



**IBM**

**International Technical Support Centers**

**SQL/DS RELEASE 3.5 USAGE GUIDE**

GG24-3049-00

SQL/DS Release 3.5  
Usage Guide

Document Number GG24-3049

April 15th, 1986

International Technical Support Center  
Boeblingen

## First Edition (April 1986)

This edition applies to Version 1 Release 3.5 of the Structured Query System/Data System (Program Number 5748-XXJ), for use with VM/SP and VSE/SP.

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM program product in this document is not intended to state or imply that only IBM's program product may be used. Any functionally equivalent program may be used instead.

The information contained in this document has not been submitted to any formal IBM test and is distributed on an 'As Is' basis without any warranty either expressed or implied. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer's ability to evaluate and integrate them into the customer's operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

No performance data may be abstracted or reproduced and given to non-IBM personnel without prior written approval by Business Practices.

Publications are not stocked at the address given below. Requests for IBM publications should be made to your IBM representative or to the IBM branch office serving your locality.

A form for reader's comments is provided at the back of this publication. If the form has been removed, comments may be addressed to IBM International Technical Support Center, Dept. 3616, Bldg. 7030-75, Schoenaicher Strasse 220, 7030 Boeblingen, West Germany. IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

(c) Copyright International Business Machines Corporation 1986

## ABSTRACT

This document serves as a guide to SQL/DS Release 3.5 users. It is intended to be used by account SE's and product specialists helping customers develop recovery/backup procedures in an SQL/DS Release 3.5 environment.

This document covers a summary of the new facilities in Release 3.5, the installation requirements, guidelines for the usage of the backup/recovery facilities, and information available for problem determination and diagnosis.

DASYS

(60 pages)



The performance data contained in this document was obtained in a controlled environment based on the use of specific data and is presented only to illustrate techniques and procedures to assist IBM personnel to better understand IBM products. The results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data in their specific environment.

### HOW TO USE THIS PUBLICATION

This document covers the functions, facilities and usage aspects of SQL/DS Release 3.5.

The emphasis of this document is in chapter 3 which describes the usage of the new facilities in Release 3.5. The second chapter covers the installation of Release 3.5. It is not intended to replace the installation guide for Release 3.5 but rather act as a supplement to the information stated there.

### HOW THIS BULLETIN IS ORGANIZED

This publication comprises three major chapters:

#### **Enhancements in Release 3.5**

This chapter describes all the new functions and facilities in SQL/DS Release 3.5.

#### **Installation**

This chapter includes the installation requirements and a summary of the steps for installing SQL/DS Release 3.5.

#### **Usage Guidelines**

This chapter describes the usage of the new backup/recovery functions. The chapter comprises two main sections.

The first section describes the facilities that can be used during warm-start log recovery.

The second section covers the facilities that can be used for forward recovery. Functions / facilities that are common to both sections are not repeated here. Readers should familiarize themselves with

the new functions described in the first section before reading the second section.

## RELATED PUBLICATIONS

- SQL/Data System Operation for VM/SP, GH24-5071
- SQL/Data System Terminal User's Reference for VM/SP, SH24-5067
- SQL/Data System Installation for VM/SP, SH24-5044
- SQL/Data System Diagnosis Guide for VM/SP, SY24-5230
- SQL/Data System Diagnosis Reference for VM/SP, SY24-5232
- SQL/Data System Messages and Codes for VM/SP, SH24-5070
- SQL/Data System Planning and Administration for VM/SP, SH24-5043
- SQL/Data System Operation for VSE, GH24-5020
- SQL/Data System Terminal User's Reference for VSE, SH24-5017
- SQL/Data System Installation for VSE, SH24-5015
- SQL/Data System Data Base Services Utility for VSE, SH24-5046
- SQL/Data System Diagnosis Guide for VSE, SY24-5229
- SQL/Data System Diagnosis Reference for VSE, SY24-5231
- SQL/Data System Messages and Codes for VSE, SH24-5020
- SQL/Data System Planning and Administration for VSE, SH24-5014

## TABLE OF CONTENTS

<b>1.0 Enhancements in SQL/DS Release 3.5</b>	<b>1</b>
1.1 User Archive and Restore Capability	1
1.1.1 Performance Aspects	1
1.1.2 Invocation	2
1.1.3 Special Considerations	3
1.2 Archiving the Log	3
1.2.1 Performance Aspects	4
1.2.2 Invocation of the LOG ARCHIVE Function	6
1.2.2.1 At SQL/DS Shutdown Time	6
1.2.2.2 Operator Command (LARCHIVE) While SQL/DS is Running	7
1.2.2.3 When ARCHPCT is Exceeded	7
1.2.2.4 When Performing a Data Base Archive	7
1.2.2.5 When Restoring the Data Base	7
1.2.3 Invocation of the Log Restore Function	8
1.2.4 Special Considerations	9
1.2.4.1 Keeping Track of the Log Tapes	9
1.2.4.2 Adding DBSPACE or DBEXTENT	10
1.2.4.3 Checkpoint Before the Log Archive	10
1.2.4.4 Log Continuity	11
1.3 Directory Verification	11
1.3.1 Performance Aspects	12
1.3.2 Invocation of Directory Verify Function	12
1.4 Diagnostic Support for Filtered Log Recovery	13
1.4.1 Where the Output is Displayed	14
1.4.2 Output Messages	15
1.5 Filtered Log Recovery and Data Base Access	15
1.5.1 Invocation	16
1.5.2 Commands Syntax and Control Keywords	16
1.5.2.1 BYPASS UNDO WORK	16
1.5.2.2 ROLLBACK COMMITTED WORK	17
1.5.2.3 DISABLE DBSPACE	18
1.5.2.4 ENABLE DBSPACE	18
1.5.3 Rules for Control Keywords	19
1.5.4 Special Considerations	20
1.6 Tape Blocking Archive and Restore (VSE only)	20
1.7 Miscellaneous	21
1.7.1 SQLRMEND EXEC to Drop the Resource Manager Code	21
1.7.2 SQLLEVEL EXEC to Identify the Release Level of SQL/DS	21
<b>2.0 SQL/DS Release 3.5 Installation</b>	<b>23</b>
2.1 Resources Required	23
2.1.1 Minidisk Requirements (VM/SP)	23
2.1.2 Disk Space Requirements (VSE/SP)	24
2.1.3 Discontiguous Saved Segment (VM/SP)	25
2.2 Installation	25
2.2.1 Installing SQL/DS Release 3.5 in VM/SP Environment	25
2.2.2 Installing SQL/DS Release 3.5 in VSE/SP Environment	27
<b>3.0 Usage Guidelines</b>	<b>29</b>
3.1 Recovery after System Failures	29



3.1.1	Interpreting Diagnostic Display Information . . . . .	29
3.1.1.1	ARI126E DBSS TERMINATION DURING REDO PROCESSING . . . . .	30
3.1.1.2	ARI126E DBSS TERMINATION DURING UNDO PROCESSING . . . . .	30
3.1.1.3	ARI126E DBSS TERMINATION DURING FORWARD PROCESSING . . . . .	31
3.1.1.4	ARI126E DBSS TERMINATION DURING ROLLBACK PROCESSING . . . . .	31
3.1.2	Filtered Log Recovery and Data Base Access . . . . .	31
3.1.2.1	BYPASS UNDO WORK . . . . .	32
3.1.2.2	ROLLBACK COMMITTED WORK . . . . .	33
3.1.2.3	DISABLE DBSPACE . . . . .	34
3.1.2.4	ENABLE DBSPACE . . . . .	36
3.1.2.5	DROP DBSPACE and DROP TABLE Considerations . . . . .	36
3.1.2.6	Summary Messages . . . . .	36
3.1.2.7	Examples . . . . .	38
3.1.2.8	Guidelines . . . . .	39
3.2	Forward Recovery . . . . .	42
3.2.1	Data Base Recovery . . . . .	42
3.2.1.1	Directory Verify . . . . .	42
3.2.1.2	SQL/DS Archive / Restore . . . . .	47
3.2.1.3	User Archive / Restore . . . . .	47
3.2.2	Log Recovery . . . . .	47
3.2.2.1	Log Archive / Restore . . . . .	48
3.2.2.2	Filtered Log Recovery . . . . .	50
3.2.3	Special Considerations . . . . .	52
3.2.3.1	Restore Fails . . . . .	53
3.2.3.2	Log Recovery Fails . . . . .	54
3.2.3.3	Error Persists . . . . .	54
3.2.3.4	Summary of Forward Recovery Actions . . . . .	56
<b>Appendix A. Performance Aspects of Release 3.5 Functions . . . . .</b>		<b>57</b>
<b>Index . . . . .</b>		<b>59</b>

## LIST OF ILLUSTRATIONS

Figure 1. Minidisk Requirements . . . . .	23
Figure 2. Starter Data Base Space Requirements (VSE) . . . . .	24
Figure 3. DCSS Space Requirements (VM/SP) . . . . .	25
Figure 4. LUWs with Dependencies Example . . . . .	39
Figure 5. Directory Verify Example . . . . .	44
Figure 6. Performance Aspects . . . . .	57



## 1.0 ENHANCEMENTS IN SQL/DS RELEASE 3.5

This chapter describes the new functions and facilities in SQL/DS Release 3.5 relative to the previous release (Release 3). The main focus of the enhancements in this release is on the backup and recovery areas to improve the reliability, availability and serviceability of SQL/DS.

### 1.1 USER ARCHIVE AND RESTORE CAPABILITY

The objective of this facility is to allow the users to take advantage of the backup/restore facilities offered by VM or VSE. This may improve backup/restore performance, especially by allowing parallel backups/restores of the directory and the data extents.

SQL/DS Release 3.5 allows you to use non-SQL/DS facilities for your data base archives and restores. Data base archives/restores done using non-SQL/DS facilities are called **User Archives / Restores**. The facilities that can be used include, but are not restricted to:

- **VMBACKUP** or **VM/SP DDR** utility  
in a VM/SP environment
- **VSAM Backup/Restore** utility  
in a VSE environment

User archives and restores can be used for the directory and DBEXTENTS but not for the log.

#### 1.1.1 PERFORMANCE ASPECTS

The time taken for archive and restore using either a user or an SQL/DS facility depends on the data base utilization. SQL ARCHIVE does not copy completely empty groups of 512 data pages to tape. If the data base is more than 80% full (DBEXTENT pages allocated), the performance improvement using user archive / restore should be significant. The lower the utilization, the less advantageous the user archive will be. If the utilization is low enough, an SQL/DS archive will outperform the user archive.

Another performance advantage using user archives is the ability to restore / archive different minidisks (VM) or VSAM clusters (VSE) concurrently:

- **In a VM/SP environment using DDR**, you can have a number of virtual machines running concurrently, each using DDR to back up or restore a subset of the minidisks that comprise the data base.
- **In a VSE environment using VSAM Backup/Restore**, you can have multiple partitions running concurrently, each using VSAM Backup/Restore to back up a subset of VSAM clusters that comprise the data base.

With both DDR and VSAM Backup/Restore, it is also possible to specify alternate tape drives. This can reduce the total time required to do an archive, as it eliminates tape rewind/unload and operator intervention delays. This parallel processing can greatly reduce the total time required to archive/restore the data base relative to archiving/restoring it serially.

### 1.1.2 INVOCATION

To invoke a user archive, use the UARCHIVE parameter with the SLENDD command.

#### SLENDD UARCHIVE

When the command is issued, SQL/DS will :

1. Let all logical units of work end.
2. Write information to the log to indicate that a user archive will be taken.
3. Terminate with a message to remind the operator to take an archive of the directory and DBEXTENTS.
4. The operator should then backup the data base.

The next time SQL/DS is warm-started, it will prompt the operator as to whether the user-archive was executed successfully.

If the response from the operator is 'Y' (yes), SQL/DS indicates in a log record that the user archive was completed and reclaims the log space.

If the response from the operator is 'N' (no), SQL/DS prompts the operator whether he wants to reissue the user archive or to continue processing.

If the user decides to take the user archive, SQL/DS will terminate without writing any record to the log because the last one is still in effect.

If the user decides to continue without taking the user archive, SQL/DS will write a record to the log indicating that no user archive was done. In this case, the log space is not freed.

To restore the data base, first restore the directory and DBEXTENTS from tape through the user restore facilities (SQL/DS has to be shutdown). Then start-up SQL/DS with STARTUP=U. SQL/DS will prompt the operator whether the data base was restored successfully.

If the response is 'Y' (the user restore was done successfully) and,

- if LOGMODE=A, SQL/DS will apply all changes in the log.
- if LOGMODE=L, SQL/DS will take an archive of the log and continue to restore the log archive tapes associated with the user archive.

If the response is 'N', SQL/DS will terminate. The operator can then do the user restore.

### 1.1.3 SPECIAL CONSIDERATIONS

You can switch between SQL archives and user archives as often as you like. However, following are some considerations when you are using user archive/restore :

1. When you need to migrate a data base between VM/SP and VSE, SQL/DS archive / restore must be used.
2. When an on-line archive must be performed to reclaim the log space while users are still accessing the data base, an SQL archive must be used. This can be an SQL data base archive or an SQL log archive. (For details on log archiving see below.)
3. You have to warm-start SQL/DS (STARTUP=W) after an SQLEND UARCHIVE. If you have to restore your data base immediately after taking the user archive, SQL/DS should be started with STARTUP=W and not with STARTUP=U.

## 1.2 ARCHIVING THE LOG

Log archiving provides you with the facility to archive the log to reclaim log space. The log archives can be used as a backup for forward recovery. Taking a log archive will be faster than the data base archive because the log is usually smaller than the DBEXTENTS and the directory.

In previous releases, when SQL/DS was running with archiving enabled and the log space filled to a user specified point (ARCHPCT), an online data base archive had to be performed to reclaim the log space. The

performance of the users accessing the system was adversely affected. With the SQL/DS Release 3.5 log archiving facility, only the log needs to be archived. This reduces the time during which SQL users will be affected.

Another advantage of archiving logs is the ability to restore your data base to the latest state even if the most current copy of your data base archive is damaged. This can be accomplished by restoring a back-level data base archive copy and applying all changes recorded in the log archives.

Following shows an example of restoring a back-level data base archive with subsequent log archives :

```
Log Archive 1
Data Base Archive I
Log Archive 2
Log Archive 3
Data Base Archive II
Log Archive 4
Log Archive 5
```

In this example, two data base archives were taken. Log archive 1 is triggered by data base archive I. Log archive 3 is triggered by data base archive II.

Suppose you need to restore your data base from an archive copy and data base archive II is damaged. With LOGMODE=L, you can restore data base archive I and apply log archives 2, 3, 4, 5 to bring the data base to the latest state.

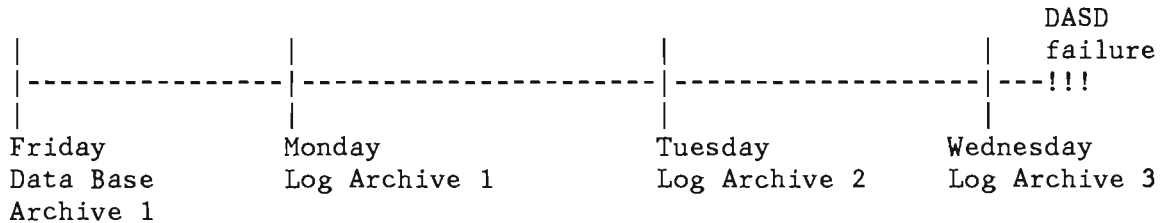
In a VM/SP environment, you can archive the log to a minidisk by changing the FILEDEF statement in SQLSTART. However, if you want to assign different log archives to different minidisks, you have to change the FILEDEF statement each time to reflect the different minidisk extents, and for that you need to shutdown the data base machine.

### 1.2.1 PERFORMANCE ASPECTS

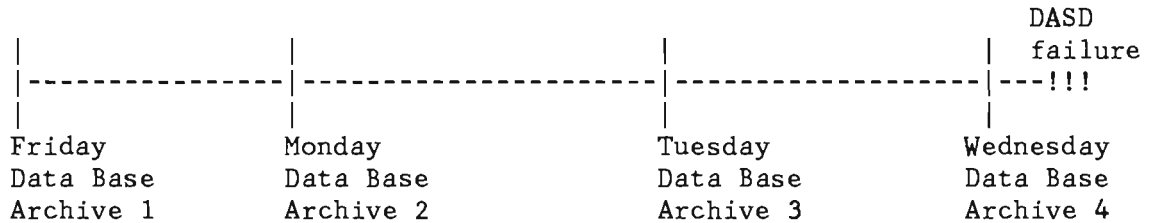
1. The time required to archive the log is directly related to the number of entries in the log. For more details please refer to Appendix A.
2. Log archive time will be the same regardless of whether or not dual logging has been specified. This is because only one log copy is archived in either case. However, the time required to restore the log will be longer in the dual logging case because the archived copy is restored to both logs.

3. Archiving the log will usually be faster than archiving the data base because the log size is usually smaller than the directory and DBEXTENTS.
4. The restore will usually take longer. Consider the following two situations with LOGMODE=L and LOGMODE=A.

CASE I LOGMODE = L



CASE II LOGMODE = A



In CASE I, you need to restore data base archive 1 and apply log archives 1, 2, 3 and the current log to bring the data base to the latest state. When restoring the logs, SQL/DS has to redo all changes recorded in the log tapes. The time taken for the restore depends very much on the amount of activity since Friday - it will be as though the logical units of work are re-executed.

In CASE II, you only need to restore the data base archive taken on Wednesday - data base archive 4. SQL/DS needs to redo the changes recorded in the current log. The time taken will be much shorter than restoring the log archives in CASE I.

5. SQL/DS schedules a checkpoint before the log archive starts. The checkpoint has to wait until no logical unit of work is active. No other LUW will be started until the checkpoint has completed. All SQL/DS activity continues while the log archive is taken. However, if a log overflow condition is encountered, all SQL/DS activity will stop until the log archive is completed. Please refer to "Checkpoint Before the Log Archive" on page 10 for more details.



## 1.2.2 INVOCATION OF THE LOG ARCHIVE FUNCTION

You have to enable log archiving by running with LOGMODE=L.

If you switch to LOGMODE=L from a different LOGMODE, SQL/DS ensures that a data base archive was created previously. A data base archive is required before you can run LOGMODE=L because the log contains only changes. If a data base archive has not been taken, SQL/DS will terminate with a message to inform the user to take a data base archive.

**Note:** You also have to take a data base archive in order to run LOGMODE=L after you performed a COLDLOG.

Once you are running with LOGMODE=L, the log archiving function can be invoked implicitly or explicitly via operator commands.

During archiving, SQL/DS displays control information to assist the external labelling of archive tapes.

The control information describes whether it is a log or data base archive, and the date and time the archive is taken. This information will assist the operators in externally labelling the tapes. The information is maintained by SQL/DS in the history area residing on the log extent.

For each log archive, SQL/DS displays the label information once and only one record is written in the history area. Hence the user has to maintain his own information for log archives that consist of multiple tape volumes.

The following messages are displayed before each archive is taken. The messages identify the type of archive (DATA BASE ARCHIVE or LOG ARCHIVE), the time and date (based on processor's time-of-day clock) when the archive is taken.

---

```
ARI257I NEW LUWS AND THE LOG ARCHIVE PROCESS MAY NOT CONTINUE
ARI257I UNTIL 0 ACTIVE LUWS COMPLETE
ARI293I ARCHIVE STARTING
ARI239I EXTERNAL LABELING OF THIS ARCHIVE IS:
ARI239I     TYPE:      LOG ARCHIVE
ARI239I     TIMESTAMP: 02-17-86 10:57:58
ARI299A READY ARCHIVE OUTPUT VOLUME. REPLY CUU
```

---

### 1.2.2.1 At SQL/DS Shutdown Time

You can initiate a log archive during shutdown when SQL/DS is running in multiple user mode. This is invoked by issuing an SQLEND LARCHIVE

command. All SQL/DS activity will be terminated before the log archive is taken.

#### 1.2.2.2 Operator Command (LARCHIVE) While SQL/DS is Running

On-line log archiving can be invoked in multiple user mode via the LARCHIVE operator command.

#### 1.2.2.3 When ARCHPCT is Exceeded

ARCHPCT is a user specified parameter (default is 80%) which determines how full the log can become before SQL/DS invokes the archiving facility. With LOGMODE=L, when the log is filled to the specified percent a log archive will be invoked rather than a data base archive as with LOGMODE=A.

#### 1.2.2.4 When Performing a Data Base Archive

With LOGMODE=L, when a data base archive is invoked (either via ARCHIVE or SQLEND ARCHIVE commands), a log archive will be taken before archiving the data base. Log archive is taken first because a data base archive would free log space in the current log. By taking a log archive, the log continuity is maintained.

#### 1.2.2.5 When Restoring the Data Base

With LOGMODE=L, a log archive will be taken after the data base is restored but before the log tapes are applied. This is done to 'save the contents' of the current log before it is overwritten by log archives. This log archive will only be taken when there is information in the log. For example, SQL/DS will not archive the log if you have done a log archive immediately before the restore.

After the data base is restored and all log tapes are applied, SQL/DS will again ask you to archive the log. An example of the events :

---

```
restore the data base
archive the log
apply all log tapes
archive the log
```

---

### 1.2.3 INVOCATION OF THE LOG RESTORE FUNCTION

SQL/DS provides facilities to help the operators ensure that the correct tapes are mounted when restoring the data base.

#### 1. Checking the currency of data base archive used for restore:

When restoring the data base using a data base archive copy, the operator will be notified if the tape is not the most recent data base archive taken. The operator will be able to correct the action if he had mounted the wrong tape, or ignore the message if he wanted to restore a back-level data base archive.

The following messages are displayed **after** you have restored a back-level data base archive.

---

```
ARI289I RESTORING DIRECTORY DISK
ARI290I RESTORING DATA DISK
ARI291I SYSTEM RESTORE FROM DIRECTORY DISK AND DATA DISK
ARI291I OF DATA BASE ARCHIVE COMPLETED
ARI251D THE DATA BASE ARCHIVE CURRENTLY MOUNTED IS NOT
ARI251D THE MOST RECENT DATA BASE ARCHIVE KNOWN TO SQL/DS.
ARI251D REPLY 'IGNORE' TO CONTINUE, OR 'CANCEL' TO RESTART
ARI251D THE RESTORE PROCESS WITH THE MOST RECENT
ARI251D DATA BASE ARCHIVE.
```

---

#### 2. Checking log tapes used when restoring the tape archives:

When applying the log tapes after restoring the data base archive, the operator is informed of the sequence of log tapes associated with this data base archive. He is also informed of the next tape to mount if he is applying a sequence of log archives. SQL/DS will also notify the operator if the proper tape is not mounted.

The following messages are displayed after the data base archive is restored. SQL/DS identifies the set of log archives associated with the data base you restored and the next log archive to apply.

---

ARI260I THE RESTORE SET FOR THIS DATA BASE ARCHIVE IS AS FOLLOWS:  
ARI238I DATA BASE ARCHIVE 02-17-86 10:17:13  
ARI261I LOG ARCHIVE 02-17-86 10:25:53  
ARI261I LOG ARCHIVE 02-17-86 10:49:07  
ARI261I LOG ARCHIVE 02-17-86 10:57:58  
ARI239I EXTERNAL LABELING OF THIS ARCHIVE IS:  
ARI239I TYPE: LOG ARCHIVE  
ARI239I TIMESTAMP: 02-17-86 10:25:53  
ARI250D THE ABOVE INFORMATION DESCRIBES THE NEXT LOG ARCHIVE  
ARI250D TO BE USED IN THE RESTORE PROCESS.  
ARI250D REPLY 'CONTINUE' TO RESTORE THIS LOG ARCHIVE, OR  
ARI250D 'STOP SYSTEM' TO INTERRUPT THIS RESTORE PROCESS, OR  
ARI250D 'END RESTORE' TO END THIS RESTORE PROCESS AND SQL/DS.

---

- If the reply is 'CONTINUE', SQL/DS will continue to restore the log archive.
- If the reply is 'END RESTORE', no more log tapes will be restored. This means that the log continuity is broken. Once the log continuity is broken, you cannot restore subsequent log archives in this restore set.

END RESTORE is useful if you want to discard a log archive that contains incorrect updates. You restore the data base and before restoring this log archive, you can enter END RESTORE to discard subsequent log archives.

- If the reply is 'STOP SYSTEM', you can temporarily stop the restore process. This is useful if you are uncertain whether to continue or to stop the restore process; or if you want to specify another EXTEND file to be used with Filtered Log Recovery. The next time you start up the data base machine, it will continue to prompt you with the same messages.

## 1.2.4 SPECIAL CONSIDERATIONS

Additional considerations when running LOGMODE=L :

### 1.2.4.1 Keeping Track of the Log Tapes

You should maintain a 'restore list' based on the control information supplied by SQL/DS for log archives and data base archives. This is extremely important if single log archives span several tapes.

#### 1.2.4.2 Adding DBSPACE or DBEXTENT

You also have to ensure that the log and directory are synchronized. You always have to take a data base archive after you have added DBSPACES or DBEXTENTS. These changes are recorded in the directory and not recorded in the log(s). Data base archives taken before the changes will not be synchronized with the log(s) - for example, the log might contain updates to those new DBSPACES.

#### 1.2.4.3 Checkpoint Before the Log Archive

SQL/DS always schedules a checkpoint immediately before the on-line log archive. This checkpoint is taken only when there is no active logical unit of work. The checkpoint agent requests a data base (DB) lock which allows the existing active logical units of work to finish and prevents any new access to the data base.

SQL/DS displays a message giving the number of active logical units of work when a log archive is requested. When all the active logical units of work are completed, SQL/DS takes the checkpoint and creates a log archive. During the creation of the log archive, new logical units of work can resume access to the data base.

The checkpoint will be delayed if there are long running logical units of work. In this case, any new LUWs will experience a long delay because the checkpoint agent is waiting for the long running LUW and the new LUWs are waiting for the checkpoint to complete. You can use the FORCE command to end the long running logical unit of work. You can use the SHOW LOCK command to find out which logical units of work are delaying the checkpoint. You have to use the SHOW LOCK command from the SQL operator console; if it is issued from ISQL, it is considered a new logical unit of work. Hence, it will also be waiting for the checkpoint to complete.

Normally after the checkpoint completion and during the creation of the log archive, new logical units of work can resume access to the data base. This is not true if any of the following conditions occurred:

- a short-on-storage condition for a storage pool.
- SQL/DS log becomes full.
- COMMIT WORK or ROLLBACK WORK is issued by a logical unit of work that updated data in a nonrecoverable storage pool.
- if the log archive is invoked by the ARCHIVE command while running with LOGMODE=L. The user can resume access to the data base after the log archive is taken and while the data base archive is running.

All logical units of work wait until the log archive is completed in these cases.

#### 1.2.4.4 Log Continuity

You have to ensure that log continuity is maintained. Log continuity is disrupted by:

- COLDLOG
- switch of logmodes
- END RESTORE during recovery
- log re-configuration

Please refer to "Log Archive / Restore" on page 48 for details on using a back-level archive copy to restore your data base.

SQL/DS uses a **history area** to keep track of information required for recovery actions. This history area resides on the last physical page of the log. One record is created for each COLDLOG, data base archive, log archive, restore, switching of log modes. SQL/DS knows the log archives that are associated with a data base archive based on information in the history area.

If the history area is erased (for example, CMS FORMAT RESERVE command), SQL/DS cannot identify the log archives associated with a particular data base archive. In this case, you can restore the data base archive but you cannot apply subsequent log archives.

### 1.3 DIRECTORY VERIFICATION

The objective of the Directory Verify function is to inform the user of the discrepancies in the directory and hence allow the user to avoid archiving an inconsistent directory.

The directory contains the mapping information between logical and physical pages, allocation of the physical pages and other internal control information for the SQL/DS data base. Directory information can become inconsistent. This can happen through:

1. incorrect use of some administration facilities (for example, ADD DBSPACE, ADD DBEXTENT ...), or
2. possible errors in SQL/DS code.

It is very rare for a directory to become corrupted, however, if not known in time, full recovery may be very cumbersome.

In prior releases, discrepancies in the directory, for example, inconsistencies between logical and physical pages allocated to a DBSPACE, were normally not known to the users until SQL/DS ended

abnormally. When this happened, the user had to restore the data base from an archive copy to make the data base operational. However, if the discrepancies were not detected before the archive was taken, then they were recorded in the archived data base. When this archive copy was used to restore the data base, the problem still existed. In a situation like this, the user would have to go back to a prior archive copy that did not contain the problem. This definitely resulted in loss of data since the log could not be used to bring the data base back to the point before the failure. The process required to recover the data base up to the point before the failure might be very difficult and time consuming.

In Release 3.5 the Directory Verify function will check for each DBSPACE page (disabled DBSPACES will also be included) if:

- free class entries in the page map table are valid
- the physical pages allocated belong to the correct storage pool
- real page numbers are valid
- the bit map entry indicates that the page is allocated

Messages will be displayed on the console if there is any discrepancy. The user still has to restore the data base if there is any discrepancy in the directory. But assuming that this archive copy was verified, then there is no need to go to a back-level copy.

### 1.3.1 PERFORMANCE ASPECTS

The time taken for Directory Verify processing is proportional to the number of allocated DBEXTENT pages in the data base. For more details please refer to Appendix A.

### 1.3.2 INVOCATION OF DIRECTORY VERIFY FUNCTION

The Directory Verify function can be invoked during multiple user mode shutdown with and without archiving.

---

SQLEND	NORMAL	DVERIFY
	ARCHIVE	DVERIFY
	UARCHIVE	DVERIFY
	LARCHIVE	DVERIFY
	DVERIFY	

---

If you specify data base archive with the DVERIFY parameter and an error is detected in the directory, the data base archive will / should not be taken. However, if SQL/DS is started with LOGMODE=L and you request an archive, then the log archive will be taken even though there is an inconsistency in the directory. The reason is that the inconsistency is in the directory and not in the log and the changes recorded in the log are required for recovery.

- If UARCHIVE is specified and there is an inconsistency, SQL/DS will display a warning message that a user archive should not be taken and it will end with a return code of 516. You should not take any archive of this data base. If you do, the archived data base cannot be used for future restore.
- If ARCHIVE is specified and an inconsistency is detected, SQL/DS will not take the data base archive.

Following is a summary of steps taken by SQL/DS if it detects an inconsistency during Directory Verify and LOGMODE=L :

---

SQLEND DVERIFY	display messages on problem pages
SQLEND ARCHIVE DVERIFY	archive the log display messages on problem pages data base archive will not be taken
SQLEND UARCHIVE DVERIFY	archive the log display messages on problem pages warning message: user archive should not be taken
SQLEND LARCHIVE DVERIFY	display messages on problem pages archive the log (in this sequence)

---

With other LOGMODEs, only the messages are displayed; no log archive will be taken.

#### 1.4 DIAGNOSTIC SUPPORT FOR FILTERED LOG RECOVERY

The primary objective of Diagnostic Support for Filtered Log Recovery is to provide appropriate diagnostic display information on certain SQL/DS abnormal terminations. Based on the information provided, SQL/DS system programmers can determine how to use the **Filtered Log Recovery and Data Base Access** facility to make the data base operational as soon as possible.



The second objective of Diagnostic Support for Filtered Log Recovery is to provide diagnostic information for IBM Support Personnel.

Diagnostic display information is issued only for SQL/DS abnormal terminations which occur during DBSS processing. This information is provided when a failure occurs while DBSS is performing one of the following activities :

- forward (normal) processing accessing PUBLIC or PRIVATE DBSPACES.
- a ROLLBACK process accessing PUBLIC or PRIVATE DBSPACES.
- log recovery during an SQL/DS warm-start or during a data base restore from an archive. Log records are used to either back out (UNDO) uncommitted data base changes or to re-apply (REDO) committed data base updates.

Diagnostic display information for SQL/DS abnormal terminations which occur in RDS processing or DBSS processing that is not accessing PUBLIC or PRIVATE DBSPACES, is not provided because :

- Filtered Log Recovery and Data Base Access cannot be used as the recovery actions for these types of errors.
- The type of errors are **not** caused by :
  - bad (or corrupted) data in the data base.
  - bad data in the SQL/DS log records.
  - data in the data base cannot be handled by SQL/DS.
  - log records cannot be correctly processed by SQL/DS.
  - SQL/DS log records have become out-of-sync with the data in the data base (for example, log record wants to delete a row that does not exist in the data base).

#### 1.4.1 WHERE THE OUTPUT IS DISPLAYED

The output of the diagnostic information is displayed :

- for VM/SP, on the SQL/DS data base machine console (SQLSTART EXEC always spools the console output).
- for VSE, on the device(s) specified by the SQL/DS initialization parameter DSPLYDEV (SYSLSST file and/or the system console).

## 1.4.2 OUTPUT MESSAGES

A new message ARI126E will be displayed as the first line of output followed by diagnostic information. The messages displayed can be used in connection with the Filtered Log Recovery commands. For details refer to "Interpreting Diagnostic Display Information" on page 29.

## 1.5 FILTERED LOG RECOVERY AND DATA BASE ACCESS

The objective of this facility is to allow the user to bring up SQL/DS after start-up failures during warm-start or recovery.

Certain DBSS errors cause the data base machine to terminate abnormally. When restarting the data base machine after the abnormal end, normal recovery processing for a warm-start or log recovery after restoring an archive may not be successful.

The Filtered Log Recovery facility of Release 3.5 alleviates this problem. It provides facilities for users:

- to bypass UNDO processing during log recovery.
- to rollback work that was committed to the data base.
- to disable DBSPACES that might cause SQL/DS termination or contain inconsistent data.

Usually the operations that caused SQL/DS to terminate abnormally are recorded in the log. Filtered Log Recovery allows you to **bypass these operations**. Bypassing the operations causes portions of the data to be logically inconsistent. But possibly the greater portion of the data base is ready for use. These inconsistencies have to be 'manually' removed by the user.

The commands available to process the log selectively are **BYPASS UNDO WORK** and **ROLLBACK COMMITTED WORK**. For details on these commands please refer to "Filtered Log Recovery and Data Base Access" on page 31.

Filtered Log Recovery and Data Base Access also includes commands that allow you to disable and enable DBSPACES. When data inconsistencies are created because of the **BYPASS UNDO WORK** or **ROLLBACK COMMITTED WORK**, you can use the **DISABLE DBSPACE** command to prevent any access to the DBSPACE that contains inconsistent data. The **DISABLE DBSPACE** command can also be used to disable DBSPACES that have internal problems; for example, if users receive error messages whenever they try to access a certain DBSPACE. The diagnostic display information support will help you to determine which DBSPACE is to be disabled (please refer to "Interpreting Diagnostic Display Information" on page 29 for more detail).

**Note:** Filtered Log Recovery can also be used to rollback committed user errors. However, you should use it only as a last resort. Attempting this is especially dangerous in environments where data is shared by multiple users, each of them doing their own updates. If you really have to use it, use it **with great caution**. Please refer to "LUWs with Dependencies" on page 38 for examples of rolling back logical units of work that are dependent on others.

**Note:** The problem described above will, of course, also exist if committed work has to be rolled-back to recover from DBSS failures, but this may be the only way to make the data base operational again. In any case, it is the user's responsibility to fix logical data inconsistencies introduced through the use of these facilities.

### 1.5.1 INVOCATION

To invoke the Filtered Log Recovery and Data Base Access facility, you have to specify **EXTEND=Y** as initialization parameter when you are starting up the SQL/DS machine. The default is **EXTEND=N**. **EXTEND=Y** may only be specified with **STARTUP=U, R, W** (or **F** for **VSE**). Other **STARTUP** values cause SQL/DS initialization to end.

**EXTEND=Y** tells the SQL/DS start-up process to read extended start-up commands. The commands and control keywords have to be supplied in a CMS file for VM/SP and in SYSIPT for VSE.

### 1.5.2 COMMANDS SYNTAX AND CONTROL KEYWORDS

This section covers the syntax for the four commands **BYPASS UNDO WORK**, **ROLLBACK COMMITTED WORK**, **DISABLE DBSPACE** and **ENABLE DBSPACE**. The control keywords and parameters that can be used are included.

#### 1.5.2.1 BYPASS UNDO WORK

---

```
BYPASS UNDO WORK WHERE
  LUWID id id1 id2....
  ALL
  DBSPACE dbspaceno dbspaceno1 dbspaceno2....
  USERID userid userid1 userid2 ....
```

---

The control keywords are :

**LUID id id1 id2....** identifies the logical unit of work to be bypassed. To bypass more than one logical unit of work, separate the ids by blanks. You can obtain the id from the diagnostic display.

**DBSPACE dbspaceno dbspaceno1 dbspaceno2 ....** indicates that UNDO processing is to be bypassed for logical units of work that update the DBSPACE specified. Separate the dbspacenos with blanks if you want to specify more than one DBSPACE.

SQL/DS works on a logical unit of work basis. If a particular DBSPACE is specified, all logical units of work that changed data in this DBSPACE are bypassed. This includes all the changes made by these affected logical units of work on other DBSPACEs.

You cannot specify DBSPACE 1 to bypass undo processing for logical units of work that updated DBSPACE 1.

**USERID userid userid1 userid2 ....** identifies that UNDO processing is to be bypassed for logical units of work that were done by this userid. To specify more than one userid, separate them with blanks.

**ALL** indicates that all logical units of work to be undone are to be bypassed. Once you specify ALL, you cannot use other control keywords or another BYPASS UNDO WORK command in the EXTEND file.

### 1.5.2.2 ROLLBACK COMMITTED WORK

---

ROLLBACK COMMITTED WORK WHERE  
LUID id id1 id2 ....  
DBSPACE dbspaceno dbspaceno1 dbspaceno2 ....  
USERID userid userid1 userid2 ....  
DATE mm-dd-yy  
TIME hh:mm:ss  
TO

---

The control keywords are :

**LUID id id1 id2 ....** identifies the logical unit of work to be rolled-back. See above.

**DBSPACE dbspaceno dbspaceno1 dbspaceno2 ....** indicates that logical units of work which updated the specified DBSPACE are to be rolled-back. See above.

**USERID** *userid userid1 userid2 ....* identifies the logical units of work to be rolled-back for specified users. See above.

**DATE** *mm-dd-yy* indicates the date from which committed work will be rolled-back. The format of the date has to be *mm-dd-yy*. You can specify the date to rollback committed logical units of work in the current log or, if you are restoring a data base, the date when logical units of work were committed in the log archives. If **DATE** is omitted but the **TIME** control keyword is specified, default is the current date.

**TIME** *hh:mm:ss* indicates the time from which logical units of work are to be rolled-back. The format *hh:mm:ss* is required. If **TIME** is omitted but the **DATE** control keyword is specified, default is 00:00:01. This is the time the log record for the LUW was written (LUW start time) and not the time when the LUW was committed.

**TO** this parameter is necessary if you want to specify a range for **TIME** and **DATE**. You have to specify a starting **DATE / TIME** before a **TO** and an ending **DATE / TIME**.

### 1.5.2.3 DISABLE DBSPACE

---

**DISABLE DBSPACE** *dbspaceno dbspaceno1 dbspaceno2 ....*

---

**DBSPACE** *dbspaceno dbspaceno1 dbspaceno2 ....* identifies the **DBSPACES** to be disabled. **DBSPACE 1** cannot be disabled.

### 1.5.2.4 ENABLE DBSPACE

---

**ENABLE DBSPACE** *dbspaceno dbspaceno1 dbspaceno2 ....*

---

**DBSPACE** *dbspaceno dbspaceno1 dbspaceno2 ....* identifies the **DBSPACES** to be enabled.

### 1.5.3 RULES FOR CONTROL KEYWORDS

1. Except for ALL which cannot be specified with other parameters, you can specify more than one control keyword for the BYPASS UNDO WORK / ROLLBACK COMMITTED WORK commands. They can be specified in any order.
2. Except for LUWID, the control keywords are ANDed, the parameters within a control keyword are ORed.
3. LUWID is always ORed to the other control keywords you specified.

For example, the following BYPASS UNDO WORK command will bypass the UNDO processing for logical units of work with LUWID 2BE and also all LUWs created by user SQLUSER1 or SQLUSER2 which caused changes in DBSPACES 5 or 6 or 7.

---

```
BYPASS UNDO WORK WHERE
  DBSPACE 5 6 7
  LUWID 2BE
  USERID SQLUSER1 SQLUSER2
```

---

In the following example, the ROLLBACK COMMITTED WORK command causes logical units of work that are created by user SQLUSER1 on DBSPACES 5 or 6 or 7 from 10:00 a.m. to 2:00 p.m. on 03-01-86 to be rolled-back.

---

```
ROLLBACK COMMITTED WORK WHERE
  DBSPACE 5 6 7
  USERID SQLUSER1
  DATE 03-01-86
  TIME 10:00:00
  TO
  DATE 03-01-86
  TIME 14:00:00
```

---

The characteristics of the input control file for the extended start-up commands are described below :

- All commands must be in upper case.
- Lines beginning with an asterisk (\*) in column one are taken as comments. Blank lines are allowed in the control file.
- Each command or control keyword must be on a line by itself.
- You can type your commands anywhere between columns 1 - 72.

#### In the VM/SP environment:

- The ddname in the FILEDEF command must be ARIEXTND.

- The EXTEND input file must be 80 byte fixed length records.

**In the VSE environment:** If SQL/DS is invoked with EXTEND=Y to run an application in single user mode, you have to ensure that the application does not use SYSIPT.

#### 1.5.4 SPECIAL CONSIDERATIONS

Following are some additional considerations when Filtered Log Recovery is used :

1. SQL/DS reads the extended commands and checks the log for logical units of work that satisfied the control keywords and parameters. It checks also the logical units of work dependencies - LUWs which updated the same tables as those LUWs to be rolled-back or bypassed.
2. SQL/DS does not check the logical units of work dependencies across different log archives.
3. If there are several failures during the log recovery, you might have to start up SQL/DS several times to gather a pattern of the failures.

#### 1.6 TAPE BLOCKING ARCHIVE AND RESTORE (VSE ONLY)

Prior releases of SQL/DS used 4K blocks for tape I/O during data base archive and restore.

The new blocking facility in SQL/DS Release 3.5 improves the performance of the VSE archive / restore functions by blocking together multiple 4K byte blocks into a 28K block for tape I/O.

It uses less processor time, less tape space and hence might reduce the tape volumes. The total archiving time is about the same. However, there would be time savings in tape rewinding and tape handling if tape volumes are reduced.

This new facility is used automatically on output to tape during archive. No parameter is needed to invoke it. When restoring from the tape, the VSE data base restore facility will handle either the 4K blocksize (archive taken prior to SQL/DS Release 3.5) or the 28K blocksize (archive taken using SQL/DS Release 3.5). Additional virtual and real storage will be required due to the larger blocksize (the data transfer between DASD and tape for data base archive and restore).

**Note:** When migrating to VM/SP SQL/DS, SQL/DS Release 3.5 on VM/SP can restore a VSE data base archive input regardless of whether the blocksize is 4K or 28K.

## 1.7 MISCELLANEOUS

Miscellaneous enhancements in SQL/DS Release 3.5 that cannot be classified under the above areas:

### 1.7.1 SQLRMEND EXEC TO DROP THE RESOURCE MANAGER CODE

SQLRMEND EXEC allows the storage occupied by the resource manager code and the control blocks to be freed from the DMSFREE area of the user's virtual machine. If DCSS for the resource manager is used then only the control blocks are freed. This EXEC can be useful for the user who executes programs that access the SQL/DS data base (and hence the resource manager code and control blocks are established) from a CMS EXEC and after which, he needs to execute a program that does not access the SQL/DS data base. In this case, SQLRMEND can be executed to make the storage available to the non-SQL programs. Otherwise, the storage will only be freed after the EXEC has ended.

### 1.7.2 SQLLEVEL EXEC TO IDENTIFY THE RELEASE LEVEL OF SQL/DS

This EXEC will provide the version, release and modification level of SQL/DS on which you are working. Please note that the values provided are incorporated in the EXEC itself. The release level given will always be Version 1 Release 3 Modification 5.





## 2.0 SQL/DS RELEASE 3.5 INSTALLATION

The intent of this chapter is to describe the changes in installation planning and requirements from Release 3 and a brief description of the steps for Release 3.5 installation. The detailed installation steps and procedures are documented in the SQL/DS Release 3.5 installation guide and the program directory delivered with the Release 3.5 tape.

### 2.1 RESOURCES REQUIRED

This section covers the installation requirements including disk space, discontinuous save segment allocations (VM/SP) and changes in real storage requirements from Release 3.

#### 2.1.1 MINIDISK REQUIREMENTS (VM/SP)

SQLDBA machine System minidisk	Block size	Virtual Address	Access mode	3330 Cyls	3340 Cyls	3350 Cyls	3375 Cyls	3380 Cyls	FB-512 Blocks
Work Disk	1024	191	A	23	57	10	16	10	8100
Service Disk	1024	193	V	81	200	35	56	35	28350
Production Disk	1024	195	Q	27	63	12	16	10	10530

SQL/DS Starter Data Base	Block Size	Virtual Address	3330 Cyls	3340 Cyls	3350 Cyls	3375 Cyls	3380 Cyls	FB-512 Blocks
Directory disk	512	200	64	156	29	47	34	23490
Log disk	4096	201	22	50	10	13	12	9600
Data disk (recommended)	4096	202	212	480	96	125	119	92160
(Minimum)	4096	202	66	171	30	51	36	24300

Figure 1. Minidisk Requirements

**Note:** We have installed Release 3.5 on 3370 disks. Following are some of the our findings:

1. The space allocated for 195 gives 30% free space for future expansion.

2. The space utilization for 193 is 94%. This value is too small when you use ARISCOR to do maintenance. (The EXEC will do a copyfile with replace option for ARISQLLD LOADLIB which occupies about 6000 FB-512 blocks or about 21% of the total size allocated.) We suggest that you either enlarge the disk or erase the file ARISQLLD LOADLIB before you invoke the EXEC ARISCOR.

### 2.1.2 DISK SPACE REQUIREMENTS (VSE/SP)

Following depicts the approximate size of the library under VSE/SP Version 2. Please refer to the installation guide for the space requirements of VSE/SP Version 1.

Approximate Space Requirement : 14,228 Library Blocks

Following shows the space requirement for the starter data base.

SQL/DS Starter Data Base	VOLSER	3330 Cyls	3340 Cyls	3350 Cyls	3375 Cyls	3380 Cyls	FB-512 Blocks
Directory disk	xxxxxxx	64	156	29	47	34	23490
Log disk 1	xxxxxxx	22	50	10	13	12	9600
Data disk 1 (recommended)	xxxxxxx	212	480	96	125	119	92160
(Minimum)		66	171	30	51	36	24300

Figure 2. Starter Data Base Space Requirements (VSE)

### 2.1.3 DISCONTIGUOUS SAVED SEGMENT (VM/SP)

There is an increase in DCSS requirements :

---

SYSNAME	SYSSIZE	Number of SYSHRSG (SYSSIZE/4)	SYSPGCT (SYSSIZE/4)	Hexadecimal Origin (Load Address) Beginning at 12MB
SQLRMGR	64K	1	16	X'C0000'
SQLISQL	384K	6	96	X'C10000'
SQLSQLDS	832K	13	208	X'C70000'
SQLXRDS	832K	13	208	X'D40000'

Figure 3. DCSS Space Requirements (VM/SP)

---

## 2.2 INSTALLATION

This section gives a summary of the steps involved in installing SQL/DS Release 3.5 in both VM/SP and VSE/SP environments.

### 2.2.1 INSTALLING SQL/DS RELEASE 3.5 IN VM/SP ENVIRONMENT

This section covers the steps to install Release 3.5 in a VM/SP environment. The detailed description is given in the installation guide.

1. Planning for the installation:

Prepare the minidisk allocation documented in "Minidisk Requirements (VM/SP)" on page 23, the DMKSNT entries required for the SQL/DS Discontiguous save segments (DCSS) and the virtual machine requirements for the installation. Virtual machines (userid) required are MAINT, SQLDBA (or a userid for the data base machine) and another userid (normal VM/SP user with 191 R/W disk and 2M virtual storage) to complete the installation process.

2. Logon to userid MAINT.

3. Attach a tape drive to MAINT as 181, mount the SQL/DS Release 3.5 tape.

4. Rewind the tape.

5. Use the VMFPLC2 command to load the installation EXECs and MEMO as below:

**VMFPLC2 LOAD \* \* A (EOF 2**

6. Invoke the EXEC ARISYSIN to load the production and service minidisks:

**ARISYSIN**

7. Logoff MAINT.
8. Logon to SQLDBA (or your data base machine id).
9. Setup data base machine profile.
10. Invoke EXEC SQLDBINS to generate a data base as below:

**SQLDBINS DBNAME(databasename) STARTER(YES)**

11. Start data base machine in multiple user mode and #CP DISC.
12. Logon to userid to complete the installation process:

- a. Setup user profile.
- b. Initialize the user machine using SQLINIT EXEC as below:

**SQLINIT DBNAME(databasename)**

- c. Run sample program EXECs (SQLASMC, SQLCBLC, SQLFTN, SQLPLI) to verify the installation. Please note that you must have the necessary language compilers installed to be able to run these sample programs.
  - d. Invoke ISQL to verify the installation and change the SQLDBA password.
  - e. Issue some SELECT statements to verify the data base.
  - f. Enter EXIT to leave ISQL.
13. It is recommended that you run your SQL/DS with save segments. Use EXEC ARISAVES to save the segments and SQLGENLD to generate the bootstraps.
  14. Include the DCSSID parameter with the name of the DCSS bootstraps when you start your data base machine.

## 2.2.2 INSTALLING SQL/DS RELEASE 3.5 IN VSE/SP ENVIRONMENT

There are no major changes between SQL/DS Release 3.5 and SQL/DS Release 3 for installation.

Following gives a summary of the installation steps in a Version 2 environment. For more details, and if you are installing under Version 1, please refer to the installation guide.

1. Planning for the installation:

Choose a virtual partition with at least three megabytes. During the installation, input is from SYSIPT and the SQL/DS source library. Output is written to the operator console, SYSLST, and the SYS001 workfile.

2. Update the CICS tables for SQL/DS.
3. Define the library to be installed.
4. Install the libraries via MSHP.
5. Catalog the Base procedures.
6. Perform SQL/DS link-edit.
7. Define the VSAM master catalog if you are installing a new SQL/DS Release 3.5 data base. Skip this step if you already have a VSAM master catalog.
8. Define VSAM data sets for the data base if you are installing a new data base.
9. Generate the data base.
10. Install the data base components.
11. Grant SCHEDULE authority if the CICS application name is not DBDCICS.
12. Start the data base machine in multiple user mode.
13. Verify the installation by running ISQL and preprocessing the sample programs.

**Note:** The optional product shipped with VSE/SP Version 2 is SQL/DS Release 3. If you install SQL/DS Release 3.5 you may have to remove Release 3 from your system.



SQL/DS Release 3.5 enhances both:

- the log recovery facility
- the backup / restore facilities

There are two types of recovery facilities provided by SQL/DS:

1. recovery after a system failure (warm-start recovery)
  - log recovery
2. recovery after a DASD failure (forward recovery)
  - data base restore, **plus**
  - log recovery

In the first part of this chapter we describe the usage of the new functions which apply to log recovery during a warm-start.

The second part contains usage guidelines for forward recovery. As far as log recovery (during forward recovery) is concerned, only the changes to warm-start log recovery are described.

### 3.1 RECOVERY AFTER SYSTEM FAILURES

Recovery from system failure is automatically done by SQL/DS and requires no user intervention. During a warm-start, SQL/DS checks the log for work to be undone or redone. If an error occurs during this process, the warm-start will fail. In SQL/DS Release 3.5, facilities are available to enhance the availability of your data base: Diagnostic messages are displayed when DBSS errors occur and filtered log commands can be used to bring up the data base even if normal warm-start fails.

#### 3.1.1 INTERPRETING DIAGNOSTIC DISPLAY INFORMATION

Diagnostic display information helps the user to determine whether a DBSS operation failed, which operation failed and what caused the failure.

Based on the information displayed, users can decide what actions to take to make the data base operational again. These commands and actions are explained under "Filtered Log Recovery and Data Base Access" on page 31.



DBSS errors are caused by :

- bad (or corrupted) data in the data base.
- bad data in the SQL/DS log records.
- data in the data base that cannot be handled by SQL/DS.
- log records that cannot be correctly processed by SQL/DS.
- SQL/DS log records that have become out-of-sync with the data in the data base (for example, a log record wants to delete a row that does not exist in the data base).

### 3.1.1.1 ARI126E DBSS TERMINATION DURING REDO PROCESSING

A DBSS error occurred during warm-start or forward recovery log processing. DBSS was trying to redo changes made by a committed logical unit of work.

In the following example, a REDO error occurred for a logical unit of work, (LUWID 238C), updating a table with TABID -32767 in DBSPACE 8. The page accessed before the failure has the address 38B000 and has a logical page number of 88 which is a DATA page.

---

ARI126E DBSS TERMINATION DURING REDO PROCESSING	
LUWID = 238C	USERID = SQLDBA
DATE = 02-18-86	TIME = 15:44:44
OPERATION = UPDATE	DBSPACE = 8
TABLE-ID = -32767 (8001)	
PAGE-ADDRESS = 38B000	PAGE-TYPE = DATA
PAGE-NUMBER = 88	

---

### 3.1.1.2 ARI126E DBSS TERMINATION DURING UNDO PROCESSING

A DBSS error occurred during warm-start or forward recovery log processing. DBSS was trying to undo changes made by an uncommitted logical unit of work.

In the following example, an UNDO failure is caused by a logical unit of work with LUWID 1127 which did an INSERT into table with TABID -32767 in DBSPACE 8. The page with address 38A000 is empty - not allocated.

---

```
ARI126E DBSS TERMINATION DURING UNDO PROCESSING
LUWID = 1127                USERID = SQLDBA
DATE = 02-20-86            TIME = 19:38:59
OPERATION = INSERT         DBSPACE = 8
TABLE-ID = -32767 (8001)
PAGE-ADDRESS = 38A000      PAGE-TYPE = EMPTY
```

---

### 3.1.1.3 ARI126E DBSS TERMINATION DURING FORWARD PROCESSING

A DBSS error occurred during normal data base access.

In the following example, DBSS was performing an OPEN CURSOR on behalf of a logical unit of work with LUWID 1637. The page accessed has address 3A1000 and it is the first header page in DBSPACE 12.

---

```
ARI126E DBSS TERMINATION DURING FORWARD PROCESSING
LUWID = 1637                USERID = SQLDBA
OPERATION = OPEN CURSOR     DBSPACE = 12
TABLE-ID = -32767 (8001)
PAGE-ADDRESS = 3A1000      PAGE-TYPE = HEADER
PAGE-NUMBER = 80
```

---

### 3.1.1.4 ARI126E DBSS TERMINATION DURING ROLLBACK PROCESSING

A DBSS error occurred while it was trying to rollback changes made by the LUW identified by LUWID.

Please refer to the appropriate Diagnosis guide for detailed descriptions of the fields in the Diagnostic Display.

## 3.1.2 FILTERED LOG RECOVERY AND DATA BASE ACCESS

This facility is used when SQL/DS fails during log recovery. It does not resolve the problem caused by DBSS errors, however, it allows you to make your data base operational again while your problem is being fixed.

Filtered Log Recovery and Data Base Access facility allows users to:

- bypass undo processing during log recovery. This allows you to bypass operations that caused SQL/DS to fail during log recovery.

- rollback a committed logical unit of work that caused REDO failures during log recovery.
- disable a DBSPACE to prevent the users from accessing a DBSPACE that contains inconsistent data. It can also be used to take an erroneous DBSPACE 'off-line' to prevent SQL/DS system abnormal termination.
- enable a DBSPACE to allow users to resume access to a previously disabled DBSPACE.

There are four commands available for Filtered Log Recovery and Data Base Access.

### 3.1.2.1 BYPASS UNDO WORK

Changes in the data base caused by uncommitted LUWs are undone during SQL/DS log recovery processing.

BYPASS UNDO WORK can be used to make the data base operational again when start-up fails due to a DBSS UNDO processing failure. After a BYPASS UNDO, partial changes caused by uncommitted logical units of work are contained in the data base. The user data in the data base may be inconsistent. To prevent the users from accessing the inconsistent data, you should consider disabling the DBSPACES that are affected by the BYPASS UNDO WORK.

**When do you use it?:** BYPASS UNDO WORK is used in connection with the diagnostic display information when SQL/DS failed during log recovery UNDO processing.

**How to use it?:** The diagnostic support will display the information (the DBSPACE number, USERID, LUWID and so on) from the log record that SQL/DS was trying to 'undo' when the failure occurred. You can use the information to decide which control keywords for the BYPASS UNDO WORK command should be used. Normally you would use the LUWID for the first failure that you encountered. If the failure occurs again but on other logical units of work, you should try to detect a pattern in these failures.

BYPASS UNDO WORK allows you to specify the DBSPACE with USERID or ALL as control keywords. If you need to bypass an UNDO process for a specific USERID on a certain DBSPACE, you can combine both DBSPACE and USERID as the control keywords. Please refer to "Commands Syntax and Control Keywords" on page 16 for a description of the syntax and control keywords available for BYPASS UNDO WORK.

DISABLE DBSPACE only affects forward processing but not log recovery processing. You have to use BYPASS UNDO WORK to bypass the UNDO processing of changes for a DBSPACE that you wish to disable.

You cannot bypass UNDO processing on DBSPACE 1. This is restricted to minimize the inconsistencies in the data base. The system catalogs will reflect the true current state of the data base. Hence, **do not use DBSPACE 1 for user data.**

Before any UNDO is bypassed a summary message is displayed which identifies all LUWs which will be bypassed. Logical units of work which are not undone but updated the same tables are displayed. After examining the summary messages, you have to answer a prompt whether to continue or to cancel the start-up. Please refer to "Summary Messages" on page 36 for a description.

### 3.1.2.2 ROLLBACK COMMITTED WORK

The command allows you to rollback logical units of work that were committed.

ROLLBACK COMMITTED WORK is an extended BYPASS REDO. A BYPASS REDO would only ensure that changes after a checkpoint would not be redone; ROLLBACK COMMITTED WORK ensures that all changes of a logical unit of work are undone (those after the checkpoint are ignored and those before the checkpoint are undone).

#### When do you use it?

1. Primarily after a DBSS REDO failure during log recovery.
2. To rollback committed user errors : but use it for this purpose only with extreme care.

**Beware of inconsistencies:** For example, LUW1 inserts a new row. Subsequently, LUW2 updates this new row. If you rollback LUW1 using the ROLLBACK COMMITTED WORK command, LUW2 will be affected. (You will get a REDO failure for LUW2.) Hence, we would recommend that you use ROLLBACK COMMITTED WORK only as a last resort to rollback committed user errors.

Data inconsistencies may also be caused by the use of these commands to avoid warm-start failures. However, this may be the only way to make your data base operational as soon as possible after the failure.

**How to use it?:** You can rollback committed LUWs identified by LUWID or any combination of TIME, DATE, USERID and DBSPACE number. Please refer to "Commands Syntax and Control Keywords" on page 16 for a detailed description of the commands and control keywords.

To make the data base available after a DBSS error, use the diagnostic display information to determine the control keywords you should use.

If you are using it to rollback user errors, you should use the LUWid to rollback the work if possible. However, if the LUWid is not known, you

can use USERID together with the DATE, TIME and DBSPACE parameters to back out the errors.

**Note:** If you want to find out the LUWid for work that was committed, specify a broader range in the EXTEND file and you can gather the LUWid information from the summary messages. Then cancel the start-up and be more specific in a new EXTEND file.

**What LUWs can be rolled-back during STARTUP=W ?:** There are two cases to consider for ROLLBACK COMMITTED WORK :

- LOGMODE=Y

You can force the rollback of any logical unit of work that was completed after the last checkpoint. You do not need the DATE and TIME control keywords.

If SQL/DS terminates normally, it will take a checkpoint before it ends. Hence, you cannot rollback any committed LUWs after normal termination.

- LOGMODE=L or A

In addition to rolling back logical units of work that were committed after the checkpoint without DATE and TIME parameter, you can also rollback committed work recorded in the current log. You need to specify the TIME and DATE parameters. You can omit the DATE parameter if you want to rollback logical units of work that are done on the current date. You can also specify a time span (for example, from TIME 10:20:00 TO TIME 10:50:50 on three separate lines in the input control file).

**Re-apply changes made by logical units of work that were rolled-back:**

Once a logical unit of work is rolled-back via ROLLBACK COMMITTED WORK, it can only be recommitted to the data base by manually resubmitting a new logical unit of work or restoring from an archive. For details please refer to "Filtered Log Recovery" on page 50.

### 3.1.2.3 DISABLE DBSPACE

This command allows you to disable the DBSPACE(s) which is (are) causing problems. Disable DBSPACE allows you to take the problem DBSPACE 'off-line' until the error can be corrected, for example by manually recreating the updates of inconsistent user data or by restoring an archive copy. This will allow users to continue working on other data and reduces the risks of inadvertently causing another crash. It also prevents users from working with data that is inconsistent.

Disabling a DBSPACE will affect only forward processing. The log recovery and restore process always assumes that DBSPACEs are enabled. The

specified DBSPACES will be disabled only when SQL/DS initialization completes.

There are several instances when you should disable a DBSPACE:

- when the DBSPACE contains inconsistencies caused by BYPASS UNDO WORK or ROLLBACK COMMITTED WORK. This will prevent accessing incorrect data.
- when the DBSPACE repeatedly causes DBSS errors. With the diagnostic display information, you can determine which DBSPACES are to be disabled. This will prevent access to the DBSPACE that caused the SQL/DS system to fail.
- when data in it is logically inconsistent because of user errors.

**Accessing a Disabled DBSPACE:** The following message will be displayed if a user tries to access tables residing in a disabled DBSPACE:

---

```
ARI503E AN ATTEMPT TO ACCESS A DBSPACE FAILED
ARI503E BECAUSE IT IS DISABLED.
ARI505I SQLCODE = -711  ROWCOUNT = 0
ARI504I SQLERRP: GENCODE SQLERRD1: -1 SQLERRD2: -200
```

---

However, the authorization commands such as GRANT and REVOKE on tables in the disabled DBSPACE will execute successfully.

**DBSPACES SYS0001 and SYS0002 ... SYS000n:** DBSPACE 1 (SYS0001) cannot be disabled; SQL/DS initialization will fail if you try to disable it. It is impossible to withdraw access to this DBSPACE. All data manipulation commands such as SELECT, INSERT and so on, all data definition commands such as CREATE TABLE, ALTER TABLE and so on, as well as authorization commands GRANT and REVOKE need to access or update the SQL/DS catalogs. Optimization also needs the SQL/DS catalogs. All catalogs are stored in DBSPACE SYS0001.

If you disable DBSPACE 2 (SYS0002) ISQL, Data Base Service Utility, preprocessing and the access modules and views that are in SYS0002 cannot be accessed.

You can disable other access module DBSPACES such as SYS0003 ... SYS000n. However, applications cannot be executed if their access modules are stored in the disabled DBSPACE.

Each time SQL/DS is started, the message ARI121I will be displayed on the console to inform you of the DBSPACES that are disabled.

### 3.1.2.4 ENABLE DBSPACE

This command enables DBSPACE or DBSPACES that were previously disabled.

DBSPACES that are disabled persist across sessions. You have to invoke the ENABLE command explicitly (in the next SQLSTART) to resume access to a DBSPACE that was disabled.

When you restore the data base from an archive copy, the DBSPACES that are disabled are those that were disabled when the data base archive was taken.

### 3.1.2.5 DROP DBSPACE and DROP TABLE Considerations

Normally with STARTUP=W, you cannot rollback a logical unit of work that contains a DROP DBSPACE or DROP TABLE command.

If the LUW with the DROP TABLE command took place after the last checkpoint before failure, you can rollback this LUW containing the DROP TABLE command with STARTUP=W.

DROP DBSPACE always causes a checkpoint and you cannot rollback a DROP DBSPACE command with STARTUP=W. SQL/DS normal termination always takes a checkpoint. Please refer to "DROP DBSPACE and DROP TABLE considerations" on page 51 for more details.

### 3.1.2.6 Summary Messages

SQL/DS analyzes the log to determine the logical units of work that will be affected by a ROLLBACK COMMITTED WORK or BYPASS REDO WORK. There are two types of messages:

1. ARI212I - which informs the user of the logical units of work that will be rolled-back or bypassed. The message (ARI237I) that follows indicates the USERID, TIME, DATE, DBSPACE number and OPERATION for each LUW to be rolled-back / bypassed. Message ARI210I will be displayed if no logical unit of work will be rolled-back or bypassed.
2. ARI211I and ARI214I - Information on the LUWs that will be impacted by the logical units of work rolled-back or bypassed. SQL/DS checks which logical units of work are working with the same tables as those to be rolled-back or bypassed. LUWid, DBSPACE number, USERID and TABLE-ID are displayed.

The displayed logical units of work may be updating or deleting objects that no longer exist or may contain inconsistent data. This

could cause new DBSS REDO or UNDO failures or propagation of the inconsistencies. This may lead to :

- DBSS termination due to FORWARD processing (message ARI126E).
- 'should not occur' SQLCODEs.
- incorrect data in the data base.

The user should read this information carefully before answering the message ARI213D to CONTINUE or CANCEL. If CONTINUE is chosen, the ROLLBACK COMMITTED WORK changes are processed and initialization continues. If CANCEL is chosen, initialization is terminated. The data base and log are left unchanged.

Following is an example to rollback a logical unit of work specifying the DATE, TIME and USERID. The messages for BYPASS UNDO WORK are similar.

---

```
ROLLBACK COMMITTED WORK WHERE
DATE 02-11-86
TIME 08:50:00
USERID SQLDBA
ARI283I LOG ANALYSIS COMPLETE
ARI212I SUMMARY INFORMATION OF THE RECORDS TO BE ROLLED BACK :
ARI212I LUWID = 4B3C                      USERID = SQLDBA
ARI212I DATE  = 02-11-86                  TIME   = 08:51:49
ARI237I DBSPACE 10
ARI237I   1 UPDATES
ARI211I THE FOLLOWING LOGICAL UNITS OF WORK MAY BE IMPACTED BY THE
ARI211I REQUESTED BYPASS OR ROLLBACK OPERATION(S):
ARI214I LUWID = 4B3D                      USERID = SQLUSER1
ARI214I DBSPACE = 10                      TABLE-ID = -32761 (8007)
ARI213D ENTER 'CONTINUE' TO CONTINUE SQL/DS INITIALIZATION OR
ARI213D 'CANCEL' TO END THE INITIALIZATION.
```

---

In the above example, logical unit of work LUWID 4B3C updated a table in DBSPACE 10. This LUW is initiated by user SQLDBA at the DATE and TIME displayed. Notice that the TABLE-ID is not displayed.

The logical unit of work LUWID 4B3D may be dependent on LUWID 4B3C. If LUWID 4B3C is rolled-back, LUWID 4B3D may be affected. To find out which table has the indicated TABID, you should check the TABID column in SYSTEM.SYSCATALOG. (Since SQL is not running at this point in time, you need to have a printout of SYSCATALOG available.)

If there are several tables with the same DBSPACE and a number of logical units of work are to be rolled-back and impact several other rolled-back LUWs, it is difficult to tell which LUW is impacted by which rolled-back LUW.



### 3.1.2.7 Examples

Following are examples of using the commands in both VM/SP and VSE environments :

- **In the VM/SP environment:**

---

```
FILEDEF ARIEXTND DISK rollbk10 extend a
SQLSTART DB(PRODDB01) PARM(LOGMODE=L,EXTEND=Y)
```

Contents of file ROLLBK10 EXTEND A

```
* Control file to rollback logical units of work
* that are committed to DBSPACE 10 and user JAMES.
*
* Disable the DBSPACE 10.
```

```
ROLLBACK COMMITTED WORK WHERE
  DBSPACE 10
  USERID JAMES
DISABLE DBSPACE 10
```

---

- **In the VSE environment:**

---

```
// JOB FILTERED LOG RECOVERY
// EXEC PROC=ARISLIBP
// EXEC PROC=PRODDB01
// EXEC PGM=ARISQLDS,SIZE=AUTO,PARM='LOGMODE=L,EXTEND=Y'
* Control file to rollback logical units of work
* that are committed to DBSPACE 10 and user JAMES.
*
* Disable the DBSPACE 10.

ROLLBACK COMMITTED WORK WHERE
  DBSPACE 10
  USERID JAMES
DISABLE DBSPACE 10
/*
/&
```

---

**LUWs with Dependencies:** Following gives an example of the result of rolling back an LUW on which a second LUW is dependent. You should be prepared for these types of inconsistencies if the summary messages indicate dependencies. It is the user's responsibility to deal with them.

Original Value	1st LUW	2ND LUW	RESULT	
			If 1st LUW is Rolled-Back but 2nd LUW is not	
			SALARY	TAX
Example 1 SALARY= 50000	UPDATE SALARY = 60000	UPDATE TAX =SALARY*.1 =6000	50000	6000
			Note : In this case, SALARY and TAX columns are not synchronized	
Example 2 SALARY= 50000	UPDATE SALARY = SALARY*1.1 = 55000	UPDATE SALARY =SALARY*1.1 =60500	50000 or 60500	
			Note : The value for SALARY is unpredictable. It depends on when LUW2 took place and when checkpoint occurred.	
Example 3	INSERT a row with EMPLOYEEID = 10	UPDATE SALARY for EMPLOYEEID =10		
			Note : If 1st LUW is rolled back. REDO failure will occur if SQL/DS has to redo 2nd LUW during log recovery	

Figure 4. LUWs with Dependencies Example

### 3.1.2.8 Guidelines

The objectives of the recovery actions are to:

1. Bring up the data base as soon as possible after the data base machine failure to allow users to partially access the data base. The data that caused the failure may be made inaccessible to avoid causing further system failures or producing incorrect results.
2. Allow corrective actions to be thought out while users are still accessing portions of the data base.

The actions to recover the data base might involved manually undoing or redoing all the changes. This can be very complicated if too many applications are involved. To ensure that your data base is still intact after you have manually recreated all the changes, you should always do

a Directory Verify and take an archive if the verify result indicates no problems.

### Recovery Actions for FORWARD Processing Failures

1. Try to start up the data base machine again. In most cases this should be successful.
2. If you cannot bring up the data base, the error messages that you get will be UNDO or REDO processing failures. Please refer to the respective sections depending on these messages.
3. If you can start up the data base, run Directory Verify. In most cases no problems should be indicated.
4. If Directory Verify shows that there are inconsistencies in the directory, please refer to "Forward Recovery" on page 42 to remove these.
5. If you can bring up the data base and Directory Verify shows that there is no discrepancy in the directory, continue with your normal operation.
6. In case the forward processing failure occurs again take the procedures outlined below :
  - a. Detect the pattern of the failure :
    - If the problem occurs in the index pages of a particular table (the same DBSPACE number, TABLE-ID and INDEX-ID appear repeatedly), try to drop and recreate the index.
    - If the same TABLE-ID appears repeatedly, it is possible that the data in the table cannot be accessed. You can use the TABLE-ID and DBSPACE number displayed as the search conditions to determine the table name and creator from the SYSTEM.SYSCATALOG. Inform the creator and users with DBA authority to stop accessing the table. To prevent other users from accessing the table, use the REVOKE command to revoke the privileges.
    - If the same DBSPACE number appears repeatedly, it is possible that some of the pages in the DBSPACE cannot be accessed. Use the DISABLE DBSPACE command to prevent any users from accessing the data in that DBSPACE.
    - If the USERID appearing in the diagnostic display is always the same, ask the user to stop executing the application.
  - b. Recover the affected parts of the data base.

This will allow you to make the data base operational again. However, the data affected is not accessible.

- 1) Try to unload the DBSPACE. You have to enable the DBSPACE if you had previously disabled it.
  - 2) Acquire another DBSPACE from another storage pool.
  - 3) Reload the DBSPACE.
  - 4) Drop the primary DBSPACE.
- c. If you cannot unload or DROP DBSPACE, restore the data base from a previous archive copy and apply all the log archives.

**Recovery Actions for ROLLBACK Processing Failures:** The actions to recover the data base from DBSS errors caused by ROLLBACK processing failures are similar to those described above for FORWARD processing failures.

**Recovery Actions for REDO Processing Failure during STARTUP=W:**

1. If time permits, you should do forward recovery. This will recover all the logical units of work in single thread mode and it will normally solve your problems.
2. To make the data base operational as soon as possible without forward recovery :
  - a. Restart the data base machine with STARTUP=W and EXTEND=Y.
  - b. Use ROLLBACK COMMITTED WORK to rollback committed LUWs that caused the REDO processing failure. Each time SQL/DS encounters a REDO failure, it terminates. The first time you encounter a REDO failure during warm-start, you might want to use LUWID to prevent SQL/DS redoing that particular logical unit of work. However, if you encounter several failures during the warm-start, you should detect the failure patterns and use other control keywords to rollback committed logical units of work.
    - Use DBSPACE as the control keyword if all the failures are in the same DBSPACE.
    - Use USERID if all failures show the same userid.
    - Use multiple LUWIDs as the control keyword if no DBSPACE or USERID can be detected.
  - c. After using the ROLLBACK COMMITTED WORK to make the data base operational again, it might contain inconsistent data. You have to manually recreate the changes.

It can happen that the applications or the logical units of work to be manually redone are too complicated and you need more time to plan for it. You should disable the DBSPACE to prevent further access to the data that is inconsistent.

- d. As an alternative to manually repairing the data, use forward recovery as soon as possible.

### **Recovery Actions for UNDO Processing Failure during STARTUP=W:**

Please refer to "Recovery Actions for REDO Processing Failure during STARTUP=W:" on page 41 for more details. Use BYPASS UNDO WORK instead of ROLLBACK COMMITTED WORK. With BYPASS UNDO WORK, you can use the control keyword ALL if it is apparent that the whole log cannot be processed at all.

## **3.2 FORWARD RECOVERY**

Forward recovery is necessary to recover from DASD failures. It may be also necessary to ultimately recover the data base from user data inconsistencies introduced by filtered log commands.

Forward recovery involves:

- restoring a data base archive and
- log recovery.

If you are running with log archiving, then log recovery includes applying all the log archives.

Normally you should be able to recover the data base from your archive copy and apply all log archives. There might be exceptions when recovery fails. The following sections cover the considerations for forward recovery, the usage of the new facilities to help in performing forward recovery and the actions to take when recovery fails initially. For more details refer to "Recovery after System Failures" on page 29 for actions to take for log recovery processing failures.

### **3.2.1 DATA BASE RECOVERY**

Before archiving your data base, run Directory Verify to ensure that what you are going to archive is intact and can be used for future restore.

#### **3.2.1.1 Directory Verify**

**Guidelines:** To ensure that there is no inconsistency in your archive copy and it can be used for restore, you should:

1. always verify your directory before the data base archive is taken. Whether you should do the Directory Verify before log archiving,

depends on the time taken for verification and the time allocated for archive.

2. verify your directory periodically on non-archive shutdowns if your data base is archived infrequently. Frequent verifications will help to detect the inconsistency earlier and reduce the time to recover.
3. invoke the Directory Verify function after a significant amount of data has been updated or loaded into the data base, or some critical data has been modified in the data base.
4. invoke the Directory Verify function if you experience persistent SQL/DS abnormal terminations in the DBSS (watch for message ARI040E for modules with ARIY..).
5. invoke Directory Verify after a series of recovery actions to ensure that the data base recovery was successful.

If Directory Verify indicates errors, the recovery actions recommended are the same as for recovery from DASD failures. Please refer to "Special Considerations" on page 52 for more details.

**Directory Verify Examples:** This section contains examples of Directory Verify. We created errors in the directory to show the various messages returned.

#### 1. Successful Directory Verify

If the directory is intact, SQL/DS displays the following messages:

```
ARI216I SQL/DS DIRECTORY VERIFY FUNCTION COMPLETED WITHOUT  
ARI216I DISCREPANCIES.
```

In this case, an archive will be taken if requested. If UARCHIVE is specified, you will be informed to take a user archive for the directory and the DBEXTENTS but not the log(s).

#### 2. Physical Page Allocated Belongs to a Wrong Pool

The directory contains mapping information between DBSPACE pages and physical pages. All physical pages assigned to a DBSPACE must be from the same storage pool. Directory Verify will detect the discrepancy if physical pages assigned to pages in a DBSPACE belong to different pools.

Following is an example of messages returned by SQL/DS after the directory is verified to be inconsistent.

---

### sqlend uarchive dverify

```
ARI028I SQL/DS IS TERMINATING
ARI065I SQL/DS OPERATOR COMMAND PROCESSING COMPLETE
ARI218E DBSPACE 6, PAGE '80'X IS ALLOCATED
ARI218E TO PHYSICAL PAGE '6'X OF STORAGE
ARI218E POOL 1 INSTEAD OF STORAGE POOL 2.
ARI218E DBSPACE 6, PAGE '88'X IS ALLOCATED
ARI218E TO PHYSICAL PAGE '58'X OF STORAGE
ARI218E POOL 1 INSTEAD OF STORAGE POOL 2.
ARI218E DBSPACE 6, PAGE '1D8'X IS ALLOCATED
ARI218E TO PHYSICAL PAGE '349'X OF STORAGE
ARI218E POOL 1 INSTEAD OF STORAGE POOL 2.
ARI218E DBSPACE 6, PAGE '1D9'X IS ALLOCATED
ARI218E TO PHYSICAL PAGE '4F'X OF STORAGE
ARI218E POOL 1 INSTEAD OF STORAGE POOL 2.
ARI216I SQL/DS DIRECTORY VERIFY FUNCTION COMPLETED WITH
ARI216I DISCREPANCIES.
ARI245E WARNING: A USER-ARCHIVE SHOULD NOT BE TAKEN UNTIL
ARI245E DISCREPANCIES IN THE DIRECTORY ARE CORRECTED.
ARI032I SQL/DS HAS TERMINATED
ARI043I SQL/DS RETURN CODE IS 516
```

Figure 5. Directory Verify Example

---

The messages returned indicate that :

- five pages of the DBSPACE number 6 are inconsistent.
- the physical pages allocated to DBSPACE number 6 are from a wrong storage pool.
- the first page that is inconsistent is logical page '80'X. Page '80'X is a page of DBSPACE number 6 which belongs to storage pool 2. However, physical page '6'X is a page of a DBEXTENT from storage pool 1.
- the warning message ARI245E is displayed (because UARCHIVE is specified); the user archive should not be taken. If ARCHIVE is specified instead, SQL/DS will not take the archive.

You may still be able to unload the DBSPACE and recreate all the tables.

Don't try to drop the DBSPACE before correcting the problem. In our test, we tried to drop the table and the SQL/DS system terminated abnormally with the dropping 'half-done' - there is still an entry in the SYSTEM.SYSDBSPACES and SYSTEM.SYSDROP.

### 3. Invalid Page in the Page Map Table

Each DBSPACE has a page map table stored in the directory. For each page in the DBSPACE, there is an entry in the page map table which

points to the physical page number allocated. For each entry in the page map table, the Directory Verify facility checks if the physical page number assigned is a valid page.

---

```
ARI234E INVALID PHYSICAL PAGE 'AA03F8'X FOR
ARI234E PAGE '88'X OF DBSPACE 12.
ARI216I SQL/DS DIRECTORY VERIFY FUNCTION COMPLETED WITH
ARI216I DISCREPANCIES.
```

---

In this example, the physical page number 'AA03F8'X that is assigned to logical page number '88'X of DBSPACE 12 is invalid. There is no physical page with the number 'AA03F8'X.

If you do not take any action after an invalid page number is detected in the page map table, any access to these pages will cause the SQL/DS DBSS component to fail. Following is the message returned when we tried to access the table that is assigned to this page :

```
ARI040E SQL/DS SYSTEM ERROR OCCURRED - ARIYI07 03
```

You should not drop the DBSPACE with the invalid physical page number before the problem is corrected as this causes the SQL/DS machine to abend.

Normally you will not be able to unload the DBSPACE.

#### 4. Invalid Free Class in the Page Map Table

To avoid scanning DBSPACE pages to find a page that will have enough space to accommodate a new row, SQL/DS maintains summary information about the free space available in each page of the DBSPACES. The information is maintained in the Page Map Tables. Each entry in the page map table contains a FREECLASS designation that identifies the range of free space available on the reference page. Please refer to the appropriate Diagnosis Reference manual for a range of valid free classes.

Following is an example of having invalid free class in the page map table:

---

```
ARI233E INVALID FREECLASS FOR PAGE '88'X OF
ARI233E DBSPACE 10 (BLOCK '1'X).
ARI216I SQL/DS DIRECTORY VERIFY FUNCTION COMPLETED WITH
ARI216I DISCREPANCIES.
```

---

In this case, one of the pages allocated to DBSPACE 10 has an invalid freeclass. BLOCK '1'X is a directory block number of the page map



table for DBSPACE 10. BLOCK '1'X is the one that contains the entry for page '88'X of DBSPACE 10. A user might still be able to access the tables in this DBSPACE, but the result may be incorrect.

In the example that we tested, we actually have 40 rows in the table. After the freeclass was invalidated, SELECT \* FROM tablename gave us 16 rows with no error message returned; the other 24 rows are within the page that has an invalid freeclass. All rows in a page with an invalid freeclass cannot be accessed. To recover the data base, you have two choices :

- a. Restore the archive copy.
- b. If it is easier to get all the data in that DBSPACE from another source, we would recommend that you drop this DBSPACE, acquire a new one and reload the data. You should recreate the views and indexes if there are any and re-grant the authorization on tables and views. If you can perform this step successfully, your data base is brought up to date.

## 5. Invalid Bit Map Entry for Allocated Page

There is an entry in the bit map for each physical page of the DBEXTENT, one bit per page. 0 indicates that the page is allocated and 1 otherwise.

Following is an example of having an invalid entry in the bit map for a physical page that is allocated. The physical page is already allocated to a DBSPACE and it is pointed to by the page map table entry of that DBSPACE. But the bit map entry indicates that the page is still not allocated.

---

```
ARI217E PHYSICAL PAGE '25'X ALLOCATED TO  
ARI217E DBSPACE 10, PAGE '80'X NOT  
ARI217E ALLOCATED IN ALLOCATION BIT MAP.  
ARI216I SQL/DS DIRECTORY VERIFY FUNCTION COMPLETED WITH  
ARI216I DISCREPANCIES.
```

---

In this example, physical page '25'X (indicated in the page map table) is assigned to DBSPACE 10 logical page '80'X. However, the corresponding entry in the bit map for physical page '25'X is 1 instead of 0.

With this type of error in the directory, a user can continue to execute the SELECT command and thereby unload the DBSPACE. However, the SQL/DS machine will terminate abnormally during the checkpoint if there were modifications to the tables in this DBSPACE. We did several updates to fill up the log buffer pages and hence cause a checkpoint to be taken.

The checkpoint caused the SQL/DS machine to terminate abnormally with the following error message (DBSS component failed) :

ARI040E SQL/DS SYSTEM ERROR OCCURRED - ARIYI36 05

and mini-dumps display.

You should not drop the DBSPACE before the problem is fixed as it will cause a checkpoint to be processed. We did a DROP DBSPACE and that caused the SQL/DS machine to terminate although the message ARI500I SQL PROCESSING WAS SUCCESSFUL was displayed. You can unload the DBSPACE and disable it to prevent further access.

### 3.2.1.2 SQL/DS Archive / Restore

The SQL/DS archive and restore facility is still available in Release 3.5. The only change is in the tape blocking (in VSE only) from 4K to 28K to speed up the archive and restore process. The usage of this facility to archive and restore the data base is identical to that of previous releases.

### 3.2.1.3 User Archive / Restore

VSAM Backup/Restore in VSE and DDR in the VM/SP environments can be used to archive or restore the data base in a shorter time. The few exceptions where SQL/DS archive / restore must be used are :

1. when you need to migrate an SQL/DS data base between VM/SP and VSE.
2. when an on-line archive must be performed to reclaim the log space while users are still accessing the data base.

### 3.2.2 LOG RECOVERY

Log recovery in forward recovery is similar to warm-start log recovery. SQL/DS has to redo all the logical units of work recorded in the log and undo changes made by uncommitted logical units of work - this can happen if a data base archive was taken while users were still accessing the data base. With log archiving, the set of log archives has to be applied in the same sequence as they were taken.

### 3.2.2.1 Log Archive / Restore

Archiving the log is faster than archiving the DBEXTENTS and Directory. However, it will take longer if you need to restore the data base. If you are running with LOGMODE=L, you should consider taking data base archives periodically.

You have to maintain log continuity in order to apply the log archives associated with a data base archive. Log continuity is broken if any of the following events occurred :

1. COLDLOG
2. log(s) reconfigured
3. log modes switched. There is an exception when you switch from LOGMODE=L to LOGMODE=A and back to LOGMODE=L without taking an archive while running with LOGMODE=A.
4. END RESTORE when restoring the log archive

A little elaboration on this point :

- Supposing the following events are the series of data base archives and log archives you have done:

- Data Base Archive 1
  - Log Archive 1
  - Log Archive 2
  - Log Archive 3
  - Data Base Archive 2
  - Log Archive 4
  - Log Archive 5

- Supposing you need to restore your data base and you use data base archive 2. When you restore data base archive 2, following are the steps taken by SQL/DS :

- a. reads the identification on the archive tape.
- b. writes a record in the history area to indicate a RESTORE is done.
- c. checks into the history area to identify the data base archive copy by reading the history area from bottom to top until it finds data base archive 2.
- d. determines the log archives associated with data base archive 2 by reading forward until the RESTORE record is reached. The set of records is referred to as the 'restore set'.
- e. copies the restore set records after the RESTORE record.

f. displays the restore set to the console via messages.

If you restore all the log archives (log archive 4 and 5), the history area remains as what SQL/DS has copied.

```
Data Base Archive 1
Log Archive 1
Log Archive 2
Log Archive 3
Data Base Archive 2
Log Archive 4
Log Archive 5
RESTORE
Data Base Archive 2
Log Archive 4
Log Archive 5
```

Suppose you enter 'END RESTORE' after applying log archive 4. Then SQL/DS will delete the log archive 5 record.

```
Data Base Archive 1
Log Archive 1
Log Archive 2
Log Archive 3
Data Base Archive 2
Log Archive 4
Log Archive 5
RESTORE
Data Base Archive 2
Log Archive 4
```

At this point, if you found out that you had made a mistake and would like to restore log archive 5, there is one way for you to restore it. You cannot restore data base archive 2, as this will cause only log archive 4 to be restored. You have to restore the back-level archive data base archive 1 and apply log archives 1,2,3,4,5.

After the restore, you continue normal operation and you have two more log archives 6 and 7.

```
Data Base Archive 1
Log Archive 1
Log Archive 2
Log Archive 3
Data Base Archive 2
Log Archive 4
Log Archive 5
RESTORE
Data Base Archive 2
Log Archive 4
Log Archive 6
Log Archive 7
```

At this point, if you found out that you had made a mistake and would like to restore log archive 5, there is one way for you to do it. You cannot restore data base archive 2, as this will cause log archive 4,6,7 to be restored. You have to restore the back-level archive data base archive 1 and apply log archives 1,2,3,4,5. However, you have then lost all the changes in log archives 6 and 7 (you cannot apply log archives 6 and 7).

### 3.2.2.2 Filtered Log Recovery

The considerations described in "Filtered Log Recovery and Data Base Access" on page 31 for Filtered Log Recovery during warm-start also apply for Filtered Log Recovery during forward recovery. Therefore, only additional considerations are described below.

**BYPASS UNDO WORK:** If an archive was taken while users were still accessing the data base and this archive copy is used for data base restore, uncommitted logical units of work have to be undone.

If you had issued BYPASS UNDO WORK during warm-start to bypass the UNDO process for certain logical units of work, you have to reissue the command if you want the UNDO process to be bypassed for the same logical units of work during forward recovery.

**ROLLBACK COMMITTED WORK:** With STARTUP=R or STARTUP=U, you can use ROLLBACK COMMITTED WORK to rollback any LUW that was committed after the last data base archive.

With LOGMODE=L, you can rollback logical units of work that are recorded in the previous log archives.

1. Create an EXTEND file to rollback committed work with TIME, DATE and DBSPACE number as control keywords.
2. Restore the data base with the EXTEND=Y parameter.
3. You can specify different EXTEND files for several log archives by entering 'STOP SYSTEM' after each log archive restored and changing the EXTEND file.

The actions caused by the ROLLBACK COMMITTED WORK commands are recorded in the current log.

When restoring your data base from log archives, you should reissue the ROLLBACK COMMITTED WORK command to rollback logical units of work you previously rolled-back during warm-start.

A log archive which was taken **prior** to a warm-start where committed LUWs were rolled-back, does not contain any information about that ROLLBACK COMMITTED WORK. Therefore, these LUWs would have to be rolled-back again

during forward recovery if the user does not want the changes to appear in the data base.

A log archive that was taken **after** a warm-start where committed LUWs were rolled-back, does contain the information about that ROLLBACK COMMITTED WORK. Therefore, the changes caused by these LUWs will not appear in the data base after the restore completed, even if the LUWs are not explicitly rolled-back again during forward recovery.

Since this is also true for the current log which is archived in the process of restoring the data base, **LUWs rolled-back from the current log cannot be recovered through forward recovery.**

**DROP DBSPACE and DROP TABLE considerations:** If a logical unit of work to be rolled-back contains DROP TABLE or DROP DBSPACE commands, following rules apply :

1. The ROLLBACK COMMITTED WORK command must contain the DBSPACE control keyword. All the DBSPACES affected by the DROP commands must be specified via the DBSPACE control keyword.
2. The ROLLBACK COMMITTED WORK command must not contain other control keywords except LUWID. That is, USERID, DATE, and TIME control keywords are not allowed.

The message ARI256E will be displayed if these rules are violated.

```
ARI256E ROLLBACK BY DBSPACE 12 MUST BE SPECIFIED IN THE
ARI256E EXTEND FILE
```

The reasons for this special case for DROP TABLE and DROP DBSPACE are that:

- SQL/DS doesn't log the specific rows that were dropped by DROP TABLE or DROP DBSPACE.
- For each DROP command, SQL/DS uses two logical units of work. The second logical unit of work actually performs the DROP function and executes after the COMMIT WORK for the logical unit of work containing the DROP command completes.

With STARTUP=R and LOGMODE=A or L, you can rollback DROP TABLE or DROP DBSPACE commands after the last data base archive. For simplicity, assume the log contains only a DROP DBSPACE command. To rollback a DROP DBSPACE command, specify EXTEND=Y when restoring the data base. The EXTEND file contains the ROLLBACK COMMITTED WORK command with DBSPACE as the control keyword. You need not specify DBSPACE 1 although it is updated by the DROP DBSPACE command.

---

ROLLBACK COMMITTED WORK WHERE  
DBSPACE 8

The sequence of events :

```
Data Base
Archive 1           checkpoint
|-----|
|     DROP DBSPACE
|     # 8
|
|           ....|-----|
|           Restore Data      Result :
|           Base Archive 1    DROP DBSPACE 8
|           STARTUP=R        is rolled-back
|           EXTEND=Y
```

---

**Summary Messages:** SQL/DS cannot detect logical units of work dependencies across several log archives. For example, if LUW2 in the second log archive is dependent on LUW1 in the first log archive, SQL/DS will not detect that LUW2 is impacted if LUW1 is rolled-back. Hence, it is the user's responsibility to take care of logical units of work dependencies between several log archives.

If you are restoring your data base using LOGMODE=L, SQL/DS displays the summary messages after each log archive restored. If you need to specify different EXTEND files for each log archive, you can stop SQL/DS before restoring the log and restart the data base machine with a new EXTEND file. If you use one EXTEND file throughout the whole restore process, the commands within the EXTEND file will be applicable for all log archives.

### 3.2.3 SPECIAL CONSIDERATIONS

Forward recovery should be initiated to recover the data base:

- from a DASD failure.
- after Directory Verify indicated errors.
- ultimately after Filtered Log Recovery has been used to warm-start SQL/DS after errors and the possible discrepancies have not been removed manually.

The normal procedure to recover the data base includes :

1. restore the data base from a verified copy
2. apply the log archives if you are running LOGMODE=L

3. run Directory Verify again
4. take a data base archive if DVERIFY indicated no errors

In most cases this will result in an intact data base with consistent user data because the changes for all logical units of work are done in single thread mode.

There might be exceptions where failures occurred in the process of recovering the data base. This chapter covers the areas:

- restore fails
- log recovery fails
- error persists

### 3.2.3.1 Restore Fails

This can happen if the archive tape is damaged. (You should consider retrying the restore process because it might be due to tape drive malfunction.)

- If you are running LOGMODE=L:
  - You can restore a back-level data base archive and apply all log archives if you have not done any of the following since the back-level archive :
  - COLDLOG
  - added DBSPACE or DBEXTENT
  - re-configured the log(s)
  - interrupted a previous restore by END RESTORE
  - switched log modes
  - If you switched to LOGMODE=A but, without taking an ARCHIVE switched back to LOGMODE=L immediately, log continuity is maintained.
- If you are running with LOGMODE other than L:
  1. invoke SQLLOG to do a COLDLOG.
  2. restore a back-level data base archive. This archive should have been taken with SQLEND ARCHIVE, SQLEND UARCHIVE or on-line ARCHIVE with no LUW accessing the data base.



3. all changes since the back-level restore have to be recreated manually.

### 3.2.3.2 Log Recovery Fails

1. If the restore fails during the log recovery, re-invoke the restore using Filtered Log Recovery. Please refer to "Filtered Log Recovery and Data Base Access" on page 31 for more details.
2. Remove the inconsistencies introduced through the use of Filtered Log Recovery commands.
3. Shutdown SQL/DS with Directory Verify (SQLEND VERIFY).
4. If the result of the Directory Verify indicates no errors, take an archive at this point and continue with your normal operation. The following covers the exceptional case where problems are still not resolved.

### 3.2.3.3 Error Persists

This could happen if the single thread log recovery did not solve the problem.

1. If the problem DBSPACES are the user DBSPACES, there are several possibilities to consider :
  - Case 1.** You can UNLOAD and DROP the DBSPACE : This is the case when you have invalid freeclass entries in your directory. The page that has an invalid free class cannot be unloaded.
    - a. unload the DBSPACE.
    - b. drop the DBSPACE.
    - c. acquire another DBSPACE.
    - d. reload the DBSPACE.
    - e. recreate the data that could not be unloaded. (Data that is within the pages with invalid free class could not be unloaded.)
    - f. recreate the indexes if there are any.
    - g. recreate the views if there are any.

- h. re-grant authorizations to the affected users to access the views and tables.
- i. shutdown the data base with Directory Verify and take an archive if the discrepancies are resolved.

**Case II.** You cannot UNLOAD and DROP the DBSPACE : This is the case for all other DVERIFY errors.

- a. Use Filtered Log Recovery commands to rollback all LUWs which changed the DBSPACE with problems. Please refer to "Filtered Log Recovery and Data Base Access" on page 31 for more details.
- b. Manually recreate all the units of work that were rolled-back.
- c. Shutdown your data base with Directory Verify. If the discrepancy in the directory is resolved, take an archive.
- d. If the problem persists, the last resort is to remove all changes recorded in the log that caused the DBSPACE to become inconsistent.
  - 1) Do a COLDLOG.
  - 2) Restore the data base from the archive copy.
  - 3) Manually recreate all the changes that were removed from the log.

2. Recovering the data base due to directory discrepancy in system DBSPACES, SYS0001, SYS0002, ...SYS000n.

- a. Report the problem to IBM.
- b. Use Filtered Log Recovery commands to rollback all LUWs which changed the SYSTEM DBSPACES. Please refer to "Filtered Log Recovery and Data Base Access" on page 31 for more details.
- c. Manually recreate all the units of work that are rolled-back.

For SYS0001 DBSPACE, all data definition and authorization commands will update the system catalogs. Preprocessing applications will also update the catalogs. You have to redo all the rolled-back work that updates the catalogs in order to bring the data base up to date.

For SYS0002, ... SYS000n DBSPACES, redo all the rolled back preprocessing, CREATE PROGRAM, VIEW, DROP PROGRAM and VIEW.

- d. Shutdown your data base with Directory Verify. If the discrepancy in the directory is resolved, take an archive.

- e. If the problem persists, the last resort is to remove all changes recorded in the log that caused the DBSPACE to be inconsistent.
  - 1) Do a COLDLOG to remove all the updates in the log.
  - 2) Restore the data base from the archive copy.
  - 3) Manually recreate all the changes that were removed from the log.

#### 3.2.3.4 Summary of Forward Recovery Actions

1. Start forward recovery.
2. Restore Data Base Archive and apply all Log Archives.
3. If Data Base restore fails;  
    Use a Back-Level copy and go to 2.
4. If Log Recovery fails;  
    Use Filtered Log Recovery commands and go to 2.
5. Run DVERIFY.
6. If DVERIFY indicates errors :
  - If errors in User DBSPACEs
    - If errors Recoverable  
        UNLOAD Data, DROP DBSPACE, Recreate Data  
        go to 5
    - Else  
        Use Filtered Log Recovery  
        go to 2
  - Else  
        Use Filtered Log Recovery  
        Go to 2.
7. Archive Data Base.
8. End Forward Recovery.

## APPENDIX A. PERFORMANCE ASPECTS OF RELEASE 3.5 FUNCTIONS

The results were obtained using VMMAP under VM/SP. The SQL/DS data base machine is the only user active on the VM/SP system.

- 4361-5
- VM/SP Release 4
- 3370 DASD

The number of DBEXTENTs pages in use as shown by the SHOW DBEXTENT command is 1036 pages. The data base was generated with the sample data base parameters.

The log size is 9600 FB-512 blocks. INSERTS and DELETES were done to fill the log to the percentages shown below.

---

Activities Measured	measured in seconds		
	Elapsed Time	Total CPU Time	Total SIOs
Larchive (Log 36% full)	32	8	922 900
Larchive (Log 55% full)	42	10	1340 1200
Larchive (Log 80% full)	50	15	1918 1700
SQLEND NORMAL	4	1	74 100
SQLEND DVERIFY	13	4	243 200
SQLEND ARCHIVE	675	299	28800

Figure 6. Performance Aspects

---



## A

ARCHPCT 3, 7  
ARISCOR EXEC 23

## B

BYPASS REDO 33  
BYPASS UNDO WORK 15, 32, 42, 50  
  command syntax 16  
  inconsistencies 35  
  summary messages 36

## C

COLDLOG 6, 11, 48, 53  
continuity of log 9, 11, 48

## D

DASD failures 42  
Data Base Recovery  
  SQL/DS archive /restore 47  
  user archive 47  
  user restore 47  
diagnostic display  
  FORWARD processing failure 31  
  REDO failure 30  
  ROLLBACK processing failure 31  
  UNDO failure 30  
diagnostic support 13  
directory verify 11, 40, 53  
  different pool 43  
  examples 43  
  guidelines 42  
  invalid bit map entry 46  
  invalid free class 45  
  invalid page number 44  
Invocation 12

UARCHIVE 13

DISABLE DBSPACE 15, 34  
  command syntax 18  
DVERIFY 11

## E

ENABLE DBSPACE 36  
  command syntax 18  
END RESTORE 9, 11, 48, 53  
enhancements 1  
EXTEND  
  file 9, 20, 50  
  parameter 16, 20, 41, 50

## F

FILEDEF statement 4, 19  
Filtered Log Recovery 15, 31, 50  
  BYPASS UNDO WORK 15, 32, 50  
  command syntax 16  
  diagnostic support 13  
  DISABLE DBSPACE 15, 34  
  DROP TABLE or DBSPACE 36, 51  
  ENABLE DBSPACE 36  
  Examples 38  
  guidelines 39  
  invocation 16  
  ROLLBACK COMMITTED WORK 15,  
    33, 50  
  summary messages 36, 52  
forward processing failure 31,  
  37, 40  
forward recovery 42, 52, 56

## I

Installation 23  
  DCSS 25  
  disk space (VSE) 24  
  minidisks (VM) 23

ISQL 10, 26, 35

L

log archive  
  checkpoint 10  
  DB lock 10  
  dual log 4  
  END RESTORE 9, 11, 48, 53  
  history area 11, 48  
  invocation 6  
  label information examples 8  
  labelling the tapes 6  
  log continuity 11  
  log restore 9  
  performance aspects 4  
  STOP SYSTEM 9, 50  
  using minidisks 4  
log restore invocation 8  
LOGMODE 3, 6, 9, 13, 34, 48

M

MAINT 25  
MSHP 27

R

recovery problems 53  
REDO failure 30, 33, 41  
restore list 9  
restore set 9, 48  
ROLLBACK COMMITTED WORK 15, 33,  
  41, 50  
  command syntax 17  
  inconsistencies 33

summary messages 36  
ROLLBACK processing failure 31,  
  41

S

SHOW LOCK 10  
SQL  
  ARCHIVE 1, 47  
  checkpoint 5, 10, 33, 34, 46  
  disk space required (VSE) 24  
  minidisks required (VM) 23  
SQLLEVEL EXEC 21  
SQLRMEND EXEC 21  
STOP SYSTEM 9, 50  
system failures 29

T

tape  
  alternate drives 2  
  blocking 20  
  label information 6

U

UNDO failure 30, 32, 42  
user archive 1, 47  
user archive and restore  
  DDR 1  
  invocation 2  
  performance aspects 1  
  VMBACKUP 1  
  VSAM Backup/Restore 1  
user restore 1, 47

**READER'S COMMENTS**

**Title:      SQL/DS Release 3.5 Usage Guide**

You may use this form to communicate your comments about this publication, its organization or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Comments:

Reply requested    Name : \_\_\_\_\_  
                          Yes / No      Job Title : \_\_\_\_\_  
  Address : \_\_\_\_\_  
  \_\_\_\_\_



-----fold-----

IBM International Technical Support Center  
Dept. 3616, Bldg. 7030-75  
Schoenaicher Strasse 220  
7030 Boeblingen  
West Germany

-----fold-----

**READER'S COMMENTS**

**Title: SQL/DS Release 3.5 Usage Guide**

You may use this form to communicate your comments about this publication, its organization or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Comments:

Reply requested    Name : \_\_\_\_\_  
Yes / No         Job Title : \_\_\_\_\_  
                         Address : \_\_\_\_\_  
                         \_\_\_\_\_

-----fold-----

IBM International Technical Support Center  
Dept. 3616, Bldg. 7030-75  
Schoenaicher Strasse 220  
7030 Boeblingen  
West Germany

-----fold-----

READER'S COMMENTS

Title: SQL/DS Release 3.5 Usage Guide

You may use this form to communicate your comments about this publication, its organization or subject matter with the understanding that IBM may use or distribute whatever information you supply in any way it believes appropriate without incurring any obligation to you.

Comments:

Reply requested Name : \_\_\_\_\_

Yes / No Job Title : \_\_\_\_\_

Address : \_\_\_\_\_

\_\_\_\_\_

-----fold-----

IBM International Technical Support Center  
Dept. 3616, Bldg. 7030-75  
Schoenaicher Strasse 220  
7030 Boeblingen  
West Germany

-----fold-----

GG24-3049-00

SQL/DS Release 3.5 Usage Guide

GG24-3049-00

PRINTED IN THE U.S.A.



GG24-3049-0

A standard 1D barcode is located in the bottom right corner of the cover, within a white rectangular box. The barcode consists of vertical black bars of varying widths on a white background.