Function Overview

The ADARAI utility prepares the recovery log (RLOG), lists the information contained in the RLOG, creates the job control statements to recover the database, and disables ADARAI logging.

Transaction recovery is provided whenever an Adabas session is abnormally terminated. The Adabas autobackout routine, which is automatically invoked at the beginning of every Adabas session, removes the effects of all interrupted transactions from the database. See the restart/recovery information in the *Adabas Operations* documentation.

However, when a database data set (ASSO, DATA, or WORK) is destroyed, it is necessary to restore and regenerate the database to recover the lost data.

The Adabas Recovery Aid utility, ADARAI, can be used to automate and optimize *database* recovery. It records and reports all information needed to recover the database and builds the recovery job stream (JCL/JCS), which is the basis for reexecuting the jobs performed from the time of the last SAVE to the point of failure and error.

Note:
The job stream generation function is not yet available under z/VSE.

This chapter covers the following topics:

- Concepts and Components

Concepts and Components

The Adabas Recovery Aid comprises two components:

- an interface (ADARAC) to collect information as relevant events occur against the database; and

- a utility (ADARAI) to list the information collected, generate jobs to recover the database or files on the database, or deactivate recovery logging.

This section covers the following topics:

- The Collection Interface

- Recovery Log (RLOG)

- Generation: The Unit of Recovery

- Retaining Noncurrent Generations

The Collection Interface

The collection interface is called by the nucleus and by all utilities to record information about each event that occurs; for example, a nucleus stop/start, a utility execution, or an event generated by the Adabas Online System.

Recovery Log (RLOG)

The interface records all event information into a Recovery log file (RLOG) for use by the utility component. The RLOG stores the information about data sets, utility parameters, and protection logs needed to build the recovery job control. The RLOG data set is DD/RLOGR1.

In a nucleus cluster environment, all nuclei use the same RLOG. Concurrent updates to the RLOG are controlled by a lock.

Notes:

1. Sequential data sets used by the utilities whose runs are logged on the RLOG must be kept and available for any recovery operation; for example, the DD/EBAND input to an ADALOD LOAD operation.
2. ADADBS file changes are now recorded on the RLOG data set.
3. Information recorded in the RLOG generally exceeds that required for recovery; it can also be used as a record of events that have occurred on a database over a period of time.

Generation: The Unit of Recovery

Information is stored on the RLOG by generation, the logical unit used for recovery.

A *generation* includes all activity between consecutive operations of:

- ADASAV SAVE/RESTORE (database),

- RESTORE GCB, or

- SAVE DELTA/RESTORE DELTA (database).

The first generation includes the first operation and extends to (but excludes) the second. A new generation is started when a database can be recovered in full after the previous operation.

Generations may be normal, restricted, or erroneous:

- A generation is labeled "normal" if a full save was available when it started and no unusual events occurred while activities were being logged in it.

- A generation is labeled "restricted" when certain events occur during the logging cycle that make it impossible for ADARAI to rebuild the database without user intervention. ADARAI generates a job, but the job will not run without help from the user. For example, if the Work data set is decreased in size, the user must create a Work data set with the original size so that the recovery job can run correctly up to the point where the Work data set size was decreased.

- A generation is labeled "erroneous" when errors occur during the logging cycle, for whatever reason. ADARAI generates a job, but the job will not run without changes.

Note:
When a generation becomes restricted or erroneous, Software AG recommends that you start a new generation as soon as possible by performing an on- or off-line save of the database. If the Adabas Delta Save Facility is installed, a SAVE DELTA will start a new generation.

Retaining Noncurrent Generations

Noncurrent generations provide a history of operations that have affected the database for use in problem resolution or for audit purposes.

Access to noncurrent generations is essential if an attempt to recover a database fails after the RESTORE step in the recovery job is executed. At this point, the generation being recovered becomes the current generation. If it then becomes necessary to rebuild the recovery job, the generation being recovered will be an older generation.

The RLOG retains the number of generations specified by the MINGENS parameter during the ADARAI PREPARE step. ADARAI recycles generations when the number stored on the RLOG reaches the number specified by the MINGENS parameter.

When a new generation plus those already stored exceed the available RLOG space, one of two events will occur:

- if the minimum number of generations as specified by MINGENS can be maintained, the oldest generation is overwritten; otherwise

- the RLOG is placed *out of service* by setting a flag in the RLOG control block. In this case, data is no longer logged.