**ς software** ᴬᴳ

# Adabas Review

## Introduction

Version 4.3.2

September 2009

Adabas Review

This document applies to Adabas Review Version 4.3.2 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

## Table of Contents

# 1 Introduction

Adabas Review is a set of tools for monitoring the performance of Adabas environments and the applications executing within them. Information retrieved about Adabas usage can be used to tune application programs for maximum performance with minimal resources.

The accompanying user profile system is used by the system administrator to define access privileges to Adabas Review.

Adabas Review runs in

- local mode in the Adabas address space. Under z/VM, Adabas Review runs in local mode only.
- hub mode as a server in its own address space with a client interface in the Adabas address space.
- batch mode as a batch job

**Note:** In the z/VM environment, Adabas Review is available only in local mode.

Hub mode is required for Adabas Parallel Services and Adabas Cluster Services environments.

The Adabas Review Introduction is organized in the following topics:

| |
|---|
| ● **Collecting Data in Local Mode** |
| ● **Collecting Data in Hub Mode** |
| ● **Collecting History Data** |
| ● **Reports** |
| ● **Autostarted Reports** |
| ● **Repositories** |
| ● **User Profile Systems** |

# 2 Conventions

This following syntax conventions are used in this documentation when describing ADARUN parameters and operator commands:

- The vertical bar ( | ) separates parameter options or values, one of which must be chosen.
- Curly brackets or braces ( { } ) delimit the series of options or values.
- (Square) brackets ( [ ] ) indicate a term that may optionally be specified.
- The underline ( _ ) identifies the default value.
- Three dots preceded by a comma or a slash ( , ... or / ... ) indicates that the previous term can be repeated but must be preceded by the comma or slash as indicated.
- A range of values is indicated by two instances of a variable (for example, "fnr", "rabn", "csn") separated by a hyphen (for example, "fnr - fnr"). The first instance represents the low value and the second the high value of the range. The range specification comprises the two numbers separated by the hyphen with no intervening spaces.
- Parentheses ( ( ) ) are part of the term and must be included as part of the parameter statement.

Dataset names starting with DD are referred to in the Adabas documentation with a slash separating the DD from the remainder of the dataset name to accommodate VSE/ESA dataset names that do not contain the DD prefix. The slash is not part of the dataset name.

# 3 Collecting Data in Local Mode

In local mode, the Adabas Review processor is installed as an extension to ADALOG.

> **Note:** Adabas Review supports only command log record layout format 5 (CLOGLAY-OUT=5).

The data collection process is partly accomplished by the Adabas Review processor, which is an Adabas processor with two modes of execution: interactive and batch.

- In batch, the processor runs as a batch job that processes sequential Adabas command log datasets.
- In interactive mode, data is collected by providing code (RAOSLOCL) that runs as an extension to ADALOG. Adabas calls this module and passes information about resource usage for each command processed by the Adabas nucleus.

This module, in conjunction with the Adabas Review processor and an intermediate Review buffer accumulates and tabulates the Adabas information based on various user-defined data requirements.

The Adabas Review data may be

- displayed in an online environment from the Adabas Review user interface;
- saved automatically in the Adabas Review repository;
- printed automatically when the Adabas nucleus terminates; or
- downloaded directly to a personal computer (PC) using Software AG's Entire Connection.

The following graphic shows the Adabas Review data collection process for local mode.
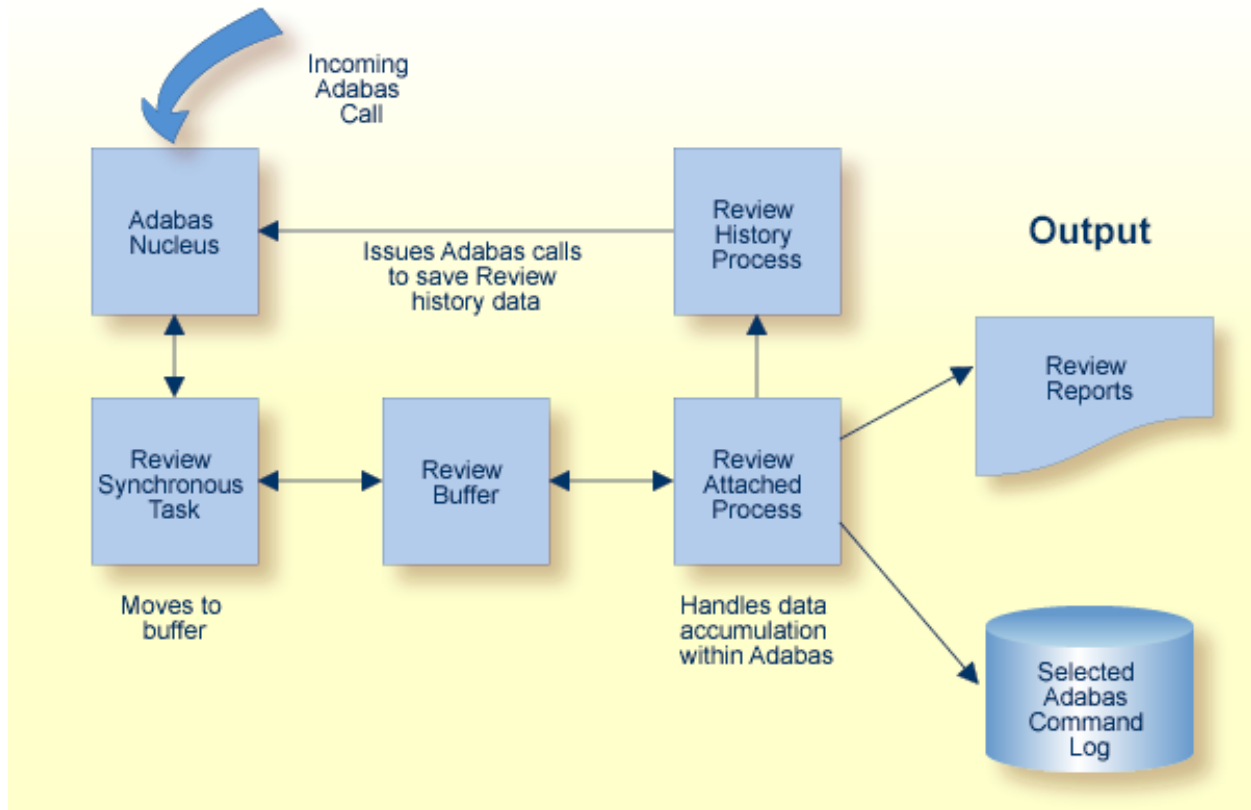
# 4 Collecting Data in Hub Mode

In hub mode, Adabas Review uses a client/server approach to collecting data:

- an interface (the client) resides on each Adabas nucleus; and
- the hub (the server) resides in its own address space, partition, or region.

The interface uses the existing Adabas interregion communication process: ADALNK, ADASVC, and ADAMPM. This process is consistent across the targeted platforms for Adabas Review. If systems are networked correctly, hub mode supports a multiple platform, multiple operating system, Adabas database environment.

This chapter covers the following topics:

# The Server Hub

The Adabas Review hub is a centralized data collector and reporting interface that combines proven components of Adabas and Adabas Review.

It handles the data consolidation and reporting functions for monitoring an Adabas database, including usage information related to applications, commands, minimum command response time (CMDRESP), I/O activity, and buffer efficiency.

The interactive reporting facility allows you to pinpoint problems quickly, providing detailed and summary data about Adabas activities. Specific information about each database is also available.

The centralized collection server has several advantages:

- A single hub collects information from multiple Adabas nuclei, Adabas Parallel Services (support for single operating system image multiprocessing) clusters, or Adabas Cluster Services (support for IBM's parallel sysplex environment) clusters.
- Because a single hub can support multiple Adabas nuclei, the number of Adabas Review nuclei required to support an enterprise-wide distribution of Adabas nuclei is reduced. This minimizes resource requirements and increases performance.
- Isolating the Review subtask from the Adabas nucleus enhances the performance of the Adabas main task and minimizes the impact of future Adabas releases on the functioning of Adabas Review.

The hub comprises

- ADAREV, a logic module that manages and supervises the incoming Review data calls and requests;
- REVHUB, a module to establish and maintain the environmentals for Adabas Review;
- the Review DB nucleus and subsystems including RAOSAUTO, the autostarted report parameter generation routine, and RAOSHIST, the historical data population routine.

## The Client Interface

The Adabas Review interface constructs and then transmits the Review data from the Adabas nucleus to the Adabas Review hub. An Adabas Review interface is integrated with each Adabas nucleus that is monitored.

The interface comprises

- ADALOG, the Adabas command logging module;

- ADACLX, the Adabas command log extension module that is responsible for acquiring additional information not present in the Adabas command log record; and

- ADARVU, which handles the environment conditions for ADACLX and the Adabas API requirements for transmitting the Review data to the Adabas Review hub.

## Interface Calls

To maximize performance, the ADARVU module issues an "optimistic" call from an Adabas nucleus to the Adabas Review hub without waiting for a completion or "post" from the hub; ADARVU assumes that the Review data was successfully passed to the hub.

However, ADARVU does perform an initialization step to ensure that the hub is active prior to any command processing by the Adabas nucleus. If the hub is not active, ADARVU informs you using WTOs and/or a user exit. If a user exit is used, you are given the option to wait for the hub to be activated, or continue initialization and call the hub only when it is active.

On the hub side of the call, the elimination of the cross-memory "post" call enhances performance by reducing the overhead of active communication with the Adabas clients. This allows the hub to remain a passive data collector.

## Example Client/Server Environment

The following graphic shows the major components of the Adabas Review interface (Adabas nucleus address space) and hub (Adabas Review hub address space) in a client/server architecture .
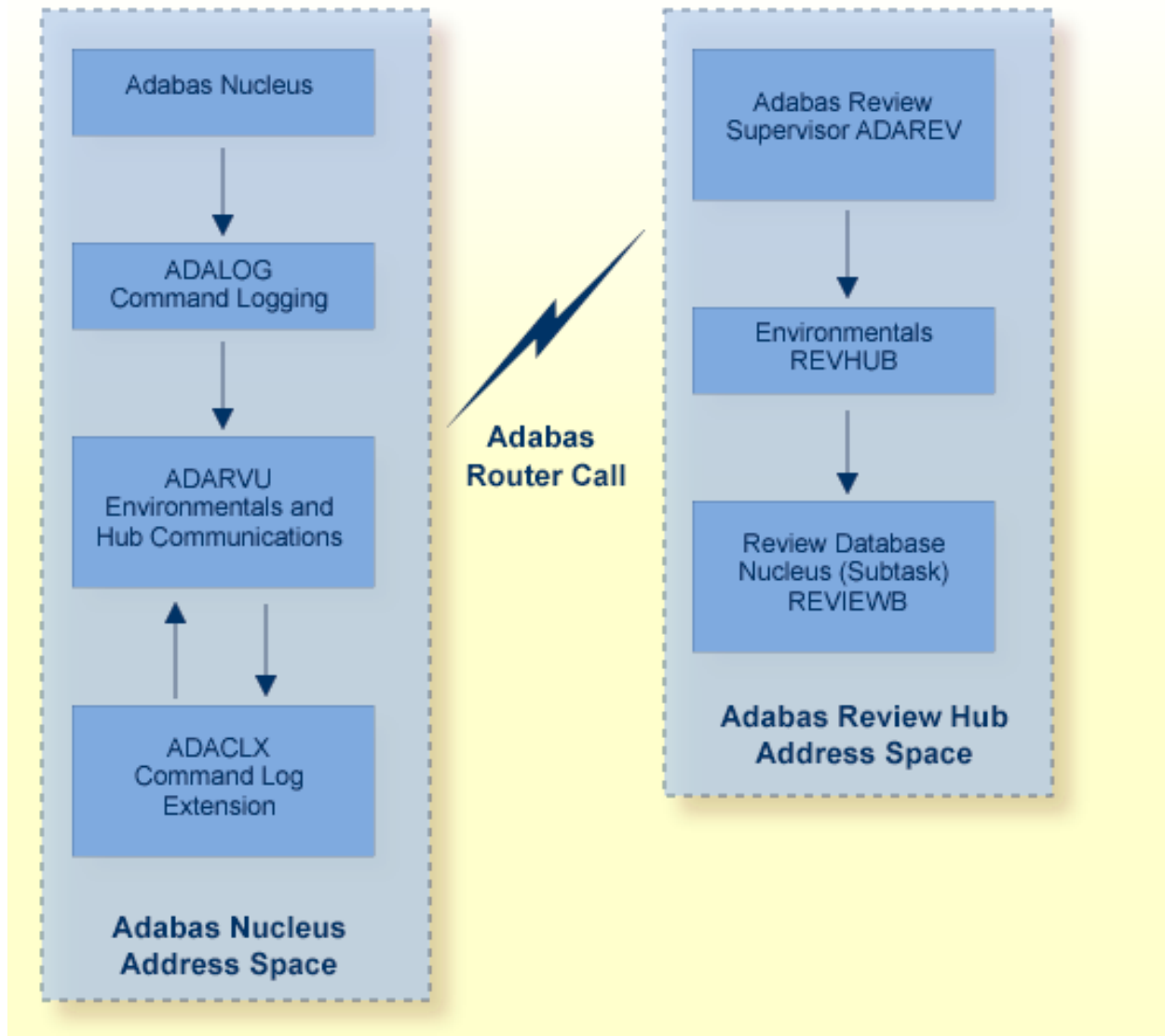
## Command Log Processing

The data collection process is partly accomplished by the hub (server) component REVIEWB, the Adabas Review command log processing routine.

> **Note:** Adabas Review supports only command log record layout format 5 (CLOGLAY-OUT=5).

REVIEWB has two modes of execution: interactive and batch.

- In interactive mode, it runs as an Adabas Review *subtask*; that is, a unit of work that the operating system treats as separately detachable.

■ In batch mode, it runs as a batch job that processes sequential Adabas command log datasets.

At initialization, REVIEWB reads any autostarted report definitions the user has defined and collects data according to the reports' criteria. REVIEWB also processes requests to start, view, and purge reports from the Adabas Review online system.

### Interactive Mode

In interactive mode, Adabas responds to requests and calls the interface module ADARVU from ADALOG (the Adabas command logging module) if `REVIEW=hubid` is specified in the Adabas initialization parameters. Adabas passes to ADARVU information about resource usage for each command processed by the Adabas nucleus.

Adabas Review link routine exits are used to pass TP system and Natural information from the user's address space (origin of the Adabas call) to the Adabas address space and, using an extension of the Adabas user buffer, on to ADARVU.

ADARVU queues Adabas command log records received from ADALOG to REVIEWB through an intermediate REVIEW-BUFFER in the hub (server) address space. REVIEWB processes the records, accumulating and tabulating various data according to the criteria specified in any user-defined reports that are active.

The resulting nucleus statistics may be

■ displayed in an online environment from the Adabas Review user interface;

■ saved automatically in an Adabas file called the Adabas Review repository;

■ printed automatically when the Adabas nucleus terminates; or

■ downloaded directly to a personal computer (PC) using Software AG's Entire Connection.

### Batch Mode

As a batch job, Adabas Review processes Adabas command log records from a sequential dataset. If you use Adabas dual command logging, you must first use the Adabas utility function ADARES CLCOPY to generate a sequential command log dataset suitable for input into Adabas Review.

When Adabas Review executes as a batch job, input report parameters that define the data collection criteria selected by the user are read from statements in the RVUPARM dataset or the RVUAUT1/RVUAUT2 datasets. These statements can be generated using the GENCARD statement.

The storage allocated for reports is exactly the same as that for Adabas Review executing in interactive (online) mode. However, since REVIEWB is reading the command log records directly from a sequential file, no REVIEW-BUFFER is allocated.

# Example Hub Mode Data Collection Process

The following graphic shows the Adabas Review data collection process for hub mode. When monitoring multiple databases, Adabas Review allows you to switch from one database to another and provide reports for each.
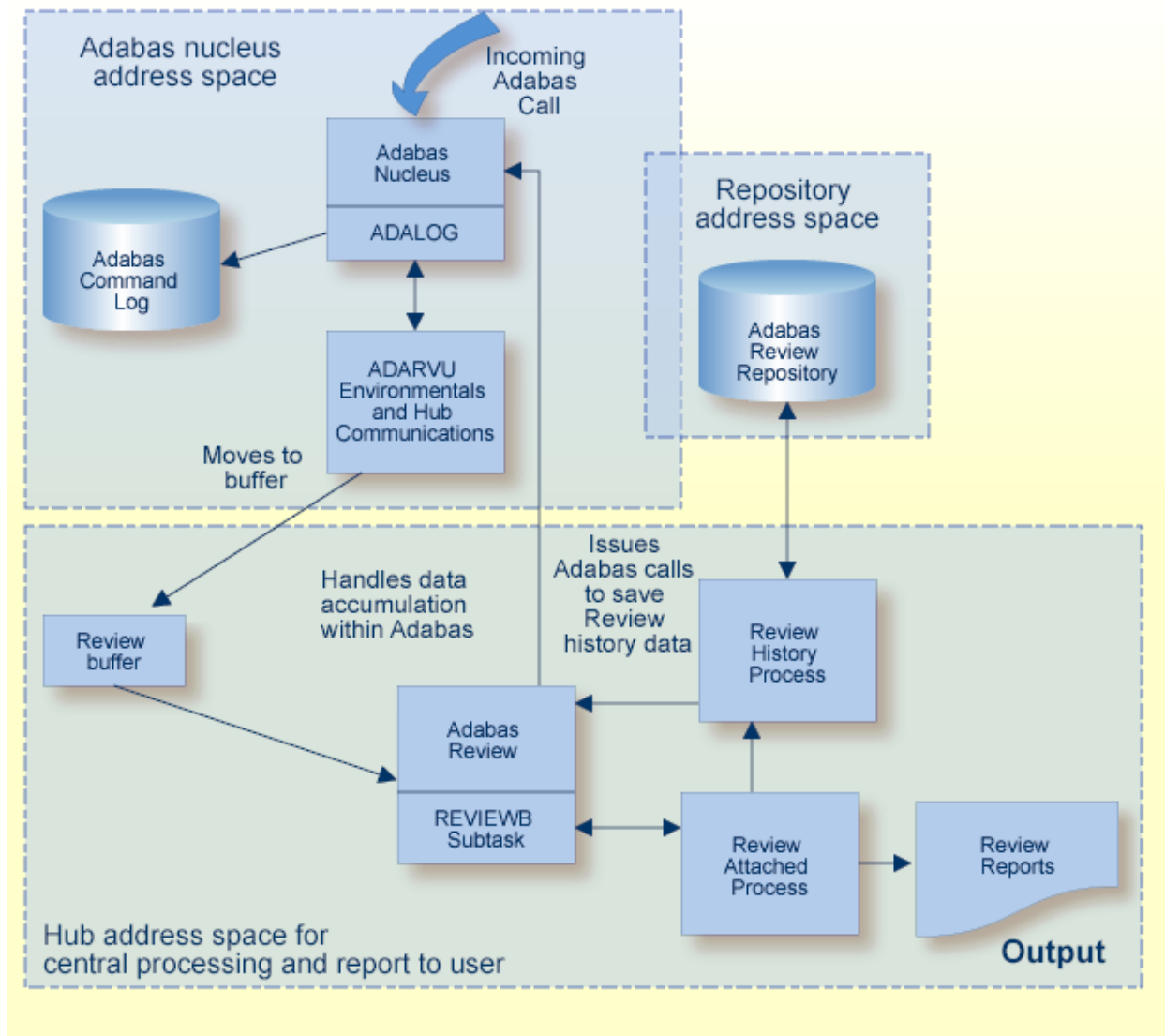
# 5   Collecting History Data

History data collection is controlled by RAOSHIST, the Adabas Review historical data population routine.

This chapter covers the following topics:

# OS/390, z/OS, VSE/ESA, and BS2000

In OS/390, z/OS, VSE/ESA, and BS2000 environments, RAOSHIST executes as a subtask of Adabas Review.

At initialization, RAOSHIST reads the RVUALT dataset to determine if there are any historical records from the previous Adabas Review session that should be written to the Adabas Review repository.

History records are written to RVUALT if the database on which the Adabas Review repository resides is unavailable during Adabas Review termination. This is always the case

- in local mode when the repository resides on the terminating database.
- in hub mode when the repository resides on a database that has already terminated.

During normal execution of Adabas Review, RAOSHIST is responsible for adding history records generated by detail history reports and by summary history reports that have a history interval.

# z/VM

In the z/VM environment, RAOSHIST is executed by an EXEC or a job, and adds records read from the RVUALT file to the Adabas Review repository file. All history records are written to the RVUALT file in z/VM.

# 6  Reports

Online or as a stand-alone batch job, Adabas Review processes Adabas command log records and generates reports according to user-defined reporting criteria. The flexible reporting structure of Adabas Review allows you to view the same data in many different ways.

Adabas Review uses a set of instructions called a "report definition" to specify the types of data to be collected. Prepared report definitions supplied with Adabas Review may be modified and custom reports may be created.

Report definitions can be created or modified using menu-driven Natural programs. Report options and processing rules allow you to specify the conditions under which the data is to be captured. Report definitions are kept in the Adabas Review repository.

When a report definition is saved, Adabas Review generates a Natural display program that is executed when you wish to view the collected data. This display program is executed

- to view data currently being collected by the interactive Adabas Review processor.
- to retrieve and display or download historical data that has been saved and stored in the Adabas Review repository.

> **Note:** No display program is created for buffer pool reports.

The Natural display program generates normal Natural `FIND` and `READ` statements against a Review DDM to access the data being collected by Adabas Review. By default, the Natural program displays the data at the Adabas Review user's terminal. Options exist, however, to download the data directly to a personal computer (PC).

Once the report definitions are edited and saved, the reports can be started. Starting a report tells the Adabas Review data collection process to start accumulating data based on the report definition parameters.

Adabas Review users can display buffer pool information, display active databases, and access the Adabas Online System, if it is available. Adabas Review administrators are also allowed to define and display target objects.

# 7 Autostarted Reports

Adabas Review reports can be set to start automatically whenever Review initializes.

Then RAOSAUTO, the Adabas Review autostarted report parameter generation routine, generates the report definition control statements and writes them to one of two parameter files, RVUAUT1 or RVUAUT2, alternating between them by writing to the older file.

When Adabas starts, the files are read by Adabas Review using the `RVUAUT1` and `RVUAUT2` statements in the job stream. Refer to the section *Batch Processor Job Control Requirements* for more information.

> **Note:** Under OS/390 and z/OS, the installation procedure defines the statements `RVUAUT1` and `RVUAUT2` so that they point to members of a PDS. To avoid constant compression of these datasets, the statements may point to sequential datasets.

RAOSAUTO automatically regenerates the control statements for all autostarted reports when you make changes to an autostarted report, delete an autostarted report, or modify the target definition for the database being monitored by the reports.

In exceptional circumstances (e.g., the source library becomes too full and requires compressing), you can force regeneration of the control statements for all autostarted reports by either issuing the `GENAUTO` command or entering the parameters manually using batch parameter statements.

Additionally, when you issue the `GENCARD` command, `RAOSAUTO` generates report parameter cards for user-specified reports and directs them to a user-specified output file.

In OS/390, z/OS, VSE/ESA, and BS2000 environments, `RAOSAUTO` executes as a subtask of Adabas Review and is only active when

- an autostarted report definition is saved in the online system;
- `GENAUTO` (or `GENCARD`) is executed in the online system; or
- a database target definition is saved in the online system.

In the z/VM environment, RAOSAUTO executes as an EXEC. The GENAUTO and GENCARD commands are not available in z/VM.

# 8 Repositories

The Adabas Review repository is an Adabas file used for storing report definitions, historical data, and target definitions. In hub mode, it cannot run in the hub address space.

Depending on the configuration at your site, more than one Adabas Review repository may be associated with your system. For example, if your site is running Adabas Review against more than one database, you may choose to have an Adabas Review repository for each database.

The Review commands `SETFILE` (or `SET`) and `SETALL` (or `SETA`) may be used to access different Adabas Review repositories and the reports stored on them.

# 9 User Profile System

Adabas Review administrators use the user profile system to generate profiles that define access rules for Adabas Review users. Access rules specify the functions that a particular user is allowed to use.

User profiles may be created for new users, changed for existing users, and purged when no longer required.

A user profile is not required for each user. Review provides a default profile to allow access for users who do not have a profile defined.

When a user logs on, Adabas Review searches for the user's profile. If one is not found, the default profile is used.

If the default profile is customized so that the access rules meet the needs of the majority of Adabas Review users, the need for individual user profiles can be eliminated.

If a user has access needs that are different from the majority, a user profile can be created to accommodate those needs. Such a profile is generated by customizing a copy of the default profile.

# 10 Storage Requirements

Adabas Review must allocate storage to execute. Storage is required for

- the Adabas Review hub, if used;
- the REVIEW-BUFFER, used as a queueing area for Adabas command log records;
- reports that are executing;
- users accessing the database from the Adabas Review online system; and
- work areas used in Adabas Review's processing.

The type, purpose, and size of these storage areas is discussed in the following sections.

Adabas Review allocates storage "above the line" whenever it is permitted by the architecture of the machine and the operating system on which it is executing.

In OS/390 and z/OS environments, Adabas Review allocates all storage from MVS subpool 5. This allows you to accurately determine the exact amount of storage Adabas Review is using with an MVS monitoring package.

This chapter covers the following topics:

## Storage for the Hub

Review 4.3 uses the high performance Trans-Port protocol as its primary method of sending data from the ADABAS nucleus to the Review Hub. When running in this mode, the `NC` and `NAB` ADARUN parameters mentioned in the paragraphs below are not particularly meaningful. The storage that `NC` and `NAB` control is used for Review administrative traffic only, and the workload from the ADABAS nuclei does not require the storage managed by these parameters. These two parameters can be left at the default values if all nuclei are using Trans-Port.

If you use Adabas Review in hub mode, the hub has a separate storage requirement for its operating queues and working areas. The queues are used to buffer the incoming command log records from the clients until the records can be sent to REVIEWB.

Two queues, both controlled by the database administrator (DBA), are used by the Adabas Review hub: the command queue (sized using the ADARUN parameter `NC`) and the attached buffer (sized using the ADARUN parameter `NAB`). See the *Adabas Operations Documentation* for more information.

## Command Queue

The command queue stores information about the client nucleus such as job name, internal ID, etc. Each entry in the command queue represents one command log record from a client if Trans-Port is not being used.

An entry exists for the time that a command log record is queued and awaiting selection from the hub until the time that the record is sent to REVIEWB. Once the command log record is sent to REVIEWB, the entry is released from the command queue. The command queue is also used for administrative calls from the Review user interface.

This means that the command queue must be large enough to accommodate the backlog of command log records from the client nuclei. If the command queue is too small, it is possible that command log records will be dropped by the hub.

The ADARUN parameter that controls the command queue size is `NC`. The value of this parameter should be set higher for the hub than it is for individual client nuclei.

The `NC` value should be set to handle the arrival rate based on

- the number of clients;
- their respective command processing limits;
- the processing power of the CPU(s); and
- the priority settings of the nuclei and hub address spaces, partitions, or regions.

**Example**

If a client nucleus can process 2000 command per second, then the expected arrival rate at the hub is also 2000 commands log records per second.

There is no general rule for estimating the NC requirements for a particular hub. However, in this example, you could start with NC=1000 and monitor the results.

## Attached Buffer

The attached buffer is used to store the contents of the command log records and their associated data extensions.

As with the command queue, an element within the attached buffer is allocated to hold the command log record for the duration of time that the record is queued for selection, up to the time the record can be sent to REVIEWB. The element is freed once the record is sent to REVIEWB.

Also like the command queue, the attached buffer must be large enough to hold the queued command log records for the time required to stage the records for REVIEWB. Software AG recommends setting the parameter high to ensure that command log data is not dropped by the hub.

The ADARUN parameter controlling the attached buffer size is `NAB`. The value of this parameter should also be set higher for the hub than it is for individual client nuclei.

The `NAB` value must be large enough to buffer the data passed by the client nuclei. The amount of data passed by a client nucleus depends upon the Adabas Review report requirements, i.e., whether control buffers are required, or the I/O list option is being used.

**Example**

The average size of a command log record and extensions, excluding control buffers, is 2500 bytes.

One approach would be to compute

```
NAB = (NC * 2500 / 4096)
```

where 4096 is the size of one `NAB` segment. If NC=1000 (see the example in section *Command Queue*), the starting value would be

```
NAB = (1000 * 2500 / 4096) = 610
```

This computation assumes that there are no control buffers or I/O list elements being passed to the hub.

## Storage for the REVIEW-BUFFER

REVIEW-BUFFER is used to queue Adabas command log records to be sent to REVIEWB. In hub mode, it is located in the hub (server) address space.

The `BUFFER-SEGMENTS` parameter specifies the size of the REVIEW-BUFFER. Each buffer segment is 128 bytes. The default value for `BUFFER-SEGMENTS` is as follows:

| Operating System | Default Value |
|---|---|
| OS/390 or z/OS | 2000 (250 kilobytes) |
| VSE/ESA | 700 (87 kilobytes) |
| z/VM or BS2000 | 400 (50 kilobytes) |

### For OS/390, z/OS, VSE/ESA, and BS2000

A larger REVIEW-BUFFER provides a larger queueing area for command log records being sent to REVIEWB and decreases the possibility that Adabas will have to wait for REVIEWB to process these records in the event that REVIEW-BUFFER becomes full.

### For z/VM

In the z/VM environment, the fact that REVIEWB does not execute as a subtask of Adabas Review but is called synchronously only when the REVIEW-BUFFER is full has the following consequences:

- A *larger* REVIEW-BUFFER means that REVIEWB is called less often but with more command log records to process. Since Adabas is unable to process incoming calls until REVIEWB has processed the contents of the REVIEW-BUFFER, a considerable delay may be experienced when REVIEWB is called.

- A *smaller* REVIEW-BUFFER means that REVIEWB is called more often but with fewer command log records to process so that the processing delay is less for each processing cycle.

> **Note:** The Adabas RVUEXI parameter `CMS-FULLSYNCH` may be used to force REVIEWB to process each command log instead of waiting for the REVIEW-BUFFER to become full.

## Storage for Reports

### For Control Blocks

When a report is started, either using autostarted report definition parameters or by an online Adabas Review user, storage is allocated for control blocks that define the criteria for the collection of the data.

Typically, the storage allocation for control blocks is two (2) kilobytes, but may be as much as four (4) kilobytes if the report is a history report or the report specifies the collection of many fields.

### For Data Collection Areas

In addition to the report control blocks, storage is allocated for the collection of data. The data collection areas are allocated in two (2) kilobyte pieces and a subsequent data collection area is only allocated when the current area is full.

**Total Storage Limit**

The total storage allocation for a report is limited by the MAXSTORE report parameter. When the total storage allocation for a report is equal to the MAXSTORE value, the report is marked as inactive and stops accumulating data. When a report is purged, all storage associated with the report is deallocated.

# Storage for Online Users

Adabas Review's online system uses Adabas calls to start, view, or purge a report. Each request requires that Adabas Review perform some processing to fulfill the request.

- Each request from the Adabas Review online system results in the allocation of a piece of storage (about 500 bytes or $^1/_2$ kilobyte) that is deallocated when the request has been satisfied.

- To maintain the integrity of each request, Adabas Review allocates an area for each user requesting Adabas Review to service a request. For example, a request to view a report requires an Adabas call for each record that is to be viewed online.

- In cases where more than one user is viewing the same or different reports, Adabas Review must remember the status of each user between Adabas calls.

# Storage for Work Areas

Adabas Review allocates storage for work areas and areas used for reading from and writing to files. These areas are typically small and are kept and used throughout the time that Adabas Review is active.

# 11 Summary of Features

The major features of Adabas Review include the following:

- **Support for Adabas 7, Adabas Parallel Services, and Adabas Cluster Services**
  Adabas Review supports multiple Adabas nuclei clustered under a single database ID.

- **Multiple database monitoring**
  Adabas Review can monitor several databases (DBIDs) and provide reports for each.

- **Client/server technology**
  Adabas Review can be configured in a straight-forward, uniform, and consistent way under all targeted operating system platforms.

- **Centralized data collection, reporting, and control**
  The hub is a true data collection point unencumbered by dependencies on internal operations and structures of the Adabas nucleus.

- **Reduced overhead**
  (hub mode) with the elimination of the Review DB subtask from the Adabas address space, partition, or region.

- **Customized command logging**
  Adabas Review can perform physical command logging of selected commands and/or buffers based on parameters within the Adabas Review report definitions. Each command log corresponds to a particular report definition, and contains only those records that have been selected as meaningful for the end user.

- **Customized analysis tools**
  Adabas Review reports may be customized to perform functions such as cost accounting and application debugging.

- **History data collection**
  Data collected for Adabas Review reports can be written to the Adabas Review repository at specified intervals and stored as history data. The data can be used in trend analysis.

■ **Supplied reports**
More than 40 prepared report definitions are provided with Adabas Review. These report definitions may be used without modification, or edited to suit specific site requirements.

■ **Report creation function**
The Edit Report function provides interactive screens to help you create a report or edit an existing one.

■ **Wide range of data types**
More than 340 data fields are available for specifying the types of data to be collected by Adabas Review.

■ **User-defined data fields**
Version 4.2 provides a mechanism for defining up to five data fields to meet user specifications.

■ **Online data retrieval**
Adabas Review reports are executed online or in batch mode. Data collected by Adabas Review reports may be viewed online.

■ **Online display of Adabas availability**
The Available Adabas Nuclei (AA) function displays a profile of the databases that are monitored by Adabas Review.

■ **Access to Adabas Online System**
Adabas Review provides access to Adabas Online System, an online Adabas maintenance product.

# Index