

Adabas Is . . .

Adabas, the adaptable data base, is a high-performance, multithreaded, database management system for mainframe platforms where database performance is a critical factor. It is interoperable, scalable, and portable across multiple, heterogeneous platforms including mainframe, midrange, and PC.

This chapter covers the following topics:

- Operational Highlights
 - Operating Environments
 - Supported Data Models
 - Operating Structure
 - Executing Adabas
-

Operational Highlights

High Availability

Adabas is designed for operation 7 days a week and 24 hours a day. Space is managed dynamically (read *Adabas Space Management*), files can be loaded and unloaded, backed up and restored, and system performance can be analyzed without interrupting the active database.

Storage Space Optimization

Adabas stores data in compressed form to reduce space requirements. Since modern databases are measured in gigabytes (1000 megabytes) or even terabytes (1000 gigabytes), the savings in disk space can be considerable. Reduced space requirements also mean that the input/output (I/O) system is more efficient.

Performance

Performance is the key factor of Adabas, which includes a number of features to enhance it. For instance, a number of set-up parameters are available for fine-tuning the database operating environment, and many of these can be modified while the database is active.

Fault Tolerance

Adabas recovers automatically after an abnormal database or system termination. Each time an Adabas database is started, an automatic check is initiated to determine whether the database previously terminated cleanly or an active transaction was interrupted. If a transaction was interrupted, Adabas automatically resets all changes of the uncompleted transaction so that the database is consistent.

Operating Environments

Adabas 8 for mainframes supports the following operating environments:

- SNI's BS2000
- IBM's z/OS, VSE, and z/VM

Mainframe Adabas can be used in distributed environments with Adabas on:

- Digital's VAX and Alpha AXP under OpenVMS
- IBM's AS/400 under OS/400
- Wang under VS
- IBM's PC and compatibles under OS/2 or Windows/NT
- a variety of supported UNIX platforms

As the telecommunication interface, mainframe Adabas supports TP monitors such as Software AG's Com-plete; and other popular monitors such as TSO, CICS, IMS/DC, TIAM, and UTM.

Software AG's multisystem processing tool Entire Net-Work provides the benefits of distributed processing by allowing you to communicate across a network with Adabas and other service tasks.

Support for network access methods is implemented in the form of line drivers. Mainframe Entire Net-Work provides drivers for VTAM, IUCV, DCAM, CTCA (channel-to-channel adapter), TCP/IP, and XCF.

Supported Data Models

Adabas is a relational-like database in that:

- it stores information in tables in which rows represent individual data *records* and columns represent *fields*; and
- separate Adabas files can be linked logically by a common field.

Adabas differs from true relational databases in that it:

- stores many data relationships physically, resulting in fewer demands on CPU resources than true relational databases, which create all relationships logically at runtime.
- supports repeating groups of fields.

Adabas separates data relationships, management, and retrieval from the actual physical data and stores the physical data independently. It provides flexible access techniques and performs both simple and complex searches quickly and efficiently. The independence of the data from the program minimizes the need to reprogram when the database structure changes.

Logical data relationships can be created as needed. Adabas can accommodate any representational and access requirements dictated by the user environment. Each individual corporate user can decide how to view data in the system, and can alter data relationships dynamically-without altering the database or existing programs.

In contrast to systems that require a single model for all data, Adabas allows you to choose any structure your application requires. You can access the same data using your choice of data-model perspective:

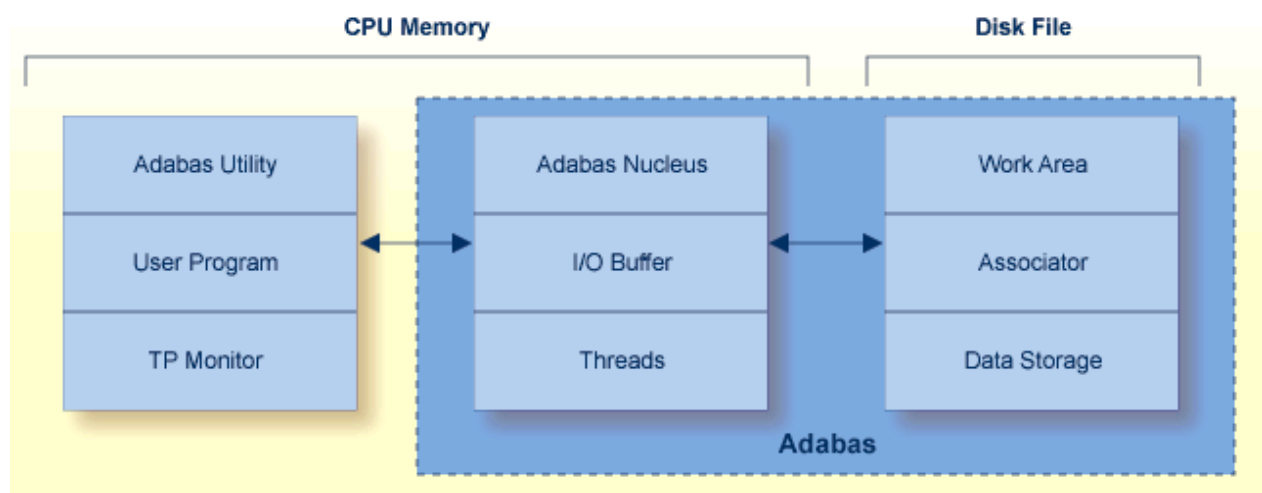
- relational including nested relational (tables within tables)
- entity relationship, with proven ability to support structural objects
- hierarchical; network
- geographical
- text

These data models can be combined within a single business solution; multiple solutions can view Adabas data using different data models.

As new requirements develop, Adabas evolves in both scope and complexity without redesign of the database or reprogramming of application systems. For example, field and access keys may be added to an Adabas file at any time without reloading or reorganizing the file.

Operating Structure

The following figure shows the operating structure of the Adabas system.



Nucleus, I/O Buffer, and Threads

The Adabas nucleus and input/output (I/O) buffer are loaded into main memory at startup. The *nucleus* is a set of programs that drives Adabas, coordinates all work, and translates user program statements into Adabas commands. All programs access Adabas files through the nucleus. All database activities such as data access and update are managed by the Adabas nucleus. In most cases, a single nucleus is used to manage a single physical database.

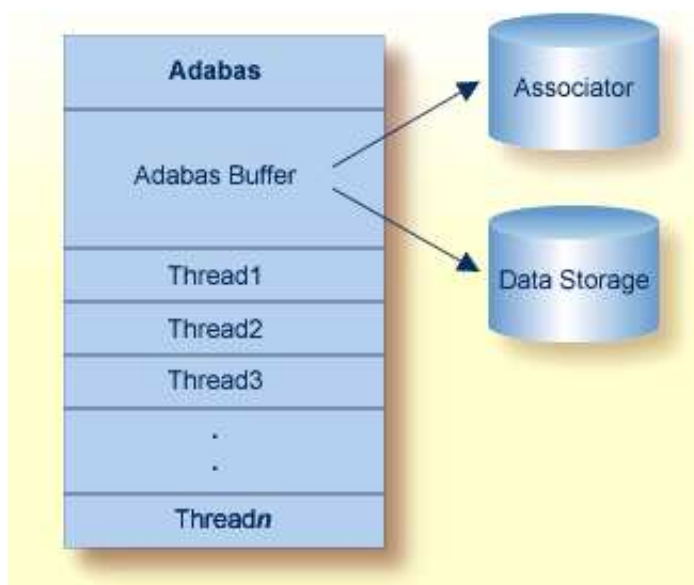
Note:

For information about running multiple nuclei against a single physical database under a single operating system image, read *Adabas Parallel Services* or under multiple z/OS images, read *Adabas Cluster Services*.

The Adabas I/O buffer area, which can be resized for each Adabas session, contains the most frequently used data and data relationships; it helps to minimize physical input/output (I/O) activity and thus saves computer time. It contains blocks read from the database and blocks to be written to the database:

- For blocks read from the database, a buffer algorithm ensures that the most frequently accessed blocks stay in memory. When a block from the database is needed, the buffer content is checked to determine if the block is already in memory, thus avoiding unnecessary reads.
- Multiple updates are accumulated in a block before being written (flushed) to the database.

Adabas provides multithreaded processing to maximize throughput. If I/O activity suspends command processing in an active thread, Adabas automatically switches to another thread. The user may set the number of 8-kilobyte threads to be used for an Adabas session up to a maximum of 250.



Data Storage, Associator, and Work

The Data Storage, Associator, and Work components are physical disk areas:

- Data Storage contains raw data, generally in compressed form.
- The Associator contains information about data relationships.
- The Work area contains the data protection area and temporary storage for intermediate results during complex search operations or distributed transaction processing.

Read *Adabas Design* for more information about these database components.

Utilities, User Programs, and TP Monitors

Utilities

Database services such as loading or deleting files are handled by an integrated set of online and batch-mode utility programs. Most utilities can be run in parallel with normal database activity to preclude interruption of daily production.

Adabas utilities provide initial design and load operations, backup/restore/recovery routines, database modification routines, and audit/control/tuning procedures. Read *Adabas Utilities* for a brief explanation of each utility.

User Programs

The nucleus is called using a batch or online user program written in:

- Natural, Software AG's fourth-generation application development environment, or some other fourth-generation language
- Assembler, or a third-generation programming language such as FORTRAN, COBOL, or PL/I (the REXX/VSE interpreter language is also supported) that uses the powerful and flexible Adabas direct call interface. Each Adabas call is accompanied by a parameter list identifying buffers defined in the user program that are used to transfer information to and from Adabas.

Note:

Read *Adabas Native SQL* for information about Adabas Native SQL, a precompiler for Ada, COBOL, FORTRAN, and PL/I programs. Read *Adabas SQL Gateway* for information about Adabas SQL Gateway, an SQL interface to Adabas.

Data from VSAM, DL/I, IMS/DB, SESAM, or TOTAL database structures can be transferred to and stored in Adabas using Adabas bridge products. The original, unmodified programs continue operating with their original data access commands while the bridge products intercept the data access commands and translate them to Adabas direct calls. Read *Adabas Bridges* for more information about Adabas bridges.

Special User Programs: Triggers and Stored Procedures

The Adabas triggers and stored procedures facility, an integral part of Adabas, can be used with Natural (read *Natural Application Development Environment*) to write and manage *triggers* and *stored procedures* in the Adabas server environment. *Triggers* are systematically used programs that are started automatically based on an event; they can be used to ensure referential integrity, for instance. *Stored procedures* are programs used by a number of different clients that are executed by Adabas as a result of a special user call. Storing these programs in an Adabas file on the server reduces the amount of data traffic to and from the server. Read *Using Triggers and Stored Procedures* for more information about Adabas triggers and stored procedures.

TP Monitors, Adalinks, and the Adabas API

Since most systems do not allow a standard call to Adabas, Software AG provides an application program interface (API) to translate calls issued by an application program into a form that can be handled by Adabas. The Adabas API is available across all supported mainframe platforms for both batch and online operations.

Online operations are controlled by teleprocessing (TP) monitors, which serve as telecommunication interfaces to Adabas. Supported TP monitors are listed in *Operating Environments*. Software AG provides versions of the Adabas API that are specific to particular TP monitors. *Adalink* is a generic term that refers to the portion of the API that is specific to a particular TP monitor.

Batch applications are supported in both single-user and multiuser mode (read *Modes of Operation* for a discussion of these modes). The Adabas batch API uses a standard calling convention that is supported by all major programming languages through their CALL mechanisms. Most mainframe operating systems allow batch application modules to be linked either with the batch API or with ADAUSER.

Software AG strongly recommends linking batch application programs with the Adabas version-independent module ADAUSER. The ADAUSER module can optionally be linked with the Adabas API. ADAUSER provides upward compatibility with Adabas releases and a degree of isolation from future changes to the API or to mechanisms that handle interregion communication between the user and the nucleus (read *Modes of Operation*).

A client running under IBM's OpenEdition can access Adabas. An OpenEdition application containing calls to Adabas can be linked with either the batch API or ADAUSER.

Executing Adabas

- Session Types
- Storage Areas
- Modes of Operation
- ADARUN Startup Parameters
- Session Control

Session Types

Three types of sessions can be identified for Adabas:

- The *Adabas session* starts when the nucleus is invoked and ends when the nucleus is terminated. An Adabas nucleus is invoked using job control specific to a particular operating system that contains Adabas startup, or ADARUN, parameters.
- A *user session* is either a batch mode program or a person using a terminal. A user session can occur only during an Adabas session; that is, when the Adabas nucleus is active. It is a sequence of Adabas calls optionally starting with an open user session (OP) command and ending with a close user session (CL) command.
- A *utility session* is executed in batch, or online using the Adabas Online System. Some utilities require the Adabas nucleus to be active; others do not. ADARUN startup parameters are also used for executing utilities.

Storage Areas

The Adabas nucleus and each user program or Adabas utility is executed in a separate storage area defined by the operating system. The name of the storage area depends on the operating system:

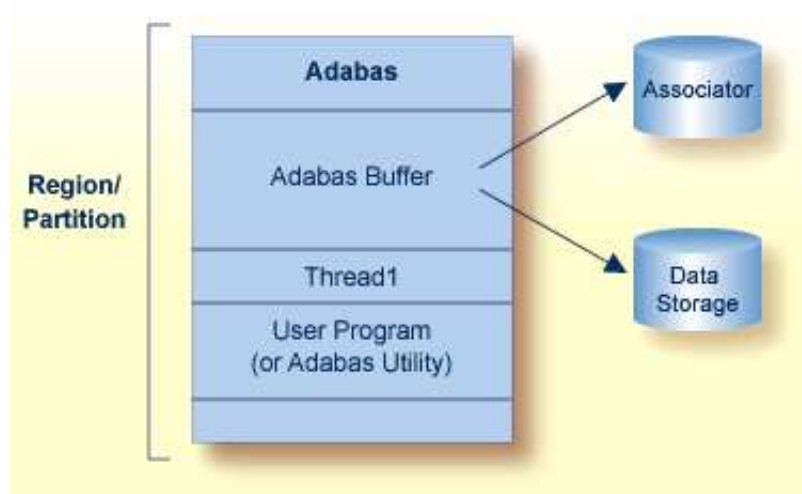
BS2000	task
z/OS	address space, data space, hiperspace, 64-bit virtual space
z/VM	virtual machine
VSE	partition, address space, data space

For consistency and simplification, Adabas documentation refer to all non-VM areas (task, address space, partition, etc.) as *regions*. VM areas are called *virtual machines*.

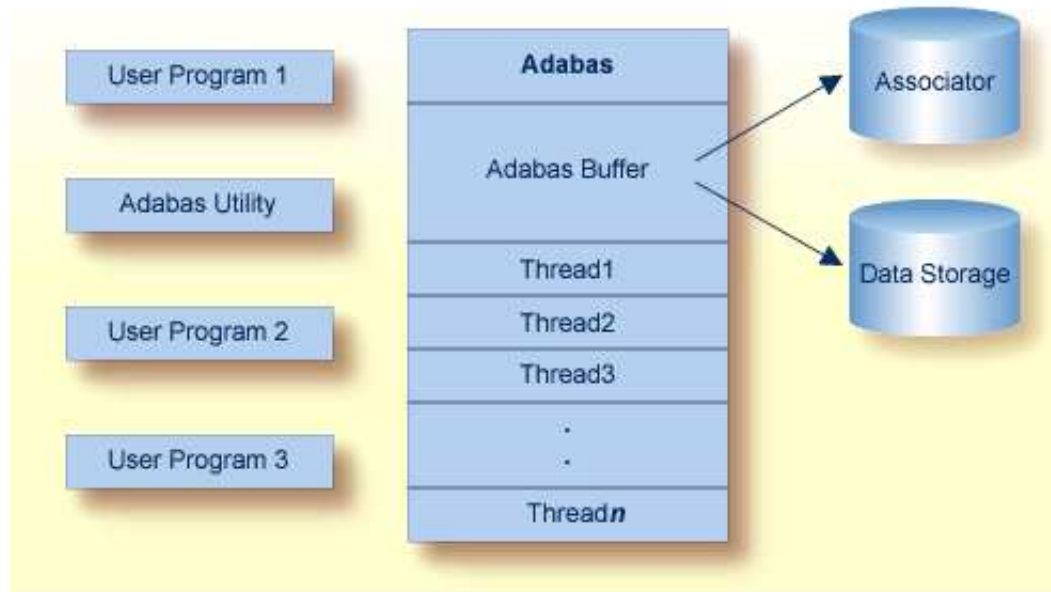
Modes of Operation

Adabas supports two modes of operation: single-user and multiuser.

Single-user mode is in effect when a user program (or Adabas utility) is executed in the same partition/region as the Adabas nucleus.



Multiuser mode is in effect when the Adabas nucleus is located in a separate partition/region. It is the most efficient and therefore the recommended mode of operation.



When using Adabas in multiuser mode, interregion communication is handled by Adabas in a manner that takes optimum advantage of the communications facilities offered by the various operating systems.

For single-user mode, the appropriate Adabas nucleus JCL must be included with the job control for the utility or user program.

ADARUN Startup Parameters

The ADARUN control statement defines and starts the Adabas operating environment. The ADARUN control statement also starts Adabas utilities. ADARUN

- loads the ADAIOR module, which performs all database I/O and other operating-system-dependent functions;
- interprets the ADARUN parameter statements; then loads and modifies the appropriate Adabas nucleus or utility modules according to the ADARUN parameter settings; and
- transfers control to Adabas.

The ADARUN statement, normally a series of entries each specifying one or more ADARUN parameter settings, is specified in the DD CARD (z/OS, z/VM, or BS2000) or CARD (VSE) data set.

Session Control

Adabas provides several ways to monitor and control Adabas, user, and utility sessions:

- Adabas *operator commands* can be entered from the operator console during an Adabas session or during utility operation.
- The ADADBS OPERCOM utility function can issue operator commands to the Adabas nucleus. Adabas then issues a message to the operator confirming the command execution.

- For those using Adabas Online System (demo or full version), you may be able to execute functions corresponding to operator commands while an Adabas session is active using menu options or direct commands.

Operator commands can be used to terminate an Adabas or user session; display nucleus or utility information; log commands; and change Adabas operating parameters or conditions.

Adabas *direct call commands* can also be used to open and close a user session. Read *Direct Call Interface* for more information about direct call commands.