



# Glossary

---

## Adabas Version 7.4.2



This document applies to Adabas Version 7.4.2 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.

© Copyright Software AG 1999-2003.  
All rights reserved.

The name Software AG and/or all Software AG product names are either trademarks or registered trademarks of Software AG. Other company and product names mentioned herein may be trademarks of their respective owners.

# Table of Contents

Glossary . . . . .	1
A . . . . .	1
B . . . . .	2
D . . . . .	2
F . . . . .	3
G . . . . .	4
I . . . . .	5
N . . . . .	5
O . . . . .	5
P . . . . .	6
R . . . . .	6
S . . . . .	7
T . . . . .	7
U . . . . .	8
W . . . . .	8



# Glossary

Glossary entries exist for the following starting letters:

---

## A

### Adabas

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.

### Adalink

A generic term for that part of the Adabas API (application program interface) that is specific to a particular teleprocessing (TP) monitor. The Adabas API is used to link application programs to Adabas. The actual module name depends on the TP monitor being used; for example, the module name for linking to a batch or TSO program is ADALNK, and for CICS, the module name is ADALNC. The term "Adalink" refers to the module appropriate for the given environment. The terms "Adalink(s)" and "ADALNK(s)" are synonyms.

### ADARUN

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 DDCARD (OS/390, MVS/ESA, MSP, VM/ESA, or BS2000) or VSE/ESA CARD dataset.

### Address converter

Adabas stores each database record in a Data Storage block identified by a relative Adabas block number (RABN). Each record's RABN is kept in a table called the address converter. The address converters, one for each database file, are stored in the Associator. Address converter entries are in ISN order (that is, the first entry tells the RABN location of data for ISN 1, the 15th entry holds the RABN location of data for ISN 15, and so on).

### Address space

The storage area assigned to a program task/work unit. In OS/390, MVS/ESA, or MSP, an address space is a region; in VSE/ESA, a partition; and in BS2000, a task. In this manual, the term "region" is used as a synonym for "partition" and "task".

## B

### Buffer flush

Associator and Data Storage blocks in the buffer pool that have been updated since the last "buffer flush" have write flags set "on". When the buffer is "flushed", these blocks are written to Associator and Data Storage datasets, respectively. The "flushed" blocks remain in the buffer pool with their write flags set "off".

A buffer flush can be synchronous or asynchronous. Update commands cannot be selected during a synchronous flush, but can be selected during an asynchronous flush.

## D

### Data compression

Data compression significantly reduces the amount of storage required. It also permits the transmission of more information per physical transfer, resulting in greater I/O efficiency.

Adabas retains data records in compressed form. It defines and executes compression at the field level. Three compression options are supported: default compression, null suppression, and fixed format. The last two options are added as field options.

*Default compression* deletes trailing blanks in alphanumeric fields and leading zeros in binary fields. An *inclusive length byte* (ILB) at the beginning of the field indicates the total number of stored bytes, including the ILB. Thus, if "Susan" is entered in a "first-name" field defined with a 20-character length and default compression, its stored size will be six bytes: five bytes for the letters of the name, plus one byte for the ILB. In addition, empty fields in a record are not stored; an empty field is replaced by a one-byte *empty field counter* (EFC). Adabas can store up to 63 contiguous empty fields in a single hexadecimal byte.

*Null suppression* (NU field option) adds to default compression in that searches on descriptor fields defined with null suppression do *not* return records in which the descriptor field is empty.

Fields defined as *fixed format* (FI field option) do not include a length byte and are *not* compressed. This option actually saves storage space for one-byte fields or fields that are nearly always full (e.g., a field containing the social security number).

### Database

In Adabas, a "database" is a group of related files.

A *physical database* identified by its database ID number (DBID) is defined with Adabas utilities. A *single physical database* is one set of Associator and Data Storage datasets identified by a single DBID. An Adabas nucleus running in an address space allows access to the physical files in the physical database.

## Database Administrator (DBA)

Controls and manages the database resources. Tasks include defining database distribution, assigning a structure and resources, creating and maintaining programming and operation standards, ensuring high performance, resolving user problems, defining and teaching user training, controlling database access and security, and planning for growth and the integration of new database resource applications and system upgrades. Also known as the Database Analyst.

## Descriptor

A descriptor is a search key. A *unique descriptor* has a different (i.e., unique) value for each record in the file. Entries are made in the Associator's inverted list for descriptor fields, adding disk space and processing overhead requirements.

Any field can be used within a selection criterion. When a field that is used extensively as a search criterion is defined as a descriptor (key), the selection process is considerably faster since Adabas is able to access the descriptor's values directly from the inverted list without reading any records from Data Storage.

A descriptor field can be used as a sort key in a search command, as a way of controlling a logical sequential read process (ascending or descending values), or as the basis for file coupling.

A portion of a field may be defined as a *subdescriptor*; combinations of fields or portions thereof may be defined as a *superdescriptor*; a user-supplied algorithm may be the basis of a *hyperdescriptor* or a *collation descriptor*; and a "sounds-like" encoding algorithm may be the basis of a *phonetic descriptor*, which may be customized for specific language requirements. For more information, see *Adabas Design in the Concepts and Facilities*.

## F

### Field

In Adabas, a "field" is the smallest logical unit of information (e.g., current salary) that may be defined and referenced by the user.

Adabas supports four field types:

	Single Value per Record	Multiple Values per Record
Single Field	Elementary	MU
Multiple Fields	Group	PE

The two basic field types are elementary and multiple-value. An *elementary* field has only one value per record. A *multiple-value* (MU) field can have up to 191 values, or occurrences, in a single record. Each multiple-value field has a *binary occurrence counter* (BOC) that stores the number of occurrences.

When two or more consecutive fields in the FDT are frequently accessed together, you can reference them by defining a *group* field. Other than its level and Adabas short name, a group field has no attributes defined. It immediately precedes its member fields in the FDT. A higher field *level* number is used to assign the member fields to the group field. Adabas supports up to seven field levels.

A *periodic* (PE) group field defines consecutive fields in the FDT that repeat together in a record. Like the members of a normal group field, PE members immediately follow the PE group field, have a higher level number than the PE field, and can be accessed both individually and as a group. Each PE has a BOC that stores the number of occurrences.

### Field definition table (FDT)

A table that defines each file's record structure and content. There is one FDT for each database file. FDTs, stored in the Associator's fixed area, have three parts: the first is a list of the file's fields in physical record order, the second part is a "quick index" to the records in the first part, and the third part defines the files sub/superfields and sub-/super-/hyper- and phonetic descriptors.

A portion of a field ( *subfield* ) or any combination of fields ( *superfield* ) may be defined as an elementary field. Subfields and superfields may be used for read operations only. They may only be changed by updating the original fields.

### File

In Adabas, a "file" is a group of related records that have the same format (with some exceptions; see *Adabas Design* in the *Concepts and Facilities*).

The disk storage space allocated to a single Adabas database is segmented into *logical* Adabas files. A certain part of the overall space within the database is allocated to each logical file. When this space is filled with records from that file, Adabas automatically allocates more space to the file from the common free space pool. This dynamic space allocation, together with the dynamic recovery of released space, allows Adabas databases to run without intervention for long periods of time.

A *physical* Adabas file contains database records. Each physical file is identified by a file number. The number of physical files (and physical file numbers) per physical database is limited to 5000 or one less than the ASSOR1 block size, whichever is lower.

An *expanded file* is a logical file comprising physical files in one or more locations. The physical files have the same field definition table (FDT), but non-overlapping ISN ranges. The data content of at least one field (the field value criterion) determines the physical file in which a data record is located.

A *multiclient file* is an Adabas file with records accessible through an owner ID. Only records identified by the same individual or group owner ID can be accessed or updated by the related user. This allows the file to be maintained as a single Adabas file, but to be used as multiple logical files (each record group belonging to an owner ID is a "logical file"). "Super" owner IDs allow access to all records in the file.

## G

### Global transaction

A "global" transaction is a unit of work that involves changes to resources under the control of more than one database operating in one or more operating system images.

# I

## Internal sequence number (ISN)

Every Adabas record is assigned an internal sequence number (ISN) to identify the record. Each record keeps its original ISN, regardless of where it is located.

Records in a physical database file have four-byte ISNs ranging from MINISN to MAXISN. In replicated files, a record has the same ISN in all file copies. In partitioned files, the ISN ranges are non-overlapping for each physical file.

## I/O buffer

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 is loaded into main memory at startup, along with the Adabas nucleus.

The buffer 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 it is written ("flushed") to external storage.

# N

## Nucleus

The "nucleus" is a set of programs that drive Adabas, coordinate all work, and translate 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.

Adabas Parallel Services makes it possible to run a cluster of up to 31 Adabas nuclei on a single operating system against a single database. Adabas Cluster Services supports the IBM parallel sysplex environment making it possible to run a cluster of up to 32 Adabas nuclei on multiple operating systems set up as a sysplex. Again the cluster runs against a single database.

### Note:

See *Optional Extensions* in the *Concepts and Facilities* for information about running multiple nuclei against a single physical database under a single operating system image (ADASMP) or under multiple OS/390 or MVS/ESA images (Adaplex+).

# O

## Operator commands

Adabas operator commands are entered during an Adabas session or during utility operation to

- terminate an Adabas or user session;
- display nucleus or utility information;
- log commands into CLOG;
- change Adabas operating parameters or conditions.

## P

### Procedure

A procedure is a Natural subprogram that is written and tested using standard Natural facilities. The same types of parameters are passed to the subprogram whether it is a *trigger* or a *stored procedure*.

## R

### RABN (relative Adabas block number)

The basis of Adabas storage addressing. Adabas divides Data, Associator, and Work disk space into device-dependent logical blocks. The blocks in each of the three areas are numbered consecutively in ascending sequence beginning with RABN 1. The data blocks themselves as well as their addresses are referred to throughout Software AG publications as "RABNs". In other words, the sentence, "Adabas assigns RABNs 1-10 to the Associator" means ten Adabas storage blocks numbered 1-10 are assigned-not just the block numbers, whereas "Adabas assigns 50 RABNs to the Associator" means 50 blocks of storage with unspecified RABN numbers is assigned.

### Record

In Adabas, a "record" is a collection of related fields that make up a complete unit of information (e.g., all the payroll data for a single employee).

### Record buffer

The portion of the calling program's parameter area, called the user buffer, that contains the data transferred during Adabas read, search, and update operations. When reading data field definitions, Adabas also returns the field definition information in the record buffer.

### Region

This manual uses "region" to collectively refer to storage space allocated to user jobs by OS/390, MVS/ESA, MSP, VSE/ESA, and BS2000 operating systems.

### Router

A central routine for communication within the boundaries of one operating system. The routine is called by users with Adalink routines, and by targets with ADAMPM. The router's main purpose is to transfer information between the Adalink and Adabas. The router also maintains the ID table. VM/ESA and BS2000 environments divide router functions among Adalink or other Adabas functions. The Adabas SVCs in OS/390, MVS/ESA, MSP, and VSE/ESA are examples of routers.

## S

### Service

A processor of Adabas calls and issuer of replies. An Adabas nucleus is an example of a service. See also *target*.

### Session

A *user session* is a sequence of Adabas calls optionally starting with an OP command and ending with a CL command. A *user* is either a batch mode program or a person using a terminal. The uniqueness of each user is assured by the user ID, a machine, an address space, and a terminal ID.

An *Adabas session* starts when Adabas is activated and continues until Adabas is terminated. During this time, the Adabas nucleus creates a sequence of protection entries in exact historical sequence reflecting all modifications made in the database. The sequence of protection entries is written to the Work dataset (part 1) and to a protection log in blocks. Each block contains the nucleus session number, a unique block number, and a time stamp.

### Stored procedure

A "stored procedure" is a procedure executed by Adabas, but invoked directly or manually by an application.

## T

### Target

A receiver of Adabas calls. A target maintains a command queue, and communicates with routers using ADAMPM. A target is also classified as a *service*. The Adabas nucleus is a target.

### Thread

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.

### Transaction

Adabas data protection, recovery, and user restart is based on the concept of a "logical transaction": the smallest unit of work (as defined by the user) that must be performed in its entirety to ensure that the information contained in the database is logically consistent.

A logical transaction may comprise one or more Adabas commands that together perform the database read/update required to complete a logical unit of work. A logical transaction begins with the first command that places a record in hold status and ends when an ET (end transaction), BT (back out transaction), CL (close), or OP (open) command is issued for the same user.

The ET command must be issued at the end of each logical transaction. Successful execution of an ET command ensures that all the updates performed during the transaction are physically applied to the database, regardless of subsequent user or Adabas session interruption.

Updates performed during transactions for which ET commands are not successfully executed are backed out, either manually by issuing the BT command or automatically by the Autobackout routine (see *Using Adabas in the Concepts and Facilities*).

**Translator**

A process that converts a logical ID of a user's Adabas call into a corresponding physical ID for a target.

**Trigger**

A procedure that is executed automatically by Adabas when a specified set of criteria is met. The set of criteria is determined for each Adabas command sent to the DBMS and is based on the target file number and optionally the command type and/or field. The command type refers to the commands FIND, READ, STORE, UPDATE, and DELETE. The field must be in the corresponding format buffer of the command.

**Two-phase commit**

"Two-phase commit" processing ensures commercial transaction integrity by securing or rejecting transactions as a whole across separately managed resources.

**U****User**

A batch or online application program that generates Adabas calls and uses an Adalink for communication.

**W****Work load balancing**

Adabas Parallel Services attempts to balance the work load evenly across the cluster nuclei based on the number of users and the number of commands processed by each of the nuclei.