Optional Product Extensions

The add-on products discussed in this chapter are available to Adabas customers who have exercised a separate purchase agreement for the feature or product.

This chapter covers the following topics:

- Adabas Bridges

- Adabas Caching Facility

- Adabas Cluster Services

- Adabas Delta Save Facility

- Adabas Fastpath

- Adabas Native SQL

- Adabas Online System

- Adabas Parallel Services

- Adabas Review

- Adabas SQL Gateway

- Adabas SQL Server

- Adabas Statistics Facility

- Adabas Text Retrieval

- Adabas Transaction Manager

- Adabas Vista

- Data Archiving for Adabas

- Event Replicator for Adabas

- Entire Net-Work Multisystem Processing Tool

- Entire Transaction Propagator

- Natural Application Development Environment

- Predict Data Dictionary System

Adabas Bridges

Adabas Bridge technology allows you to access the DL/I (and IMS/DB) and VSAM application development environments efficiently. Emulation requires no modifications to application programs and the delay and expense of traditional conversion are avoided.

Note:
Solutions are also available for TOTAL and SESAM.

Adabas Bridge technology provides

- user application transparency (it is not necessary to change any existing application programs or third party application software using native VSAM or DL/I calls);

- support for batch and online processing environments and the RPG, COBOL, PL/I, FORTRAN, and Assembler programming languages;

- data and application integrity.

Adabas Bridge for VSAM

Adabas Bridge for VSAM (AVB) allows application software written to access data in the VSAM environment to access data in an Adabas environment. It operates either in batch mode or online (under CICS) and is available for z/OS and z/VSE operating environments.

AVB may be executed in an environment with Adabas files only; VSAM files only; or both Adabas and VSAM files (mixed environments). The ability to operate in a mixed environment means that your migration schedule can be tailored to your needs and resources. You can migrate files from VSAM to Adabas as needed, one application or even one file at a time.

AVB uses a transparency table to map the names and structures of VSAM files to the numbers and structures of corresponding Adabas files. Once a VSAM file has been migrated to Adabas and defined in the AVB transparency table, it can be bridged to Adabas. When a VSAM file is bridged, AVB converts each request for the VSAM file to an Adabas call; the Adabas file is accessed instead of the VSAM file.

When AVB is active, it intercepts each file OPEN and CLOSE request and performs a series of checks to determine whether to process the request against an Adabas file. If not, AVB passes the request to the operating system to open the referenced VSAM file.

When AVB detects an OPEN or CLOSE for a bridged file, it converts it to an Adabas command and calls Adabas to open or close the corresponding Adabas file. After the OPEN, all requests to read or update the VSAM file are passed directly to AVB.

AVB allocates VSAM control blocks and inserts information the application needs to process results as if they were returned from a standard VSAM file request. After the Adabas call, AVB returns the results to the application using standard VSAM control blocks and work areas.
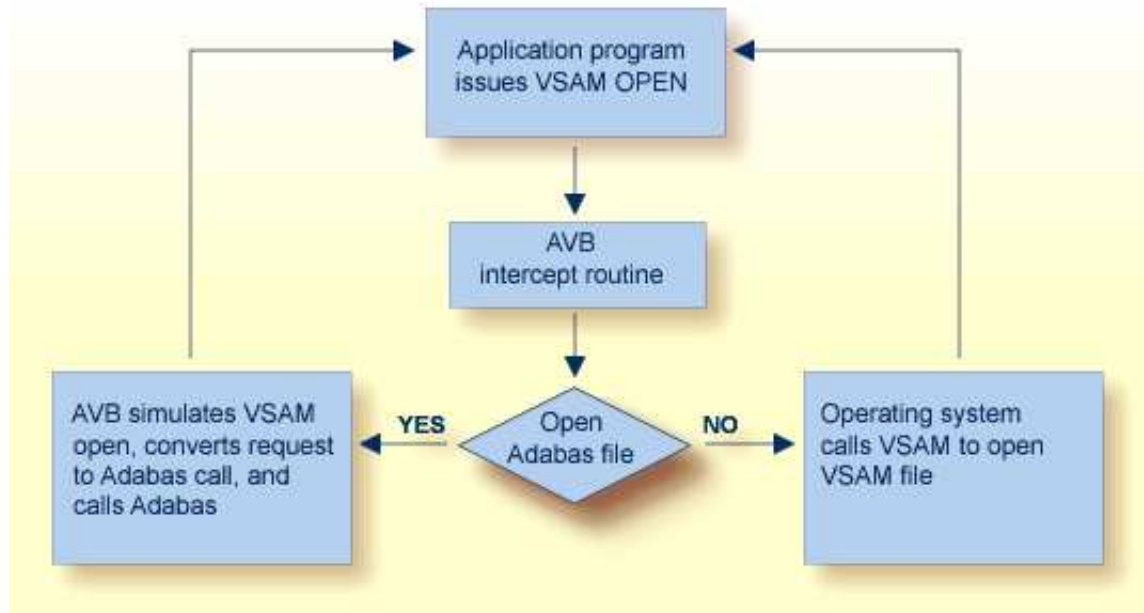
Advantages of Adabas over VSAM

The availability of Adabas in an environment in which only VSAM file structures were previously used results in the following benefits:

● Applications can be extended with the powerful indexing facilities of Adabas. You can query, retrieve, and manipulate data using efficient views and paths.

● Applications can be extended with programming languages such as Natural and SQL.

● Application programs are independent of the data structure, which reduces maintenance costs and increases programmer productivity.

● Automatic restart/recovery ensures the physical integrity of the database in the event of a hardware or software failure.

● Data compression significantly reduces the amount of online storage required and allows you to transmit more information per physical I/O.

● Security is improved by password protection at both the file and field levels and, on the basis of data values, at the record level as well.

● Adabas provides encryption options, including a user-provided key that drives the encryption process.

After migration, your application programs have the same view of data as before, but you can structure the new Adabas files to optimize the benefits outlined above.

The tables that identify files to Adabas are external to the applications and may be changed without relinking the application programs. This feature is especially useful when you want to change file or security information, or move applications from test to production status.

Adabas Bridge for DL/I (and IMS/DB)

Adabas Bridge for DL/I (ADL) is a tool for migrating DL/I or IMS/DB databases into Adabas. The term DL/I is used as a generic term for IMS/VS and DL/I DOS/VS. ADL operates either in batch mode or online (under CICS or IMS/DC) and is available for z/OS and z/VSE operating environments.

DL/I applications can continue to run without change: the migrated data can be accessed by Natural and by SQL applications if the Adabas SQL Gateway is available. ADL can also be used to run standard DL/I applications against Adabas databases.

Functional Units

ADL comprises six major functional units:

- A collection of conversion utilities automatically converts DL/I databases into Adabas files called ADL files to emphasize the special properties of these files as opposed to native Adabas files.

- As a result of the conversion process, the DL/I database definitions (DBDs) and the related Adabas file layouts are stored on an Adabas file called the ADL directory, which also contains the ADL error messages and other information.

- A menu-driven application written in Natural provides a number of online services, including reports on the contents of the ADL directory.

- A special set of integration programs uses data from the ADL directory to generate Predict definitions for ADL files, which can then be used to generate Natural views.

- A call interface allows DL/I applications to access ADL files in the same way as original DL/I databases (and both concurrently in mixed mode). It supports Assembler, COBOL, PL/I, RPG, FORTRAN, and Natural for DL/I. A special precompiler is provided for programs using the EXEC DL/1 interface.

- A consistency interface provides access to ADL files for Natural applications or programs using Adabas direct calls. This interface preserves the hierarchical structure of the data, which is important for ongoing DL/I applications.
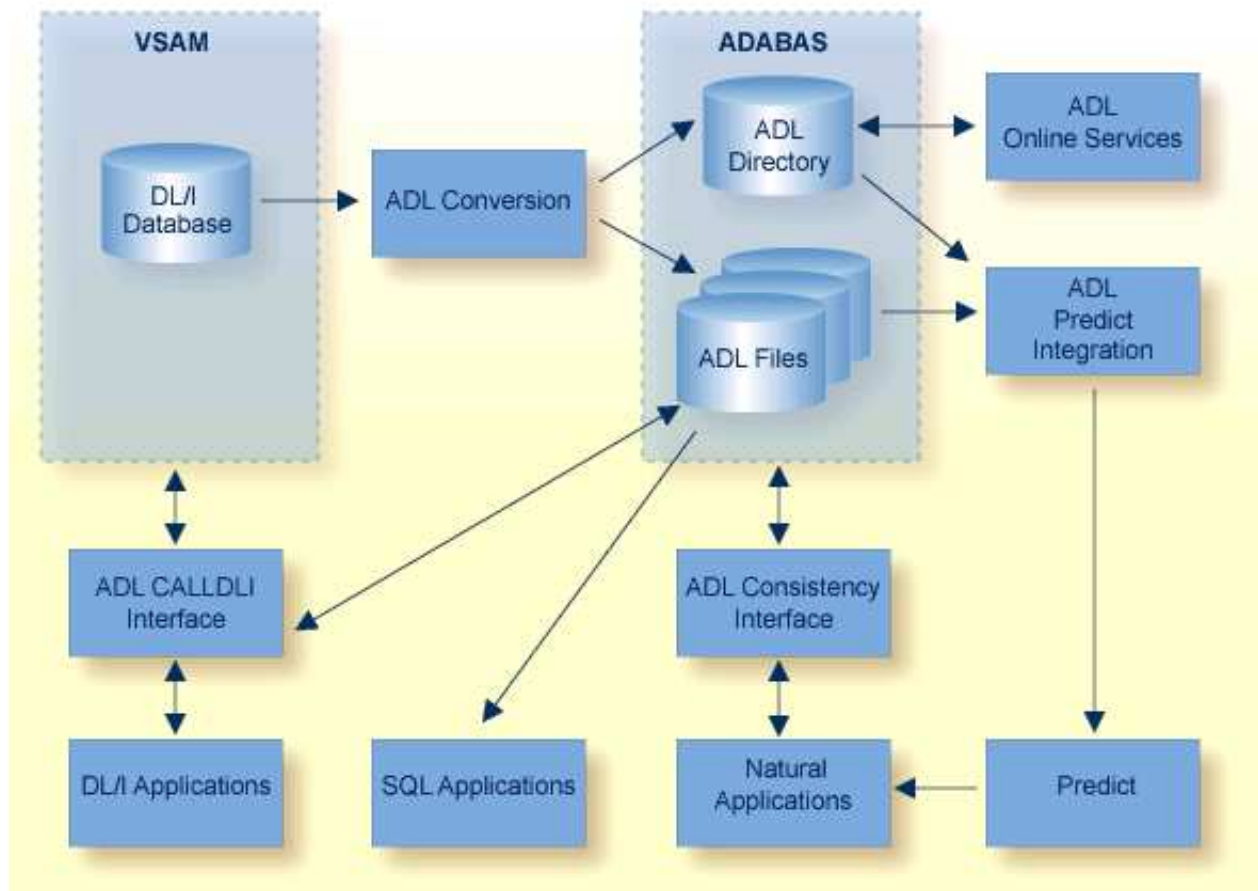
Advantages of Adabas over DL/I

The availability of Adabas in an environment in which only DL/I file structures were previously used results in the following benefits:

- Data can be manipulated by Natural, Software AG's fourth generation language;

- Data is compressed at field level automatically by Adabas;

- Deleted data records are released from storage immediately. This is in contrast to DL/I, which simply sets a flag in such records but does not release the storage used by them. With Adabas, the released space can be reused immediately for new records: there is no requirement to maintain records that are marked as deleted, and less reorganization of the database is required;

- All converted DBDs have full HIDAM functionality, regardless of the original access method;

- The length of a field can be increased without unloading and reloading the data;

- Segments can be added to the end of a DBD without unloading and reloading the data;

- Trace facilities are available for online and batch;

- The batch CALLDLI test program is available.

- Since Adabas (unlike DL/I) does not run in the CICS region/partition, online system resources are reduced.

- A symbolic checkpoint facility is available under z/VSE.

- HD databases are available under z/OS.

Adabas Caching Facility

Adabas Caching Facility helps improve system performance and make full use of ESA functions by augmenting the Adabas buffer pool in extended memory, data space, hiperspace and, in z/OS version 1.2 and above environments, 64-bit virtual storage.

Adabas Caching Facility augments the Adabas buffer manager by reducing the number of read Execute Channel Programs (EXCPs; UPAM SVCs for BS2000) to the database. This allows you to use the available operating system facilities without monopolizing valuable virtual memory resources.

Note:
Write EXCPs are always issued to maintain the integrity of the database.

Adabas Caching Facility is functionally similar to the Adabas buffer manager, but offers the following additional capabilities:

- User-specified RABNs (blocks) can be cached or fenced to make them readily accessible when demand arises, even though the activity against them is not sufficient to keep them in the active buffer pool. RABN-fencing reduces the required I/O response time if the Adabas nucleus needs to reread those RABNs.

- The Adabas Work data set parts 2 and 3 can be cached to improve performance in environments that service large numbers of complex queries. Adabas Work parts 2 and 3 serve as temporary work areas used to resolve and maintain the ISN lists of complex queries. Reducing the number of read and write EXCPs to Work parts 2 and 3 for these complex queries can decrease processing time dramatically and improve performance substantially.

- A file or range of files can be specified to cache all associated RABNs. It is also possible to cache only Associator or Data Storage blocks if required. Files can be prioritized by assigning a class of service which determines the percentage of the maximum available cache space that a given file can use and when the file's RABN blocks will be purged from the cache.

- Operator commands are available to dynamically respond to a changing database environment by modifying

    ○ the RABNs to be cached by RABN range, file, or file range;

    ○ the RABNs to be enabled or disabled by RABN range, file, or file range;

    ○ when to acquire and release the system resources used by the Adabas Caching Facility.

- When processing serial Adabas commands (e.g., read logical, read physical, histogram, and searches using non-descriptors) with the read-ahead caching option, a single EXCP is issued to read all the consecutive ASSO and/or DATA blocks that reside on a single track of the disk device. The blocks are kept in cache and are immediately available when the nucleus requests the next block in a sequence. This feature may enhance performance by reducing the number of physical read I/Os for a

3380 ASSO by as much as 18:1.

The integrity of the database is preserved because Adabas RABNs are not kept redundantly in both the Adabas buffer pool and the cache. An Adabas RABN may reside either in the Adabas buffer pool or in the cache area, but never in both. All database updates and consequently all buffer flushes occur only from the Adabas buffer pool. Unlike other caching systems, this mechanism of non-redundant caching conserves valuable system resources.

The demo or full version of Adabas Online System is required to use the online cache-maintenance application Cache Services. For more information, see the *Adabas Caching Facility* documentation.

Adabas Cluster Services

Adabas Cluster Services implements multinucleus, multithread parallel processing and optimizes Adabas in an IBM Parallel Sysplex (systems complex) environment. The Adabas nuclei in a sysplex cluster can be distributed to multiple z/OS images that are synchronized by a Sysplex Timer® (IBM). One or more Adabas nuclei may be run under a z/OS image.

Adabas Cluster Services comprises software components that ensure intercommunicability and data integrity among the z/OS images and the associated Adabas nuclei in each sysplex nucleus cluster. An unlimited number of sysplex clusters each comprising up to 32 clustered nuclei can reside on multiple z/OS images in the sysplex.

In addition to the increased throughput that results from parallel processing, Adabas Cluster Services increases database availability during planned or unplanned outages; the database can remain available when a particular operating system image or cluster nucleus requires maintenance or goes down unexpectedly.

To support a cluster environment that includes more than one operating system image, a limited Software AG Entire Net-Work library is included as part of Adabas Cluster Services (see also *Entire Net-Work Multisystem Processing Tool*). Entire Net-Work is used to send Adabas and Adabas Cluster Services commands back and forth between z/OS images. It provides the communication mechanism among the nuclei in the sysplex cluster. No changes have been made to Entire Net-Work to accommodate Adabas Cluster Services.

The ADACOM module is used to monitor and control the clustered nuclei. For each cluster, the ADACOM module must be executed in each z/OS image that either has a nucleus that participates in the cluster or has users who access the cluster database.

The Adabas Cluster Services SVC component SVCCLU is prelinked to the Adabas SVC and is used to route commands to local and remote nuclei. CSA space is used to maintain information about local and remote active nuclei, and currently active users.

The sysplex cache structure is used to hold ASSO/DATA blocks that have been updated during the session. It synchronizes the nuclei, users, and the z/OS images; ensures data integrity, and handles restart and recovery among the nuclei.

Cluster Services With Other Adabas Products

Adabas Online System communicates with all nuclei within the sysplex cluster.

Adabas Caching Facility supports clustered nuclei and can provide a performance boost to the cluster.

Advantages of Using Entire Net-Work

In an Adabas Cluster Services environment, Entire Net-Work allows users on various network nodes to query a logical database across multiple z/OS images. Users access a cluster database as they would a conventional, single-node database.

A request to Adabas can be made from within an existing application, without change. The request is processed automatically by the system; the logistics of the process are transparent to the application.

Entire Net-Work ensures compatibility by using Adabas-dependent service routines for the operating system interface as well as for interregion communication. Job control statements for running Entire Net-Work are much like those needed to run Adabas. For example, the EXEC statement invokes the ADARUN program for Entire Net-Work just as it does for Adabas, and the ADARUN parameters for Entire Net-Work are a subset of Adabas parameters.

Because status information is broadcast to all nodes whenever a target or service establishes or terminates communication with the network, there is no need to maintain or refer to database or target parameter files at a central location.

Allowing only one Entire Net-Work task on each node enforces control over the network topology by maintaining all required information in one place. This avoids confusion in network operation and maintenance. If, however, more than one Entire Net-Work task is required, this can be accomplished by installing additional routers.

Each Entire Net-Work node maintains only one request queue and one attached buffer pool for economical use of buffer storage. All buffers that are not required for a particular command are eliminated from transmission. In addition, only those portions of the record buffer and ISN Buffer that have actually been filled are returned to the user on a database reply.

Buffer size support in Entire Net-Work is comparable to that in Adabas, ensuring that all buffer sizes that are valid for Adabas can also be transmitted to remote nodes.

- Entire Net-Work XCF Option

Entire Net-Work XCF Option

Actual network data traffic is controlled by the Entire Net-Work XCF Option, an interface to IBM's cross-system coupling facility (XCF) which allows authorized applications on one system to communicate with applications on the same system or on other systems. XCF transfers data and status information between members of a group that reside on one or more z/OS images in a sysplex.

The Entire Net-Work XCF Option, which is installed on each Entire Net-Work node, provides high performance, transparent communications between z/OS images that reside on different central processors in the sysplex. Multiple connections to other nodes are supported, and the line driver's modular design permits easy addition of new access method support to the system.

A *member* is a specific function (one or more modules/routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one z/OS image in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group. Each Entire Net-Work node running the XCF line driver is identified as a different member in a group specifically set up for Entire Net-Work connectivity.
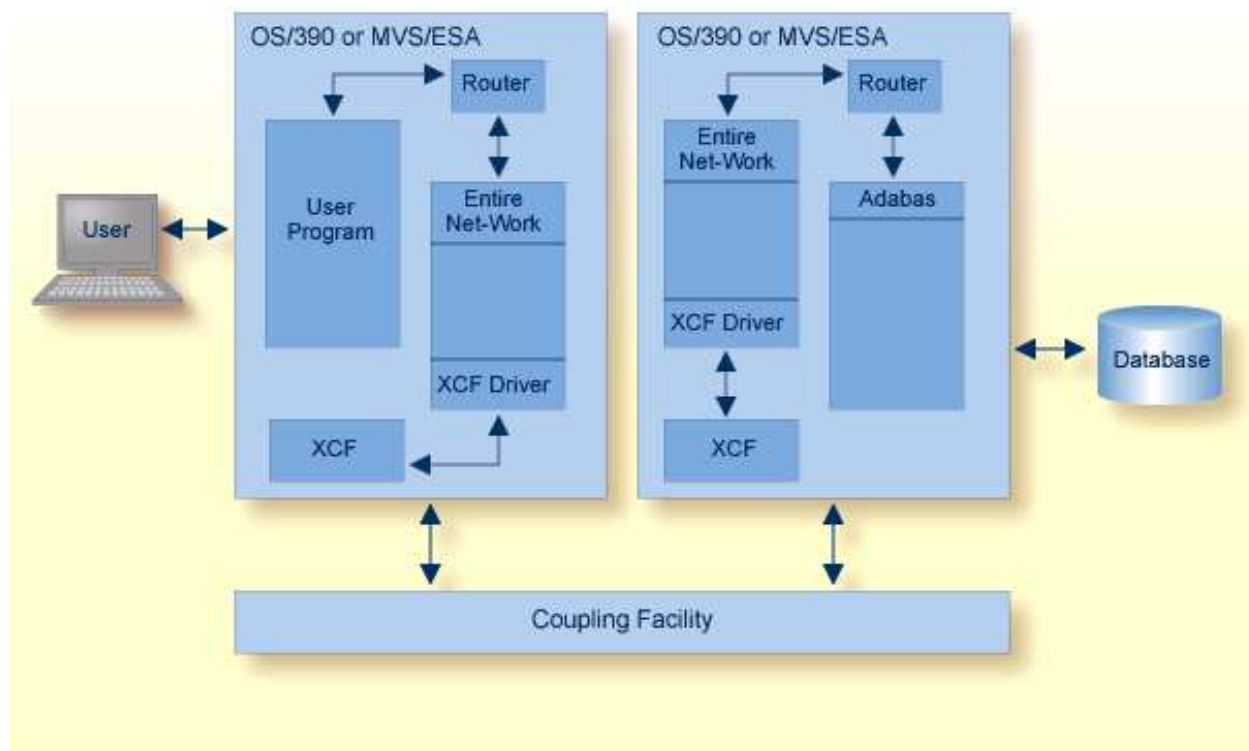
Adabas Delta Save Facility

Adabas Delta Save Facility (DSF) offers significant enhancements to ADASAV utility processing by backing up only the changed (delta) portions of Adabas databases. It reduces the volume of save output produced and shortens the duration of save operations; this increases database availability. By allowing more frequent save operations to be performed, it also reduces database recovery time.

Adabas Delta Save Facility provides for:

- more frequent saves without interrupting database availability;

- enhanced 24-by-7 operation;

- full offline saving parallel with the active database; and

- short REGENERATE duration during recovery.

Adabas Delta Save Facility achieves these objectives by saving only those Associator and Data Storage blocks that have changed (delta portion) since the last save operation. The result of this operation is called a *delta save tape*. Because a much smaller volume of output is written to delta save tapes, contention for secondary (tape, cassette etc.) storage is reduced.

Adabas Delta Save Facility can:

- maintain a log of changed database blocks (RABNs);

- create and merge interim delta save tapes while the database remains online, if required;

- consolidate delta save tapes with the most recent database save tape to create an up-to-date full save tape;

- restore the database from the most recent full save tape and all subsequent delta save tapes.

Adabas Delta Save Facility is intended for Adabas sites with one or more large, heavily updated databases that need to be available most of the time. It is particularly beneficial when the volume of data changed on a day-to-day basis is considerably smaller than the total database volume.

The demo or full version of Adabas Online System is required to use DSF. For more information, see the *Adabas Delta Save Facility Facility* documentation.

Adabas Fastpath

Adabas Fastpath optimizes response time and resources by bringing reference data closer to the user, reducing overhead, and reducing response times by servicing requests locally (that is, within the same region or partition).

Fastpath satisfies an Adabas query from within the application process, thus avoiding the operating system overheads needed to send a query to and from the database. Database activity such as command queue processing, format pool processing, buffer pool scanning, and decompression are also avoided.

Fastpath uses a query sampler to efficiently identify:

- the most commonly issued direct access queries; that is, queries where the client identifies the data being sought (e.g., ISN, search value).

- sequential access queries; that is, queries where the client identifies a series of data items related by sequence or search criteria.

The query sampler reports interactively and at shutdown the exact types of queries that can be optimized and their relative popularity.

For each type of query, Fastpath uses algorithms to recognize and retain the most popular data and discard or overwrite the least popular data. Given a particular amount of memory to use that is available to all clients within an operating system, Fastpath retains the results of popular data queries so that they can be resolved in the client process when repeated. The retained results comprise a common knowledge base that reflects the experience gained from past queries. The knowledge base is dynamic in that it is continually updated; the least popular data held there is discarded or overwritten.

A Fastpath component attached to the DBMS ensures that any changes to popular data are reflected in the results returned to the knowledge base. Fastpath data is always consistent with the DBMS data.

Before a query is passed to the DBMS, the Fastpath optimizer attempts to resolve the query from the knowledge base. If successful, the query is satisfied faster, without interprocess communication or DBMS activity. Fastpath optimizes sequences by dynamically applying Adabas prefetch read-ahead logic to reduce DBMS activity. As many as 256 data items can be retrieved in a single visit to the DBMS.

Fastpath optimization occurs in the client process, but requires no change to application systems. Different optimization profiles can be applied automatically at different times of the day. Once started, the Fastpath buffer can be left active without intervention; Fastpath reacts automatically to DBMS startup and shutdown.

For more information, refer to the *Adabas Fastpath* documentation.

Adabas Native SQL

Adabas Native SQL is Software AG's high-level, descriptive data manipulation language (DML) for accessing Adabas files from applications written in Ada, COBOL, FORTRAN, and PL/I.

Database access is specified in an SQL-like syntax embedded within the application program. The Adabas Native SQL precompiler then translates the SQL statement into a transparent Adabas native call.

Software AG's Adabas, Natural, Predict, and the Software AG Editor are prerequisites for Adabas Native SQL, which makes full use of the Natural user view concept and the Predict data dictionary system to access all the facilities of Adabas.

Using your Natural field specifications, Adabas Native SQL automatically creates Ada, COBOL, FORTRAN, or PL/I data declarations with the correct set of fields, field names, field sequence, record structure, field format and length.

Adabas Native SQL uses information about file and record layouts contained in Predict to generate the data structures that the generated Ada, COBOL, FORTRAN, or PL/I program needs to access the database. Then, as Adabas Native SQL processes the program, it records in Predict active cross-reference (Xref) information including the names of the files and fields that the program accesses.

These features help to eliminate the risk of writing incorrect data declarations in programs that access the database. In addition, they create comprehensive records in the data dictionary that show which programs read from the database and which programs update it, providing the DBA with an effective management tool.

For more information, refer to the *Adabas Native SQL* documentation.

Adabas Online System

Note:
Adabas includes a demo version of the Adabas Online System to illustrate its capabilities and to provide access to selected other services.

Adabas Online System (AOS) provides an online database administration tool for Adabas. The same functionality is available using a batch-style set of utilities.

AOS is an interactive menu-driven system providing a series of services used for online Adabas database analysis and control. These services allow a database administrator (DBA) to:

- display Adabas user statistics, monitor and control access and operation of one or all users;

- display and modify Adabas fields and files: add fields, allocate and remove file space, change file and database layout, view and remove field descriptors;

- restrict file use to utility users only, or lock/unlock file access completely.

Support is also provided for nucleus cluster environments as well as SAF security. In addition, AOS has the ability to dynamically modify ADARUN parameters.

AOS is written in Natural, Software AG's fourth generation application development facility. AOS security functions are available only if Software AG's Natural Security is installed and operating.

AOS includes functions that are comparable to the Adabas operator commands and utilities.
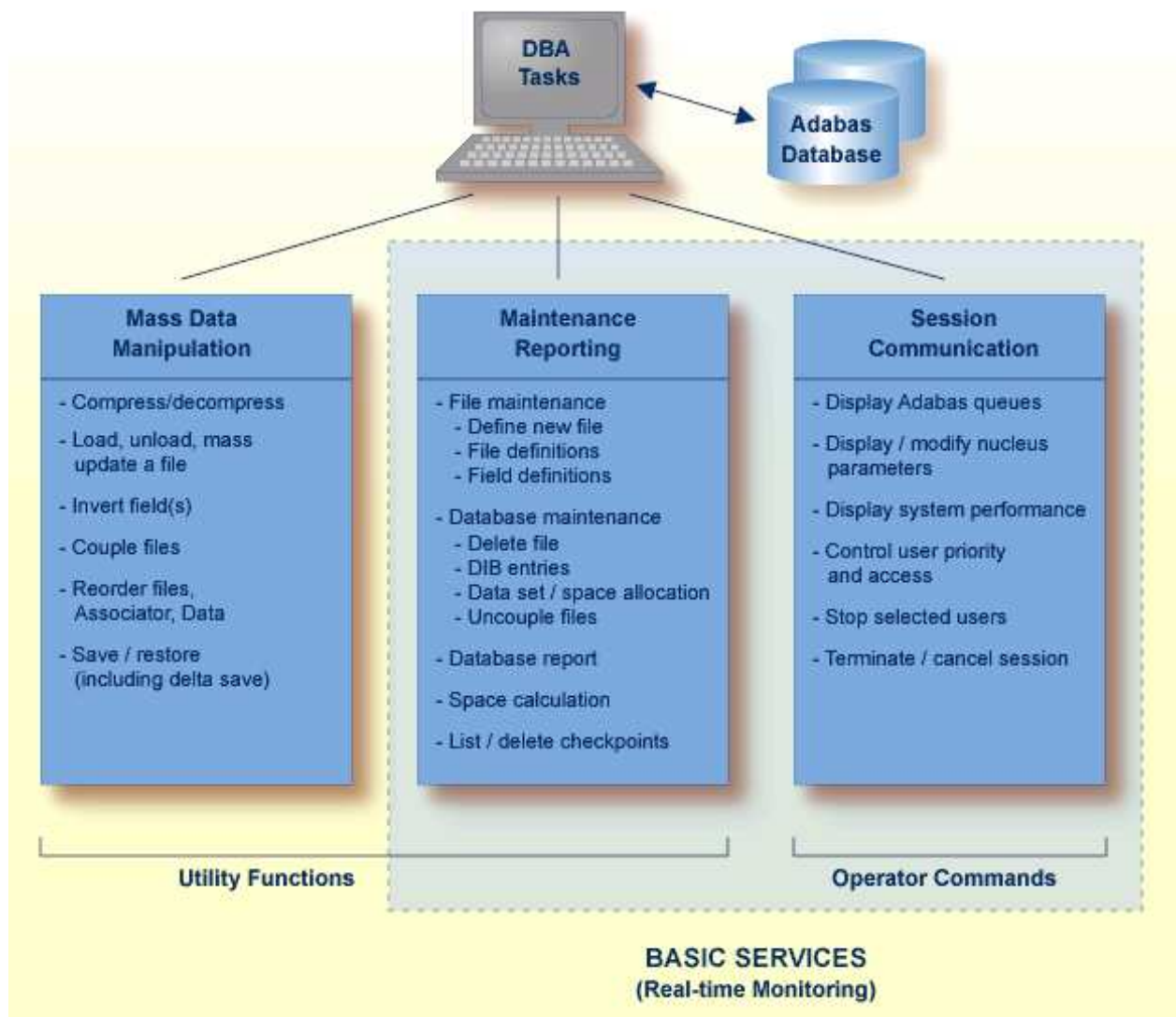
Basic Services makes it possible for the DBA to interactively monitor and change aspects of an Adabas database while an Adabas session is active. Using menu options or direct commands, the DBA can view resource status and user queues, display and revise space allocation, change file and database parameters, define a new file online, and stop a selected user or current Adabas session.

AOS is delivered in a separate data set/library from Adabas. In the initial Adabas delivery, AOS functions as a demonstration system. For full functionality, the contents of the AOS load library must be copied to the Adabas load library, or the AOS load library must be concatenated in the STEPLIB of the Adabas nucleus JCL. In addition, all AOS users (demo or full version) and Predict users must include the load module AOSASM from the Adabas load library in the link of the Natural nucleus.

For more information, refer to the *Adabas Online System* documentation.

Adabas Parallel Services

Adabas Parallel Services (formerly ADASMP) implements multinucleus, multithread parallel processing and optimizes Adabas in a multiple-engine processor environment on a single operating system image.

Up to 31 Adabas nuclei in an Adabas Parallel Services *cluster* are distributed over the multiple engines that are synchronized by the operating system.

Note:
The precursory product, Adabas support for Multiprocessing (ADASMP), provided single nucleus update and multinucleus read capability. Adabas Parallel Services provides multinucleus update capability as well.

All nuclei in the cluster access a single physical database simultaneously. A single physical database is one set of Associator and Data Storage data sets identified by a single database ID number (DBID).

The nuclei communicate and cooperate with each other to process a user's work. Compression, decompression, format buffer translation, sorting, retrieving, searching, and updating operations can all occur in parallel.

In addition to the increased throughput that results from parallel processing, Adabas Parallel Services increases database availability during planned or unplanned outages: the database can remain available when a particular cluster nucleus requires maintenance or goes down unexpectedly.

An unlimited number of Adabas Parallel Services clusters can operate in the same operating system image under the same or different Routers or SVCs; that is, an unlimited number of separate databases can be processed, each with its own Adabas Parallel Services cluster of up to 31 nuclei.

Applications see only one database target; no interface changes are required. Applications still communicate with their intended databases and communicate with an Adabas Parallel Services cluster of nuclei without modification.

For more information, refer to the *Adabas Parallel Services* documentation.

Adabas Review

Adabas Review (formerly Review Database) provides a set of monitoring, accounting, and reporting tools that enable you to monitor the performance of the Adabas environment and the applications executing within them.

Information retrieved about Adabas usage helps you tune application programs to achieve maximum performance with minimal resources.

In addition to the local mode with Adabas Review running in the Adabas address space, Adabas Review offers the hub mode, a client/server approach to the collection of performance data for Adabas:

- the Adabas Review interface (the client) resides on each Adabas nucleus.

- the Adabas Review hub (the server) resides in its own address space, partition, or region.

Cluster Services statistics gathering is supported. Both file-level (CF) caching statistics and Cluster Services locks can be monitored over a period of time and at user-defined intervals. Statistics are written to the Review History file and can be retrieved or viewed using the power of Review reporting facilities.

For additional, information, refer to the *Adabas Review* documentation.

- The Hub Server

- The Interface Client

- Interface Calls

- Example Client/Server Environment

The Hub Server

The Adabas Review hub is the data collector and the reporting interface for the user. The hub 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.

An 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.

Proven Adabas and Review components are combined in the centralized collection server (hub) with the following advantages:

- A single hub can collect information from multiple Adabas nuclei and from Adabas nucleus clusters managed by either Adabas Parallel Services or Adabas Cluster Services. This means that the number of Adabas Review nuclei required to support an enterprise-wide distribution of Adabas nuclei is minimized, resource requirements are minimized, and performance increases.

- Removing the Review subtask from the address space, partition, or region of each Adabas nucleus improves the performance of the Adabas main task. At the same time, the isolation 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 environment for Adabas Review; and

- the Review nucleus and subsystems including RAOSAUTO, the autostarted report parameter generation routine, and RAOSHIST, the historical data population routine.

The Interface Client

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 utilizes the existing Adabas interregion communication process; that is, ADALNK, ADASVC, and ADAMPM. This communication process is consistent across supported platforms.

When all supported platforms and systems are networked correctly, Adabas Review supports a multiple platform, multiple operating system, Adabas database environment.

The interface comprises the following:

- ADALOG, the Adabas command logging module;

- RAOSDAEX, the Adabas Review 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 RAOSDAEX and the Adabas API requirements for transmitting the Review data to the Adabas Review hub.
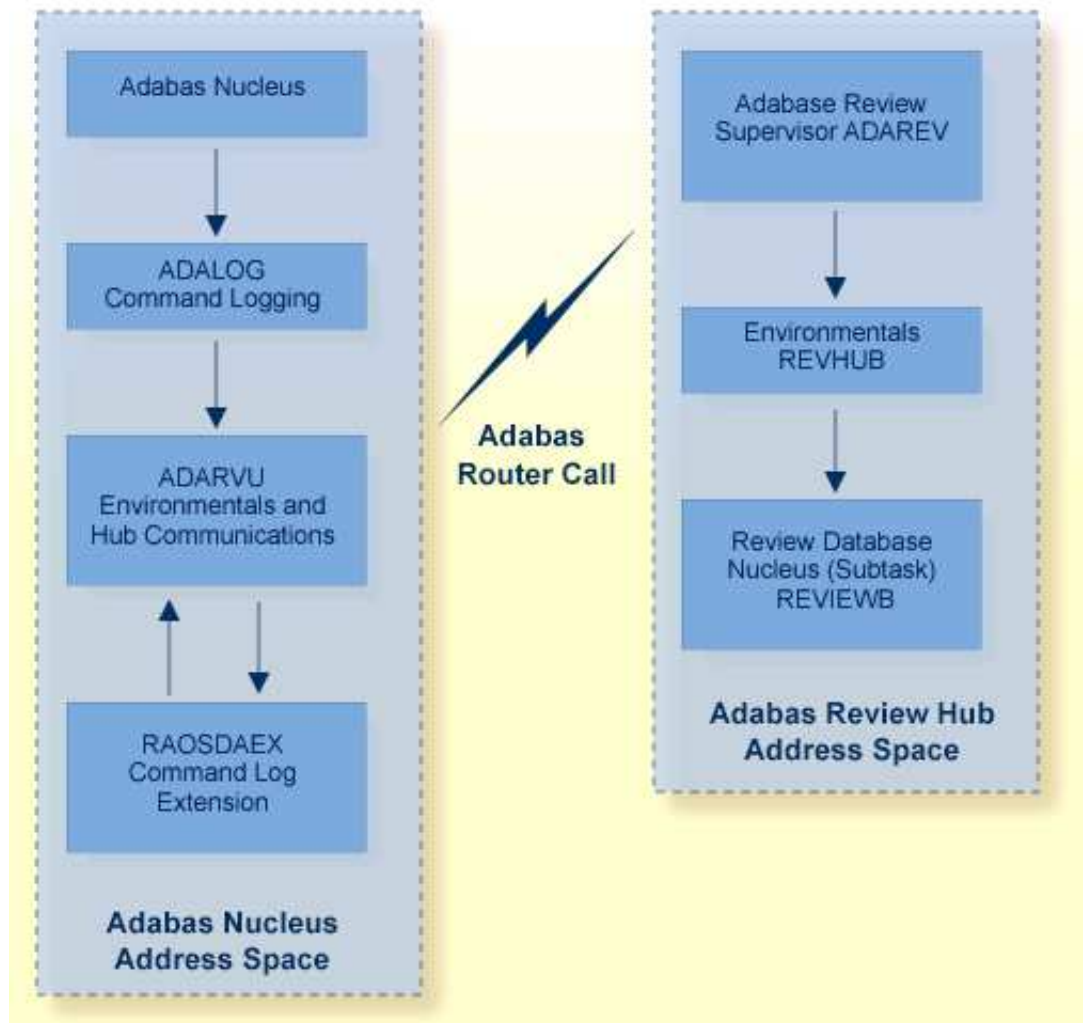
Interface Calls

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 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 figure shows the major components of the Adabas Review interface (Adabas nucleus address space) and the Adabas Review hub (Adabas Review hub address space) in a client/server architecture.

Adabas Nucleus

ADALOG
Command Logging

ADARVU
Environmentals and
Hub Communications

RAOSDAEX
Command Log
Extension

**Adabas Nucleus
Address Space**

**Adabas
Router Call**

Adabase Review
Supervisor ADAREV

Environmentals
REVHUB

Review Database
Nucleus (Subtask)
REVIEWB

**Adabas Review Hub
Address Space**

Adabas SQL Gateway

The Adabas SQL Gateway is a subset of CONNX, a unique client/server connectivity programming tool set that makes it possible to use computers in real-time interactive operation with many databases. The CONNX data access engine is unique in that it not only provides access to the databases, but it presents them as one enterprise-spanning relational data source. CONNX also offers ODBC SQL Level 2 compatibility, additional security, metadata management, enhanced SQL capability, views, heterogeneous joins, bidirectional data conversion, and enables read/write access to the data. Such technology can be used in data warehousing, data integration, application integration, e-commerce, data migration, and for reporting purposes. The technology also has a place within companies seeking to make use of disparate data sources, that need to Web-enable their data, or that have older applications storing mission-critical information.

CONNX includes the following components:

- CONNX Data Dictionary (CDD)

- CONNX ODBC Driver

CONNX OLE RPC Server (Not implemented for CONNX and VSAM)

- CONNX Host Data Server (RMS, VSAM [Implemented as CICS/C++ TCP/IP Listener/Server], C-ISAM, Rdb, and DBMS)

- CONNX JDBC Driver (Thin Client)

- CONNX JDBC Server

- CONNX JDBC Router

In addition to other databases, CONNX supports Adabas on IBM z/OS, Windows 2000/XP/2003, AIX, Solaris, Linux, HP-UX, and other platforms.

For complete information about the Adabas SQL Gateway, read the *Adabas SQL Gateway* documentation.

Adabas SQL Server

Adabas SQL Server is Software AG's implementation of the ANSI/ISO Standard for the standard database query language SQL. It provides an SQL interface to Adabas and an interactive facility to execute SQL statements dynamically and retrieve information from the catalog.

The server supports embedded static and dynamic SQL, as well as interactive SQL and SQL2 extensions. It automatically normalizes complex Adabas data structures into a series of two-dimensional data views that can then be processed with standard SQL.

Adabas SQL Server accesses and manipulates Adabas data by submitting statements

- embedded in a Natural application program.

- embedded in the third generation host languages C, COBOL, or PL/I.

- using a direct, interactive interface.

Currently, Adabas SQL Server provides precompilers for SQL statements embedded in C, COBOL, and PL/I. The precompiler scans the program source and replaces the SQL statements with host language statements. Due to the modular design of Adabas SQL Server, the functionality is identical regardless of the host language chosen.

Because certain extensions not provided for by the standard are available to take full advantage of Adabas functions, one of three SQL modes must be selected when compiling an application program: ANSI compatibility mode, DB2 compatibility mode, or Adabas SQL Server mode.

Both local and remote clients can communicate with the server using Entire Net-Work as the protocol for transporting the client/server requests. With the Adabas ODBC Client, an ODBC driver allows access to Adabas SQL Server using ODBC-compliant desktop tools. Entire Access, also ODBC-compliant, provides a common SQL-eligible application programming interface (API) for both local and remote database access representing a client-server solution for Adabas SQL Server.

Software AG is committed to making Adabas SQL Server available in most hardware and operating system environments where Adabas itself is available. The core functionality of the Adabas SQL Server will be identical across platforms.
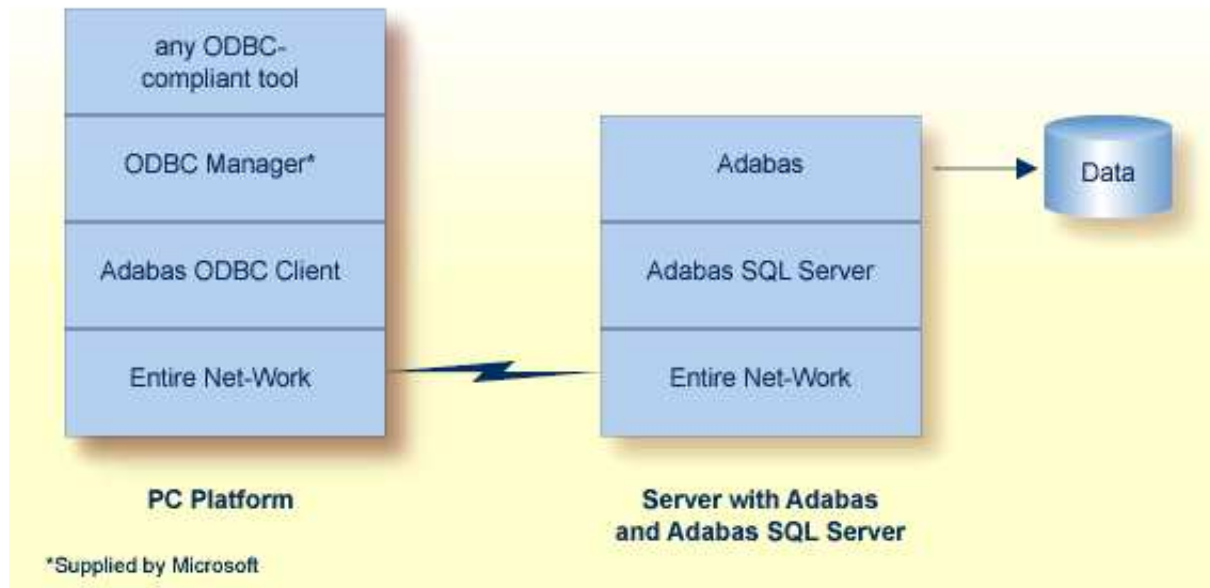
For more information, refer to the *Adabas SQL Server* documentation.

Adabas Statistics Facility

A database administrator (DBA) regularly checks the status of the database (such as disk and memory utilization) and plans for the long term, such as ensuring that the future disk space requirements can be met, based on current trends.

For Adabas, the DBA can check the status of individual databases and files using the ADAREP (Database Report) utility, the nucleus end session protocol, and ad hoc inquiries made using the Adabas Online System. This is often a time consuming process.

The Adabas Statistics Facility (ASF) provides an automated environment comprising:

- A *store program* that collects database status information during an active nucleus session. The store program is normally scheduled to run at regular intervals (for example once per day) over a period of many weeks or months to collect data that can be statistically evaluated. The store program can also be started by the DBA on an ad hoc basis, using commands in the ASF online menu system.

- A set of *evaluation programs* to interpret the statistics gathered by the store program and publish summary evaluation reports to either the screen or a hardcopy printer. Reports may also be downloaded to a PC using Entire Connection.

Database information can be collected at the start, at the end, and during a nucleus session. The start and end nucleus data, when accumulated over periods of weeks or months, gives an indication of long term database growth and permits projections of future database requirements. The nucleus performance data, such as main memory and pool usage, permits the DBA to analyze and tune the Adabas nucleus parameters.

For more information, refer to the *Adabas Statistics Facility* documentation.

Data Collection Program

ASF uses a data collection program called the *store program* to collect database status information at the start of, at the end of, or during an active nucleus session. This program is normally scheduled to run as a batch program at regular intervals (perhaps once per day) over a period of weeks or months to collect data that can be statistically evaluated. The store program can also be started by the DBA on an ad hoc basis, using commands in the ASF online menu system.

The DBA defines store profiles, each specifying a different set of databases and files to be monitored, that are specified as input to the store program when it runs. Several store programs, each with a different store profile, can run concurrently.

Approximately 170 criteria called data fields are used for monitoring Adabas databases and files. The data fields represent aspects of an Adabas database such as disk and buffer usage, thread usage, database load, ADARUN parameters, pool usage, and frequency of use of particular Adabas commands. All data fields are stored for each database and file specified in the store profile.

Start and end nucleus data, when accumulated over periods of weeks or months, give an indication of the long term database growth, and permit projections of future database requirements. Nucleus performance data, such as main memory usage and pool usage, provide information for tuning Adabas nucleus parameters.

Data Evaluation Programs

ASF uses a set of programs called *evaluation programs* to evaluate the statistics gathered by the Store Program and produce summary reports called evaluation reports that can be viewed online, printed, or downloaded to a PC.

For each evaluation report, the DBA uses the online menu system to define an evaluation profile specifying

- the databases and files for which data is to be evaluated (data for these must have been collected by a store program);

- for each database and file specified, one or more data fields to be analyzed in the report;

- the units of measurement of the data fields specified;

- upper and lower values representing critical levels for the data fields specified; and

- one of ten report types (01-10) which determines the format of the report heading.

The main types of reports are

- General evaluation: an analysis of the past and present database status. The statistical tables generated provide an overview of the status of various databases and files. The maximum and minimum values in rows of the output tables can be displayed, as well as statistical quantities such as the sum and average of the values.

- Trend evaluation: tables of projected statistics, in time steps of days, weeks, or months, until a specified end date.

- Critical Report: a report of databases and files for which the specified data fields have reached or exceeded the specified critical limits.

- Critical Trend Report: a report of databases and files for which the specified data fields will reach or exceed their critical limits within a time frame, based on an extrapolation of current trends.
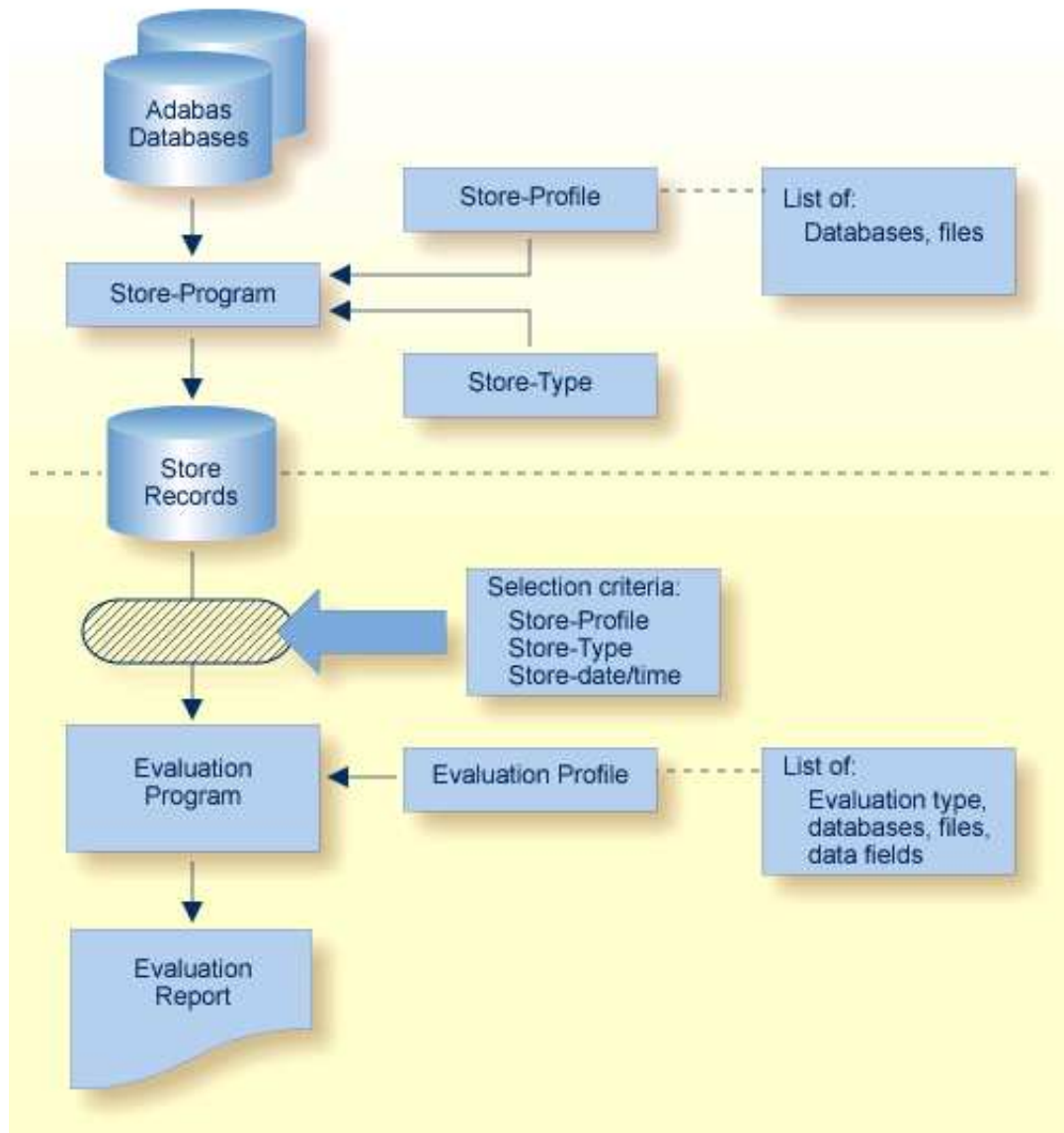
Adabas Text Retrieval

Adabas Text Retrieval is an extension of Adabas that allows you to develop applications that access both formatted and unformatted (that is, text) data simultaneously. Adabas Text Retrieval manages the index information and not the content of the data, making it possible to store the document contents at any location: Adabas, sequential files, CD-ROM, PC, etc. The call interface for Adabas Text Retrieval can be embedded in Natural, Software AG's fourth generation application development environment, or in any third generation language such as COBOL or PL/I.

Documents comprise chapters designated as either free text to be managed by Adabas Text Retrieval or formatted fields to be managed by Adabas itself. Free-text chapters comprise paragraphs and sentences, which can be individually searched.

Text substrings of an entered text are identified or tokenized by identifying a character defined in the Adabas Text Retrieval character table, by using algorithms to identify characters based on their contexts, or by using translation tables to sort or limit previously identified characters.

Document index entries are created when unformatted data is inverted. The full text can be inverted, or the inversion can be limited by a controlled thesaurus or by ignoring words in a stopword list.

Searches can be based on words, parts of words (right/left/middle truncations are possible), phonetics, synonyms, integrated thesaurus relations (broader/narrower terms), proximity operators (adjacent, near, in sentence, in paragraph), relational operators, Boolean operators, references to previous queries (refinement), or sorts (ascending or descending). Searches can be independent of structure, meaning that they may encompass any combination of free text and formatted fields. Search returns can be highlighted.

Natural users can expand the functions of Adabas Text Retrieval using Natural Document Management, which provides complete document management services.

Adabas Transaction Manager

Adabas Transaction Manager introduces distributed transaction support to Adabas. Transactions may be distributed over multiple Adabas databases in one or more operating systems (connected by Entire Net-Work). Transactions may also be distributed over non-Adabas DBMSs such as DB2, IMS, and so on. Where transactions are distributed over a non-Adabas DBMS, Adabas Transaction Manager must be configured to inter-operate with other transaction coordinators such as IBM's CICS Syncpoint Manager and/or IBM's Recoverable Resource Management Services. At any time, Adabas Transaction Manager can account for in-flight transactions, suspect transactions, participating databases, and more.

Adabas Transaction Manager (ATM) is an Adabas add-on product that:

- coordinates changes to Adabas databases participating in a global transaction;

- (function not available in first release) processes two-phase commit directions from transaction managers that take a higher-level, controlling role in coordinating global transactions such as IBM's RRMS or the CICS Syncpoint Manager allowing transactions to encompass both Adabas and non-Adabas databases operating within a single operating system image; and

- plays a key role in coordinating global transactions that change Adabas databases on more than one system image. In this case, the communication mechanism between the components across the system images in Entire Net-Work.

Each Adabas Transaction Manager instance (one per operating system image) executes in its own address space as a special kind of Adabas nucleus. Each Adabas Transaction Manager is aware of and partners with the other Adabas Transaction Managers in the distributed system and the databases they coordinate. At any time, each Adabas Transaction Manager can account for the status of the global transactions it is coordinating.

Adabas Transaction Manager addresses two basic needs of the enterprise object revolution:

- the need to deliver industrial strength enterprise objects for widespread commercial use in mainstream, critical business systems; and

- the need to spread the masses of data that Adabas customers manage more evenly across the computer(s) and organization.

Adabas Transaction Manager includes an online administration system based on Natural and available through Adabas Online System.

For more information, refer to the *Adabas Transaction Manager* documentation.

Adabas Vista

Adabas Vista allows you to partition data into separately managed files without reconstructing your business applications, which continue to refer to one (simple) Adabas file entity even though the physical data model is partitioned and possibly distributed across a wide-ranging computer complex.

Data can be partitioned across multiple Adabas database services. When a large file is partitioned across two or more databases, the processing load is actually being spread across the computer service. With more than one CPU engine in your computer, greater use is made of the parallel availability of the CPU engines.

Adabas Vista partitions are truly independent Adabas files:

- partitions need not be identical. Provided all the partitions support all of the views to be used by Adabas Vista, the files can operate with different physical layouts (FDTs). Of course, the Adabas source fields that are common to all partitions must be defined identically in each FDT.

- partitions can be maintained individually. You can size, order, and restore according to the needs of the individual partition. You do not have to make all partitions operate from the same physical constraints. You may choose some to be large, others medium, etc. You can also tune the ASSO space according to the partition.

You can select the applications for which Adabas Vista provides a single file image for all the partitions. You can also set an application for mixed access mode so that a program can access a partition directly by its real file number, even while using the single file image.

A file is usually partitioned based upon the overall dominance of a key field such as location or date: the partition criteria. However, it is possible to partition a file without a partition criteria.

Applications generally access the file with search data based to some extent on the key field. Adabas Vista minimizes processing overhead by detecting access explicitly or implicitly based upon the partition criteria, interrogating the search argument, and directing the access to the specific partition(s) needed. This is referred to as focused access.

The partition outage facility of Adabas Vista allows you to control what happens when a partition becomes unavailable. You can set sensitivity to partition outage unilaterally and allow business application to override it on a user basis. For example, if your partition criteria is location and only data in a particular location is critical to users in that location, you can set partition outage so that users are interrupted by outages in their own location but unaffected by outages in other locations. This can greatly increase the overall availability of your data, which can significantly enhance the effectiveness of your business.

The restricted partition facility of Adabas Vista allows you to hide partitions even though the data is available. You can use this facility to limit data to particular users based on role, location, or other business definition for security or performance reasons.

The consolidation facility of Adabas Vista allows you to impose a single file image upon multiple, previously unrelated files. The files may well be different but they support the same consolidated view.

Adabas Vista can be used in IBM mainframe environments (z/OS or z/VSE) with all supported versions of Adabas.

Adabas Vista supports Adabas calls from 3GL programs as well as from Natural. An online services option is available in a Natural environment.

Adabas Vista comprises a stub (client) part and a server part. By design, most processing occurs in the client process rather than the server to

- minimize CPU usage;

- minimize the impact of overhead associated with partitioning on the database service; and

- spread the load among as many CPU engines in parallel or even computers as possible.

For more information, refer to the *Adabas Vista* documentation.

Data Archiving for Adabas

Data Archiving for Adabas provides the tooling to implement a well-managed, automated, accountable, secure place to store, search and recall data archived from Adabas.

Most, if not all enterprises experience database content growth at an increasing rate. Studies show as much as 85% of database content is inactive. This pattern emerges across all types of industry. There are many reasons for the growth in data content, too many to cover here. The fact is that growth is a continuing phenomenon which pressures primary production services. At the same time, there are increasing legislative reasons why information must be kept for longer, creating even more pressure. Data Archiving for Adabas relieves the pressure by making it easy to archive and recall data on a large scale on a long-term basis.

Using Data Archiving for Adabas you can archive data from your Adabas databases (both mainframe and open system) to an Data Archiving for Adabas vault.. A vault is a flat-file store which contains all the accumulated archived data taken since the inception of the archive. Any number of vaults may be defined. For complete information, read your Data Archiving for Adabas documentation.

Event Replicator for Adabas

The Event Replicator for Adabas is composed of a family of Software AG products. The basic product can be used to monitor data modifications in an Adabas database and replicate the modified data to another application. For more information, read the following topics:

- Event Replication Overview

- Event Replicator Target Adapter

- Event Replicator Administration

- Entire Net-Work Administration

For complete information about the Event Replicator for Adabas, read the *Event Replicator for Adabas* documentation.

Event Replication Overview

Software AG's Event Replicator for Adabas allows specific Adabas files to be monitored for data modifications. Whenever any record modification (delete, store, or update) occurs in one of the monitored files, the Event Replicator extracts each modified record and delivers it to one or more target applications through a messaging system (such as webMethods EntireX or IBM MQSeries). The set of replicated files are defined in one or more subscriptions.

Note:
The term *MQSeries* is used in this documentation when referring to the product now known as *WebSphere MQ*.

The Event Replicator is an essential tool for organizations that need Adabas data modifications delivered to a target application while minimally impacting the normal processing of Adabas. The principle features of the Event Replicator include:

- Near real-time replication

- Asynchronous replication

- Guaranteed consistency and sequence of the delivered replicated data

- Replication of committed updates only

With the Event Replicator, whole Adabas files or a specific set of records can be replicated to the target location, as defined in one or more subscriptions. Data replication is asynchronous, which allows the Adabas database to operate normally while replication takes place. Only committed Adabas modifications are replicated for the predefined set of replicated files, at the transaction level.

For complete information about the Event Replicator for Adabas, read the *Event Replicator for Adabas* documentation.

Event Replicator Target Adapter

The Event Replicator Target Adapter can be used to transform and apply replicated data to a relational database, such as Oracle, DB2, Microsoft SQL Server, MySQL, or Sybase.

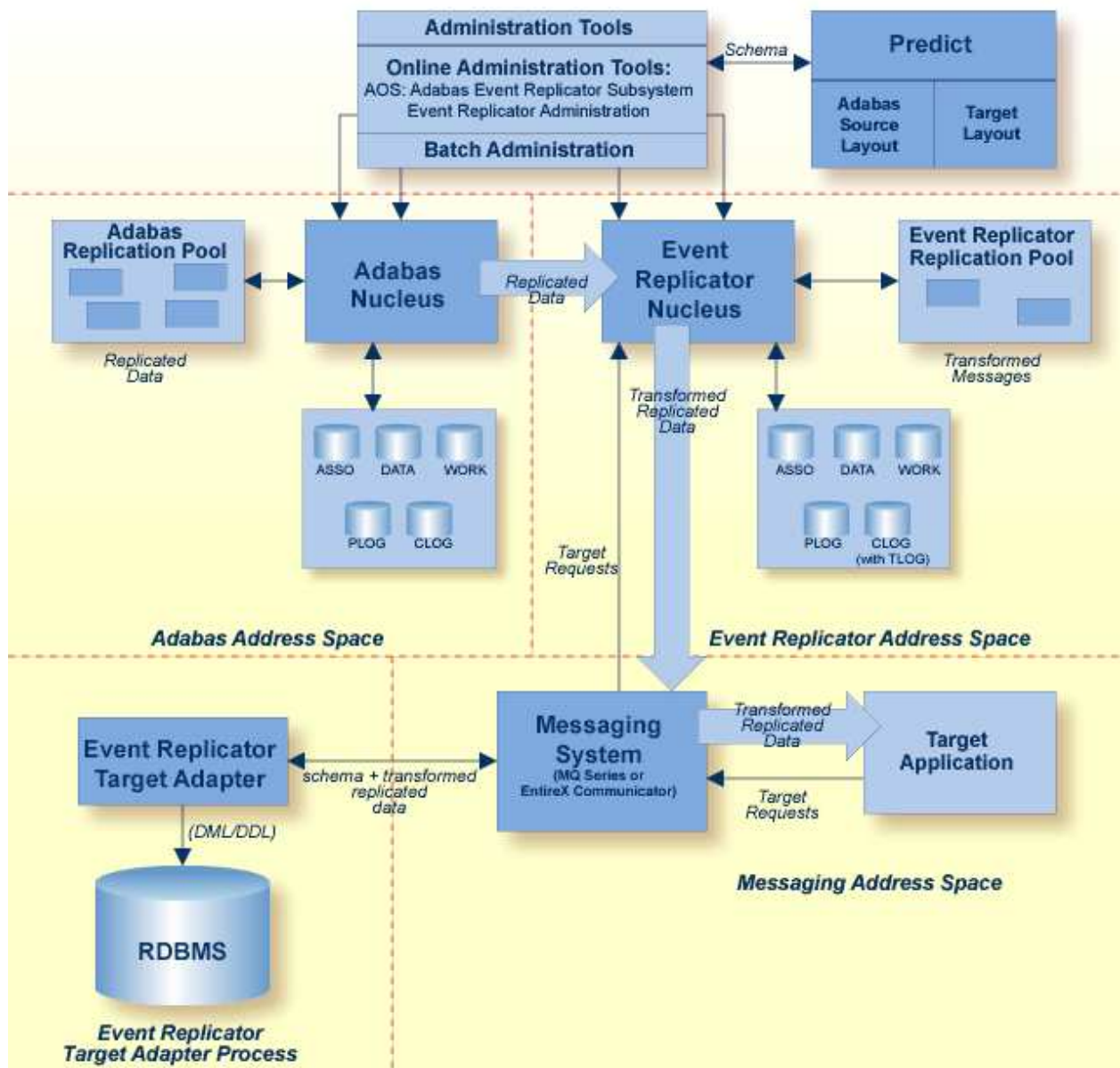The Event Replicator Target Adapter requires the use of:

- An Event Replicator for Adabas subscription for which a global format buffer field table has been generated. If no field table has been generated, the Event Replicator Target Adapter will not work.

● At least one Event Replicator for Adabas destination definition with a class type of "SAGTARG". This destination must be used by the subscription.

When a subscription definition and one or more of its associated destination definitions have been defined in this manner and if they are activated, the Event Replicator for Adabas automatically creates a schema that maps the replicated data. The Event Replicator Target Adapter uses the schema to transform and apply the replicated data to your relational database. The Event Replicator Target Adapter will dynamically create tables if they don't exist and populate the tables with Adabas data using insert, update, and delete processing as these processes occur in near real-time in the replicated Adabas file.

Event Replicator for Adabas and Event Replicator Target Adapter high-level processing are depicted in the following diagram.

Once appropriate Event Replicator subscription and destination definitions are activated and the Event Replicator Target Adapter is started, normal processing for the Event Replicator Target Adapter will transform and apply replicated data to a relational database as the data is processed by the subscription. In

addition, you can manually submit requests to the Event Replicator Target Adapter that:

- initiate an initial-state request to populate the relational database tables

- clear data in the relational database tables

- delete relational database tables and their data.

For complete information about the Event Replicator Target Adapter, read the *Event Replicator for Adabas* documentation.

Event Replicator Administration

Event Replicator Administration is a web-based graphical user interface (GUI) you can use to perform administrative tasks for the Event Replicator. The GUI uses a standard web browser that communicates with the System Management Hub (SMH), the central point of administration for Software AG's products. SMH handles user sessions and transforms user/browser interaction into requests to specific agents created for the Event Replicator that implement Event Replicator administrative tasks. SMH then forwards agent replies as HTML pages to the browser.

Note:
If you close the browser window in which the System Management Hub is running or switch to another URL, you will also terminate the session with Event Replicator Administration.

The Event Replicator Administration area in the System Management Hub includes two types of management:

- Adabas database management

- Event Replicator Server management

Using these management areas, you can manage the Adabas databases and Event Replicator Servers used by the Event Replicator for Adabas.

Area	Description
Adabas Databases	Use this area to register and unregister Adabas databases for use by the Event Replicator for Adabas.
Replicators	Use this area to register and unregister Event Replicator Servers for use by the Event Replicator for Adabas as well as to maintain the replication definitions in the Replicator system file associated with each Event Replicator Server.

For complete information about Event Replicator Administration, read the *Event Replicator for Adabas* documentation.

Entire Net-Work Administration

If you want to use Event Replicator Administration to maintain replication definitions, you must have some Software AG middleware components installed. The recommended way to do this is to install either Entire Net-Work Client or Entire Net-Work 7.2 or higher on the client side. Event Replicator Administration is shipped with Entire Net-Work Administration, which is a limited version of Entire

Net-Work 5.9 (including the Simple Connection Line Driver from its Entire Net-Work TCP/IP Option), and Entire Net-Work Client.

If you use the Entire Net-Work products listed above, the Event Replicator Servers and Adabas databases you maintain using Event Replicator Administration must be UES-enabled. Another way to establish communication between Event Replicator Administration and the Event Replicator Servers and Adabas databases is to use Entire Net-Work 2.6 on Windows and Entire Net-Work 5.8.1.

For complete information about Event Replicator Administration and Event Replicator requirements, read the *Event Replicator for Adabas* documentation.

Entire Net-Work Multisystem Processing Tool

Entire Net-Work, Software AG's multisystem processing tool, provides the benefits of distributed processing by allowing you to communicate with Adabas and other service tasks on a network-wide scope. This flexibility allows you to

- run Adabas database applications on networked systems without regard to the database location;

- operate a distributed Adabas database with components located on various network nodes;

- perform specific types of tasks on the network nodes most suitable for performing those services without limiting access to those services from other network systems;

- access Entire System Server (formerly Natural Process) to perform operating system-oriented functions on remote systems;

- access Entire Broker in order to implement your client/server applications.

Mainframe Entire Net-Work supports BS2000, z/OS, z/VSE, and Fujitsu Technology Solutions' MSP. It provides transparent connectivity between client and server programs running on different physical or virtual machines, with potentially different operating systems and hardware architectures.

Entire Net-Work is additionally available on the midrange platforms OpenVMS, UNIX, and AS/400 and on the workstation platforms OS/2, Windows, and Windows NT.

At its lowest level, Entire Net-Work accepts messages destined for targets or servers on remote systems, and delivers them to the appropriate destination. Replies to these requests are then returned to the originating client application, without any change to the application.

The method of operation and the location and operating characteristics of the servers are fully transparent to the user and the client applications. The servers and applications can be located on any node within the system where Entire Net-Work is installed and communicating. The user's view of the network targets and servers is the same as if they were located on the user's local node. Note that due to possible teleprocessing delays, timing of some transactions may vary.

Entire Net-Work insulates applications from platform-specific syntax requirements and shields the user from underlying network properties. It also provides dynamic reconfiguration and rerouting (in the event of a down line) to effect network path optimization and generate network-level statistics.

Entire Net-Work is installed on each participating host or workstation system requiring client/server capability. The configuration for a given system comprises an Entire Net-Work control module, control module service routines, and any required line driver. Each system with Entire Net-Work installed becomes a node in the network. Each node's adjacent links to other nodes are defined by name and driver type.

Each Entire Net-Work node maintains a request queue for incoming requests. This queue is similar to the Command Queue used by Adabas; it allows the node to receive Adabas calls from locally executing user/client programs, which Entire Net-Work then dequeues and transports to the nodes where the requested services reside.

Each local Entire Net-Work node also keeps track of all active network services, and therefore can determine whether the user's request can be satisfied or must be rejected. If the request can be serviced, the message is transmitted; otherwise, Entire Net-Work advises the calling user immediately, just as the Adabas router would do for a local database request.

Actual network data traffic is controlled by Entire Net-Work line drivers, which are interfaces to the supported communications access methods, such as VTAM, IUCV, DCAM, XCF, and TCP/IP, or directly to hardware devices, such as channel-to-channel adapters (CTCAs). Each Entire Net-Work node contains only those line drivers required by the access methods active at that node. In addition, each line driver supports multiple connections to other nodes; this modular line driver design permits easy addition of new access method support to the system.

The Entire Net-Work XTI interface allows users to write their own client/server applications, typically in C, which are independent of the Adabas structures. XTI is an internationally accepted vehicle for creating truly portable applications. In theory, an application created according to XTI specifications can easily be ported to any other platform that supports the XTI implementation.

The Entire Net-Work XTI implementation supports communication between programs running on the same machine and programs running on different machines. Entire Net-Work is viewed as the transport provider from the application programmer's point of view.
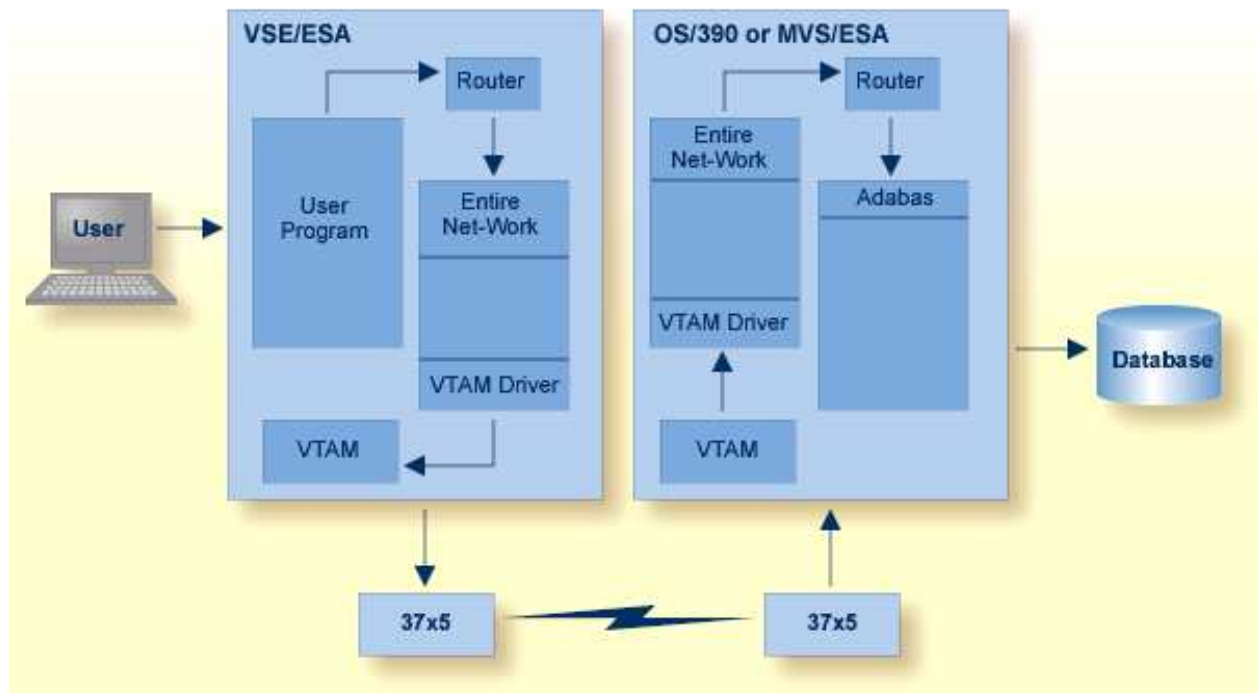
For more information, refer to the *Entire Net-Work* documentation.

Entire Transaction Propagator

Entire Transaction Propagator (ETP) allows Adabas users to have duplicate, or replicate, database files in a single database or distributed network. The copies can be distributed throughout a network to provide quick, economical access at user locations.

The concept of a distributed database provides operating efficiency and flexibility while at the same time offering almost unlimited data capacity. Such a networked database structure means that the portion of the database data needed by a particular department can be located on local systems and still be available corporate-wide as part of the common database resource.

One particularly appealing feature of distributed databases is the possibility of having duplicate copies of data at those locations where the data is needed most. This concept allows duplicate copies of a data file to be located throughout the database network, yet the copies are viewed logically by users as a single file.

Normally, a replicated file requires an intricate control process to ensure data integrity in all file copies after each change. For distributed systems with a high ratio of read transactions compared to write transactions, however, such critical control may be unnecessary. ETP provides an alternative replicated file concept using a less critical control process, but with virtually all the other advantages of replicated files. Using a master/replicate system of control, ETP resynchronizes all replicate copies with a master copy at user-specified intervals.

Natural Application Development Environment

High-level access to Adabas is provided by Natural, Software AG's advanced fourth generation application development environment and the cornerstone of Software AG's application engineering product family which includes analysis/design, code-generation, and repository facilities.

The access can either be directly from Natural to Adabas or using an Entire Access call via Adabas SQL Server.

Whereas the FDT defines the physical records in an Adabas file, Natural programs define and use logical views of the physical file to access the file. There can be two levels of views: data definition modules and user views.

Data definition modules (DDMs) are Natural modules that look much like the Adabas FDT. They consist of a set of fields and their attributes (type, format, length, etc.), and may contain additional specifications for reporting formats, edit masks, and so on.

A DDM can include all the fields defined in the FDT or a subset of them. There must be at least one DDM for each Adabas file. For example, the Adabas file Employees could have a DDM called Employees. The Natural statement READ EMPLOYEES BY NAME actually refers to the DDM rather than the physical file; the DDM links the Natural statement to the Adabas file.

You can define multiple DDMs for an Adabas file. Multiple DDMs are a way of restricting access to fields in a file. For example, a DDM for a program used by managers could include fields that contain restricted information; these fields would not be included in a DDM for a general-use program. In a workstation group, a database administrator may define a standard set of DDMs for the group.

A new Adabas FDT can be created from an existing Natural DDM. Conversely, Adabas can generate or overwrite a DDM automatically when an FDT is created or changed.

Note:
When you delete a field from an Adabas file, you must also eliminate it from Natural programs that reference it.

A Natural user view often contains a subset of the fields in a DDM. User views can be defined in the Data Area Editor or within a program or routine. When a user view references a DDM, the format and length do not need to be defined, since they are already defined in the DDM. Note that in a DDM or user view, you can define the sequence of fields differently from the FDT sequence.

Adabas access is field-oriented: Natural programs access and retrieve only the fields they need. Natural statements invoke Adabas search and retrieval operations automatically.

Adabas supports a variety of sequential and random access methods. Different Natural statements use different Adabas access paths and components; the most efficient method depends on the kind of information you want and the number of records you need to retrieve.

For more information, refer to the *Natural* documentation.

Predict Data Dictionary System

Predict, the Adabas data dictionary system, is used to establish and maintain an online data dictionary. Because it is stored in a standard Adabas file, it can be accessed directly from Natural.

A data dictionary contains information about the definition, structure, and use of data. It does not store the actual data itself, but rather data about data or metadata. Containing all of the definitions of the data, the dictionary becomes the information repository for the data's attributes, characteristics, sources, use, and interrelationships with other data. The dictionary collects the information needed to make the data more useful.

A data dictionary enables the DBA to better manage and control the organization's data resources. Advanced users of data dictionaries find them to be valuable tools in project management and systems design.

Database information may be entered into the dictionary in online or batch mode. The description of the data in the Adabas dictionary includes information about files, the fields defined for each file, and the relationship between files. The description of use includes information about the owners and users of the data in addition to the systems, programs, modules and reports that use the data. Dictionary entries are provided for information about

- network structures

- Adabas databases

- files, fields, and relationships

- owners and users

- systems, programs, modules and reports

- field verification (processing rules)

Standard data dictionary reports may be used to

- display the entire contents of the data dictionary

- print field, file, and relationship information

- print field information by file

For more information, refer to the *Predict* documentation.