

## **Adabas Fastpath**

### **Introduction to Adabas Fastpath**

Version 8.2.2

March 2013

This document applies to Adabas Fastpath Version 8.2.2.

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

Copyright © 2013 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, United States of America, and/or their licensors.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <http://documentation.softwareag.com/legal/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third-Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

**Document ID: AFP-INTRO-822-20130314**

## Table of Contents

1 Introduction to Adabas Fastpath .....	1
2 Adabas Fastpath Benefits and Features .....	3
Increased Throughput via Optimization .....	4
Application Transparency .....	4
Continuous Operation .....	8
Security .....	5
Support for Distributed Systems .....	6
3 Why is Optimization Important? .....	7
Increasing Database Management System (DBMS) Workloads .....	8
Increasing Application Processing Overhead .....	8
Conventional Optimization Solutions .....	9
4 The Adabas Fastpath Optimization Solution .....	11
Database Query Optimization .....	12
Database Read-Ahead Optimization .....	14
5 Adabas Fastpath Tools .....	17
Optimization Sampler (AFPLOOK) .....	18
Online Administration (SYSAFP) .....	20
Buffer Operation and Reporting (AFPCMD) .....	21



# 1 Introduction to Adabas Fastpath

---

This document provides an introduction to Adabas Fastpath.

The following topics are provided:

**Adabas Fastpath Benefits and Features**

**Why is Database Access Optimization Important?**

**The Adabas Fastpath Optimization Solution**

**Adabas Fastpath Tools**



# 2 Adabas Fastpath Benefits and Features

---

- Increased Throughput via Optimization ..... 4
- Application Transparency ..... 4
- Continuous Operation ..... 8
- Security ..... 5
- Support for Distributed Systems ..... 6

This section provides an overview of the benefits and features provided by Adabas Fastpath.

## Increased Throughput via Optimization

---

Adabas Fastpath provides optimization of repetitive direct access queries to data managed by Adabas, as well as optimization for sequential access to Adabas data using a read-ahead facility.

By increasing overall throughput, Adabas Fastpath can reduce

- Adabas CPU consumption
- application CPU consumption
- online response times
- batch elapsed times
- overall costs

Adabas Fastpath satisfies an Adabas query from within the application process to avoid the operating system overheads needed to send a query to and from the database. It also avoids database processing functions such as

- command queue processing
- format pool processing
- buffer pool scanning
- I/O (potentially)
- decompression

Because Adabas Fastpath optimizes a query in the client process, queries that previously required queuing (and other) overhead involved in database processing are now processed with no overhead.

In multiprocessing environments, this implies that the capacity for database query throughput is increased significantly, and to scale.

## Application Transparency

---

Adabas Fastpath provides optimization without the need to change the application systems. Because no reinvestment is needed, older systems that are difficult or impossible to maintain can also be optimized.

For example, if you are having performance problems with a system that has been running for some time and there is no budget (or relevant experience) to improve the system, you can use the Adabas Fastpath sampler to analyse which queries issued by the system can be optimized and



then instruct Adabas Fastpath to perform the optimization. Adabas Fastpath implementation is that simple.

## Continuous Operation

---

The Adabas Fastpath software is used by

- the database;
- each client job; and
- the Adabas Fastpath buffer manager.

During production operation, it may be difficult to find time for software maintenance. To minimize the time required, Adabas Fastpath provides an automatic upgrade facility. After applying maintenance, you can direct Adabas Fastpath to dynamically load a new copy of its own code. Adabas Fastpath coordinates the process across the whole operating system.

Once the Adabas Fastpath buffer is started, it can be left active without intervention. Adabas Fastpath reacts automatically to database startup and shutdown.

You can instruct Adabas Fastpath to apply different optimization profiles according to the time of day. For example, your systems may generate significantly different work during the night-time hours. You can instruct Adabas Fastpath to operate accordingly, automatically.

## Security

---

Adabas Fastpath read-ahead optimization makes a series of sequential accesses more efficient for a single client. Because no data is shared between clients, data security is not an issue.

Fastpath direct access optimization retains the results of previous queries (from any client) and supplies the results to any client. In this case, data security is indeed relevant:

- When Adabas Fastpath is instructed to optimize a direct access query that is secured under Adabas Security, it issues an Adabas response code.
- If the Adabas SAF Security Interface is installed for a database, Adabas Fastpath verifies security permission at the file/user level.

## Support for Distributed Systems

---

Adabas Fastpath can be used to optimize Adabas databases whether they are local or remote to the client process while automatically maintaining the integrity of the data in the Adabas Fastpath buffer.

- Using Entire Net-Work
- Using Clusters

### Using Entire Net-Work

When the database is accessed over the network, calls are optimized in the client process before they are transported, thus bypassing the Entire Net-Work communication path as well as Adabas command processing.

Entire Net-Work makes the location of the database transparent to the Adabas Fastpath optimization process. There are no special installation requirements.

### Using Clusters

Adabas Fastpath works in conjunction with the Adabas System Coordinator to provide support for clustered Adabas servers and for clustered applications.

The maximum benefits from Adabas Fastpath are provided to all images of the Adabas server running under Adabas Cluster Services in an IBM parallel sysplex environment. The integrity of all Adabas Fastpath buffer data is ensured no matter where the data is updated.

Adabas Fastpath supports clustered applications that allow client session movement across or within systems.

# 3

## Why is Optimization Important?

---

- Increasing Database Management System (DBMS) Workloads ..... 8
- Increasing Application Processing Overhead ..... 8
- Conventional Optimization Solutions ..... 9

This section provides information regarding the importance of using optimization facilities as provided by Adabas Fastpath in today's data processing environments:

## Increasing Database Management System (DBMS) Workloads

---

A DBMS adds significant value to a computer. Modern systems are constantly increasing the DBMS workload and the trend shows no sign of reversing itself.

The increased amount of data to be managed by the DBMS is accompanied by the need for

- more processing power; and
- improved system responsiveness.

A more efficient DBMS goes a long way toward controlling the costs associated with these increasing requirements.

## Increasing Application Processing Overhead

---

A DBMS makes it possible for systems designers to create highly complex computer systems more easily than in the past. Consequently, more and more data is stored in the DBMS.

But there is a cost.

The application system client and the DBMS server reside in separate processes within the operating system. To satisfy a DBMS query, the following must occur:

1. The client prepares the query
2. The client issues the query
3. The query is shipped to the server
4. The server processes the query and prepares the response
5. The response is shipped back to the client
6. The client receives the result

Shipping data from one process to another within an operating system is resource intensive. Experience shows that benefit is gained by minimizing this overhead.

## Conventional Optimization Solutions

---

As systems designers have recognized the rising costs associated with increased DBMS usage, some have chosen to make highly used (popular) data permanently available to the client process in the form of memory-based tables, thus avoiding some part of the DBMS activity.

However, this solution has some basic problems and associated costs:

- the application system must have special logic to use these tables; and
- the data is difficult to change (as it is no longer in the DBMS).

In an attempt to resolve the difficulties of tables in memory, other solutions preload data into memory from the DBMS. However, data copied into the client process must be synchronized with the DBMS copy to ensure that clients operate only from copies that are current. Otherwise, serious data integrity problems may be created.

To avoid the need for a synchronization mechanism, some systems preload static (unchanging) data only. Others simply ignore the possible consequences. These approaches provide limited benefit, at best.

Furthermore, the cost of adding complexity to application systems for such limited gain is often overlooked. It can cause a major degradation in the organization's responsiveness to commercial change that may only come to light at the most inopportune time.



# 4 The Adabas Fastpath Optimization Solution

---

- Database Query Optimization ..... 12
- Database Read-Ahead Optimization ..... 14

This section describes the solutions provided by Adabas Fastpath for optimizing database access:

## Database Query Optimization

---

This section describes the database query optimization provided by Adabas Fastpath:

- [The Concept of Data Popularity](#)
- [Identifying Popular Data](#)
- [Building a Query Knowledge Base](#)
- [Sharing a Query Knowledge Base](#)
- [Integrity of Query Results](#)

### The Concept of Data Popularity

Software AG's considerable experience in DBMS technology indicates that as much as 80% of DBMS queries are directed at only 20% of the data. This is commonly referred to as *data popularity*.

At any given time, some amount of data is popular. However, today's popular data may not be the same as yesterday's or tomorrow's. For computer systems, the time span of popularity is probably not days, but rather seconds or less.

The popularity of any particular data in the DBMS tends to vary according to both commercial and system activity. In this sense, the popular data in the DBMS is a constantly moving target.

To effectively optimize queries to this data, it is necessary to identify and constantly re-evaluate the data that is popular.

How Adabas Fastpath has resolved this complex issue is described in the next section.

### Identifying Popular Data

Adabas Fastpath provides the sampling tool **AFPLOOK** to identify the most commonly used data.

AFPLOOK can be used to quickly identify exactly which database queries can be optimized and their relative popularity.



## Building a Query Knowledge Base

Adabas Fastpath accumulates a knowledge base by building a number of indexed query models. Each query model represents a set of query results that are available for optimization. The results within a set are indexed by search argument, ISN, etc. to provide an efficient means for locating results.

Once started, Adabas Fastpath analyses the queries being made and the data that is currently popular. For each type of query, various algorithms are used to

- recognize and retain the most popular data; and
- discard or overwrite the least popular data.

Adabas Fastpath does not attempt to load all data because not all data is used at the same time.

Rather, you instruct Adabas Fastpath to use a particular amount of memory. Adabas Fastpath will then make the best use of that memory by retaining the results of popular data queries so that they can be resolved in the client process when repeated. Large Objects (LOBs) can be retained by Adabas Fastpath but due to their size they are unlikely to offer the best use of the memory available to Fastpath.

## Sharing a Query Knowledge Base

Adabas Fastpath's knowledge base is maintained in memory that is available to all clients within an operating system. One client may therefore benefit from the results collected from another client query.

Before a query is passed to the DBMS, the Adabas Fastpath optimizer in the client process attempts to resolve the query from the knowledge base. If the query is satisfied, response times are shortened and processing overhead is reduced significantly because:

- less time is needed for the query;
- interprocess communication is avoided; and
- DBMS access is avoided.

The least popular data held by Adabas Fastpath in the dynamic knowledge base is overwritten. This means that the longer Adabas Fastpath is active, the more certain you can be that the most popular data currently in use is retained.

## Integrity of Query Results

Using the Adabas Fastpath solution, all clients receive data that is absolutely consistent with the DBMS at all times. An Adabas Fastpath component attached to the DBMS ensures that any changes to popular data are reflected in the results returned.

## Database Read-Ahead Optimization

---

This section describes the database read-ahead optimization provided by Adabas Fastpath:

- [The Concept of Read-Ahead Sequences](#)
- [Identifying and Processing Read-Ahead Sequences](#)

### The Concept of Read-Ahead Sequences

In a direct access query, the client identifies the data being sought using search data. There is another type of query where the client wishes to access a series of data items that are related by perhaps sequence or search criteria.

This series of related access is viewed as a *sequence* by Adabas Fastpath. A sequence may be initiated by a direct access query to position to the data required but once that is done, the client enters into an iteration where the next data in the sequence is requested.

Adabas Fastpath optimizes sequences by dynamically applying *read-ahead logic* to the query so that the DBMS workload is again reduced. Adabas Fastpath uses the Adabas multifetch option to perform the read-ahead operation.

Using read-ahead, many data items can be retrieved in one access to the DBMS. Adabas Fastpath then satisfies subsequent client queries from within the client process.

### Identifying and Processing Read-Ahead Sequences

The Adabas Fastpath query sampler identifies those queries that are used for read-ahead access.

When you direct Adabas Fastpath to apply read-ahead logic to a file, an algorithm is used to decide the read-ahead rate so that the maximum possible benefit is achieved. If Adabas Fastpath simply applies a read-ahead rate of 1, 50% of the interprocess communications to the DBMS for a sequence are saved. This is because for every data item requested by the client, Adabas Fastpath causes an additional data item to be returned with it.

Read-ahead processing handles both online and batch sequences, although online sequences tend to be much shorter than batch ones.

For example, an online transaction may allow entry of a surname such as SMITH. The DBMS may contain many data items for SMITH, so a sequence is used to show the first page of SMITH data

items on a selection screen. The user may select a particular data item for more detailed processing or signal for the next page, thereby continuing the sequence.



# 5 Adabas Fastpath Tools

---

- Optimization Sampler (AFPLOOK) ..... 18
- Online Administration (SYSAFP) ..... 20
- Buffer Operation and Reporting (AFPCMD) ..... 21

This section describes the tools provided with Adabas Fastpath.

## Optimization Sampler (AFPLOOK)

---

The Adabas Fastpath command analysis sampler (AFPLOOK) can be used to determine which database commands have the most potential for Adabas Fastpath optimization.

The following are examples of AFPLOOK output.

The first screen shows the overall potential optimization for database 50:

```

12:33:10      ***** A D A B A S   FASTPATH SERVICES 8.2.1 *****      2006-05-19
              -  Afplook - Summary for DB 50      -                      F16300MA

              Afplook Started May  19, 2006 at 12:30 by   UKDEV1

----- Potential Optimization Summary -----

              Sampled Commands                Potential Optimization

Sampled File Commands      57960 (75%) <=== Sequential:           2797 ( 5%)
                               Direct Access:           30597 (53%)
                               RCs:                22194 (38%)

Excluded Commands         19115 (25%)

Totals                     77075 (100%)                55588 (72%)
-----

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit  Refr                                File  More  Menu
    
```

### AFPLOOK Potential Optimization Summary

The second screen shows the potential optimization for all files in database 50:

```

12:31:26          ***** A D A B A S   FASTPATH SERVICES 8.2.1 *****          2006-05-19
                    Afplook - Files Summary for DB 50                                F16200MA

                    Started: May 19,2006 at: 12:30 by: UKDEV1

C   Fnr      Total          Direct Access          RCs          Sequential
-   50      3282 ( 3%)          1641 (50%)          1641 (50%)
-   52      2428 ( 2%)          1214 (50%)          1214 (50%)
-   53      3144 ( 3%)          1564 (50%)          1564 (50%)
-   55      1644 ( 1%)           822 (50%)           822 (50%)
-   64       821 ( 1%)           821
-   87      1640 ( 1%)           820 (50%)           820 (50%)
-  135     12432 (10%)         7102 (57%)          5330 (43%)
X  137     51540 (43%)        28455 (55%)         16876 (33%)         3547 ( 7%)
-  139      5324 ( 4%)          2662 (50%)          2662 (50%)
-  143     19514 (16%)         8870 (45%)          7983 (41%)          1774 ( 9%)
-  144      3551 ( 3%)          1776 (50%)          1775 (50%)
-  191      3226 ( 3%)          1378 (43%)           908 (28%)           470 (15%)
-  250      9432 ( 8%)          5623 (60%)          3809 (40%)
Total     122202 (100%)

Mark to Display File Details

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help          Exit  Refr                                Sumry

```

### AFPLOOK Files Summary

The third screen shows the optimization potential for file 137 in database 50.

```

12:32:21      ***** A D A B A S   FASTPATH SERVICES 8.2.1 *****      2006-05-19
              - Afplook - Details for DB 50 File 137                -      F16210MA

              Started: May 19,2006 at: 12:30 by: UKDEV1

CC DE Direct Acc      RCs Sequential Sequences      Non-Qualifying
L3 S5      23902      20508      6752      394      Updates 32
S1 S5      54584      27093
L3 S1      23670      16889      6781      6781      Deletes 0
L1 --      6748
L9 TA      13032      13032
L9 S5      9326
              16478      7896
              Prefetch 0
              Excluded 32
              Spare 0
              No CID 0
              Max.CIDs 0
              Bad Rsp. 0
              Max.Rec.Buff.Len
Total      131262      77522      30011
Total Commands for File      253970      Direct Acc 69
              Sequential 63

Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
      Help      Exit Refr Parm      File      Menu
    
```

**AFPLOOK File Details Display**

For more information, see the section AFPLOOK.

**Online Administration (SYSAFP)**

Adabas Fastpath online administration (SYSAFP) provides a wide variety of online operational and administrative services.

The following services are provided:

Service	Description
System Settings	Maintain Adabas Fastpath configuration file.
Parameter Maintenance	Define/maintain Adabas Fastpath parameters.
Current Activity Displays	Display Adabas Fastpath buffer information.
Buffer History	Display Adabas Fastpath buffer history information.
Printing	Print Adabas Fastpath reports.
Special Services	Perform miscellaneous actions for the Adabas Fastpath components.



Service	Description
Online Sampling (APFLOOK)	Perform optimization sampling.
About Adabas Fastpath	Display Adabas Fastpath release information.

For more information, see section SYSAFP.

## Buffer Operation and Reporting (AFPCMD)

---

The tool AFPCMD provides operating and reporting functions for the Adabas Fastpath buffer. AFPCMD can be invoked using Adabas Fastpath online administration or it can be invoked using Natural in batch mode.

For more information, see section AFPCMD.

