Defining and Tuning Optimization Parameters

This section provides guidelines for defining and tuning Adabas Fastpath parameters to achieve maximum optimization.

- Overview

- Buffer Parameters

- File Parameters

- Monitoring File Parameters

- Client Runtime Controls

Overview

Adabas Fastpath optimizes Adabas commands by storing formatted data in a common memory area (Adabas Fastpath buffer area) and returning this data directly to the application without calling the Adabas database. When this is successful, an Adabas command is optimized. The Adabas Fastpath buffer is initialized and managed by the asynchronous buffer manager (AFPABM).

Adabas Fastpath tuning is an ongoing process. Because tuning is partly achieved by changing controls and measuring the results, it is important to decide how you are going to measure the success of Adabas Fastpath in a dynamic production environment. You also need to decide the period over which you will measure the performance based on procedures at your site; for example, many sites have additional weekend and/or month-end processing.

If your environment already measures overall performance, you may decide to use these same measurements for Adabas Fastpath. You may choose to use Adabas Fastpath statistics for overall commands optimized or percentage of commands optimized at the buffer, database, file, or job level. It is important to have a basis for measuring the effect of changes so that you are aware of any decrease in performance.

Adabas Fastpath parameters, which are located in the Adabas Fastpath configuration file, are defined and maintained using Adabas Fastpath Online Services, Parameter Maintenance.

Adabas Fastpath counts the commands optimized and produces other statistics needed to tune its parameters. This information is also recorded on the Adabas Fastpath configuration file. This information can be displayed using the Adabas Online Services functions Buffer Statistics and Buffer History.

The following sections describe the three Adabas Fastpath parameter types: buffer, file, and client runtime control. These interact closely in that information from one parameter type may indicate that a parameter of another type may also need to be reviewed.

Section	Description
Buffer Parameters	The Adabas Fastpath buffer is initialized and managed by the asynchronous buffer manager (AFPABM). Each buffer is defined using buffer parameters which can significantly affect the overall performance of Adabas Fastpath. This section tells you how to monitor the buffer statistics and which buffer parameters you may need to modify.
File Parameters	File parameters are used to define the databases and files which are to be optimized. This section describes how to select files for optimization, define file optimization parameters, and measure the optimization results.
Client Runtime Controls	Each job that uses Adabas issues calls through a link module that includes the appropriate Adabas Fastpath module. This module uses a client runtime control located in the Adabas Fastpath configuration file to identify the Adabas Fastpath buffer to be used by this job. This section describes the optimization parameters available at the job level and how to measure their effectiveness.

Buffer Parameters

The parameters defined for the Adabas Fastpath buffer determine the size and structure of the buffer and have a significant effect on overall performance.

You can use the Adabas Fastpath Online Services to display the main Buffer Performance Measurement screen shown below:

```
16:49:32  ***** A D A B A S   FASTPATH SERVICES 8.1.2 (I001) *****  2006-10-09
                    -  General Parameters  -                       F12100MF
               Buffer: Global ICFDCOR1 Node ID: 650
                                       Initial                    Commands
 Main Parameters            Value      Allocation     Peak Usage   Ignored
 Size(k)..Min: 2047         2048          2047             43%
 Maximum Jobs:                12                           1            0
 RB Length Limit:           2048                                        0
 Set Length Limit:          1024                                        0


 Other Parameters           Value                                  Value
 Default pacing rate:       10           Set Concurrency:          2
 Freespace Index Entries:   512          Average Item Size:        256
 Log every n minutes:       60           Keep for n days:          30
 Autorestart:               Y            Restart every n Hrs:      4
 Restart Time:              08:00        Synchronous Remote Updates: Y
 Direct Access:             ON           Read-ahead Processing:    ON
 FIND     SX/L1:            ON           READ PHYSICAL L2:         ON
 READ LOGICAL L3:           ON           HISTOGRAM    L9:          ON

 Command ==>
Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
     Help        Exit        Serv  Excl  Gen                     Node  Menu
```

Each of the main buffer parameters shown on this screen is described in the table below.

The following information is provided:

- the parameter value currently defined;

- the initial memory allocation;

- the peak usage (which includes extra areas allocated); and

- the commands ignored because of an area shortage or limit having been exceeded.

Parameter	Description
Size	The buffer size (minimum, defined, initial allocation, peak usage). If the minimum is not available, the buffer cannot be initialized. In the example, an initial allocation of 4 megabytes has been made and the peak usage is currently 7%. Any value above 97% is regarded as 100% usage (there is usually a small amount of unused residue because of fragmentation). A fully used buffer does not necessarily mean that the size needs to be increased. Other indicators must also be considered; for instance, low direct access optimization results. It is not possible to define a general formula for increasing this parameter. The best approach is to redefine the size and then measure the overall effect on performance. If performance is improved, you can repeat the process until there is no further performance gain.
Maximum Jobs	The maximum number of jobs permitted for the buffer. This value is used to preallocate the number of areas needed for all Adabas Fastpath jobs. Each Adabas Fastpath job is allocated a job area when it is started. When a job ends, the area is freed and can be reused. Peak usage is the highest number of areas which have been allocated. This parameter value should be increased if • the peak usage value is the same as the defined value; or • the number of commands ignored due to shortage of areas is high.
RB Length Limit	The maximum length of a record buffer for direct access optimization processing. Commands ignored because of the limit are also shown for this parameter. If a large number of commands is being ignored, this is an indication that this parameter value could be increased to achieve better optimization.
Set Length Limit	The maximum length of a set used for direct access optimization processing. A set is made up of the Adabas format buffer (FB) and search buffer (SB) plus an extra 12 bytes. Commands ignored because of the limit are also shown for this parameter. If a large number of commands are being ignored, this parameter value could be increased to achieve better optimization.
Other Parameters	For information on all other parameters, see the section Adabas Fastpath Parameters.

File Parameters

The section describes how to select Adabas files for optimization, and how to define individual file parameters for those files which have been selected for optimization.

- Selecting Files for Optimization

- Database Potential Optimization Summary

- Defining File Optimization Parameters

- Default File Definition for All Files in a Database

Selecting Files for Optimization

Adabas Fastpath provides the AFPLOOK tool for sampling an Adabas database and reporting on the suitability of individual files for Adabas Fastpath optimization. This section provides general guidelines regarding the contents and usage of the information provided by AFPLOOK.

Note:
For additional documentation on AFPLOOK, see AFPLOOK and Using AFPLOOK from Adabas Online Services.

The AFPLOOK samples provided in this section were created using Adabas Online Services and are based on the following conditions:

- the database is sampled in its entirety with no focus on particular files or jobs,

- the sampling session extends over a period long enough to be representative of the normal workload,

- any other databases in use are sampled and analyzed in the same way so that the whole system is optimized, not just a part of it.

The sample presents a realistic use of AFPLOOK with the goal of understanding the general workload presented to the database. The screens included in this section are taken from an in progress sampling session. Therefore, the numbers contained in the detailed screens do not correlate with the numbers in the summary screens. If necessary, you can "freeze" a sample so that detail and summary numbers do correlate by requesting the sample to stop. Sampling stops, but the numbers are retained for analysis purposes. You can then release the sample and continue the session.

AFPLOOK sampling results provide the information needed to initially define optimization parameters. You may need to adjust these parameter settings after observing the results of subsequent processing.

AFPLOOK reports on the commands processed by the database. If you are already optimizing with Adabas Fastpath, AFPLOOK does not account for the commands which are already optimized, since they are not sent to the database.

For files where optimization has been defined, AFPLOOK only counts commands that either have not been qualified or have failed optimization. Optimized direct access commands do not appear on the report at all, even in file totals. Optimized read-ahead commands do not appear either, but the command used to prime read-ahead is counted in the prefetch counter, indicating that prefetch/multifetch has already been set for these commands.

Files where no optimization has been defined show accurate figures *except where a default file 0 has been defined* for general read-ahead for the database. In this case, any file can be affected and will ignore optimized commands and show counts in the prefetch counter.

When interpreting a file sample, it is important to remember that the AFPLOOK view can be seriously affected when Adabas Fastpath is active. Because optimized commands are not seen by the sample, even the file totals and relative percentages are affected. Therefore, do not be misled into thinking that the file parameters defined are not needed when in fact, they are performing well.

Database Potential Optimization Summary

The Potential Optimization Summary screen provides potential optimization information for an Adabas database.

```
12:33:10 ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****        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

In the sample screen, the field Sampled File Commands shows a total of 77,075 commands processed by Adabas during the sample period of which 75% are considered to have optimization potential. The field Excluded Commands shows that 25% are not considered to have optimization potential.

The full optimization potential is generally not fully achievable. Experience shows that approximately 50% of the potential is most likely to be achieved. In this example, 30-40% optimization can be expected for this database which would result in an approximate Adabas CPU usage reduction of approximately 20%.

You can use PF11 to display more information on Excluded Commands.

```
12:33:41 ***** A D A B A S   FASTPATH SERVICES 8.1.2 *****          2006-05-19
               - Afplook - Summary for DB 50  -                      F16300MB

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

--------------------------- Command Analysis ------------------------------
 Rejected Commands:
         Max.Users Exceeded:
         Max.CIDs Exceeded:
         Max.Files Exceeded:
 Excluded Commands:
         Bad Commands:
         Non-File Commands:              431
         Non-File RCs:                  1718
         Excluded File Commands:        6015
         Update Commands:              10951
         Already Prefetched:                     19115 (25%)
 Sampled File Commands:                          57960 (75%)
 All Commands Seen:                              77075 (100%)
 ------- Parameters and High Water Marks      Reused User Areas:
 Max.Files: 64    Files Needed: 28       Max. DE: 32    Overflows:
 Max.Users: 100   Peak Users:  1         Max.CID: 10    Peak CIDs: 5



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

AFPLOOK Command Analysis Summary

The section Rejected Commands shows the number of commands which were rejected as a result of the maximum number of users, maximun number of Command IDs, or maximum number of files being exceeded.

The section Excluded Commands shows the number of commands excluded because of the following reasons:

- Bad Commands: commands not recognized or invalid.

- Non-File Commands: commands which are not related to file processing. For example, ET,BT,OP,CL.

- Non-File RC Commands: Release Command ID commands which are not related to file processing.

- Excluded File Commands: access commands which cannot be optimized. For example, L4,L5,L6,S4.

- Updated Commands: update commands which cannot be optimized. For example, A1, E1, N1.

- Already Prefetched: prefetching may exclude a significant number of commands that otherwise could be optimized. If so, it may be worth running the sample without prefetching to achieve a more realistic result.

The parameters used to control the AFPLOOK sample are displayed at the bottom of the screen. The limits set for these parameters may result in the rejection of commands during sampling. Rejections, if any, are also displayed on this screen. If the number of commands rejected is significant, you should increase the value for the relevant parameter and then execute the sample again.

You can use PF10 on the Database Summary screen to invoke the File Summary screen to determine the most appropriate files for optimization.

Note:
The list of files usually extends to more than one screen. Because of space restrictions, some files have been removed from this example.

```
12:31:26  ***** A D A B A S   FASTPATH SERVICES 8.1.2 *****           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 following information is provided for each file:

Field	Description
Total	The total number of commands processed for the file. The accompanying percentage can be used to identify quickly the busiest files, especially when the file summary is spread across several screens.
Direct Access	The number of direct access commands processed for the file.
RCs	The number of RC commands processed for the file. The Adabas RCcommand releases one or more Command IDs currently assigned to a user. See the *Adabas Command Reference* documentation for more information. When a direct access command is optimized, it is often possible to optimize its matching RC command.
Sequential	The number of sequential access commands processed for the file.

In the example, file 137 is by far the busiest file (43% of database commands are for this file), and 95% of the sample for file 137 has potential optimization (55%+33%+7%).

By marking File 137, further details can be displayed for the purpose of selecting the appropriate optimization parameters for this file.

```
12:32:21 ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****            2006-05-19
             - 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                             Inserts 0
  L3 S1      23670      16889        6781      6781        Deletes 0
  L1 --       6748
  L9 TA      13032      13032                             Prefetch 0
  L9 S5       9326                  16478      7896        Excluded 32
                                                          Spare 0
                                                          No CID 0
                                                          Max.CIDs 0
                                                          Bad Rsp. 0
                                                          Max.Rec.Buff.Len
 Total       131262      77522       30011                 Direct Acc 69
 Total Commands for File                      253970       Sequential 63



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

AFPLOOK File Details Display

This screen shows the commands that can be optimized and summarizes those that cannot.

The following information is provided:

Field	Description
CC DE	The Adabas command and descriptor pair for which information is being provided.
Direct Acc	The number of direct access commands which have optimization potential.
RCs	When a direct access command is optimized, it is often possible to optimize its matching RC command. This column indicates the potential number of RC commands which can be optimized.
Sequential Sequences	As a general rule, read-ahead optimization is only appropriate for descriptors with at least a 2 to 1 ratio of sequential commands to sequences. If the ratio is smaller, over-reads could occur if the descriptor is optimized. In the example, for the command/descriptor pair L3/S5, there are 6,752 sequential commands which are separated across 394 sequences yielding an average sequence length of approximately 17. Read-ahead optimization is appropriate in this case. When optimization is defined for this descriptor, Adabas Fastpath automatically determines the read-ahead factor using an accelerating algorithm: the longer the sequence, the higher the read-ahead factor. The example also shows, for the command/descriptor pair L3/S1, 6,781 sequential commands which are separated over 6,781 sequences. In this case, read-ahead is not appropriate and should not be defined for this descriptor. RC commands cannot be optimized for sequential (read-ahead) access.
Non-Qualifying	Commands that cannot be optimized. Also shown are the record buffer lengths used for direct access and sequential views.

Before defining the optimization parameters, it may be useful to have a copy of the AFPLOOK file detail statistics for the selected files to refer to when you are monitoring the results.

PF5 can now be used to define the optimization parameters as described in the next section.

Defining File Optimization Parameters

This section describes the definition of file optimization parameters using various file examples:

- Direct Access Optimization Parameters
- Read-Ahead Optimization Parameters

Direct Access Optimization Parameters

As shown on the example screen below, for file number 135, all L3 and S1 commands potentially qualify for direct access optimization.

```
12:32:21  ***** A D A B A S   FASTPATH SERVICES 8.1.2 *****        2006-05-19
              -  Details for DB  50   File 135       -        F16210MA

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

  CC DE Direct Acc       RCs  Sequential  Sequences       Non-Qualifying
  S1 AA    10188       10188                              Updates 6749
  L3 AA     3396        3396                              Inserts 0
  S1 A1     6748        6748                              Deletes 0
  S1 BS     6717

                                                         Prefetch: 0
                                                         Excluded: 0
                                                         Spare   : 0
                                                         No.CID  : 0
                                                         Max.CIDs: 0
                                                         Bad Rsp.: 0

                                                         Max.Rec.Buff.Len
  Total       27049       20332                           Direct Acc 648
               Total Commands for File       47381         Sequential


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

File 135 File Details

Direct access optimization is therefore defined as ON for L3 and S1 commands without any field level (descriptor) overrides:

```
               On/Off  Except for:   Field level overrides
 L1 Get ISN        OFF
 L3 Read Logical   ON
                                    __ __ __ __ __ __ __


 L9 Histogram      OFF
 S1 Find Records   ON               __ __ __ __ __ __ __
                                    __ __ __ __ __ __ __


 S2 Find Sorted    OFF              __ __ __ __ __ __ __
```

Defining commands for optimization without descriptor overrides allows the automatic inclusion of other descriptors that may not be visible in this example or may be used to access the file in the future. This method is a good choice for

- a reference type file where you anticipate direct access optimization; or

- a file with more than seven descriptors, all with good optimization potential, for a command.

In the next example, direct access optimization is to be performed only for L3 commands which use the field AA as primary descriptor and for S1 commands which use the field AA, A1 or BS as primary descriptor. This method is a good choice for a file with many descriptors and for which most of the database access is performed using specific descriptors.

Note:
Monitoring the performance of file optimization at descriptor level is the same whether generic command definitions are used or not. However, the result of monitoring may require that you change from generic to explicit descriptor definitions, or that you explicitly exclude certain descriptors.

```
                    On/Off   Except for:   Field level overrides
L1 Get ISN           OFF
L3 Read Logical      OFF                  AA  __  __  __  __  __  __  __


L9 Histogram         OFF                      __  __  __  __  __  __  __
S1 Find Records      OFF                  AA  A1  BS  __  __  __  __


S2 Find Sorted       OFF                      __  __  __  __  __  __  __
```

Read-Ahead Optimization Parameters

In the example below, for file 143, there is read-ahead optimization for the command/descriptor pairs `L3/S2, L3/S3` and `L3/S4`, as shown by the values in the Sequential and Sequences fields.

```
12:32:21 ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****          2006-05-19
            - Details for DB  50   File 143   -                      F16210MA

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

  CC DE Direct Acc       RCs  Sequential  Sequences        Non-Qualifying
  S1 S3     10142     10142                               Updates  16921
  L3 S2     23701     20311       6780        3390        Inserts      0
  L3 S3       500               306971         654        Deletes      0
  L3 S4     12403                12401       12401
  L3 S7       134                                         Prefetch     0
  L3 S9        99                                         Excluded 16921
                                                          Spare    :   0
                                                          No.CID   :   0
                                                          Max.CIDs:    0
                                                          Bad Rsp.:    0

                                                          Max.Rec.Buff.Len
   Total     46979     30453      326152                   Direct Acc  195
              Total Commands for File     406974           Sequential  195


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

File 143 Detail Display

In that the `L3/S3` pair has a relatively low volume of commands, only the pairs `L3/S2` and `L3/S4` are defined for optimization.

```
                    On/Off   Except for:   Field level overrides
L1 Get ISN           OFF
L3 Read Logical      OFF                   S2  S4  __        __  __  __


L9 Histogram         OFF                       __  __  __  __  __  __  __
S1 Find Records      OFF                   S3  __  __  __  __  __  __  __


S2 Find Sorted       OFF                       __  __  __  __  __  __  __
```

However, in the case of read-ahead parameters, descriptor S4 should not be defined because the number of sequential commands is equal to the number of sequences which means that no real read-ahead capability exists. Therefore, it is necessary to explicitly define descriptor overrides for L3 commands:

Note:
If descriptor S4 were defined for optimization (or implicitly allowed), a large number of over-reads would probably result.

```
                    On/Off   Except for:   Field level overrides
L1 Read by ISN       OFF
L2 Read Physical     OFF
L3 Read Logical      OFF                   S2  S3  __  __  __  __  __


L9 Histogram         OFF                       __  __  __  __  __  __  __
S1 Find Records      OFF                       __  __  __  __  __  __  __
S2 Find Sorted       OFF                       __  __  __  __  __  __  __
S8 Process ISN List  OFF
S9 Sort ISN List     OFF
```

Read-ahead parameters can also be defined by setting L3 command optimization ON and by specifying the descriptors which are not to be optimized.

```
                    On/Off   Except for:   Field level overrides
L1 Read by ISN       OFF
L2 Read Physical     OFF
L3 Read Logical      ON                    S4  __  __  __  __  __  __


L9 Histogram         OFF                       __  __  __  __  __  __  __
S1 Find Records      OFF                       __  __  __  __  __  __  __
S2 Find Sorted       OFF                       __  __  __  __  __  __  __
S8 Process ISN List  OFF
S9 Sort ISN List     OFF
```

Default File Definition for All Files in a Database

A file parameter defined with a file number value of 0 indicates a default definition for all files in the database that do not have an explicit file parameter definition. This can be useful for:

● defining read-ahead optimization, but no descriptor overrides, for all affected files.

- a database with many files for which only read-ahead optimization is appropriate.

Note:
Direct access optimization is not possible by default.

All statistics for the affected files are shown under the default. This makes it difficult to monitor individual files. For example, if over-reads are high, you may not be able to identify the file that is causing the problem.

If you do identify the file causing over-reads, you can switch off read-ahead optimization for the file by defining a file parameter for that file.

Monitoring File Parameters

The section describes how to monitor the performance of the currently defined optimization parameters.

- Reviewing File Optimization Performance

- Excluded Commands Analysis

- Direct Access Analysis

- Read-Ahead Analysis

Reviewing File Optimization Performance

You can use Adabas Fastpath Online Services to display a summary of the file optimization performance for a given file.

The following example shows the summary for file 137 in database 50.

```
15:26:19     ***** A D A B A S   FASTPATH SERVICES 8.1.2 *****     2006-06-07
            Database Details from Global A8P17004 Node ID 17004 **   F12200MB


                        Database: 17003 SVC: 254


 -------------------------- Optimization Summary ---------------------------
      Access Type          Attempts  (87%) <----------Successes  (65%)

      Read-ahead:           2,918  (32%)              1,765  (60%)
      Direct:               3,914  (43%)              2,313  (59%)
      RCs:                  1,170  (12%)              1,170  (29%)
      Others:               1,062  (11%)

      Total:                9,064 (100%)              5,248  (65%)
 ----------------------------------------------------------------------------
      Updates:                          Over-reads:              (100%)

  Set Limit(k): 4095                              Update Sensitivity: R
  Set Id Length Limit: 1024   RB Length Limit: 100   Set Concurrency: 2
  Status: 00 D0   Last RC:        Next Event:  0:00   Addr: 0FC76FD0

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

File Details Part 1

With an overall optimization success rate of 65%, both direct access and read-ahead optimization can be considered to be successful.

Nonetheless, it may be possible to obtain a higher optimization percentage by reviewing more detailed information which is provided using PF6.

```
15:30:23      ***** A D A B A S   FASTPATH SERVICES 8.1.2 *****      2006-06-07
              -  File Details for File 9     DBID 17003 SVC 254  -     F12210MF

     Statistics for commands that have optimization potential but were
     unable to be optimized:

          Unsuitable command type:                      20
          Disqualified:                               1023
          Optimization was disabled:                     0
          Multi-fetch was already set:                   0
          Command details exceed Set-id limit:          0
          Record Buffer exceeds the limit:             12




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

The information provided on this screen is described in the following sections:

- Excluded Command Analysis

- Direct Access Optimization Analysis

- Read-Ahead Optimization Analysis

Excluded Commands Analysis

The first File Detail screen for file 137 shows that 11% of the commands are excluded from optimization.

The second File Detail screen provides the specific reasons for exclusion. It shows that 1023 commands were excluded because the descriptor was not qualified. By referring back to the saved AFPLOOK output for file 137, on which the parameter selection was based, it can be seen that L3 commands with descriptor S1 were not optimized, which explains the large number of excluded commands.

Note:
The information on the right hand side of the screen is discussed in subsequent sections. In addition, a description of each exclusion category is provided in section Excluded Command Summary, together with information about the parameters that affect each reason.

Direct Access Analysis

The first File Details screen for file 137 shows that direct access optimization is 59% successful and an additional 29% of RC commands are optimized.

- Direct Access Sets
- Set Details
- Set List
- Controlling Individual Sets

Direct Access Sets

Adabas Fastpath maintains direct access data by database and file in a *set*. Each set has a unique data profile. For any given Adabas command/descriptor pair defined for optimization, multiple sets can result each with a different profile.

The screen example below shows set information for each command/descriptor pair for file 137.

This screen provides the information necessary to decide whether or not the currently defined file optimization parameters defined have been successful. A line is shown for each command/descriptor pair for which direct access optimization sets have been built. Each line shows the Adabas command, the primary descriptor and the number of direct access optimization sets that have been built for this combination.

```
10:38:27   ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****              2006-05-19
          -  Set Summary from Global  SYSCO33 Node Id 33 -             F12220MA

                        Database: 50     SVC 240 File: 137


               Number of Items....              Memory (k).... Efficiency Ratings
  C CC De Sets    In Use       Peak  Optimized   In Use    Peak   Set Memory Growth
  _ L1        1      49         56        11         44      44     0      1   0.01
  _ L3 S1     1      51         57       1557         5       6     9    311   2.99
  _ L3 S5     2     220        234       1887        30      32     1     62   0.59
  _ L9 S5     1      20         24        404         1       2     9    404   3.88
  _ L9 TA     1     120        124       1427         3       3     9    427   4.10
  _ S1 S5     1       2          8        758         7       7     9    108   0.90




   Mark with D(etail),F(ull Set List)


   Command ==>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12
       Help         Exit  Refr  Parms Look                  Cntl  Node  Menu
```

File Set Summary List

In this example, the number of sets is low. In a heavy production environment, it is possible for some command/descriptors pairs to have a large number of sets. For each set, the information described in the table below is provided. The values provided for each command/descriptor pair are an accumulation of the individual sets, except for those values provided in the column Efficiency Ratings, which are recalculated from various totals of the individual sets.

Field	Description
Number of Items	In Use: the number of items (data records) currently in the sets. Peak: The highest number of data records which has been used for the sets. Optimized: the number of times data records in these sets were used for direct access optimization. The higher the number of optimizations relative to the values for In Use and Peak, the better.
Memory	In Use: the amount of memory allocated from the Adabas Fastpath buffer currently in use for the sets. Peak: the highest amount of memory allocated from the Adabas Fastpath buffer which has been in use for the sets.
Efficiency Ratings	The efficiency ratings for set, memory and growth. These ratings are calculated from the above values to obtain measurements that can be used to judge the performance of the command/descriptor combinations.
Efficency Rating (Set)	The effort needed to maintain the sets (adding, deleting, overwriting) against the (nominal) effort saved by optimization. ● Any rating above 0 shows that, overall, the sets for this command/descriptor pair are optimizing efficiently. ● A rating of 0 means that the command/descriptor pair is consuming more effort in maintaining the sets than it is saving by optimization and should be removed from file optimization. However, all sets start with a 0 rating, so enough time must be allowed for proper calculation of this rating. In addition, sets that are used infrequently (for example, by monthly batch jobs) may show a 0 rating prior to their usage period. ● A rating of 1 is of interest only if the command/descriptor pair uses a high level of memory and has a low memory value. In this case, releasing the memory would benefit other sets or Adabas Fastpath generally.
Efficiency Rating (Memory)	The number of optimizations against the memory used. The higher the rating, the better the value of the sets in terms of memory usage.
Efficiency Rating (Growth)	Growth efficiency puts the memory value into perspective by relating it to the average for the whole buffer. A rating of 1.00 is average.

In the example screen, the `L1` command (descriptors are not applicable for `L1s`) has a set efficiency of 0. The memory value is also low. Therefore, optimization for this command should be removed unless the performance is likely to change.

Set Details

In the File Set Summary screen, the `L3` command with descriptor `S5` shows a set efficiency of 1 and a memory value that is approximately half of the average for the buffer. By using the D option next to the `L3`/`S5` pair, more detailed information for the sets used by this pair can be displayed:

```
10:38:27   ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****        2006-05-19
          - Set Summary from Global  SYSCO33 Node Id 33  -          F12220MB

                     Database: 50    SVC 240 File: 137

   Command: L3       Primary Descriptor: S5         Number of Sets: 2

   Efficiency Ratings:   Set:  1   Memory:       62 Growth:        0.59

   Items - Optimized:      1887   Added:        220 Collisions:        6

           Current:         220   Peak:         234 Free:             14

           Removed:               RLU:              Overwrites:     1433

                                                    Set Resets:

   Memory- Used(k):         30 Peak(k):        32 Free(k):           1



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

File Set Summary Detail

This screen shows the totals for the two sets that have been created for the command L3 descriptor S5 combination.

In addition to the fields previously described on the preceding screen, this detail screen shows the following fields:

Field	Description
Collisions	The number of times a set was inaccessible because concurrency for the set was exhausted.
Overwrites	The number of overwrites which have occurred for the sets.
	On overwrite occurs when Adabas Fastpath tries to insert a new data record but finds that there are no free item areas available for the set. When this happens, the set is marked for expansion and an aging algorithm is used to determine if a record is suitable for deletion. If so, this record is overwritten by the new record.
	In this example, there is an unusually high number of overwrites relative to the number of items added. It is common for sets to have overwrites, especially when first being built, but these usually decrease as a set matures.
	This summary in the example shows a typical profile of sets with a low data reusage ratio. The number of optimizations only minimally exceeds the total of Added and Overwrites, which often indicates that a data record is added to the set but then only optimized once or twice. In this type of set, data records can continue to be inserted and memory can continue to be requested until all the buffer memory is used. Optimization continues to be based on the latest record inserted, so the oldest data is no longer referenced and is therefore wasted.
	Adabas Fastpath recognizes the pattern of behavior and, because the growth rate is below average, restricts the set expansion. This properly causes the sets to overwrite old data rather than continually insert additional data.
	In some cases, it may be helpful to look at the various individual direct access sets that compose the set summary for a command/descriptor combination.
Set Resets	The number of times a complete set was emptied.

Set List

In the File Set Summary screen, entering the F option next to the L3/S5 pair invokes the Command Descriptor Set List screen:

```
10:39:56   ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****            2006-05-19
          -  Set Summary from Global  SYSCO33 Node Id 33 –            F12230MA

                Database: 50    SVC 240 File: 137    CC: L3 De: S5


            Number of Items......            Memory (k).....  Efficiency Ratings
  C   SSID       In Use      Peak  Optimized  In Use    Peak   Set Memory Growth
  _      8          218       226       1489      30      31     1      49   0.47
  _      6            2         8        398                     9     398   3.82




   Mark with D(etail),S(start/stop)


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

Command Descriptor Set List

This screen shows different behavior profiles for the two sets:

- The first set has a higher volume but low efficiency

- The second set has low volume but high efficiency.

Although together the sets show a positive contribution to the overall Adabas Fastpath efficiency, the sets do use 30k of memory from the Adabas Fastpath buffer. In other examples, the memory involved could be several megabytes. If memory shortage is causing lost optimization for other databases/files/sets or read-ahead, this command/descriptor pair could be removed from optimization as part of an overall strategy to recover allocated memory.

Controlling Individual Sets

The file parameters defining optimization can only be defined down to the command/descriptor level. While it is possible to stop and start individual sets, this is only useful on a short-term basis because the set is restarted every time the buffer, database, or file is restarted.

The Set Details screen can be displayed for reference while considering additional options for controlling individual sets.

```
10:40:31 ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****            2006-05-19
           - Set Detail from Global  SYSCO33 Node Id 33  -            F12230MB

                        Database: 50    SVC 240 File: 137
  Command: L3    Primary Descriptor: S5                     SSID: 8

   Efficiency Ratings:   Set:  9 Memory:         398  Growth:           3.82
   Items - Optimized:       1489 Added:          218  Collisions:          6
           Current:         218 Peak:            226  Free:                8
           Removed:              RLU:                 Overwrites:       1433
                                                      Set Resets:
   Memory  - Used(k):        30 Peak(k):          31  Free(k):             1
   Status:   00 C8 0E 08        Expansion:         8  Acc.Mask Entries     2
   Used      FB:    54 RB:    70 Indexes:          3  Address:      09A00234
   Lengths: SB:     9 VB:    20 Code:               Lock:         00000000
   SetID Length:    75
 Reject:
 SB AE,020,A.
 FB AA,008,A,AC,020,A,AE,020,A,AD,020,A,AF,001,A,AG,001,A.


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

Set Details

For some areas, you can set general buffer or file parameters that have an effect on individual sets. The parameters apply to all the sets within a file:

Buffer/File Parameter	Description
Set Limit (k)	Sets the maximum memory allocation for each set within the buffer or file. If a set uses a lot of memory but you want to continue optimizing the command/descriptor pair, you can restrict the file parameter to limit the maximum memory allocated to a set within the file.
Set ID Length Limit	Defines the maximum length of the unique set identifier. A set identifier comprises 12 bytes control informatin plus the Adabas search and format buffers. By reducing the value of this parameter at the file level, you can prevent optimization on certain sets within a file.
RB Length Limit	Sets the maximum record buffer length for which sets will be built. The Set Details screen shows the actual record buffer length used (in the case of variable length sets, it is the maximum RB length used so far). By restricting the value of this parameter at the file level, you can prevent optimization on certain sets within a file.

Read-Ahead Analysis

The first File Details screen for file 137 shows a successful read-ahead optimization percentage of 60%.

- Over-reads

Over-reads

During read-ahead processing, Adabas Fastpath stores a number of records that have been prefetched for a user, with the anticipation that the application will subsequently require these records. Inevitably, occasions arise when all the prefetched records are not required and are therefore discarded. Such records are counted as *over-reads*.

Adabas Fastpath uses an accelerating algorithm to decide how many records to read ahead to minimize over-reads. Consequently, it is rare to see a high level of over-reads. Because of the way the algorithm slowly accelerates the read-ahead factor, high levels of over-reads only occur on sequences with a low number of commands.

The first File Details screen for file 137 shows 668 over-reads which is 37% of read-ahead optimizations. This is high since over-reads are normally kept below 25%.

The File Details screen can be used to determine which command/descriptors are creating over-reads:

```
10:37:21 ***** A D A B A S  FASTPATH SERVICES 8.1.2 *****             2006-05-19
        - File Details for File 137   DBID 50 SVC 240 -            F12210ME
                        Expanded File N


              On/Off Adabas Field Overrides     ........Direct Access......
          L1   On
          L3   Off    S1      S5
          L9   Off    S5      TA
          S1   Off    S5
          S2   Off
  Over-reads   Default Adabas Field Overrides     ........Read-ahead.........
          L1     Off
          L2     Off
      82  L3     Off  S5
     586  L9     Off  S5
          S1     Off
          S2     Off
          S8     Off
          S9     Off



  Command ==>
 Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12
      Help        Exit              Excl  Gen                    Node  Menu
```

File Details Part 2

This screen shows that most of the over-reads (586) result from the L9 command with descriptor S5.
Referring back to the AFPLOOK File Details Display, it can be seen that the L9 sequence ratio is just
over the minimum of 2. Low command-to-sequence ratios are the main cause of high over-reads.

To reduce over-reads to less than 20%, read-ahead optimization for the L9 command for this file can be
switched off or the descriptor override can be removed.

Although you can usually identify the command/descriptor pair causing the over-reads by finding low
command-to-sequence ratios in the AFPLOOK analysis, you may occasionally need to switch off
descriptors one-by-one to determine which descriptor is causing the most over-reads.

Client Runtime Controls

The Job Details second screen displays information about the work areas used while optimizing
commands for the job. These areas are dynamically allocated and need no tuning parameters. The display
is for information only:

```
18:45:17     ***** A D A B A S   FASTPATH SERVICES 8.1.2 *****     2006-06-07
              Job Details from Global A8P17004 Node Id 17004 **     F12300MC

 Job:    UKRPE Type: TSO       Number: U36025          Started Jun  7,2006 18:42

 Job Information-------------------------------------------------------------------
   Peak Concurrency...: 2
   User Areas:  In Use: 2       Peak: 2           Recovered: 0
   CID  Areas:  In Use: 0       Peak: 3
 Read-ahead Memory Categories-----------------------------------------------------
   Size(k):  4 Peak No: 0      Usage.: 8          Failures: 0
             8         0             1                   0
            16         0             0                   0
            32         0             0                   0
 Client Runtime Controls----------------------------------------------------------
              Job End Stats..: N           Direct Access: ON
 Read-ahead Optimization Control: BAT    Read-ahead Limit(k): 0



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

Job Details 2

Read-Ahead Parameters

The Job Details 2 screen shows the defined job parameters, including the read-ahead parameters.
Although no explicit performance indicators exist, a general low read-ahead optimization level indicates
that these parameters should be reviewed.

Read-Ahead Optimization Control

The read-ahead rate is accelerated automatically by algorithm: the longer the sequence continues, the
more the rate increases. In this way, optimization is maximized and over-reads are avoided. Because batch
jobs normally use longer sequences than online tasks, different algorithms are available: BAT for batch
jobs and TP for TP monitor jobs.

You can choose either algorithm for any job. Some sites may find that the BAT algorithm works better in
their particular TP systems than the default TP algorithm. The only way to determine the best option is to
try out the algorithms and measure the results. This procedure applies as well when choosing the correct
algorithm for such job types as TSO or TIAM.

Read-Ahead Limit

The Read-ahead Limit parameter can be used to restrict the amount of memory allocated to read-ahead
optimization for the job. This can be useful if a particular job or TP monitor is running short of local
memory because of Adabas Fastpath read-ahead allocations.