

GiAPA

by iPerformance

A one-minute introduction to the new GiAPA Menu option 20

Automatically generated optimization hints for:

- **Programs**
- **Access of data bases**

New Menu Option 20 Added to the GiAPA Menu

GiAPA (c) by iPerformance

GiAPA V05M00

GiAPA Menu

POWER720 on 06E84CT LPAR 00001

KAARE

DATA COLLECTION AND ANALYSIS

- 11 Submit performance data collection
- 12 HotSpot watch of one selected job
- 13 End performance data collection
- 14 Expand and analyze collected data

DISPLAY/PRINT RESULTS

- 15 Job performance summary reports
- 16 Reports on *ALL data (when kept)
- 17 Job or user name summary
- 18 HotSpot count summaries
- 19 Program and file performance analysis
- 20 Program and file optimization hints
- 21 Collection interval summaries
- 22 File analysis based on HotSpots
- 23 Jobs having priority modified
- 24 CPU usage per current user
- 25 M3 Movex user/class statistics
- 26 User defined graphics
- 27 Start GiapaNavigator (Charts + GUI)
- 28 Work with created graphics data

IBM PERFORMANCE EXPLORER

- 31 Start PEX statistics data collection
- 32 End PEX statistics data collection
- 33 List call stack based on PEX data

DETAILED JOB TRACE

- 41 Start trace of job
- 42 End trace of job
- 43 Analyze trace job data

DATA BASE UTILITIES

- 51 Collect file check data
- 52 Run file check analysis reports
- 53 List index generations
- 55 Sort Database file

TRACK USE OF QUERY

- 64 Start RUNQRY and WRKQRY tracking
- 65 End RUNQRY and WRKQRY tracking
- 66 List RUNQRY and WRKQRY usage

EXPORT AND IMPORT GIAPA DATA

- 71 Export GiAPA raw performance data
- 72 Export GiAPA analysis results
- 73 Import GiAPA raw data or results

INSTALLATION PARAMETERS

- 74 Define loop trap exceptions
- 75 HotSpot and Optim.Hint exceptions
- 76 Maintain color palettes for graphics
- 77 Define jobnames for special analysis
- 78 Installation parameters

HOUSEKEEPING

- 81 Manage unexpanded pfr.data members
- 82 Manage expanded data members
- 83 Delete Performance Explorer data
- 84 Delete trace job data
- 85 Delete file check data
- 87 Delete RUNQRY/WRKQRY tracking data
- 89 Check if authority OK for pfr.coll.

- 99 Display GiAPA Command Menu

F2=Cmd.Line F3=Exit

Licence code type: L

Select option: █

Data library: GIAPALIB

GiAPA Menu option 20: Program and file optimization hints

```
GiAPA (c) by          Select input member for generation of          21-02-01
iPerformance         Optimization Hints for Programs and File Access  19:29:17
                     Data Library: GIAPALIB

Select (generic) pgm or file *ALL          Show savings exceeding 010 minutes

Select between two output formats
Show in HTML window:  1=All results    2=Program hints    3=File access hints
Use 5250:             4=Totals         5=All results    6=Program hints    7=File access hints

Opt  Member      Date      Text
--  -
_   E2021JAN29   210129   Data from 01 days starting Jan 29 (ExpDate 21097)
_   E2021JAN28   210128   Data from 01 days starting Jan 28 (ExpDate 21096)
_   E2021JAN27   210127   Data from 01 days starting Jan 27 (ExpDate 21095)
_   E2021JAN26   210126   Data from 01 days starting Jan 26 (ExpDate 21094)
_   E2021JAN25   210125   Data from 01 days starting Jan 25 (ExpDate 21093)
```

Output Formats: HTML Window or 5250 “Green Screen”

- HTML format includes a column chart showing potential savings – intended for management presentations
- 5250 format shows the same information in a more raw format

Select option 4 to check the total potential savings possible of run time

Selection T for Total provides overview of potential savings found

```
GiAPA (c) by                               Statistics from Automated Application Performance Analysis          21-03-30
iPerformance                               Library GIAPAUTILI           Member EXAMPLES                               13:55:30

      21,538 data collection intervals processed = data from 3 days 17 hours 45 minutes
14-08-02  5:45 date and time for first data included in analysis (YY-MM-DD hh:mm)
14-08-06  0:00 date and time for last data included in analysis (YY-MM-DD hh:mm)
103,715,178 job and task records received from Performance Collector API
37,902,572 showed resource usage --> record generated
1,147,656 different jobs and tasks found in API data
893,509 HotSpots detected (Job exceeded interval limits)
951,490 program call stacks retrieved
10,357,238 program names processed
72,473,827 open file data records processed
```

```
Source machine specifications:
GiAPA version          V05V00
System name            MAINSERV
Serial number          781X22C
Processor type         EPA1
Model & Server Model   E8B
Price group            P20
Op.System version      V7R1M0
LPAR number            021
Number of LPARs        3
Nbr of Phys. CPUs     18
Processor capacity     18.00
PVU per processor      100
Available memory Mb    457,179,136
Auxiliary storage Gb   45,897,128
System ASP Gb          34,012,316
System ASP use pct     72.0450
```

Potential Savings Found by Automated Application Performance Analysis

52	Improvements of Program Functions	2,176 Minutes
18	Improvements of File Access Method	628 Minutes
*** Total Potential Run Time Savings		46 Hours 44 Minutes

Example of Optimization Hint for a Program

GiAPA
by iPerformance

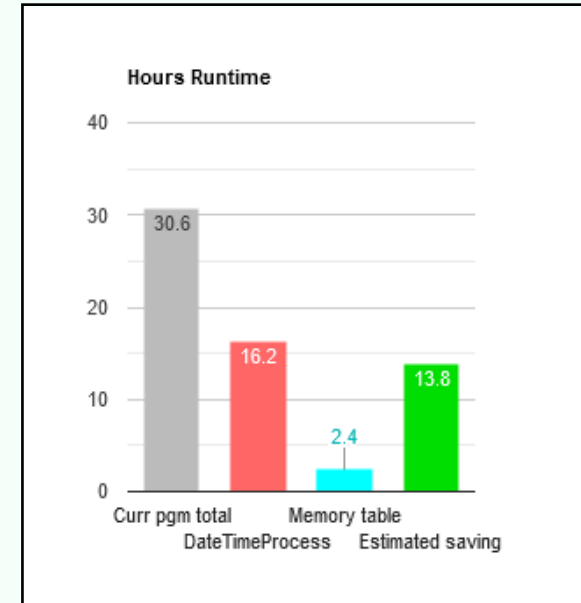
Program Optimization Hint

System: MAINSERV
781X22C LPAR 021

95.3 hours of data collected starting 2021-01-29 at 00:01

Program used RWONMN/OMENPDHPZ Calculate interest for outstanding invoices
Statement number 46900
GiAPA detected Date/time conversion or calculation found in 3907 HotSpots
Job and user UBSTVABZY4 KVKZKDV (4 jobs)
 UBSTVABZY7 KVKZKDV (4 jobs)

Estimated saving 85 % of DATETIME = 830 minutes run time
Effort required Probably < 7 hours programmer time (test not included)



Technical explanation

The process needed for date/time format conversions or calculations is rather CPU intensive

Tips on how to optimize the performance

Date/Time conversions, and calculations on date and time fields may be convenient to use, but are rather CPU intensive functions. An example is interest calculation starting with finding the number of days between two dates. If this is done for each record in a batch run, the date field calculation may be responsible for around half the CPU time used by the program. Most often such routines calculate the days elapsed between an older date and today's date, in which case the results of the calculations can be stored in an array using the older date as key. Subsequent date calculations can then be replaced by much faster binary table look-ups in the array.

[Print all pages](#)

[Print page](#)

The examples are produced 100 % automatically by GiAPA.
All jobs are analyzed, and only programs with optimization potential are selected.

Example of optimization hint for file access. HTML format selected.



File Access Optimization Hint

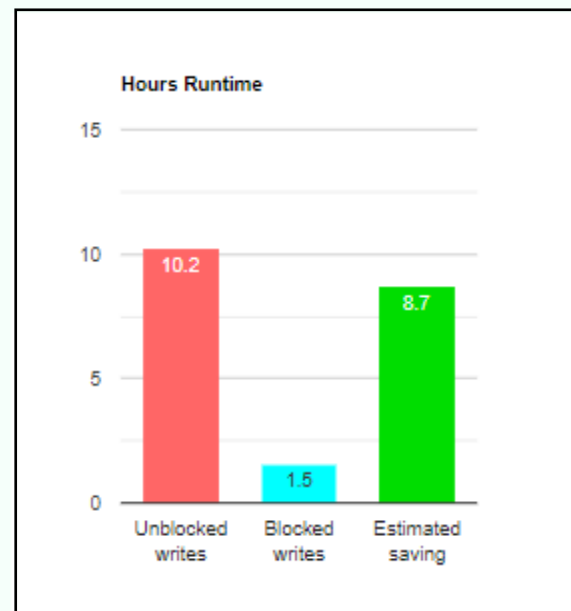
System: MAINSERV
781X22C LPAR 021

95.3 hours of data collected starting 2021-01-29 at 00:01

File accessed QTEMP/FEWXRNMP Transactions ready for main update run
Records in file 50,513,446 (Estimate based on records accessed)
GiAPA detected 1,765,955,117 unblocked writes of records found in 4,625 HotSpots

Job and user HSLAB KVKZKDV (117 jobs)
HSLAX HAHXDYM (2 jobs)
HSLIJ KVKZKDV (6 jobs)
(More job info shown by GiAPA Menu option 19, sel. 2)

Estimated saving 524 minutes run time (mainly CPU time)
Effort required Probably < 4 man-hours (test time not included)



Technical explanation

Writing records/rows one by one is inefficient. A change to use blocking would save most of the time used by these writes.

Tips on how to optimize the performance

When QDBPUT occurs as the active program in many GiAPA HotSpots it should always be considered if the much more performance efficient blocked writes could be used. If the program logic does not necessitate forcing the records to be added to the file immediately, CL statements may be used to request blocking (please refer to GiAPA Tutorial 14, slides 4, 6, 7 and 9 for more details). Data base management will in some cases not automatically use blocked writes, e.g. if access path(s) with unique keys are defined for the data. However, if user program logic assures that duplicate key values are avoided, blocking can be forced through use of CL OVRDBF statement. Blocking could cut over 80 % of the time used for writing the records.

[Print all pages](#)

[Print page](#)

The same two examples shown in "Green screen" format.

```

GiAPA (c) by
iPerformance
Program Optimization Hint
GIAPAUTILI Mbr EXAMPLES
System: MAINSERV
95.3 hours of data collected starting 2021-01-29 at 00:01
781X22C LPAR 021

Active user program RWONMW/OMENPDHPZ Calculate interest for outstanding invoices
Statement number 46900
GiAPA detected Date/time conversion or calculation found in 3907 HotSpots
Job and user names UBSTVABZY4 KVKZKDV (4 jobs)
UBSTVABZY7 KVKZKDV (4 jobs)

Hours Runtime
Curr pgm total 30.6
DateTimeProcess 16.2
Memory table 2.4
    
```

Estimated saving 85 % of DATETIME = 830 minutes
 Effort required Probably < 7 hours programmer

Tecnical explanation
 The process needed for date/time format conversions

Tips on how to optimize performance
 Date/Time conversions, and calculations on date and
 An example is interest calculation starting with findi
 batch run, the date field calculation may be responsi
 calculate the days elapsed between an older date and t
 array using the older date as key. Subsequent date cal
 array.

F2=Cmd line F3=Exit PageUp=Show previous

```

POWER720
File Access Optimization Hint
GIAPAUTILI Mbr EXAMPLES
System: MAINSERV
iPerformance
95.3 hours of data collected starting 2021-01-29 at 00:01
781X22C LPAR 021

File accessed QTEMP/FEWXRNMP Transactions ready for main update run
Records in file 50,513,446 (Estimate based on records accessed)
GiAPA detected 1,765,955,117 unblocked writes of records found in 4,625 HotSpots
Job and user names HSLAB KVKZKDV (117 jobs)
HSLAX HAHXDYM (2 jobs)
HSLIJ KVKZKDV (6 jobs)
(More job info shown by GiAPA Menu option 19, sel. 2)

Hours Runtime
Unblocked writes 10.2
Blocked writes 1.5
Estimated saving: 8.7

Estimated saving 524 minutes run time (mainly CPU time)
Effort required Probably < 4 man-hours (test time not included)

Tecnical explanation
Writing records/rows one by one is inefficient. A change to use blocking would save most of the time used by these writes.

Tips on how to optimize performance
When QDBPUT occurs as the active program in many GiAPA HotSpots it should always be considered if the much more performance
efficient blocked writes could be used. If the program logic does not necessitate forcing the records to be added to the file
immediatly, CL statements may be used to request blocking (please refer to GiAPA Tutorial 14, slides 4, 6, 7 and 9 for more
details). Data base management will in some cases not automatically use blocked writes, e.g. if access path(s) with unique keys are
defined for the data. However, if user program logic assures that duplicate key values are avoided, blocking can be forced through
use of CL OVRDBF statement. Blocking could cut over 80 % of the time used for writing the records.

F2=Cmd line F3=Exit PageUp=Show previous PageDown or Enter=Show Next
    
```

DATA COLLECTION AND ANALYSIS

IBM PERFORMANCE EXPLORER

EXPORT AND IMPORT GIAPA DATA

- 11 Submit performance data collection
- 12 HotSpot watch of one selected job
- 13 End performance data collection

- 31 Start PEX statistics data collection
- 32 End PEX statistics data collection
- 33 List call stack based on PEX data

- 71 Export GiAPA raw performance data
- 72 Export GiAPA analysis results
- 73 Import GiAPA raw data or results

By analyzing all jobs GiAPA may report optimization potential that cannot be used, e.g. when found within purchased standard software that the supplier does not want to modify.

In these cases, please use GiAPA Menu option 75 to specify the names of programs and files to be exempt from further optimization hint reporting.

INSTALLATION PARAMETERS

- 74 Define loop trap exceptions
- 75 HotSpot and Optim.Hint exceptions
- 76 Maintain color palettes for graphics
- 77 Define jobnames for special analysis
- 78 Installation parameters

HOUSEKEEPING

- 81 Manage unexpanded pfr.data members
- 82 Manage expanded data members
- 83 Delete Performance Explorer data
- 84 Delete trace job data
- 85 Delete file check data
- 87 Delete RUNQRY/WRKQRY tracking data
- 89 Check if authority OK for pfr.coll.
- 99 Display GiAPA Command Menu

- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26 User defined graphics
- 27 Start GiapaNavigator (Charts + GUI)
- 28 Work with created graphics data

- 65 End RUNQRY and WRKQRY tracking
- 66 List RUNQRY and WRKQRY usage

F2=Cmd.Line F3=Exit

Licence code type: G

Option or menu: █

Data library: GIAPALIB

GiAPA (c) by iPerformance Maintain Exceptions for HotSpot Analysis 21-03-18
 and for Hints Reporting (Menu option 20) 17:36:56

- 1) Jobs having names or user names entered below will not trigger HotSpots.
- 2) Programs and files entered below are not listed in 'Optimization Hints'

Maintenance of name list: Change by overtyping, delete by blanking,
 add by entering new name in empty space