



# Global iSeries Application Performance Analyzer

Product Sheet  
May 2009

Even most iSeries DP professionals consider application performance optimization a complex task, normally requiring expensive external experts. However, **GiAPA's** automatic data collection and analysis makes it easy for the average operator, programmer or systems analyst to locate and diagnose the vast majority of performance inefficiencies.

**GiAPA** is to our knowledge the first and only software product created with the sole purpose of locating and diagnosing performance inefficiencies in applications running under i5/OS. **GiAPA** will assist you in using your existing computer resources in an optimal way, so your hardware does not have to grow faster than the number of users or applications.

**Using less than 0.1 per cent of one CPU, GiAPA** collects performance data for all active jobs and tasks every 15 seconds, automatically storing additional details down to thread level for the jobs using the most resources.

**GiAPA** allows you to collect comprehensive performance data continuously at merely no costs. This means that when a performance problem has occurred (e.g. a user complains about bad response times during the morning), you can pinpoint the cause within the available **GiAPA** data.

## GiAPA is designed to tell you

1. which of your applications have performance problems,
2. what the problems are,
3. within which program(s), often down to source statement line number, and
4. how the application can be improved to run efficiently.

## GiAPA is not designed to tell you

5. what additional hardware to purchase in order to hide the performance problems.

## GiAPA is used for much more than speeding up applications

Since **GiAPA** keeps details on the movements of all jobs every 15 seconds, it is commonly used by operations to follow up on the every day load on the server (who did what and when), or to supply management trends on resource usage by application, week, or .....

## How does GiAPA Work?

### 1. Data Collection.

**GiAPA's** data collection retrieves performance details for all jobs and tasks every 15 seconds. Additional call stack and file usage data is collected automatically for any job causing a "HotSpot", i.e. exceeding user-defined limits for CPU percentage or Disk-I/Os within a 15 seconds interval. Data collection consumes typically somewhat less than 0.1 % CPU.

### 2. Data Expansion and Analysis.

The collected performance data stored in compressed binary format is expanded and analyzed. Jobs suffering from or potentially causing performance problems are kept individually by **GiAPA** for further analysis – other jobs or tasks are kept summarized by type and interval.

### 3. GiAPA Reports.

A large number of exception and detailed reports, including pie, bar, line, and column charts, can be requested interactively, often telling down to source statement number where to correct which performance problems, or telling in a graphical format where resources are used.

**See more specifications, and examples of GiAPA reports on [www.giapa.com](http://www.giapa.com)**

### Capacity Planning based on Peeks

Although not designed for capacity planning, the data collected by **GiAPA** can tell resources used per interval within any selection criteria available. If e.g. job or user name can be used to identify a given application, then **GiAPA** can answer a question like

“We have currently implemented the new software for 3 of our warehouses. If we add 12 warehouses more, how much additional CPW would be needed?”

This is close to impossible to answer using traditional tools, since peeks and not total usage are needed.

### Trace Job Analysis

If jobs are running very slowly without really showing any excessive use resources, the **GiAPA** standard reports will not catch the problems. Such delays, often caused by locks, can be very difficult to locate.

**GiAPA's** solution is to analyze Trace Job output file records, producing exception reports that in most such cases very clearly will show the cause of the delays.

### GiAPA Loop Surveillance

It should not happen, but it is not uncommon to hear about jobs looping. On faster machines with several CPUs such loops may continue unnoticed for hours.

Since **GiAPA** is processing all kind of performance data for all active jobs every 15 seconds anyway, such loops can trigger sending of a message to QSYSOPR.

### Which program is causing a delay?

Call stack information is automatically collected for any job using many resources within a 15 seconds data collection interval. Sufficient data is therefore available to allow statistical analysis of all heavy jobs, telling which thread, program and statement number is using how many per cent of the total run time.

### Installation parameters

**GiAPA's** flexibility is user-controlled thru parameters, shipped with a default value, but adjustable according to the speed of the CPU, etc. They are used to define the selection limits for the **GiAPA** exception reports.

### Logical Analysis of File Usage

**GiAPA** may be the only tool collecting file usage information down to individual record number. You will be surprised to learn the saving potential that can be determined from analyzing this data.

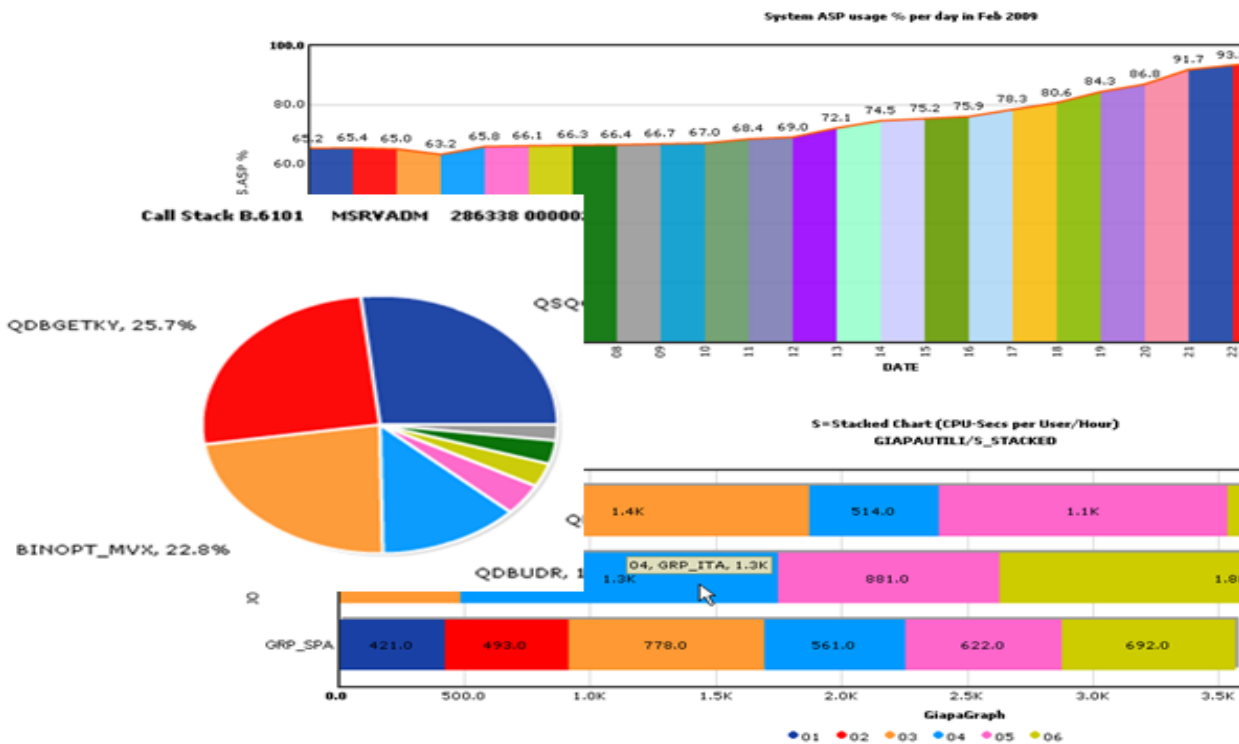
### Obtaining GiAPA Software and Documentation

The software, updates, the User Manual, and “Terms & Conditions” are available from [www.giapa.com](http://www.giapa.com). Get download passwords from your **GiAPA** contact person!

### Security – Commands and Tools used

For retrieving performance data **GiAPA** only uses IBM standard APIs and commands. Customer programs or data base files etc. or operating system parameters are obviously never modified by **GiAPA** in any way.

## Examples of Charts produced by GiAPA Graphics



**iPerformance ApS**



Skrænten 6B, DK-4220 Korsør, Denmark  
Tel. +45 5835 3305 [www.iPerformance.dk](http://www.iPerformance.dk)