

## Frequently Asked Questions

# ActiveBase Performance

Performance improvement and control

### What exactly do I need to install?

In order to run AB\*Performance, you need to download and open a tar file - the AB\*Performance server software. It can be installed on a dedicated server or on the database server, or on one of the application servers with Unix HP/Solaris, AIX, Linux or Windows OS. For testing purposes, even a desktop is enough (though we would not recommend it for a production environment).

### How large is the download and what does it include?

The download for the AB\*Performance server software ~35MB, dependant on your OS version, and includes the installation as well as documentation and a management console installation.

### How long will it take to install and configure AB\*Performance on a single database?

Installation only takes about an hour. You will first install the server software (opening a tar), followed by a PC – administrator management consol.

Predefined rules will be available immediately, and you will be able to define custom rules

### Do I need to change my Applications or Database?

No! AB\*Performance is completely transparent to both applications and databases. NO DATABASE CONFIGURATION CHANGES REQUIRED!.

The only configuration change required, is either:

1. Add a new entry to the Oracle client centralized routing flat file, (tnsnames.ora or oranames) with ActiveBase server host and listener port number. This file is centrally managed for all clients, thus a single and simple configuration change immediately affects all clients to be routed via the ActiveBase server. For testing purposes, a tnsnames.ora copy can be saved on a local directory on your client.

OR...

2. Change the database listener port to another port (hidden) and configure ActiveBase listener port to listen to the primary port (e.g. Oracle port 1521). ActiveBase includes specific functionality to support this type of configuration by enabling selective bypass to specific clients (clients are identified by an include/exclude list of program/host name and OSusers). This bypass enables all clients to connect to ActiveBase listener port, where specific clients will be routed with ActiveBase policies applied, and other applications will be routed directly to the hidden database



listener, thus bypassing ActiveBase policies (e.g. ETL processes that do not require auditing or security policies).

### What permissions do I need in order to install AB\*Performance?

You create user active, and open AB\*Performance tar file. It also comes with its OWN JVM installation, inside the installation tar. You 'start' AB\*Performance using running ./start from the installation directory. Edit the setPath file for specifying your specific installation directory.

After installation, you define the Oracle databases that run on the server. When introducing a database in AB\*Performance, you need to enter oracle username and password. This Oracle user needs to have a plan\_table and 'become user' privileges to the main schema.

### What are the resources AB\*Performance consumes as it is installed on the database server?

Overhead is negligible - about 1% CPU load, where propagation delay is about 150 microsecond per SQL statement. Keep note that AB\*Performance switch capability enables to define the specific modules, clients and IP's that will go through the rule processing (and overhead – 'use-rules' routing action), and which will automatically bypass AB\*Performance rules and connect directly to the Oracle listener ('direct' routing action).

A rule tree has been included, that enables ActiveBase to bypass all fast and efficient SQL statements with no delay, while only the long and inefficient SQL statement are caught and manipulated by the rules – accelerating them by x10-1000 times.

### What Databases does AB\*Performance support?

AB\*Performance is currently available for Oracle 8 – 11g running on UNIX (HP, Sun or IBM AIX), Linux or Windows.

### How does AB\*Performance identifies and manages Parallel Query Server sessions?

AB\*Performance can add parallel hint that reduces or increases the parallel degree of a request. In addition, it can disable parallel execution all together by adding 'alter session disable parallel' command on incoming sessions.

### How do you upgrade AB\*Performance?

Simply stop the server process (using ./stop command within the activeknowledge directory), download the tar file into active directory and untar it. Upgrading does not delete your existing installation configurations. You also will need to uninstall ((control panel -> install/uninstall programs->ActiveBase),) and reinstall the Windows client to be able to connect and administer the new version.

### What is maximum number of rules that can be implemented without effecting propagation?

No real limit or affect. AB\*Performance testing is done with up to 10000 rules.



Nevertheless based on our experience large installations do not have more than 30 - 50 active rules (although each rule fixes many different SQL statements – using ‘search and replace’ rewrite and regular expression tag place holders, partial text matching, partial execution plan matching etc.- resulting in a small BUT VERY EFFECTIVE number of rules in large production implementations. Note that in massive on-line-transaction-processing (OLTP) application, we apply rules only on selected modules as well as use only regular expression matchers and rewrite actions for best speed.

[How is ActiveBase different from native DBMS performance features and tools?](#)

Native DBMS performance tuning tools typically provide SQL optimization suggestions, with no capability of implementing this suggestion (requires manually changing the application source code, when applicable). This is not feasible when the customer has no control over the application source code (in packaged OLTP applications), and in large DW and BI environments - where reports and ad-hoc queries are created and changed daily. Oracle's DBMS Outline feature does not support applications using dynamic SQL requests and are impossible to administer and manage on a large scale. ActiveBase broad rule capabilities can fix these occurrences immediately and with no changes whatsoever.

[Why is AB\\*Performance blocking action better and safer then blocking provided by other solutions?](#)

Only AB\*Performance enables to block a specific SQL requests in any application (2, 3 or n-tier applications) without touching application code, while returning a customized notification to the user (multi-language supported).

Application connections are not torn and sessions are not killed. Only the specific request is blocked. Other requests from different users using the same connection continue with no application abstraction whatsoever.

[Can AB\\*Performance rewrite/add-hint or block SQL requests that will perform a full scan on large tables?](#)

Yes, AB\*Performance can identify the execution plan of the SQL statement, the cost and the from clause objects and apply rewrites, hints and statement blocks accordingly.

[How AB\\*Performance knows what is the database execution of a statement – even before the statement is executed in the database?](#)

AB\*Performance checks the cost and execution plan of the SQL statement by using a JDBC connection from ActiveBase server to the database, and executing 'get explain plan for SQL'. It parses the cost and the execution plan steps returned by the database and tests these results with the execution plan and cost based matching rules.



Note: if no rule includes execution plan or cost identification criteria, NO 'GET EXECUTION PLAN' IS DONE.

### How can we prevent AB\*Performance from being a single point of failure?

AB\*Performance uses several built-in mechanisms to ensure that it will never be a single point of failure. In addition, Oracle self fail-over mechanism (Failover and TAF – Transparent Application Failover) guarantees that in extreme circumstances all clients will automatically and immediately reconnect directly to the database server, completely bypassing AB\*Performance server. AB\*Performance has also a bypass routing option, that when activated, redirects all clients directly to the databases. When required, AB\*Performance can also be installed on high-availability cluster.