

Performance Top 10 Rules



Good applications have at least one thing in common – they are *really* fast. Beyond built-in OutSystems® Platform optimizations keep these best practices in mind.

1 Index your entities

Creating indexes for the most used entity attributes will significantly improve the performance of select queries at the cost of slight overhead in insert and update operations.

2 Setup database maintenance plans

Maintenance plans with reorganization of indexes and statistics update allow the database engine to optimize queries and take advantage of the existing entity indexes. Don't forget to involve the resident DBA.

3 Use Service Center reports

Use Service Center reports as the starting point for all application tuning efforts. These reports provide a full view of the slowest operations; are divided by application screens, web services, timers, queries and extensions.

4 Don't join over linked server

Cross server joins are very inefficient because the table in the linked server is completely loaded to the local DB server and then the join is performed. Cross server joins may be acceptable if the tables are small and unavoidable but as a rule cross server joins should be avoided as much as possible.

5 Configure web server memory settings

Apply the webserver tuning settings to optimize memory usage and application server performance according to the checklist best practices. Using .NET stack, process recycling affects performance and should be minimized, while in the Java stack, the virtual machine memory settings should be fine-tuned to improve availability.

6 Isolate large text and binary data

Avoid the use of 2000+ characters in a text field; data fields greater than 2000 bytes are converted into a Text data type which in turn are much slower to process and decrease performance overall. As a rule, isolate binary and/or large text fields in separate entities and only retrieve them when they are strictly necessary.

7 Avoid long running timers and batch jobs

Timers and batch jobs should be built having partial processing capability and auto-timeout mechanisms. These features will prevent run-away jobs, endless data reprocessing and data inconsistencies.

8 Simplify screen preparations

Screen preparation processing affects the user experience and therefore should be simple and fast to execute. This is especially important for web screens having a high access volume like home pages

9 Avoid using preparation data in screen actions

Don't use preparation data in screen actions because it increases the network traffic between the server and the browser. This in turn increases network usage and degrade server request processing performance.

10 Minimize the number of fields fetched from the database

Use specific structures with the necessary attributes, instead of returning the whole database entity as I/O parameters of actions. This allows the platform to optimize the queries, improving database performance.