



JTC: THE UNIQUE JAVA CACHE FOR TABLES

JTC is a unique in-memory Java cache. JTC provides the world's fastest read and write Java access to data in tabular form. JTC extends the concept of the Key-Value Store to the use of tables - independent of the complexity and limitations of an in-memory database.

JTC provides tables with a flexible structure for your applications.

Give your business processes a head start: JTC Tables can be structured flexibly - and thus offer great advantages for your daily business processes. JTC is distinctly different from all other cache solutions available on the market.

JTC searches faster than any in-memory database.

Take advantage of the world's fastest Java access: JTC's innovative technology provides your applications with tables for high-performance Java read access. You can also define derived index tables with a different sort sequence. This enables JTC to process data faster than any other in-memory database - e.g. up to 40 times faster than the in-memory database H2.

JTC saves a lot of programming effort and is easy to use.

Enjoy the convenience: JTC provides all query logics for finding and returning searched values or value ranges from tables. The intuitive API, which specializes in high-performance access, makes data search in tables simple and convenient.

JTC can be updated by your applications.

Load tables from different databases into JTC or create the tables with the required table structures directly in JTC. Write table data directly from your applications into the Java Table Cache using the configurable JTC write-API. Thus, your data is always available for access at maximum speed.

JTC is a bi-temporal in-memory store.

Define different versions of your tables and their validity period. JTC automatically searches for the valid table version for the timeframe that you have specified, and returns the data that is valid in this timeframe. Snapshots ensure data access that is consistent over time and across tables.



BETTER ORGANIZED INFORMATION.

BOI Software Entwicklung und Vertrieb GmbH

40 years of success and innovation.

Your expert for data management.

© Copyright BOI Software GmbH, Austria.

BOI LINZ

Spazgasse 4
4040 Linz, Austria

Phone: +43 (0) 732 736423 - 0

E-Mail: office@boi.at

BOI GRAZ

Friedrichgasse 30/1
8010 Graz, Austria

Fax: +43 (0) 732 736423 - 2

<https://www.boi.at>

JTC: Worldwide unique multi-client Java cache

Adapt JTC to the organization of your company: JTC is multi-client capable. It can write and read multiple independent subsystems, so-called data rooms, in the Java Table Cache. These can be perfectly configured for your organizational structure: Disjoint units that have neither users nor data in common are as well possible as instances with overlap (e.g. shared users, databases, authorizations, etc.). This results in a wide range of possible applications: from the implementation of different specialist areas to the realisation of different environments, e.g. development, testing, production, etc.

JTC guarantees absolute security for your data.

The entire dataset of the Java Table Cache can be saved in a database of your choice - automatically and in time intervals freely configurable by you. In addition, JTC permanently keeps a backup copy of all data on the hard disk. A warm start is thus possible at any time.

CONCLUSION: JTC is the convenient complete solution for the implementation of a table-based in-memory Java Table Cache. JTC convinces with its innovative and user-friendly technology when the limitations of the key-value store are hindering and an in-memory database is too complex.

JTC IN COMPARISON

Advertising is silver - facts are gold: We sent JTC into the race against Key-Value Store and in-memory database.

Here is a summary of the most important values. For a detailed table with all test results, please contact us at: sales@boi.at

| Sector | Key-Value Store | JTC | In-memory database |
|----------------------------|--|---|--|
| Local access | 30.000.000 /sec. (Integer-GET) | 3.000.000 /sec. (table access) | 200.000 /sec. (SQL queries) |
| Distributed access | ~9.000 /sec. (Integer-GET) | 3.000.000 /sec. (table access) | no distribution planned |
| Read for composite values | ✗ Access to total values only | 3.000.000 /sec. | 250.000 /sec. |
| Write for composite values | | 8.000.000 Updates/sec. | 25.000 Updates/sec. |
| Multi-client | ✗ To be implemented in the application code | Automatically managed multidimensional organizational units | ✗ To be implemented in the application code |
| Temporal data | ✗ To be implemented in the application code | Automated temporal data storage based on tables & records | To be implemented in temporal SQL or in application code |