Transbase®

PRODUCT SHEET

Transbase®
The SQL Core System

Transbase® is a relational database system which conforms to the current SQL standards. It is especially suitable for embedding in applications and web services due to its product features and licensing flexibility. Transbase® can be extended for special functions with the three currently available options Transbase® Hypercube, Transbase® Fulltext and Transbase® D.

Transbase® Product Features

Transbase® supports a range of different **types of table** for a variety of purposes:

- Standard tables with clustered data storage as primary B+ trees,
- Tables without index for speedier insertion e.g. log entries
- Remote tables in other Transbase® databases. For this the option Transbase® D is required.
- Tables in other databases which are attached via JDBC or OCI for read only purposes.
- External files in CSV format, which can be firmly integrated for the purpose of, for example, the building of a database.

Transbase® offers manifold **indexing techniques** for primary as well as secondary indexes in Transbase® tables:

- As standard index, B+ trees are used, for both tables (with primary keys) and for indexes on secondary keys.
- Furthermore a particularly compact bitmap index is available, which can be used with single column secondary indexes on INTEGER fields, and by means of the bitmap algorithm supports in particular the combination of many less selective fields.
- Beyond that, the option of Transbase® Hypercube offers a multi dimensional indexing, also for primary and secondary indexes. This option is particularly suited to multi dimensional data and in particular for Data Warehouse data.

- With the Transbase® Fulltext option full text indexes, for the indexing of text, are made available, by which means even large text repositories can be searched very rapidly for full text predicates combined with structural predicates.

Transbase® also supports **Views** – virtual tables, defined by means of an SQL query. These make it possible to create personalized views of the stored data. Thereby data can be both filtered as well as allowing the structural view of the data (columns) to be limited, or to be extended by the inclusion of calculated fields, or fields from other tables.

The **privileges** defined in the SQL standards – the rights to access tables, views or stored procedures are supported by Transbase® in their entirety. They govern the rights to read, insert, delete or alter data for each user. Rights are granted at table or column level, with the possibility to pass on these rights on a transitive basis. In particular, in combination with views, privileges enable a personalised access security, without the need to hold redundant data.

Advantages for Application Development

By using the product features described, application developers can make full use of all the features defined by the current valid **SQL standard (SQL2003)** or port applications from other database systems to Transbase® without problem. This offers optimal investment protection for such applications.

With Transbase® it is also possible to use specific language extensions which go beyond the SQL standard. As an example at this point the use of iterative queries created with the help of CONNECT BY constructions can be mentioned. The query language can thereby be enhanced considerably thereby simplifying the generation of queries within the application.

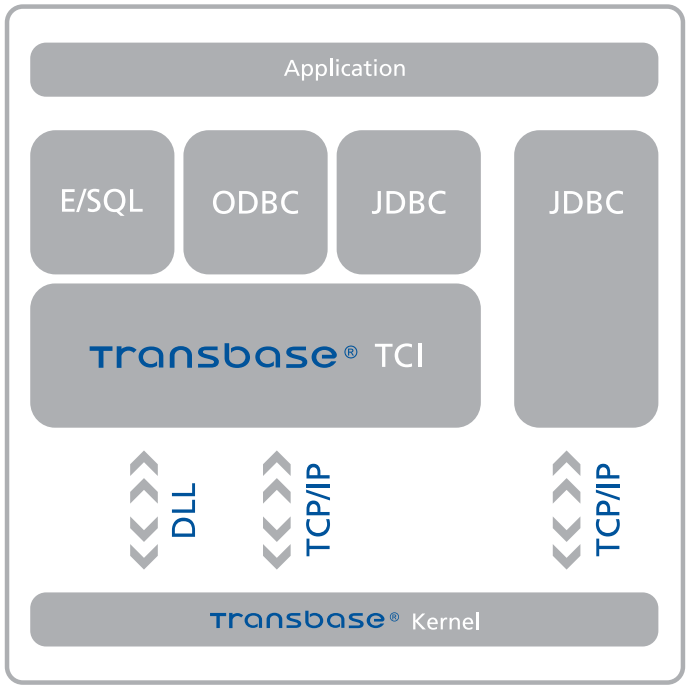
Transbase® is available with identical functionality on various **platforms**, ranging from hosts with multiple processors to single PCs, or even embedded platforms. The operating systems Windows, Linux and Unix are supported in all relevant versions. The programming interfaces in all cases are identical, so that porting to different or new platforms can be performed without problems.

The application developer can choose from a wide range of **programming interfaces**. The Transbase® Call Interface **TCI** has been developed for C and C++ programs and offers object oriented access functions to database objects. Also **ODBC** and **OLEDB** drivers are based on these, which can be used from, for example, Visual Basic applications, Delphi, or from C++ program generators. **ADO.net** also offers a direct access to Transbase® that is without the need for bringing in other drivers. The Transbase® **JDBC** driver is suited for all platform independent applications.

By means of these open, standard interfaces Transbase® is very easily integrated into the application development environment.

Illustration 1:

Interfaces



Transbase® Features

One of the significant features of Transbase® is its **ease of installation**, which means that Transbase® can be redistributed as an embedded database system along with an application without the user even being aware that the database system is in operation. The Transbase® setup can even be integrated into the application setup.

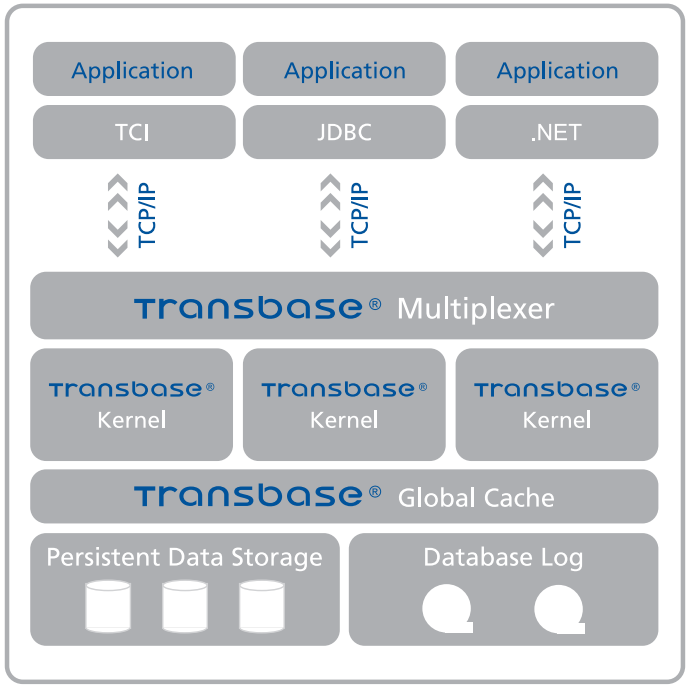
The operation of Transbase® databases requires no administration or maintenance. All **administration functions** e.g. starting the service, the implementation of changes or securing the database contents are made available via a programmatic interface and as such can be seamlessly integrated into the application.

The Transbase® **IO system** uses what is made available by the operating system and generates IO requests in parallel in order that RAID systems or databases spread across several hard disks can be used with optimal performance. In particular the IO transfers are performed directly between the disk and Transbase®, in order that the cache of the operating system is not affected. The (configurable) size of the Transbase® cache is the deciding factor for the performance; depending upon architecture and performance requirements they can be set anywhere between a few megabytes and 2 gigabytes.

The Transbase® cache is available globally as a shared memory object for all instances of Transbase® databases in main memory. It is managed by LRU (Least Recently Used). In addition each instance of Transbase® can dynamically acquire for itself local memory for such as sort operations or for interim results.

Illustration 2:

Architecture



All Transbase® operations are as a matter of course organised as **transactions** and thereby are atomic, consistent, isolated and durable (ACID). Transbase® offers two methods for ensuring the integrity of transactions: Before Images and Data Logging. In the first case, a "Before Image" is written for every page that is changed, so that in the case of an error it can be used for recovery; when a transaction commit occurs all changes it caused are committed to disk persistently. This method is best suited for mainly read operation. The second method is especially designed for large caches and for concurrent write operations, as when a transaction commit occurs, only the log of the changes must be written persistently. The changed pages can be written later as required. At the same time the log serves to provide tolerance in the event of disk failures, making flexible procedures possible for the purpose of **backing up databases** during live use.

The **query processing** itself occurs in three stages: the translation of an SQL query into an execution plan ("Operator tree"), followed by the optimizing of this plan with special consideration being given to indexes and access paths, and the resulting evaluation. The query results are calculated "on demand" but can be buffered for calculation (for scrolling cursors).

Within the framework of the optimizing, parts of the operator tree are identified which can be **processed in parallel**. During the evaluation and depending on the load threads are then generated in order to accelerate the query to the maximum. Thereby it is assured that modern multi CPU architecture is also used effectively in the event that only one query is active in the system.

Thanks to efficient algorithms and data structures the **resource requirements** of Transbase® are astoundingly small, both in terms of the hard disk and the main storage: 2 MB processor and 4 MB cache result in a total system with a very good performance. For storage on the hard disk there are various logical and physical compressions available which, in addition to sparing the use of resources, positively impact the performance. This performance characteristic is particularly important for the porting of applications to small PCs, laptops or embedded architecture.

An important feature of Transbase® is its **scalability**, which arise to the most part from the guaranteed performance behavior characteristics of B+ trees: The run times for key accesses grow only logarithmically in relation to the height of the tree. Since the B+ trees are the basis for all access paths, the end result is a perfect scalability for Transbase® databases.

Transbase® Options

Over and above the capabilities described so far, the options of Transbase® Hypercube, Transbase® Fulltext and Transbase® D offer **additional functionality**. These options can also be combined with each of the Transbase® variants (Transbase® CD, Transbase® Embedded and Transbase® Myriad).

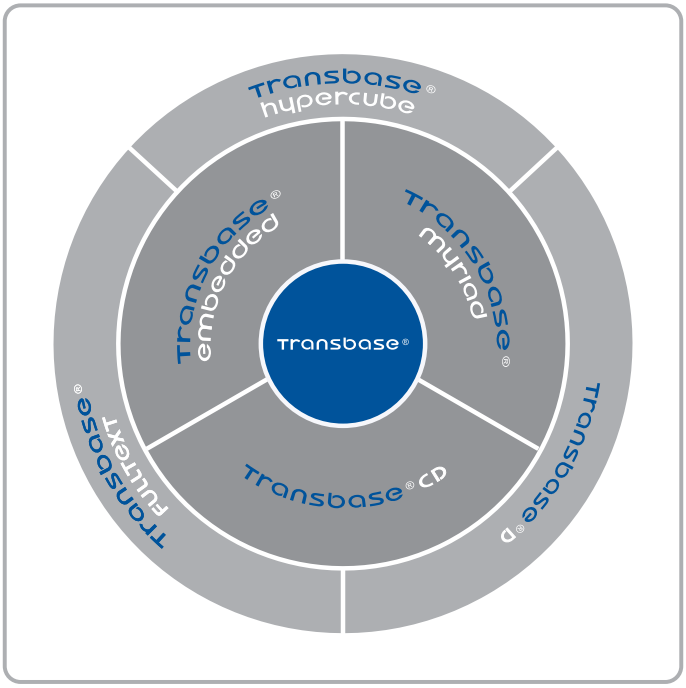


Illustration 3:

Transbase® – Variants and Options

Transbase® Hypercube offers a special index which makes multi dimensional and hierarchical indexing possible and which is completely integrated into SQL. The indexing is clustered – similar to the case with B+ trees – that is, data sets which are "close to each other" in the multi dimensional space are also "close to each other" on the hard disk and can therefore be materialized with little IO effort.

This allows all kinds of OLAP applications to be speeded up very effectively. This is particularly so in the case of large and growing amounts of data, where scalability is a major requirement. Transbase® Hypercube is responsible for maintaining the multidimensional order during change operations and thus guarantees that access times only grow logarithmically. Maintenance and support are as simple as for other Transbase® table types.

Transbase® Fulltext is a special index for the purpose of full text searches in text columns (CHAR, VARCHAR, CLOB), and is also fully integrated into SQL. This enables full text predicates to be combined with structural predicates as wished, and for them to be converted into an optimal integrated evaluation plan. The full text predicates encompass partial word searches including NORMAL expressions, and all forms of NEIGHBORHOOD searches (in the case of positional full text indexes). The algorithms for breaking up words can be influenced by flexible configuration parameters. In order to individually reduce the space requirements it is possible to define negative or positive word lists.

Transbase® D makes the processing of queries possible on a number of remote Transbase® databases.
Thus, remote tables can be addressed like local tables, completely transparently. The Transbase® optimizer distributes the queries so that in every case the maximum part queries can be sent and their results then processed. The operation of remote queries occurs thereby in parallel and in the background.

A local transaction automatically becomes a distributed transaction:
When more than a single database is involved in the write process, the transaction is automatically committed in a 2 phase protocol, thus preserving the full integrity of the transaction security.

Transbase® Licensing

The licensing of Transbase® is based on several different criteria. It is based on target platform, license type and the transferability of the runtime license.

For the various target platforms there are three classes of license: Class I covers UNIX systems. Class II covers Linux and P.C. based UNIX systems, as well as Windows servers. Class III applies to Windows workstations which are not networked.

A difference is made between Transbase® Run time Licenses and Transbase® Development Licenses. A Runtime License is necessary for the use of Transbase® applications, whereas the Development License is required for the development of the application.

Transbase® Development Licenses are available for single, four or an unlimited number of developers. For each target platform the respective Development License has to be obtained. Transbase® Development Licenses are not transferable.

Runtime Licenses for the operation of servers are licensed on the basis of the maximum number of connections and according to the number of CPUs. A growing number of connections results in degressive license fees.

Transbase® Runtime Licenses can be distributed in combination with the developed application. In this case the Transbase® Runtime License follows on the basis of the total number of static users. A static user is in this case defined as a workstation, from which access to a Transbase® database exists, regardless of whether as a stand alone workstation, or via a local network or the internet. Single and multiple workstation users are simply added. An increasing number of static users results in degressive license fees.

Contact

Transaction Software GmbH
Willy-Brandt-Allee 2
81829 München, Germany

Tel.: +49 89 / 627 09 - 0
Fax: +49 89 / 627 09 - 11

info@transaction.de
www.transaction.de
www.transbase.de