



“Enterprise-Class” Means More Than “Used in Enterprises”

By: Andy Astor
CEO, EnterpriseDB Corporation

This magazine, of course, is called Enterprise Open Source Journal, and it generally covers the benefits, challenges and trends associated with using open source software in enterprises. Not all software deployed in an enterprise, however, deserves the “enterprise-class” appellation. That term is properly reserved to describe software that may be safely relied upon for business-critical uses, software with particular qualities that themselves define the term “enterprise-class”.

In determining whether software is truly enterprise-class, open source software and closed source or proprietary software are judged by the same criteria. For example, BEA WebLogic either is enterprise-class or it isn't, JBoss Application Server either is enterprise-class or it isn't, and those determinations are made by asking the same questions. On the other hand, the set of criteria for, e.g., operating systems, is different from the set that applies to, e.g., CRM applications.

Among enterprise software, relational databases make particularly appealing open source deployments. This is largely because the most popular proprietary alternatives are stratospherically over-priced, especially for large deployments, and they are unnecessarily complex to deploy, maintain and run. In addition, the best open source databases have stable and mature code bases and, in the case of truly enterprise-class products, offer easy migration paths from existing proprietary installations.

Enterprises that choose to adopt open source databases for business-critical applications should thoroughly investigate whether the choices under consideration are truly enterprise-class. In general, every enterprise-class database shares certain characteristics with the others: Functionality, Reliability, Scalability and Compatibility.

Enterprise-Class Databases are fully *Functional*

The first and most obvious required functionality is the ability to **store and retrieve data**. Support is required for all significant data types, including numbers, characters, timestamps, money, and possibly large text and binary objects.

The second required functionality is the ability to **execute programs against the database**. In an ideal world, the database is a vital but hidden piece of the IT infrastructure that end users don't know or care exists. Client applications and reporting tools are the user's interface, and they in turn seamlessly access the underlying data stores. The corollary is that standard tasks must be available to all applications by defining them in the database as stored procedures and functions.

In today's multi-tiered processing environment, it is important to correctly partition the work, so the work is being done in the environment that is best suited for the task. For

example, data-intensive work would be done logically close to the database and user-intensive work logically close to the user. Data movement is minimized by performing aggregation of information as close to the data source as possible. Most high-end databases have procedural languages that will compile and execute within the database.

It is also critical to provide the ability to **define and enforce business rules within the database**. Defining business rules in the application is not enough. Not all updates will come through front-end applications. An enterprise database will allow the definition and enforcement of:

- Referential integrity – an order cannot be added without a valid customer; a customer cannot be deleted that has open orders.
- Constraints – customer id cannot be null; taxable indicator must be Y/N.
- Complex rules – defined by triggers to enforce cross-attribute dependencies.
- Business Functions – defined by stored functions and procedures can do common validation and business processing.

These rules are defined in the database and are enforced across all applications that affect the database.

A database must **define a transaction as the basic unit of update**. Few business events are so simple that they result in a single database update. A purchase, for example, at the very least will result in the creation of an *order* row to identify purchaser, payment and shipping information as well as one or more *order item* rows to identify what was purchased. A transaction will be considered complete if **all** of the rows are stored in the database. The industry has called this the ACID model, which stands for Atomicity, Consistency, Isolation and Durability.

- Atomicity – all of the updates of a transaction are committed together. If one fails, they all fail.
- Consistency – only valid data is written to the database. The business rules discussed above are enforced.
- Isolation – prohibits each of several simultaneous transactions from impacting the others' executions. The data created or modified by one transaction is not visible to any other until it is committed.
- Durability – a transaction, once it is committed to the database, is not lost. This usually refers to the ability to preserve all transactions occurring prior to a hardware or software failure.

Finally, an enterprise-class database must offer native, mature support for all mainstream enterprise RDBMS features, such as stored procedures, and it must feature robust implementations of important database technologies, such as Multi-Version Concurrency Control (MVCC).

Enterprise-Class Databases are *Reliable*

Obviously, the database must be running to serve data to the enterprise. There are three main factors in reliability:

- Stability – the database must not crash in the hardware and software environment in which it runs.

- Recoverability – in the event of a hardware or software failure, it must be possible to restore the database to its pre-failure state.
- Failover – in the event of a hardware or software failure, a standby database must be able to assume the database duties without loss of transactions. This may be an option that is deferred in an initial or non-mission-critical application, but the architecture of the database must not prevent its future implementation.

In addition, an enterprise-class database will predictably exhibit behavior expected by enterprise database developers, including extensive use of error messages and warnings in the case of exceptions.

The final reliability criterion is that a commercial-quality, always-available, global support organization is available to guide the enterprise through the installation, deployment and execution of the database.

Enterprise-Class Databases are *Scalable*

An enterprise-class database must not have any limits on its storage capacity or processing volume. Quality of service and response levels must be maintained as the load on the database increases. Spikes in activity cannot result in severe degradation of performance. OLTP applications, for example, cannot be deployed on a database that is not enterprise-class from a scalability perspective.

Enterprise-Class Databases are *Compatible*

An enterprise-class database must adhere to standards and must not require proprietary coding. Too often, an application with proprietary coding must ultimately be ripped out and replaced at great expense. A large community of developers and administrators can build and support applications that are standards-compatible. Commercial applications are more likely to be available for standards-compliant databases, as third-party database application developers prefer to address large markets.

That said, certain proprietary RDBMS systems, notably Oracle, are sufficiently ubiquitous that the ability to execute applications written for them can fairly be considered a necessary criterion for enterprise-class databases. An enterprise-class database should feel comfortable and familiar to many enterprise developers, including the huge community of Oracle developers.

Conclusion

Organizations considering the adoption of an open source RDBMS for business-critical applications are well-advised to choose a truly enterprise-class product. This will be a fully functional, ACID-compliant database that will perform reliably, scale to meet the organization's high-volume needs and run the organization's legacy applications and widely available commercial applications without modification.

A good starting point for the investigation would be the EnterpriseDB 2005 RDBMS that is available from the EnterpriseDB Corporation. EnterpriseDB Corporation significantly enhanced PostgreSQL, the world's most advanced open source database, to create an

enterprise-class RDBMS that runs Oracle applications without modification. More information can be found on the company's website, www.enterprisedb.com.