Objectivity/DB: Solving The Most Demanding Data Management Challenges

Background
Data management challenges are more complicated than ever before, and today’s software innovators are increasingly looking for industrial-strength products that allow them to build systems to manage massive volumes of complex data.

From the world’s largest databases and real-time, distributed, multi-source data challenges to complex inter-related data fusion systems with demanding performance challenges, Objectivity/DB is becoming the standard for building data fusion applications in Government and Complex Manufacturing.

Systems based on Objectivity/DB are able to meet and exceed high performance specifications that require systems to ingest, fuse, store, correlate and easily navigate immense volumes of data at extremely high ingest rates.

Business Rationale
Objectivity/DB’s distributed architecture saves organizations thousands to hundreds of thousands of dollars by eliminating costly equipment purchases and upgrades. The technology also enables real-time information processing, so analysts can make decisions based on What Is rather than What Was.

Security and Intelligence
In today’s security landscape, government agencies have more complex data to analyze from more sources than ever before. Objectivity/DB meets these challenges head on. The technology is already at work in applications that are central to government agencies involved in security and intelligence initiatives, including:

- Security analysis & cryptanalysis
- Social network analysis
- Knowledge and case management
- Mapping terrorist networks
- Geospatial data fusion
- COMINT, ELINT, HUMINT and SIGINT

The Technology
Objectivity/DB is a pure object-oriented database that is heterogeneously compatible across multiple hardware platforms, operating systems, and languages. Current language support includes Java, C++, C#, Python, and Small-Talk. An application written in any of these languages can persist objects into a database and retrieve objects written in any of these languages. Write objects with Java and read with C++. Modify using C++ and read again using Python. Modify with Python and read again with Java.

No O-R Mapping Layer: Objectivity/DB does not use an Object-Relational mapping layer to store objects in a relational database, greatly reducing the costs of development and maintenance.

Objectivity/DB databases and applications can be embedded in devices, installed on single machines, or widely distributed across thousands of machines.

ACID Transactions: Objectivity/DB provides completely support for ACID transactions.

Replication: Objectivity/DB databases can be replicated to multiple locations so that applications can have local copies of relevant data. All of the necessary controls (locking and quorum negotiation) are managed by the Objectivity/DB software.
Objectivity/DB: Solving The Most Demanding Data Management Challenges

**No Database Server:** Objectivity/DB does not rely on a database server like many relational databases. Objectivity/DB consists of a small application library linked into the application, a lock-server process and a page-server process. The lock-server and the page-server are lightweight applications that manage locks and data pages for all of the Objectivity/DB applications accessing a federated database.

**Schema Evolution:** Objectivity/DB provides complete support for Schema Evolution, allowing schema definitions to evolve over time to handle new requirements without jeopardizing existing data. The data shapes are evolved as needed to reflect the schema shapes.

**Object Clustering:** Objectivity/DB allows you to place related objects of different types close to each other on disk, often on the same disk page. This feature can greatly accelerate application performance.

**Scalable Collections:** Objectivity/DB provides a broad range of scalable collections to allow data indexing in several different ways, including Sets, Lists, and Maps.

**Object Relationships:** Objectivity/DB relationships alleviate the need to perform joins to access related data by allowing the creation of persisted “pointers” between objects that applications can then follow to retrieve related objects. This capability is considerably faster than using SQL joins.

**Fully Distributed:** Objectivity/DB supports many data models. Organizations can distribute the applications and leave the data on a centralized server, or distribute the data across hundreds or thousands of computers and support centralized applications, or distribute the applications and the data. Developers can also use replication to move copies of data closer to their point-of-use.

**Zero Administration:** Objectivity/DB has a Zero Administration philosophy - there are very few operations that require administrative action. Most activities take place when the system is initially installed and deployed. After installation, standard system administration is usually all that is needed.