

Top 10 Reasons Friends Don't Let Friends Use Flat Files

10 Reasons in a Nutshell

1. Multiplatform data portability, indexing, and retrieval
2. Ensuring data consistency and avoiding corruption
3. Distributed application data
4. Automated, assured synchronized data
5. Concurrent writes for modern applications
6. Built-in NoSQL and SQL support
7. Reporting
8. Automated defragmentation
9. Built-in enterprise security
10. Automated disaster recovery

Need More Reasons?

Visit www.actian.com/zen for product, service, and solution information, papers, success stories, developer news and much more.

Application developers face ever-shrinking design, test, and deployment cycles, particularly in the age of agile development and delivery. And performance is everything, particularly with mobile applications and resource-constrained IoT applications. Also, application data is mushrooming as developers strive to break away from the competition with decision support, personalization, and multichannel UX to maintain state and context across time, platforms, and user behaviors. These trends are at odds with having to build out basic data management functionality for flat file systems enmeshed in their operating environments so that they can be called through various APIs or from CLIs.

Alternatively, an external database can be used, but this often puts a heavy tax on financial, compute, IT, and support resources. With Actian Zen, application developers, product managers, business analysts, and others within OEMs, ISVs and enterprises can embed a complete data management solution directly into their applications at a lower TCO while gaining a powerful, secure, and scalable engine. By moving to the Actian Zen database family, developers can trade the drawbacks of flat files for the following advantages.

1. Multiplatform data portability, indexing, and retrieval

The ease of data portability and usage across file systems and APIs is challenged by a multitude of file systems, such as NTFS (Windows), ext3 (Linux), APFS (MacOS) or NFS for file sharing. Important features like block journaling, data checksums, file system encryption, and data deduplication may be natively supported only partially or not at all. Basic data management, including indexing and selective data retrieval, generally must be hand-coded, which varies from developer to developer. Actian Zen's key-value store design provides portability, indexing, fast and efficient storage of data, and retrieval through API calls from several standard programming interfaces, while avoiding file compatibility problems.

2. Ensuring data consistency and avoiding corruption

To ensure flat file data consistency and avoid corruption, developers must code create, read, update, and delete (CRUD) logic to store or retrieve data with care, and coding may need to vary across APIs and file systems. If concurrent reads or writes must be supported across multiple threads, file systems do not ensure consistency, so a database with full ACID support is a must-have, particularly if the file system acts as a temporary cache yet has a spotty network connection. ACID capability is mandatory if the application and associated data are exposed to machine crashes — otherwise partial writes can leave data in an inconsistent state. The Actian Zen database family provides native CRUD and ACID capabilities out of the box, across virtually any platform, without any programmer effort.

3. Distributed application data

Applications are increasingly modular, adopting micro services architectures and spanning on-premise and cloud environments, multiple VM and container instances, and also mobile clients and IoT gateways and devices. Aside from managing use of local memory by the application, developers must increasingly handle synchronization, deduplication, and other needed features across various file systems. Architecting shared memory and processes to manage data through multiple APIs slows design and coding and adds ETL overhead, as well as maintenance and support nightmares. The Actian Zen database family provides server capabilities to share data with other Actian Zen databases wherever they are, on other VM instances, on remote on-premise servers, or in clients on smart phones and IoT devices, without changing data types and file formats.

4. Automated, assured synchronized data

The need for local data is increasing, even where applications are still monolithic and extend to mobile devices only through a thin client web browser interface. If not by architectural requirements then by hard experience, data must not be lost on the client during points of network disconnection. Actian Zen Edge and Zen Enterprise provide an ultra-thin client data store, automated data synchronization with the server, and built-in reconnect after loss of connectivity.

5. Concurrent writes for modern applications

File systems can handle concurrent reads and writes through file sharing, but the risk of data inconsistency and file corruption rises. For example, IoT gateways and mobile clients with several IoT edge devices will require database support for downstream, parallel control and management of applications and data. Actian Zen servers can orchestrate this activity across all of these devices.

6. Built-in NoSQL and SQL support

Most application developers prefer standard APIs to access data, while DBAs prefer SQL. In addition to full ANSI SQL support, all Actian Zen family products provide API-level access to the underlying key-value

store — no different from a file system. This method, often called NoSQL, is supported by the Btrieve 2 API for native Java, C/C++, and SWIG-generated bindings to scripting languages such as Python, Perl, and PHP.

7. Reporting

File systems do not have built-in, configurable reporting tools, an essential part of evaluating past performance or governance of the data managed by business analysts and others tasked with deriving value from the data in compliance with various regulations. Actian Zen has a built-in reporting engine that handles SQL query of any or all data within the database in parallel with its operation yet without any performance degradation or data duplication.

8. Built-in enterprise security

A significant part of an application's security vulnerability involves its data. For the developer of a packaged application or one deployed remotely, using a file system hinders management of security risks — you don't know if the deployed system is doing encryption, if files are exposed to hacking, or other details of network and file access control. Actian Zen provides access control and the ability to assign specific permissions key to protecting user privacy for GDPR, PCI, HIPAA and other regulatory requirements. Actian Zen also provides AES 192-bit encryption for data at rest and in transit.

9. Automated defragmentation

File fragmentation degrades performance over time and consumes memory resources. Because packaged application providers are generally unable to maintain deployed environments, Actian Zen Enterprise and Edge include automatic defragmentation. This built-in feature can run unattended on recently opened files with settings to work on files larger than 10MB, or with at least 15% fragmentation, or more than 5% of records out of order.

10. Automated disaster recovery

Actian Zen has complementary tools for auditing, backup, and data synchronization across multiple distributed applications.

