Progress® DataDirect®

DataDirect JDBC Drivers vs Free Drivers At A Glance

DataDirect JDBC Driver Features

1. **Unrestricted Performance**
   Driver response time and data throughput are consistent regardless of the runtime environment or data access model and stay consistent even as more application users are added.

2. **Codeless Configuration**
   Features and functionality can be added, configured, or tuned for any application without changing application code, regardless of environment or data access model.

3. **Resource Efficiency**
   Driver use of application runtime CPU and memory resources is minimized and can be tuned in the driver as needed to fit specifc runtime environment parameters or limits. Application data throughput stays high even when deployed into a virtualized environment.

4. **All-In-One Deployment**
   A single driver JAR file is required regardless of the Java environment or application requirements. Advanced functionality does not require the use of an external DLL or shared library.

5. **Streamlined Standard**
   No proprietary extensions to the JDBC specification are required by the driver for any supported data source.

Free JDBC Driver Problems

1. **Non-Scalable Architecture**
   Overall driver response time or data throughput performance is poor when deployed into different JVMs, within ORM frameworks or app servers, or as the number of application users increases.

2. **Unavailable, Inaccessible Functionality**
   New database or driver functionality is not available across all supported JVMs or hardware or within ORM frameworks or app servers because these technologies do not allow developers access to the JDBC code to enable features and functionality.

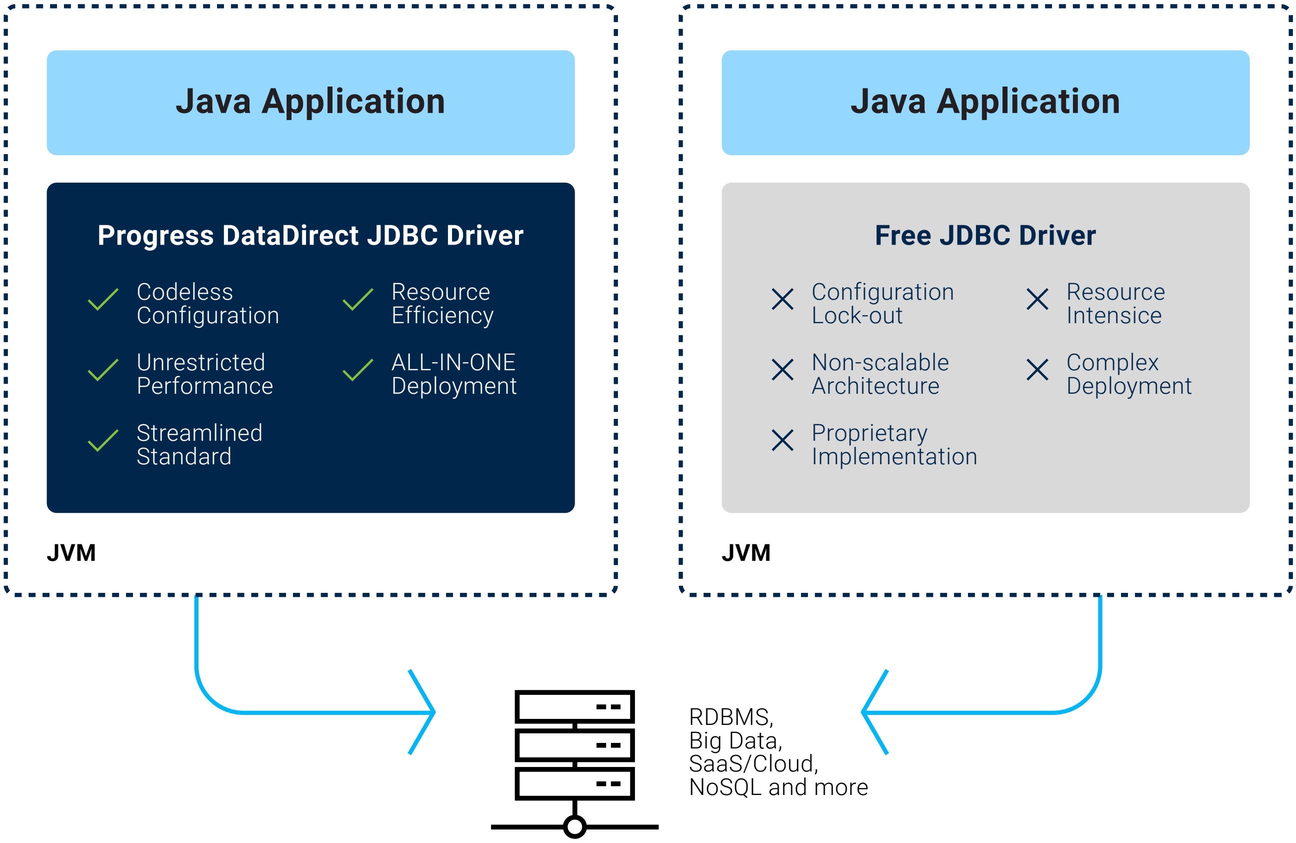
3. **Resource Intensive**
   Application resource usage targets cannot be achieved because the driver uses excessive CPU cycles and consumes vast memory resources during data access. Data throughput in virtualized environments is particularly poor. Tuning options, if available are inaccessible or limited.

4. **Complex Deployment**
   Multiple JAR files are required to support deployment across different JVMs or hardware and to access all supported versions of a particular database. Bulk data loading, security, high availability and XA features are examples of functionality that require the use of an external DLL or shared library.

5. **Proprietary Implementation**
   Each driver requires the use of proprietary code to support features such as BLOBs and CLOBs, high availability and XA. With each data source that an application must support, the amount of datasource-specifc code that must be maintained increases.

© 2018 Progress. All Rights Reserved.

DataDirect JDBC Drivers vs Free Drivers At A Glance



Java Application

Progress DataDirect JDBC Driver

- ✓ Codeless Configuration
- ✓ Unrestricted Performance
- ✓ Streamlined Standard
- ✓ Resource Efficiency
- ✓ ALL-IN-ONE Deployment

JVM

Java Application

Free JDBC Driver

- ✗ Configuration Lock-out
- ✗ Non-scalable Architecture
- ✗ Proprietary Implementation
- ✗ Resource Intensice
- ✗ Complex Deployment

JVM

RDBMS,
Big Data,
SaaS/Cloud,
NoSQL and more

DataDirect JDBC drivers offer the same client-side, single-tier, 100% Java architecture of free JDBC drivers, but address the limitations of many of the free native JDBC drivers available today.

Learn more

© 2018 Progress. All Rights Reserved.

Progress and DataDirect are trademarks or registered trademarks of Progress Software Corporation and/or one of its subsidiaries or affiliates in the U.S. and/or other countries. Any other trademarks contained herein are the property of their respective owners.

© 2018 Progress Software Corporation and/or its subsidiaries or affiliates. All rights reserved.