

■ ■ ■ white paper



# 5 Ways to Simplify Access to Adabas & Natural with Data Virtualization

5 Ways to Simplify Access to Adabas & Natural with Data Virtualization

Revised July 2015

© Rocket Software, Inc. or its affiliates 1990 – 2015. All rights reserved. Rocket and the Rocket Software logos are registered trademarks of Rocket Software, Inc. Other product and service names might be trademarks of Rocket Software or its affiliates.



# 5 Ways to Simplify Access to Adabas & Natural with Data Virtualization



Your business depends on your ability to integrate disparate data sources scattered across the IT infrastructure in a cost efficient and timely manner. For today's enterprise applications to function effectively, they need seamless access to information regardless of whether that information is stored in distributed servers or mainframe systems, the format of the information, or its geographical location. Despite the high compatibility and interoperability of application technologies today, there are unique challenges when it comes to accessing information stored within Software AG's database Adabas and its application development platform, Natural.

Despite some organizations moving data off the mainframe, Adabas has endured built on a reputation of high performance and security. Today's middleware technology is challenged to effectively resolve the fundamental differences between the way Adabas represents its data and the way the client applications



want to use that data. Regardless of the initiative driving the need for integration, whether delivering information through a web interface or bringing data into a business intelligence (BI) and analytics solution, developers confront daunting hurdles when it comes to integrating with Adabas and or Natural. This paper covers the challenges faced by organizations when integrating Adabas and Natural and how new technologies like data virtualization can eliminate technology roadblocks provide universal data access, and enable legacy applications to be easily re-used. It will help explain Rocket® Data Virtualization (DV) and how it can be a crucial tool in extending the value of your mainframe data and programs, particularly in support of new business services that require real-time, comprehensive access to data.

## Today's Market Scenario for Adabas/ Natural Interfaces

In today's market, interfaces to Adabas and Natural are very limited in scope. Most interfaces have only one focus and cannot be used for any other purpose. Companies have to equip their developers with several different interfaces to cover the challenges they face in developing applications that must interface to Adabas and Natural. Currently, there is not one product that can interface with all three layers of an Adabas/Natural application—direct to data, using the Natural business logic layer, or using the Natural screen or presentation layer. Some solutions are tied to a specific technology and provide no centralized management across other integration streams. Other connectivity options might use a gateway server as an intermediary between the distributed applications and the Adabas or Natural source, resulting in scalability issues or I/O bottlenecks. Mainframe resident adapters provide native access, but are CPU intensive because they run all of their transformation-related processing on the mainframe general purpose processor.

Without a comprehensive solution, many companies end up with a collection of products in their application development environment that offer no interoperability with one another and limit the ability to access or share data stored in Adabas.

## Challenges Accessing Adabas and Natural

Organizations needing direct, facile access to Adabas or Natural face significant challenges because there is an inherent gap between the distributed development environment and the mainframe platform. Here is a list of the top five challenges developers face when trying to access information stored in Adabas

### **1. Adabas Does Not Provide a Native Interface**

The first challenge in developing applications that must interface with Adabas and Natural is simply being able to access Adabas on the mainframe, because Adabas does not have native support for SQL access. The underlying reason for this difficulty is that Adabas uses a non-relational database structure while SQL is used with relational databases. The only way to link these two environments is to employ a data integration tool to provide reliable non-relational to relational mapping.



## **2. No Single Tool Covers All Three Integration Layers**

An Adabas/Natural application consists of three layers—the database, the application logic and, in some cases, a screen-based application layer. Most interfaces available are limited because they only access one of these layers. Developers often need to access all three layers, depending on the application, but lack a single interface to do so. Without one interface to access all three layers, organizations have to purchase separate, costly interfaces for each layer.

## **3. The Black Box Effect**

A “black box” does not give developers visibility into the device or system. A black box puts developers at a disadvantage, because it does not provide monitoring and debugging utilities necessary for efficient application development. Without debugging capabilities, errors will not be caught early—and errors that surface later in the development process are more difficult and costly to fix. In addition, the lack of monitoring capabilities makes it harder to diagnose problems once the application is in production. Without proper diagnostics, a problem with an application could lead to a service disruption.

## **4. Integration Can Require Additional Hardware**

Most Adabas/Natural interfaces on the market do not run on the same platform as Adabas, and therefore require additional hardware, such as a dedicated gateway system. Adding hardware to the infrastructure entails adding points of failure and potentially reducing application performance.

## **5. Lack of Integration with Mainframe Security**

Most interfaces do not integrate with the mainframe’s security, or they require additional components, which is an increasingly critical issue with today’s focus on data security and privacy. Organizations need to maintain a secure mainframe environment that allows appropriate access for authorized applications, without exposing mainframe systems and data to unauthorized and possibly malicious entities.

# A Comprehensive Solution for Real-time Access to Adabas and Natural

Rocket® Data Virtualization (DV) is the industry’s only mainframe-resident data virtualization solution for real-time, universal access to data regardless of location or format. Rocket DV was built to make integration as seamless as possible for the programmer—it transforms Adabas or any non-relational mainframe data into a familiar relational format so that developers do not have to learn new skills.

Rocket DV can support a number of data integration use cases—from SQL access, to data virtualization (eliminating the need to physically move the data) to its unique brand of data replication using Extract



Virtualize and Load (EVL). The product's performance, scalability, and compatibility with current development tools reduce development time and increase the developer's ability to respond rapidly to new business requirements.

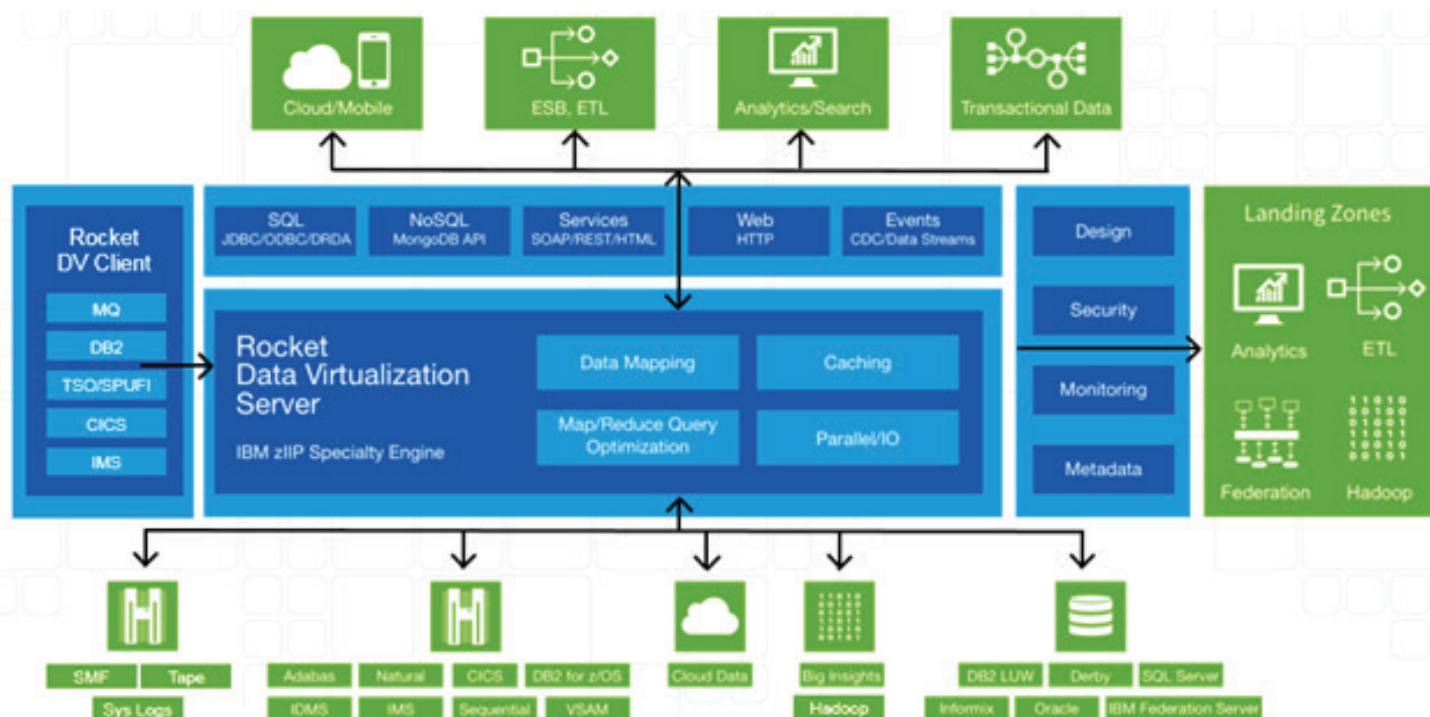


Figure 1. Rocket DV mainframe-resident data virtualization server that uses the IBM System z Integrated Information Processor (zIIP) to run up to 99% of processing.

## Universal Access to Adabas

Rocket DV provides developers with direct, real-time SQL access to Adabas and the ability to leverage native Adabas features. This enables simple, reliable, scalable, and secure enterprise-class integration between business applications running on distributed platforms and Adabas via ODBC, JDBC, and REST/Web Services.

With Rocket DV, the data held within Adabas is accessible by any application that uses standard SQL syntax. Rocket DV converts the SQL statement from the application into native Adabas calls and then converts the returned information into a result set for return to the distributed application.

Rocket DV provides high-performance integration with mainframe Adabas databases to deliver the following benefits:



Benefits of Rocket Data Virtualization for Adabas	
SQL Access	Provides SQL access via the ODBC or JDBC Adapters. The Rocket DV Interface for Adabas supports standard SQL syntax such as SELECT, INSERT, UPDATE, and DELETE. Data from Adabas is returned to the calling application as a relational result set.
Metadata Support	<p>Offers full relational metadata support for dynamic integration with development environments and business intelligence tools. Relational database management systems maintain comprehensive catalogs that describe schemas of the database, such as the tables available, the columns they contain, availability of indexes, relationships between the tables, and stored procedures. The Rocket Data Virtualization Facility extracts metadata from Adabas field definition tables (FDT) and PREDICT DDM.</p> <p>Application programmers can use standard metadata calls from their tool of choice to retrieve the metadata information which describes the layout of the Adabas objects. The names of these objects can be readily manipulated within Rocket DV to make them more meaningful to the application developer without impacting the structure of the database itself. With access to an internal view that contains metadata information from the underlying native structures, developers gain flexibility while maintaining the high-performance and production characteristics on the host, with no mainframe coding required.</p>
Support for Native Adabas Structures	<p>To ensure optimal application performance, the Rocket DV Interface for Adabas exploits native Adabas performance features, simplifying access while preserving these unique Adabas features. Rocket DV offers support for:</p> <ul style="list-style-type: none"> <li>• Native Adabas syntax, including ET and BT transaction logic.</li> <li>• Options to either flatten to or dynamically build virtual sub-tables to represent PEs (periodic groups) and MU (multi-value) based on the parent Adabas file, layering a relational structure over the Adabas multiple occurrence data.</li> <li>• Native Adabas search operations with multi-descriptor processing, which allows use of “where” criteria to search on a super descriptor, sub or hyper descriptors.</li> <li>• Native Adabas datatypes, including date and time, packed and unpacked numbers, and binary.</li> <li>• Adabas file level security.</li> </ul>
Native z/OS SAF Security	To reduce the complexity of integrating Adabas security with standard mainframe security products, the Rocket DV Interface for Adabas provides features that simplify the use of RACF, ACF2 and CA-Top Secret in an Adabas environment.
Aggregation of Multiple Files with SQL Joins	The Rocket Data Virtualization Interface for Adabas allows developers to SQL join up to five Adabas files with other virtualized tables in DB2, IMS, VSAM and sequential.
Cursor Processing	Rocket DV provides host cursor processing for more explicit and efficient control of aggregated data. This feature allows developers to bring data together from different files and maintain detailed control of the current record in use.



## Universal Access to Natural

Natural from Software AG is a popular fourth generation language (4GL) that generally deploys within the CICS environment. Developers can use the Rocket DV REST/Web Services interface to either call a Natural program via SQL metaphor or wrap and re-use the back-end services written in Natural for CICS or batch environments. This enables programmers to develop applications using their standard tools, while allowing existing applications to run undisturbed in their native transaction processing environment.

Interoperability with Natural is facilitated by a Software AG component called the Advanced Communications Interface (ACI). Rocket DV provides a native replacement for the ACI. This ensures that Natural programs can continue to interoperate with external systems without modification, while taking advantage of the range of connectivity options and the robust feature set presented by Rocket DV.

Additionally, Rocket DV dynamically allocates CICS or Batch resources on demand. These services remain available to accept work requests according to installation policy. Incoming requests from connected applications use the Rocket DV's Data Mapping Facility to correctly format a request for a Natural program, which is then handed to the ACI service. The Natural program runs as normal and hands the response back to the Rocket DV Server, which formats the response for the calling application.





# Features of Rocket Data Virtualization Interface to Natural

Rocket DV significantly reduces the complexity of integrating back-end services written in Natural for CICS or batch environments. Key features include:

Benefits of Rocket Data Virtualization for Natural	
SQL Call RPC Access to Natural Programs	Enables developers to execute a Natural program and return the data as a relational result set, making it easy for the distributed application to access the result set and manipulate the data.
Multiple Column, Multiple Records Returned	Developers can return large result sets out of a Natural program columnized in a tabular form with multiple records and multiple columns.
Direct Access to Natural Applications	Allows direct access to Natural through standard ODBC, JDBC, or REST/SOAP.
Metadata Support	Provides full metadata support for rapid application development. The Rocket DV DMF extracts metadata from Natural data area definitions and returns this information using standard metadata calls. This simplifies application development by ensuring full compatibility with common IDEs without affecting the structure of the Natural program itself.
Input/Output Data Checking and Transformation	Provides automated data conversion and validation for input and output to Natural applications.
Conversational or Non-Conversational Programs	If a developer requires several roundtrips to the same Natural program maintaining ET logic, Rocket DV supports conversational programs.
CICS Connection Fail Over	In the unlikely event a CICS region fails, Rocket DV can route subsequent requests to another region without interruption to the service.
Resource Protection	Manages starting and stopping Natural servers, based on need and excessive usage.
Interface to 3270	Provides is an additional interface to the presentation layer of Natural. This interface allows Natural 3270 screens to be unobtrusively accessed by the application for “screen- scraping” and enables presentation layer data to be aggregated and wrapped into single ODBC/JDBC requests to the mainframe.



The Rocket DV Server correctly handles all datatype translation issues between Natural and the calling client application.

## Advantages Gained by Rocket Data Virtualization

Rocket DV offers the most comprehensive Adabas and Natural interface available, delivering the following advantages:

### ❖ High Performance Data Architecture

The need for real-time information requires a high performance data architecture that can handle the extreme volumes and unique requirements of mainframe data. Rocket DV includes several query optimization features, such as parallel I/O and MapReduce. This significantly reduces the query elapsed time when accessing large mainframe non-relational databases by splitting queries into multiple threads that read the files in parallel.

### ❖ Reduced Mainframe Overhead

Rocket DV is unique in its ability to run up to 99% of its data virtualization processing in the mainframe System z Integrated Information Processor for significantly reduced mainframe total cost of ownership.

### ❖ Universal Connectivity

Rocket DV enables distributed applications to access mainframe resources through a breadth of flexible industry-standard APIs - ODBC, JDBC, REST, SOAP, JSON, HTTP, HTML, and XML.

### ❖ Less Complexity

Rocket DV has fewer "moving parts" than generalized messaging solutions often used to access Natural, providing greater reliability and higher performance. Rocket DV's thin client architecture is deployed directly on the mainframe with Adabas, requiring no new hardware such as a middle-tier gateway server. This simplicity translates to fewer potential points of hardware failure.

### ❖ Scalability

Rocket DV is designed to provide consistent scalability and performance for enterprise deployments by load balancing work across multiple servers and efficiently reusing network connections. These capabilities enable it to handle large numbers of concurrent users while ensuring consistent performance and reliability, and has been proven on distributed computing environments running millions of transactions per day.

### ❖ Tight Integration with Mainframe Security

Rocket DV offers a significant advantage over other Adabas and Natural interfaces by providing secure access to the mainframe. It has comprehensive facilities to ensure that authenticated end-users can be identified without requiring a user ID so that organizations can leverage existing



mainframe security infrastructure and reduce the overhead associated with defining and implementing security for new applications accessing Natural programs. Rocket DV supports Adabas file name security and file security, and allows implementation of security measures through RACF, ACF2, or CA-Top Secret without utilizing proprietary Adabas security offerings.

## Conclusion

Adabas and Natural are still used extensively by organizations around the world because they continue to deliver high value as a data repository, and maximizes that value by delivering the data stored within Adabas to new, dynamic applications throughout the enterprise. However, the challenges of integrating today's application development environments with mainframe environments make this a daunting task. Despite technological challenges and the limitations that most development environments have in connecting to mainframe data, it is still possible for organizations to make that connection and gain additional value from their Adabas data using Rocket Data Virtualization.

While enabling application development to integrate with Adabas and Natural may have been a challenge in the past, it is both feasible and economical with Rocket DV. The solution provides a seamless interface between the programmer's contemporary development environment and the previously impassable legacy Adabas and Natural environment. Now application developers can concentrate on the applications and leave the Adabas integration to Rocket DV.

And because Rocket DV is quickly deployed and easily maintained, it does not create an additional burden on already stretched developers. Rocket DV is a true asset to an organization's application development environment.

Rocket DV is the industry's only mainframe resident data virtualization solution, and this leadership is based on a foundation of customers who have successfully utilized Rocket Software technology to maximize the use of their mainframe. Rocket's commitment to open standards enables organizations to rely on Rocket DV and continue to maximize their Adabas investment for the long term, no matter what new development environments emerge on the horizon. The development gap between modern applications and the mainframe is eliminated when Rocket DV is deployed.

For the business, Rocket DV has a significant impact on the bottom line. Due to rapid deployment and low total cost of ownership, the solution delivers a quick return on investment and continues to contribute by boosting developer productivity, enabling development of high-performance applications, and saving money by eliminating the need for multiple interface purchases.

Rocket DV maximizes an organization's Adabas and mainframe investments so capital does not need to be diverted into new data storage options. The Rocket DV Interfaces for Adabas and Natural are economical



solutions that bring an organization's IT infrastructure together without major infrastructure purchases and disruptive data migrations by bridging the gap between leading edge applications and mainframe data.

## About Rocket Software

Rocket Software is a leading global developer of software products that help corporations, government agencies and other organizations reach their technology and business goals. 1,100 Rocketeers on five continents are focused on building and delivering solutions for more than 10,000 customers and partners—and five million end users.

Rocket DV enables mainframe relational and non-relational data to seamlessly integrate with big data, analytics, and web/mobile initiatives; eliminating the need to move or replicate data, and with significantly reduced costs, complexity and risk.

- ❖ The industry's only mainframe-resident data virtualization solution for real-time, universal access to data, regardless of location or format.
- ❖ Support for Data Providers: IBM Big Insights, Hadoop, MongoDB, DB2, Oracle, Microsoft SQL Server, VSAM, IMS, Adabas, and others.
- ❖ Support for Data Consumers: Cloud, Mobile, Analytics, Search, ETL, as well as ODBC, JDBC, REST, SOAP, JSON, HTTP, HTML, and XML.
- ❖ Reduced mainframe TCO: Engineered to divert up to 99% of its integration related processing to the System z Integrated Information Processor (zIIP).
- ❖ Universal DB2 Support: Applications using DB2 can now seamlessly integrate with any non-DB2 data source with the same ease of functionality.
- ❖ Asymmetrical Request/Reply: Any mainframe application (Batch, Started Task, IMS, IDMS, Natural) can interface with DV to request data for itself or another applications.

Our customers tell us that IBM System z—the mainframe—is still the best platform in the world for running their critical business applications. And those applications generate and access large data volumes—big data. Increasingly, those applications and data must connect with other applications within the enterprise and even outside the enterprise. Rocket has deep domain expertise and world-class technology to keep the data where it belongs and move the analytics closer to the data.

