

**THE ENSIVA DATA SERVER...
...TRANSCENDING THE JDBC/ODBC BOTTLENECK**

ENSIVA WHITE PAPER

Data Data everywhere but not a byte to read!
-Anon

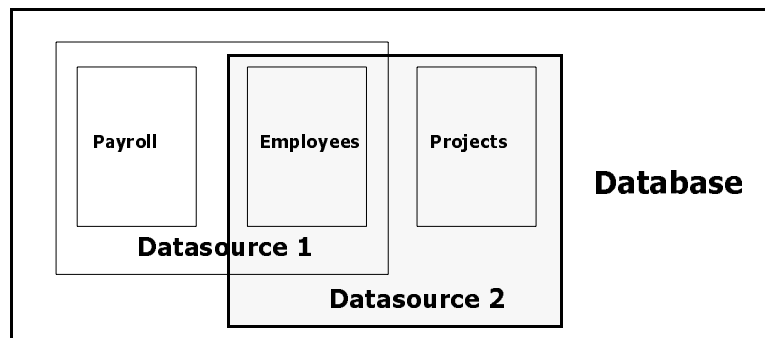
Introduction

Ensiva is all about dynamic interactivity, catering to different devices and modes, by voice, wireless and traditional browsers, all over the Internet. Dynamic interactivity implies dynamic exchange of information between the end-user and some information source, which could either be another user or a data source. When there is even a mention of the term data source or database or RDBMS in the software development world, one immediately thinks in terms of ODBC/JDBC. This paper explores how Ensiva transcends the ODBC/JDBC paradigm by providing an optimized and programming-free interface to the RDBMS world.

The importance of providing optimized and easy data access to both designers and end-users warrants the need for a separate server whose sole role is to provide database services to the other servers. Ensiva calls this server the Data Server and it is the gateway between the business logic and the actual data.

Database, Database Schema and Datasource

A database schema is the information on how a database is structured in terms of tables and columns. A database is the actual repository of data. Datasource definitions provide a smaller and more manageable “window” into the database. Figure 1 is a simple diagram illustrating these definitions:



Here “Datasource 1” is a logical grouping of two tables, Employees and Payroll. Similarly, “Datasource 2” is a logical grouping of two tables “Employees and Projects”. If there are developers who need access to payroll information, then the Ensiva administrator can grant them access to Datasource 1. For developers who need access to projects information, Datasource 2 is defined. Physically, a datasource is merely a definition and not a duplication of the data whose schema it encompasses. It is possible to define a datasource on non-RDBMS data, such as flat files and other datasources such as LDAP will be supported shortly.

Ensiva Servers

The Ensiva Data Server is a bridge between other Ensiva servers and the datasources along with their underlying data.

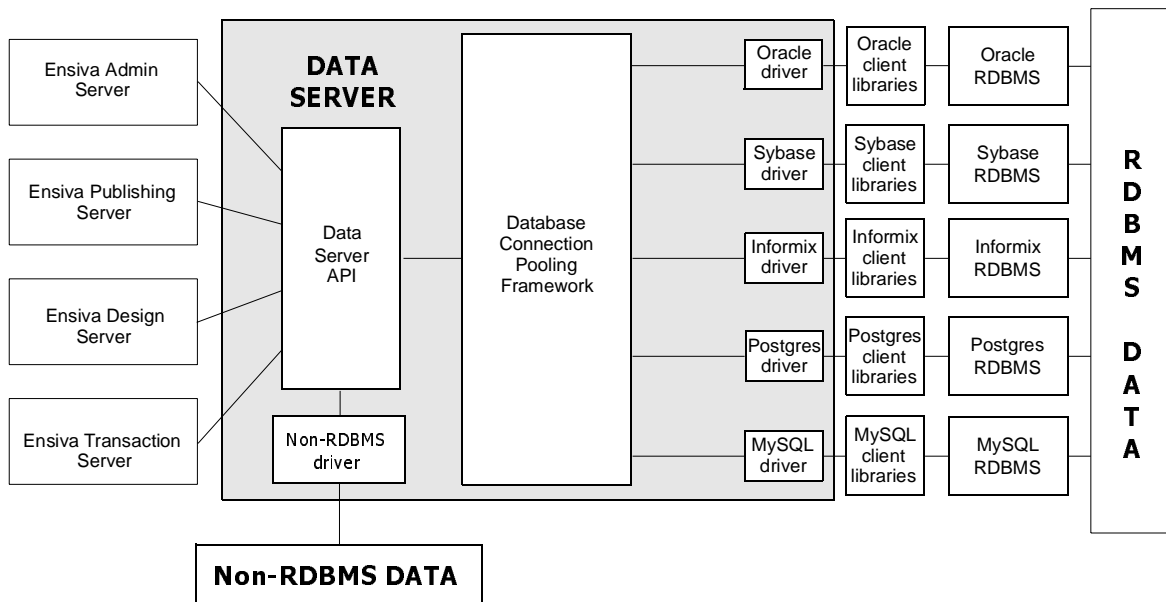
To understand the design philosophy behind the Ensiva Data Server architecture, it is important

to know that database interaction falls into three categories, based upon the following three user types:

- Administrator**
 An administrator needs access to the database schema so he can create a datasource definition. He also provides appropriate permissions to this datasource. The Ensiva Administration Client, a Java-based Graphical User Interface, interacts with the Ensiva Data Server through the Ensiva Administration Server.
- Designer**
 A designer needs to know the underlying schema behind the datasource so he can use the items in the datasource in his applications. This could be for the purpose of publishing these items or for updating them based upon end-user requests. The Ensiva Designer, another Java-based Graphical User Interface, interacts with the Ensiva Data Server through the Ensiva Designer Server.
- End-user**
 End-user uses the applications deployed by the designer and transparently interacts with the datasources used in the applications. End-user, of course, is neither aware of the datasource, the underlying RDBMS or any such details. The End-user, using browsers such as HTML, WML and Voice, interacts with the Ensiva Data Server through the Ensiva Publishing and Transaction Servers.

Figure 1. displays the architecture of the Ensiva Data Server and how it interacts with other Ensiva servers.

Figure 1. Data Server Architecture



The Ensiva Data Server serves as a bridge between Ensiva servers and RDBMS data as well as non-RDBMS data such as flat files. The essential components of Ensiva Data Server are the Ensiva drivers. These RDBMS-specific drivers are written to handle SQL features and datatypes specific to a particular RDBMS. The Data Server itself also provides a connection pooling framework and a high-level data-independent API which is used by other Ensiva servers in their implementation. It is not important for the Ensiva developer to know this API. In fact, knowing it, serves no purpose, because it is neither required nor can be used by the developer directly. By encapsulating any RDBMS specific features inside the Ensiva drivers, the Ensiva Data server provides the Data Server API layer which has common set of API calls independent of a particular RDBMS. This API is used by other Ensiva servers, such as the Administration, Publishing, Design and Transaction servers. This makes the Ensiva architecture very extensible since support for additional RDBMS or non-RDBMS data can be added by simply writing a driver for it. Also, support for additional features in an RDBMS can be provided by simply enhancing the driver.

Services Provided by Data Server to Other Servers

As mentioned above, the Data Server's role is to serve as a bridge between the data world and Ensiva's other servers. It provides different functionalities to the different server. Indeed the different servers' needs vary and the Data Server should be able to cater to all these needs and is designed accordingly.

Services provided by Data Server to Design Server

The Design server provides services to the Ensiva designer. Therefore, the services provided by the Data Server to the Design Server are those which an application designer would need. Such services include providing access to the definition of the datasource, the database tables and columns in it and, of course, the data types of these columns.

The Design server ensures that appropriate security is enforced so that designers are provided access only to those datasources to which they have been granted permission by the Ensiva Administrator.

Services provided by Data Server to Publishing and Transaction Servers

The Publishing and Transaction Servers together form the backbone of the Ensiva run-time environment. The Publishing Server uses the Data Server to retrieve information from the application datasources and then delivers it to the end-user in a language and format, as defined in the application. The Transaction Server manages database updates requested by the end-user. The actual database updates are performed by the Data Server.

Services provided by Data Server to the Administration Server

The Data Server provides two important services to the Administration Server:

- Providing information required to manage datasources.
Management of datasources include tasks such as creating of datasource definitions, granting selected designers access to them, deleting them, modifying their definitions and specifying the connect permissions required for the Ensiva Data Server to access the underlying RDBMS.

- Providing access to administration information that is stored in the Ensiva System database.

Ensiva stores all administration information required for the proper functioning of the Ensiva system in a RDBMS database. The datasource referring to this database is called the Ensiva System datasource and the Ensiva Administrator can choose any RDBMS available to store this database. In fact, several configurations can be simultaneously stored in different datasources and the Ensiva system can be booted from one of several available configurations.

The administration information stored in the Ensiva System datasource is made available to the Ensiva administration modules in the Ensiva Administration Server as well as in other Ensiva servers, by the Data Server. To the Data Server, the Ensiva System datasource is just like any other datasource. Other servers access information from this datasource just as the Publishing and Transaction Servers use the Data Server to access application datasources.

ODBC/JDBC vs. Data Server

No discussion on RDBMS database interaction can be completed without including an understanding of the ODBC/JDBC standard. The similarity between the Ensiva Data Server and an ODBC/JDBC driver is that the latter also provides connectivity and interaction with RDBMS databases. However, the similarity ends there. While JDBC/ODBC is an API for developers, the Ensiva Data Server provides this functionality to the Ensiva application without the need for any programming.

Though database connection pooling is provided in some ODBC/JDBC drivers, this feature may not be fully supported or in some cases the support may be inconsistent. Ensiva Data Server provides Data Connection pooling for all RDBMS it supports and it supports almost the entire spectrum of popular RDBMS. Perhaps the most important distinguishing factor between the Ensiva Data Server and ODBC/JDBC drivers is the performance advantage that the Data Server provides over the use of ODBC/JDBC drivers. The Data Server uses Native RDBMS APIs in its implementation and therefore is able to provide the maximum throughput that the RDBMS engine can support. This is clearly an important issue, specially when developing Voice-driven applications, where end-users are sensitive to any delays in system response. The delay in data interaction quickly propagates to the end-user, jeopardizing its usability.

Providing Database Transparency

Most application developers have to constantly deal with database schema changes, connection ids and passwords, RDBMS version dependencies and other such tasks which can be a productivity drain on their main task: to develop the application. Ensiva eliminates this by providing a very simplified notion of a datasource. The application developer does not have to deal with issues such as the actual RDBMS version, the connection id/password required to access the underlying RDBMS or even the actual location of the database. The datasource abstraction is made possible by the Data Server.

Ease in Database Administration

The Ensiva Administrator manages these issues through the Ensiva Administration client. This

central management of datasources ensures consistency in their definitions and access privileges. It also simplifies the task of the Database Administrators (DBAs) by providing database access information to a single entity: The Ensiva Administrator. The Ensiva Administrator can then impose additional restrictions when assigning access to these databases to the different Ensiva application. No further DBA intervention is required.

Extensible architecture of Data Server

Data server currently supports Oracle, Sybase, Informix, MySQL and Postgres RDBMS. However, it is worth pointing out that the architecture of Data Server is very extensible. All information specific to a particular RDBMS is encapsulated in the so-called database drivers which serve as the bridge between RDBMS native API interfaces and common API provided by the Data Server used by other Ensiva servers. In this way it is very easy to provide support for any other RDBMS. Just plug-in the database driver into the Data Server and you are ready to use the new RDBMS! In fact, any data driver, not just RDBMS can be plugged in and in fact there are plans to provide a datasource abstraction for LDAP in the next Ensiva version. Currently, flat file drivers are already supported.

Data Server as a Heterogeneous Database Junction

Data Server provides access to different types of RDBMS through the datasource abstraction. Applications that use the datasource need not be aware of the underlying RDBMS, though this information is accessible to the application developer. An application developer could possibly embed several datasources in a single application (page or document) without been concerned or aware of their heterogeneity!

The mapping of different datatypes to a standard, is handled by the Data Server. This enables the application developer to focus on the business logic aspect of the data rather than been concerned with the RDBMS-related issues.

Datasource Connection Pooling

In a performance-sensitive environment, every millisecond counts. Ensiva was designed ground up to optimize performance at all levels. In a database intensive application, the time to connect to a RDBMS server can be substantial, depending on the RDBMS. Ensiva provides extensive built-in support for database connection pooling in the Data Server. The Ensiva Data Server can be made to pre-open several connections when it boots for each datasource defined within its context. This can be tuned based upon the application requirement. For example, a datasource could be used in multiple heavy usage applications, in which case the Ensiva Administrator may keep a large number of pre-opened connections to this datasource. Another datasource used in an application with relatively less frequent access could have a lesser number of pre-opened connections. The set of pre-opened connections is often referred to as a "connection pool."

When an application requires a database connection, it uses the existing connection from the database connection pool rather than opening a new one. When the application no longer needs the connection, it returns the connection to the connection pool for reuse. This results in substantial performance enhancement and is an important part of the Data Server functionality.

While the concept of connection pooling and the advantages offered are relatively a standard offering in most application servers, Ensiva's Data Server provides a tunable connection pooling environment for a heterogeneous set of databases with absolutely no programming required!

Leveraging Performance Benefits of RDBMS Native API

Using native RDBMS API interfaces certainly provides substantial performance advantages as compared to JDBC/ODBC. The run-time access to RDBMS data is significantly improved. Unlike some other “standard mapping” solutions, Data Server is not implemented on top of ODBC or JDBC. Data Server uses native RDBMS drivers and implements intelligent caching and pre-fetching algorithms. The operation broker implements optimized access strategy based on specific features provided by the RDBMS, such as multiple-row fetches and array interfaces. In the absence of the Data Server, application developers would have to code at the native RDBMS API level to leverage the RDBMS-specific performance enhancement features. Ensiva’s Data Server delivers access to the database through its GUI development tool leaving the critical task of optimizing database access to the Data Server.

Common Abstraction for Flat files

The Ensiva Data Server provides datasource abstraction for non-RDBMS data also. Currently only flat files are supported but support for LDAP data is forthcoming. Files with fixed-size fields as well as delimiter separated fields are supported. Support for flat files makes it easy to publish data from flat files such as from server logs.

The flat file datasource abstraction enables the Ensiva application developer to access the different fields of the file like a database table with no programming requirement.

Security Provided by Data Server

While the key motivation behind the design of the Data Server was based upon performance considerations, it was not achieved at the expense of data security. Indeed, the design considerations balanced ease in database administration with clear, unambiguous rules on how the database security is to be administered and propagated.

The security rules for datasource administration are simple:

The RDBMS DBA provides the Ensiva Administrator with one or more connections associated with different access privileges and therefore having different connect strings, such as user-id and password. Using these connect strings, the Ensiva Administrator defines datasources. Multiple datasources, which use the same connect strings, could be defined. This enables the Ensiva Administrator to impose additional restrictions on the tables to be included in a datasource.

Once the Ensiva Administrator has defined these data sources, he grants access to them to different Ensiva application designers. Application designers in turn use these datasources to create and deploy applications. When they deploy an application, they can specify the users who can access the application, thus implicitly propagating the datasource access to the end-user.

Conclusion

The features and advantages of the Data Server are summarized here:

- Abstraction from the complexities of RDBMS programming interfaces. No efforts from the application are required to manage data in RDBMS. Both update and retrieval operations in the databases are performed transparently.
- Abstraction from the complexities of knowing authorization scheme for a particular database. What the developer gets is the datasource name which can be used to access the database schema.

- Support for flat files. Flat files can be easily published in any Ensiva application without any programming. The Data Server manages the reading and parsing of the records presenting the information to the application as a series of fields.
- Data Server uses native RDBMS interfaces which provide superior performance throughput and far outperforms other database access methods like ODBC/JDBC.
- Data Server provides the required security mechanism to enable the Ensiva Administrator to provide selective access to different datasources.
- Support for heterogeneous databases. Data Server supports all major commercial and open source RDBMS such as Oracle, Sybase, Informix, MS-SQL, DB2, MySQL and Postgres. With Ensiva, it is easy to develop applications which have to access multiple heterogeneous datasources in a single page/document.
- Data Server provides a tunable connection pooling mechanism which further enhances the performance of applications developed in the Ensiva environment by saving the time required to establish a connection with the RDBMS engine.

Data Server provides the backbone for database interaction in the Ensiva environment. Its significance is enhanced because Ensiva itself uses a database-driven mechanism for managing configuration of its own servers and resources.