

**HiT**SOFTWARE®

# **The Business Value of HiT Software Data Integration Tools: Allora and DBMoto**



## Introduction

Data integration is critical to the success of enterprises today. Competition constantly drives businesses to find better ways to handle information, including having centralized stores of data available in real-time for informed decision-making; buying new applications that use existing data without adding overhead to existing applications; and improving overall cost-effectiveness and productivity in accessing and maintaining enterprise data.

Some of the most challenging data integration scenarios involve gathering data from multiple databases across multiple platforms. Solutions are typically evaluated based on low cost of ownership, respect for the integrity of existing systems, lightweight installation, ease-of-use, predictability, and open, extensible architecture.

This white paper presents the issues facing businesses with data integration needs. It then compares the business value of the principal approaches to database integration: doing it yourself, investing in a single-vendor enterprise solution, or integrating data via specialized tools to provide an accelerated custom solution.

## Expectations for Data Integration Tools

Successful business operations across all sectors depend on the availability of data. In an ideal world, enterprises would design their information systems from the ground-up with the latest available technology to serve their specific business needs. Realistically, however, enterprises are constantly struggling to find ways to integrate new applications with heterogeneous legacy applications and databases that have accumulated over the company's years in business. Following are the major concerns that any successful data integration solution will need to address.

### Continuous Operation

To be cost effective, any new data integration solution should ideally be implemented and maintained without disrupting existing applications and databases.

When adding a new application that relies on an existing database, it is advisable to replicate data from a production system to avoid impacting performance and to isolate the production system from potential problems because of the new application. Replication typically involves no installation of a third-party component on the production server. For example, customers that rely on Oracle or DB2 databases for their business often prefer to create a database replica using Microsoft SQL Server, MySQL or another database when creating a new web application. This way, they are not disrupting production systems and can leverage many Internet tools available with lower cost database servers. Replication can be one-way, bi-directional, a one-time event (migration) or continuous (real-time mirroring).

A new approach that in recent years has gained significant momentum is the use of integration servers and web services in conjunction with legacy data sources to query legacy data and transform results to XML for input into new applications. The production system or legacy data source remains untouched. This approach provides maximum interoperability and makes it easy to integrate messages/XML data into any process or with any application.

Companies should look for solutions based on data connectivity standards. Relational database connectivity products, from database vendors or third party companies, support any version of any database, any file format, and any application, including legacy applications and their future versions. .NET, OLE DB providers and ODBC drivers supply the Windows developer market, while JDBC drivers are the standard used by the Java community. JDBC type 4 drivers achieve performance and portability by eliminating the need for gateways. Once new applications and standard drivers are in place, there is no wait for the latest version of the data integration tool to support the new release of your database engine: Future database releases are automatically supported.

## **Interoperability with Legacy Systems**

Data replication should involve no installation of third-party components on a legacy database server. A one-way transfer (refresh) of data or bi-directional transfer (mirroring) to a smaller, more cost-effective database server such as Microsoft SQL Server or MySQL provides access to legacy data without impacting the legacy system in any way.

Whether performing query-only access to legacy data sources or writing back to legacy databases via web services or applications, transformation of data from/to XML is quickly becoming an alternative data integration solution that complements direct SQL approaches from traditional data replication middleware.

## **Support for Data Warehousing**

A data warehouse typically stores all enterprise data, allowing users to extract data for business applications such as tracking business trends and facilitating forecasting or planning efforts. Data integration and ETL tools (Extract, Transform and Load) are used to populate data warehouses, where complex transformations can be required to make the data suitable for end users. Tools are available for building data warehouses or operational data stores in batch or real-time and for giving users access to enterprise data.

## **Secure Data Access**

With more applications sharing and integrating databases over public networks like the Internet, security concerns grow. It is important for any data integration solution to address data security issues including the encryption of data, user authentication, and safe data transport. Making mission-critical data easily accessible in real-time by business users has grown parallel with the need to protect data from unauthorized users. Information is the blood of most business operations today, and its loss, corruption or embezzlement can mean the end of the business altogether. Security can be built within the network, the protocol layer or the data layer. Many solutions exist, but they are often complex and not always compatible with the existing framework. For this reason, it is critical to choose data integration solutions that fully support standard security packages.

## **Data Availability**

High availability of production database servers is a significant concern when choosing a data integration solution. Data replication tools allow companies to access data from production database servers without placing a load on the servers and with a complete guarantee that the data will not be accidentally modified. By creating a mirror of critical database servers, these tools unlock company data and make that data readily available to web or business intelligence applications. It is important to note that the database replica does not need to be from the same vendor as the source database. In fact, the customer will often choose a target database vendor with the most functional and extensive application development framework meeting their internal requirements in a cost-effective way.

Additionally, with the rise of XML, the lingua franca of data exchange, it is becoming more critical to view corporate data according to specific semantics or XML schemas defined by industry standard committees. Hiding the complexity of SQL queries, XML transformation tools access production servers in order to make relational data readily available in a text-based format and readable by any individual or to store XML data in databases in relational format. XML is a standard where both semantics and the content of the data are naturally expressed, and it has become the reference for any integration project. XML transformation tools can exist alongside production database servers and perform transformations without affecting the availability of the servers.

## **Heterogeneous Database Support**

Most companies maintain databases from multiple vendors; it is therefore important for a data integration solution to include the ability to easily replicate data within a heterogeneous database environment that has evolved with each IT cycle. Replication tools provide a means to perform a one-time transfer of data from source to target, as well as continuous mirroring or synchronization between databases. Replication is often used to populate operational data stores or to distribute data between a central database and multiple, remote subsets of that database that operate independently between updates. An ideal replication tool should provide replication to and from leading database vendors including Oracle, DB2, SQL Server, Sybase or MySQL, or even any database via ODBC, OLE DB, .NET or JDBC database connectivity protocols. In brief, they should support data replication and transformation through the use of the full set of SQL functionality including stored procedures. XML also plays a key role in heterogeneous environments as it is platform and application agnostic.

## **Replication and Heterogeneous Platform Support**

In organizations where enterprise data and applications reside on different hardware platforms, data integration tools and solutions that offer platform independence or are available on multiple platforms can initially seem appealing for ease-of-use reasons. However, tools that are available on different platforms can also have limitations. Some features and functionality may be lost or modified in porting the tool to a new platform. Software vendors may not synchronize new releases of their products on multiple platforms and may not be able to maintain all products at the same level. For these reasons, a solution that runs on a common platform, but is able to access and work with heterogeneous platforms, is often preferable. This

approach guarantees that functionality will be the same on all platforms, thereby reducing training and support costs.

## **Scalability**

In choosing a data integration solution, the issue of scalability must be considered. Even if initial estimates of the amount of data involved are accurate, the solution you choose should continue to perform as the amount of data grows over time. Relational database vendors have a proven track record of reliability and scalability, including enhancements to support parallel processing and load balancing. Any data integration solution should also demonstrate reliability and robustness in the face of large amounts of data. It is worth asking for customer references and test cases, as well as assessing the stability of tools by evaluation and length of time on the market.

## **Open Systems**

In a typical business environment that consists of heterogeneous database servers and applications running on different platforms, adding a data integration solution that consists of a proprietary server and related adapters is a significant risk, in terms of cost, maintenance, and dependence on a single vendor. If enterprise information systems have grown based on choosing the best tool for a specific job, companies should use the same approach for data integration. A data integration solution should allow enterprises to keep their systems open to the addition of further suitable components as the need arises.

Database vendors usually want to lock customers into the database they provide. Even if they offer cross-database solutions, they are most often for moving data from other databases to theirs. Using an independent application is the only way to assure that the integration will work well in situations where heterogeneous databases must cooperate.

## **Cost**

Costs involved in data integration solutions fall into four areas:

- Initial acquisition cost, including licensing, consulting and support fees
- Initial and ongoing employee labor costs, including training of employees, implementation costs and modification costs
- Maintenance costs associated with upgrades, additional required components
- Back-out costs associated with the impact of moving to a different vendor or if the integration vendor ceases operations.

Enterprises should also consider the fact that when making company-wide financial decisions, executive managers tend to look more favorably on investment in immediate revenue generators. For this reason, a low-cost alternative to data integration needs is critical for the success of the project.

## Data Integration Options

After examining the needs and issues associated with data integration, enterprises then need to assess the data integration solutions available. Solutions range from a do-it-yourself approach (where an IT team plans and implements the applications needed) to a complete single-vendor enterprise solution that requires substantial investment of both money and time.

### Do-it-yourself Solutions

IT teams often address immediate data integration needs through manual scripting, often using homegrown code (such as SQL, Perl, and Java) to integrate data. However, this approach can quickly grow unmanageable for some or all of the following reasons:

- **Maintenance**  
As IT group members change and new technology is introduced, knowledge of already-implemented solutions is lost, so maintenance becomes a problem when there is an upgrade to a database or a new operating system release is introduced. Maintenance is always by far the largest cost factor of any system.
- **Portability**  
Scripts and code that were designed as stopgap measures to solve immediate problems have an insidious way of becoming permanent. A solution works for a while, but the developer never has time to go back and implement a more generalized approach. Then the same approach is needed for a different platform. The original script is platform-dependent, however, and it turns out that the developer has left the organization. Additional significant investment is now needed to port the original stopgap solution to the new platform.
- **Labor cost**  
When choosing to develop a data integration solution in-house, it is easy to overlook the labor costs associated with understanding and learning the existing systems, evaluating an in-house solution, and then learning about and implementing the new solution. Also, the time invested in a developer or group of developers is lost if a key group member leaves the company.
- **Non-optimal solutions**  
In-house development groups have highly-specialized knowledge of existing systems. However, when implementing a solution that requires new languages or new technology, groups may come up with a less-than-optimal result. There may be performance problems or critical GUI features missing. The end result is a loss of business efficiency for enterprises in spite of the significant development investment.
- **Increasingly complex projects**  
Typically, in-house development efforts solve an immediate problem in the least amount of time possible. In the interests of saving money and resources, limited thought goes into developing a generalized solution that can be scaled up to meet future needs.

- **Time to availability**  
With in-house solutions, IT teams often need to work on new projects in addition to their current responsibilities. For example, in a small web-based business, the web master who is skilled at writing scripts may be asked to solve a data integration problem, but at the same time has a critical full-time job keeping the web site up and running to serve sales. Development efforts will come second to the daily demands of keeping the web site running, therefore delaying the implementation of the data integration solution.

The benefits of a homegrown solution include the availability of in-house expertise on existing systems and clear identification of data integration needs, as well as a custom solution to the problem at hand. However, while it might appear that in-house development is virtually free, the cost-effectiveness of developing homegrown data integration solutions is largely an illusion. While paying the salaries and benefits of the development team for an IT project, the opportunity to put developers on more strategic projects for the company business goals is missed.

A packaged integration software application may provide the solution. While initial financial investment is required to purchase the package, if in-house resources are put to best use identifying the package that fits their needs, the solution can be cost-effective.

## **Single-vendor Enterprise Solutions**

One possible approach to a packaged solution is the purchase of a single-vendor enterprise solution. The appeal of an enterprise-wide solution is that it could take care of all data integration needs for the foreseeable future while dealing with a single company that has expertise in the data integration arena.

Single-vendor data integration solutions can come from these sources:

- Database vendors, who supply tools to convert and integrate existing data sources and then use their own database environment as a basis for an enterprise-wide data integration solution
- Application server vendors, who supply adapters for different databases and a development environment with components for building applications
- Data Integration specialists, who either have full product solutions or provide solutions through a combination of consulting and pre-built components.

With any of these options, you typically need to accept and deal with the following issues:

- **Cost of acquisition**  
A single-vendor solution can be very powerful, with great transformation or replication capabilities that integrate existing heterogeneous environments, perform complex transformations, and interface with a variety of well-known database and application servers. However, the entry price for a single-vendor data integration solution can range from \$100,000 to \$500,000 or beyond, depending on hardware, number of CPUs, and operating system. Additional costs are incurred for installation and setup, training, support and ongoing consulting for changes to the initial configuration.

Enterprises should consider that professional services fees can surpass the initial licensing fees for single-vendor data integration solutions because of the level of complexity involved with all-in-one integration platforms.

Many single-vendor data integration solutions offer more features than a single enterprise typically uses. Consequently, when purchasing a solution of this type, companies may be paying for features that they do not need.

- **Proprietary engine**  
Single-vendor solutions typically make use of a proprietary engine and/or proprietary languages. These usually require complex setup and tuning using vendor consultants. Any changes or customization to the initial setup require either additional investment in consulting or investment in employee training to manage the system internally.

Enterprises should also consider that proprietary languages are usually comprehensive, but not easily extensible. If enterprise requirements change, the solution companies choose needs to be flexible enough to adapt.

If evaluating a single-vendor solution, enterprises need to monitor installation and setup of product in their own environment. This will give them a good sense for the complexity of the product and how flexible it may be.

- **Implementation issues**  
When making a significant investment in a single-vendor solution that involves complex installation and setup, it makes most sense to implement the solution enterprise-wide. This can mean interruptions and down-time in different departments as the solution is implemented. Implementation may also require additional resources, man and machine, to run existing systems alongside the new ones until the new implementation is tested and found reliable. An implementation schedule should be carefully planned for minimum impact to daily business.

## **Specialized Tools for a Custom Solution**

In most cases, the benefits of a set of well-designed, packaged integration tools outnumber those presented by custom-built solutions or a single proprietary enterprise solution. Appropriately chosen tools can provide the transformation power, flexibility and scalability needed to resolve any data integration problem.

This approach could be called “use what you have; buy what you don’t.” It takes advantage of all existing investments and expertise in the enterprise but allows for easy, accelerated implementation of data integration solutions that meet specific needs.

- **Low Total Cost of Ownership (TCO)**  
The most comprehensive packaged integration tools keep integration costs down via the following:
  - An intuitive graphical user interface for mapping data between one system and another, avoiding the need for costly custom coding.

- Optimization of development resources as programmers can plug into more strategic development projects.
- Fast implementation and deployment, which reduces the cost of the entire integration process.
- Leverage of existing industry-standard protocols and formats via a broad range of database connectivity (ODBC, OLE DB, JDBC, .NET), standard languages like XML, SQL, Java and Microsoft platform programming languages, and messaging standards like JMS, MSMQ, SOAP for disparate platforms.

The biggest integration cost of a homegrown solution is labor: training and learning, and custom code development and maintenance. Employing tools that accelerate development and deployment while using the expertise of existing staff significantly reduces labor costs. Also, limited investment in products is needed to provide flexibility, performance and scalability.

At the other end of the scale, costs associated with implementing an enterprise-wide single vendor solution are typically high initial license fees, a deployment cycle that can involve significant down-time, and consulting fees to initially customize the solution and then maintain it. These solutions are often proprietary and so complex that they require specialized consultants that come at a high premium. Customers, however, are looking for solutions that empower their own technical staff.

- Accelerated implementation/faster time to market  
For organizations with quick-time-to-market requirements, packaged software offers short learning curves, easy design tools and highly-focused solutions. They can target a specific data integration issue within the organization, choose a tool and solve the problem within a reasonable budget and timeframe. It is often surprising to see what small IT teams can accomplish when they are provided with the right development tools. In many cases, a single developer achieves better results where armies of developers and consultants previously failed.
- Non-proprietary and standards-based  
Many of the available data integration tools rely on standard languages and components. For example, tools might use scripting with Javascript or VBScript to customize data transformation between one database and another or between relational data and XML.

Organizations should look for data integration tools based on standard data access protocols such as .NET, JDBC, ODBC and OLE DB. These drivers/providers are commonly available for almost all database types, so data integration tools that use them typically offer integration for heterogeneous databases.

In using standard data access protocols, data integration tools are more flexible and cost-effective. Development efforts are put into solving data integration problems rather than re-creating efficient data access components. This approach guarantees an extensible solution that can evolve over time rather than being locked into a proprietary solution.

- **High Reliability and Proven Performance**  
When developing an integration application in-house, there is a time-consuming iterative testing process. On the other hand, a packaged integration solution has already been tested and offers high reliability with consistent performance. Testing efforts are therefore significantly reduced. The number of development cycles of a software product is directly proportional to its quality, and there is no better guarantee for a software component than to know that large and fast-growing companies are relying on it for their mission-critical business applications.
- **Easy availability**  
Data integration tools are often available for online download so that prospects can easily judge for themselves of the ease-of-use and functionality without requiring an army of consultants at prohibitive costs. Companies that have chosen to make their product readily available over the web are depending on the ease-of-use of the software tool to be successful. As a consequence, these companies are less likely to disappoint their customers.
- **Technical support**  
Even the most easy-to-use software tools sometimes require the support and advice from an expert to overcome a technical roadblock and speed software development. Technical support needs to be fast, precise, concise, helpful and cordial at all times and support multiple time zones in our global development economy. A knowledge base, documentation and a large community of users are also important.

## **HiT Software Data Integration Tools**

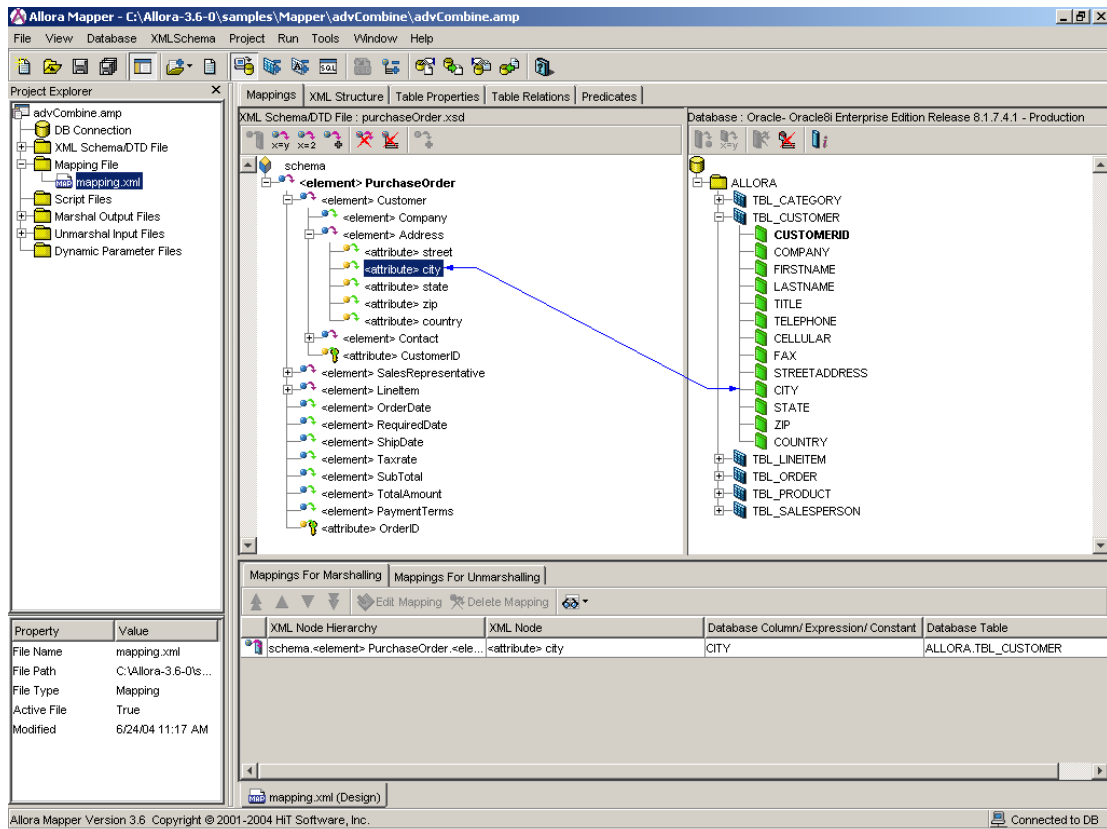
For over 10 years, HiT Software integration middleware has been recognized for its ease-of-use, scalability and performance. HiT Software is known for the following:

- Expertise in data access tools
- Use of easily-recognized industry standards
- Data access products that support heterogeneous databases

To serve enterprise data integration needs, HiT Software offers Allora for bi-directional transformation of data between XML and relational databases; DBMoto for data replication between databases; and SafeConduct for secure point-to-point communication between 2 systems.

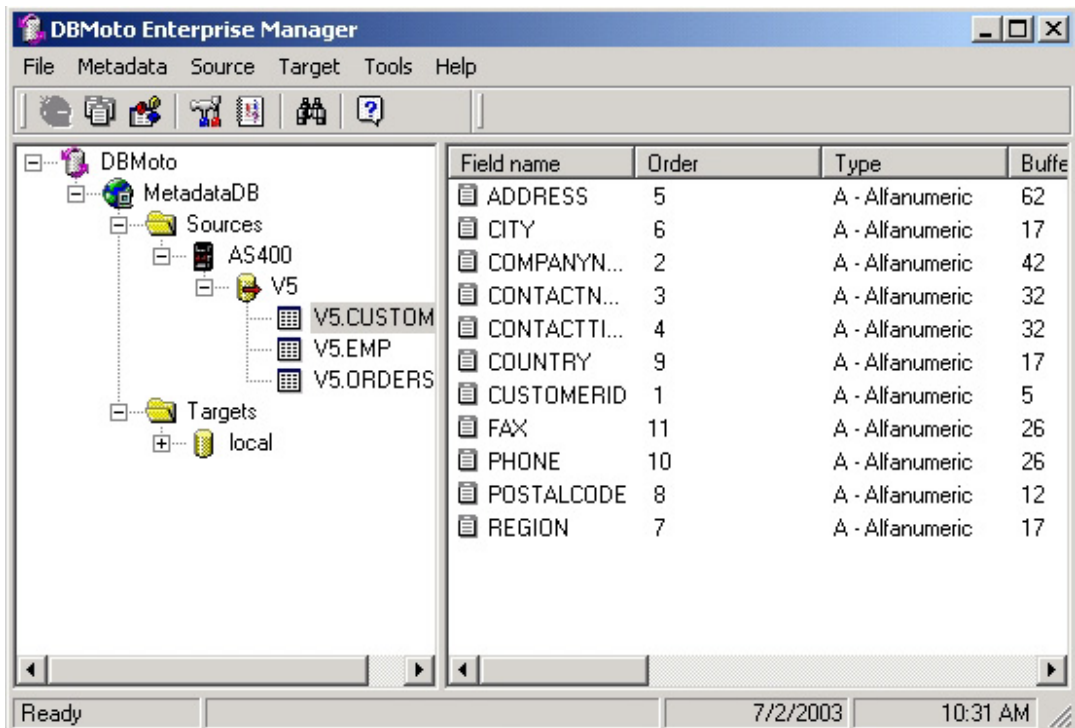
## Allora

Allora leverages leading-edge XML mapping and database technology to give application developers bi-directional access to relational databases without the need for complex SQL or XSLT programming. Allora speeds development and deployment through use of an intuitive Mapper to accurately and transparently transform data structures between XML elements/attributes and relational database structures. Once Allora mapping is in place, full bi-directional XML-RDB access is enabled. Allora gives developers a simple, consistent XML interface to relational data. Allora is particularly well suited to address the challenging task of importing XML schema-defined data into databases in relational format and according to database schemas.



## DBMoto

DBMoto performs refresh and real-time data replication for enterprise server and desktop replication needs. All major database platforms such as IBM DB2 UDB (including iSeries/AS400 and zOS), Oracle, Microsoft SQL Server, Sybase Adaptive Server Enterprise, MySQL, Microsoft Access and others are supported. In Refresh mode, DBMoto reads data, applies administrator-defined mapping rules, and writes the result on the target database. In Mirroring mode, DBMoto performs a real time incremental replication based on transaction logs. A third option, bi-directional mirroring, allows enterprises to keep databases synchronized. Powerful, easy to use administration wizards make setup a snap. And, unlike other data replication solutions, DBMoto offers unlimited control over data replication and transformation using standard scripting language.



## SafeConduct

SafeConduct brings the benefits of the Secure Sockets Layer (SSL) v3.0 standard, including digital certificate authentication and 256-bit data encryption, to any point-to-point application data traffic. Many enterprise applications today access sensitive data across the Internet or VPNs without employing data encryption or client authentication. As such, legacy application data traffic is susceptible to unauthorized access or tampering and thus raises serious information privacy and quality risks. SafeConduct transparently removes these risks by securely protecting virtually all application data access without impact to existing applications. SafeConduct can be used with a variety of application architectures including client-server, host-slave, or

multi-tier. Examples include applications accessing databases, file transfer utilities, terminal emulation sessions, email access, or remote monitoring applications. These third-party or in-house developed applications are transparently secured and do not require changes to the application source code.

Together, these three products offer the following features for an efficient, cost-effective, lightweight solution for data integration needs:

- **Power and ease-of-use combined**

HiT Software's Allora and DBMoto are characterized by their ability to transfer large volumes of data efficiently between source and target using sophisticated transformations. These tools provide smaller organizations or departments of large companies with a feature-rich and highly configurable product for integration design, deployment, and management. HiT Software mapping interfaces are both easy to use and administer, so they can help increase the productivity of staff members regardless of skill sets and levels. All HiT Software tools minimize the impact of system changes and new user requirements in addition to reducing project risk. HiT Software integration products read data in legacy information systems using standard data access technologies such as JDBC, .NET, ODBC, or OLE DB on a different machine. The target system is therefore never modified.

- **Additional data security**

To respond to security concerns, HiT Software has added an additional optional level of data security by using Secure Sockets Layer (SSL) to establish secure point-to-point communication between 2 systems. SafeConduct builds an invisible, secure channel between two TCP/IP nodes, can run on any platform, and provides 256-bit data encryption. As with all HiT Software products, it is non-intrusive and does not disrupt existing applications and systems.

- **Lightweight and flexible**

Whereas most data integration products process data through a proprietary engine, Allora and DBMoto are lightweight and do not require any heavy-duty installation of a server component. Transformations can be performed interactively or called from simple APIs leveraging Java or .NET technologies, which have become long-standing standards.

- **Interoperability via industry standards**

By leveraging connectivity standards like ODBC, OLE DB, JDBC and .NET, HiT Software is able to support databases on any hardware platform with maximum scalability. With over 10 years experience in developing and deploying relational database middleware to large industry players like UPS and American Airlines, HiT Software supports over 20 database vendors for use with Allora, its XML-to-RDB transformation tool. Additionally, DBMoto, the data replication tool, can be used with most major RDBMS via .NET, OLE DB, and ODBC connections.

## **Conclusion**

For large enterprises with global operations or small companies running data operations on a couple of PC servers, data integration is an ever-changing issue that all businesses face. One week, a major supplier might request a change in business procedures that require all business partners to process XML documents. Another week, a new corporate web application might need to access data residing on a secure database server without jeopardizing the performance and reliability of the server. To respond to these demands effectively, IT personnel need to move quickly and efficiently. The optimal solution includes a suite of flexible data integration tools that operate on open standards and a willing IT crew to implement cost-effective, extensible solutions. HiT Software's Allora and DBMoto products empower all IT teams to develop reliable, scalable tools for database integration projects.