

The PDFs of *eAI Journal's* Data Integration department are sponsored by:



**Data Junction** sets the standard for data integration with award-winning technology, customer support and technical services. Right now, more than 40,000 customers invested in Data Junction technology to solve their most pressing integration concerns, making our software the most widely deployed in the world. What integration challenges are you facing? Discover how Data Junction's solutions fit into your integration world. For an on-line demonstration of the power and versatility of Data Junction, contact us at 800.580.4411 or [info@datajunction.com](mailto:info@datajunction.com)

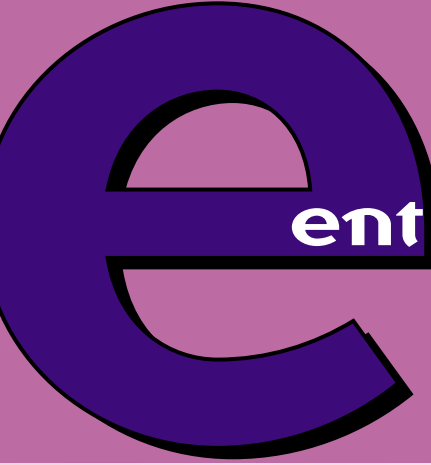


Data Junction Corporation  
5555 North Lamar Blvd.  
Ste. J-125  
Austin, TX 78751  
tel: 512-452-6105 ext. 256  
fax: 512-467-1331  
[www.datajunction.com](http://www.datajunction.com)



**This article is a PDF version of the one that appeared in a recent issue of *eAI Journal*, the leading resource for e-business, application integration, and Web services.**

Integrating the Interconnected World™



# enterprise integrity



By DAVID MCGOVERAN

## Data Integration, Part I

Most enterprise application integration (EAI) work is about making data from one application syntactically and semantically accessible to another application. Although it's not the only contributor, the high cost of integration projects is directly related to the difficulty of data integration. The level of difficulty has two primary components: data transport and data incompatibility. With widespread investment in messaging infrastructures, the data transport problem has been reduced to an all but routine engineering task.

Although we rarely give much consideration now to the causes of data incompatibility and how to solve the problem permanently, the topic consumed many gifted minds in the 1970s and '80s. Today, we concentrate on the syntactic issues, largely ignoring the semantic ones. Data incompatibility can be understood as having several causes. Beginning this month, we'll discuss those causes and some keys to ensuring that multiple data sources can be integrated with minimal effort. Usually, the causes (and solutions) can be discussed in the simplified context of two data sources.

The simplest of all causes of data incompatibility are syntactic. By syntactic aspects, we mean those portions of the problem involving format and structure. The simplest of these is field-to-field conversion. For example, the internal or stored representation of numeric data may be as a character string in one data store and as a binary number in another. Usually, conversion between representation types is pretty straightforward — as long as the data type is legitimate — although there are a few well-known problems with precision when converting most floating point representations.


Even when the error due to a single conversion is insignificant, care must be taken because these conversion errors can add up when conversions are automated. Not long ago, a client discovered that the cumulative conversion error in a particular financial transaction was costing thousands of dollars daily. The same floating point data item was repeatedly undergoing round-trip conversion between front-end and back-end systems. Remember that data transformation engines aren't immune to these problems since precision is inherent in each representation.

Record-to-record conversion introduces its own difficulties unless requirements are exceedingly simple. Rearranging a record and changing field representations is little more than a set of combined field-to-field conversions. Such changes

remain simple because they maintain field integrity. Once field integrity is broken, life becomes more complex because we now enter the realm of semantic transformations. As long as one-to-one functional translations suffice to meet the requirements, the developer can be ignorant of semantics. Any time fields are combined or decomposed to create new fields within the output record, a semantic transformation has occurred, too. We'll come back to semantic transformations and the problems they entail in a future column.

Data transformations that require records from multiple data sources necessarily blur the line between syntactic and semantic transformations. Even if the transformation can be described in terms of one-to-one functional translations, you should still justify combining multiple source records on semantic grounds. Question whether you can guarantee that the business meaning of data elements involved in the transformation will remain constant relative to each other. When the data sources are independent, this is more difficult than the data integrator might imagine and typically outside their sphere of control. We'll return to this issue and its myriad implications in the future, too.

All too often, we depend on automating the syntactic aspects of the process of data integration rather than solving the overall problem. Easy to use drag-and-drop data transformation tools and XML have greatly facilitated syntactic efforts. Many standard data sources — ranging from DBMSes to packaged applications — now generate data formatted in XML. This makes the task of specifying data transformations even easier.

However, if we consider that almost all this effort ignores both the semantics of the source and the target, it should be obvious that the simplicity is deceptive. When we trade in a deep understanding of corporate data assets for rapid delivery of data integration, the hidden cumulative costs will seriously degrade — if not eventually outweigh — any potential benefits or return on investment. The next few columns will discuss the implications of that deep understanding on your enterprise integrity. 

*David McGoveran is president of Alternative Technologies, Inc. He has more than 20 years' experience with mission-critical applications and has authored numerous technical articles on application integration. e-Mail: mcgoveran@alternativetech.com; Website: www.alternativetech.com.*