# The Last Mile Of Data Integration

A Whitepaper

Table Of Contents

XML Global Technologies, Inc. is a software developer and vendor of XML solutions. More information is available at www.xmlglobal.com.

GoXML Transform is covered by US Patent No. 5,909,570

Transform Your Information
Transform Your Business

## Executive Summary

Just as telecommunication networks can change radically over the last mile, integration solutions also differ in the last mile of the journey between systems, applications, and databases. A telecommunications network may switch from high-bandwidth fiber optics to low-bandwidth copper as it connects to its endpoints. Integration broker and EAI systems may employ similar standards and methodologies in moving information between systems and applications, but the methods that they use to map and transform data between various structures and formats – that last mile in any integration solution – vary widely.

Best practices in data mapping and transformation employ specialized transformation tools that include key elements:

- **Separation of input formatting, mapping and transformation rules, the transformation engine, and output formatting** provides flexibility, especially in dealing with multiple types of data formats.

- **Input and output formatters** are needed to handle various data formats and structures, like EDI, XML, and relational.

- **A consistent, graphical mapping tool** that presents data structures and elements in an intuitive manner.

- **A rules-based transformation engine** that will process defined mapping and transformation rules.

- **Transformation functions** that will handle common data manipulation tasks without the need for programming code.

- **Standards-based interfaces** are important to allow connectivity to various systems and databases both within and across organizational boundaries.

GoXML™ Transform, from XML Global, is one product that embraces all of these best practices and ensures smooth traveling over the last mile of data integration.

## Background

Integration platforms go a long way in making it easier to integrate data between dissimilar systems and databases. They typically provide robust communication, security, process design, and transaction monitoring components. The more progressive solutions have recently embraced XML as a standard format for transmitting data and defining metadata (definitions of data elements and structures). They have also embraced the Internet, along with its related technologies and standards, as a low-cost, ubiquitous transmission vehicle. Ultimately, these integration platforms must reach into a database, application, EDI document, flat file, or some other data store to retrieve or update information. Integration platform vendors typically supply adapters and sometimes pre-built mapping templates for popular data formats.

These solutions may get you started on this last mile, and if your needs happen to match those of the chosen vendor's prototype, all of your needs may be met, with maybe just a bit of tweaking. There will inevitably be a gap though. How wide that gap is, and how you deal with it, may very well be the deciding factor in the success of your integration solution.

There are basically three alternatives for dealing with that integration gap:

- **Live with it.** You may actually be one of the lucky ones that fit the vendor's mold, and can change your processes to deal with the occasional exceptions. You may even have enough influence with the vendor to persuade them to close the gap in a future release.

- **Fix it yourself.** Your IT department may have the skill set to write some custom code that can be injected into the process. If the perceived gap is small and the endpoints of your integration process are static, this may be a tempting quick fix. If not, this solution will leave you with a brittle system that is not ready to change as quickly as your business does – any integration solution is only as strong as its weakest link.

- **Use a transformation tool.** There are tools that are specifically designed to map and transform data from one structure to another. Some of them are based upon the XSLT standard, and they are ideally suited for situations where your last mile involves transforming data from XML to a presentation format, like HTML. If at least one of your endpoints is in a relational or EDI format,

Transform Your Information
Transform Your Business

your options are more limited. Using an XSLT-based transformation tool in that situation will probably involve XSLT coding, which is what you are trying to avoid with a transformation tool. The remainder of this paper will present you with criteria for evaluating specialized transformation tools to take you down that last mile of data integration.

## What To Look For

The two main reasons that you would consider a data transformation tool are flexibility and reliability. While performance and scalability are obviously key criteria in selecting a tool that will be part of a robust integration solution, they could conceivably also be achieved with a homegrown, custom-coded solution.

The flexibility to adapt to changing data formats, or to quickly add additional solutions as systems or trading partners are added, is a big reason to buy a transformation tool rather than build your own on a case-by-case basis.

The reliability and predictability of a robust transformation tool will allow you to accurately plan, and consistently implement, integration solutions.

The following set of evaluation criteria for transformation tools is presented with those two reasons in mind.

■ **Separation of input formatting, mapping and transformation rules, the transformation engine, and output formatting** – Such separation provides flexibility, especially in dealing with multiple types of data formats. GoXML Transform is one transformation tool that provides this separation. In fact, XML Global owns a United States Patent based on this methodology.

■ **Input and output formatters** – A transformation tool should understand different structured input data formats, such as XML, EDI (X12, EDIFACT, and extensions like HIPAA), and relational databases. Each format needs to be handled in unique ways relating to its particular structure and syntax rules. When transforming data, an input formatter parses the incoming data source, applies syntax rules consistent with its general format, and may impose structure constraints specific to the input data type. For example XML input might be validated against a DTD or XML Schema. There

is no standardized way of representing metadata for EDI structures, however a flexible tool will provide a way to define EDI metadata separately from the application code. In other words, rather than burying EDI format definitions in the compiled application code, that information should be stored in a separate dictionary or repository that can be modified or extended by the integrator if the out-of-the-box definitions supplied with the tool are not sufficient. GoXML Transform stores its EDI dictionaries in XML format for easy portability and editing. The same principles, in reverse, apply to output formatters.

■ **A consistent, graphical mapping tool** – Input and output data structures should be displayed in a uniform, graphical format. Regardless of the formats being mapped, a consistent graphical display allows integrators to intuitively define mapping and transformation rules the same way every time. A hierarchical, tree-like representation of nested structures instantly shows relationships between data elements while hiding repeating loops. A graphical mapping action that complements the structural display makes the process of mapping quite simple and intuitive. Also, the ability to hide or show various layers of detail provides both granular control and manageability – a feature that comes in handy with really large and deep structures. A sample screen from GoXML Transform illustrates the above principles.
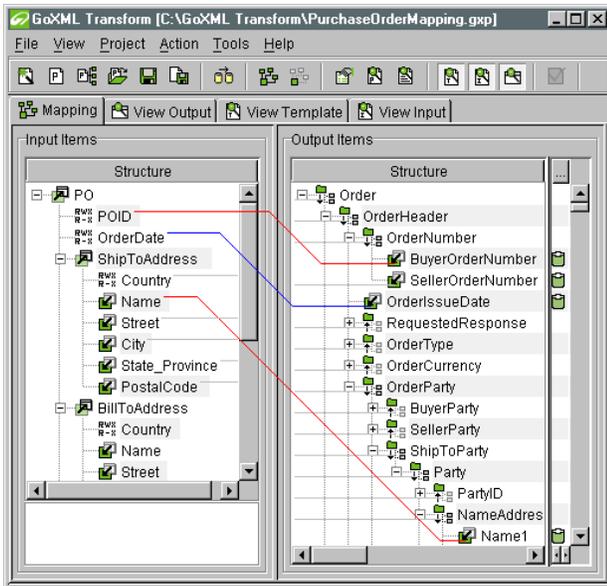
■ **A rules-based transformation engine** – It is a good idea to have a separate transformation engine that will process defined mapping and transformation rules. Decoupling the engine from the rule definitions allows the rules to be developed by one or more integrators for deployment to one or more centralized processing engines. The rules or the engine may also then be updated without affecting one another.

The features listed so far should be part of any good transformation tool, and you can use those criteria to develop your short list of candidate tools. A conceptual layout of the components described is shown here.
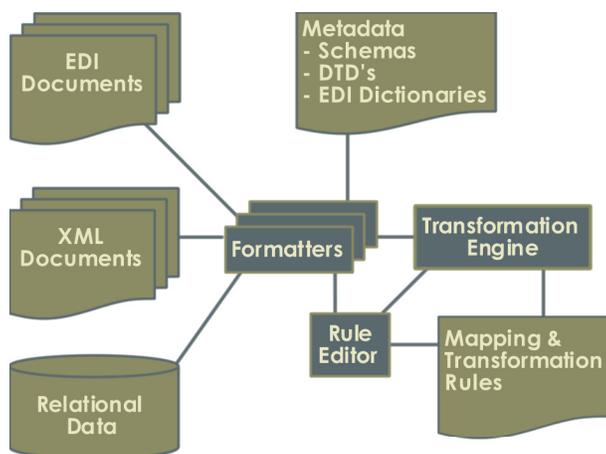


**Transformation System Components**

Other file formats, like CSV and fixed length text files, are not shown in this diagram for the sake of

clarity, but a good transformation tool should support them.

Evaluating the remaining features will help you make your final decision on which tool to choose.

■ **Transformation functions –** In addition to mapping equivalent data elements to each other, a transformation tool should allow you to manipulate the data as it is going from one format to another. Typically, one or more data elements will have to have its contents modified before it can be moved into the output structure. Often, a code in an input element will be used to look up a related description in a table, which will then be placed in the output element. Elements are sometimes combined, split apart, or supplemented with constant data. A library of functions that behave much like common spreadsheet functions should handle common data manipulation tasks without the need for programming code. A screen from GoXML Transform's function wizard illustrates this concept below.
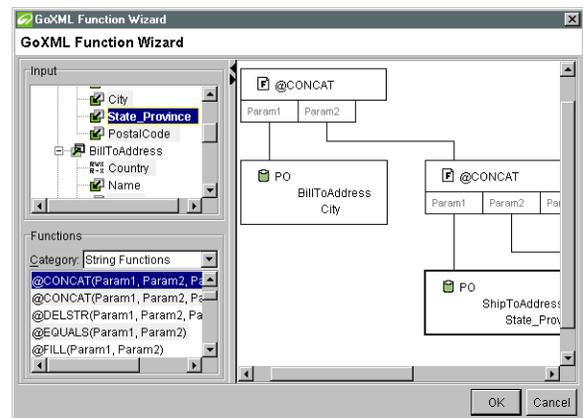


■ **Standards-based interfaces** – Remember that we started evaluating transformation tools in the context of filling a gap within a larger integration solution. Just as standards-based adapters are needed to connect the 'last mile' of copper wire to a telecommunications fiber optic backbone, you will want standards-based interfaces to embed your transformation tool into various integration platforms. Look for a tool to support Web Services, ebXML, SOAP, HTTP, and links to message queues at a

minimum. Those will give you hooks to most modern integration platforms. An SDK with a robust set of API's to the transformation engine is also a valuable feature.

## Conclusion

At XML Global, we have plenty of experience with data integration projects that adopt XML, and we have found that system integrators typically encounter a gap in the last mile of integration – where data structures must be mapped in a repeatable way between dynamic data sources. For anything other than the simplest projects, it usually pays to add a tool that specializes in mapping and transformation to your integration solution. Successful integrators follow certain best practices in transformation and data mapping. GoXML Transform, from XML Global, is one product that supports those best practices by meeting all of the criteria mentioned above, and ensures smooth traveling over the last mile of data integration.

XML GLOBAL

Transform Your Information
Transform Your Business