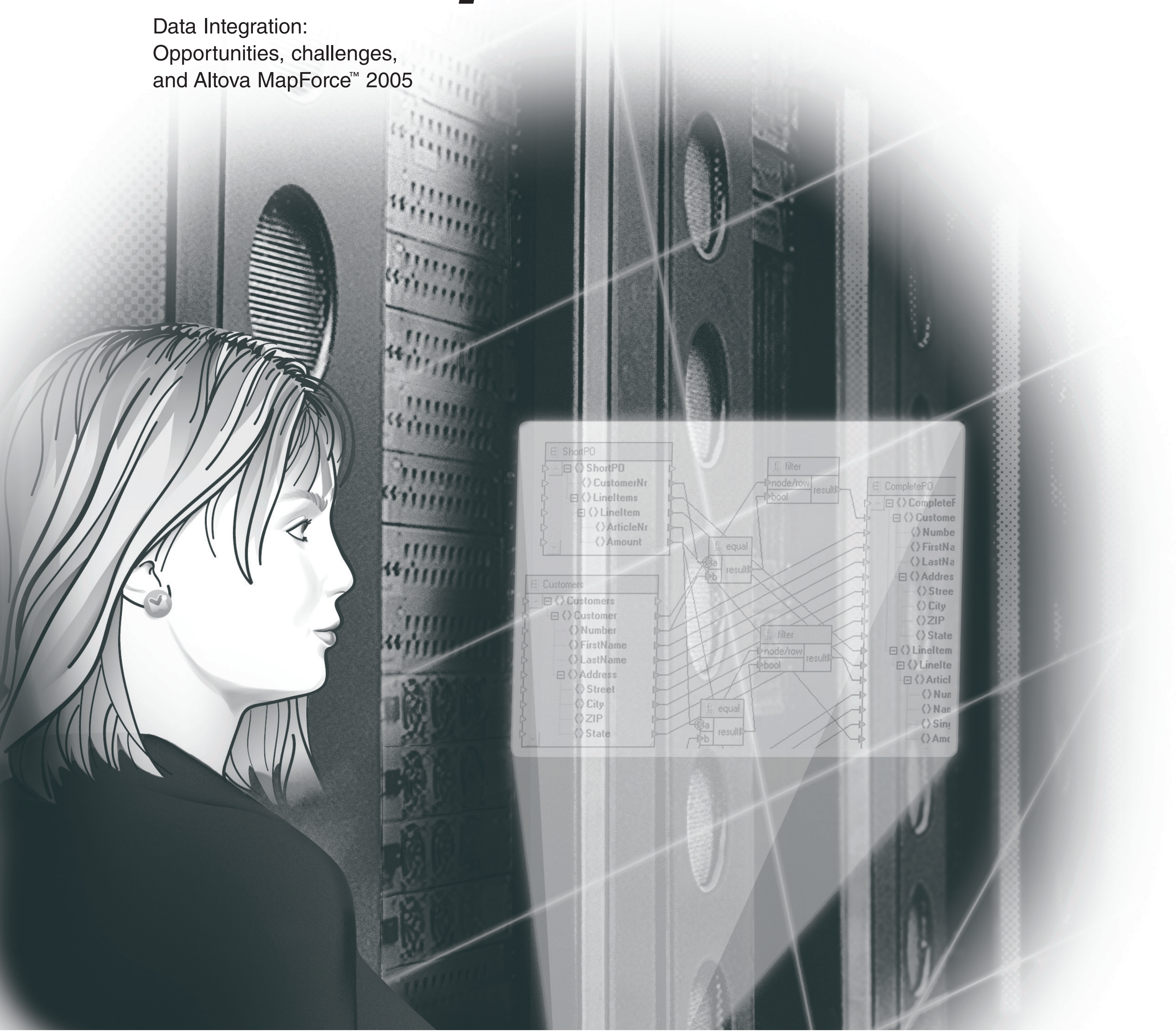
ALTOVA®

WhitePaper

Data Integration:
Opportunities, challenges,
and Altova MapForce™ 2005



Altova, Inc. • 900 Cummings Center, Suite 314-T • Beverly, MA, 01915-6181, USA • Tel: 978-816-1600 • Fax: 978-816-1606 • Web: www.altova.com

Executive Summary

The ability to seamlessly connect with customers, partners, and co-workers is vital for success; yet most enterprises store and exchange data in dissimilar formats, such as databases, EDI systems, text files, and, increasingly, XML-based applications. The ability to map between these different formats is mission-critical, and Altova MapForce™ 2005 offers a unique and powerful approach to easing the pain of systems integration. MapForce 2005 provides an easy-to-use interface for mapping any combination of XML, database, flat file, and EDI data. And, it auto-generates mapping code in XSLT 1.0, XSLT 2.0, XQuery, Java, C++, or C# for use in custom data integration applications.

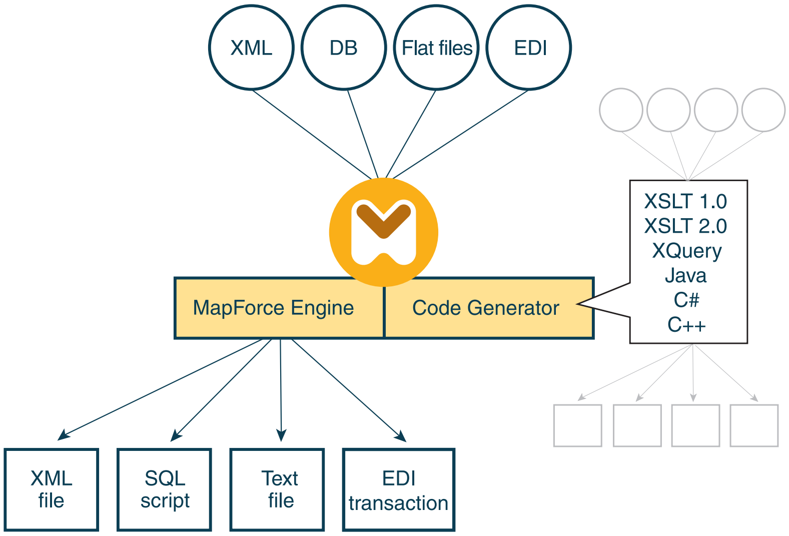


Figure 1: The MapForce 2005 data mapping model

Introduction

Data is king in today's information-driven economy. Business-to-business (B2B) e-commerce has yielded figures in the trillions of dollars and shows no signs of slowing. At the same time, organizations have realized the benefits of real-time communications technologies for unifying geographically separated offices. To remain competitive, companies must seamlessly connect with customers, suppliers, and internal business units – all of which often store and process data in different formats.

Standing behind the explosion in e-commerce are open, standards-based technologies such as XML, which promise to unify enterprise data and enable advanced Web services and Services Oriented Architectures (SOA). However, the majority of existing enterprise data is stored in business applications and database systems, which are generally neither standards-based nor readily extensible. In addition, many large enterprises employ Electronic Data Interchange (EDI) systems for exchanging business information with partners. Though these systems have proven extremely effective and enjoy wide spread use, they are often not interoperable with other systems, are complicated to develop and deploy, and may not allow real-time transactions.

The challenge is to integrate data from these various sources in a standards-based, cost effective solution that can take a company securely into the future.

To achieve effective data integration, today's businesses require applications that can access core business systems through scaleable, tool-generated code that facilitates information re-usability and interchange. The ability to efficiently map data from disparate types of data formats is necessary for a truly unified view of corporate data; however, the process is often time consuming and cost prohibitive.

Altova MapForce™ 2005 offers a unique and powerful approach to easing the pain of systems integration and enabling value-adding enterprise data services.

The Challenge – integrating multiple data formats

There are various formats for storing and exchanging data in use today, the most widely used being relational databases, flat files, Electronic Data Interchange (EDI), and, increasingly, XML.

Relational Databases

Today, the dominant storage mechanism for structured enterprise data is the relational database, which has proven itself an efficient tool for storing, searching for, and retrieving information from massive collections of data. Relational databases specialize in relating individual data records grouped by type in tables. Developers can join records together as needed using SQL (Structured Query Language) and present one or more records to end users as meaningful information.

The relational database model revolutionized enterprise data storage with its simplicity, efficiency, and cost effectiveness. Relational databases have been prevalent in large corporations since the 1980s, and they will likely remain the dominant storage mechanism for enterprise data in the foreseeable future.

Despite these strengths, relational databases lack the flexibility to seamlessly integrate with other systems, since this was not historically a requirement of the database model. In addition, although relational databases share many similarities, there are enough differences between the major commercial implementations to make developing applications to integrate multiple products difficult. Among the challenges are differences in data types, varying levels of conformance to the SQL standard, proprietary extensions to SQL, and so on.

Programmatic access to relational databases is achieved through both standard and proprietary binary protocols and system APIs including ODBC, JDBC, Net8, ADO, and others. Although some implementations are highly efficient, and many companies have made efforts to adapt to database vendors' different implementations of the most common standard, ODBC, substantial complexity remains when it comes to connectivity.

Beyond merely connecting applications to data sources, integration issues become even more problematic. Different enterprise applications store data in different ways. Names of tables and fields differ. Formats of identical data items are incompatible. Relationships between records are organized to meet completely different requirements. Certain sets of tables are optimized for fast updates, while others are optimized for fast lookups, and so on. Moreover, some applications take advantage of proprietary extensions to the SQL standard available in only one vendor's software, resulting in data models that would be impractical were the application developed for another database system.

Without question, integrating enterprise information is one of the most challenging areas of IT, even where the data is housed in the well-known relational model and even in the case where the database servers are from the same vendor. Adding to the complexity is the requirement to store data from other common sources such as EDI and XML applications in relational databases to unify enterprise information. The ability to efficiently and effectively integrate data from these various systems is a mission-critical requirement in today's enterprise.

Electronic Data Interchange

Long before the Internet made B2B electronic trade standard practice, there was EDI. Electronic Data Interchange (EDI) is a set of widely-used formats for exchanging information electronically. Though there are several EDI standards in use worldwide, the most prevalent are American National Standards Institute (ANSI) X12 and United Nations Electronic Data Interchange for Administration Commerce and Transport (UN/EDIFACT). Today X12 is the *de facto* EDI standard in the United States and much of North America, while EDIFACT is the most prevalent international EDI standard. Both standards were designed to be flexible enough to support the electronic transmission of virtually any type of data, such as purchase orders, invoices, shipping notices, medical and insurance claims, and much more.

EDI has proven effective for reducing the necessity of paper-based business processes, and the accompanying expense and error. This enables more efficient day-to-day business transactions between corporate partners and branches. In addition, electronic interaction has made the real-time supply chain possible by accelerating event notifications and the resulting transactions between large corporate partners. The use of the X12 and EDIFACT standards has allowed organizations across diverse industries to increase efficiency and productivity by exchanging large amounts of information with partners and other companies in a quick, interactive, standardized way.

When EDI technology first emerged, early adopters built systems around costly dedicated leased lines called value-added networks (VANs). More recently, though, cost-conscious companies are routing EDI traffic over common protocols using the public Internet.

However, because EDI implementations are typically infrastructure, training, and maintenance intensive, smaller companies have been less able to partner with EDI-enabled enterprises, and vice versa. This problem is particularly troublesome in rapidly changing industries where innovations often appear in less established firms.

As e-business has evolved, the number of networked businesses of every size has exploded. This Internet-driven increase in the number of prospective customers and partners, with its competitive and cost advantages, is too significant to ignore. Still, many companies have substantial existing investments in EDI technology, and those that do not already have EDI systems in place require a means to map EDI data to the newer data storage and interchange formats now receiving broad industry support.

With the increasing prevalence of commodity Internet infrastructure, there exists an opportunity to build "plug and play" applications that integrate EDI vocabularies with other common data formats. These new alternatives offer a cost effective way for organizations, regardless of their size, to effectively exchange information electronically with partner companies in a variety of data formats.

Flat Files

Another common data format is the flat file. Flat file formats are supported by a wide array of legacy systems and popular enterprise applications, such as accounting software packages, banking solutions, CRM systems, standard UNIX applications, etc., as well as by Microsoft Excel®, and are often used as an interchange format for transferring information between different applications, including databases. Flat file formats come in a variety of flavors and include fixed width, comma separated value (CSV), and tab-delimited text files.

Given the number and variety of applications in use across various industries today, the flat file format has proven invaluable in the pre-XML era for allowing users of disparate tools to share information. However, though flat files are supported by many applications, they generally require additional processing to interoperate with common data formats such as XML and EDI.

eXtensible Markup Language

XML is one of the newer technologies in the world of information interchange, but it now plays a central role in data management, transformation, and exchange. Since its introduction to industry in the late 1990s, XML has achieved widespread support and adoption among all the leading software tools, server, and database vendors. As importantly, XML has become the *lingua franca* for data by lowering the cost of processing, searching, exchanging, and re-using information.

XML provides a standardized, self-describing means for expressing information in a way that is readable by humans and easily verified, transformed, and published. This allows both information workers and automated applications to better find and use the information they need. In addition, data can be transmitted to remote services anywhere on the Internet using XML-based Web services to take advantage of the new ubiquity of connected software applications.

The openness of XML allows it to be exchanged between virtually any hardware, software, or operating system. Simply put, XML opens the door for information interchange without restriction.

XML and XML-based technologies such as XML Schema, XSL (eXtensible Stylesheet Language), WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol), etc., are all open standards that can be used in conjunction with any programming language or platform. This is to say that XML technologies can be used on and between any combination of database, application run-time, and operating system – a characteristic that's essential for integration with heterogeneous systems.

Though XML provides much of the basis for B2B information exchange over the Internet today, it was not designed to replace existing EDI, flat file, and database technologies. Rather, the flexibility and openness of XML allow it to co-exist with and complement non-XML data formats and technologies.

Because of its built-in interoperability, XML gives organizations the ability to leverage existing investments while adopting a powerful new technology, and, at the same time, it actually increases the usefulness of those existing systems by adding the flexibility required for real-time data exchange across departmental and geographic boundaries and through system and programmatic constraints.

However, XML in and of itself is not a data integration panacea. Successfully integrating XML with other data formats requires the development of applications that integrate system interfaces and map between data structures.

Developing Data Integration Applications

Data integration frameworks and applications offer the potential to unify business data while capitalizing on the particular strengths of relational database, EDI, flat file, and XML systems that have made them staples of modern business.

Businesses have the option of choosing a middleware or server-based data integration platform, but these solutions are proprietary and often extremely expensive. In addition, off-the-shelf point products offer a quick solution for very specialized integration scenarios but lack the flexibility necessitated by the pace at which business and technology requirements evolve today. A third and more viable option is building customized data integration applications that are cost effective, flexible enough to adapt to changes, and don't force businesses to be locked in to a particular software vendor.

Depending on an organization's specific needs, customized applications can be created to, for example:

- Store XML and EDI data in a relational database
- Convert database, flat file, and EDI data to XML for use in advanced Web services or content management applications
- Convert CSV data to an EDI format (or vice versa) to allow EDI systems to interoperate with popular spreadsheet and accounting applications
- Transform an XML file to conform to another XML format to enable an e-business transaction
- Convert database data to a flat file format for use in a business application
- Process data from one or more databases to conform to the structure or naming conventions of one or more other databases
- And countless other scenarios

Despite their advantages over other solutions, customized data integration applications are often extremely complex, and developing them can be complicated, expensive, and time consuming. For an application to be effective, developers must understand the underlying theory and implementation intricacies of each data format, which is no small feat. Moreover, the different runtime characteristics of the primary business applications the data stores support and the data model design choices that make those applications effective play a considerable role in defining the integration opportunities and constraints.

An enormous amount of enterprise application code is tied up in data access, verification, and management. Data processing functions are required to transform source data to the target format. Many services-oriented applications require multi-step mappings that merge multiple structures into a logical model. Writing the hundreds or thousands of lines of lines of infrastructure code required to procedurally perform intricate operations or merge the XML payloads of multiple Web services before updating one or more relational databases, for example, is not merely tedious – it is cost prohibitive, inflexible, and error prone. Further, no matter the shape the job market, few firms will find it easy to justify, attract and retain enough of the highly skilled developers who could routinely take on such challenges.

Developers need an affordable tool that simplifies away as much complexity as possible and enables them to easily define mappings that account for the unique aspects of each integration scenario. Only that increase in productivity can allow an enterprise data integration project to achieve the kind of success business demands.

The Solution – Altova® MapForce™ 2005

Altova, a pioneer and leader in XML-based technologies and tools, extended its focus to include data integration tools with the release of its MapForce data mapping utility in 2003. Since then, MapForce has received awards and rave reviews from developers working to integrate not just XML, but database, flat file, and EDI data, as well.

MapForce 2005 allows you to define mappings for any combination of XML, database, flat file, and EDI data, with support for mapping:

- XML to XML, databases, flat files, and EDI
- Databases to XML, databases, flat files, and EDI
- Flat files to XML, databases, flat files, and EDI
- EDI to XML, databases, flat files, and EDI

Mapping is not limited to one-to-one relationships - MapForce 2005 allows you to mix multiple sources and multiple targets to map any combination of different data sources in a mixed environment.

MapForce 2005 abstracts away the complexity of data mapping by allowing you to create customized mappings visually. When you load two or more data structures into the design window, MapForce 2005 represents their hierarchical structure visually, and you map between them by dragging connecting lines between matching elements in the source(s) and target(s) and inserting data processing rules. Then, with the click of a button, MapForce 2005 autogenerates code in XSLT 1.0, XSLT 2.0, XQuery, Java, C++, or C# for use in a customized data integration application. In addition to generating code to implement a mapping, the built-in MapForce engine allows you to execute a mapping to view and save the output at any time. This feature is especially useful for one-time data transformations.

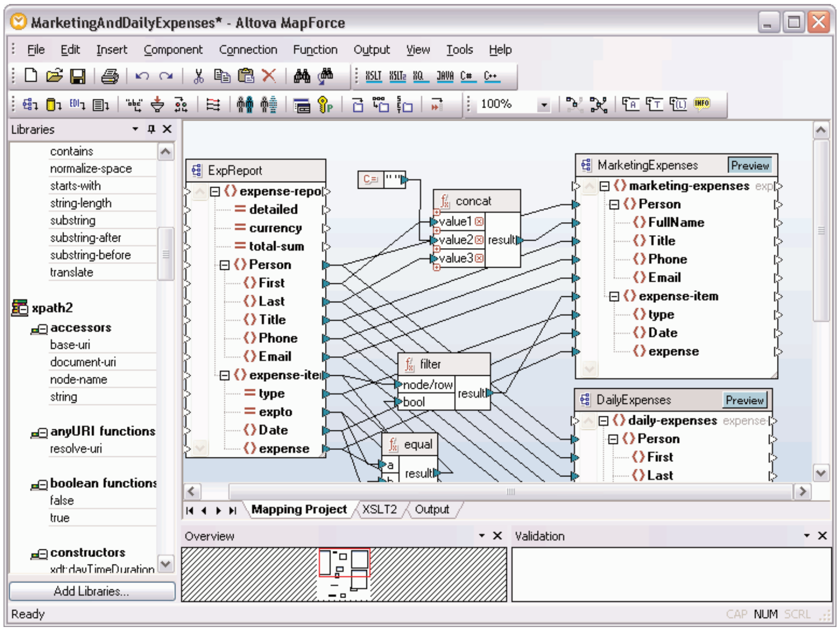


Figure 2: Intuitive MapForce™ 2005 User Interface

Because you design mappings with a visual interface, you don't have to understand the specific details of how to programmatically access the data formats being mapped. The advantages of this aspect of the tool cannot be overstated – what used to require a team of highly-skilled and specialized XML, flat file, database, and EDI experts can now be accomplished quickly, easily, and error-free using MapForce 2005.

As you are designing a mapping, MapForce 2005 provides a comprehensive library for building advanced data processing functions to perform any type of computational operation on data to make it adhere to the content model of the target. The functions allow you to specify mapping rules based on conditions, Boolean logic, string operations, mathematical computations, or any user-defined function. Many built-in functions support an unlimited number of parameters, allowing you to perform complex mathematical or concatenation functions with ease.

MapForce 2005 supports advanced multi-stage data processing logic, meaning you can pass the output of one function into the input of another function, chaining them together as required, before completing the data transformation. In Figure 1, data processing functions are represented by the boxes joining the connecting lines between the sources and target. Instead of writing and testing all the code required to make these data manipulations, MapForce 2005 lets you specify data processing rules using a visual interface, then performs reliable code generation for you.

In addition, MapForce 2005 includes a powerful visual function builder for defining and saving complex functions. Performing multi-stage data processing operations can be somewhat complicated and time consuming if you have to recreate it from scratch each time you require the same result. With the visual function builder, you can combine the various functions required to perform a particular data manipulation into one function, and then save it for use in all future integration projects. Defining functions this way allows the efficient re-use of functions and saves a considerable amount of time by allowing you to automate repetitive tasks. The visual function builder all but eliminates the need to write functions in other programming languages.

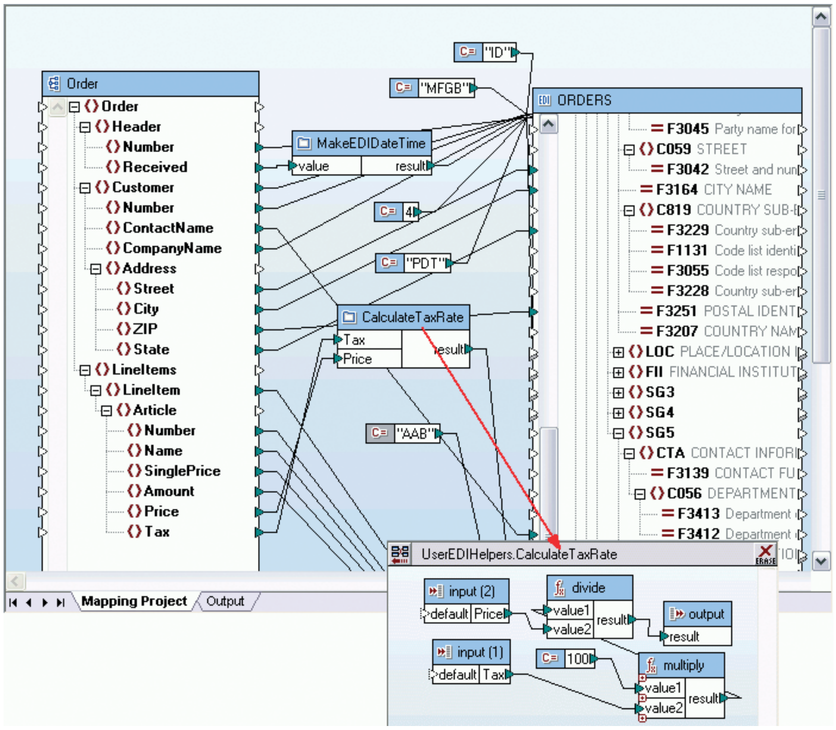


Figure 3: MapForce 2005 visual function builder

In addition, when using MapForce™ 2005 to generate code for a data integration application, you can design mappings that utilize transformation input parameters. This allows the input of outside parameters used in your mapping transformations at runtime, so that transformation results can be influenced by the calling program.

After your mapping design and data processing rules are complete, MapForce 2005 auto-generates the code required to implement the mapping. MapForce 2005 ensures that data transformation code is written consistently across an entire integration project, because it's auto-generated according to industry standards and globally defined parameters, rather than having multiple engineers manually implement the code. This approach to data integration ensures compatibility and interoperability across different platforms, servers, programming languages, and database environments, removing the limitations imposed by proprietary data integration solutions. In addition, code consistency helps reduce and isolate software bugs while improving overall code readability and reusability.

XML-to-XML Mapping

With its powerful XML integration capabilities, MapForce 2005 allows XML-to-XML mapping, and it also supports XML as the target or source of any mapping involving EDI, flat file, or database data.

For data integration projects that require XML-to-XML mappings, MapForce 2005 can easily transform data from one XML Schema to any other XML Schema. For example, a common requirement is transforming one company's XML-based purchase order (PO) to correspond to a different company's order processing system to enable online e-commerce transactions. To make this transformation using MapForce 2005, you simply open the XML Schema for each company's purchase order and drag connecting lines between corresponding elements of the source and target. MapForce 2005 supports mixed content mapping, allowing you to map between elements that contain text and sub-elements, and it includes an option to automatically connect child elements as the mapping is designed. Mapping between multiple source and target schemas is supported as well.

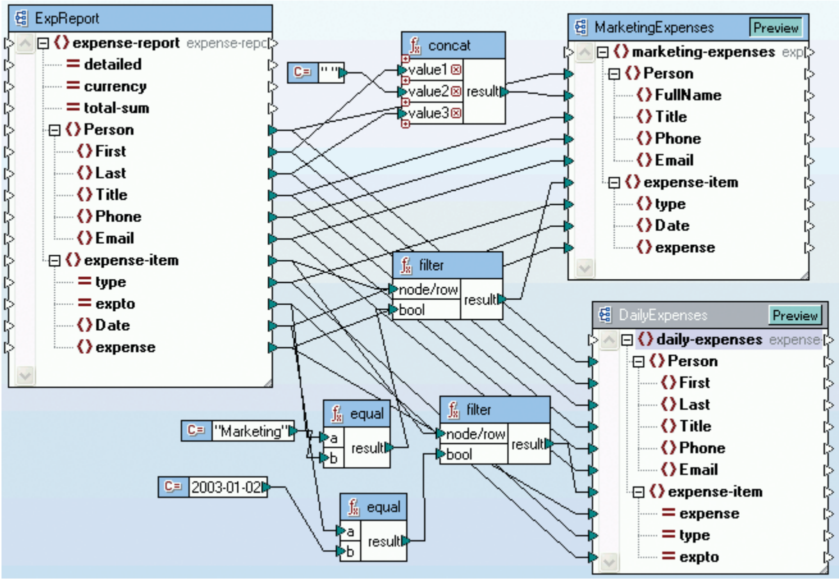


Figure 4: Designing an XML-to-XML mapping

For XML-to-XML transformations, MapForce 2005 supports the ability to import existing XSLT 1.0/2.0 function libraries to use in addition to the libraries included with the product. This allows you to leverage work you've already completed and also take advantage of other industry-standard XSLT libraries for defining data transformation functions.

Once your design is complete, MapForce 2005 can auto-generate XSLT 1.0/2.0 or XQuery transformation code, or Java/C++/C# program code to implement the mapping.

XSLT and XQuery Transformation Code Generation

In addition to generating XSLT 1.0 stylesheets for XML-to-XML mappings, MapForce 2005 includes the industry's first production-grade implementation of the important new XSLT 2.0 specification. The 2.0 specification represents a significant upgrade compared to version 1.0, extending the language with features that increase productivity, improve code quality and reusability, and provide powerful new functions. The new features in XSLT 2.0 are designed to make transformation stylesheets more efficient and more powerful, allowing you to produce higher-quality applications in less time. The option of generating XSLT 2.0 stylesheets with MapForce 2005 lets you harness the powerful new data processing capabilities of XSLT 2.0 in your applications today.

When you choose XSLT 1.0 or XSLT 2.0 code generation, MapForce 2005 generates the appropriate XSLT stylesheet behind the scenes. At any time, you can preview and save the stylesheet by clicking on the XSLT tab at the bottom of the main design window. To facilitate testing your mapping code, you can associate a sample XML instance document with the XML Schema to view the results of a transformation. This feature is also useful for one-time transformations, as you can design a mapping and save the output very easily.

MapForce 2005 also allows you to perform XML-to-XML mappings with XQuery. The ability to generate XQuery code allows you work with XQuery in a native implementation, with MapForce 2005 acting as a visual XQuery builder. You can view the XQuery generated by MapForce 2005 in the XQuery tab of the design window. Once generated, this code can be executed in an XML-capable database, or you can view the results of the XQuery using the output preview feature enabled by the built-in MapForce Engine.

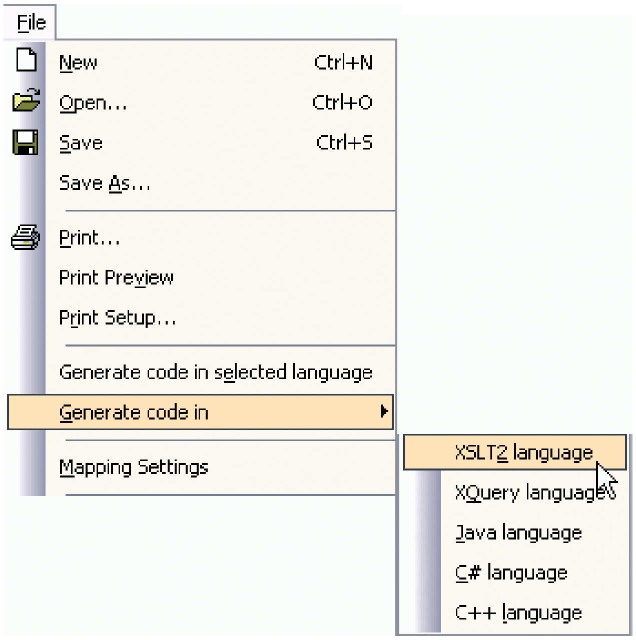


Figure 5: Generating transformation or program code

Program Code Generation

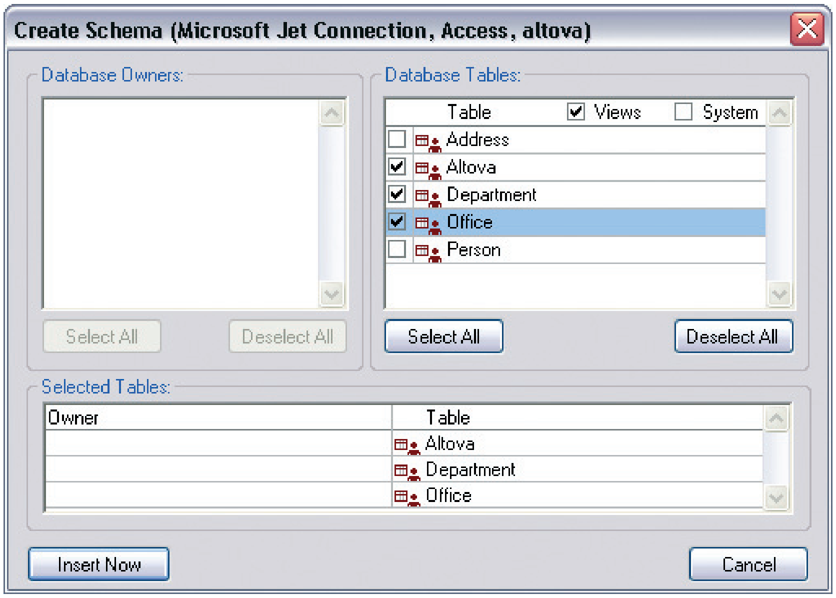
You can also use MapForce 2005 to auto-generate software program code in Java, C++, or C#. The ability to generate program code in various programming languages can provide tremendous performance benefits in applications that require XML transformations. Instead of relying only on standard XSLT processors, in which transformations are processed in an interpretive way, MapForce 2005 allows you to produce lightning-fast transformations by providing program code that can be compiled into your applications. Compiled transformation code will consistently out-perform any standard XSLT processor approach.

As with XSLT and XQuery generation, you can view and save the output of any mapping executed in Java, C++, or C# by clicking the output tab.

Database Mappings

Due to the nature of the relational database model and variations between different products and implementations, integrating database data with XML, flat files, EDI, and other databases can present some of the most challenging aspects of a data integration project. MapForce 2005 removes these challenges, allowing you to develop mappings to and from relational databases using the same user-friendly visual design process described above. MapForce 2005 supports the ability to map to and from all major databases, including Microsoft® Access®, SQL Server, Oracle®, MySQL®, Sybase®, and IBM DB2®. Other databases are also supported using ADO or ODBC connectivity.

When you load a database structure in the design window, MapForce 2005 automatically interprets the database schema, lets you pick available database tables and views, and recognizes table relationships. Once you have confirmed your selection, MapForce 2005 displays all chosen top-level and related tables in a hierarchical tree structure. As with XML-to-XML scenarios, designing mappings that involve databases is as easy as dragging connecting lines between data structures and inserting data processing functions to filter and process the data as needed.



Create Schema (Microsoft Jet Connection, Access, altova)

Database Tables:

	Table	☑ Views	□ System
□	Address		
☑	Altova		
☑	Department		
☑	Office		
□	Person		

Selected Tables:

Owner	Table
	Altova
	Department
	Office

Figure 6: Loading a database in MapForce

When a database is the target of a mapping, MapForce 2005 allows you to control how data is written to the database. An easy-to-use Database Table Actions dialog box allows you to define the columns within the selected table to determine what action (e.g., INSERT, UPDATE, DELETE) should be executed in the database. This provides unprecedented flexibility in manipulating database rows in response to XML, flat file, EDI, or other database data.

The MapForce 2005 database key settings allow you to customize how primary and foreign key values will be added to a database. You can either provide values for keys from within MapForce 2005, or you can let the database system handle the generation of auto-values.

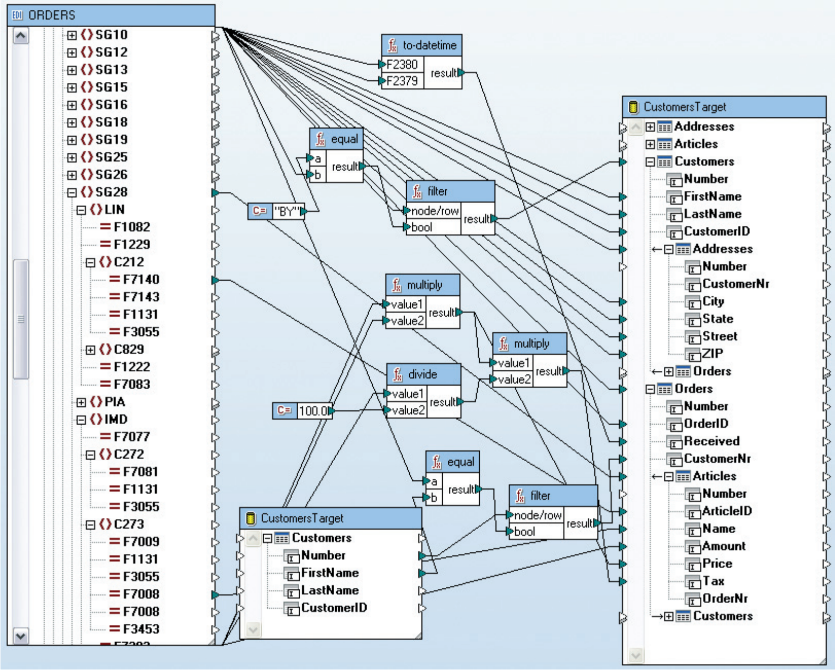


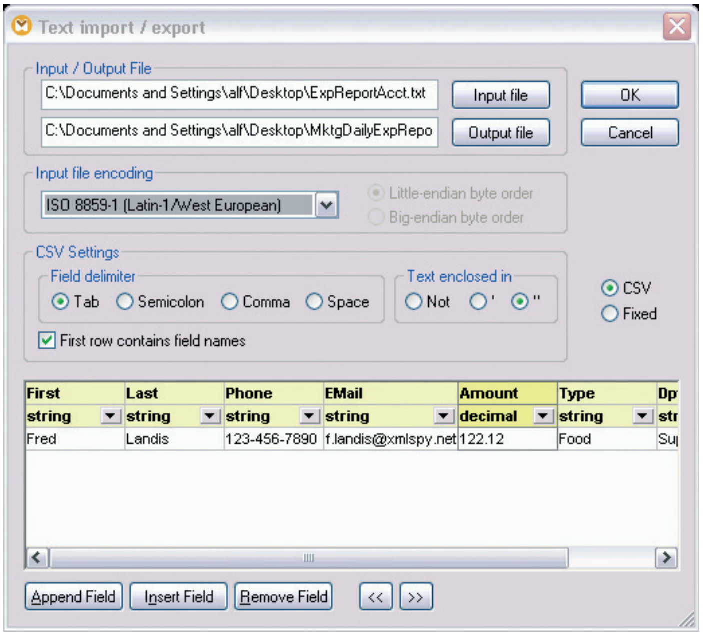
Figure 7: A mapping using database and EDI sources

Once you have finished defining the data mappings and data manipulations, MapForce™ 2005 auto-generates Java, C++, or C# code required for your application. Again, this code is compatible and interoperable across different platforms, servers, program languages, and database environments.

The built-in MapForce Engine provides an instant output preview. Mappings to a target XML Schema produce an XML output document, mappings to flat files have output in CSV or fixed-length text files, and mappings to EDI produce either EDIFACT messages or X12 transaction sets, depending upon which standard you are working with. Mappings to a database produce output in the form of SQL scripts (SELECT, INSERT, UPDATE and DELETE statements), which can be run against your target database directly from within MapForce 2005.

Flat File Mapping

MapForce 2005 supports flat files as the source or target of a mapping. When you insert a file in the design window, MapForce 2005 presents you with a dialog box that allows you to specify whether the file is CSV or fixed length, then select the respective settings associated with the file. The data in the document is displayed in a window, and you can append, insert, and remove fields as well as change field header names and values as required before importing the file.



Text import / export

Input / Output File

C:\Documents and Settings\alf\Desktop\ExpReportAcct.txt

C:\Documents and Settings\alf\Desktop\MktgDailyExpRepo

Input file encoding: ISO 8859-1 (Latin-1/West European)

○ Little-endian byte order
○ Big-endian byte order

CSV Settings

Field delimiter: ● Tab ○ Semicolon ○ Comma ○ Space

Text enclosed in: ○ Not ○ ' ● "

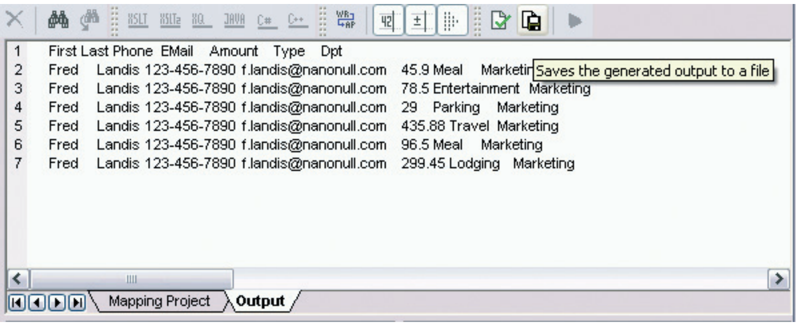
● CSV ○ Fixed

☑ First row contains field names

First	Last	Phone	EMail	Amount	Type	Dp
string	string	string	string	decimal	string	str
Fred	Landis	123-456-7890	f.landis@xmlspy.net	122.12	Food	Su

Figure 8: Inserting a CSV file in MapForce

As you design your mapping, the MapForce Engine allows you to preview the output to track your progress.



```
First Last Phone  EMail    Amount   Type    Dpt
Fred    Landis 123-456-7890 f.landis@nanonull.com   45.9 Meal     Marketin
Fred    Landis 123-456-7890 f.landis@nanonull.com   78.5 Entertainment  Marketing
Fred    Landis 123-456-7890 f.landis@nanonull.com   29   Parking    Marketing
Fred    Landis 123-456-7890 f.landis@nanonull.com   435.88 Travel  Marketing
Fred    Landis 123-456-7890 f.landis@nanonull.com   96.5 Meal     Marketing
Fred    Landis 123-456-7890 f.landis@nanonull.com   299.45 Lodging   Marketing
```

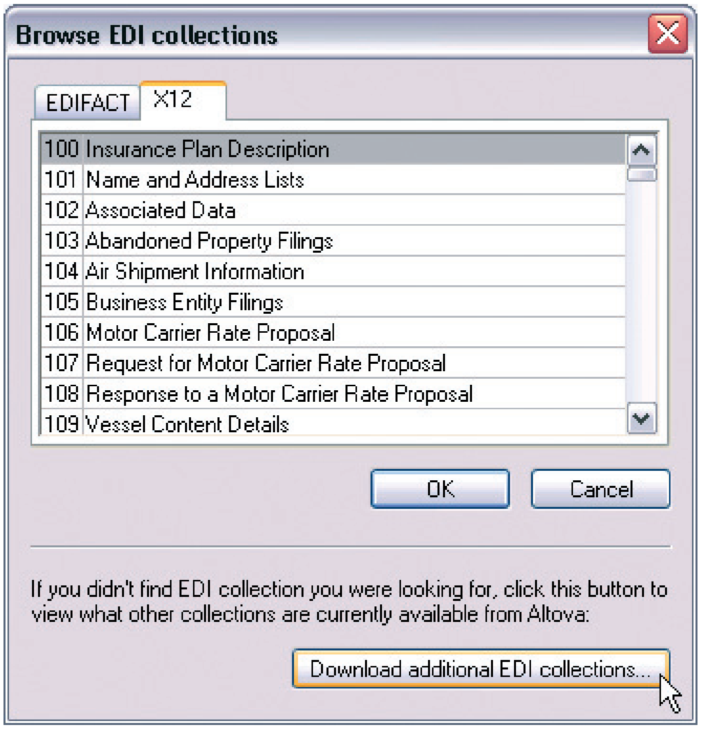
Saves the generated output to a file

Figure 9: Previewing output to a flat file

EDI Mapping

MapForce 2005 supports EDI as the source and/or target of any mapping, allowing you to map EDI messages to XML, databases, flat files, and other EDI formats.

MapForce 2005 ships with all the transaction sets specified in ANSI X12 and the hundreds of messages covered by the most recent UN/EDIFACT standard. All the EDIFACT standards for previous years (back to 1993) are also available for free download so that you can customize MapForce to work with the standard your organization supports. Built-in support of these messages and their underlying cross references is crucial to practical EDI data mapping.



100	Insurance Plan Description
101	Name and Address Lists
102	Associated Data
103	Abandoned Property Filings
104	Air Shipment Information
105	Business Entity Filings
106	Motor Carrier Rate Proposal
107	Request for Motor Carrier Rate Proposal
108	Response to a Motor Carrier Rate Proposal
109	Vessel Content Details

If you didn't find EDI collection you were looking for, click this button to view what other collections are currently available from Altova:

Figure 10: EDIFACT and X12 EDI messages

The unique capability of MapForce 2005 to easily integrate EDI data allows organizations to leverage their investments in EDI technology while incorporating EDI data with other common formats. In addition, EDI integration allows smaller companies to effectively exchange information with partners that employ EDI systems. The ability to open the previously inoperable EDI model to Internet based e-commerce and relational database stores allows businesses of any size to reap the benefits of real-time information exchange.

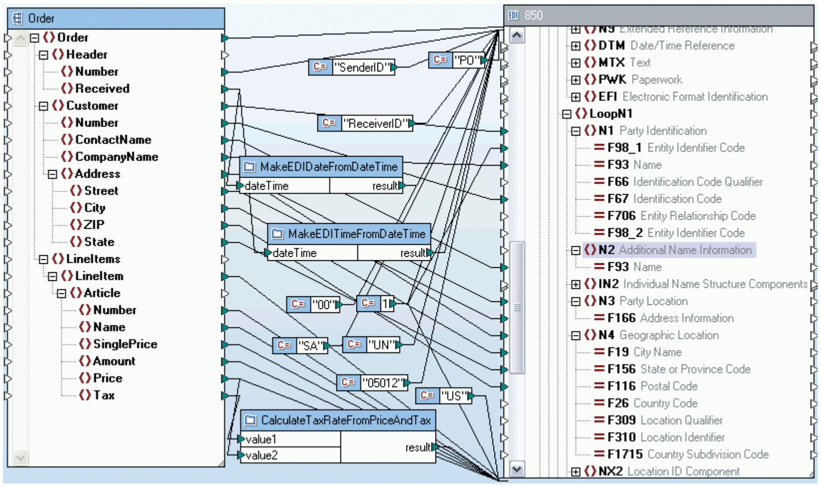


Figure 11: Mapping to EDI X12

MapForce 2005 for Advanced Integration

The unique MapForce 2005 approach to data integration addresses the challenges commonly associated with developing customized integration applications. Its intuitive, visual design paradigm simplifies mapping to the extent that you no longer have to understand the intricacies of XML, flat files, EDI, and the slew of relational databases in use today. You can focus on the business logic of your application and leave the generation of low-level infrastructure code to MapForce 2005. With the click of a button, it auto-generates XSLT 1.0, XSLT 2.0, XQuery, Java, C++, or C# code, and you can preview the output of any transformation directly in MapForce 2005.

MapForce 2005 is the first retail-priced, general purpose data integration tool to support all major relational databases and programming languages, and it provides an easy, cost effective way to develop and integrate database, XML, flat file, and EDI technologies. Using MapForce 2005 simplifies and accelerates the development of data integration applications to make information leverage a reality.

Conclusion

The value of an organization's business information is directly proportional to its ability to share that information internally and externally. It's no longer acceptable for the data storage and exchange systems to operate in a vacuum.

Organizations large and small are realizing the benefits and necessity of integrating business data from disparate sources. Open, standards-based technologies such as XML have achieved widespread adoption as much for their flexibility to represent and exchange data as for their power to create the applications of the future through integration with legacy and database systems.

Altova MapForce 2005 is a robust tool for data integration that simplifies and accelerates software development. It's the only data mapping product that allows you to map any combination of XML, database, flat file, and EDI data, making it the most comprehensive, easy-to-use, value-priced data mapping tool on the market today.

For more technical details about the features of MapForce 2005,
visit the Altova MapForce 2005 product pages at
www.altova.com/products/mapforce/data_mapping.html
and the MapForce 2005 Feature Matrix at www.altova.com/matrix_m.html.

You can download a free 30-day trial of MapForce 2005 at
www.altova.com/download.html.

About Altova

Altova accelerates development and integration projects with software, services, and solutions that enhance productivity and maximize results. As an innovative, customer-focused company and the creator of XMLSpy, MapForce, and other leading software development and data integration tools, Altova is the choice of over 1.5 million clients worldwide and virtually every Fortune 500 company. With customers ranging from vast development teams in the world's largest organizations to progressive one-person shops, Altova's line of software applications and custom-tailored professional and educational services fulfills a broad spectrum of business needs. Altova is an active member of the World Wide Web Consortium (W3C) and is committed to delivering standards-based platform-independent solutions that are powerful, affordable, and easy to use. Altova was founded in 1992 and has headquarters in Beverly, Massachusetts and Vienna, Austria.

Visit Altova on the Web at http://www.altova.com.

Resources

"A Focus on XSLT 2.0: Understanding the Development and Business Benefits." October 2004. InfoTrends/CAP Ventures Analysis Report (http://www.capv.com/home/Downloads/10.22.04.pdf).

Fitzgerald, Michael. 2001. Building B2B Applications with XML. New York: John Wiley and Sons Publishing.

Goldfarb, Charles F., Prescod, Paul. 2002. Charles F. Goldfarb's XML Handbook, 4th Edition. Upper Saddle River, NJ: Prentice Hall PTR.

Hendrickson, Jim. 2003. "Exploding Myths and Examining Realities of E-Business in the Internet Era." Sterling Commerce white paper.

Hollander, Dave. 2001. "Re-wiring EDI: XML Makes New Connections." ADTmag.com.

Hunter, David, et al. 2000. Beginning XML. Birmingham, UK: Wrox Press Ltd.

Simon, Solomon H. 2001. Emerging Business Technology Series: XML. New York: McGraw Hill.