Microsoft .NET Framework | Microsoft Visual Studio .net

Microsoft .NET Customer Solution

Software Labs

Data Integration Solution Boasts Ease of Use

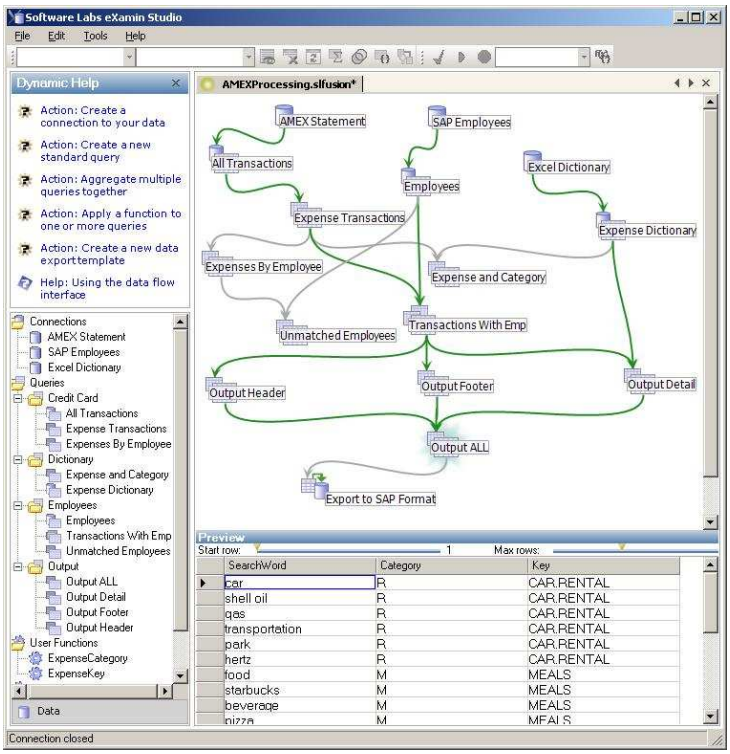
Published: January 2005

Software Labs develops advanced data management solutions. When designing eXamin Fusion, Software Labs considered Java, but chose to build the product with C#, Windows Forms, Microsoft ADO.NET, and the Microsoft .NET Framework. Windows Forms made it easy to give the product a rich graphical user interface, and ADO.NET made it easy to write connectors for a wide variety of databases.

Data integration is an issue for almost any business. As soon as data is stored in two places, especially if the formats are different, it's a good bet that someone will need to combine the data for purposes unforeseen by the original data collectors, and possibly for purposes not even related to the original applications.

Software Labs develops advanced data management solutions for companies that need to improve business process efficiencies and business intelligence. Its focus includes data integration, migration, and process automation.

Software Labs wanted to develop a new product for managing data between systems that cannot natively share information. The company considered two major development environment options, the Microsoft® .NET Framework and Java.



Java's advantage is that it works across many platforms, but Software Labs judged that unimportant for this product. "There's little need for a rich client product to run cross-platform," says Pradeep Tapadiya, President and CEO of Software Labs, and lead developer of eXamin Fusion. "[Microsoft] Windows® is the standard client platform for our customers because of the applications: people run Microsoft Office to get their jobs done."

Fast Facts	
Number of developers to build application	5
Number of months to build application	18
Number of modules	50

Software Labs decided to build the eXamin Fusion product with Microsoft Visual C#® .NET development tool, using Windows Forms and Microsoft ADO.NET. The development team used Microsoft Visual SourceSafe® version control system for source code

Software Labs

Solution Overview

Customer Profile

Software Labs develops advanced data management solutions for companies who need to improve business process efficiencies and business intelligence. Its focus includes data integration, migration, and process automation.

Business Situation

Software Labs wanted to develop a new product for managing data between systems that cannot natively share information.

Solution

After considering Java, Software Labs decided to build the product with Microsoft® Visual C#® .NET, using Windows® Forms and Microsoft ADO.NET.

Benefits

- The Microsoft .NET Framework classes reduced the need to write libraries.
- Windows Forms made it easy to develop a rich graphical user interface with good ease of use and ease of learning.
- ADO.NET made it easy to develop connectors for a wide variety of databases.

Software and Services

Microsoft Windows XP
Microsoft SQL Server™ 2000
Microsoft Visual Studio® .NET 2003
- Visual C# .NET

Microsoft Visual SourceSafe
Windows Forms
ADO.NET

management, NUnit as a unit testing framework, and BugTracker to manage software defect tracking. After initial design, the team went into a spiral mode of development based on extreme programming. The design work was done by all 5 developers.

The Software Labs team built a core system to provide data management in the form of access, integration, migration, and export. This solution was designed so that additional data sources can be supported through the creation of new data connectors. The functionality after data connection is common among all the data connectors.

eXamin Fusion has four major parts:

- Creating a connection to data sources
- Getting data from the data sources
- Transforming data
- Exporting the data or preparing it for integration with other applications.

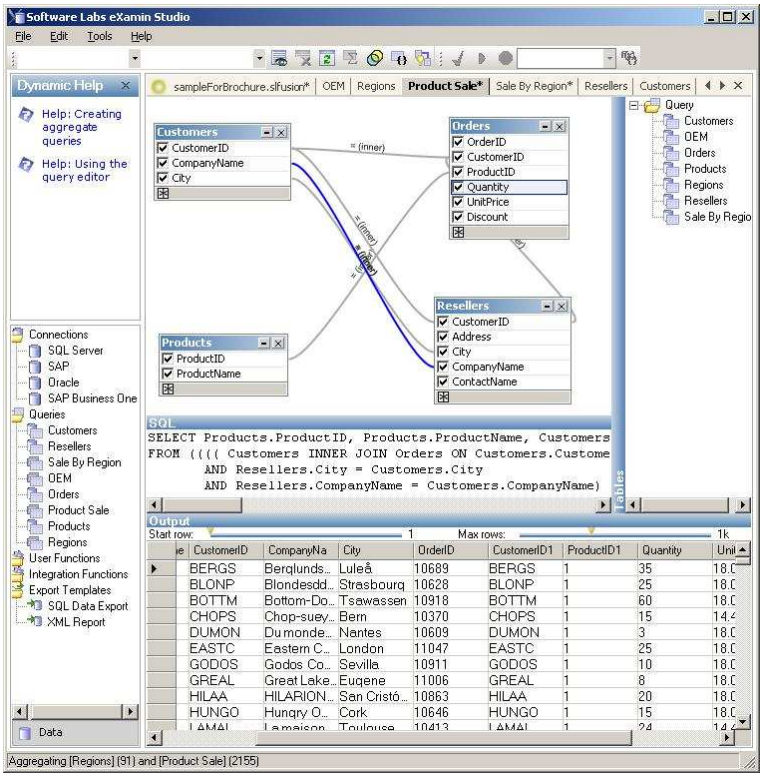
On the "creating a connection" level, eXamin Fusion offers an open internal connection interface. Software Labs has defined a set of interfaces for adding future data connectors. The interfaces include metadata, tables, and columns. eXamin Fusion supports OLE DB-compliant applications as data connectors.

The core product does not distinguish between one connector and another from a functionality standpoint, but most of the product's internal data connectors use ADO.NET. At the "getting data" level the product offers two standard query types: relational and advanced.

At the "transform data" level the product provides aggregate queries and transform queries. Aggregate queries combine data from multiple sources. Transform queries are the product's most powerful feature and provide flexibility through an expression language. Over 100 functions are available, and users can define their own functions in C#, Microsoft Visual Basic® .NET development system or Microsoft JScript® .NET development software. These functions can even be used to fetch data from a Web service.

On the "exporting" level, eXamin Fusion has an export interface via each data connector. Exporting is wizard-driven to map tables, columns, and so on There is also an integration function for more complex needs and users can write their own integration functions.

"Development in Visual C# is amazing," says Tapadiya. "If we hadn't used the .NET Framework, I



don't know how long it would have taken us. We are able to write functions quickly, test them with NUnit, and then check them into Visual SourceSafe."

Test-first programming doesn't eliminate debugging, however. "You always want to see what's going on down in your code," says Tapadiya. "We try to use exceptions only when they are needed. The [Microsoft] Visual Studio® debugger is terrific for tracking down and fixing the source of the exceptions thrown."

Tapadiya and his team used Windows Forms to quickly implement a rich, graphical user interface, shown in the two screen shots above, which gave the product good ease of use and ease of learning. Tapadiya praises Windows Forms, especially when coupled with the Visual Studio .NET development system forms designer, for "RAD, visual design, ease of use, and the simplicity of the model."

The team wrote their code in C#. "We had C++ backgrounds, and we all felt right at home in C#," says Tapadiya. "In addition, it is simple to understand and maintain C# code."

Tapadiya says he "could talk all day" about ADO.NET. "In previous database access technologies, like ODBC and OLE DB, client programs could be tough to write," he explains. "ADO.NET is fantastic, because it simplifies the model and reduces the effort needed to write database code. ADO.NET saved us a *lot* of time when we were writing our connectors."