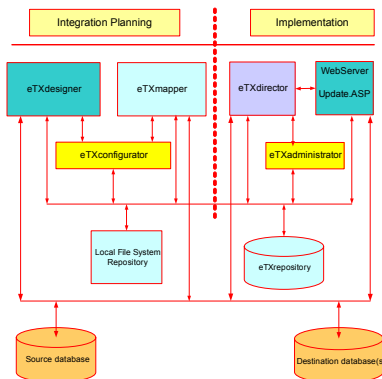


## Introduction

Many organizations today are faced with the complex task of integrating front-end applications such as customer-relationship management (CRM) software, order entry systems and corporate portals with back-end systems such as enterprise resource planning (ERP) and other custom, homegrown applications. In many cases, the common objective is to provide management with real-time views of business processes, better interaction with business partners and new avenues for increased efficiencies. For these organizations, there is an immediate need for Enterprise Application Integration (EAI) tools to facilitate this process. EAI tools offer the ability pass data between multiple legacy and enterprise systems through a single messaging hub, rather than building countless custom interfaces to link each system. The most significant benefits of using EAI tools are speed, cost savings and flexibility. The eTXIntegration Suite provides quick, simple and robust integration between various applications – all at an affordable price. In essence, it allows organizations to “Do More with Less and Do it Fast!”

## Architecture Overview

The eTXIntegration Suite is a standards-based, components oriented EAI tool that provides an easy to use, yet highly effective method of integrating two or more applications and their underlying databases. It consists of a suite of components that are designed to guide the user from the planning phase all the way to the final execution of the application integration.



**Figure 1: eTXIntegration Suite Architecture**

These components allow users to design, map and integrate transactions between disparate systems in real time. The eTXIntegration Suite automatically generates XML Schema Documents (XSD) and XML Style Sheet Language Transformation (XSLT), eliminating the errors and frustrations typically associated with manual generation. It uses World Wide Web Consortium (W3C) standards for its XSD and XSLT formats and has a repository to keep track of the Meta data on the transactions for audit purposes. The repository can be SQL Server, DB2 UDB, Oracle or any other SQL-compliant database. By maintaining the Meta data for the transactions, users can easily change the transaction design and the conversion associated with each transaction, even as the business requirements change. This is an important aspect of the system’s simplicity, as it empowers users to perform a variety of integration functions effortlessly. The system also uses Open Database Connectivity (ODBC), HTTP and HTTPS as its main communication and access technology for databases and Web servers. Figure 1 shows the architecture for the eTXIntegration Suite.

## Integration Process

The integration process consists of four easy steps: design, map, configure and integrate. The “design” step starts by using the eTXdesigner to create the “source” and “target” transactions. The source transaction refers to the hierarchical structure and all the data elements that will be extracted from the transaction source system. The target transaction refers to the hierarchical structure and data elements that will be input into the target system or end-point. The eTXdesigner allows you to design simple or complex transactions involving an unlimited number of table or view objects in a hierarchical structure with one-to-one or one-to-many relationships.

The “map” step involves using the eTXmapper to map the data elements between the source and target transactions and specify any transformation that needs to be done to the source data elements. In this step you can perform

string manipulation, arithmetic operation between various source transaction data elements, combine multiple source data elements and map to a single target data element. You can also split a single source data element to multiple target data elements, comprise an external program to process any of the source data elements or access an external database to do look-ups for the data element values. The eTXIntegration Suite even offers you the ability to use Visual Basic or Java Script to perform custom data manipulation as needed. At the end of the mapping process, the eTXmapper auto generates the XSLT document for the transformation.

In the “configure” step, the eTXadministrator is used to define the source and target systems and their ODBC connection definitions, the transaction name and its parameters (such as the source and target XSDs and which XSLT name to use) and the origin and destination systems of the transaction. It also allows you to choose which adaptors to use to write to the target system and gives you the option of setting up hub and spoke or point-to-point configurations. The initiation of a transaction can be done via triggers in the database or through the polling agent service that monitors the source system for changes. The criteria for detecting changes in the source system can be based on a time stamp value, data element values or a combination of both.

In the final, “integrate” step the eTXdirector starts up the various Windows NT services to facilitate the integration between each end-point you have established in your integration project. At this point, all of your source and target systems will be sharing data in real time. Figure 2 shows the integration process steps involved.

## Integration Example

The example shown in Figure 3 (page 3) depicts a procurement system sending out an invoice transaction to various financial systems that are based on different databases. The user enters a new invoice or updates an existing invoice in the procurement system. This business event triggers a transaction that has been defined and configured as per the steps outlined above. The eTXIntegration Suite gets a message (i.e. event notification) that an invoice has been added or changed for a particular invoice number and passes on the name of the transaction to be triggered. On receipt of this notification, it gathers up all the relevant information, such as the source and target XSDs and the target systems it needs to send the transaction. Once the data is gathered, it completes the transformation and sends the transactions to the target systems. The transactions that are sent to the databases are under full transactional integrity constraints. If the transactions fail for any reasons, the error recovery process takes over and keeps trying until the transaction succeeds or an email notification is sent to the system administrator (provided in the configuration step). An audit trail of the entire transaction is kept in the logs and the repository, including the actual transaction data.

## Summary

The eTXIntegration Suite provides a simple and affordable way to integrate applications. Not only do you save money at time of implementation, but you also have the flexibility to do future integration without incurring additional programming expenses, resulting in much lower Total Cost of Ownership (TCO). The eTXIntegration Suite can meet and exceed the expectations of its users because of the following differentiators that set it apart from the competition: Inexpensive, quick setup, standards based, no programming involved and low risk implementation. We guarantee it.

Figure 2: eTXIntegration Process

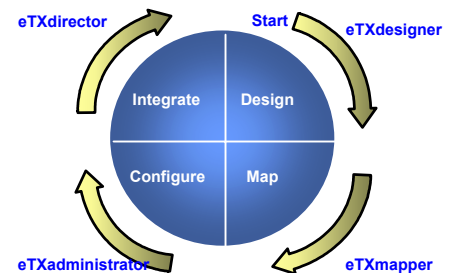


Figure 3: Integration Example

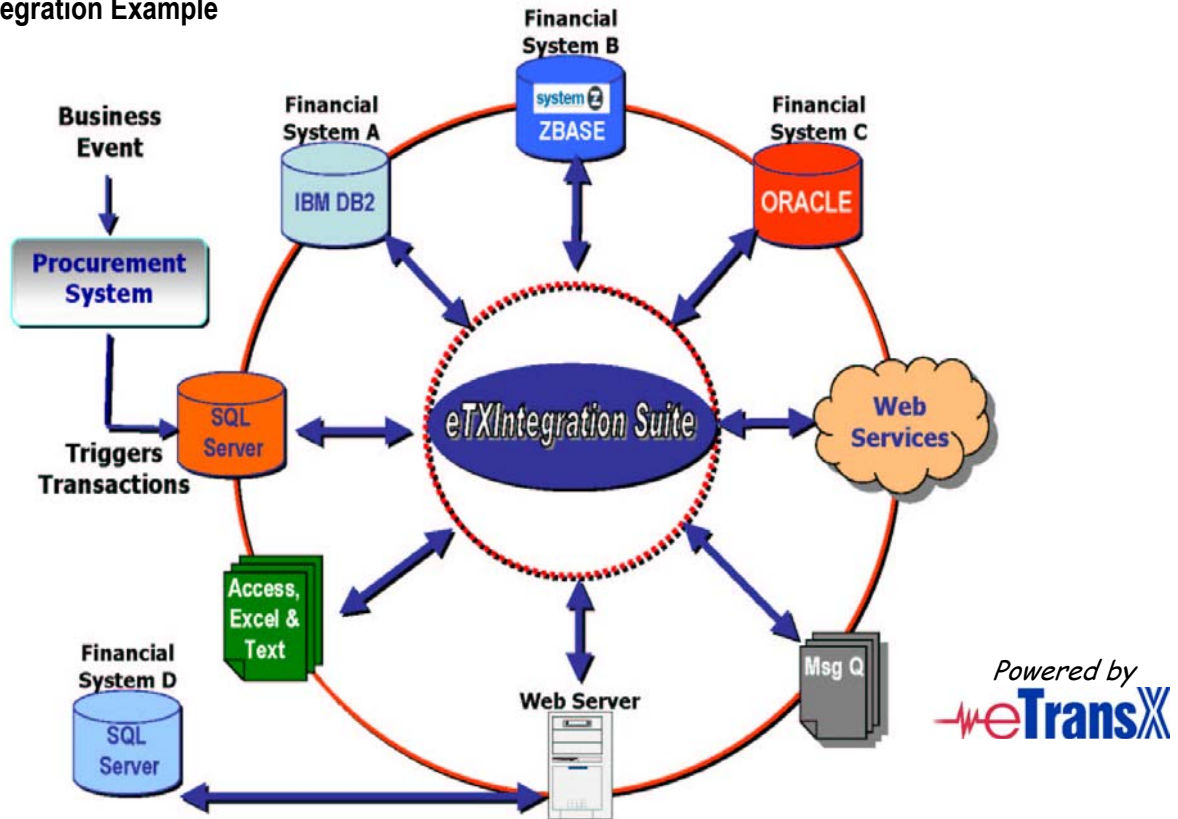


Figure 4: Screen Shot –  
Transaction Monitor

QueueId	TxnId	TxnName	ObjctId	DateCreated	Control	StartTime	EndTime	Message
1	1	Product_Info_Response	7	Sep 10 2003 9	C	Sep 10 2003 10:20:26 AM	Sep 10 2003	Succeeded
10	9	Invoice_Txn	INV-001 IBM	Aug 23 2003 1	C	Aug 23 2003 12:04:42 PM	Aug 23 2003	Succeeded
10	9	Invoice_Txn	INV-001 IBM	Aug 23 2003 1	C	Aug 23 2003 12:04:42 PM	Aug 23 2003	Succeeded
10	9	Invoice_Txn	INV-001 IBM	Aug 23 2003 1	C	Aug 23 2003 12:04:42 PM	Aug 23 2003	Succeeded
10	10	Product_Info_Response	16	Sep 10 2003 4	C	Sep 10 2003 04:22:03 PM	Sep 10 2003	Succeeded
100	100	Product_Info_Txn	6	Sep 11 2003 7	C	Sep 11 2003 07:41:56 PM	Sep 11 2003	Succeeded
100	100	Product_Info_Response	23	Sep 11 2003 7	C	Sep 11 2003 07:41:51 PM	Sep 11 2003	Succeeded
101	101	Product_Info_Txn	21	Sep 11 2003 7	C	Sep 11 2003 07:41:58 PM	Sep 11 2003	Succeeded

Count for All Transactions is: 1551

**Quick Search**  
 StartDate: [ ] End Date: [ ] Source Server: [ ] Target Server: [ ]  
 Txn Errors  Invalid Server/Txn  Txn Completed  All Transactions [Search] [Clear]

**Advanced Search**  
 Field: [Queue Id] Operator: [=] Criteria: [ ]  
 or  or  or [Clear] [Search]

**Data To Be Updated**  
 Control Flag: [ ] Target Server: [ ] Retry Count: [ ] [Clear]

[Purge] [Update] [Export]

Double Click to View XML Document