What Does IBM WebSphere TX Do?

- It takes any kinds of data from their native forms
  
  **ANY Data**

- Processes them together, natively, with NO CODE
  
  **Many-to-many Integration**

- And outputs them into their native target formats
  
  **ANY Data**

Ex. - Hierarchical Data, Binary Data, Packed Data, Tabular Data, Relational Data, Nested Structures, Mixed-Type Data, and on and on...
Transformation Vision - Simplified

- Transform Anything
- Transform Anywhere
- Everyone Transforms

The Need for Transformation

“Transformation is the hard, ugly part of integration”

- Ted Friedman, Research VP, Gartner

  "Organizations pursuing service-oriented integration involving document-oriented, proprietary, legacy or modern XML-oriented data standards... should consider products capable of automating complex data transformation as an important building block of their integration architecture."
  Massimo Pezzini, VP and Distinguished Analyst, Gartner

- “Two of the biggest and most significant challenges of dealing with data are transformation and semantics.”
  Zapthink

- “There are many integration styles and challenges, each with their own unique demands; what is consistent across every one of them is the need for data transformation and mediation.”
  IBM
The Challenge and Transformation

- About 35 percent of an enterprise’s software budget is spent on maintaining the multitude of point-to-point application links already in place. — Gartner Group

1. Each system has its own data definitions, business rules and interfaces
2. Providing granularity to application’s business functions requires ability to maintain contextual usage and data meaning across systems
3. Interactions between applications expose data and business rule anomalies which result in high maintenance costs and inflexibility to change
4. The Validation and Use Rules must align with the Transformation needs and execute in the same unit of work with Transformation
5. Ever increasing integration logic points with duplicate or slightly modified templates rob from anticipated benefits and efficiencies

Real World Transformation Demands Demand a Broad Range of Capabilities

Almost all projects that require these today, do so via programming, even if they are using middleware as the platform (e.g. Java, C++, etc)

Only WTX can provide all of these… …AND DOES SO WITHOUT CODING!
What is the problem of Transformation
Processing Information with all the business rules and usage mechanisms in tact…

What Makes WebSphere TX so Different?

- Code-Free Design and Deployment
- In-Process Data Validation
- High Throughput Execution of Complex Transformations and Enhancements
- One Engine, Multiple Deployment Options
Examples of Complex Transformation Challenges Addressed by WebSphere TX

- Data Enhancement
  - Lookups
  - Data Logic and Routing
  - Data Validation
  - Context Based Data and Usage Rules

- Many to Many Transformation
  - Single-Transaction, interdependent data sets, conversions and logic
  - Mixed Data and Source/Target Types
  - Dependent Result Sets, Nested Structure Dependencies

- Complex Data Transformation
  - Nested, Semi-structured And Hierarchical Data Types
  - Dependent Inputs And Outputs
  - Binary, Packed, EBCDIC, ASCII, Mixed Character Data

One Engine, Multiple Deployment Options

- WebSphere TX is the one transformation engine that can be reused across all applications and infrastructures in the enterprise
  - Deploy the same WebSphere transformation asset across:
    - ESB
    - SOA
    - Web Services
    - SOAP
    - EJB
    - HTTP
    - Event driven
    - Batch
    - Java
    - C/C++
    - COBOL
    - Unix
    - Linux
    - Windows
    - z/OS
      - Batch
      - CICS
      - IMS
      - USS
    - 30+ Adapters
    - ETL
    - Data quality
    - Enterprise Applications
    - Application Servers
    - WebSphere Message Broker
    - WebSphere ESB
    - WebSphere Process Server
    - WebSphere Partner Gateway
    - WebSphere Application Server
    - WebSphere DataStage
WebSphere TX is further enhanced by Vertical Product Packs

**Financial Services**
- SWIFTNet
- SWIFTNet Funds
- SEPA*
- Many Services Based Solutions
  - FIX
  - FedWire, Chipps
  - NACHA
  - BAI, BAII2
  - AL3
  - ACORD
  - Etc.

**Health Care**
- HIPAA
- NCPDP
- HL7
- Many Services Based Solutions
  - HealthCare Hub
  - Payment Processing
  - Clearing House Processing

**CPG, Manufacturing**
- ERP/CRM Integration
- EDI (ANSI X12)
- TRADACOMS
- ODETTE
- EDIFACT
- Many Services Based Solutions

**Embedded Software**
- Integration to the Enterprise as an embedded service
- Telecom
- Enterprise Apps
- Shop Floor Systems, etc...

---

WebSphere TX – A Long And Storied Road

Proven

Reliable

Innovative

Mature

Jan 1994

Jan 1995

June 1995

June 1996

Dec 1999

Aug 1997

June 2000

Sept 2001

Jan 2002

Jan 2003

July 2004

V6.0

V6.5

V6.7

V7.5

V8.0

V8.1

V...

...
Global Financial Institutions rely on WebSphere Transformation Extender

- Fortune 1000 Coverage
  - 9 of Top 10 Commercial Banks
  - 6 of Top 15 Diversified Financials
  - 4 of Top 12 Life and Health Insurance Companies
  - 8 of Top 14 Property and Casualty Insurance
- Significant Impact on Capital Markets:
  - $3 Trillion of transactions flow through WebSphere TX daily
  - 40% of London securities transactions are powered by WebSphere TX
- WebSphere TX for STP:
  - 100 international banks
  - 70 investment managers
  - 40 broker dealers
  - 10 global custodians
  - Major Stock Exchanges
  - European Clearing Houses

WebSphere TX is widely used in all industries
WebSphere TX: Universal Transformation & Reuse

- Transform, validate, and enrich any document, message or complex data
- Deliver trustworthy information for critical business initiatives
- Meet regulatory compliance requirements
- Codeless development; universal reuse and deployment

WebSphere TX: A Unified Transformation Environment across the WebSphere portfolio

- ONE Transformation Engine for the Enterprise!
  - Powerful transform capabilities without coding
  - Natively handles any data type
  - Solves really hard transformation problems
  - Complements existing portfolio of products
  - Delivers new sales opportunities

For batch environments (as a standalone engine)
For WebSphere Message Broker
For WebSphere Process Server
For WebSphere ESB
For WebSphere Partner Gateway
For 3rd Party Integration Products (as an embeddable engine)
Developing a WebSphere TX Solution

The Components of a WebSphere TX Solution

- A WebSphere TX solution is composed of the following:
  - Adapter
  - Type Tree
  - Rules
  - Type Tree
  - Adapter

In other words…
- An adapter is a technical connector which can be “plugged” into a source or destination - without parsing!
- A Type Tree is a graphical representation of meta-data
- Rules define how a Source structure is transformed to a Destination structure
- Together these components form a WebSphere TX Map
The Components of a WebSphere TX Solution

- However…many to many mapping is also possible:

- A Map can perform many tasks:
  - Validation
  - Enrichment,
  - Content Based Routing
  - etc. SOURCE FILE

Orchestrating Transformations
Developing a WebSphere TX Solution

- We usually talk about three simple steps:
  - Describe the data...
    - Data that is handled by IBM WebSphere TX, are described in “Type Trees” that define the data structure and semantics.
  - Transform the data...
    - Data is mapped between the source and destination using drag-and-drop.
  - Deploy the and execute the transformation
    - Multiple deployment models
      - Batch
      - Event Driven
      - Tightly-coupled (through APIs)
      - Loosely-coupled (RMI)

WebSphere TX Design Environment

- Describe the Data
  - Properties
  - Structure
  - Validation Rules

- Define Transformation Rules
  - Interfaces
    - Structure
    - Formats
    - Semantics
  - Resources
    - Communication
    - Adapters

- Deploy and Manage
  - Flows
  - Events
  - Monitoring

- Tools
  - Type Designer
  - Database Interface Designer
  - Map Designer
  - Message Broker
  - Launcher
  - IMS
  - Batch / JCL
  - CICS
Describe the Data...

### IBM WebSphere TX Type Trees
- The Type Trees define the structure of the data to be handled in IBM WebSphere TX.
- This XML structure is then represented as an IBM WebSphere TX Type Tree.

WebSphere TX Design Environment

- **Describe the Data**
  - Properties
  - Structure
  - Validation Rules

- **Define Transformation Rules**
  - Interfaces
    - Structure
    - Formats
    - Semantics
  - Resources
    - Communication
    - Adapters

- **Deploy and Manage**
  - Flows
  - Events
  - Monitoring

**Type Designer**
- Database Interface Designer

**Map Designer**

**Message Broker**
- Launcher
- Batch / JCL
- IMS
- CICS
Map Designer

- Create maps to:
  - Validate data from data sources
  - Identify data targets
  - Specify data transformation logic (rules)

Identifying Data Sources and Targets

- Identify type tree and type tree components that define data target
- Identify type of data and location of data
WebSphere TX Adapters

**Messaging**
- IBM Message Q & Tuxedo
- Candle Riana
- IBM WebSphere MQ
- IMS
- Microsoft MNSQ
- Oracle AQ
- TIBCO Rendezvous

**Applications**
- Siebel
  - BIM, EIM, OE COM
  - SAP R/3 & BW
  - ALE, RDD, DML, DXOB
  - BW, EDI, BAPI
- PeopleSoft
  - Message Agent API
  - Open Query, ZDI
- QualityStage

**Database**
- DRZ
- Informix
- VODIC *
- VOLE DB
- Oracle *
- Microsoft SQL Server
- Sybase
- HP NonStop SQL/MX & SQL/MP

**Communication**
- CICS *
- COM
- CORBA
- E-MAIL *
- FTP/FTPS *
- HTTP/HTTPS
- JAVA Class
- JAVA Gateway
- SOAP
- SOCKET
- VAN

**Utility**
- Archive (ZIP/TAR)
- Base 64
- Batch
- File *
- GZIP/ZLIB
- NDSS
- DAP
- MIME
- Quoted Printable
- Shell Script
- Sys *
- DMAP *
- XML *

**Security**
- S/MIME
- OpenPGP
- SSL Module

* Adapters supported on z/OS, varies by execution method

---

**Step 1 – Define the data structures**

- Pre-defined Importers for automatic generation
  - XML Schema, DTDs
  - COBOL Copybooks
  - Database (Tables, Vues, Procédures Stockées)
  - 3rd Party Applications
    - SAP: BAPI, Idoc, DXOB, BDC
    - PeopleSoft 7: Business Document, Message Agent
    - PeopleSoft 8: Application Messaging, Component Interface
    - Siebel: COM Business Object, EAI, EIM
    - WSDL (Web Services)
    - EJB API
    - Java Class
    - Java Messaging Service
    - Others: Tuxedo, Flat File, PL 1...

- Pre-defined libraries ( Packs)
  - EDIFACT
  - SWIFT
  - SEPA *
  - ACORD
  - HIPAA
  - HL7

* En cours
Built-in Validation (Restrictions)

Items have extensive validation options
- Enumerations
- Exclusions (with substitution)
- Patterns
- Ranges

Additional Built-in Validation
- Component Rules allow cross-field validation
- Incorporates semantic validation directly into the parsing process
Additional Built-in Validation

- Component Rules support aggregate functions
- Use component rules to specify conditions (business rules) that must be met for a particular component to be valid

Step 2 - Creating transformation rules

- Drag and Drop
- No Procedural Coding!!
- Easily represent complex business rules
Calling external functions and programs

Rejecting Errors
Step 3 – Deploy and execute the transformation map

- Option “Build” creates a compiled map
  - For the execution server
  - Automatically accounts for “byte-order” and code page differences

Example – Developing a WebSphere TX Map

1. Define the Data Structures
2. Define the transformation rules
3. Deploy and execute

Multiple Execution Methods
- API
- Batch / JCL
- Event-Driven / ESB
  - WebSphere Message Broker
  - WebSphere ESB
  - WebSphere Process Server
  - WebSphere Partner Gateway
  - WebSphere Application Server
  - WebSphere DataStage, etc.
WebSphere TX General Architecture

Design Studio

- Type Designer
- Map Designer
- Integration Flow Designer
- Importers
- Database Interface Designer
- Unit Test Engine

WebSphere TX Server
- WTX Core Engine
- WTX Adaptors
- WTX Compiled Maps (.mmc)

Server A

WTX for z/OS
### WTX for z

**Transformation and execution on z/OS**

<table>
<thead>
<tr>
<th>Product Offering</th>
<th>Execution Method</th>
</tr>
</thead>
</table>
| WebSphere Transformation Extender with Command Server | - Batch / JCL  
- CICS  
- IMS TM  
- IMS/DC |
| WebSphere Transformation Extender with Launcher USS Edition | - Event-driven, multi-threaded, multi-process transformation engine |
| WebSphere Transformation Extender (embedded edition) | - APIs & Interfaces for: Cobol, Java, C, EJB  
- For use with WAS, WPS, Cobol programs, etc. |
| WebSphere Transformation Extender for Message Broker USS Edition | - WTX Parser  
- WTX Map Node |

### WTX for Z/OS - Batch

- « Pure » Batch execution mode invoked through JCL
  ```plaintext
  // SET WTXLIB=LIBRARY.WTX.REL8.XPL LOAD <= WTX LOAD LIB  
/DESTX EXEC PGM=DESTX,REGION=OM,  
PARM="SEPAPP /IF1 INPTSEPA"  
/STEPJLIB DD DISP=SHR,DSN=LIBRARY.WTXLIB
  ```

- SMP/E installation

- Supports the following sources/targets natively
  - Files (QSAM, VSAM)
  - DB2
  - FTP
  - MQ

- EXIT function allows existing COBOL and C programs to be called from a WTX Map
WTX for Z/OS – Batch - continued

- Administration through standard z tools (ISPF, etc.)
  - No need to learn a new monitoring tool

- No permanent active processes
  - WTX is only active when called by the JCL
  - When map transformation ends
    WTX is no longer active

- Integrates seamlessly into existing production procedures
  - Does not require changes to customer job and system administration processes

WTX for Z/OS - CICS

- Transactional execution mode through a CICS transaction

- Runs natively in CICS – inline and batch modes supported

- SMP/E installation

- Supports the following sources/targets natively
  - VSAM Files
    - key-sequenced
    - entry-sequenced
    - relative-record data sets
  - DB2
  - MQ
  - Transient data queues
  - Temporary storage queues
  - Storage buffers (COMMAREA)

- EXIT function allows existing COBOL and C programs to be called from a WTX Map.
WTX for Z/OS – CICS - continued

- WTX z/OS CICS can be invoked:
  - From a clear terminal by using the **DIST** transaction
    - i.e. DSTX MYMAP -IF1 INPUTF
  - From a user program by starting the **DIST** transaction in the **EXEC CICS START** command
  - From a trigger transient data queue by initiating the **DIST** transaction
  - By calling the **DSTXCICS** program through an **EXEC CICS LINK**

- Administration through standard z tools (CEBR, etc.)

- Integrates seamlessly into existing production procedures

WTX for Z/OS – IMS TM & IMS/DC

- Allows WTX Maps to be programmatically executed through IMS

- Relies on WTX Platform APIs and high-level language programs

- C/C++ and COBOL programs can invoke WTX through the IMS execution option

- WTX Maps participate in the IMS transaction

- **EXIT** function allows existing COBOL and C programs to be called from a WTX Map
WTX for WebSphere Message Broker

WebSphere Transformation Extender for WMB USS Edition

- WebSphere TX is an additional Node on the Message Broker design palette
- Embed WebSphere TX transformations within Message Broker Message Flows
- Interoperates with Message Broker V5.x or 6.0
- Fully compatible with Message Broker transaction control
- Takes full advantage of Message Broker scalability
- Full support for multiple sources and multiple targets within a single transaction scope
- Installs in minutes

“The most powerful Any-to-Any Broker on the Market”
WTX for WebSphere Message Broker Components

- **Parser**
  - Transforms bit streams to Message Trees (MQInput) and Message Trees to bit streams (MQOutput)
  - Transforms Message Trees to Message Trees (RCD)

- **Plug-in node**
  - Transforms anything to anything
  - Icon installs into Message Flow palette
  - Easy configuration

WTX for WebSphere Message Broker Plug-In Node

The Plugin node can be used in message flow like any other node.
Chief Value Differentiators of WebSphere TX to alternative transformation methods

- Powerful Transform capabilities with No Coding
  - Lowers cost to implement, maintain and re-use
  - High-Throughput, Complex Transformations
  - Cost per transaction very low

- Natively Handles Any Data Type
  - Never resort to “flattening” or “re-structuring data”
  - Data integrity and meaning are never lost

- In Process Data Validation
  - Greatly reduces development efforts and implementation costs

- Solve Really Hard problems, in less time, with one common design method