

Enterprise service bus solutions
To support your IT objectives



WebSphere® software



Extending your ESB for complex data.

Delivering business agility, transforming vital business information and helping ensure content quality with WebSphere Transformation Extender

By Colin P. Taylor,
Product Marketing Manager, IBM WebSphere



Compliance and agility — when transforming your vital business information

ESB transformations: Protecting information integrity

A service oriented architecture (SOA) allows organizations to update and reorchestrate business processes to meet changing market conditions, reusing existing applications that are running on existing hardware. The most successful organizations implement SOA with a focus on business value, and tested and proven techniques. They work smarter.

Data *integrity* can be defined as the trustworthiness and dependability of information. Maintaining this integrity across a highly distributed SOA process is a more complex challenge than maintaining integrity on a single, closed system. Businesses succeed by focusing on three key components of data integrity:

- *Transactional integrity*
- *Information integrity*
- *Human interaction integrity*

Information integrity applies to data cleansing, data warehouses and master-data management. When you examine information integrity from the perspective of the enterprise service bus (ESB), a cornerstone of SOA, you can see how the transformation of complex data can help in meeting the interface requirements of internal business applications and trading-partner systems.

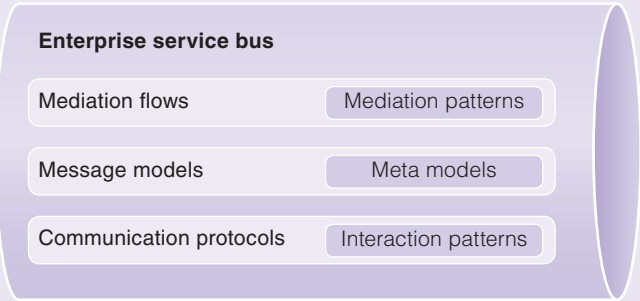


Figure 1. Basic conceptual model for an ESB

Application reuse requires more than composite applications can deliver

A key feature of ESBs is to mediate, or *transform*, information stored and retrieved in existing back-end applications and data stores for use by new business service applications (see Figure 1). These business-service applications are composite applications, designed to reuse existing interfaces by using the integration technologies of an ESB.

For many enterprises, application reuse requires more-advanced capabilities than ESB products generally provide. They require:

- *Noninvasive integration of application-interface formats*
- *Integration of existing enterprise business suites with multiple interfaces*
- *Integration of enterprise applications with adapters at the data tier*
- *Synchronization of data between applications*
- *Integration of electronic data interchange (EDI) systems migrated from vendor access networks (VANs) to the Internet*



Business-to-business (B2B) transactions add more requirements that, if coded into the applications themselves, detract from the business logic and threaten to increase the maintenance costs required year on year:

- *Regulatory compliance*
- *Compliance with guidelines for implementing industry standards*
- *Bilateral service-level agreements with trading partners*

Business process management (BPM) solutions that choreograph the automation of business processes using composite service applications also provide functions to insert business rules into code and mediations for basic transformation. But these BPM solutions are not necessarily designed for handling larger data structures and complex, variable-structured documents. They're not designed to collect and correlate multiple input sources that have different data formats, and they're not designed to process these sources together as one entity while preserving the original relationships and context.

Many of these solutions are intended for new, pure-XML projects. BPM solutions that use an XML-based ESB will need additional support for non-XML data to avoid writing additional pre- and post-processing utility applications. And some of the XML Schemas now being published as templates are very large. The real world is far from being a pure-XML world. The world of reuse requires more IT flexibility to deliver the agility that *smart* businesses need.

What constitutes complex data?

Although complexity is caused by more than simply the size of the document or message to be processed, size can present a challenge to some parsers. Techniques are available to chunk a message and reassemble it. Fan-in and fan-out techniques can handle the complexity of aggregating information from multiple messages and documents. Correlation and synchronization of activities fall under the heading of complex-event processing, and techniques are available in ESBs, for example IBM WebSphere® Message Broker technologies, to accomplish this. But complex-event processing should not be confused with complex data.

EDI formats fall into the category of complex data, typically having an envelop structure permitting multiple transactions of multiple transaction types (for example, purchase order, sales order and invoice). EDI formats were developed at a time when the cost of a transaction over a managed value added network (VAN) service was significant enough to warrant batching of transactions.

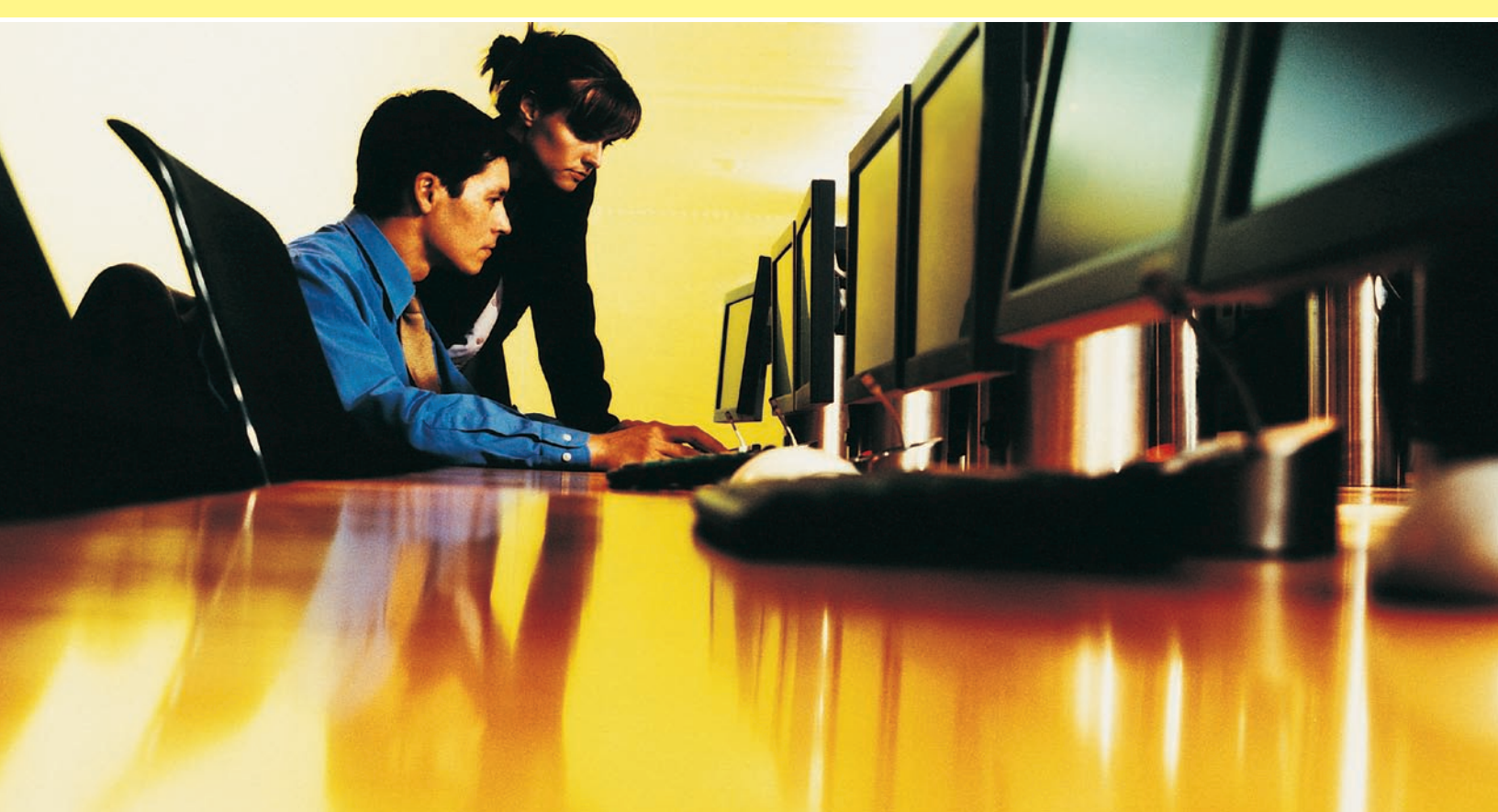
A second characteristic of complex data, also demonstrated in EDI and early industry standards, is the need to keep the byte count to the necessary minimum. Although some fields have fixed lengths with white space filling unused bytes, others use delimiter characters (for example, = and #) to separate one variable-length item from the next. The order of the items is usually fixed, with 123**ABC to indicate that the middle item has no data. A template can be defined in advance for parsers

to decode the content. However, at least one standard permits the delimiter characters to be defined and redefined within the document or message itself. You might start processing with * as the delimiter, and later in the document, need the parser to change it to #. Or you might even use both at the same time. This requires more-sophisticated parsing techniques than usually found in an ESB.

A widely used format in the finance arena is called Financial eXchange (FIX) Protocol (www.fixprotocol.org). FIX Protocol is a global standard for the electronic exchange of trading information. When first introduced, it used a tag=value structure as shown in this example:

```
8=FIX.4.19=11235=049=BRKR56=INVMGR34=23552=199806  
04-07:58:28112=19980604-07:58:2810=157
```

This format is still used by many organizations today. FIX Protocol tags can be designated as mandatory or optional. Different users of trading applications might require a specific tag or value that is listed as optional or perhaps not defined at all within a message. FIX provides a large suite of tags and permits some that can be *user-defined*, providing an additional layer of flexibility and the opportunity for customization. User-defined tags enable developers to customize and differentiate varying trading applications and systems, offering different functions for different business purposes. But the tags also add to the complexity. Exchange partners are expected to set up bilateral agreements for using optional and custom tags. The FIX *engine* (in our case, the ESB implementation) must have the right mechanisms in advance to process this data correctly. More importantly, it would be inefficient and costly to maintain if these setups had to be performed for every partner relationship.



FIX also highlights the next level of complexity: standards evolution. FIX, Version 4.19 is only one of many versions in use. An ESB would need to accommodate syntactic differences of FIX, Version 4.0 and Version 4.1 through to Version 5.0. FIX versions 4.1 to 4.4 are also available as XML (FIXML) with DTD templates. FIXML, Version 4.4 and Version 4.5 have XML Schema templates. And further adding to the complexity, FIX, Version 5 can contain XML and tag=value structures. Mediations based on the XSLT language would struggle to handle these mixed standards.

We have used FIX as just one example of the potential for high maintenance when using complex data without the right tools for the job. Similar complexities exist when using industry-standard EDI formats, such as Health Insurance Portability and Accountability Act (HIPAA) in health care, Association for

Cooperative Operations Research and Development (ACORD) in insurance, Society for Worldwide Interbank Financial Telecommunication (SWIFT) in finance, ASC X12, and Electronic Data Interchange For Administration, Commerce, and Transport (EDIFACT) in retail commerce. In addition to these complexity concerns, FIX 5.0 Service Pack 1 is in beta test and due for final release in March 2008 (www.fixprotocol.org/fix5.0sp1).

This variability of content is also not confined to B2B and EDI exchanges; it exists in COBOL application data (COBOL copybook formats). And SAP document templates are often highly customized to in-house requirements. The overall data structure of one application can be so different from that of another that the data, in effect, must be turned *inside out*.

Content validation for complex data

There are several business reasons to check the content as well as the syntax of complex data formats when transforming data:

- *Service level agreements*
- *Guidelines for implementing industry standards*
- *External rules and regulations for business services*
- *Compliance with regulatory mandates*

The SWIFT network regulations can penalize firms that repeatedly submit invalid messages. The Single Euro Payments Area (SEPA) initiative has introduced new XML-based templates for Pan-European payment transfers in the European community, where the implementation guidelines are extensive, written in local languages and still evolving as the initiative is rolled out across more countries. And HIPAA compliance is a key driver for health-care providers and insurers in the U.S., with a requirement to adopt the latest release (version 5010) in 2008.

It might seem prudent to have validation-checking programs regardless of business need. But when you have to write lengthy code or scripts to implement content validation, the performance overhead is often given the higher priority, and the amount of validation checking in production systems is limited to essential items.

Businesses run the risk of inserting bad data into applications, or sending correctly formatted but incomplete information. This is where it is necessary to apply business and technical rules, and compute derived values. Performing these operations within the ESB minimizes the invasive impact on existing applications, and avoids the need to write *fat adapters* to perform pre- and post-processor validation in addition to the connectivity role that adapters provide.

With an ESB enhanced with transformation capabilities, business service applications can focus on the business logic without the overhead of a lot of validation logic. Performing content validation in conjunction with transformation can result in fewer rejections by target applications or external business services. It helps to reduce manual intervention to fix and resend. And it enables more transactions to be completed the first time.

Many implementations of transformation services within an ESB or process-automation server convert input sources to a generic (canonical or superset) data representation before processing, and then convert back to individual application-specific business objects. This technique has the advantage that any application-specific variant of the data can be generated for target applications. However, the interrelationships of the content in the original sources might be lost. For many scenarios with one or two input documents this is manageable, but as the number of input sources that need to be processed together increases, and the size and complexity of source documents grow, it is more efficient to implement a transformation service that maintains the original source relationships and cross dependencies during processing.

And on a simple expense basis, the cost of a separate rules engine to maintain the relationships and transformation engine to tackle the problem might be prohibitive.



WebSphere Transformation Extender

IBM WebSphere Transformation Extender is a universal data-transformation engine designed to thrive on the complexity and diversity of information integration.

WebSphere Transformation Extender solves the toughest transformation problems without programming. It delivers high-performance transformation, particularly when dealing with multiple applications, complex data types and diverse platforms.

WebSphere Transformation Extender enhances ESB solutions from IBM.

Code-free design and deployment

There is no language to WebSphere Transformation Extender. The transforms and data process are all maintained within the spreadsheet-like GUI, and you never need to actually write code to handle complex transforms. You create portable *transformation objects*. The WebSphere Transformation Extender Design Studio tools can be used stand-alone or within the Eclipse open standard workbench of the ESBs from IBM. And thanks to a built-in file adapter, you can easily use local test data to analyze and debug transformations on your desktop before creating platform-specific versions for the runtime platform of choice.

Flexible and efficient asset management

The transformation assets generated within the design studio are file based and so lend themselves to version control in any in-house code-management system, such as IBM Rational® ClearCase® software. There is no runtime repository either, so additional database licenses just to support the solution are avoided.



Self-describing data model

WebSphere Transformation Extender uses data in its source format and has a unique mechanism for describing data in its native form. WebSphere Transformation Extender is able to handle complex and mixed data types using one design environment. It can process up to 100 input sources with one transformation and route content to 100 destinations.

Native support for XML Schemas

WebSphere Transformation Extender Design Studio can import XML Schemas directly into the mapping tool, automatically presenting it to the designer as a type tree representation ready to apply functions from a menu to build transformation and validation rules.

Data validation as part of the transformation process

Data is validated against content rules and context usages as part of the transformation process. You do not need to write separate logic or have separate executions in order to provide extremely rich data validation.

High throughput of complex transforms and enhancements

WebSphere Transformation Extender has a unique many-to-many model of transforming and processing data, which allows it to process all transforms, lookups and data enrichments with only one pass at the data, making it one of the top-performing transformation engines on the market.

Multiple deployment options, including z/OS and Linux on System z

Using the same design environment, you can deploy transformations to a number of runtime environments, including embedded, stand-alone batch and event-driven scenarios, across a number of operating system platforms, including IBM z/OS® and Linux® on System z™.

With WebSphere Transformation Extender, you can create *functional maps* that perform specific mapping functions. These maps can be thought of as similar to subroutines in traditional programming languages. An example of a functional map is a map that is processed every time a particular segment occurs in a transaction. For example, when processing health-care patient information, you might want to create a functional map that formats the social security number and the internal patient I.D. every time a patient I.D. segment is encountered. Iterating back and forth over the source data is a common activity for computing new values from related fields and working with cross dependences in batched transaction elements.

WebSphere Transformation Extender Industry Packs

Industry Packs are used in conjunction with WebSphere Transformation Extender Design Studio to provide organizations with an infrastructure that:

- *Enables compliance with government and industry mandates.*
- *Controls administrative costs.*
- *Streamlines business processes.*
- *Facilitates accuracy and timeliness of information.*
- *Reuses existing business systems.*
- *Adapts to new technologies as they emerge.*
- *Integrates multiple systems and standards.*
- *Automates cross-organizational exchanges.*

WebSphere Transformation Extender provides the following Industry Packs:

- *EDI*
 - *X12*
 - *EDIFACT*
 - *TRADACOMS*
 - *ODETTE*
 - *EANCOM*
- *Health care*
 - *HIPAA EDI*
 - *HL7*
 - *NCPDP*
- *Insurance*
 - *ACORD*
- *Financial services*
 - *FIX*
 - *NACHA*
 - *SEPA*
 - *SWIFT FIN*
 - *SWIFT FUNDS*



Industry Packs accelerate the development of customer- and industry-specific transforms and provide prebuilt validation checks on the content and structure of data. Used with the Design Studio, they remove any need to code this logic into applications. Industry Packs contain ready-to-use templates for *type trees* and maps. *Type trees* provide the data-dictionary-like specification of how the data is structured for inbound or outbound processing. A *map* defines the instructions for transformation and validation processing. IBM data architects take the paper documents and do the hard work of translating them into mapping rules using the prebuilt function library in the Design Studio. The templates are carefully designed for reuse and with user customization in mind. For complex data, several classes of functional type trees can be derived:

- *Industry subset type trees*
 - *Clients can customize (prune) templates for an industry subset of the possible functional groups specific to an industry or group of companies.*
 - *These help improve mapping of runtime performance, because there are fewer functional-group partitions to test during validation.*
- *Multiversion type trees*
 - *In order to support different trading partners, it is often necessary to create a type tree that represents multiple EDI version releases. For example, one partner might send #850 purchase orders from ANSI version 3040 and another partner might send #850 purchase orders from ANSI version 3050.*
- *Multipartner type trees*
 - *A multipartner type tree allows you to specify particular partners and groups of partners by business relationships, such as customer, distributor or supplier.*
- *Multistandard type trees*
 - *In order to support requirements for different partners, it is often necessary to create a type tree that represents multiple standards. This might be needed, for instance, if some partners use ANSI X12 and others use EDIFACT.*

Industry Packs might also include data files to simplify lookup processing and utility programs to help with validation. For example, Industry Packs for financial services might look up bank identifier codes and country codes to expand them into readable form. This data might already be available in your database, in which case it is easy to override the file lookup process used in functional testing with a database lookup process for the production system.

Often EDI partner identification is not the identification method used in your application. In this case, you need to modify your map to look up your application's partner identifier based on the EDI partner identification information. For this you need a cross-reference data source that contains the EDI partner identification, the corresponding customer identification code that is used within your application and a type tree that defines that cross-reference data. To manage the coordination of partner-specific information, you will probably need a *partner profile* data file that contains trading partner information, such as ISA envelope identifiers and control numbers.

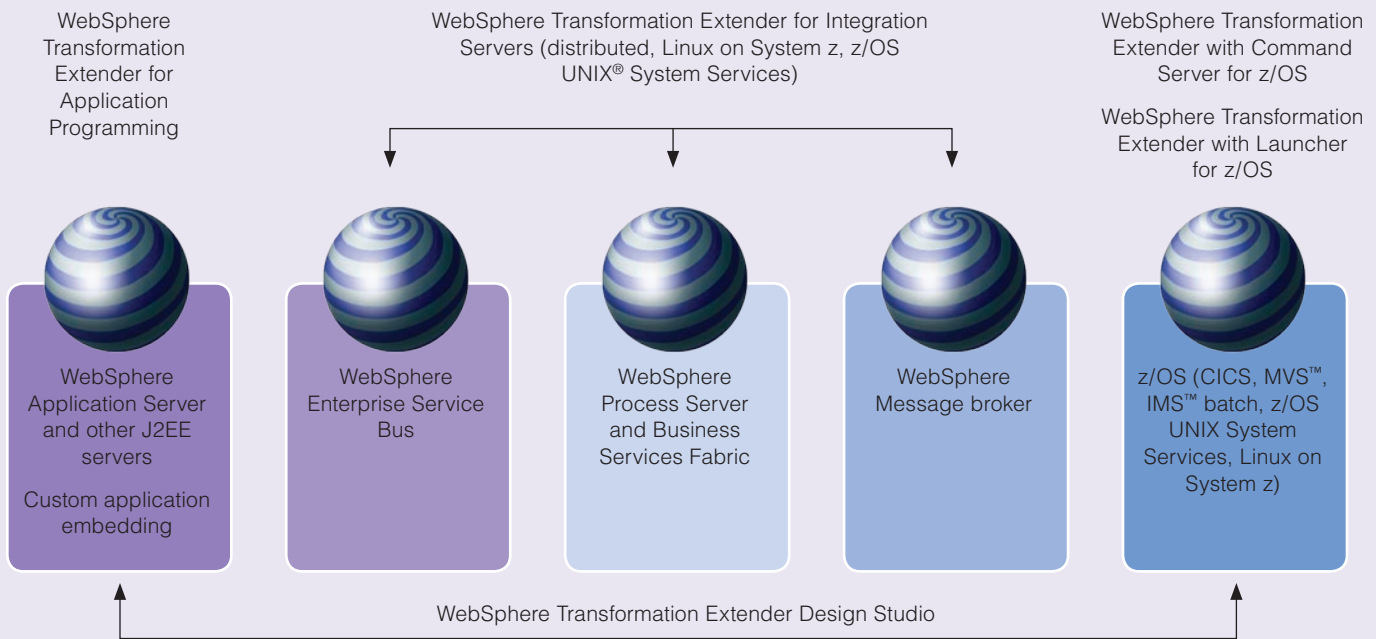


Figure 2. One transformation engine, multiple deployment scenarios

Complementary WebSphere solutions for SOA

Figure 2 shows that the WebSphere Transformation Extender road map is enabling more ESB and BPM products from the IBM WebSphere family to offload complex data transformation and content validation to its dedicated engine. IBM WebSphere Transformation Extender for Application Programming can be used with IBM WebSphere Application Server as well as third-party ESBs and Java™ 2 Enterprise

Edition (J2EE) servers. The engine can also be used under the transactional control of IBM CICS® software or in batch operations natively on the IBM System z platform.

IBM WebSphere Transformation Extender for Integration Servers is specifically designed for tight coupling with ESB and business process integration (BPI) solutions.



Extending WebSphere Message Broker

IBM WebSphere Message Broker is IBM's universal and extensible ESB with comprehensive routing capabilities and in-flight message processing. WebSphere Transformation Extender extends WebSphere Message Broker capabilities for scripted transformations with a macro approach similar to a spreadsheet, which simplifies transformation and validation capabilities for handling multiple, complex transactions for messages and documents.

Not only do the WebSphere Transformation Extender design tools plug in to the workbench used by the message-flow designer, but the designer can also drag, drop and configure a WebSphere Transformation Extender transformation activity in the same way any message flow is designed, giving a seamless user experience.



Extending WebSphere Enterprise Service Bus

Based on proven WebSphere Application Server technology, including its security model, IBM WebSphere ESB delivers standards-based connectivity and integration that allow you to create and deploy applications that support Web Services standards.



Data that is brought into this ESB is converted into a service data object before it is consumed by a business service application. Similarly, data passing from the application is converted to formats understood by the external systems reused in the solution.

WebSphere Transformation Extender can be used to perform these transformations for both simple and complex data. The ESB server supports data bindings that convert XML data to and from service data objects. If you have a format other than XML, you can create your own custom data binding for Java Message Service (JMS), enterprise information system (EIS), HTTP and WebSphere MQ messaging. To transform complex data, you can use the WebSphere Transformation Extender bindings provided by WebSphere Enterprise Service Bus to hand off processing and let the transformation engine do the hard work, according to the mapping rules you have configured independently. As well as bindings, an additional graphical wizard is provided to take existing business objects and define the type tree data model for use within the WebSphere Transformation Extender map designer tool.

Extending WebSphere Process Server

BPM enabled by SOA improves business processes and

hastens business innovation through the combination of business acumen and software capabilities.

Process automation is defined as supplementing a manual process with an automatically controlled alternative through the orchestration and integration of technology and human assets. It can reduce costs, increase efficiency and enable process compliance. *Process choreography* delivers an SOA solution based on a common model to orchestrate, mediate, connect, map and run underlying IT functions as part of their business processes.

IBM WebSphere Process Server contains IBM WebSphere Enterprise Service Bus, which mediates disparate services, helping to maximize reuse of assets wherever they are—regardless of vendor or platform, or whether they are built by companies themselves or provided as part of packaged applications.

The role of WebSphere Transformation Extender for integration Servers here is to reuse applications in order to provide the business objects used by business services, and deliver *compliant* information exchanges with external business. Equally, the choreography can delegate a complex processing task to WebSphere Transformation Extender, treating the product as a reusable application in its own right. It integrates with databases and applications in a single transformation activity, preparing and validating data so that it is ready to be used in downstream process activities. If the next activity requires a formatted report to be sent to an employee, WebSphere Transformation Extender can assemble the report, from as many as 100 input sources.

Extending WebSphere Business Services Fabric

IBM WebSphere Business Services Fabric is a comprehensive

SOA offering that extends the BPM platform. It adds dynamic BPM capabilities to process choreography when assembling and managing composite business applications.

Like WebSphere Transformation Extender, it offers optional industry content packs. WebSphere Transformation Extender Packs contain industry-standard transformation templates, while content packs for WebSphere Business Services Fabric contain process-level assets. IBM WebSphere Business Services Fabric is an end-to-end solution that includes the process server and tools of WebSphere Process Server.

Extending WebSphere DataPower Integration Appliance XI50

IBM WebSphere DataPower® Integration Appliance XI50 is another ESB offering from the WebSphere family. It provides an easy-to-use, high-performance, security-hardened hardware device for SOA. Its configuration-driven approach eliminates the need to program and provides a quick and easy way to configure and secure services. It offers streamlined processing of multistep transactions—including simplified configuration of common ESB scenarios, such as looping, branching, fan out and parallel processing—to help you build your ESB solution more quickly. With the IBM WebSphere DataPower Integration Appliance XI50, you simply plug it in, turn it on and connect services to build your ESB solution. WebSphere Transformation Design Studio provides a



The IBM WebSphere DataPower® Integration Appliance XI50

common tooling suite for both WebSphere Transformation Extender engines and the WebSphere DataPower XI50 appliance. The XML transformations of WebSphere DataPower XI50 appliance can be modeled, analyzed and functionally tested, deploying to the appliance. Unlike WebSphere Transformation Extender, the appliance is intended for wirespeed performance with comprehensive support for Web Services security standards. It favors repetitive, simple-to-medium-complexity formats. When compliance with industry standards is a priority, you can turn to WebSphere Transformation Extender, which can leverage the memory and storage power of its host server to undertake more-intensive processing tasks and provides Industry Packs to accelerate solution delivery.

Summary

For clients considering either an ESB or a process-automation server, WebSphere Transformation Extender provides the additional function, versatility and usability to complete the solution, helping to:

- *Extend the range of message formats and protocols that can be handled.*
- *Enable validation and processing of complex data without writing script or code.*
- *Deliver tools that plug in to an existing development environment or run stand-alone, providing a studio for ESB appliances.*
- *Provide predefined industry templates for EDI, health care, insurance and financial services to accelerate development.*

For existing WebSphere Transformation Extender customers:



- *WebSphere Message Broker and the various ESBs available in the WebSphere portfolio provide event-driven services for transformation with high availability, scalability and performance, and a broad range of connectivity options.*
- *WebSphere Process Server provides event-driven business process automation and choreography along with ESB capabilities, all based on the high availability and scalability of a WebSphere Application Server core.*

WebSphere Transformation Extender is the primary data-transformation and content-validation engine for the WebSphere portfolio. With WebSphere Transformation Extender, you can eliminate the need for custom programming for integrating and transforming data, regardless of complexity. You have the capability to respond quickly and with flexibility to rapidly evolving, strategic

business requirements. You have the ability to leverage the value of your data, applications and enterprise systems to achieve their full potential.

For more information

For more information about WebSphere ESB and BPM solutions, visit:

ibm.com/software/integration

For more information about WebSphere Transformation Extender, visit:

ibm.com/software/integration/wtx



© Copyright IBM Corporation 2008

IBM Corporation
Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
03-08
All Rights Reserved

IBM, the IBM logo, ibm.com, CICS, ClearCase, DataPower, IMS, Rational, System z, WebSphere and z/OS are trademarks of International Business Machines Corporation in the United States, other countries or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product and service names may be the trademarks or service marks of others.