

Transformer - the foremost message transformation tool available for today's financial institutions

Contents

1. Transformer Overview and Background

[Overview](#)

[Trace Financial and Messaging](#)

[Trace Financial Background](#)

2. Transformer Features and Benefits:

[Object concepts brought to messaging](#)

[XML everywhere - Common approach to XML and non-XML](#)

[Line of business message sets e.g. SWIFT ISO 7775 & 15022, FIX](#)

[Sophisticated Design Workbench for the business analyst](#)

[Pure Java open architecture for easy deployment everywhere](#)

3. Transformer Components and Processes

[Message Definition](#)

[Definition of Mappings](#)

[Test Facilities](#)

[Impact Analysis](#)

[XML/Non-XML Interoperability](#)

[Transformer & the dictionary based enterprise](#)

[Easy deployment everywhere](#)

[Transformer and web services](#)

1. Transformer Overview and Background

Product Overview

Transformer is Trace Financial's new pure Java and XML based data dictionary transformation tool aimed at sophisticated enterprises that have complex messaging requirements. It streamlines the administration and maintenance of the transformation process and enables compliance with new messaging standards in a fraction of the time traditionally required.

Trace Financial and Messaging:

Trace Financial have been delivering innovative solutions to complex financial messaging situations for over twenty years. Transformer is the combination of this experience with the latest ISO messaging concepts plus Java/XML/Web services technologies. All of this functionality is delivered in a super look and feel GUI that empowers the business analyst to take control of business messaging for the enterprise. **Trace has an unsurpassed six years Gold accreditation under the SWIFTReady EAI program.**

About Trace Financial Ltd:

Trace Financial Limited, a wholly owned subsidiary of Trace Group plc, is a specialist in the provision of application integration software, corporate actions processing and real time trading solutions for the financial markets. The company has implemented Cloverleaf®*finance* globally as a solution for investment banks, commercial banks, retail banks, broker dealers, asset managers and pension funds. Trace Group plc, established in 1974 and listed on the FTSE, is one of the UK's leading providers of IT solutions for the financial, insurance, reinsurance and property markets. www.tracegroup.com or email info.transformer@tracegroup.com

Cloverleaf is a registered trademark of Quovadx Inc.


[Back to Contents Page](#)


[Back to Contents Page](#)


[Back to Contents Page](#)

2. Transformer Features and Benefits:

Object concepts brought to messaging

Transformer makes the enterprise dictionary a reality and provides an elegant re-usable solution to the most complex of messaging situations.

- Transformer delivers a dictionary based approach for message definitions and mappings allowing maximum re-use, minimising errors and managing complexity with power and elegance.
- Users can build their own in-house Data Dictionaries - as many as they wish.
- Users can also produce point to point mappings.
- Transformer makes a reality of the “Syntax-Neutral” enterprise dictionary.
- It allows the definition of messages in terms of underlying business object types and automatically handles the formatting of both incoming and outgoing messages to these business object types.

XML everywhere

Transformer enables an enterprise XML messaging strategy that encompasses all legacy systems and external networks no matter what messaging standards they support.

- All enterprises have to deal with proprietary non-XML as well as XML based messaging. Transformer uses a common information structure model for XML and non-XML messages, so:
 - there is a separation of syntax and semantics, and
 - **any** message can be expressed in XML syntax.
- With Transformer all message data is available in an XML form **without doing a single mapping** and all message definitions (including proprietary definitions) are automatically available as an exported XML schema.
- The Enterprise can operate in a pure XML way with all messages sent or received as their XML equivalent.
- Transformer insulates users from non-XML situations.
- XML/non-XML inter-operability is a reality.

Line of business message sets

Transformer meets the messaging needs of any enterprise in any industry. We have implemented parsers and message definitions for message groups that deliver immediate added value for wholesale financial institutions.

- Trace specialises in finance industry messaging such as SWIFT, OMGEO and FIX etc.
- Transformer can easily import any XML schema for industry standard message sets.
- Transformer is packaged with a set of parsers for processing industry standards including SWIFT and FIX.
- Transformer also supports generic definitions including Fixed-Width, Comma Separated, Hierarchical as well as XML.
- This means **any** message definition can be defined to Transformer.
- Furthermore, Transformer is fully extensible and allows users to define their own parsers and functions.

Sophisticated Design Workbench for the business analyst

- Transformer’s intuitive Design Workbench allows a business analyst to build and view dictionary definitions as well as the mappings between definitions.
- Exposure to message syntax is minimised. This means that Transformer can be used by business analysts.
- In addition to the extensive set of supplied routines, users can define their own functions and rules to be executed as part of the parsing and mapping process
- The advanced Design Workbench allows for ease of use in viewing, navigating and maintaining message definitions and their components.
- The speed at which definitions and mappings are built allows Transformer to deliver a rapid and reliable time to market for new services or changes to existing services.

- Messages become more understandable to business users. You use <SecurityId> for Security Identifier and not ':35B:'.
- The Design Workbench also includes powerful testing facilities and tools to assist impact analysis.

Powerful testing tool -

- The user can configure and retain test data as part of the configuration process.
- Transformer will compare test results with pre-configured expected results and report on discrepancies.
- Over time the user will accumulate a wealth of test situations for automated system or regression testing.
- Risk is reduced. Confidence and quality are improved.

Reference View/Impact Analysis -

- Transformer provides the ability easily to assess where component definitions are used in the project thus helping to determine all the implications when changes are required.
- Comprehensive definition of the change process requirements reduces risk and improves project quality by giving confidence that all impacted areas have been highlighted and addressed.

Pure Java open architecture for easy deployment everywhere

- Transformer is Pure Java technology and is fully J2EE compliant.
- Transformer has a small footprint with platform independent deployment.
- Integration with J2EE compliant Web services, EJB, JMS is easy.
- Transformer can also be integrated with in-house message brokers as well as with Trace's own Cloverleaf®*finance* message broker.


[Back to Contents Page](#)


[Back to Contents Page](#)


[Back to Contents Page](#)

3. Transformer Components and Processes

Message Definition

Definition of messages is the fundamental building block of any messaging infrastructure.

Typically this means coping with a wide range of legacy proprietary message structures as well as more modern XML based standards.

Transformer is designed to work with all your messaging requirements, XML based and non-XML based, in a common way. Transformer will automatically render any non-XML message definition or message instance with its XML equivalent. **This makes Transformer an ideal tool for the enterprise struggling with the co-existence of non-XML technology whilst strategically standardising on an XML approach.**

Transformer message definition support goes much further than this. The concepts of 'vocabulary management'; of 'ISO15022 standards'; of 'Enterprise Data Dictionaries'; are all delivered by Transformer.

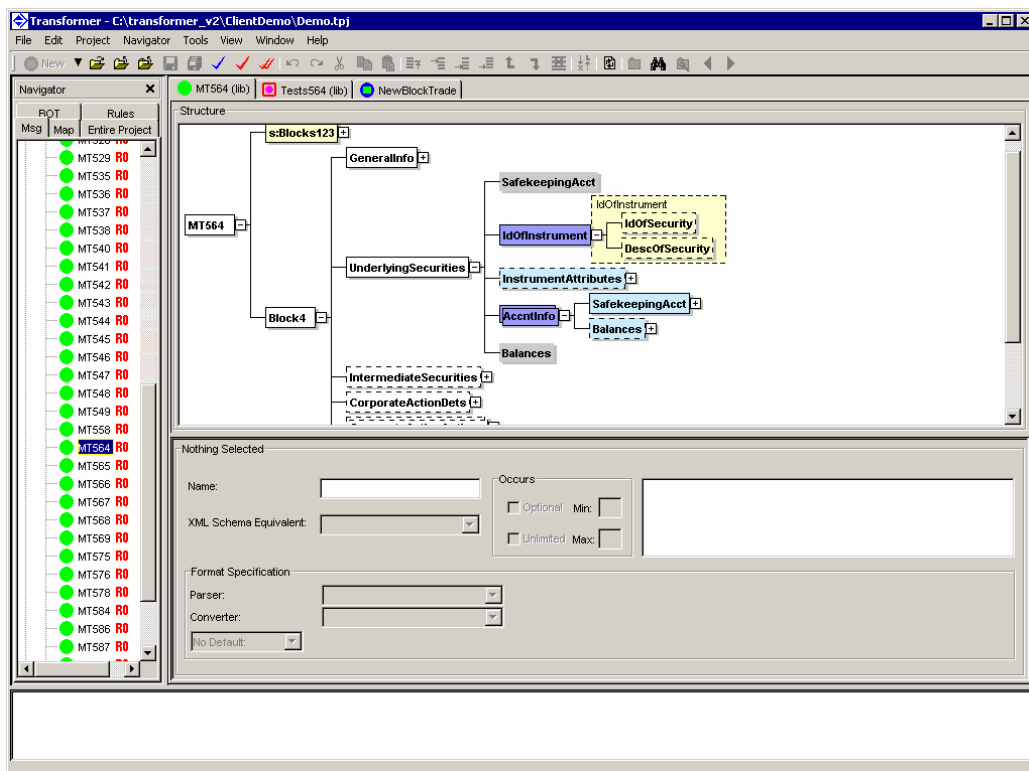
In addition concepts such as data typing and objects and re-use are fundamental. This means Transformer copes with the most complex of messaging constructs in an elegant and powerful way.

Schema import/export facilities allow Transformer to be used as the authoring tool by those responsible for maintaining the enterprise dictionary schema.

Transformer can also be used alongside industry standard version control tools for safe and secure management of development, test and live deployments.

For the finance industry certain message definitions are available as XML schema and these can be imported directly into Transformer. Other message groups such as SWIFT (15022 and 7775) and FIX are available from Trace. Furthermore Trace maintains these message definitions as part of its ongoing commitment to this marketplace and as part of its provision of SWIFT ReadyGold accredited messaging solutions for the past six years. Additional message groups and parsers can be added by the users themselves or developed for them by Trace.

By way of illustration this screen shot shows the definition of the MT564 SWIFT message within Transformer and makes use of the component and re-use facilities.



Definition of Mappings

Years of experience implementing complex messaging systems has taught us that any graphical tool has to do a lot more than draw a few lines and give the impression that the mapping process can be simplified.

Any user seduced into thinking that an application will remove complexity will fail to ask the right questions and only find out the hard way that the lack of sophistication of most products actually makes them unusable.

Transformer mapping features deliver all the benefits of re-use combined with all the flexibility for integrated user extensions (in Java) to ensure that any mapping requirement can be accommodated. Transformer will future-proof organisations where mapping requirements will increasingly become a strategic issue demanding a strategic approach.

Transformer's Mapping Definition GUI allows a mapping between messages or components to be defined in terms that a business analyst understands, with full validation and help at every stage. The design and validation is proactive at design-time, starting with the business analyst choosing the type of message or component involved. This contrasts with the way in which transformation rules can be written as an XSLT definition, which is essentially a reactive definition aimed at interpreting a document presented at run-time. XSLT is technically oriented, Transformer is oriented towards the business analyst.

A further illustration of this orientation is the fundamental Transformer concept of business object types. Mappings and condition tests work at the business object level. E.g. Transformer knows that the amount 12345678,9 in a SWIFT message means the same as 12,345,678.90 in another standard, and that the date 20040301 in one standard is earlier than 01/04/04 in another standard. There is no need to think of the mapping in terms of characters - the conversion between business objects and the characters that represent them happens automatically when a message is read in or written out. The technical issues are resolved automatically and the user is left to focus on the business significance of amounts or dates.

A sample of the mapping definition screen is shown below.

The screenshot displays the Transformer Mapping Definition GUI. The interface is divided into several panes:

- Navigator:** Shows a tree view of the project structure, including 'MappingDefinitions' and 'LookupTables'.
- From Data:** Lists source components such as 'BrokerBlockRef', 'CounterpartyRef', 'OriginalRef', 'BrokerId', 'Counterparty', 'DealerCode', 'BuySellIndicator', 'Security', 'SecurityDescription', 'IssueType', 'BrkQuantity', 'Price', 'BrkDealType', 'Conditions', 'BrkTradeDate', and 'BrkTradeTime'.
- Action:** A table of mapping actions. The selected 'Copy' action is detailed below:

Action	Details
OriginatorOfMessage	BrokerId to ...OriginatorOfMessage
RecipientOfMessage	Counterparty to ...RecipientOfMessage
Copy	BrokerBlockRef to ...#text
CopyText	TEMP-Timestamp to ...DateTimeOfSentMessage/#text
Copy	"NEVM" to ...FunctionOfTheMessage/#text
InstructingParty	Counterparty to ...InstructingParty
ExecutingBroker	BrokerId to ...ExecutingBroker
Copy	BrokerBlockRef to ...MasterReference/#text
Copy	"ISIN" to ...NumberingAgencyCode/#text
Copy	Security to ...SecurityCode/#text
Copy	SecurityDescription to ...#text
CopyText	"TRAD" to ...#text
Copy	BuySellIndicator to ...BuySellIndicator/#text
Copy	Currency to ...CurrencyCode/#text
CopyText	Price to ...DealPrice/Amount/#text
IF	(IssueType = "01" OR IssueType = "51")
- To Data:** Shows the target data structure, including 'CTMTradeLevel', 'TradeLevel', 'SubmitHeader', 'ProtocolVersion', 'OriginatorOfMessage', 'RecipientOfMessage', 'Userid', 'SendersMessageReference', '#text', '@Syntax', 'DateTimeOfSentMessage', and 'TradeLevelBody'.
- Action: Copy:** A detailed view of the selected 'Copy' action.
 - Input:** BrokerBlockRef
 - Output:** #Level/SubmitHeader/SendersMessageReference/#text
 - Description:** Performs a simple copy between a single source and destination.
 - Input:** The source of the data for a Copy or CopyText Action.

Testing Facilities

Message definitions and mappings need to be tested as they are developed.

Over time the enterprise will accumulate a growing library of tests in parallel with expected results.

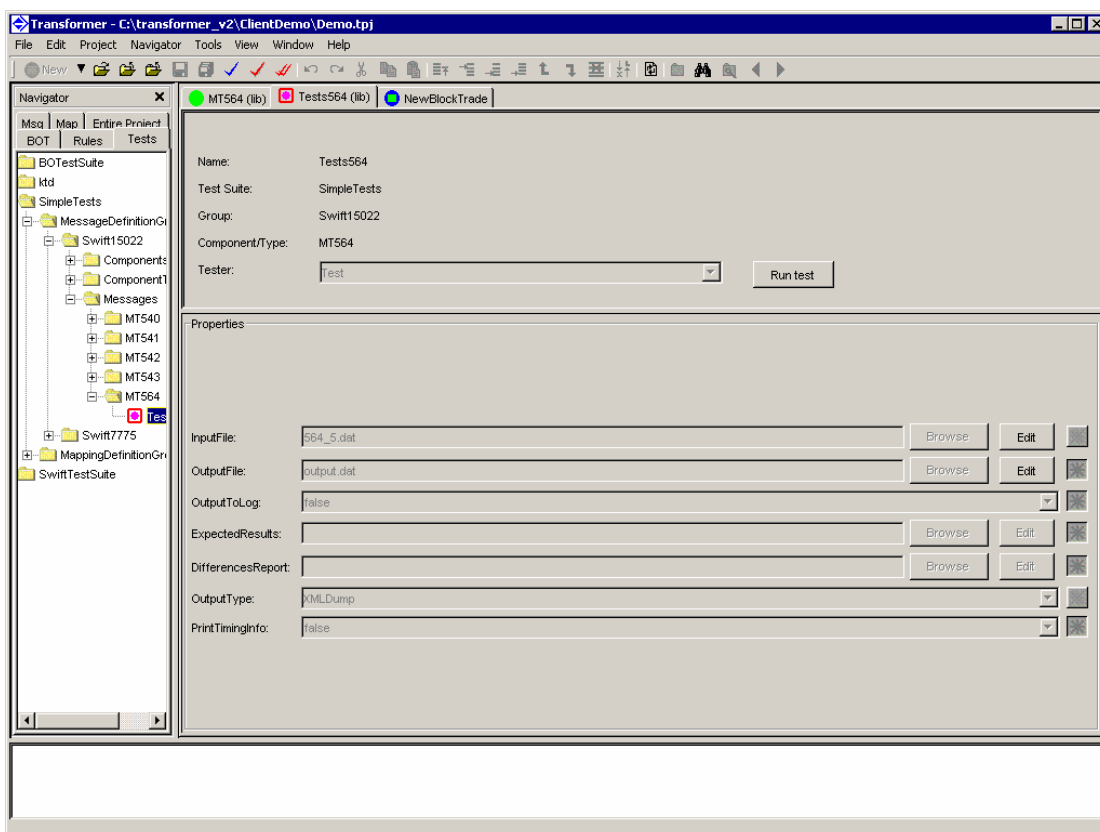
Transformer allows the user to manage this process and execute tests as an integrated part of the development cycle and within the Transformer environment.

This can include one off tests for individual message types up to suites of regressions tests for whole message groups.

The infrastructure within Transformer allows the user to deploy and deliver a quality service wherever messaging and transformation are needed within the enterprise.

This infrastructure empowers the business user to build quality components and combine this with the maximum re-usability of these components. These two ingredients will greatly assist analysts charged with delivering messaging solutions for the enterprise.

This snapshot of the testing screen illustrates some of the facilities available.



[Back to Contents Page](#)

[Back to Contents Page](#)

[Back to Contents Page](#)

Impact Analysis

Ongoing maintenance for regulatory or other reasons is a major feature of any messaging infrastructure and is an area that accounts for much of the effort, cost and risk associated with business messaging projects.

Within Transformer the impact of any change needed for regulatory, new developments or maintenance purposes can be assessed quickly.

Work can be accurately planned from the resulting information.

Areas needing additional analysis can be identified.

Areas to be tested can be noted.

The whole management of the change is made safer and easier.

This facility combines with the numerous re-use features: Re-use of message definitions; Re-use for creating mappings; Re-use within Java for user extensibility. The net effect is to:

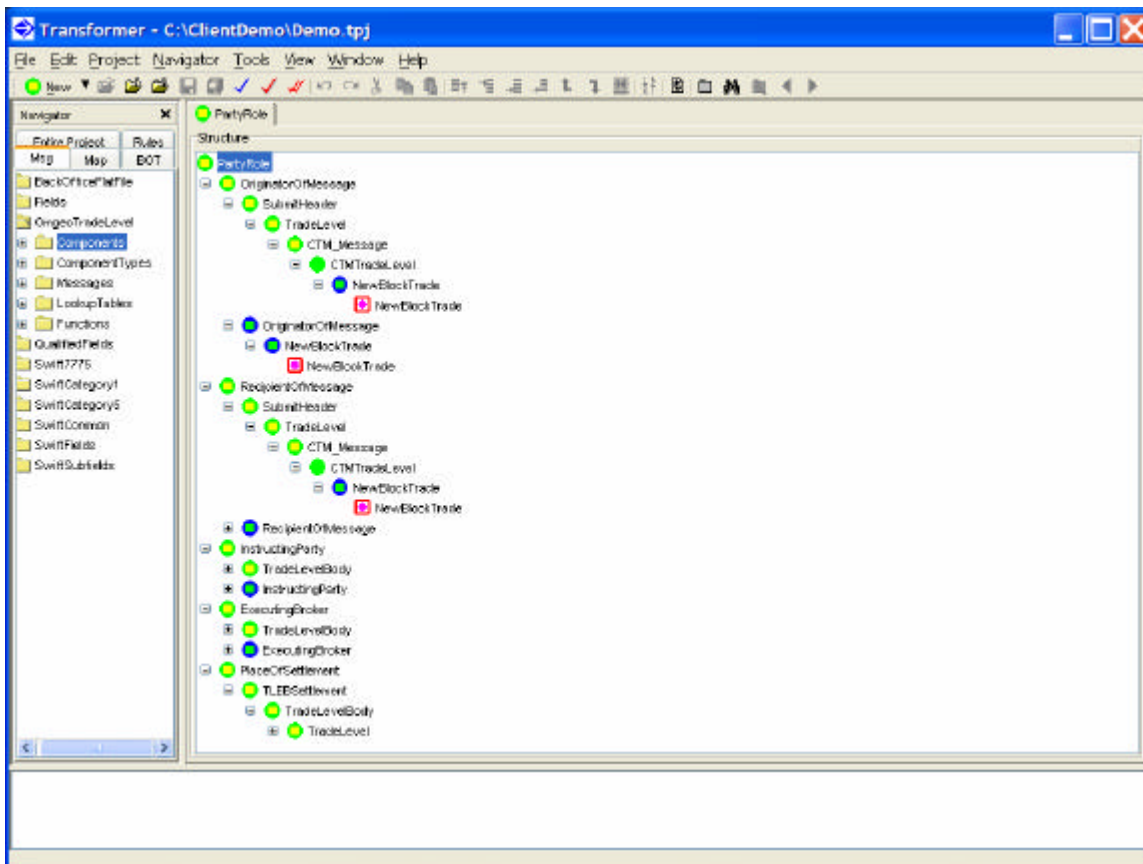
Minimise ongoing maintenance and support costs;

Minimise time to market for interfaces;

Maximise quality.

This means strategic and complex messaging infrastructures are possible and which, without Transformer, would become unmanageable, over complex and ultimately fail.

This screen shows where a message definition *component* called PartyRole is used across the OmgeoTradeLevel *message group*. The symbols indicate whether the entity is a component, a message, a mapping or a test. This display shows 'PartyRole' is used in the *component* 'OriginatorOfMessage' which in turn is used in 'SubmitHeader' etc. until we reach 'CTM_Message' which is used in the *message* 'CTMTradeLevel' which is used in the *mapping* 'NewBlockTrade' as well as the *test* 'NewBlockTrade'. The rest of the display shows other entities (components, messages, mappings or tests) that use 'PartyRole'.



Back to
Contents
Page

Back to
Contents
Page

Back to
Contents
Page

XML/Non-XML Interoperability

Trace understands that, in the real world, non-XML based systems and messaging standards are a significant feature for any enterprise. Legacy back office systems or networks such as SWIFT leave the financial enterprise with the need to accommodate both XML and non-XML messaging standards for the foreseeable future.

It is a fundamental design concept of Transformer that the user must be able to handle XML and non-XML messaging in a common way.

If an XML schema is available for a particular message group then this can be automatically imported into Transformer.

For non-XML structures or ones where an XML schema is not available then the message structure can be defined using the Transformer GUI. Once the definitions are created then the Schema Export facility allows the definition to be produced as an XML schema (even though the original message definition may not have been XML based).

This immediately provides the enterprise with a full XML representation of all its messaging standards - both proprietary and XML.

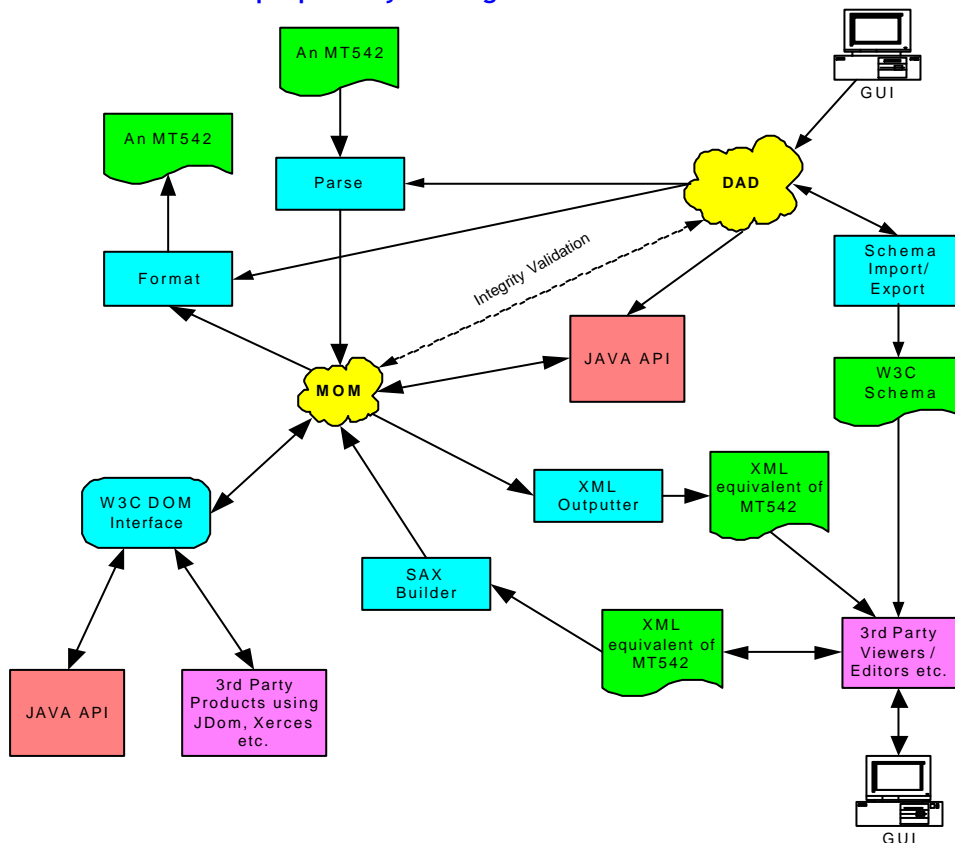
In the following diagram one can see how the Transformer architecture facilitates XML and Non XML interoperability.

The DAD (Data Dictionary) is where the message definitions are held;

The MOM (Message Object Model) holds the actual instances of particular messages;

When an actual message (in the example below we have used a SWIFT MT542 message) is received, the MOM will allow the same message to be output as an XML message consistent with the schema available from the DAD. This, of course, is a two way process; send Transformer an XML formatted message and the Transformer runtime will output the correctly formatted proprietary message structure.

This facility allows the enterprise to conduct all its messaging using an XML standard. Transformer insulates the users from proprietary message standards.



Transformer & the dictionary based enterprise

Transformer meets the entire messaging needs of a dictionary based enterprise or an enterprise wishing to exploit the benefits of a dictionary based approach to message strategy.

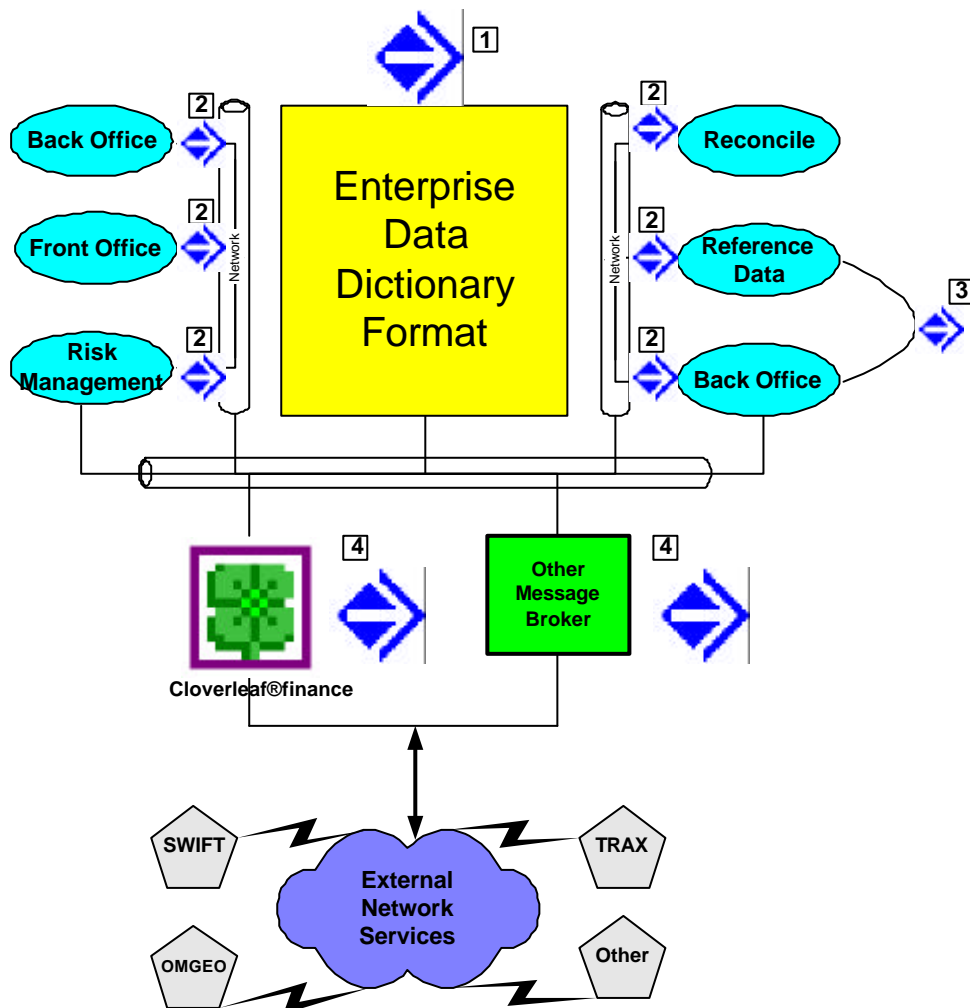
These benefits are significant and include:

- Many fewer interfaces when compared to point-to-point connectivity;
- Legacy systems are more easily replaced with 'best of breed' systems because interfacing is handled independently of the business systems;
- Departments find it easier to exchange data because they are all working to and from common standards;
- Data becomes more meaningful information because it is expressed in terms of business objects rather than in terms of technical message formats.

The diagram below illustrates the differing roles that Transformer can play within the enterprise messaging environment:

1. Transformer is the authoring tool for the enterprise schema;
2. Transformer is the transformation tool for in-line use with internal systems to/from standard dictionary formats;
3. Transformer is the transformation tool between system A and system B;
4. Transformer is the transformation tool alongside batch systems or systems handling external connectivity such as Cloverleaf finance or other message broker.

Transformer in the Enterprise



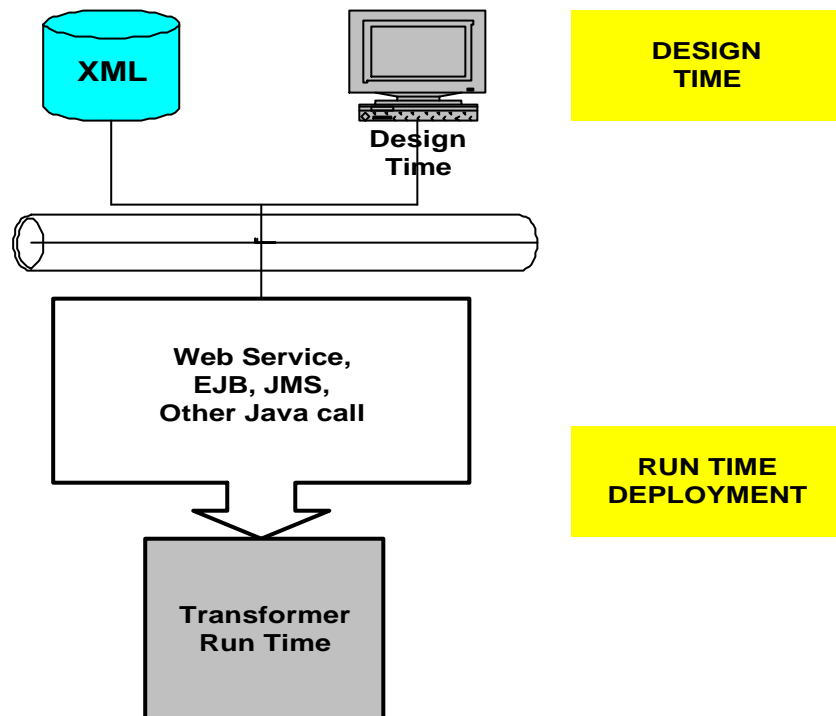
Easy deployment everywhere

Another key feature of any strategic messaging solution is that it should be deployable easily and everywhere.

Transformer operates in two modes:

Design time for the messaging analysts who are sophisticated users and understand the power of Transformer's features as well as the complexity of the messaging systems sufficiently to define message definitions and mappings.

Run time deployment which comprises Java classes and XML metadata which can be deployed to departments for integration with their local infrastructure including web service, J2EE or other architecture.



Transformer and web services:

A deployment option for the Transformer Runtime Component is as a Web Service on any Application Server that provides support for Java 1.4.2 or later.

This allows interoperability with client applications (both browser and non-browser) and other Web Services running on Java-based or non-Java-based platforms.

Client requests and Web Service responses use the standard XML-based SOAP protocol over HTTP (or HTTPS). Use of SOAP gives the possibility of using digital signatures for authentication and HTTP (or HTTPS) reduces problems with access through firewalls.

This method of deployment also gives the added value of scalability and security provided by the Application Server. The deployment is achieved using a wrapper to interface with the Application Server that will make the appropriate calls to the Transformer Runtime API.