

# **Metabula DataBridge Overview Paper**

*Metabula Limited, July 2004*

*Intentionally Left Blank*

## Introduction

In today's world, information is king. New technology is emerging at an accelerating pace and software consumers are increasingly a) more aware of the need to innovate, and b) more confused about how to put a solution together. The IT structure in which these users operate has evolved to solve specific problems over time. This evolution has resulted in the accumulation of out-of-date software and hardware solutions, which cannot easily be replaced.

The connectivity supplied by the Internet allows businesses to improve the way in which they operate. eCommerce is a way in which goods and services can be supplied and managed over the Web, reaching a vast audience. eBusiness is a concept, which describes how businesses themselves can communicate and do business using the Internet.

A major problem to be addressed by software consumers is the accumulation of multiple disparate systems (over a number of years), which use out-of-date technology. Updating these systems to use the Internet is either not physically possible, or too expensive in terms of time and cost. At the same time, these systems are needed to keep the business running and cannot be removed easily.

### *Metabula DataBridge*

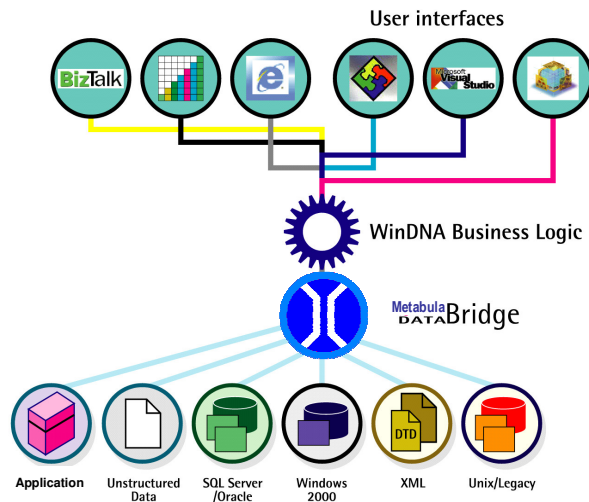
Businesses want to streamline and become more efficient and effective. Metabula DataBridge supports software consumers in their drive to a) create more open systems, and b) reach a wider audience. It provides the glue between the various systems in order to keep them running alongside a new eBusiness oriented set of systems. It allows the business to keep functioning, and provides consolidated information to be used in new solutions.

By implementing a Metabula DataBridge solution, the software consumer will have a single point of contact to the data sources of existing systems, allowing eBusiness transactions to plug into the existing business framework. This will allow the Internet to enable increased throughput with minimal business impact.

The main features of Metabula DataBridge are summarized below:

- Non-invasive. Leave existing systems and business processes untouched.
- Single point of entry. Allow new eBusiness solutions to be built against a single interface, which distributes all necessary inter-system calls behind the scenes.
- Separate back-end from front-end. Allow new investment in data centers without affecting front-end functionality.
- Secure transactions. Allow updates to be carried out across multiple systems with confidence.
- Help identify and rectify data quality issues.
- Produce integrated and consolidated datasets from multiple sources.

Metabula DataBridge integrates and consolidates data from multiple sources in order to provide new and meaningful information for applications and web pages. The data is brought together using Microsoft UDA (Universal Data Access), a key element of the Microsoft Win DNA architecture, which is built upon the OLE DB and ADO data access methods. The results of this process are an OLE DB rowset of data, which can be manipulated in any OLE DB consumer environment (e.g. Visual Basic, Active Server Pages, and Crystal Reports).



To do this the Metabula DataBridge stores meta data about the various schemata, which constitute the physical data sources. The schemata are brought together into a single business model, which is exposed as OLE DB. Metabula DataBridge also has the ability to identify data weaknesses in the source systems using this meta information.

## Concepts

### *Multiple Data Sources*

#### The problem

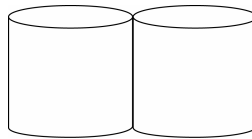
Modern businesses have evolved into major consumers of IT. This process has not always been smooth, due to the incredibly fast nature of IT advancement. In reality, no enterprise has a single master record or database for all the business objects that need to be stored. New applications need to make sure that any updates to the data are reflected in more than one place. This usually leads to either a) no improvements being added to the existing infrastructure, or b) a complete rewrite of the existing systems at a very high price.

Metabula DataBridge allows multiple data sources to be linked (joined) together as virtual tables. It encapsulates the rules, performs the joins and keeps the data structures and physical locations hidden from any consumer applications. These applications will only need to read/write from one place, therefore substantially reducing their complexity and ensuring that they are isolated from any future changes to the physical data.

#### The concept

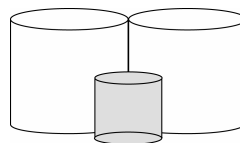
We have the same data objects stored in numerous physical locations but they can be identified as the same thing. Some items of data (e.g. 'Name', 'Age') may even be duplicated, so the source systems have overlap. Without knowing the history of the data, there is no obvious way of knowing which data is up to date and which is not.

1a. Data from multiple sources:



This is where the same data object (e.g. 'Pump P101') exists in more than one place with different sets of attributes on each source.

1b. Data from multiple sources with overlap:



This is where the same data object exists in more than one place and its attributes are duplicated in more than one source.

### An example of the problem

System A contains the following data:

ID	Name	Age
1	Andy	25
2	Fred	27
3	Tony	43

System B contains the following data:

ID	Name	Salary	Extension
1	Andrew	18000	123
2	Fred	43000	433
3	Antony	17000	762

Both systems contain information about three people. In this case a person has a unique ID. The 'Name' data is duplicated twice; therefore any data read/writes have to occur within all physical sources that contain this data. Any updates need to be carried out in a secure transaction environment in order to keep consistency across the source systems.

### A solution to the problem

ID	Name	Age	Salary	Extension
1	Andy	25	18000	123
2	Fred	27	43000	433
3	Tony	43	17000	762

The two source tables (from different vaults) need to be brought together in one 'virtual' table such as the one above. When conflicts occur (such as 'Name'), then simple mapping rules can decide on which value to use. In this solution, the virtual table always receives its data from System A in preference to System B. This rule may change according to who is viewing the data (e.g. an engineer may prefer System A, whereas an accountant may prefer System B), a technique known as 'context'.

## *Integrated Data*

### **The Problem**

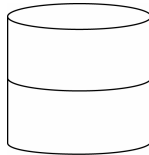
The market today is global. It is not uncommon for duplicate data to emerge as a short-term response to the request for information by consumers in different places. Whereas duplicating data is simple, the process of managing that duplicated data is extremely difficult. Real world objects of a particular type (e.g. 'Staff') may exist in more than one place. This works well for a localized view of the data. However when a higher-level view is required, bringing the data back together is costly and time consuming. Data objects may also be duplicated in different systems leading to huge identification problems (e.g. identifying which data is correct).

Metabula DataBridge allows multiple data sources to be linked (appended) together as virtual tables. It encapsulates the rules that allow duplicates to be identified and highlights areas of conflict.

### **The concept**

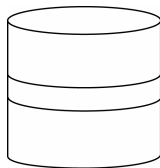
We have the same data types stored in numerous physical locations. A higher-level view is required that brings these systems together so that the data may be viewed through a single point of entry. Some data objects records may even be duplicated, in which case there is no obvious way of knowing which data is correct and which is not.

#### 2a. Integrated Data:



This is where the same data type (e.g. 'Pump') exists in more than one place but no data records are duplicated.

#### 2b. Integrated Data with overlap:



This is where the same data type (e.g. 'Pump') exists in more than one place. The two locations contain some duplicated records.

### An example of the problem

System A contains the following data:

ID	Name	Extension	Age
1	Andy	123	25
2	Fred	433	27
3	Tony	762	43

System B contains the following data:

ID	Name	Age	Salary
1	Tony	42	£32000
2	Richard	34	£18000
3	Mark	29	£43000
4	Tom	25	£24000

Both systems contain information about people, where each person has a unique ID. The 'Tony' record is duplicated in two places with inconsistent data (on the 'Age' attribute). Any read/write legacy software is unaware of the other dataset and may be dealing with out-of-date information. No higher-level view is available on all rows in all the tables.

### A solution to the problem

ID	Name	Age	Salary	Extension
1	Andy	25		123
2	Fred	27		433
3	Tony	43	£32000	762
4	Richard	34	£18000	
5	Mark	29	£43000	
6	Tom	25	£24000	

The two source tables (from different vaults) need to be brought together in one virtual table such as the one above. Reporting needs to be carried out where conflicts occur so that the database administrator can rectify them. When presenting a higher-level view, a rule can be implemented to determine which value to use (e.g. in this case, use the first value encountered).

## Consolidated Data

### The Problem

Over long periods of time, the expertise in an organization about which data is correct and which is not can dissipate. New applications, web pages and high-level reports cannot be guaranteed to produce results according to a consolidated managed data set. This seriously affects the quality of data within an organization and impairs the productivity of the enterprise as a whole.

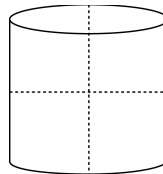
The time to re-key and re-verify the data is not available. Current live applications need to provide data today in order for the user community to make valid business decisions. Even a data set that is 95% accurate leads to a better business than one, which is not to that standard. Data needs to be consolidated (i.e. brought together in an intelligent way) to present the best possible quality of data to the user community. If there are data inconsistencies in multiple vaults, simple rules are not enough. Over time, data inconsistencies need to be reported and resolved. In the short term however, the data needs to be consolidated into the 'optional' situation given the state of the data.

Metabula DataBridge allows multiple data sources to be linked (joined and appended) together as virtual tables. Membership and identification rules can be defined to identify common objects within different sources and the characteristics that define their objects as belonging to the same 'set'.

### The Concept

We have the same data types stored in numerous physical locations. Some data objects (rows) may even be duplicated, so there is no obvious way of knowing which data is correct and which is not. A higher-level view is required to bring these systems together so that the data may be viewed through a single point of entry.

3. Consolidated data:



The same data type and data object exists in numerous different forms and locations. Complex rules are required to unite the data set.

### An example of the problem

System A contains the following data:

Name	Age	Salary
Andy	25	£10000
Fred	27	£15000
Tony	43	£20000

System B contains the following data:

Name	Salary	D.O.B
Fred	\$15000	22-Jul-75
Antony	\$15000	4-Feb-76
Richard	\$30000	23-Dec-57

These systems contain information about three people. There is no unique identifier, so they can only be identified by the values of their fields. In this case, there are two distinct objects named 'Andy' and 'Richard'. There may be one or two 'Fred' objects. Also, are 'Tony' and 'Antony' the same object? These are complex questions which need to be solved a) over time by identifying and reporting data quality issues, and b) immediately using a best guess algorithm to allow business users to make important decisions with the best available data.

#### A solution to the problem.

Name	Age	Salary
Andy	25	£10000
Fred	27	£15000
Fred	25	£12000
Antony	43	£17500
Richard	42	£37000

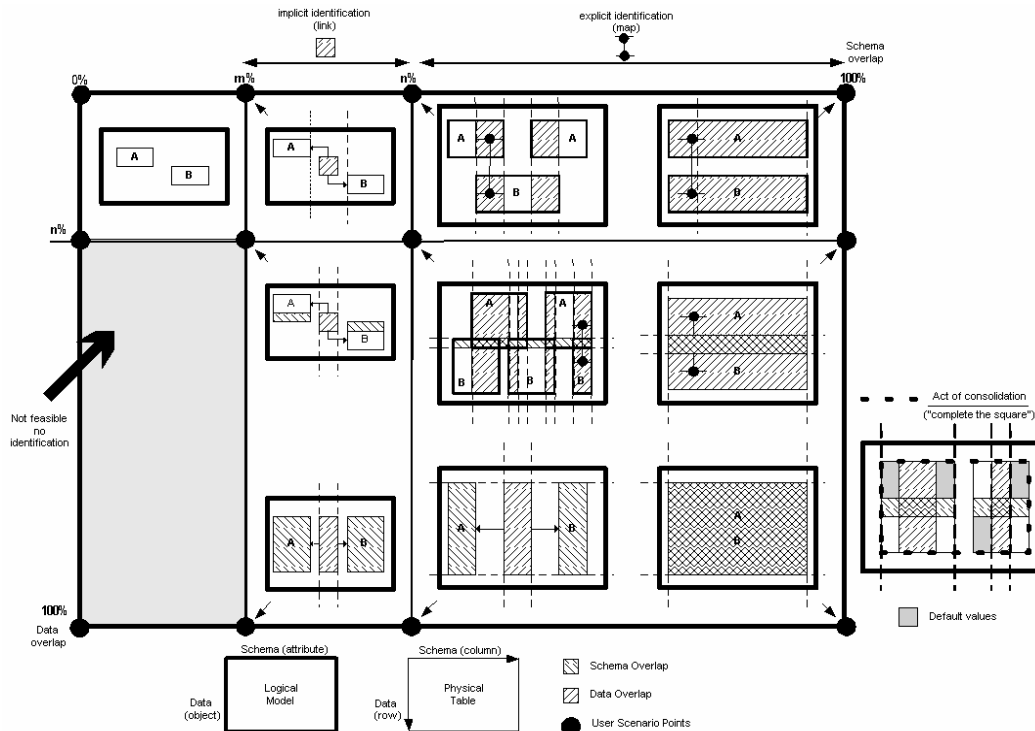
The two source tables (from different vaults) need to be brought together in one virtual table such as the one above. The data is presented in a consolidated rowset. A considerable amount of processing occurs before it is presented to the user to determine what data is correct. A number of rules are used to transform the data into the consistent higher-level view. Even this table cannot be considered completely correct, so reports need to be generated and action taken within the organization.

The rows were generated as follows:

- The result 'Age' is in years.
- The results 'Salary' is in Sterling (Dollars are converted accordingly).
- 'Andy' and 'Richard' only occur once and so are piped straight through to the results.
- 'Fred' is identified as two separate objects as both the 'Age' and 'Salary' delta values exceed the set limits.
- 'Antony' and 'Tony' are identified as a single object as the delta values are within limits. The rows are consolidated.

## Schema and Data Overlap

The diagram below is a ‘consolidation roadmap’. It explains the concepts discussed in the previous section in terms of schema and data overlap across a number of different data sources. As an example, consider the output of Metabula DataBridge to be one logical table defined by schema and data axes. The mapping task performed by the system positions each physical table within the logical space according to its ‘coordinates’ in terms of records (rows) and fields (columns).



Data from multiple sources with no common characteristics is shown as the example in the left hand corner. In this case, no bridging is possible. Data that is exactly equivalent (same objects and attributes) is shown in the right-hand corner. Between these extremes lie a series of scenarios where there is a percentage overlap of both schema (attributes) and data (objects). The overlap in schema enables identification of objects in multiple sources that are subject to data overlap.

The pure ‘multiple source’ problem corresponds with a scenario where there is 100% data overlap and sufficient (n%) schema overlap to identify equivalent objects in each source. The pure ‘integrated data’ problem corresponds with a scenario where there is 100 % schema overlap and 0% data overlap

The ‘consolidation’ problem covers all other scenarios where there is a degree of schema and data overlap. Note that in the worst case scenario (in the center of the roadmap) the problem of setting default values against individual sources becomes critical. Metabula DataBridge ‘completes the square’ to provide a ‘virtual table’ that hides the complexity of overlapping physical sources.

*Intentionally Left Blank*



## **2. DataBridge Constructor**

This application populates the meta information in the MMB vault. It provides a user interface for the configuration of the physical sources, the logical model and the mapping information between the two. It can also import and export schemata using XML.

## **3. DataBridge Inspector**

This component offers two functions: audit and inspect. Audit enables the content of a single logical class to be analyzed for duplication and formatting inconsistencies. Inspect (Not available in first release) uses the meta information in the MMB vault to run complex reporting queries across the physical source systems.

## **4. XML Toolset**

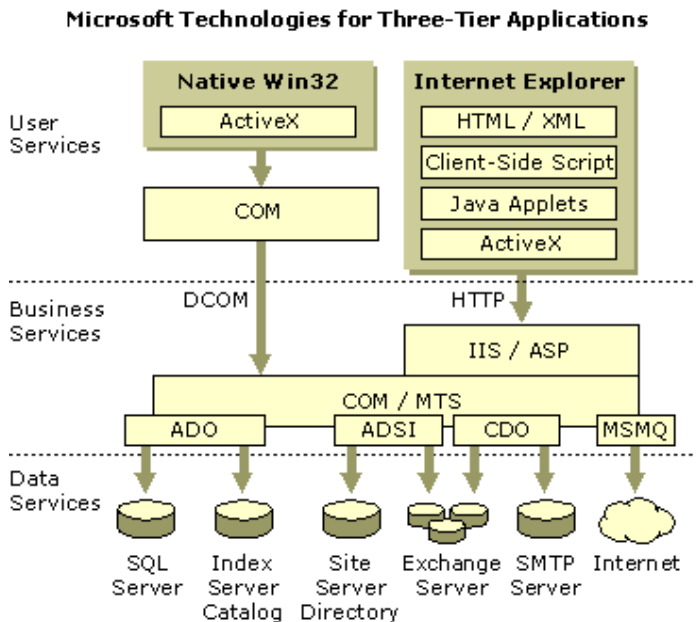
This is a number of smaller applications, which allow Metabula DataBridge to interact with the world using XML and Biztalk. In particular, support is provided for the import of logical models from Rational Rose and the export of these models as XML DTDs.

## **5. DataBridge Administrator**

A number of administration/utility applications to support Metabula DataBridge configuration and administration:

- User Manager
- Connection Setup
- Vault Status interrogation.

## Supporting Technology



Metabula DataBridge is built using Microsoft tools on Microsoft platforms. The key technologies in use are:

### 1. COM (Component Object Model)

Allows Metabula DataBridge to be built of many smaller components, which together form a whole. This means that Metabula DataBridge will not work on non-Microsoft platforms. DCOM is an extension of COM, which allows components to run on disparate physical machines. COM+ includes a few enhancements, notably MTS.

### 2. OLE DB: ADO (ActiveX Data Objects)

The data access methods used for querying the physical sources and used for querying DataBridge Engine. ADO can be used by Visual Basic or Active Server Pages to use the Bridge

### 3. ASP (Active Server Pages)

Web pages that contain server script. The server script is executed by IIS to generate HTML for the client. The server script will usually consist of ADO commands that query the Metabula DataBridge for results.

#### **4. MTS (Microsoft Transaction Server)**

Allows an environment where multiple updates can be controlled as one (so if one fails, they all fail).

Metabula DataBridge uses MTS directly.

#### **5. XML (Extensible Markup Language)**

A language that provides a common structure for sharing information across the Internet.

#### **6. MSMQ (Microsoft Message Queue)**

Allows asynchronous use of legacy systems and servers. Update requests can be queued and carried out when the server is available.

In addition, Metabula DataBridge uses an embedded Distributed Query Processor. This DQP provides transaction integrity on these sources that support two phase commit (e.g. SQL server, Oracle).

A separate paper, 'Metabula DataBridge and Microsoft's Universal Data Access Strategy', describes how Metabula DataBridge fits within the context of the Win DNA environment.

## Deployment

### *Choosing an Architecture*

The deployment architecture can change according to:

- **Size of expected workload**

If the workload is expected to be high, then the size and performance of the server machines need to be high. If the workload is expected to be small, (perhaps the system is to be used by a single person), then the software can be installed on the client itself.

- **Tiered architecture**

The client may wish to have remote clients with no software on them except a web browser. All of the Metabula software needs to reside on a designated server.

- **Expected Skill of target user group**

The users themselves may vary in skill. Skilled users will be able to issue SQL commands to the Metabula DataBridge using ADO or OLE DB. Unskilled users will use the system with web browsers and high-level tools.

- **Level of control required**

If a high-level of control over the deployment is required, then the clients should be kept as thin as possible.

A technical deployment architecture should be chosen early. The architecture is dependent upon key factors in the problem domain, which determine what software is required. These factors should be analysed by the client, with the aid of Metabula technical consultancy, in order to choose the correct solution configuration. The technical architecture may vary over time. For example, it may be required that the system will support five people for the first six months before rollout to one hundred people. The enterprise data architect may wish to construct a Metabula MetaBase (MMB) 'off-line' on their client machine, rather than a publicly viewable server machine.

## Software Configuration

The Metabula DataBridge software can be configured to perform differently in certain situations. These situations, and the options available, are the topic of this section. The recommended option is always held as the default and is automatically setup when Metabula DataBridge is deployed. These options are configured using DataBridge Administrator.

The software can be configured according to a number of factors:

- **Cost**

How much physical expense is involved in the deployment

- **Licensing agreement**

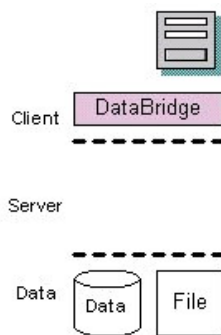
Does the client already have licensing agreements in place (e.g.: Microsoft site license)?

- **Transition policy**

Does the client allow read/write on major systems to occur?

### Single User

This option requires all of the software to be installed onto a client machine, with no server.

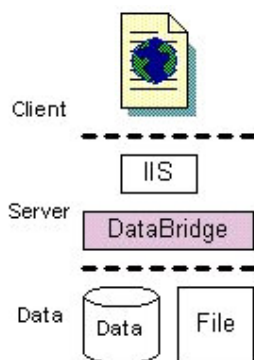


In summary, we have no server in the middle layer. The Metabula DataBridge components are installed directly onto a client (or a number of clients). The clients communicate directly with the data source via the local Metabula DataBridge OLE DB Provider. The client uses Metabula DataBridge directly through ADO, OLE DB or OLE DB aware third party products. This scenario should be common at the beginning of a project, when developers and enterprise information engineers wish to 'get to know' the software and its capabilities. At this time, the MMB is constructed with the correct mapping information.

The benefits of this approach:	The drawbacks of this approach:
<p>Easy to install.</p> <p>Low impact to the enterprise of configuration changes and testing.</p>	<p>Single client access to the Metabula meta vault.</p> <p>If more than one client installation exists, the MMB is not synchronized.</p>

### Multiple Web Users

This is the most common deployment option. The majority of Metabula DataBridge users will interact with the application through HTTP and a web browser only.



In summary, the Metabula DataBridge software is installed onto a server machine (or a number of servers). The server is running as a web server such as MS IIS, which serve/s HTML pages using ASP. The clients request and receive pages over HTTP and LAN. The DataBridge administration software resides on the server. This is the most scalable and manageable solution. The servers can be replicated as many times as is required, in order to achieve a better client performance (as long as the MMB is kept the same on all of the servers). The web server uses the application to create HTML for the client using ASP, JSP or an alternative.

The benefits of this approach:	The drawbacks of this approach:
<p>Easy to control the installation for a large number of users.</p> <p>Easy to increase the number of users supported by increasing the server capability, or by duplicating the server.</p> <p>High security as clients have access to HTML only.</p> <p>Good for unskilled users.</p> <p>Clients are platform and browser-independent.</p>	<p>If more than one server installation exists, the MMB must be the same on all servers.</p>

### *Licensing*

The way that Metabula DataBridge is licensed depends on the deployment information. There is no limit to the number of clients or client applications that can access the application at any one time. Of course, the number of clients involved may have an effect on the performance of a Metabula DataBridge installation.

The licensing model is characterized by:

- (A) Functional breakpoints: Audit
  - Inspect
  - Provide
- (B) Scaling breakpoints: Number of sources
- (C) Temporal breakpoints: Expiry date

License types can be either evaluation or full. Detection is based on a broadcast-response mechanism whereby new installations confirm whether other copies of Metabula DataBridge are running upon at startup time.

## Components

### *DataBridge Engine*

DataBridge Engine is an OLE DB provider. This means that, by itself, the software does not have a user interface. It provides data in OLE DB rowsets to consumer applications, which have to connect to the DataBridgeEngine and issue SQL commands. DataBridgeEngine also provides catalog information to clients (tables, columns etc).

Typical consumer applications include:

- Rowset Viewer
- DataBridge Constructor
- Visual Basic
- Crystal Reports

The SQL supported by DataBridge Engine is constrained against:

- Logical mode constructs.
- Mappings (read/write - the physical to logical mappings that produce the rowsets).
- Expressions.
- Physical schema constructs.

The current release offers the following SQL support:

<b>Supported SQL</b>		
<b>SQL Keyword</b>	<b>Parser Supported</b>	<b>DataBridge Supported</b>
SELECT	YES	YES
SELECT named column	YES	YES
WHERE	YES	YES
DISTINCT	YES	YES
ALL	YES	YES
GROUP BY	YES	YES
ORDER BY	YES	YES
HAVING	YES	YES
AS (identifier aliases)	YES	NO
JOIN	YES	NO
Embedded queries	YES	NO
CREATE SCHEMA AUTHORISATION	YES	NO
CREATE TABLE	YES	NO
GRANT privileges	YES	NO
CREATE VIEW	YES	NO
INSERT	YES	YES
UPDATE	YES	YES
FETCH	YES	NO
COMMIT	YES	NO
ROLLBACK	YES	NO
CREATE VIEW	YES	NO

## Supported SQL (continued)

SQL Keyword	Parser Supported	DataBridge Supported
DECLARE CURSOR	YES	NO
CLOSE CURSOR	YES	NO
DELETE	YES	YES

Future releases will support additional SQL constructs against more complex mappings.

### *DataBridge Constructor*

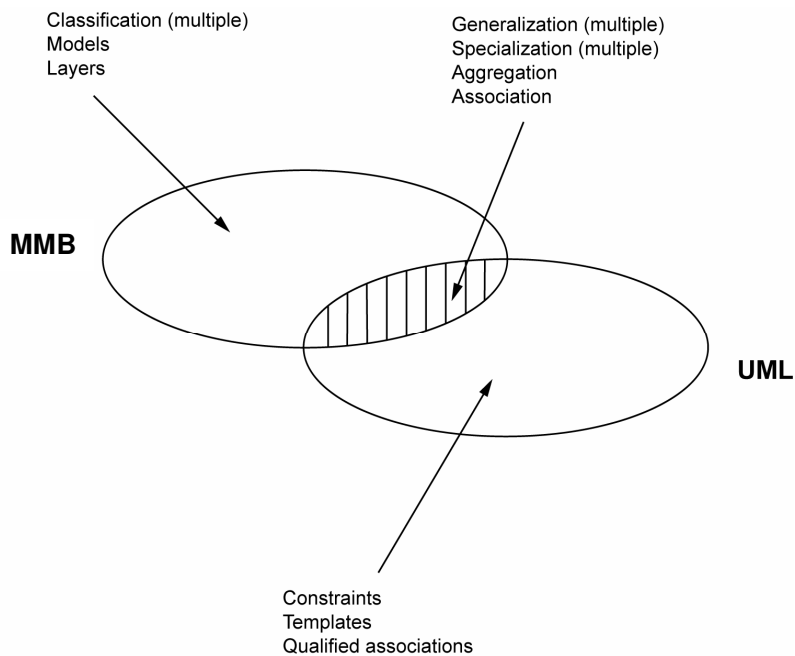
DataBridge Constructor is the major design tool for configuring the Bridge. It has the responsibility for importing physical schemata, designing a logical model and creating the mapping information between the two. BridgeConstructor should be used when deploying Metabula DataBridge or changing the MMB information.

The major function points supported are:

- Importing a new physical vault.
- Updating physical vault information. This is required for when more tables/columns are required from the physical source or when the schema information has changed in some way (e.g. path).
- Viewing the physical data. SQL queries can be issued against the physical sources.
- Exposing a model. Create new subclasses in the logical model window.
- Create physical-to-physical mappings. Create join information by dragging one physical field onto another physical field.
- Create physical to logical mappings. Create the mapping information by dragging physical field onto the logical field.
- Adjust the logical attribute weightings. Change the preferred physical field from the popup menu on the logical attribute.

The logical model that will be supported in this release of DataBridge is that defined by the UML standard for Class Diagrams\* rather than Metabula's MetaBase (MMB)

The relationship between UML (class diagram) and MMB is shown below:



\* UML State Transition and Interaction models are not supported in this release.

### *DataBridge Inspector*

This application is used for identifying data quality issues by using the meta information to query the physical source information. It has two aspects: audit and inspect (not available on first release).

It provides audit functionality on the content of a single class with respect to:

- Duplication - write sets of columns
- Formatting - lead, trail, alpha, numeric
- Coding - against lookup tables
- Distribution – nulls, singletons, counting of discrete sets
- Pattern identification

Inspection functionality will include:

- Physical Vaults
- Physical tables
- Physical columns
- Logical classes
- Logical attributes
- Identifying logical attributes that are not linked to an entity in the physical schema
- Identifying inconsistencies in data content
- Identifying duplications in data content
- Identifying consolidation requirements in data content.

In addition guidelines for development of OLEDB consumer applications that can perform repair processing via Metabula DataBridge are available.

### *DataBridge Administrator*

DataBridge Administrator should be used when you need to configure the connection information used by the Metabula DataBridge components in order to connect to the MMB.

The information stored is:

- Host, Port, Username, Password, Ownership. The connection parameters used to connect to the meta vault
- Connection string. Used by DataBridge Constructor only when connecting to the DataBridge Engine.

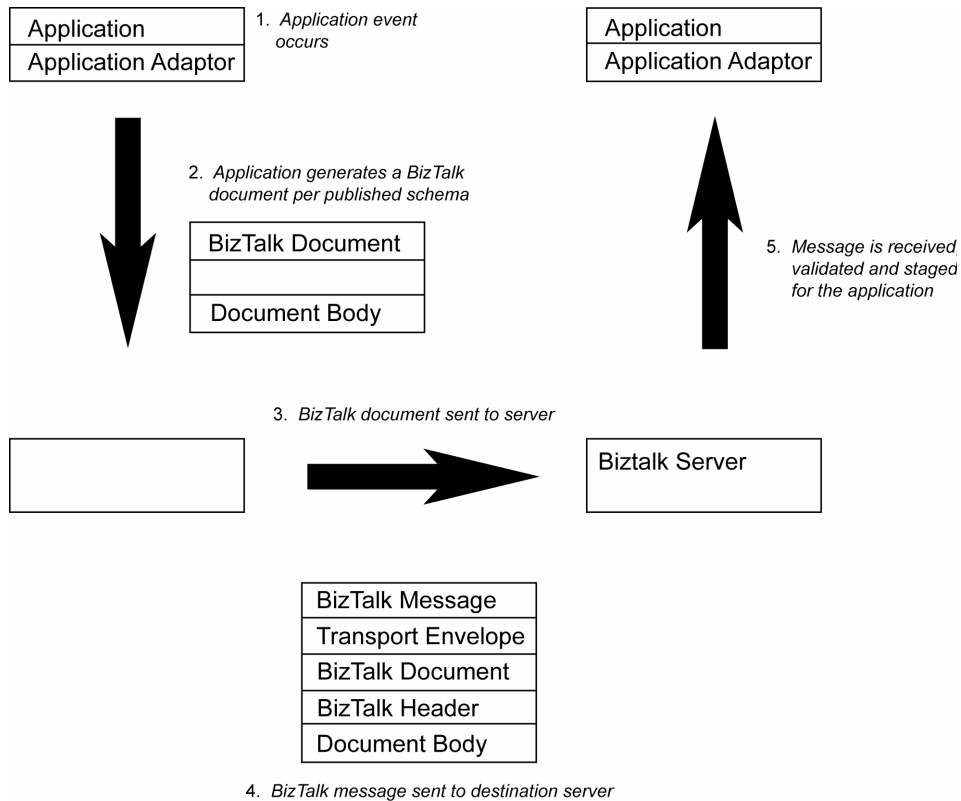
These values are read when the application starts and saved when the OK button is clicked. If you change the connection parameters, a 'Test' button is provided to check that they are correct. Vault online status can also be verified with this component. DataBridge Administrator also provides the user manager configuration functionality that enables users define against the logical model to be linked to users defined on physical sources.

## XML ToolSet

This toolkit supports the import and export of UML logical models in XML DTD format. XML tags can be a subset of those defined by the BizTalk framework or defined by the OMG standard. Data (Object) content can be exposed through an XML interface against specific (restricted) DTD's. These DTD's define 'clusters' of data that extract specific sets of objects from Metabula DataBridge that correspond to specific business views.

## Integration with BizTalk

The diagram below shows the basic processing behind the BizTalk framework:



The format of document bodies (encoded in XML) within BizTalk is domain-specific (it is not possible to invent a totally generic BizTalk solution for everyone's business needs). As with other methods of application integration, there has to be some unique code written for every application of BizTalk.

Separate applications can be written that can take XML files generated by Metabula DataBridge and wrap them with the required BizTalk extensions to produce a BizTalk document. This document could then be sent to a receiving BizTalk node using an appropriate transport.

## References

- Metabula DataBridge and Microsoft's Universal Data Access Strategy.
- The Core Meta Model: A Comparison With UML.
- Layers and Models in a Metabula MMB System