Table of Contents

Data Storage

Any data management application needs to store data. The data in **Aware IM** is encapsulated in business objects (see Business Objects as Carriers of Data) and so **Aware IM** needs to store instances of business objects.

In **Aware IM** most business objects are stored in a database. Database management systems (DBMS) are complex software and development of applications using traditional methods requires a lot of effort dedicated to the implementation of database-related issues, such as a thorough design of database tables that includes relationships between the tables; interface to database management software and database transactions.

Aware IM takes care of the database-related issues automatically

Aware IM performs database-related tasks automatically behind the scenes. When configurators define business objects, attributes, queries, etc they do not need to worry about database tables and relationships between them. **Aware IM** automatically creates database tables when new business objects are defined and automatically alters them when object definitions change. When object definitions do change **Aware IM** makes every effort to preserve the existing data whenever possible. For example, when a new attribute is added to a business object the existing data for this business object is not affected in any way or when the attribute type is changed from Number to Plain Text the existing numbers are automatically converted to text. Obviously when an attribute is deleted from a business object the existing data for the attribute is deleted as well. In any event, if the existing data is going to be affected **Aware IM** warns the configurator about the possibility of the data loss and indicates which configuration changes will affect the data (see also Publishing Business Space Version]).

In the Operation Mode when users create or modify instances of the business objects **Aware IM** performs all the database-related tasks behind the scenes creating and modifying rows of data as required.

Aware IM supports different data types.

Aware IM supports most data types including images and documents, small and large text, dates, numbers, binary data etc. Again configurators need not be concerned with the actual representation of these data types in a specific DBMS – **Aware IM** takes care of all these issues.

Aware IM is designed to be DBMS independent.

Aware IM is designed to be independent of a particular database management system used by its clients. Due to quite significant differences in capabilities and functionality of different DBMS implementations **Aware IM** does include the code specific to a particular DBMS. This code however is

well isolated and supporting more database vendors is just a matter of time. At the moment **Aware IM** has been tested with IBM Cloudscape/Derby, MySQL, Microsoft SQL Server/SQL Server Express/Azure, Oracle, PostgreSQL and Maria DB. More database systems will be supported in the near future.

Aware IM is designed to support different data storage media.

Database is not the only place where business objects can be stored in **Aware IM**. **Aware IM** abstracts the concept of the data storage so in principle business objects may be stored in other types of data storage media, such as memory or LDAP. At the moment persistence in memory only is supported (apart from persistence in a database). Memory persistence can be useful when business objects are created for temporary purposes only and are not stored permanently in the system. For example, an instance of a business object may be created only to trigger business rules that perform some calculations attached to this object (see Business Rules as Carriers of Business Logic). Configuration of a persistence type for a business object is described in the Specifying General Properties section.

Aware IM takes care of transactions and data integrity issues.

Aware IM makes every effort to preserve the integrity of data – for example, when an instance of a business object is deleted, all references to this instance in other business objects (if any exist) are deleted as well. Also if some data operation fails all data operations that occurred prior to the failed operation within one logical request are automatically rolled back, so the data stays consistent. For example, if money is transferred from one account to another one in two steps (withdrawal and deposit) and the deposit step fails for whatever reason, withdrawal is rolled back as well. This multi-step operation is called a *transaction* in database systems. For more details on how **Aware IM** handles transactions see Rules and Transactions.