



**AUTOMATED CONVERSION TECHNOLOGIES OF INFORMATION SYSTEMS**

**Go.Net**

## **100% Automatic Conversion of Magic Applications to Microsoft .Net Environment**

The advanced, unique technology developed by *Go Up Technology* enables to perform completely automatic conversion of Magic applications in any version to Microsoft.Net.

Automatic conversion permits the application methodology to be retained, while enabling redesign of the application's architecture and GUI.

***Go Up Technology* enables complete automatic conversion in a quicker, shorter, more effective and more economical process when compared to alternative Solutions.**

- Completely automatic conversion, with no additional manual work required.
- Combine optimization with in the conversion process to improve the new system.
- Embed Advanced capabilities in the converted application during the conversion process
- Reduce maintenance costs as a result of standardization of applications.
- No more dependence on licenses.
- Automatic conversion means that the system can be operational very quickly.
- The new converted system is similar to the old one, so assimilation is quick and easy.
- The new code is readable according to all rules of development in .Net platform, efficient and easy to maintain

## ***Content of the Conversion***

### **1. Method**

Converting information systems developed with MAGIC Software to Microsoft. Net framework requires thorough preliminary work and the preparation of efficient methodologies.

Since we already have an existing working application there is a need to preserve the business logic of the existing system.

The Conversion process will be performed in several stages

#### **1.1 Consolidation of the Technological Standards**

Consolidating the technological standards according to which the Conversion process will be performed. That, together with the organization personals and with an agreement on the combined frameworks, available libraries and the user interface libraries that will be used in the Conversion process.

#### **1.2 Learning the Application in Magic Environment**

Meeting with Magic's specialists and learning of the system, its functionality, the System's interfaces, acquaintance with the system's database and its information Security system, as well as identification of the programs that are not in use, and All the information we can get about the application.

#### **1.3 Deciding on application module priorities**

Dividing the application into different modules and prioritizing according to the Correlation of the modules, the testing stages and the inception of working in The new environment.



## AUTOMATED CONVERSION TECHNOLOGIES OF INFORMATION SYSTEMS

### **1.4 Taking Care of the Database**

Converting the database to SQL environment (if needed), Implementation a Relational schema in the database, reclamation the connections between the tables, and normalizing the data and re-indexing the tables.

### **1.5 Pilot (optional)**

There is an option to Implementation of the pilot – in accordance with the standards and libraries that were consolidated , we implement a pilot and present it to the customer target personnel.

### **1.6 Converting the Application**

Converting the application in accordance with the standards and libraries that were consolidated, while implementing notes taken in the pilot phase.

### **1.7 Testing**

Acceptance tests that will be performed by programmers and framework professionals of the organization will follow the initial test performed by us. After debugging the hitches that were discovered during those tests, the system will be transferred to users' tests.



## **2. Automatic Conversion Process**

During the conversion process, automatic tools, Developed by *Go Up Technology*, will be used. Those tools are based on unique technology that was developed by our company. The technology enables the decoding of the Magic application source code in an absolute manner and the creation of objects, properties and relationships between them, that will lead to learning and optimization processes during the conversion stages. The rewriting process of the new code is performed regardless to the standard of the old code, and apart from preserving functionality, the automatic tools produce standard code, clean of errors and easy to maintain.

Another benefit of the automatic tools is that they are enabled to create a code for the User Interface (UI) that is easy to maintain also at the design level, and not only at the code level.

The use of the automatic tools shortens the work processes and makes them more efficient, preserves the maximum functionality, and preserves unified and intact standard of the created code.

In addition, the automatic tools assist in rewriting of each of the layers, including building the Data Access Layer (DAL), the Business Logic (BL), and of course, the User Interface and reports.

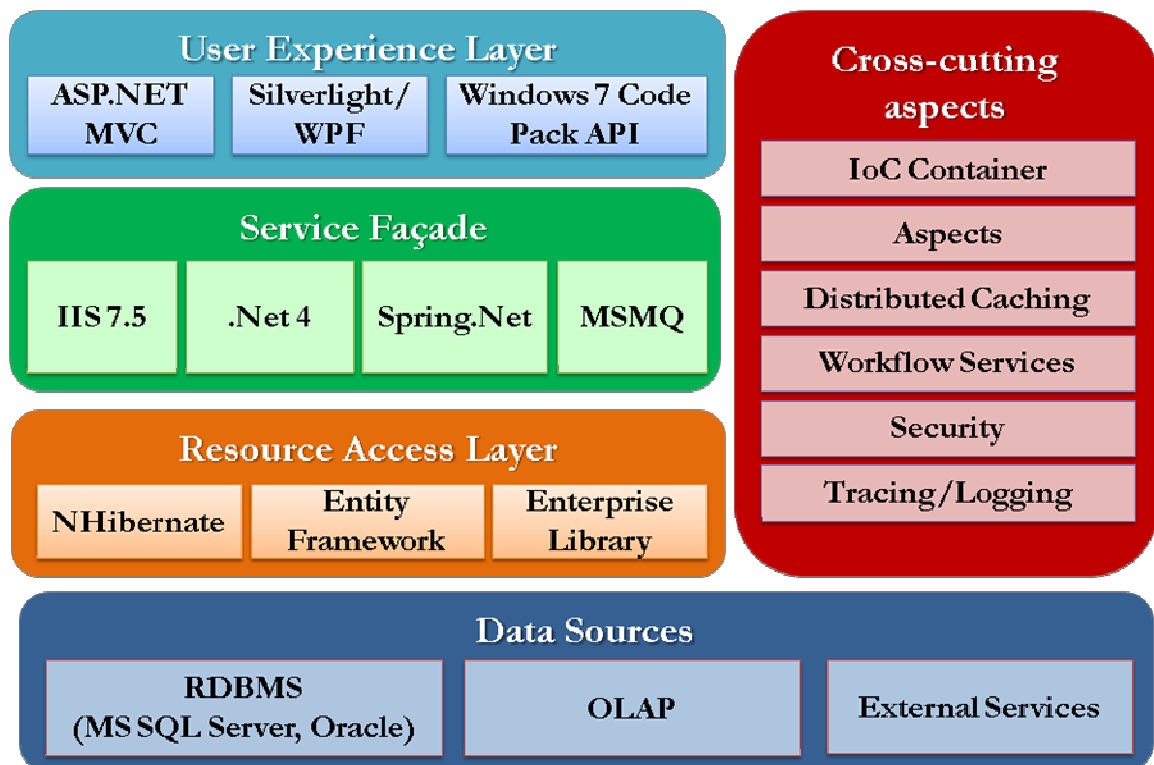
Likewise, the automatic tools are integrated in the processes of building Stored Procedures (SP) to the databases, and assist in the database analysis for the creation of a relational scheme in the database tables.

### 3. Architecture

The architecture of the converted application is based on a multi-layer model, which enables separation between the Data Access Layer, the Business Logic layer, and the User Interface layer.

This architecture enables clear distinction of the responsibility fields of the different layers, simplified the system maintenance, and is a standard in the industry of information systems' implementation. This model is also known as N-Tier Model.

diagram of the system's architecture:



We see high importance in using Architectural Patterns such as Inversion of Control (IoC), Aspect Oriented Programming (AOP), and Design Patterns, in order to implement Pluggable Architecture. Our goal is to build flexible system that allows the use of different databases, changing modules without harming other modules, easy integration and unit tests, etc.

## **4. Technology and Libraries**

### **4.1 Destination Language and the Convert Platform**

The converting will be to C# 4.0 according to all the Microsoft.Net framework methodology. while using the progressive abilities of Net 4., such as parallel programming, dynamic types, etc.

- The new code will be readable, efficient and easy to maintain.
- The conversion will be to Web environment, using one of the following technologies: ASP.Net, MVC, and Silverlight; or to Desktop environment, using Win Forms or WPF, according to the client's demand.
- Object Oriented Design (OOD) will be used to enable the code reuse, and to provide easy access for the components' maintenance.
- Using abstraction layer in access to the database.
- Using AJAX and JQuery qualities in User Interfaces (relevant only for ASP.Net/MVC), in order to create user experience that is pleasant as possible, and enable the use of many components that are available online.
- Services: in the conversion process to Net. Environment in the given model, we are getting an open system that allows us to reveal services easily using Web Services, and to provide advanced integration capabilities with other systems and environments. The services will be developed and publicized according to the accepted standards in the SOA field.

#### **4.2 Access to Databases**

The access to the databases will be done using Object Relational Mapping (ORM) layer, which enables data coupling between the data model – as is declared in the database , and its representation in the application.

In addition, ORM enables flexibility in choosing the database provider (Oracle, Microsoft SQL Server, etc.).

Stored Procedures will be used when relevant (usually in reports) in the database layer, in order to save network traffic and to enhance executions of complex pulling outs.

Equally important, there will be wide-scale use of caching technologies (First Level, Second Level, Distributed Cache), to enhance executions in working with complex entities and to minimize the number of database accesses.

#### **4.3 Application Security**

The application security in the new code is based on standard mechanisms that are available in .Net Framework. Authentication mechanism for user identification will be implemented using Forms/Windows Authentication, depends on the system running environment. In addition, Role Based Security mechanism will be implemented in the relevant placed, to create user authorizations based on users and groups. This mechanism is usually executed using standard Membership Provider, unless the client asks differently.

#### **4.4 Combined Libraries During the Conversion process**

Some Software Factories will be combined in our system model when needed. Application Blocks from the Enterprise Library of Microsoft's Patterns and Practices (P&P).

Most progressive and leading Open code libraries like:

- NHibernate for ORM.
- Spring .Net, Windsor Castle for IoC and AOP
- Common Logger for Logging will enable flexibility in choosing a provider.

The libraries combination will be performed for the most stable version at the time of execution.

#### **4.5 Downloaded Components in the client**

After wide scan of the system, if there will be a need in downloading components to the client station, that need will be taken care of specifically. The solution will be performed, if possible, using standard .Net components (User Controls), or Silverlight in case of progressive user experience, which demands unique abilities.

ActiveX technology can also be combined when needed.

#### **4.6 Reports**

There are several possibilities of reports' production using .Net environment.

- Microsoft SQL Server Reporting Services with The most updated version that enables progressive reporting abilities and wide array of possibilities including the ability of producing reports by the end-user (ad-hock reporting).
- Crystal Reports (less recommended) - producing the reports using Crystal Reports – an old-timer tool with many functionalities.
- Third-part tools available in the market.