

USoft Product Information
USoft for .NET



USoft for .NET

Rules Services for the Microsoft Environment

Combining two paradigms for efficient Solutions

When you develop, deploy and maintain business software, you are looking for fast time-to-market and acceptable Total Cost of Ownership (TCO). In this arena,

- programmers traditionally prefer the unlimited but time-consuming power of 3GL programming environments,
- while system analysts are more often in favor of declarative systems.

In reality, you need both paradigms if you really want to allow for fast application development, support large systems, limit TCO and improve time-to-market.

Business Rules Intelligence

We have a strong background in business rules processing. This is reflected in its USoft technology. Like its predecessor Rapid Application Development but with today's technology, business rules processing leads to a drastic reduction of TCO.

The strong market demand for application integration has inspired to develop **USoft Rules Services** to deliver core business rules technology to Service-Oriented Architecture.

USoft Rules Services apply business rules automatically and intelligently when changes occur in the database. Your business logic is enforced in a central location, just like your business data are stored in that central location.

USoft Definer contains the interfaces needed to configure and maintain your business rules and associated data structures.

Rule-driven support for .NET components

USoft Rules Services support .NET components¹. We support .NET because of its versatility, the completeness of the available class libraries (in e.g. C#), and the scalable architecture.

This support is based on USoft's Rule Driven Method Invocation (RDMI) technology, where methods of 3GL code or OO components can be invoked by business rules.

.NET components can be approached using web service invocations. Alternatively, they can be invoked via COM, using the REGASM.EXE tool from the .NET environment.

Application externalisation

Integrating USoft rule-based applications in a .NET environment requires application externalisation. USoft Rules Services are the building block for deploying externalised application services. Application externalisation defines what functional possibilities the application offers, and how end users are authorised to use it. Three techniques for application externalisation can be distinguished:

- SQL access via ODBC, JDBC or HTTP via a dedicated funnel account that is configured with the appropriate authentication and authorization.
- Access through an XML interface; rule-based processing of incoming XML messages.
- Web Service based interfaces (USoft 7.x).

All techniques allow re-use of existing rules. The advantage of the first technique is the ad-hoc approach with (initially) low cost. Because no application interface is formally defined, this technique requires strong communication between developers on both sides of the externalised interface.

The second technique requires initially more investment since a more formal (XML) interface is agreed upon and implemented. The versatility of USoft's support for XML and XSL makes this technique an attractive option for the USoft 6.x series.

The third technique requires USoft 7. Advantages over the second technique include the ease of use of web services (for example, the use of UDDI) and the conformance to standard web service protocols. The fact that both XSL and USoft job structures can be used for this type of application externalisation is an added bonus.

.NET connectivity to Rules Services

Deployment of USoft Rules Services in a .NET environment makes use of web services protocols such as HTTP and SOAP for the exchange of XML formatted documents. Alternatively, Rules Services can be approached by ADO/ODBC and JDBC, either for the exchange of XML-formatted documents or for direct access to application data with SQL.

Embedding Rules Services in portlets

Another powerful option is to externalize USoft applications through portlets or web parts. This time, the application is externalized in USoft-generated web pages. These web pages are end user interfaces. They may run as a stand-alone application or they can be embedded in any external web application: ready-mades (SAP, Siebel...); web applications built in enterprise environments such as .NET, Weblogic or WebSphere; message routing solutions like MS BizTalk workflow engines like SeeBeyond; content management software like MS Sharepoint.

Developing portlets on top of USoft repositories is straightforward and low-cost. Communication between the embedding environment and USoft is versatile. This approach allows fast and easy configuration of browser user interfaces with very limited programming. The portlets allow users to perform transactional application tasks such as entering new customers or processing orders. Changes to the underlying business model are automatically reflected in the portlets. This greatly enhances maintainability.

A strong combination: .NET and USoft Rules Services

The combination of .NET and USoft Rules Services has the advantage that different aspects of a problem can be solved using the fastest and most appropriate solution for each:

- .NET is an excellent environment for creating customized, attractive, and functionally rich browser-based application modules.
- USoft Rules Services greatly expedite the implementation of all data-oriented business logic.

The combination of these two paradigms gives you fast time-to-market and great-looking browser interfaces without the burden of having to write 3GL-type business logic.

Using Rules Services, business rules are easily recognized, identified and deployed. Rules are clarified and tested in a default working prototype which adds considerable design-time value.

What are Web Services?

Web services are self-contained business functions that operate over the Internet and allow businesses to offer automated services to each other. They are written to strict specifications to enable collaboration. Some of the more established functions at this stage are messaging, directories of business capabilities, and descriptions of technical services.

Increasingly, companies use web services to automate parts of their supply or fulfillment chain.

For example, a company can now let its suppliers view inventory levels and replenish stock without processing explicit purchase orders. But suppliers can also build on these basic features to provide better services to the customer.

Parties work from the same basic web services design and then add value and business advantage to meet the specific needs of their customers. Web services standards mean that many systems developers can enter the market, thus increasing competition and bringing down cost. Competition among vendors also encourages product innovation and prevents you from being locked in with a specific vendor, computer type, or software.

Web services are still in their infancy. Not all standards are fully tested, and many potential business uses remain unexplored. But companies should start planning for web services, and ask vendors for their plans to support web services.

Capabilities

We have a pool of experienced .NET, Java Enterprise and USoft developers, who work with a range of enterprise environments including .NET, WebSphere, WebLogic and JBoss. In the field of interconnectivity, we offer expertise in back-office integration with ERP systems like SAP. In addition, we offer training and coaching services for the implementation and use of USoft Rules Services and USoft Web Designer.