

Web Services

Get familiar with ebXML Registry

Learn the basics about the ebXML Registry standard

Summary

This article is the first of a series of short articles that will introduce readers to the industry's various Web services standards. These articles will provide a quick introduction to a standard, its background, underlying architecture, benefits, status, and industry adoption. As some of the content might be a depiction of the author's viewpoint, readers are encouraged to refer to the links provided in [Resources](#) to gain a deeper understanding of a particular standard. This article focuses on ebXML Registry from OASIS. (*1,000 words*; **April 25, 2005**)

By Ash Parikh

Much like Electronic Data Interchange (EDI), electronic business XML (ebXML) was developed to enable business to be conducted electronically over the Internet. Custom protocols and proprietary messaging formats between trading partners created a deterrent to the widespread adoption of EDI. ebXML, on the other hand, provides an open and flexible alternative, allowing enterprises anywhere, anytime, and anywhere to engage in meaningful collaborations.

ebXML supports the following tenets of a typical B2B infrastructure:

- XML-based messaging
- Trading partner agreements
- Registry

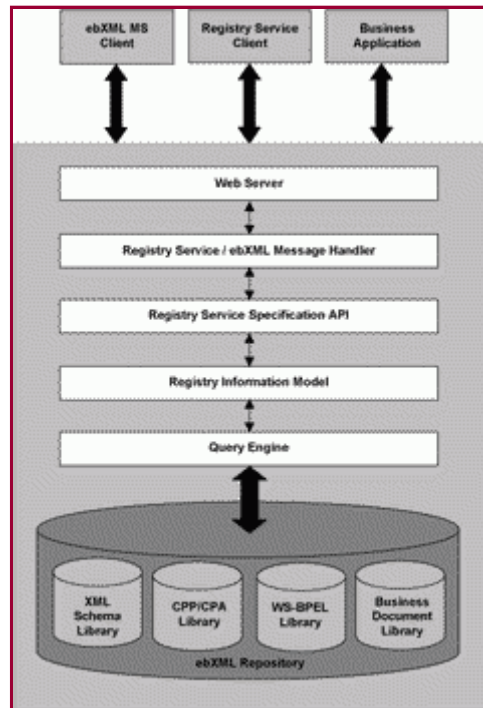
ebXML is a joint sponsorship effort by the United Nations Centre for Trade Facilitation and Electronic Business (UN/CEFACT) and the Organization for the Advancement of Structured Information Standards (OASIS), with additional support from a multitude of industry leaders.

In this column, I briefly explore the ebXML Registry standard, its architecture, maturity, industry adoption, and how an ebXML registry can use a service-oriented architecture (SOA) repository.

What is ebXML Registry?

Like its predecessor, Universal Description, Discovery, and Integration (UDDI), ebXML Registry also facilitates seamless and automatic inter-enterprise collaborations. This feature lends itself well to enabling true integration between trading partners and supports the overall goal and vision of an SOA in which applications communicate and share functionality without human interaction. Specifically, both these standards enable enterprises to discover each other, exchange XML-based messages, and engage in meaningful collaborations based on their individual trading-partner agreements and public business processes.

As depicted in the figure below, an ebXML registry can have a persistence mechanism for enterprises to share and store information as registered content. Per the ebXML Registry Services Specification, there are "a set of services that enable sharing of information between interested parties for the purpose of enabling business process integration between such parties based on the ebXML specifications."



An architectural approach for an ebXML registry. Click on thumbnail to view full-sized image.

ebXML Registry's Registry Information Model (RIM) describes entities that store information in an ebXML registry, such as the `RegistryEntry` entity, which contains an ebXML object, and the `ClassificationNode` entity, which is used to create classifications or ontologies for these objects.

As one would guess, to be of any use, this registered content needs interaction. Hence, ebXML Registry also provides two interfaces, `ObjectManager` and `ObjectQueryManager`, that enable interaction with the registry.

To get a comprehensive understanding of inter-registry cooperation, security, and content management with respect to ebXML Registry, I encourage readers to reference information available in the ebXML Registry Technical Committee's subcommittees (SC) such as:

- ebXML Registry CC (Core Components) Review SC
- ebXML Registry Cooperating Registries SC
- ebXML Registry Security Services SC
- ebXML Registry Semantic Content Management SC

What is an SOA repository?

An SOA repository is a persistence mechanism that stores information published to an SOA registry. An SOA registry is a resource that enterprises share to publish, discover, and consume Web services, thus enabling dynamic and loosely-coupled B2B collaboration. Content such as XML Schemas, Document Type Definitions (DTDs), and Web Services Description Language (WSDL) documents, can be persisted in an SOA repository, which is then used in an SOA registry to enable a subscribe/publish model for persisted services. Per the Registry Information Model specification, the RIM or SOA repository **may** be implemented within an ebXML registry in the form of a relational database schema, object database

schema, or some other physical schema. It may also be implemented as interfaces and classes within a registry implementation. I recommend the use of an XML database as the persistence mechanism since the artifacts to be stored are primarily XML-based.

What are the benefits of an ebXML registry?

An ebXML registry provides the following benefits:

- Discovery and maintenance of registered content
- Support for collaborative development, where users can create XML content and submit it to the registry for use and potential enhancement by the authorized parties
- Persistence of Web Services Business Process Execution Language (WS-BPEL), WSDL, and business documents during interactions between trading partners
- Secure version control of registered content
- Federation of cooperating registries to provide a single view of registered content by seamless querying, synchronization, and relocation of registered content
- Event notification via email or Web services

What is the status and industry adoption of ebXML Registry?

As of April 2005, the technical committee released a committee-approved specification dated March 12, 2005 for version 3.0 of ebXML Registry.

As documented in [Resources](#), ebXML Registry enjoys adoption by numerous industry leaders in varying degrees. For example, by linking to an ebXML registry, Adobe LiveCycle Form Manager enables the management and access of enterprise forms via a Web-based portal.

Using ebXML Registry in a Java world

JAXR, the Java API for XML Registries, provides a standard API for publication and discovery of Web services through underlying registries such as ebXML Registry.

JAXR performs registry operations over ebXML Registry and defines a unified information model for describing a registry's contents. JAXR allows any JAXR client to access and interoperate with any ebXML registry that is accessible via a JAXR provider. In this sense, JAXR provides a Write Once, Run Anywhere (WORA) API for registry operations, simplifying Web services development, integration, and portability.

The development lifecycle of a JAXR-based application requires the developer to do the following steps:

- Establish connection to the registry as shown in the snippet of code:

```
...
// define configuration properties
String publishURL = "https://www-
3.ibm.com/services/uddi/v2beta/protect/publishapi";
String queryURL = "http://www-
3.ibm.com/services/uddi/v2beta/inquiryapi";
Properties props = new Properties();
props.setProperty("javax.xml.Registry.lifeCycleManagerURL",
publishUrl);
props.setProperty("javax.xml.Registry.queryManagerURL", queryUrl);
props.setProperty("javax.xml.Registry.factoryClass",
"com.sun.xml.Registry.uddi.ConnectionFactoryImpl");
try {
    // Create the connection, passing it the
    // configuration properties
    ConnectionFactory factory =
```

```

        ConnectionFactory.newInstance();
        factory.setProperties(props);
        connection = factory.createConnection();
        System.out.println("Created connection to Registry");
    } catch (Exception e) {
        e.printStackTrace();
        if (connection != null) {
            try {
                connection.close();
            } catch (JAXRException je) {}
        }
    }
}
...

```


- Perform query and/or update operations
- Check output
- Close the connection

Conclusion

ebXML Registry provides a set of services that allow interested parties to share information to enable business process integration. The shared information may be persisted as objects in a repository. The submitted content may be in the form of XML Schemas and documents, process descriptions, and/or information about trading partners. Though registries may provide an index for these information artifacts, the persistence of the same could be in repositories. Such information artifacts could be used to facilitate ebXML-based B2B partnerships and transactions.

With an ebXML registry, you may register your business process and also store the same. The focus of the repository, however, is for the developer at design time to be able to download and reengineer the schema for a business process.

I encourage readers to get current information on the specification from the [ebXML Registry Technical Committee page](#).

I would like to thank Robert Smik, lead architect with the Enterprise Applications Group at Raining Data Corporation, and Vivek Kondur, senior developer with Source-N for their contributions to this article. I would also like to thank Frank Cohen, director of solutions engineering who heads the developer community XQster.com, for technically reviewing this article. 

About the author

[Ash Parikh](#) is the director of technology and development for the Enterprise Applications Group at Raining Data Corporation. He is a named expert in the field of SOA and distributed computing and has presented and authored abstracts for OASIS Symposium 2005, Delphi BPX Summit 2004, Delphi Enterprise On-Demand 2004, JavaOne 2004, JavaOne 2003, BEA e-World 2002, and JavaOne 2002. Parikh has more than 15 years of IT experience and is an active member on a number of Java Specification Requests in the Java Community Process and in OASIS technical committees. He is also the president of the Bay Area Chapter of the Worldwide Institute of Software Architects. Parikh is the collaborating author of *Oracle9iAS Building J2EE Applications* (Osborne Press, November 2002), and has also authored several technical articles in journals such as *JavaWorld*, *XML-Journal*, *Java Pro*, *Web Services Journal*, *ADTmag*, *Softwaremag.com*, and *Java Skyline*.

Resources

- ebXML Website:
<http://www.ebxml.org/>
- ebXML Registry TC page on the OASIS Website:
http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=regrep

- ebXML Registry Services Specification:
<http://www.ebxml.org/specs/ebRS.pdf>
- ebXML Registry Information Model:
<http://www.ebxml.org/specs/ebRIM.pdf>
- Java API for XML Registries JAXR:
<http://java.sun.com/xml/jaxr/>
- Adobe LiveCycle Form Manager:
<http://www.adobe.com/products/server/formmanager/main.html>
- Sun Microsystems implementation:
<http://sunflash.sun.com/articles/42/4/opt-dev/4596>
- Korea Trade Network implementation:
<http://www.gxmlhub.com/english/index.html>
- XML Global implementation:
<http://www.xmlglobal.com/>
- XML.gov registry implementation:
<http://www.xml.gov/registries.asp>
- DISA: Open Travel Alliance and Interactive Financial Exchange Forum implementation:
<http://www.disa.org/drive/>
- OASIS ebXML Registry RI:
<http://sourceforge.net/projects/ebxmlrr>
- ebXMLsoft implementation:
<http://www.ebxmlsoft.com/news/pr/press-release-26apr2002.html>
- For more articles on Web services, browse the **Java and Web services** section of *JavaWorld's* Topical Index:
http://www.javaworld.com/channel_content/jw-webserv-index.shtml



[HOME](#) | [FEATURED TUTORIALS](#) | [COLUMNS](#) | [NEWS & REVIEWS](#) | [FORUM](#) | [JW RESOURCES](#) | [ABOUT JW](#) | [FEEDBACK](#)

Copyright © 2004 JavaWorld.com, an IDG company