Web Services take your Omnis Applications to the Next Level

The Omnis Web Services component lets you integrate Web Services into your Omnis applications, or create Web Services from your own Omnis code.

The **Omnis Web Services** (OWS) component opens up the Omnis Studio IDE to interoperability with existing third-party Web Services for adding application features and functionality. Omnis Studio enables developers to extend their Omnis web applications by allowing interaction with other services and application components offered elsewhere on the Web. In addition, developers can publish Web Service methods from within their own Omnis Studio web applications to expose functionality to other applications in a Service Oriented Architecture (SOA) environment.

What are the Benefits?

By using the Omnis Web Services component, developers can enhance the functionality of their Omnis Studio applications with third-party Web Services and gain the following benefits:

Interoperability and Data Integrity

- Better integration and reuse of tried and tested code and functionality in your apps
- Access business functionality regardless of platform or implementation to link data
- Exchange data across departments or businesses in disparate locations

Productivity and Innovation

- Quickly integrate reusable application components to add functionality into your apps, opening up new and improved business solutions
- Exchange data between different applications and different platforms
- Modularize software to create new combinations for new customers and markets
- Reduce complexity and development time

New Revenues and Greater Efficiency

- Generate new revenue from enhanced or streamlined applications, together with more efficient use of existing resources
- Third-party Web Services integration can make app development quicker and easier to maintain
- Save on development time and maintenance costs

How do I get a Web Service into Omnis?

To incorporate a Web Service into your Omnis Studio app you need access to its *WSDL (Web Services Description Language)* file. This is a special file that defines precisely what the Web Service does, including its methods, as well as its input and output data types. The WSDL file can be a local file, but is more usually provided over the internet and accessed via HTTP.

Omnis Studio provides a very helpful wizard that references the WSDL file to create and compile a *Java object*. The wizard also creates an Omnis *Object class* which is derived from the Java object and encapsulates all the methods (functions) available in the remote Web Service: you can call these methods in your Omnis code to access the functions in the Web Service. There is also an option in the wizard to create a form (window) which you can use to test the Web Service, or use as the basis for your own forms in your Omnis application.

Having created the Omnis Web Services object and incorporated its methods into your library, you will need to deploy your application using the Omnis App Server, which is setup in the same way as for any Omnis web or mobile app (note the Java jar files need to be deployed with your library as well). There are tech notes on the Omnis website describing how to setup the Omnis App Server and Web Service apps.

How do I create my own Web Service?

In addition to consuming third-party Web Services in your Omnis code, you can expose your own Omnis code as a Web Service. Each operation or function within an Omnis Web Service corresponds to a method in a remote task in your Omnis library. There is a wizard to generate the WSDL file for your Omnis based Web Services. Such Web Services can be accessed by any other application, via direct or remote HTTP via the Omnis App Server and a standard Web Server.

Software Requirements and Licensing

To develop with the OWS component, you must use the Omnis Studio Professional edition, version 4.2 or higher. As of Omnis Studio 5.2, the component can be found inside the Studio ***xcomp*** folder. To use the OWS component during development, you should purchase one development license per developer, whereas deployment is free of charge.

Note: When accessing and using a particular Web Service you should refer to the relevant product or licensing documentation. To use a Web Service for testing or commercial purposes, you may need to obtain an access ID and/or license key. In some cases, Web Services may be chargeable.

Tigerlogic

www.tigerlogic.com/omnis

© TigerLogic Corporation 2012. All Rights Reserved. 12/2012