# Standards Devkit 1.0

## Copyright 2011 ExxonMobil Upstream Research Company

License notice   
Standards DevKit, version 1.0   
Copyright 2011 ExxonMobil Upstream Research Company

The following Energistics (c) products were used in the creation of this work:

* WITSML Data Schema Specifications, Version 1.4.1
* WITSML API Specifications, version 1.4.1
* WITSML Data Schema Specifications, Version 1.3.1.1
* WITSML API Specifications, version 1.3.1
* PRODML Data Schema Specifications, Version 1.2
* PRODML Web Service Specifications, Version 2.0

All rights in the WITSML™ Standard and the PRODML™ Standard, or any portion thereof, which remain in the Standards DevKitshall remain with Energistics or its suppliers and shall remain subject to the terms of the Product License Agreement available at [http://www.energistics.org/product-license-agreement](http://ctt.marketwire.com/?release=818988&id=968440&type=1&url=http://www.energistics.org/product-license-agreement).

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License.

You may obtain a copy of the License at   
[http://www.apache.org/licenses/LICENSE-2.0](http://ctt.marketwire.com/?release=818988&id=968443&type=1&url=http://www.apache.org/licenses/LICENSE-2.0)

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

# Using the Devkit

## Referencing the devkit

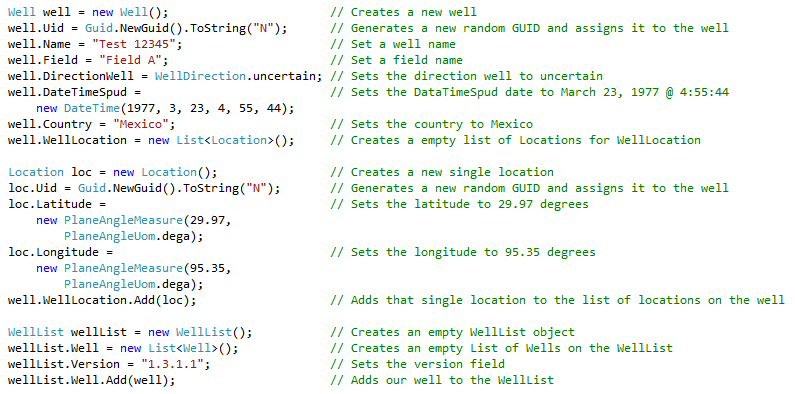
In order to use the devkit, you must first add a reference to the devkit from your Visual Studio project.

1. Right Click the project name and click “Add Reference”
2. a. If you are working in the same solution as the devkit project, click on Project and add a reference to “DataAccess”.  
   b. If you are working in a separate solution, browse to “Energistics.DataAccess.dll” and add that as a reference
3. At the top of your code, add a using statement for the namespace(s) that you will be using. (See “DataAccess Project” under “Solution Structure” below for a list of available namespaces)

## Creating and manipulating objects

Once you have referenced the devkit, you can begin to write code. The following example creates a list of wellbores and sets several values:

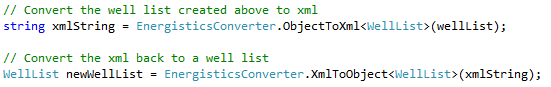
### Usage:



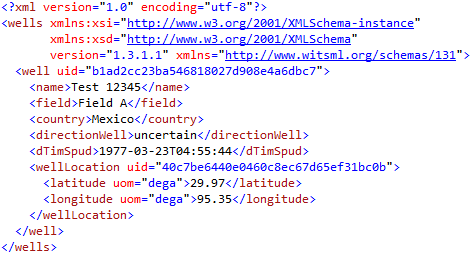
## Converting between XML and objects

The devkit stores everything it works with as objects that are easily manipulated through .NET code. However, to save or exchange this information it needs to be in XML format. The conversion between the two is easy. Here is how to convert the well we just created to an XML string.

### Usage:



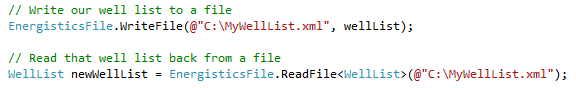
The “xmlString” value generated above would look like this:



## Working with XML files

An XML file is simply a XML string written to a file. You could handle the file operation yourself, but the devkit provides convenience method to help you do that.

### Usage:



## Calling Web Services Directly

The devkit supports calls to both WITSML and PRODML web services. These calls use the same parameters and syntax that the official WITSML / PRODML web services use. Please refer to the appropriate documentation on using these web services.

### Example usage:



## Using the WITSML Web Service convenience class

For added convenience, a WITSMLWebServiceConnection class has been created to simplify calls to WMLS. You can always call the web service directly as shown above, but this class is intended to make basic calls easier.

### Usage:

### 

# Solution Structure

## SampleBrowserApp Project

This project contains a sample browser application written using the devkit. It utilized many of the devkit features and demonstrates some of the more difficult concepts such as using reflection when you don’t know at compile time which datatypes that you will be using.

## UnitTests Project

This project contains Unit Tests which serve as sample code as well as providing regression testing functionality to the devkit itself.

# EnumValues

## Using EnumValues

WITSML / PRODML support the use of an EnumValues.xml file that can extend the allowed additional values for certain data types. The devkit supports this by compiling the provided enumValues.xml file into the devkit so the values can be used like any other enumeration.

### Usage:

Message m1 = new Message();  
m1.ActivityCode = ActivityCode.Abandonment;

## Using EnumValues Extensions

In some cases, you may have your own EnumValues.xml file that contains additional allowed values that are not included in the EnumValues.xml file compiled with the devkit. In that case, there is an included executable that can create a separate dll that contains your extensions.

### Sample enumValues.xml file:

## 

### Generating Extension Class:

EnumValuesExtensionGenerator.exe enumValuesExtensions.xml Company.Namespace WITSML131

### Extension Class Usage:

Message m1 = new Message();  
m1.ActivityCode = Company.Namespace.ActivityCodeExtension.MyCustomActivityCode;

## Using custom runtime EnumValues

Sometimes EnumValues cannot be compiled ahead of time and must be created at runtime.

### Usage:

Message m1 = new Message();  
m1.ActivityCode = new ActivityCode("My Activity Code");