

WITSML Product Certification Form

Please complete this form and return to Jana Schey (Jana.Schey@Energistics.org) in advance of or along with the signed license agreement. Examples of processed forms as well as the downloadable license agreement are available on the Energistics website via Standards Resource Centre->Product Certification Program.

Q1	Product Name, Version, Description
A1	<p>Polaris Guidance RDS (Remote Drilling System) – Version 2.5.</p> <p>Polaris RDS brings directional drilling and MWD data logging services to the Internet enabling remote users to have access to rig site information in real-time at the office or on the go. It is a rugged embedded data logging system specifically designed for Directional and MWD data collection and reporting.</p>
Q2	Vendor Name, Logo, Description
A2	Polaris Guidance LLC – Polaris is an innovative developer and manufacturer of wired and wireless remote drilling systems for the oilfield drilling sector. (http://www.polarisguidance.com/)
Q3	Product Availability (from, to)
A3	August 1 st , 2012 onwards
Q4	Form Submitter Information
A4	Richard Gonsuron, Director, Engineering
Q5	WITSML Functional Coverage. This is described in light of definitions provided directly on the form of functions of Client products as producers and/or consumers, of Server products as Receiving, Delivering, and/or Managing Servers. Functionality is summarized by checking which of sixteen overall functions are exhibited by the product.
A5	<p>Check all that apply. Explain limitations and/or special circumstances in the Comments area.</p> <p>Notes:</p> <ul style="list-style-type: none"> • Functional coverage is organized according to five kinds of product functionality with respect to the WITSML Standards: two kinds of client functions and three kinds of server functions. • A product may exhibit multiple kinds of functionality. • The terms client and server are used here exclusively with respect to the WITSML Server API interfaces. Clients issue requests to servers. Servers receive and respond to requests from clients. • Behaviors for products that do not use the WITSML Server API are classified in an analogous manner.

The five product classifications of WITSML functional coverage are:

Client Products

1. WITSML Producer Client -- a product that generates or otherwise obtains data that is formulated as WITSML object instances and sent to a WITSML Server to be incorporated in that server's data population. Examples of such products include products that pick up real-time data from sensor devices, format it, and send it to a server; and products that extract data from data stores, format it, and send it to a server.
2. WITSML Consumer Client -- a product that issues requests for data as queries of subscriptions to a WITSML Server and then receives data as query responses or subscription publications. Examples of such products include products that acquire data from a server, possibly reformat it, and delivery it to an application program or viewer utility.
- 1&2. Products the exhibit combined Producer and Consume Client functions may be application programs that operate directly on a WITSML Server, such as a mudlogging application or a pore pressure analysis application.

Server Products

3. WITSML Receiving Server -- a product that performs WITSML Server functions in general and, in particular, acquires data from external sources. Data acquisition may be through WITSML API interfaces or other mechanisms.
4. WITSML Delivering Server -- a product that performs WITSML Server functions in general and, in particular, delivers data to external destinations. Data delivery may be through WITSML API interfaces or other mechanisms.
5. WITSML Managing Server -- a product that performs WITSML Server functions in general and, in particular, supports requests from authorized client applications to augment (extend), modify, or delete (part or all) WITSML object instances.
- 3&4. The general understanding and expectation is that a product characterized as a WITSML Server supports both Receiving and Delivering Server functionality.
- 3&4&5. The addition of Managing Server functionality allows a WITSML Server product to do more than store and forward data, such as supporting data quality management client applications that help ensure the integrity and quality of data content in a Server data population.

General Functions

6. Virtually all products associated with the WITSML Standards will issue and/or process WITSML Server General Functions to determine the capabilities and version of a server product.
1. WITSML Producer Client A product that (generates and) sends WITSML object instances to a destination process:
 - 1a Sends to a WITSML Server using AddToStore interface
 - 1b Otherwise

	<p>2. WITSML Consumer Client A product that requests and receives WITSML data from a source process:</p> <p>2a <input checked="" type="checkbox"/> Queries a WITSML Server using GetFromStore interface 2b <input type="checkbox"/> Subscribes to a WITSML Server using Publish interface 2c <input type="checkbox"/> Otherwise</p> <p>3. WITSML Receiving Server A product that performs the WITSML Server interfaces and receives data from source processes:</p> <p>3a <input type="checkbox"/> Receives WITSML object instances via AddToStore interface 3b <input type="checkbox"/> Otherwise receives WITSML object instances 3c <input type="checkbox"/> Receives non-WITSML form data treated as if it were WITSML object instances or a virtual equivalent</p> <p>4. WITSML Delivering Server A product that performs the WITSML Server interfaces and delivers data to destination processes:</p> <p>4a <input type="checkbox"/> Delivers WITSML data in response to queries via GetFromStore interface 4b <input type="checkbox"/> Publishes WITSML data in response to subscriptions via the Publish interface 4c <input type="checkbox"/> Otherwise delivers WITSML data 4d <input type="checkbox"/> Delivers non-WITSML form data derived from WITSML object instances or a virtual equivalent</p> <p>5. WITSML Managing Server A product that performs the WITSML Server interfaces and manages (augments, changes, deletes portions, or deletes entirely) WITSML object instances or a virtual equivalent:</p> <p>5a <input type="checkbox"/> Processes modification requests via AddToStore, UpdateInStore, DeleteFromStore interfaces 5b <input type="checkbox"/> Otherwise processes modification requests</p> <p>6. WITSML General Functions A product that issues general WITSML Server interface requests to a WITSML Server:</p> <p>6a <input type="checkbox"/> Issues GetVersion and/or GetCapabilities A product that performs the general WITSML Server interfaces: 6b <input type="checkbox"/> Processes GetVersion and/or GetCapabilities</p>
Q6	<p>WITSML Data Coverage. This is described by indicating for each of twenty types of data whether the data is Delivered and/or Received by the product. Provision for additional data type support is given. Special limitations between Functions and Data must be explained.</p>
A6	<p>Mark D for Deliver and R for Receive, as applicable. If all functions do not apply, note either functions supported or functions not-supported, e.g. supported by 1a. Explain other limitations or special cases in the Comments area.</p> <p><input type="checkbox"/> Realtime <input checked="" type="checkbox"/> Well</p>

	<input type="checkbox"/> R Wellbore <input type="checkbox"/> D Log & WellLog <input type="checkbox"/> D Trajectory & Traj. Stn. <input type="checkbox"/> Message <input type="checkbox"/> Mud Log <input type="checkbox"/> Rig <input type="checkbox"/> Survey Program <input type="checkbox"/> Target <input type="checkbox"/> Fluids Report <input type="checkbox"/> Operations Report <input type="checkbox"/> Risk <input type="checkbox"/> Formation Marker <input type="checkbox"/> Conventional Core <input type="checkbox"/> Sidewall Core <input type="checkbox"/> Cement Job <input type="checkbox"/> Tubular <input type="checkbox"/> BHA Run <input type="checkbox"/> WB Geometry (Wellbore Geometry) <input type="checkbox"/> Other, specify: _____
Q7	Additional Information (optional)
A7	
Q8	Form Submission Date.
A8	August 1st, 2012