Petroware

JWitsml

High performance programming library for WITSML

APPLICATIONS

- Full-scale real-time E&P applications
- Real-time geo-steering applications
- WITSML adapters for cloud services
- Advanced WITSML viewers for browsing remote WITSML servers
- WITSML real-time apps for mobile units
- Real-time dashboards with alarms/ triggers
- Accessing real-time log data for visualization and derivation of Quick Look properties
- Moving data between WITSML databases
- Moving data between WITSML and corporate databases
- Populating WITSML servers with log data from LIS, DLIS, LAS, CSV, Excel etc.
- Extracting real-time data and export to LIS, DLIS, LAS, CSV, Excel etc.
- Report drilling information to the authorities
- Real-time drilling or logging simulators
- WITSML server validation, benchmarking, and performance test applications

FEATURES

- CRUD support of all WITSML types
- Supports WITSML 1.3, 1.4 and 2.0
- Access through HTTP/SOAP
- Full ETP/WebSocket implementation
- Includes the comprehensive Energistics unit of measure database
- Supports Java 8+

BENEFITS

- High performance, small footprint
- Simple and well-documented API
- Thread-safe
- Online tutorials
- For MS/Windows, Mac OS/X, Linux, and Android

WITSML

<WITSML/>™

Wellsite information transfer standard markup language (WITSML) is an open standard for defining and transmitting technical drilling data between disparate systems and companies in the E&P petroleum industry.

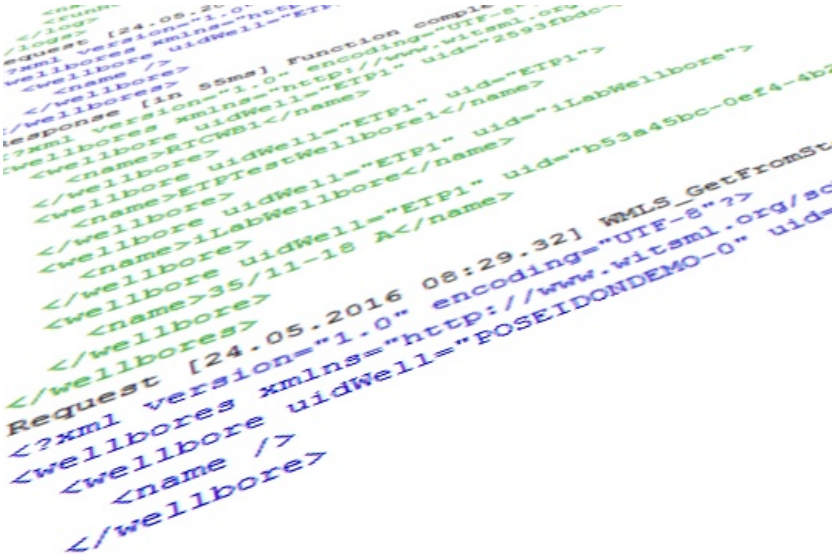
WITSML is a technology which allows previously incompatible or proprietary systems to interoperate and share data.

The WITSML standard is being developed and maintained by Energistics (an affiliate of the Open Group), a nonprofit consortium designed to develop and deploy open data exchange standards in the oil and gas industry.

What is JWitsml?

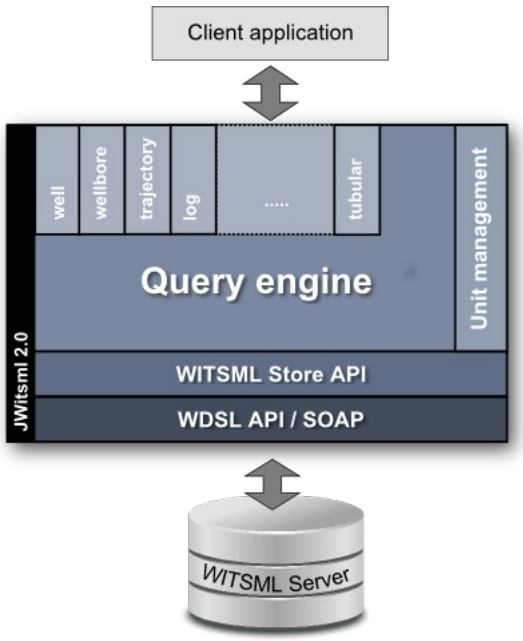
JWitsml is a Java library for accessing data on WITSML servers. The goal of the JWitsml project is to support WITSML client software with WITSML data through a clean, complete, well documented, efficient, and simple to use Java programming interface.

Many tools exist for browsing WITSML data. To access the data for performing advanced computations or create custom services and applications, JWitsml is the solution.

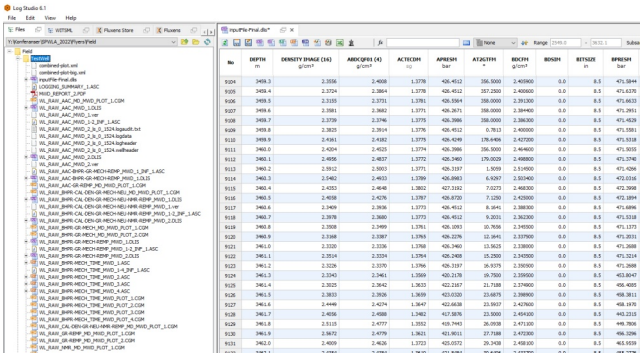


JWitsml contains comprehensive features for connection logging so that all client requests and server responses can be monitored and analysed.

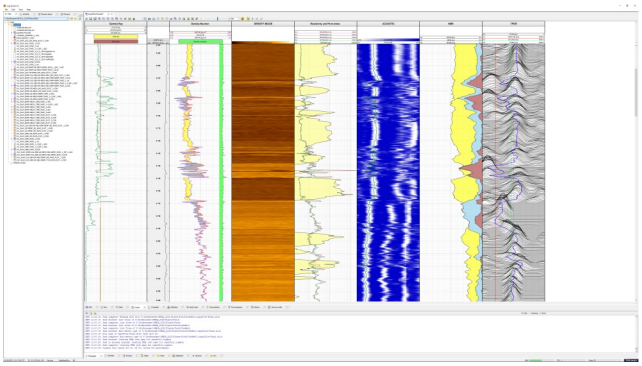
JWitsml



The architecture of JWitsml



Log Studio showing WITSML tabular data in a live stream.



Log Studio showing WITSML live log plots.

Architecture

JWitsml has a layered architecture as shown above. The bottom layer contains the implementation of the WITSML WSDL API specification or (optionally) the HTTP/WebSocket specification. It contains a custom implementation of the SOAP protocol that makes it possible to use JWitsml on the Android mobile platform.

The next level is the WITSML Store or the ETP (Energistics Transfer Protocol) accessor which provides XML data type queries according to the WITSML definitions.

The query engine is responsible for converting between the Java API calls of the client and the XML specification syntax of the server.

Queries (create/read/update/delete) specified by the client are converted to the equivalent (and minimal) XML syntax and sent to the server through the lower layers. The server response is disassembled and converted back to the Java model.

The top layer defines all WITSML types as Java objects with getters and setters for all defined properties. This is the clients convenient view of the WITSML data model.

The Unit Manager is an optional service provided for the client to do unit conversions on numeric data.

Log Studio

Log Studio is the Petroware reference implementation for WITSML and related real-time technologies. The application contains a rich set of functionalities for working with digital well-logs and related E&P data.

Note that the pure WITSML data access is only a small part of building a full-scale real-time E&P application. A client application must be able to do concurrent access across multiple object sources and propagate the data asynchronously into a thread-safe data model. Requested log curves may reside in different log sets, each measured against separate Z references. Log data may be organized in runs that must be spliced live. The amount of data may be vast, and the client must be prepared to do multiple queries to complete each server request. The client program must gracefully handle unstable network connections and should be able to operate 24/7 without interruption.

Log Studio addresses all these issues.

Petroware AS

Sjøhagen 3
N-4016 Stavanger
Norway

E-mail: info@petroware.no

<JWitsml/>

Petroware