

JWitsml ETP

High performance programming library for WITSML

APPLICATIONS

- Full-scale real-time E&P applications
- Real-time geosteering applications
- Advanced WITSML viewers for browsing remote WITSML servers
- WITSML real-time apps for mobile units
- Real-time dashboard with alarms
- Accessing real-time log data for visualization and QL-computations
- Moving data between WITSML databases
- Moving data between WITSML and corporate ER databases
- Populating WITSML servers with log data from LIS, DLIS, LAS, CSV, MS/Excel etc.
- Extracting real-time data and export to LIS, DLIS, LAS, CSV, MS/Excel etc.
- Report drilling information to the authorities
- Real-time drilling or logging simulators
- WITSML server validation, benchmarking and performance test applications

FEATURES

- Full CRUD coverage 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
- Based on the latest Java 8 technology from Oracle

BENEFITS

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



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, a non-profit 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 extremely simple to use Java programming interface.

There exists many tools available for browsing WITSML data, but in order to get access to the actual data for performing advanced computations or create custom applications, a library like JWitsml is required.

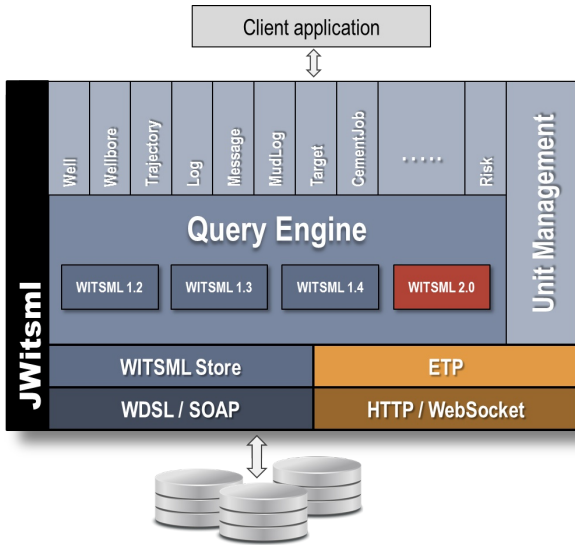
```

<?xml version="1.0" encoding="UTF-8" ?>
<wellbores xmlns="http://www.witsml.org/witsml-1.3" uidWell="ETP1" uid="2593f8dc-..."/>
<wellbore />
</wellbores>
Response [in Ssmal Function comple
?xml version="1.0" encoding="UTF-8"
wellbores xmlns="http://www.witsml.org/witsml-1.3" uidWell="ETP1" uid="2593f8dc-..."/>
<wellbore uidWell="ETP1" uid="iLabWellbore">
  <name>ETPTestWellbore</name>
</wellbore uidWell="ETP1" uid="iLabWellbore">
<wellbore uidWell="ETP1" uid="b593a45bc-0ef4-4b7-...>
  <name>35/11-18 A</name>
</wellbore uidWell="ETP1" uid="b593a45bc-0ef4-4b7-...>
</wellbores>
Request [24.05.2016 08:29.32] WMLS_GetFromSt
<?xml version="1.0" encoding="UTF-8" ?>
<wellbores xmlns="http://www.witsml.org/witsml-1.3" uidWell="POSEIDONDEMO-0" uid="..."/>
<wellbore />
</wellbores>

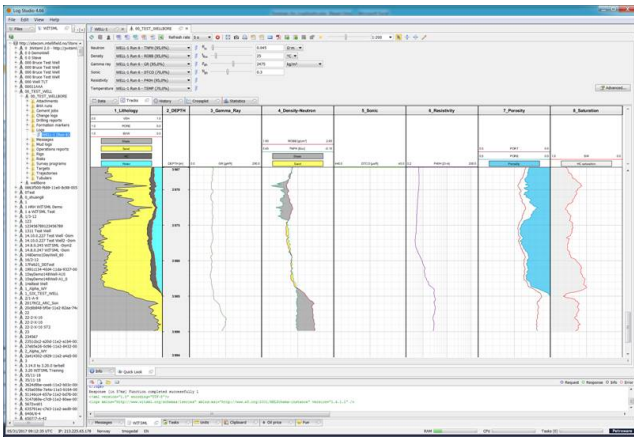
```

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

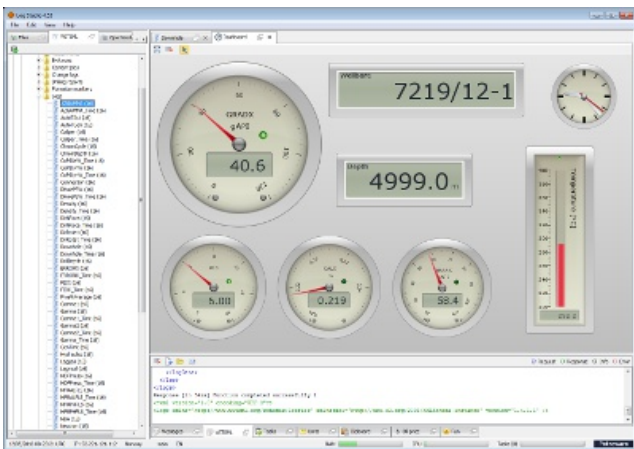




Basic architecture of JWitsml.



Log Studio with live Quick Look interpretation on a real-time WITSML stream.



Log Studio showing a custom dashboard with selected properties of a drilling well from a WITSML server.

Architecture

JWitsml has a layered architecture as shown. 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 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 in order to do unit conversions on numeric data.

Log Studio™

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

Note that the pure WITSML data access is only a tiny 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 possibly 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 of these issues.

Petroware AS • Logtek AS
 Ipark
 Professor Olav Hanssens vei 7
 4021 Stavanger
 Norway
 Phone: (+47) 51 87 40 80
 e-mail: info@petroware.no



<JWitsml/>
<http://petroware.no/jwitsml>

*Mark of Petroware AS
 Other company, product, and service names
 are the properties of their respective owners.
 Copyright © 2017 Petroware AS. All rights reserved.