PRODML Technical Usage Guide

For PRODML v2.0

PRODML Overview	The PRODML standard facilitates data exchange among the many software applications used in production operations, which helps promote interoperability and data integrity among these applications and improve workflow efficiency.
Version of Standard	2.0
Abstract	This guide describes PRODML, its capabilities, key concepts, and related resources.
Prepared by	Energistics and the PRODML SIG and its various working groups
Date published	9 December 2016
Document type	usage guide
Keywords:	standards, energy, data, information, production operations, production reporting

Document Information	
DOCUMENT VERSION	1.0
Date	9 December 2016
Language	U.S. English

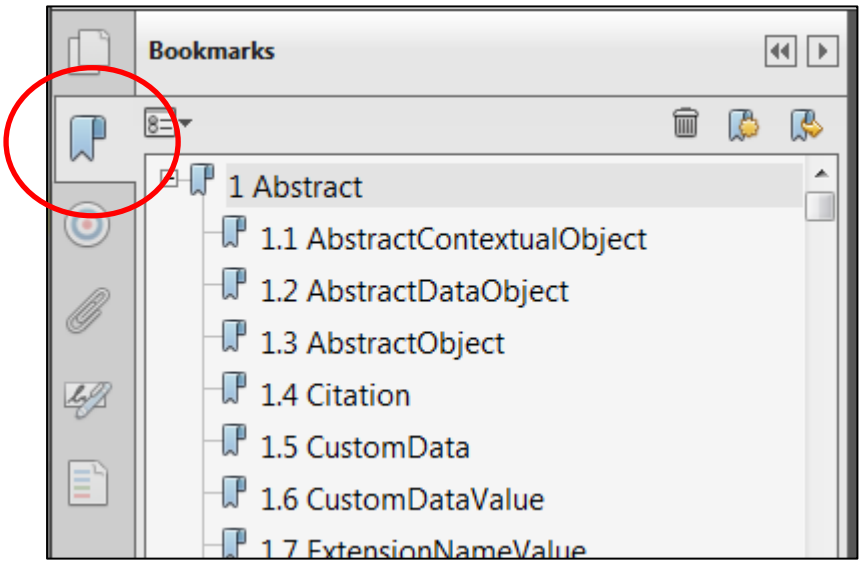
Acknowledgement

Energistics would like to acknowledge and thank all those who contributed to the development of the PRODML standard. Where appropriate, individual chapters acknowledge specific contributors. In particular, we would like to thank Laurence Ormerod for his leadership and commitment to Energistics, our members, and the industry.

Amendment History			
Std. Version/ Doc Version	**Date**	**Comment**	**By**
2.0 / 1.0	9 Dec 2016	First release of PRODML on the Energistics Common Technical Architecture.	PRODML SIG, various work groups, and Energistics

Table of Contents

Because this document is so long, no table of contents was generated inside the document. Use the table of contents pane in Acrobat by clicking on the icon shown below, which is located at the far left in an open PDF document.



1 Introduction to PRODML

PRODML is an upstream oil and gas industry standard for the exchange of data related to production management. This domain covers the flow of fluids from the point where they enter the wellbore to the point of custody transfer, together with the results of field services and engineering analyses required in production operation workflows.

The production phase of an oil or gas field covers a wide range of data transfer requirements. PRODML covers some of these requirements but does not yet present a coherent view of the whole domain and PRODML's role within it. A "top-down" full development of such a large scope of data transfers is too large a task for an essentially voluntary effort. Hence, PRODML development has been a "bottom-up" development approach, driven by member company project requirements.

PRODML has been developed by Energistics members—which includes oil companies, oilfield service companies, software and technology companies, regulatory agencies and academic institutions—through the PRODML special interest group (SIG), and its various work groups and project teams.

1.1 PRODML Overview

PRODML is an XML-based data-exchange standard that facilitates reliable, automated exchange of data among software packages used in production management workflows. It defines a set of data objects that can be implemented into software so it can read and write to the PRODML standard format, which allows PRODML-enabled software to more easily share and use this data.

Underpinning PRODML's domain-specific data models is the Energistics Common Technical Architecture, which provides various technical standards and methods upon which PRODML builds.

The table below lists the main segments of data transfer within production; PRODML standards are in daily commercial use in all of these segments.

Segment	Scope
Production volumes	Internal use, to regulators, to partners.
Completion and well services	Hardware and operations over producing life.
Surveillance	Monitoring production in the wellbore, the reservoir (near wellbore); and on the surface.
Optimization	Decision support both for design and operational optimization.

NOTE: As described in Section 1.1.2, completion and well services is now incorporated into WITSML. But, per the integration of the Energistics domain standards (see Section 2.4), WITSML data objects are easily available to user of PRODML.

These segments are defined in more detail in Error! Reference source not found., which shows some of the main data scope requirements within each segment. This figure is intended to be illustrative rather than exhaustive. For each item of data scope, the v2.0 coverage of PRODML is shown, together with a high-level estimate of current usage—the two together giving some idea of maturity. The bands of high-medium-low scores have the following meanings:

1. PRODML Coverage
    a. High: strong coverage where a multi-company group of domain experts has specified the requirement and validated the model.
    b. Medium: reasonable coverage where significant work has been done, but where the requirement was not the primary aim of the effort, or where the effort was focused on one aspect of the requirements, not general coverage.

c. Low: a low level of coverage, either where the scope required is only partially covered or not at all, of where work has been donated and incorporated but based on one company's views with no peer review.

2. PRODML Usage

a. High: widespread uptake over a material number of companies/situations. The standard is therefore well-tested and known to be complete.

b. Medium: material uptake but in a limited number of cases, e.g. for one purpose, or by a small number of companies. The standard is therefore workable but may need adapting for high usage.

c. Low: usage limited to one or two companies, or not used at all. The standard therefore may need work and adaption before it can be used more widely.

Segment	Scope Required	PRODML Coverage within scope	PRODML Usage
Production volumes	Volume reports daily	High	Medium
	Volume reports monthly	High	Medium
	Operational Reports	Medium	Low
	Networks	Medium	Low
Completion and well services	Initial completion	High	Medium
	Workover and well service change history	High	Medium
	Completion correct details at any instant	High	Medium
	Well services - details of each job	Low	Low
	Artificial Lift equipment	Low	Low
	Workover - details of each job	Medium (WITSML)	Low
Surveillance	Well tests (steady state production tests)	High	Medium
	Producing well performance	Medium	Low
	Downhole point measurements - time series	Low	Low
	Gathering system point measurements - time series	Low	Low
	Surface equipment surveillance	Low	Low
	Downhole distributed measurements	High	Medium
	Transient well tests/formation tests	Low	Low
	Gradient surveys/other artificial lift surveillance	Low	Low
	Production logging	Low	Low
	Fluid samples and analysis	High	Low
Optimization	Well simulation (inc operational and design usage)	Low	Low
	Network simulation (inc operational and design usage)	Low	Low
	Reservoir to production simulation and optimization	Low (RESQML)	Low

Figure 1–1. High-level scope of sub-domains within production segments, showing PRODML maturity.

The following sections describe in more detail the capability and usage within the segments of production data outlined above.

1.1.1 Production Volumes

The origins of PRODML were focused on the use of the product volume data object. This has been adopted on a large scale on the Norwegian Continental Shelf (NCS) where all producing fields report to partners and the regulator using this standard. For more information, see Chapter 24.

The product volume data object has been used by other member companies for volume reporting but adoption has not been large, and feedback has been that this is owing to the complexity of this model.

The simple product volume reporting capability (SPVR) has been developed and added to version 2.0 and the aim throughout this development has been to keep the model as simple as possible, while still covering most, if not all, requirements.

The coverage of requirements in this area is therefore believed to be high using SPVR or product volume if the most flexible approach is needed. Maturity is medium, since uptake outside the NCS is low.

1.1.2 Completion and Well Services

A joint PRODML-WITSML project during 2012-2013 resulted in a comprehensive completion domain capability. The model comprises data objects for:

- The well completion equipment itself (assembled into multiple strings, e.g. tubing, casing, etc.); couplings between elements of the completion such as plugs and packers, cement.
- The connections to the reservoir, by position, by type (perforations, gravel pack, etc.).
- The time span of each item of equipment and each connection, so that the configuration at any time can be reported.
- A “ledger” of jobs performed to change the completion equipment, which ties in with the time spans of equipment per the previous list item.

The set of objects does not address in any detail the different kinds of well service or workover “jobs”.

Artificial lift is only dealt with at a skeletal level.

WITSML itself deals with some of the drilling rig type operations.

The capability described has been used in a large commercial setting and by the companies involved, to various degrees.

The scores above reflect this experience and capability.

NOTE: This whole segment, although developed largely under the auspices of PRODML, is now packaged as part of WITSML. For more details, see the *WITSML Technical Usage Guide* (Chapter 8). Although this document does not refer further to the completion segment of production management, it is vital to know that the level of coverage and maturity described is available in the Energistics portfolio.

1.1.3 Surveillance

Production surveillance activities consist of a wide variety of measurement and analysis methods. PRODML coverage of this whole universe of data types is limited. The areas highlighted above as being covered to a high degree do have high-detail models and these are all described in this document.

Usage of some of these (well performance, distributed fiber optics, fluid sampling for example) is shown as low because these are recent developments, either in version 2.0 or in version 1.3, which preceded it. There does appear to be significant interest in uptake of these sub-domains.

1.1.4 Optimization

Within the production management domain, there is somewhat of a hierarchy of requirements. For example: it is hard to report surveillance without a description of what is being measured, and it is hard to optimize something without both a description of what is being optimized, and the measurements which are input to the optimization.

The optimization segment has low scores for its sub-domains simply because the foundational data models are not yet sufficiently developed to support major optimization workflows. For example: well simulation (commonly known as nodal analysis) needs a model covering the following elements:

- The physical hardware in the well—available through the Completion segment since 2014.
- The paths by which fluid flows through the hardware—no model developed.
- The properties of the connection to the reservoir—available through the Completion segment.

- The flow properties of the near-wellbore reservoir and of the connection to the reservoir—no model developed for well simulation, though gridded properties available in RESQML.
- The fluid properties flowing in the system—available from v2.0 in 2016.
- The control of flow—chokes, artificial lift, etc.—no model developed.
- The reporting of well simulation results (flowrates, sensitivities, flowing gradients, etc.)—no model developed.

Much of the input data can be modelled using other segments of PRODML and other Energistics domain standards, but the specific data needed to run a simulation using this base physical data has not yet been modelled.

Therefore, in general, the coverage of optimization is low, being limited to partial solutions. The segment is a good opportunity for future development.

1.2 Use Cases

Energistics aims to focus its development on use cases defined by its SIGs and its members. Use cases for specific domains are included in the parts/chapters for each specific domain.

1.3 What's new in PRODML v2.0?

1.3.1 Energistics Common Technical Architecture (CTA)

One of the goals of digital oilfield technology is to break down domain silos in E&P workflows. In support of this goal, Energistics is working to better harmonize its domain-based standards, including PRODML, RESQML (earth modeling), and WITSML (wells, drilling, well-related operations).

Energistics is also moving towards sharing resources and establishing processes across its standards where this approach makes sense, for example, with coordinate reference systems, units of measure, and file packaging conventions. These shared resources are referred to as the Energistics Common Technical Architecture (CTA). A key factor driving the update of PRODML is the move to the CTA.

For more information on the CTA, see Section 2.1, Section 2.3, and the *CTA Overview Guide*, which is included in the PRODML download.

1.3.2 PRODML Domain Capabilities

PRODML v2.0 also features some brand new domain capabilities and sets of data objects, which include:

- **Simple product volume reporting (SPVR)**. Intended initially for reporting for partners is North American joint operations, the SPR provide a simplified, reliable way to provide production volumes to non-operating partners. For more information, see Part II (which begins with Chapter 3).
- **Fluid and PVT analysis (PVT).** A set of PRODML data object that can be used to consistently capture and communicate fluid and pressure-volume-temperature (PVT) analysis data covering sample acquisition, laboratory analysis, fluid system characterization, and property generation for upstream technical workflows. For more information, see Part III, which begins with Chapter 8.
- **Distributed acoustic sensing (DAS)**. DAS is a fiber optic technology used for oil and gas surveillance applications, which include wellbore, pipeline, and surface facility monitoring. The DAS data objects were developed to provide an industry-defined, vendor-neutral format for exchanging the large volumes of data associated with DAS. It builds on recent work to define data-exchange standards for data associated with distributed temperature sensing (DTS), which is also part PRODML. For more information, see Part V, which begins with Chapter 17.

1.4 Audience, Purpose and Scope

This guide is intended for both business/domain people who use data and applications, and for information technology (IT) professionals—software engineers, data managers and developers—who want to implement PRODML into software.

This guide describes how the objects have been designed and are intended to be used. Use of XML means the standards can be implemented on many platforms and in many programming languages, so no specifics on implementation are given.

Chapter 2 gives a technical overview of PRODML, and subsequent chapters go through each sub-domain.

1.4.1 Audience Assumptions

This guide assumes that the reader has a good general understanding of programming and XML, and a basic understanding of the exploration and production (E&P) domains and related workflows.

1.5 Resource Set: What Do You Get When You Download PRODML?

PRODML is a set of XML schemas (XSD files) and other technologies freely available to download and use from the Energistics website. When you download PRODML, you get a zip file structured as shown in Figure 1–2, which contains the resources listed in the tables below in these 2 main groups:

- **PRODML-specific** (Section 1.5.1). The schemas and documentation specific to PRODML. Note in Figure 1–2 (left) (red square):
  - ProdmlAllObjects.xsd is schema that includes all other schemas in shown.
  - The ProdmlCommon schema includes objects shared by other PRODML data objects.
- **Energistics Common Technical Architecture** (Section 1.5.2) (in the *common->v2.1* folder) Components of the Energistics Common Technical Architecture (CTA), which is a set of specifications, schemas, and technologies shared by all Energistics domain standards.

To download the latest version of this standard, visit: http://www.energistics.org/production/prodml-standards/current-standards

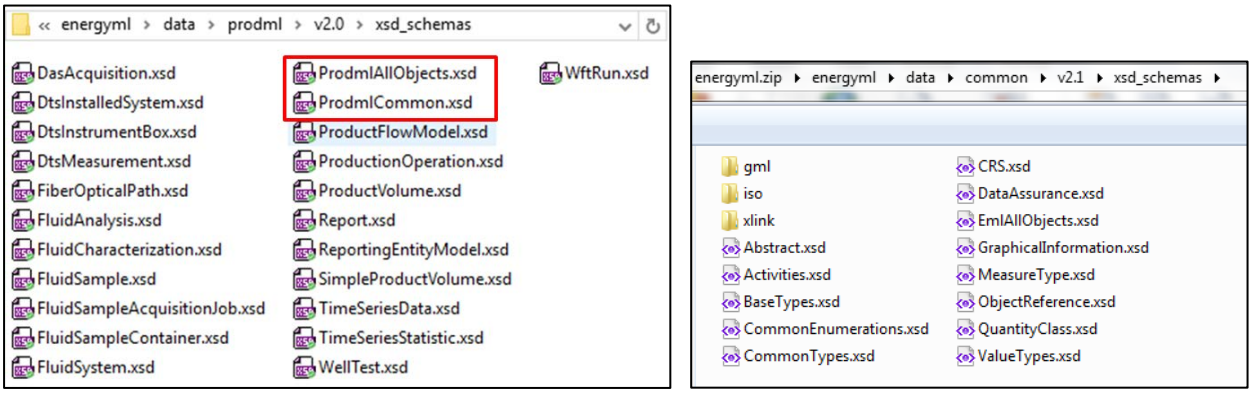


Figure 1–2. You download the PRODML standard as zip file from the Energistics website. It contains the resources described in the two tables below. The figure shows (left) the PRODML schemas (xsd files) and (right) the Energistics common schemas.

1.5.1 PRODML-Specific Resources

The PRODML download will install the folder structure described in Section 2.4. Within the specific prodml\v2.0 folder will be found the resources listed in the table below.

	Document/Resource	Description
	Prodml→v2.0 (folder)	
1.	xsd_schemas (folder)	Schemas for all of the data objects in PRODML v2.0. This folder contains all top-level objects outlined in Section 2.2 and map onto the sub-domains of PRODML as shown in Figure 2-3.

	Document/Resource	Description
2.	xml_example (folder)	One example data object for each schema. These examples are not intended to represent "engineering data" examples of use cases. They are merely valid examples of the xml complying with each schema.
3.	ancillary (folder)	Contains supporting material, which includes a mapping of v1.3 data objects to v2.0 data objects.
	doc (folder) contains the following documents:	
4.	*PRODML Technical Usage Guide* (This document.)	Provides an overview of PRODML and details about the sub-domains and supporting data objects. If just getting started with PRODML, begin with this document.
5.	PRODML UML Data Model (XMI file)	The entire UML data model that developers and architects can explore for better understanding of data objects, definitions, organization, and relationships. Energistics saves the UML model as an XMI file, a format that can be imported by any UML data modeling tool.
6.	*PRODML Technical Reference Guide*	Generated from the UML model, lists and defines all elements in the data model (for easy reference).
7.	*PRODML Product Volume, Network Model & Time Series Usage Guide*	Documentation for previously published versions (v1.x) of these PRODML data objects: • Product Volume • Product Flow Model (Network) • Time Series Some minor updates have been made to this document for v2.0, and some of the material that is most relevant to v2.x usage of these data objects is incorporated into the *PRODML Technical Usage Guide.*
8.	PowerPoint presentations	Presentations for the following PRODML sub-domains: • Simple Product Volume Reporting (SPVR) • DAS • DTS • PVT • Wireline Formation Tester (WFT)
9.	**doc→example (folder)**	This folder contains sub-folders for each of the following sub-domains. Each folder contains data files for extended worked examples, which are also explained in detail in the corresponding chapters of this *PRODML Technical Usage Guide.* • DAS • DTS • PVT • SPVR
10.	prodml_schema_overview.html	Launches an html, hyperlinked navigation through the: • Schemas (HTML versions of the schemas are in the

	Document/Resource	Description
		doc→schema folder) • Example XML files (links to the examples in the prodml→v2.0→xml_example folder)

1.5.2 Energistics CTA Resources

These resources are included in a standards download, unless otherwise specified in the table, for use with version 2 and higher of all Energistics domain standards. Not all standards use all of these technologies. The technologies used within PRODML are listed in Section 2.3.

	Resource/Document	Description
1.	Energistics common XSD files	Schemas (XSD files) for Energistics CTA data objects, which are contained in the folder named *common* and are further described in this document. NOTE: The *common* folder is Included as part of the package when you download any of the Energistics domain standards.
2.	UML Data Model (XMI file)	The UML data model that developers and architects can explore for better understanding of data objects, definitions, organization, and relationships, in context. Used to generate the XSD files and technical reference documents. NOTE: Objects in the CTA are in the folder named *common* and included in the UML for each Energistics domain standard. Energistics saves the UML model as an XMI file, a format that can be imported by any UML data modeling tool.
3.	*Energistics Common Technical Architecture Overview Guide* (For CTA, start here.)	Provides an overview of the components that comprise the Energistics CTA.
4.	*Energistics common Technical Reference Guide*	Lists and defines packages, data objects, elements, and relationships for the objects in the CTA *common* folder. Produced from the common UML package from which Energistics *common* XSDs are produced.
5.	*Energistics Packaging Conventions (EPC) Specification*	Specifies the Energistics Packaging Conventions (EPC), which is the set of practices to store multiple files as a single entity for data transfer; this single entity is referred to as an “EPC file” (or sometimes, an “Energistics package”). EPC is an implementation of the Open Packaging Conventions (OPC), a container-file technology standard.
6.	*Energy Industry Profile of ISO 19115-1 (EIP)*	An open, non-proprietary exchange standard for metadata used to document information resources, and in particular resources referenced to a geographic location, e.g., geospatial datasets and web services, physical resources with associated location, or mapping, interpretation, and modeling datasets. The EIP is an ISO Conformance Level 1 profile of the widely adopted international standards ISO 19115-1:2014 which provides XML implementation guidance with reference to ISO Technical Specification 19115-3:2016.
7.	*Energistics Identifier Spec*	Describes the syntax and semantics of data object identifiers used in Energistics data exchange standards, which include UUIDs and URIs, and object reference.
8.	*Energistics Unit of Measure Standard* (Must be downloaded separately:	A dictionary, grammar specification, and related documentation, which provide a consistent way to define, exchange, and

	Resource/Document	Description
	http://www.energistics.org/asset-data-management/unit-of-measure-standard)	convert between different units of measure. All Energistics standards (PRODML, WITSML, PRODML, etc.) must use this dictionary; other industry groups are also using it. Key data objects and components of the UOM spec are implemented in Energistics *common*.
9.	*Energistics Transfer Protocol (ETP)* (Must be downloaded separately: *http://www.energistics.org/standards-portfolio/energistics-transfer-protocol*)	A data-exchange specification that enables the efficient transfer of real-time data between applications. Specifically envisioned and designed to meet the unique needs of the upstream oil and gas industry and, more specifically, to facilitate the exchange of data for Energistics domain data standards (RESQML, WITSML, and PRODML).

1.5.3 Documentation Updates

Energistics is committed to providing quality documentation to help people understand, adopt, and implement its standards. As uptake of the standards increases, lessons learned, best practices, and other relevant information will be captured and incorporated into the documentation. Updated versions of the documentation will be published as they become available.

1.5.3.1 v2.0 Documentation Status

The focus for PRODML v2.0 has been on developing data objects together with comprehensive documentation for the new sub-domains that are listed above. You can find solid content in this guide to help guide you through the design and intended use of these new objects, and also the earlier DTS sub-domain objects.

Creation of documentation for the previously published versions of PRODML (some of which had little or no user documentation) is under development. This version 2.0 has good overview information as to the design and intended usage of all the older data objects. However, these older data objects typically do not have worked examples. Other material available is referenced or included where available.

1.5.3.2 Conventions

	Document/Resource	Description
	Document Hyperlinks: Internal	Though no special text-formatting convention is used, all document-internal section, page and figure number references in this and all PRODML and Energistics documents are hyperlinks. The table of contents is also hyperlinked.

2 Technology Overview of PRODML

Each of Energistics domain standards—RESQML, WITSML and PRODML—leverages components in the Energistics Common Technical Architecture (**Figure 2**–1).

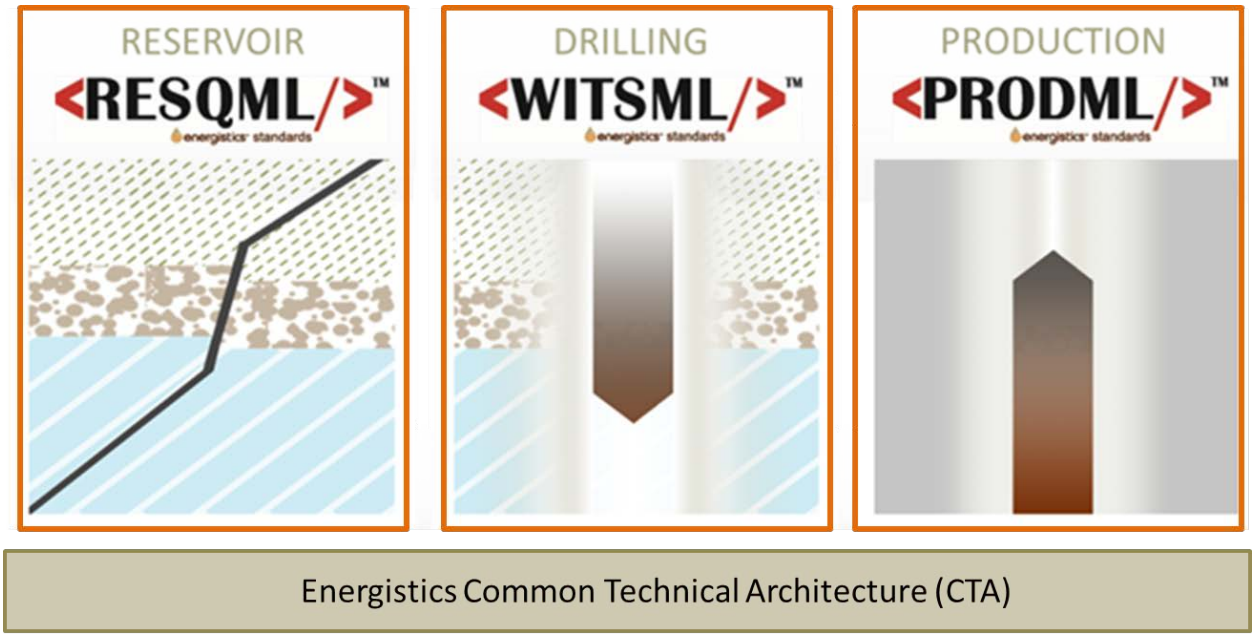


Figure 2–1. The Energistics family of standards includes WITSML, RESQML and PRODML, which share the Energistics Common Technical Architecture.

PRODML comprises a set of XML schemas covering a range of sub-domains of the whole scope of production management in upstream oil and gas. Additional components of PRODML are located within the Energistics Common Technical Architecture. PRODML is available as a downloaded zip file, which installs the schemas, together with examples and documentation, into a folder structure which includes the CTA elements.

This chapter describes:

- An overview of the main CTA components and the purpose of each.
- PRODML schema top-level data objects and the mapping of these onto the sub-domains of production management as outlined in Chapter 1.
- Usage of the elements of CTA in PRODML v2.0.
- Common expected usage of other Energistics domain standards by PRODML.

2.1 CTA: Main Components and What They Do

NOTE: For more information about standards used in the CTA, see the *CTA Overview Guide*.

Each Energistics domain standard (RESQML, WITSML, and PRODML) has its own set of schemas, which include a <domain>common package (e.g., ProdmlCommon) of schemas for consistency across each ML (refer to Figure 2–1). The underlying technology to define the schemas (XSD files) for the objects, artefacts, data, and metadata is XML, with HDF5 used for large numeric arrays. Each instance of a top-level data object must be identified by a universally unique identifier (UUID).

Each domain ML leverages components of the Energistics CTA.

Energistics standards are designed using the Unified Modeling Language (UML), which is also used to produce the schemas and some documentation.

- **Energistics *common* schemas.** The *common* package is standardized across all Energistics domain standards; for each of the MLs, the same *common* package is included in the download. Like the Energistics domain schemas, these schemas are also XML XSD files. Data object schemas can be considered in these categories:
  - Mandatory (for example, AbstractObject, ObjectReference, objects related to units of measure (UOM), etc.).
  - Optional, available for use if wanted (for example, the Data Assurance and Activity Model objects).
  - Objects defined by Energistics specifications (see next bullet), which may be optional or mandatory, depending on the specification and domain ML.
- **Energistics specifications** describe objects and behaviors for handling mandatory and optional functionality across domains. For example, units of measure, metadata, and object identification are mandatory. Other standards, such as packaging objects together for exchange, are optional or ML-specific. Related data objects are implemented in the Energistics *common* schemas. The specs describe additional behavior requirements. For the complete list of CTA resources, see Section 1.5.2.
  - **Energistics Transfer Protocol (ETP)** is the Energistics spec that serves as the new application programming interface (API) for all Energistics domain MLs. Initially designed to replace the WITSML SOAP API, ETP is based on the WebSocket protocol. It delivers real-time streaming capabilities and is being expanded to provide CRUD (create, read, update and delete) capabilities.
- **Information technology (IT) standards.** Energistics standard's leverage existing IT standards for various purposes. For example:
  - The Unified Modeling Language (UML) is used to develop the data model and produce the schemas and some documentation.
  - XML is used to define the data object schemas (XSD) and instances of data (XML files).
  - UUID (as specified by RFC 4122 from the IETF) is used to uniquely identify an instance of a data object.
  - HDF5 is used when needed as a companion to the XML data object to store large numeric data sets.

2.2 Mapping of the PRODML Sub-Domains onto Top-Level Objects

PRODML is defined by a set of schemas. These are found in the xsd_schemas folder (for more on the folders, see Section 2.5). The schemas are a set of top-level objects (as defined by the CTA), other than the two schema files highlighted with a red box in Figure 2–2.

The two schema files which are not top-level objects are:

1. Prodml All Objects: this includes all the other objects into one schema file. This is provided because certain software development environments can efficiently consume this file.
2. Prodml Common: this has various elements that are used by multiple PRODML top-level objects and therefore are put into one common location. This fulfils a similar role for PRODML as the schema files in the Energistics *common* package do for all the Energistics domain standards.

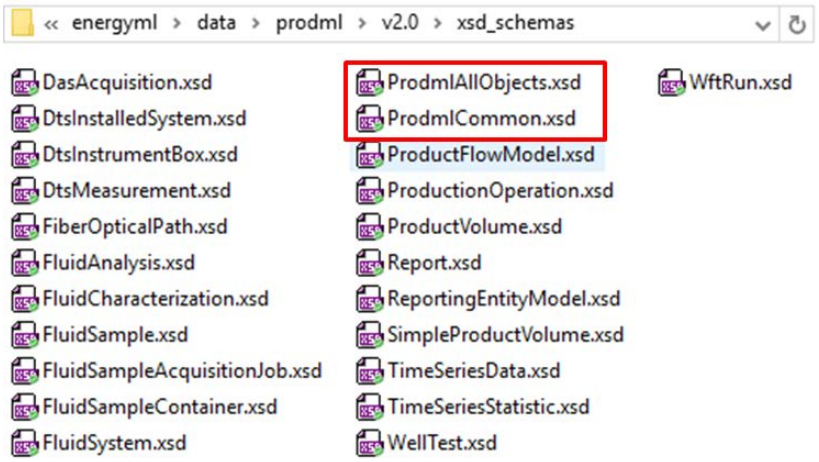


Figure 2–2. PRODML Schema files.

Most of the mappings of these schemas to the sub-domains of PRODML are evident from the file names. **Figure 2**–3 shows the mapping from PRODML sub-domains onto data objects. One additional factor is that the simple product volume schema file is an abstract top-level object, packaged in one schema file. There are five non-abstract top-level objects which are defined for this sub-domain. These schemas are shown with blue text. It can be seen that the more recent sub-domains tend to use multiple top-level objects (from 2 to 6 each in version 2.0), while the older PRODML sub-domains commonly have a 1-to-1 mapping to data objects (which tends to make those older schemas more complex).

	PRODML Sub-Domain										
Top Level Object Schema	**PRODML General**	**SPVR ***	**PVT**	**DTS**	**DAS**	**Wireline Formation Testing**	**Product Volume**	**Product Flow Model**	**Time series**	**Welltest**	**Production Operation**
DasAcquisition.xsd					1						
DtsInstalledSystem.xsd				1							
DtsInstrumentBox.xsd				1							
DtsMeasurement.xsd				1							
FiberOpticalPath.xsd				1	1						
FluidAnalysis.xsd			1								
FluidCharacterization.xsd			1								
FluidSample.xsd			1								
FluidSampleAcquisitionJob.xsd			1								
FluidSampleContainer.xsd			1								
FluidSystem.xsd			1								
ProdmlAllObjects.xsd	1										
ProdmlCommon.xsd	1										
ProductFlowModel.xsd								1			
ProductionOperation.xsd											1
ProductVolume.xsd							1				
Report.xsd	1										
ReportingEntityModel.xsd		1									
SimpleProductVolume.xsd **											
AssetProductionVolumes		1									
ProductionWellTest		1									
WellProductionParameters		1									
TerminalLifting		1									
Transfer		1									
TimeSeriesData.xsd									1		
TimeSeriesStatistic.xsd									1		
WellTest.xsd										1	
WftRun.xsd						1					

* SPVR = Simple Product Volume Reporting

** *Simple Product Volume is abstract*, from which these SPVR objects are derived

Figure 2–3. Mapping between PRODML sub-domains and schema files.

The Report top-level object is a “header” type object, which has been retained in this version of PRODML for continuity, but in version 2 it is less likely to be needed.

Having selected the sub-domain in which to work, you can open the schema files, and/or import the .XMI file containing the UML model (see Section 1.5.1) and import it into a UML tool for visualization. You should do this in conjunction with reading the rest of this chapter, and the chapters of this document concerned with the relevant sub-domain. You can also navigate to the appropriate example files (again, for details, see Section 1.5.1).

2.3 PRODML Use of CTA

As noted previously, PRODML makes extensive use of the CTA.

All the PRODML sub-domains use data objects, which are modelled in UML and defined by XSD schemas, and which use common data objects and units of measure.

DAS uses the additional CTA elements of:

- HDF5 (for high-volume data).
- EPC (for packaging multiple files together).

Other sub-domains could use the EPC, for example, multiple PVT XML files and other non-Energistics documents used to report PVT data could be packaged together using EPC. However, this has not been documented or tested.

The Energistics Transfer Protocol (which is not downloaded with PRODML; see Section 1.5.2) could potentially be used to transfer XML data objects created by use of PRODML. This capability also remains undocumented and untested.

Wider use of EPC and ETP are candidates for further PRODML development.

In terms of the Energistics common data types, PRODML makes extensive use of a package of data types called Value Types. These types are to cover measurements where the measurement conditions act as a qualifier to the measured value:

- Pressure: whether absolute or relative/gauge pressure has been measured; if relative or gauge, then the reference/atmospheric pressure must/may be provided.
- Volume, Density and Flowrate: where the pressure and temperature conditions of the measurement have a profound impact on the underlying “value” of the measurement. A choice is available—either to supply the pressure and temperature of measurement, or to choose from a list of standards organizations’ reference conditions. Note that the enum list of standard conditions is extensible, allowing for local measurement condition standards to be used.

The four types are called “xxxValue” where “xxx” is one of the four measurement types listed immediately above. The PressureValue is shown in **Figure 2**–4, and the other three types in **Figure 2**–5.

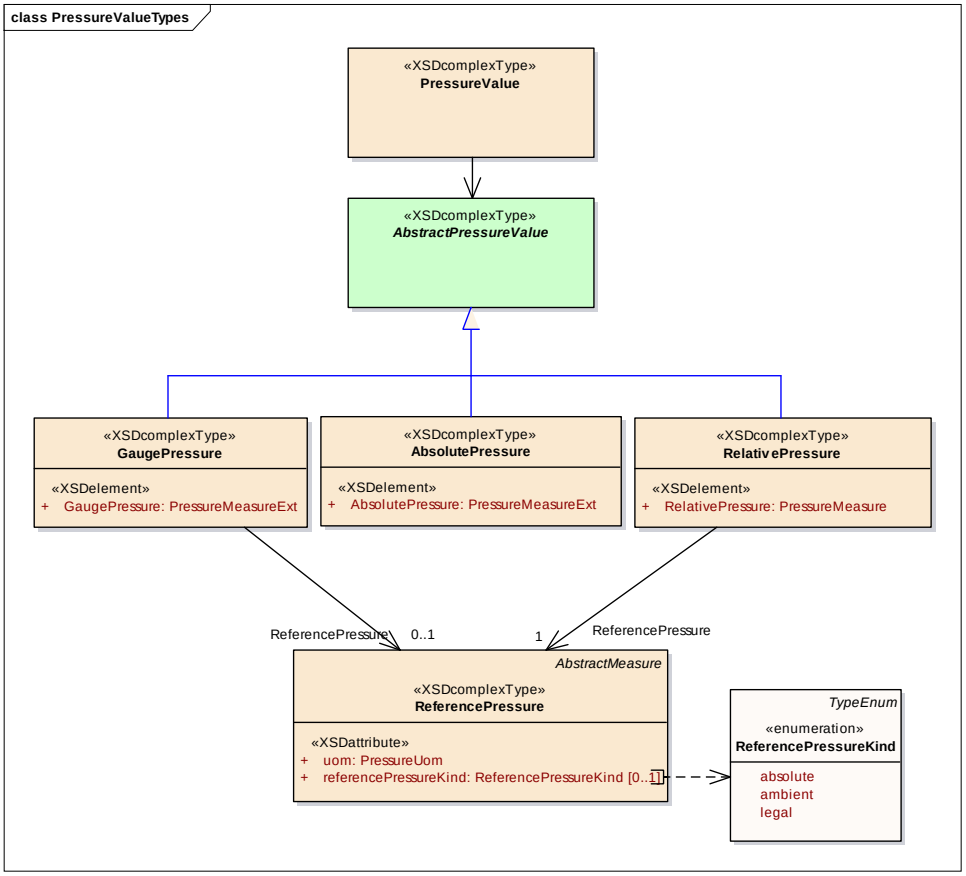


Figure 2–4. PressureValue data type.

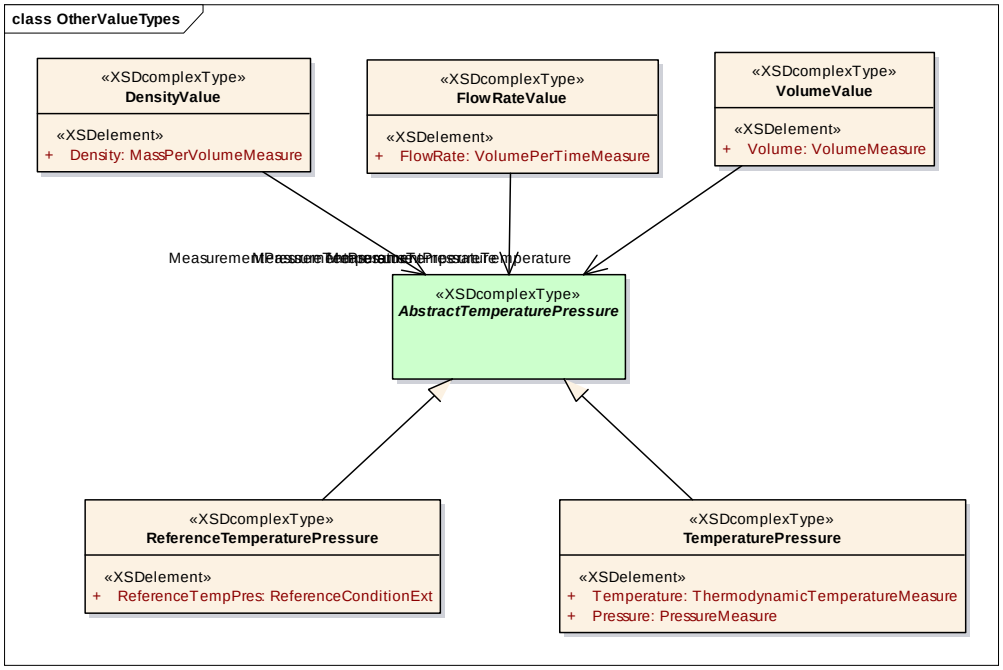


Figure 2–5. Density, Flowrate and Volume Value data types.

2.4 PRODML Use of Other Energistics Domain Standards

With version 2 of PRODML and WITSML, these domain standards join RESQML in using the Energistics CTA. This upgrade makes it possible to easily integrate data from the three domain standards. PRODML sub-domains are expected in common use (although these are not mandatory usages) with certain other WITSML and RESQML data objects.

The usage pattern is that data that is useful to the PRODML sub-domain can be referenced (using the data object reference element in Energistics *common*) from certain PRODML data objects. These are generally either:

- Well/wellbore/completion/rock-fluid unit feature references for reference to flow, sample or measurement sources. These are contextual references and so some degree optional. See **Figure 2**–6, pink cells (value 1).
- Log/Channel Set/Channel references so that number arrays can be transferred using these data objects, and referenced from PRODML. Note that for DAS arrays, HDF binary files are used in a similar manner to these log usages. These are references to objects which transfer essential data, and hence to use these, the WITSML schemas are required. See **Figure 2**–6, red cells (value 2.

Figure 2–3 lists these usages by referencing as outlined above, from each PRODML sub-domain to other domain data objects.

Other ML Element	SPVR *	PVT	DTS	DAS	wftRun	Product Volume	Product Flow Model	Time series	Welltest	Production Operation
WITSML: well	1	1							1	
WITSML: well completion	1	1								
WITSML: wellbore	1	1	1	1	1				1	
WITSML: wellbore completion	1	1								
WITSML: log					2					
WITSML: Channel Set			2							
WITSML: Channel					2					
RESQML: Rock Fluid Unit Feature		1								

* SPVR = Simple Product Volume Reporting
1 = a reference to an object from another ML for contextual information
2 = a reference to an object from another ML for transfer of essential information

Figure 2–6. Mapping between other Energistics "ML" objects and their usage in PRODML

The WITSML resources can be obtained by following the links at http://www.energistics.org/drilling-completions-interventions/witsml-standards/current-standards. They will install as shown in **Figure 2**–8.

2.5 Installed Components

This section explains the folder structure and content that is installed when any of the Energistics standards is downloaded. It then describes the specific PRODML content.

When the PRODML download is obtained from Energistics, it is a zip file. This file can be unzipped into a convenient location. The contents unzip into a folder called *energyml*, and a folder under this is created, called *data*.

If the Energistics ETP data transfer protocol is added (this is not part of the standard PRODML download), then at the same location, a second folder called *protocols* will be created. See **Figure 2**–7.

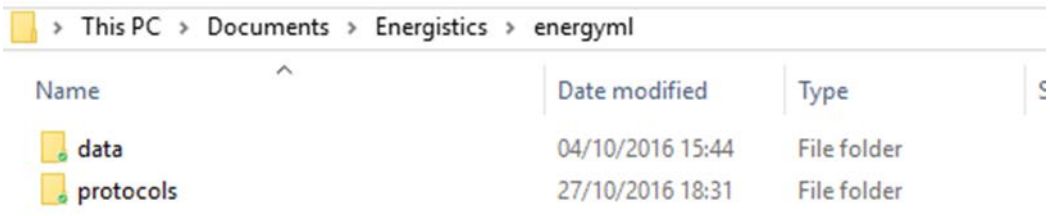


Figure 2–7. Energistics folder structure.

If PRODML has been downloaded, the *data* folder contains two folders:

- *common* contains the elements of the CTA, which are common across PRODML, RESQML and WITSML.
- *Prodml* contains the specific PRODML content.

Keeping these folders in this relative location means that the schemas all reference the correct paths to common elements. If another ML is added, then at the same location, a third folder called (e.g.) *Witsml* is created. See **Figure 2**–8**.** To use the WITSML objects as required by certain of the PRODML sub-domains (see Section 2.4); WITSML is required and will unzip as shown.

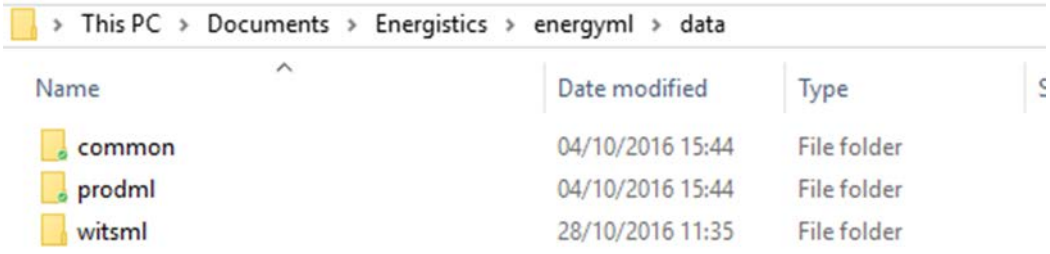


Figure 2–8. *Data* folder contents, example showing PRODML and WITSML installed.

For the contents of the common folder, please see Section 1.5.2, and for full details of these CTA elements, see the *CTA Overview Guide*.

The prodml folder contains a v2.0 folder (and later versions will appear here, e.g. v2.1) and within this is the folder structure shown in **Figure 2**–9.

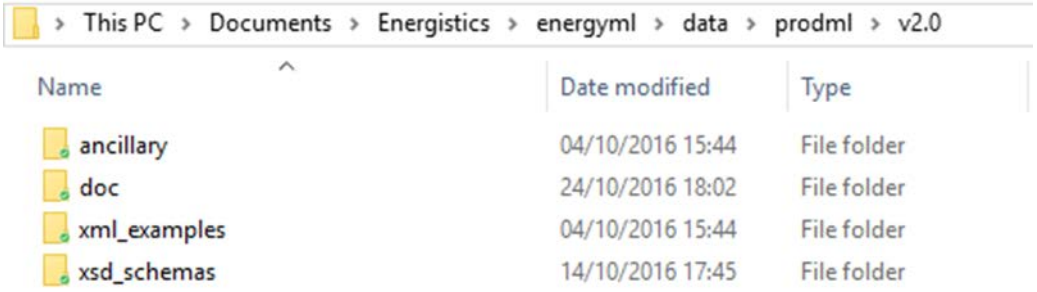


Figure 2–9. Contents of Prodml\v2.0 folder.

Details of the contents of these folders are supplied in Section 1.5.1.

Part II: Simple Product Volume Report

Part II contains Chapters 3 through 7, which explain the set of PRODML data objects for simple product volume reporting.

Acknowledgement

The detailed input into and leadership of this project by the following companies is gratefully acknowledged: P2 Energy Solutions, Oxy, Accenture, Energysys, Halliburton. The contributions in terms of helping set requirements from the following companies is also highly appreciated: IHS, WellEZ, Infosys, Anadarko, Baker Hughes, Chevron, QEP, Schlumberger, Statoil.

3 Introduction to Simple Product Volume Report Objects

This chapter serves as an overview and executive summary of the Simple Product Volume data objects.

3.1 Overview of Simple Product Volume Reporting, the NAPR Project, and the earlier PRODML models devised for the Norwegian Continental Shelf

NAPR stands for North America Production Reporting. The objective of the NAPR project was to analyze and extend (if needed) PRODML schemas necessary for the reporting of daily and monthly production related data to non-operating joint venture (NOJV) partners, and royalty owners in North America. The result is the Simple Product Volume package of PRODML, first released as part of PRODML v2.0.

Although the focus of the project has been North America requirements, the hope is that the data standard can be used worldwide.

For a quick overview or to be able to make a presentation to colleagues, see the slide set: Worked Example SPVR.pptx which is provided in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files.

Note that the Simple Product Volume capability is, as the name suggests, a standard which aims to cover the minimum requirements for volume reporting. It was developed following widespread feedback that the capabilities within earlier versions of PRODML, whilst comprehensive, were also complex and therefore hard to understand and to implement.

The earlier version of volume reporting does remain part of PRODML. It comprises principally the data object Product Volume, and the supporting objects Product Flow Model, WellTest and Production Operation. This reporting standard has been adopted (and extensively developed) for use on the Norwegian Continental Shelf (NCS). The NCS reporting is based on version 1.x of PRODML. The data objects concerned have been migrated to Energistics CTA standards but otherwise left unaltered, for compatibility with previous work. A fuller description can be found in Chapter 24, Product Volume; Chapter 25, Product Flow Model; Chapter 27, WellTest; Chapter 28, Production Operation.

3.2 The Business Case

Some of the challenges of reporting production data to other relevant entities are:

- Lack of a standard (protocol, format and mechanism).
- Associated requirement for a significant amount of manual effort both in creating data to be transmitted, and receiving/consuming it.
- NOJV partners are increasingly requesting more detailed data (daily and monthly), resulting in a significantly increased quantity of data to be transmitted.
- Timelier reporting needed to support decision making (e.g., marketing, participation in projects) and internal reporting (accruals, etc.).

Therefore the benefits expected are:

- Streamlined approach to communicating production related data.
- Less manual effort and thus reduced cost.
- Greater accuracy (due to less manual intervention/preparation).
- Greater insight into the status of properties operated by others (OBO) supporting more informed decision making.
- Quicker access to data from joint venture partners.

3.3 Scope of Simple Product Volume Data Objects

The scope of this section pertains specifically to the Simple Product Volume data objects. Data in scope for the current version includes:

- The standards established and lessons learned by the Norwegian Production Reporting Guidelines using the Product Volume object first released in PRODML v1.1, and retained (updated to Energistics Common Technical Architecture style) in PRODML v2.0. Most of the data transfers supported by this standard have been included in the Simple Product Volume package. See the note in Section 3.1 concerning the earlier data objects.
- The previous work by the National Data Repository workgroup has been considered and leveraged, where appropriate, in developing North American standards.

Out of scope are:

- Regulatory reporting in North America, which is complex and varies by state.
- Regulatory reporting in the rest of the world, although it is hoped that in many cases the current data model will be adequate for this purpose.
- Security infrastructure, such as user identity and authentication, for example used in transfer of data between partners.

3.4 Use Cases: Overview

The Simple Product Volume data objects support these use cases:

1. Receive monthly and/or daily production data directly from the operating partner;
2. Transmit monthly and daily production data directly to a participating joint venture partner;
3. Provide historical production data (various frequencies and date ranges possible) for divested properties;
4. Obtain historical production data (various frequencies and date ranges possible) for acquired properties;
5. Transmit monthly and/or daily production data on multiple wells to a central data exchange (optionally including a list of participating entities by well so that the exchange can limit access to the production data as appropriate);
6. Obtain monthly and/or daily production data from a central data exchange.

For the detailed use cases, see Chapter 7.

3.5 Data Object Overview

The set of Simple Product Volume Reporting data objects work together to cover the following capabilities:

- Describe the Reporting Entities:
  - List of entities, any arrangement of hierarchies, reference the data for the physical entity
- Report Volumes per Reporting Entity:
  - Production, injection, dispositions, deferred production and inventories, transfers to other entities, terminal lifting
  - For any time period; any type of produced or service fluid
- Transfer of event-driven reports and supporting data:
  - Well tests, well production parameters

4 Simple Product Volume: Use Cases, Data Types, and Workflow

This chapter provides a brief over of the business context for using the Simple Production Volume data objects, the use cases they support, key data types, and a high-level workflow.

4.1 Business Context: Asset Production Volume Flows

In general, oil and gas fields consist of largely discrete “assets”, administered by different operators. This means that production flows are measured and reported on a per-asset basis. Importantly, the financial imperative of custody transfer ensures that any inter-asset transfer flows or exports are also measured and to a high standard.

An asset in this context is a set of interlinked oil production infrastructure administered by a single operator. **Figure 4**–1 shows the generalized asset group flow. The Simple Product Volume data objects are designed to handle the requirements for reporting the production flows of an asset as generalized in this figure.

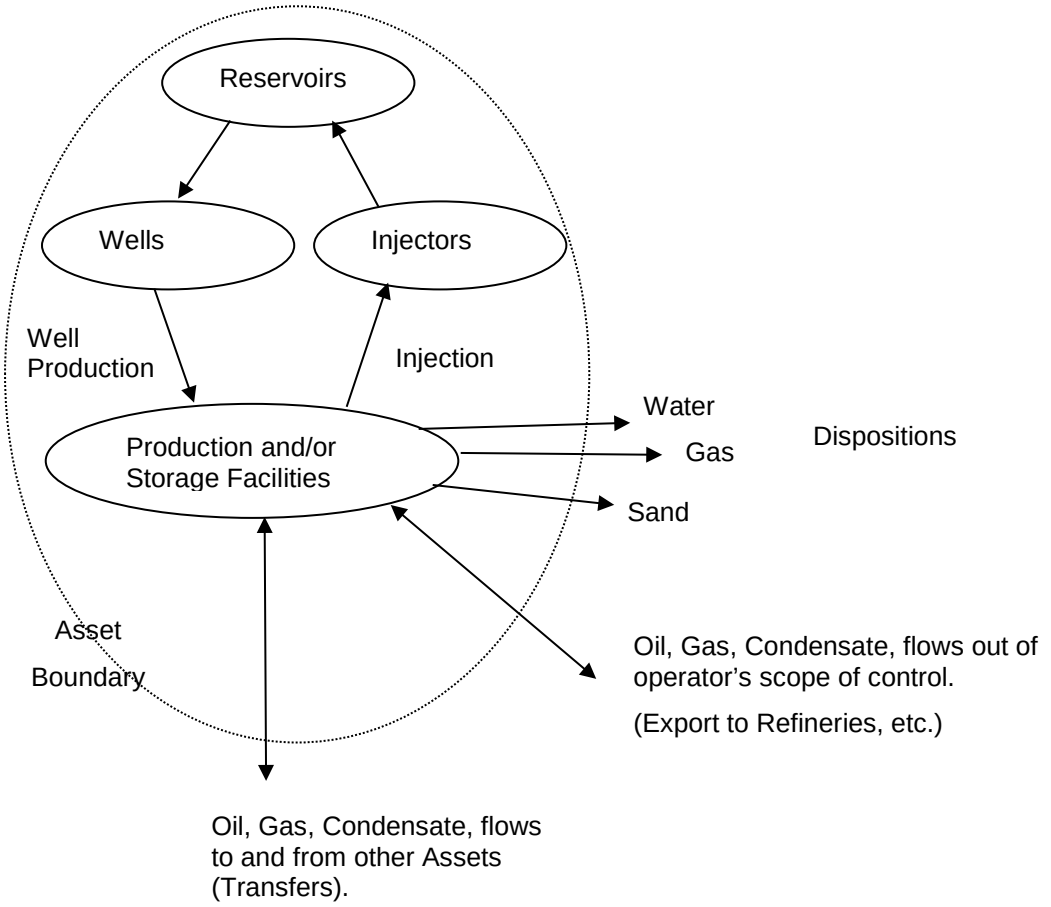


Figure 4–1. Asset production flows.

4.2 Use Cases: Overview

For details of these use cases, see Chapter 7.

1. **Receive monthly and/or daily production data directly from the operating partner.** The goal is to provide to others with working and/or revenue interest in jointly owned properties: production data for a fixed duration necessary for monitoring, decision-making, forecasting and reporting, and financial record-keeping associated with operated properties.
2. **Transmit monthly and daily production data directly to a participating joint venture partner.** This is the "mirror image" of Use Case 1.
3. **Provide historical production data (various frequencies and date ranges possible) for divested properties.** The goal is to reduce cost and effort and support automation of data room presale activities and generate added value for properties being sold, as well as for those purchasing properties.
4. **Obtain historical production data (various frequencies and date ranges possible) for acquired properties.** The goal is to reduce cost and effort of post-acquisition data loads and improve completeness and accuracy of loaded data.
5. **Transmit monthly and/or daily production data on multiple wells to a central data exchange**. The goal is to ensure that when data is exchanged, the sender can specify the rules for access to the data, such that the receiver may enforce privacy rules specified by the data owner. Therefore, the transmission should optionally include a list of participating entities by well so that the exchange can limit access to the production data as appropriate.
6. **Obtain monthly and/or daily production data from a central data exchange**. This is the "mirror image" of Use Case 5. The functionality depends on the data exchange having information concerning the rights of different users to access specific items of data.

4.3 Key Data Types

The data types required for Simple Product Volume include:

- Well identification information (name, unique well identifier, owner well number)
- Composition of hierarchies for reporting (e.g., groups of wells in a field)
- Total produced volumes of oil, gas, NGL, and water by well
- Volumes sub-divided by other entities such as formation, string, lease, field
- Volumes sub-divided by final disposition (sales, fuel, injection, etc.)
- Reporting of periodic production over different periods (e.g. daily and monthly)
- Temperatures and pressures
- Wellhead measured volumes (where applicable)
- Producing status
- Hours operated/down with downtime reasons and comments
- Well tests
- Parties authorized to access data by well

The data model is described Chapter 5 (page 26) and a full worked example showing these data types is described in Chapter 6 (page 32) and included in the downloaded PRODML v2.0 zip file.

4.4 Typical Workflow

A data exchange using the Simple Product Volume data object is expected to proceed as follows:

1. Establish the "reporting entities" (e.g. the wells, completed intervals, fields, etc., against which volumes are reported) between sender/reporter and receiver/partner. The sender needs to send the

details of the reporting entities just once (and updated them if they change). Subsequent transfers only need to include the volumes being reported against the previously-transferred entities.

2. On an agreed periodic basis, transfer the volumes associated with these entities.
3. On an event-driven basis, transfer other important ancillary data, e.g., well test data, well production parameter changes, tanker lifting, or transfers of product in and out of the asset.
4. Once per periodic or event-driven transfer, describe the fluids whose volumes are being reported (e.g., oil, gas, water and any details of these).
5. If the configuration of the asset changes, transfer the changes/updates. For example, if a new well is completed, the sender must send to the receiver the new well information and its place in the reporting entity hierarchy.

5 Simple Product Volume: Data Model

This chapter gives an overview of the Simple Product Volume data objects, how they are used, and how they are related.

Chapter 6 gives a detailed description of the worked example (that is downloaded with the Simple Product Volume standard) and also gives more details on the data model and how it works.

5.1 Overview of the Simple Product Volume Data Object

Figure 5–1 shows the main data types and relationships between them that comprise the Simple Product Volume data objects; this set of objects works together to cover these capabilities:

- Describe the reporting entities:
  - List of entities,
  - Any arrangement of hierarchies,
  - Reference the data for the physical entity
- Report volumes per reporting entity:
  - Production, injection, dispositions, deferred production and inventories, transfers to other entities, and terminal lifting
  - For any time period, any type of produced or service fluid
- Transfer supporting data:
  - Well tests, well production parameters

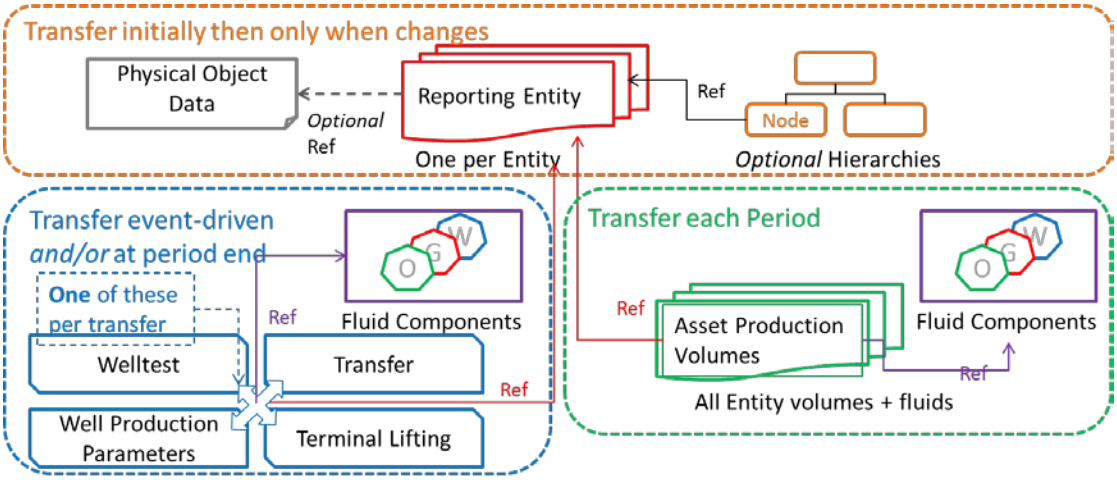


Figure 5–1. Overall data model for Simple Product Volume showing reporting entities section (orange border), periodic volumes section (green border) and event-driven section (blue border).

The Reporting Entity object is important because all the volume and event-driven data transfers use it to refer to the "thing" whose properties they are reporting. **Figure 5**–2 shows a UML diagram of the data model (simplified to the top levels); it shows how the other types of data objects reference back to the reporting entity for identity. (NOTE: The UML model is used to produce the PRODML schemas and is provided (as an XMI file) when you download the standard.)

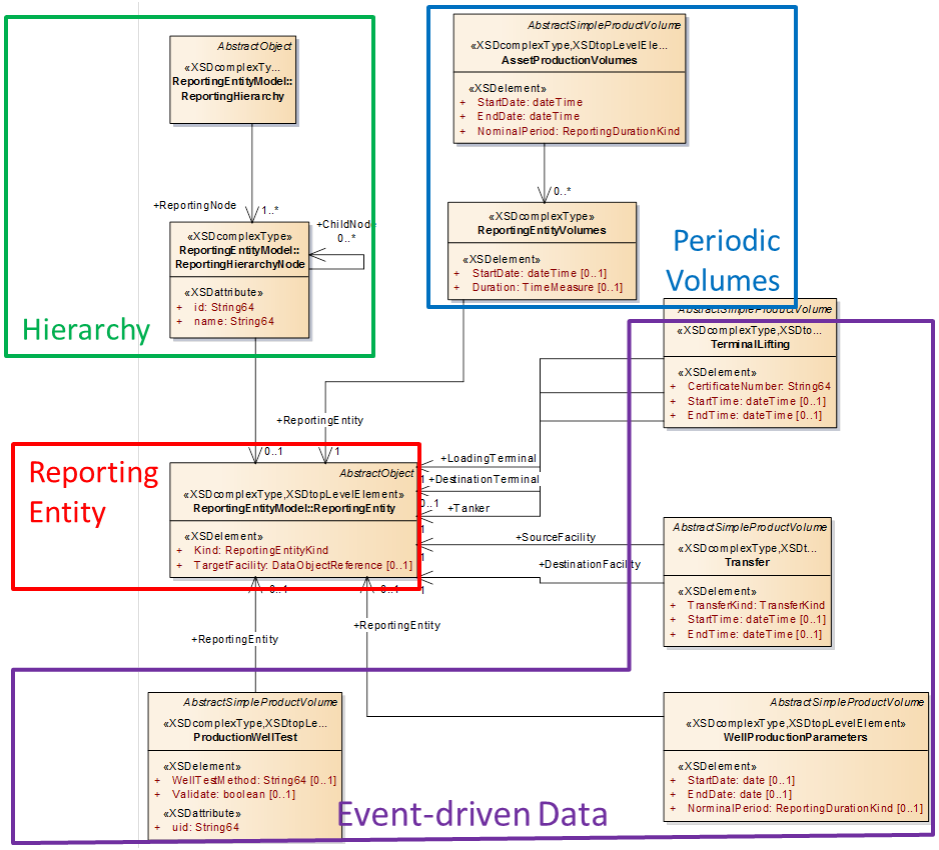


Figure 5–2. Top-level UML model diagram showing central role of reporting entity.

5.2 Describing the Reporting Entities

It is important to be certain that volumes are being reported against the correct component within an asset; that is, production is reported for the correct well. For this reason, one function of the Simple Product Volume data objects is to satisfy the requirements of identity of asset components, or “reporting entities” as they are called here. The reporting entity functionality is sub-divided into: defining the entities, showing how they are arranged in hierarchies, and linking them to any available detailed descriptions. The following sections describe this.

5.2.1 Reporting Entity Defined

A reporting entity refers to a physical, organizational or geographic “thing” that production data is reported against. For example: wells, fields, leases, business units, countries or states are reporting entities.

At its basic level, the reporting entity data object is simply a “placeholder” object, which all other Simple Product Volume data objects reference. That is, the object identifies the name or ID of the entity against which production data is being reported, but not much other information about it.

Optionally, you can provide other additional data for a reporting entity, using one or both of these methods:

- Define hierarchies to give appropriate context. For example, a hierarchy might be: business unit, fields within a BU, wells within a field, and wellbores within a well (see Section 5.2.2).
- Reference a physical data object. For example, a reporting entity that is a well or wellbore can reference a fully defined well or wellbore in WITSML (see Section 5.2.3).

Additional information:

- Reporting entities may be of different kinds, which are enumerated in the model (for list of enumerations, see Section 7.2):
  - Physical asset, e.g. well
  - Geographical, e.g. State
  - Organizational, e.g. Company
- The behavior of reporting entities according to Kind is not enforced; for example, the schema has no mechanism to restrict well tests being associated only to reporting entities whose kind is “well”.
- Aliases can be used within a reporting entity to enable alternative identifiers to be used (e.g., a well’s name in multiple systems, if they differ).

5.2.2 Reporting Entities can be Organized into Hierarchies

The key features of the Hierarchy data object are as follows:

- Hierarchies can be defined for reporting entities. To view this conceptually, see **Figure 5-1** and as a UML construction, see **Figure 5-2.**
- As many hierarchies can be included as are required for different purposes. For examples of the use of different kinds of hierarchy, see **Figure 6-6**.
- Hierarchies have a name, but no enumerated kind.
- Hierarchies have a root “reporting node”, which then can have as many child “reporting hierarchy nodes” in as many levels as required (Figure 5-2).

5.2.3 Reporting Entities Can Reference “Full” Data Objects

Optionally, reporting entities can reference a data object containing data about the physical object concerned (see Figure 5-1 and the link marked “Optional Ref”). Section 5.2 explains how a reporting entity is essentially a “placeholder” for the volume reporting. It contains very little actual data about the entity. However, you can provide a link to another data object that contains more data about an entity; for example, a well reporting entity can link to a well data object in WITSML.

This reference is done using a data object reference, which is a mechanism in the Energistics CTA that can reference any v2.0 data object from any of the Energistics standards (PRODML, WITSML, and RESQML).

For more information on how data object reference works, see 6.2.2.

For a list of example current Energistics data objects, see the right hand column in the table of Reporting Entity Kinds in Section 7.2.

5.3 Asset Production Volumes

The Asset Production Volumes object transfers *all* volume data for *all* reporting entities. This section explains the details of the use of this data object. The worked example in Chapter 6 covers all these usage patterns with an illustration of actual use. Some general points are:

- Each reporting entity is taken in turn and all its volumes are reported. The Reporting Entity Volumes contains all the volumes for a single reporting entity. See Figure 5-2 (blue box) where the overall period (time start and time end) is contained in the asset production volumes “container” and then a repeating pattern of reporting entity volumes follows.
- Reporting entity volumes reports all volumes for a single reporting entity, and contains a reference back to the reporting entity using its UUID for reference.
- Although named “volumes” in line with industry usage, different quantities may be reported, such as volume, mass, and energy content.

- Where an actual volume measurement is reported, this will of course be dependent on the measurement conditions of temperature and pressure. The purpose of SPVR is to transfer volume and similar quantities for internal, partner and regulator reporting, not to transfer field operational measurements. Hence the assumption is made that all volumes have been expressed at the same temperature-pressure conditions for the current transfer. The Standard Conditions element is mandatory in all the SPVR objects. A choice is available – either to supply the temperature and pressure for all the volumes which follow, or to choose from a list of standards organizations' reference conditions. Note that the enum list of standard conditions is extensible, allowing for local measurement condition standards to be used. See Figure 2-5 which shows the Abstract Temperature Pressure class: this is the type for standard conditions in all SPVR objects.
- Use is also made within SPVR of the Pressure Value type, allowing absolute or relative pressures. See Section 2.3 and **Figure 2-4**.

5.3.1 Fluid Component Catalog

Fluid components are specified by selecting them from the Fluid Component Catalog, which lets you specify fluid by component types or composition. NOTE: The Fluid Component Catalog is transferred only once per asset production volume data object; all volumes contained within reference the appropriate components from the catalog.

Within an asset production volumes data object, each instance of a quantity refers to one of the members of the Fluid Component Catalog in that same asset production volumes, using the UID for reference.

The fluid components in the catalog include:

- stock tank oil
- natural gas
- formation water
- pure fluid component
- plus fluid component
- pseudo fluid component

The first three are aimed at black oil descriptions and the second three at compositional descriptions; however, the two kinds may be mixed quantities of produced fluid, which can be described in either black oil or compositional terms, or both. The schema shows the attributes of each type of fluid component.

Note that the Fluid Component Catalog and the Fluid Components are contained in the PRODML Common package, and are shared with the PVT data objects, so are compatible with lab analysis reports created using PRODML v2.

5.3.2 Different Volume Types

Asset production volumes support transfer of these types of volumes:

- Production:
  - Production
  - Injection
- Inventory:
  - Opening Inventory
  - Closing Inventory
- Dispositions:
  - flared
  - sold
  - used on-site

  - fuel
  - vented
  - disposal
  - gas lift
  - lost or stolen
  - other

- Movement in or out of an asset:
  - Terminal Lifting
  - Transfer

- Deferred Production

These different types of volumes, as required, all sit within a single reporting entity volumes element in the asset production volumes object. Each quantity refers to a common Fluid Component Catalog as described in Section 5.3.1.

Note that the two kinds of movement in or out of asset can be reported either as elements within the periodic asset production volumes object, or in an event-driven manner with standalone Terminal Lifting and Transfer data objects (see Section 5.5).

The worked example (Chapter 6 shows many of these different volume types.

5.4 Well Tests and Well Production Parameters

Well tests and well operating parameters can be transferred as part of the Simple Product Volume capability. Both these types of transfer are designed to be transferred upon events happening or upon demand, rather than periodically as for asset production volumes. For this reason, they are available as standalone objects (See Figure 5-2, purple box).

The Production Well Test object and the Well Production Parameters use the data object reference mechanism (described in Section 5.2.3 and shown in Figure 5-1 and Figure 5-2) to reference a single Reporting Entity. The kind of Reporting Entity is expected to be a well (or possibly a wellbore or wellbore completion). Note however that the behavior of reporting entities according to the kind enumeration is not enforced—that is, the schema has no mechanism to restrict well tests being associated only to a reporting entity whose kind is “well”. The checking of this enum would need to be validated by software implementing the Simple Product Volume object.

Both the Production Well Test object and the Well Production Parameters object use the same reference to fluid components in a fluid component catalog as the Asset Production Volumes object (described in Section 5.3.1). Each instance of production well test or well production parameters has its own fluid component catalog so that it can behave as a standalone data object.

5.5 Transfer and Tanker Lifting

Volumes associate with transfers and tanker liftings can be transferred as part of the Simple Product Volume capability. Again, both these types of transfer are designed to be transferred upon events happening, rather than periodically as for asset production volumes. For this reason, they are available as standalone objects (see Figure 5-2, purple box). However, unlike the production well test object and the well production parameters objects, they can also be transferred embedded within an asset production volumes data object. The reason for this is an operator may wish to do an end-of-period (typically end-of-month) report showing all production and all dispositions of production, including all transfers and tanker liftings. When used within asset production volumes in this way, transfers and tanker liftings are to be identified as specific types of disposition.

The transfer object and tanker lifting object both use the same data object reference described above (and shown in Figure 5-2). However, they differ in that they refer to multiple reporting entities.

Transfer: 2 Reporting Entities:

- source facility
- destination facility

Tanker Lifting: 3 Reporting Entities:

- loading terminal
- destination terminal
- tanker

For facility and terminal entities, the reporting entity should be some kind of facility. For oil tanker and tanker trucks, the reporting entity is expected to be tanker. As noted above, the behavior of reporting entities and kind is not enforced by the schema, so it needs to be validated by the software (that the data objects are implemented in).

Both the Transfer object and Tanker Lifting object use the same reference to fluid components in a fluid component catalog as does the Asset Production Volumes (described in Section 5.3.1). Each instance of transfer and tanker lifting has its own fluid component catalog so that it can behave as a standalone data object. However, where they are used within Asset Production Volumes (see above), then they share the fluid component catalog of the parent asset production volumes.

6 Simple Product Volume: Worked Example

This chapter contains details of the worked example included with the download. The contents of the example are covered in the same order as the model has been described in Chapter 5.

In addition to the example data files which are in *energyml\data\prodml\v2.0\doc\examples\SPVR*, the SPVR example also includes:

- A presentation file found at: *energyml\data\prodml\v2.0\doc\Worked Example Simple Product Volume.pptx*
- A spreadsheet file found at: *energyml\data\prodml\v2.0\doc\examples\SPVR\Worked Example Simple Product Volume Spreadsheet.xls*

Note that extensive use is made of the data types "xxxValue" where "xxx" can be volume, flowrate or density, allowing for these measurements to be reported at specific pressure and temperature, or pressure, allowing absolute or relative/gauge pressure to be reported. See Section 2.3.

6.1 Reporting Entities: Physical and Commercial Layout

A set of Reporting Entities are provided to support the example of a small field containing 5 wells but with sufficient complexities to demonstrate the main features of the data transfer model.

Figure 6–1 shows the reporting entities for the example. The physical reporting entities include:

- 5 wells (Well 1, 2, 3, 4, 5, which intersect 2 reservoirs that each have 3 contact intervals):
  - Reservoir A
    - Contact Interval 1-A, 2-A and 3-A
  - Reservoir B
    - Contact Interval 3-B, 4-B and 5-B
- Note that Well 3 intersects both reservoirs at Contact Intervals 3A and 3B.

Commercially, these wells sit in Lease X. A group called *ABC Interest* (perhaps land owners) has an interest in Wells 1-4, and the *XYZ Company* (e.g., a non-operating partner) has an interest in Wells 3-5. So ABC Interest and the XYZ Company have commercial interest; therefore, are candidates to receive reports about production from these assets/reporting entities.

The XML for the example is explained below, but you can begin to see the various physical and commercial reporting entities that might arise from this simple example.

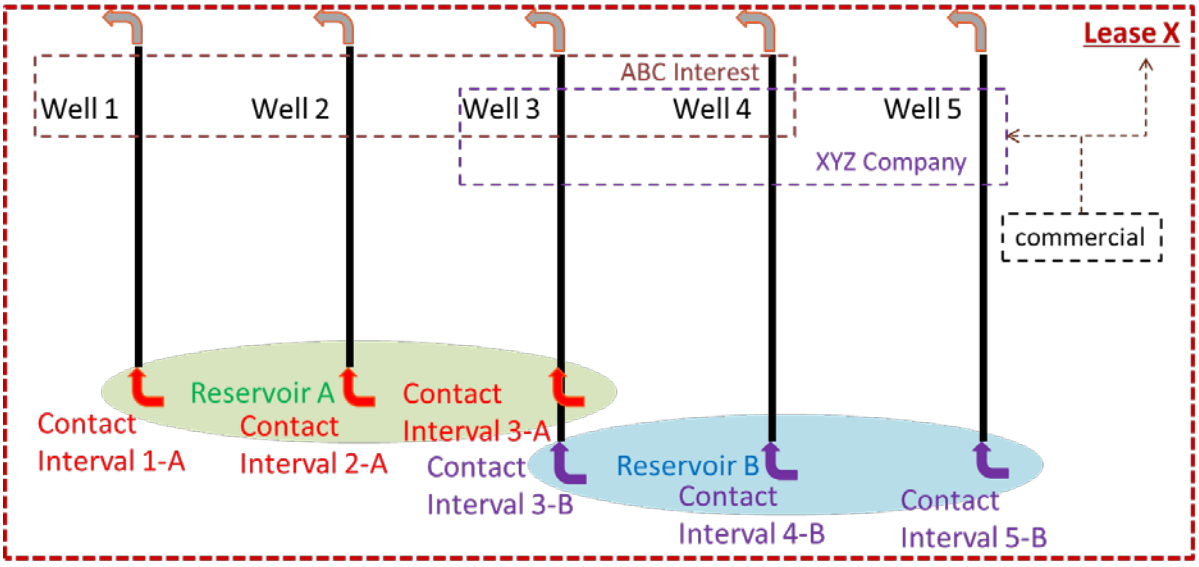


Figure 6–1. Reporting entities: physical and commercial layout.

The asset description is completed by **Figure 6**–2, which shows the reporting entities outside the asset's direct control and which play a part in transfers and terminal Liftings. These are a separate Lease which transfers production to Lease X, and an oil tanker, Barge 99, which after a tanker lifting, transports production to Terminal Z.

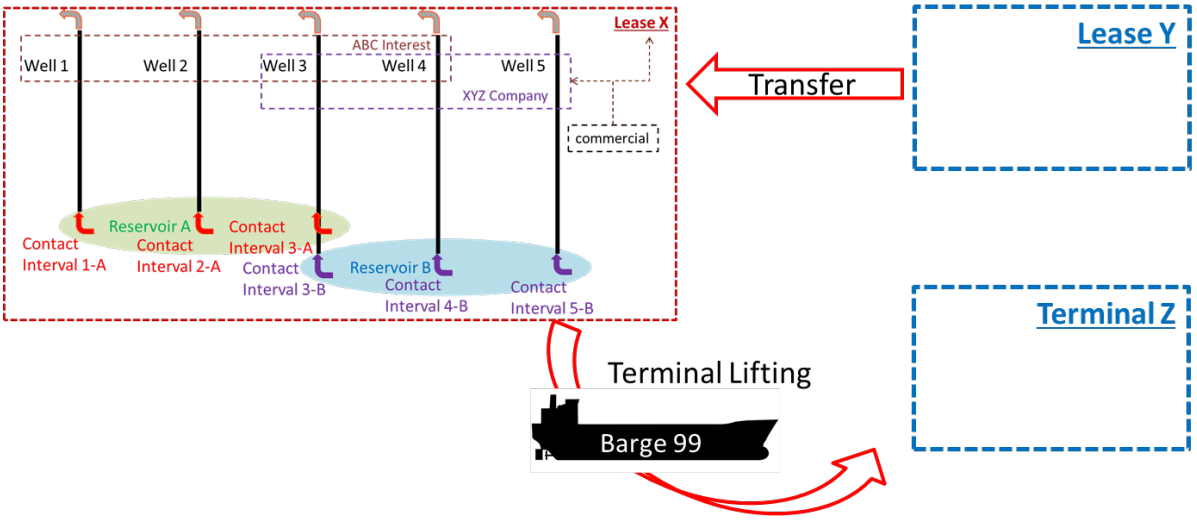


Figure 6–2. Reporting entities: other related facilities.

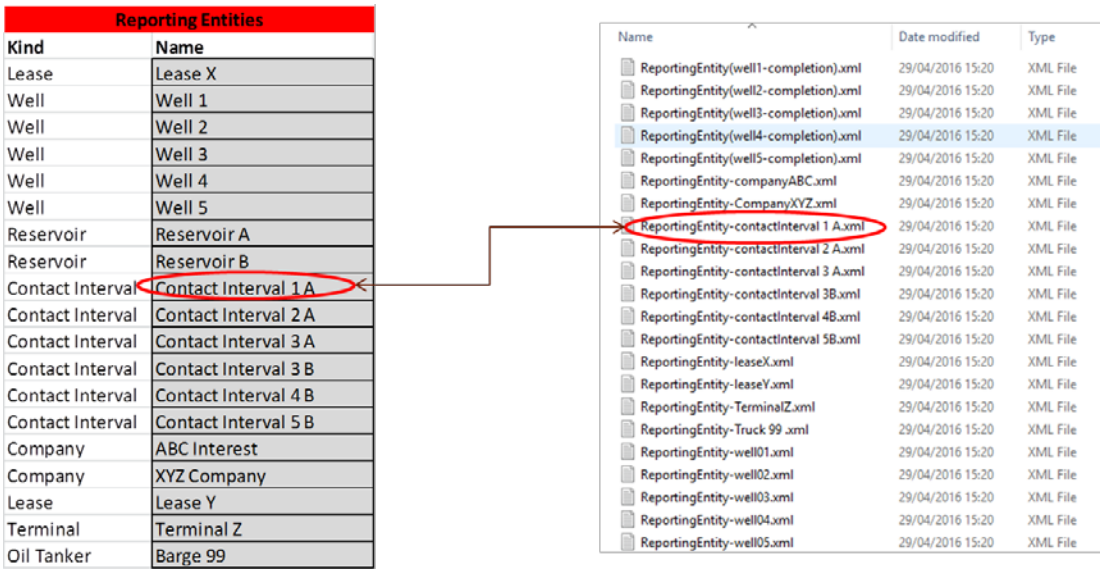
6.2 Reporting Entities: Organization in the Data Transfer

With the set of facilities visualized as shown in Section 6.1, you need to create a set of reporting entity data objects to define them.

6.2.1 XML Objects

The set is listed in Figure 6–3. The left side of the figure lists the entities by kind and name, and the right side shows a view of the folder containing one XML reporting entity data object per entity. In the worked example, these files can be found in the sub-folder "ReportingEntities".

Note that this set of data is expected to be transferred only at the start of the reporting relationship. Any new entities (for example, a new well), require that the new XML data object be transferred when needed.

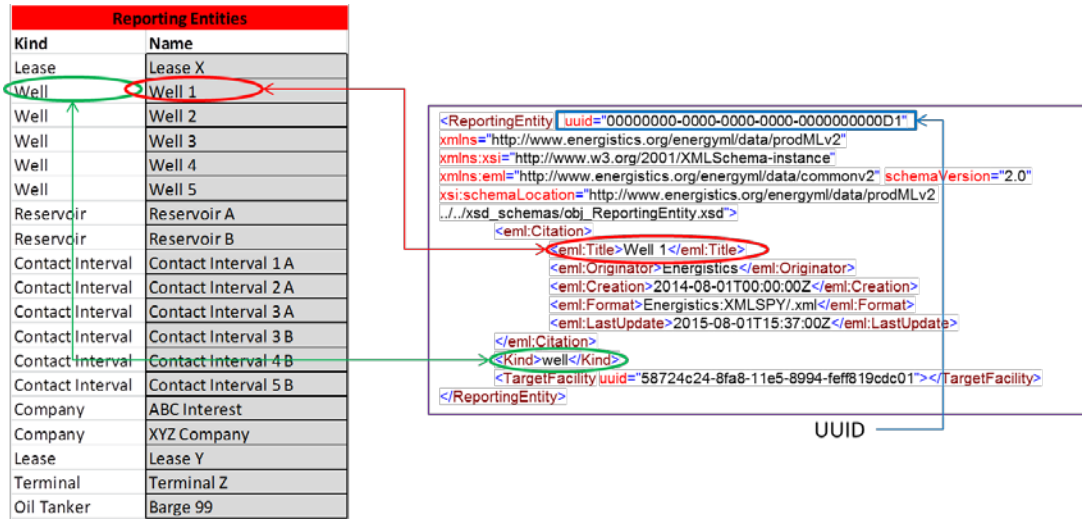


Reporting Entities	
Kind	Name
Lease	Lease X
Well	Well 1
Well	Well 2
Well	Well 3
Well	Well 4
Well	Well 5
Reservoir	Reservoir A
Reservoir	Reservoir B
Contact Interval	Contact Interval 1 A
Contact Interval	Contact Interval 2 A
Contact Interval	Contact Interval 3 A
Contact Interval	Contact Interval 3 B
Contact Interval	Contact Interval 4 B
Contact Interval	Contact Interval 5 B
Company	ABC Interest
Company	XYZ Company
Lease	Lease Y
Terminal	Terminal Z
Oil Tanker	Barge 99

Figure 6–3. Data objects in the test data set: reporting entities.

Figure 6–4 shows the details of one reporting entity object. Important to note are:

- The UUID is used in the other objects (hierarchies, asset production volumes, production well test, well production parameters, transfer and terminal lifting) to identify the reporting entity(ies) concerned (blue box).
- Title is the element in the Energistics Common Citation class to represent the name (red oval).
- The kind is the enum of reporting entity kinds (green oval).



Reporting Entities	
Kind	Name
Lease	Lease X
Well	Well 1
Well	Well 2
Well	Well 3
Well	Well 4
Well	Well 5
Reservoir	Reservoir A
Reservoir	Reservoir B
Contact Interval	Contact Interval 1 A
Contact Interval	Contact Interval 2 A
Contact Interval	Contact Interval 3 A
Contact Interval	Contact Interval 3 B
Contact Interval	Contact Interval 4 B
Contact Interval	Contact Interval 5 B
Company	ABC Interest
Company	XYZ Company
Lease	Lease Y
Terminal	Terminal Z
Oil Tanker	Barge 99

Figure 6–4. Details of the reporting entity object.

6.2.2 Reference to a WITSML Well Data Object

Optionally, a reporting entity can reference a data object containing full data. An example of this is shown in **Figure 6**–5. The XML on the left shows a reporting entity object, in this case a well. In this example, more information about the well is desired, so a link to a WITSML well, object is specified. The right side of the figure shows the referenced WITSML well object.

The kind of reporting entity being a well, linked to a Well object, is not enforced but is expected (red circle).

The link is made using the Energistics Common data object reference class (from the Energistics CTA), which is named Target Facility. The key data element is the UUID, which is the same as the well object's own UUID (blue box).

In the worked example, these "full data" object files can be found in the sub-folders "Wells" and "wellboreCompletions". Examples are provided for all the 5 wells and 6 wellbore completions listed in Figure 6–4.

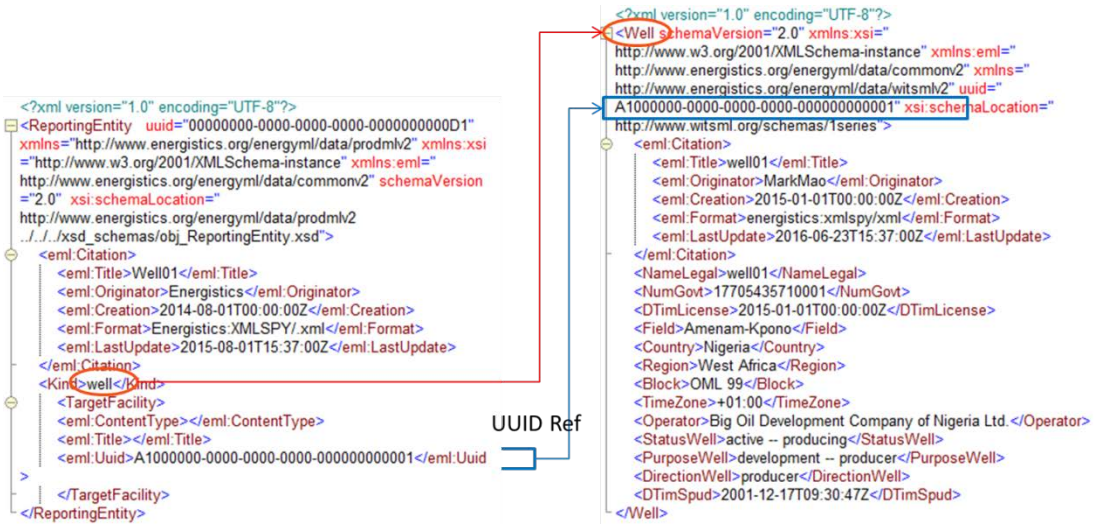


Figure 6–5. Optionally, reporting entities can reference a data object containing full data.

6.2.3 Defining the Hierarchies

The worked example contains 3 example hierarchies. As noted above, any number can be created to show the context of the reporting entities. **Figure 6**–6 shows the hierarchies included. The left side of the figure lists the hierarchy names and underneath, the hierarchy of nodes. The right side shows these as 3 XML data objects of type reporting hierarchy. These can be found in the sub-folder "ReportingHierarchies". The 3 hierarchies demonstrate:

- Typical order for volume reporting: lease *(field*)/well/contact interval *(layer)*
- Order for reservoir offtake reporting: lease *(field*)/reservoir/contact interval *(layer)*
- Order for commercial relationships, e.g. for defining access to data: company *(commercial entity*)/well

Again, you only need to update the hierarchy object when changes are made (e.g., a new well comes on line, etc.).

Note that the third hierarchy supports the requirement of Use Case 5 and 6 (see Section 4.2) in which reporting is done via a third-party "hub". This hierarchy makes it possible to list all the commercial entities associated with the asset being reported, and which require access to data pertaining to particular reporting entities, e.g., the different well interests shown in Figure 6–1.

Hierarchies		
By Lease, Well and Layer	By Lease, Reservoir and Layer	By Company and Entity
Lease X	Lease X	ABC Interest
Well 1	Reservoir A	Well 1
Contact Interval 1 A	Contact Interval 1 A	Well 2
Well 2	Contact Interval 2 A	Well 3
Contact Interval 2 A	Contact Interval 3 A	Well 4
Well 3	Reservoir B	XYZ Company
Contact Interval 3 A	Contact Interval 3 B	Well 3
Contact Interval 3 B	Contact Interval 4 B	Well 4
Well 4	Contact Interval 5 B	Well 5
Contact Interval 4 B		
Well 5		
Contact Interval 5 B		

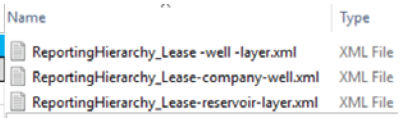


Figure 6–6. Worked example reporting hierarchies.

The structure of a hierarchy data object is shown in **Figure 6**–7. The left side of the figure shows one of the example hierarchies. The right side shows some of the XML hierarchy data object. Note:

- The root node of a hierarchy is the reporting node (red circle).
- Child nodes nest under this root and each other (blue circle).
- The use of reporting entity as the data object reference back to the reporting entity object (for well 01 in this example) using UUID (green box).

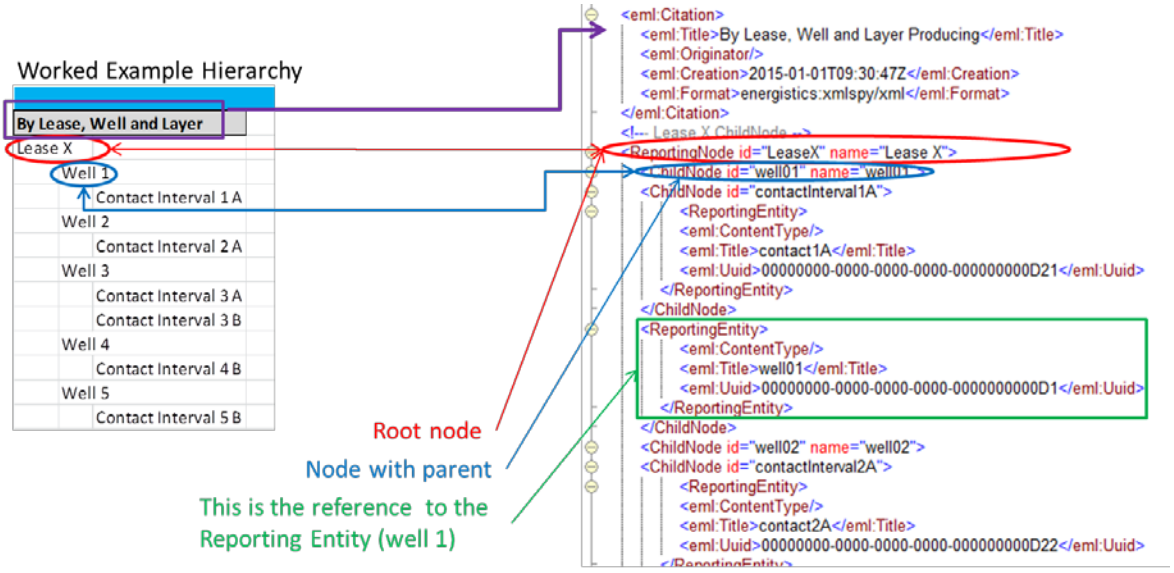


Figure 6–7. Hierarchies showing reference to reporting entities and the root node

6.3 Periodic Volume Reporting using Asset Production Volumes

6.3.1 Worked Example Content

With the assets defined, now you can transfer volume data on a periodic basis. There are many combinations possible. **Figure 6**–8 shows the worked example content illustrating:

- The reporting entities used (left hand column, blue box)
- The types of volume used (column headings across the top, orange box)
- The instances of volume (entries in the matrix where the word shows the qualifier for the volume (see below for details, purple box).

The list of “Examples Shown” below the matrix shows how the figures and explanations below work through the different usages. The colors correspond to the entries in the matrix (row headings, column headings, or cell entries). The explanation below has sub-headings that refer back the numbers in the “Example Shown” list in Figure 6–8.

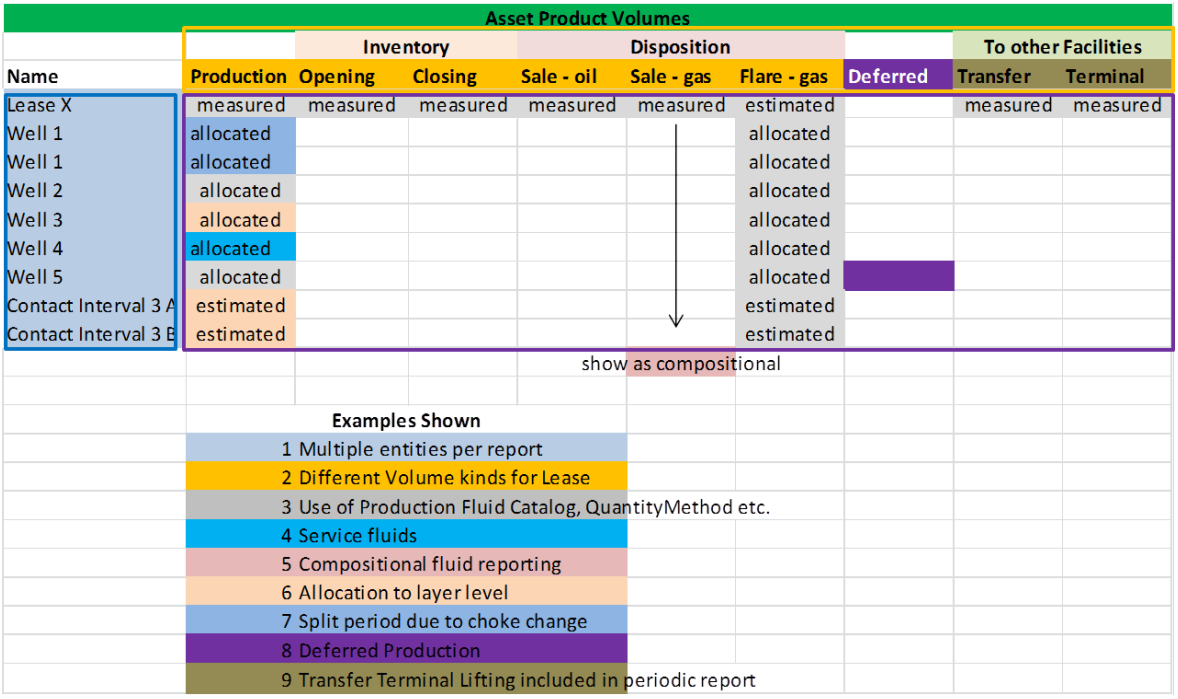


Figure 6–8. Content of asset production volumes example.

The worked example contains one asset production volumes XML data object, found in the worked example root folder (**Figure 6**–9).

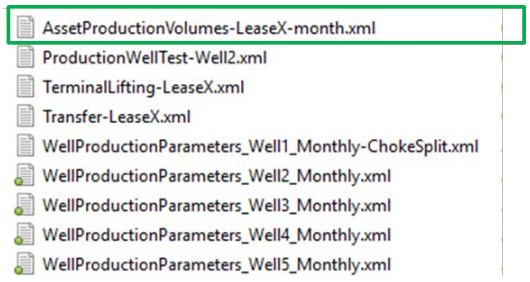


Figure 6–9. Periodic transfer with one asset production volumes XML data object.

So far as possible, the worked example has realistic numbers for the volumes, which are shown in **Figure 6**–10 and can be found in the "Numbers" worksheet of the spreadsheet "Worked Example NAPR.xlsx. **Figure 6**–11 shows how the numbers in the spreadsheet reconciled. **Figure 6**–12 shows the timeline of the worked example. The events on this timeline are represented in the various data objects transferred.

Asset Production Volumes		Production				Opening Inventory	Closing Inventory	Dispositions			Deferred Production	Other Assets	
		Product Fluid			Service fluid			Sale	Sale	Flare		Terminal Lifting	Transfer
Reporting Entity	Reporting Entity Volumes period	Oil	Water	Gas	Methanol	Oil	Oil	Oil	Gas	Gas	Oil	Oil	Oil
Lease X	Month	60,000	985	26,000,000		10,000	20,000	50,000	25,000,000	1,000,000		100,000	50,000
Well 1	Days 1-15	4,800	78.80	2,080,000						80,000			
Well 1	Days 16-31	7,200	118.20	3,120,000						120,000			
Well 2	Month	12,000	197.00	5,200,000						200,000			
Well 3	Month	12,000	197.00	5,200,000						200,000			
Well 4	Month	12,000	197.00	5,200,000	120					200,000			
Well 5	Month	12,000	197.00	5,200,000						200,000	828		
Contact Interval 3 A	Month	6,000	98.50	2,600,000						100,000			
Contact Interval 3 B	Month	6,000	98.50	2,600,000						100,000			
	Key:	measured	allocated	estimated					compositional		analytics model		

Figure 6–10. Volumes reported in the worked example. (For a copy of this spreadsheet, see the file named *Worked Example Simple Product Volume Spreadsheet.xls* included in the example download.)

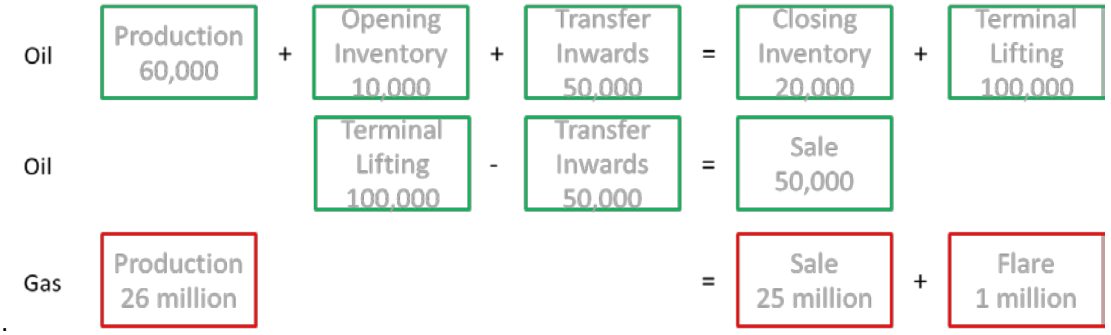


Figure 6–11. Reconciliation of quantities.

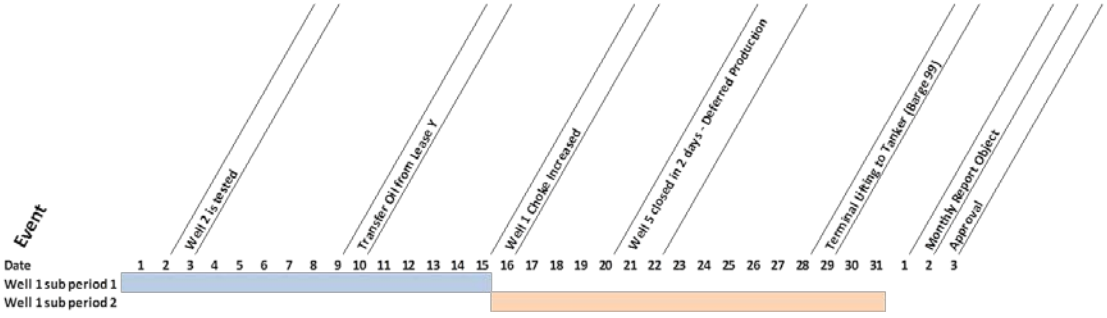


Figure 6–12. Worked example timeline.

6.3.2 Worked Example Volumes Walkthrough

The worked example shows the various kinds of volumes and is explained by the content and order shown in Figure 6–8.

Note that the volumes and flowrates in this report are assumed to be at the standard conditions which are provided one per asset production volumes transfer

#1 Multiple Entities Per Report

The asset production volumes object contains a repeating reporting entity volumes element, which repeats for each reporting entity. **Figure 6**–13 shows the asset production volumes XML data object, with a snippet showing some of the reporting entity volumes elements (in collapsed state, green box). The reporting entity element references the reporting entity (here, for Contact Interval 3B, red box). The standard conditions are transferred only once (and are mandatory) and apply to all volumes in this transfer.

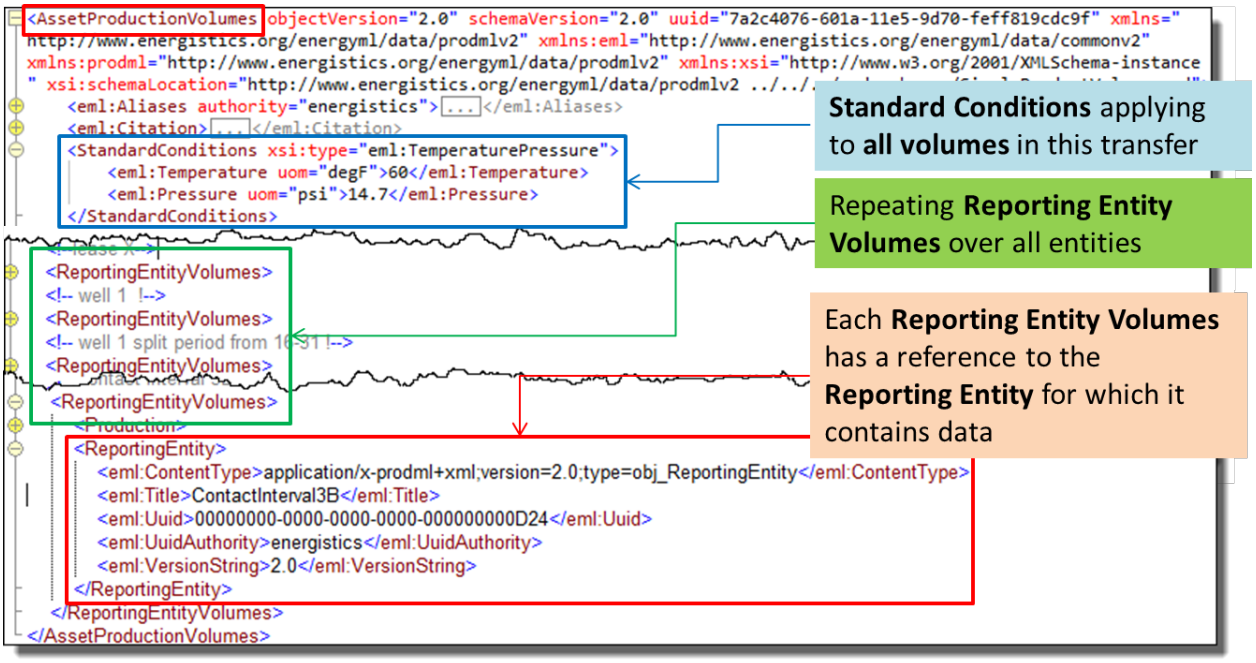


Figure 6–13. Asset production volumes has repeating reporting entity volumes per reporting entity.

#2 Different Volume Kinds Per Lease

If you open any reporting entity volumes element, you can see the different volume types included (per the matrix in Figure 6–8). **Figure 6**–14 shows the volumes for the Lease X parent asset.

- Shown (details collapsed):
  - Disposition: Terminal Lifting
  - Disposition: Transfer
  - Disposition: Product Sale
  - Inventory: Opening
  - Inventory: Closing
- Shown (details expanded):
  - Production
- Not shown but available:
  - Injection
  - Deferred Production

```
<ReportingEntityVolumes>
  <!-- TerminalLiftingDisposition 100,000 barrels !-->
  <Disposition xsi:type="TerminalLiftingDisposition">
  <!-- TransferDisposition 50,000 barrels !-->
  <Disposition xsi:type="TransferDisposition">
  <!-- sold oil 50,000 barrels !-->
  <Disposition xsi:type="ProductDisposition">
  <!-- sold gas 25,000 cubic feet !-->
  <Disposition xsi:type="ProductDisposition">
  <!-- flared gas !-->
  <Disposition xsi:type="ProductDisposition">
  <!-- ClosingInventory 20,000 !-->
  <ClosingInventory xsi:type="ProductFluid">
  <Production>
    <QuantityMethod>measured</QuantityMethod>
    <!-- production oil 60,000 !-->
    <ProductionQuantity productFluid="00000000-0000-0000-0003-00000000091" xsi:type="ProductFluid">
      <Volume uom="1000 bbl">60</Volume>
      <ProductFluidKind>oil - gross</ProductFluidKind>
    </ProductionQuantity>
    <ProductionQuantity productFluid="00000000-0000-0000-0003-00000000092" xsi:type="ProductFluid ">
      <Volume uom="1E6 ft3">26</Volume>
      <ProductFluidKind>gas - dry</ProductFluidKind>
    </ProductionQuantity>
    <ProductionQuantity productFluid="00000000-0000-0000-0003-00000000093" xsi:type="ProductFluid">
      <Volume uom="bbl">985</Volume>
      <ProductFluidKind>water - processed</ProductFluidKind>
    </ProductionQuantity>
  </Production>
  <ReportingEntity>
  <!-- OpeningInventory 10,0000 !-->
  <OpeningInventory xsi:type="ProductFluid">
</ReportingEntityVolumes>
```

Figure 6–14. Kinds of quantity contained within each reporting entity volumes.

#3 Use of Production Fluid Catalog and Quantity Method

Where the volumes are expanded in **Figure 6**–14, you can see the reference to the corresponding fluid component in the fluid component catalog. Remember, each transfer requires only one fluid component catalog. It lists and characterizes all components needed. These could be any of the following, as examples:

- One each of oil, gas, water, in a simple black oil system.
- Multiple oils, etc., with different qualities such as gravities.
- Compositional with pure, pseudo and plus fractions.

Figure 6–15 shows this in more detail, with the fluid component catalog for the asset product volumes example (left side of the figure, red box). The figure also shows three black oil type fluid components and below, some pure fluid components and plus fluid components.

The right side of the figure shows the production quantity element within (in this case) a production volume element, with the water used as an example as a reference (blue box). Note that because this referencing is within one XML data object, a UID rather than a UUID is used (the UID only needs be unique within the data object).

Figure 6–15. Fluid components transferred once per asset production volumes object and then referenced.

Figure 6–16 shows more details of the volume reporting.

- Every volume has a quantity method, which is an enumeration listing the ways in which that quantity was determined (red box).
- Every volume has a product fluid reference (per above, and blue box), which gives the **physical** properties of the component.
- Every volume also has a product fluid kind, which is a simple enumeration to provide information about the **product** that the production quantity represents (green box).

NOTE: It is not *mandatory* to include the reference to the fluid component *or* to product fluid kind. but obviously *at least one* is needed.

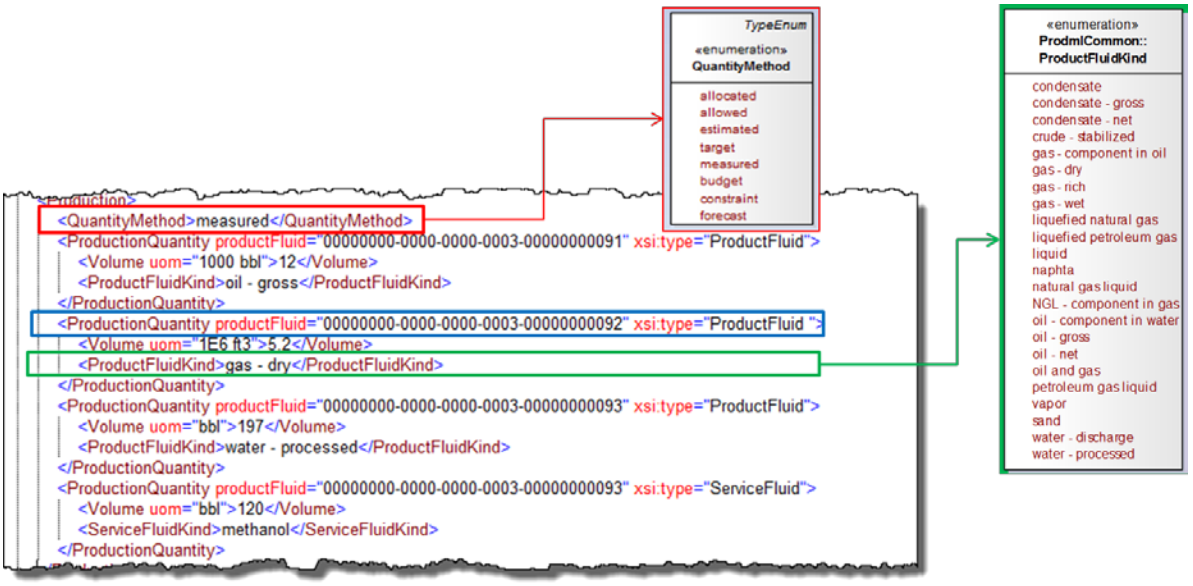


Figure 6–16. Quantity methods and product fluid kind can be included in the volumes.

#4 Service Fluids

In addition to product fluids, service fluids can be reported. **Figure 6**–17 shows an example. Service fluids are reported only by a service fluid kind enumeration. NOT by a reference to the fluid component catalog (a product fluid ref for the product fluid types). In other words, service fluids has no catalog, only the enumerated list shown in Figure 6–17 (purple box).

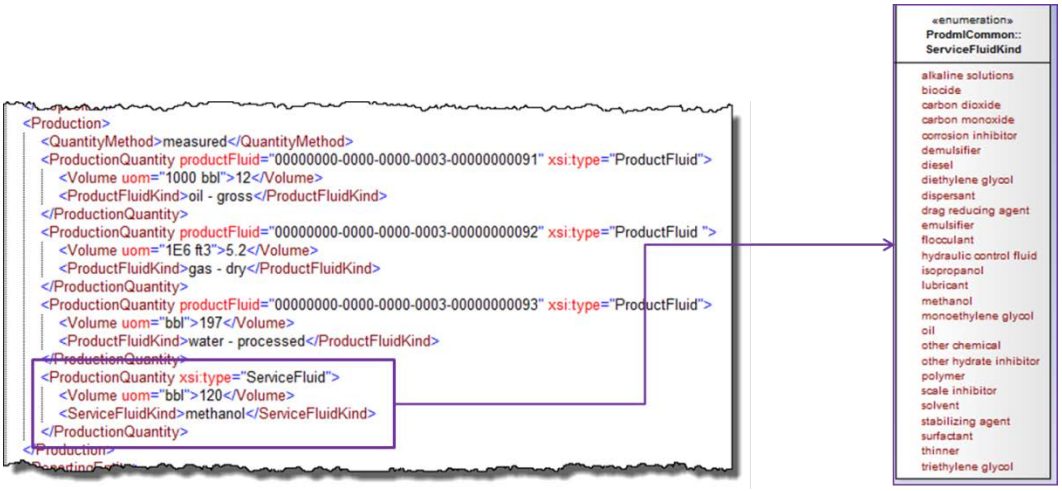


Figure 6–17. Service fluids can be included in the volumes.

#5 Compositional Fluid Reporting

Figure 6–18 shows how compositional reporting of fluid is done. In this example, the fluid component catalog (left side of figure) has in addition to the black oil components, compositional components C1, C2 and C3 (green box) and a C10+ plus component (red box). Pseudo and plus components can contain additional data about them in the catalog (e.g., molecular weight).

The right side shows the use of the overall composition element, which is optional for any volume. It shows the references to the compositional components and their fractions—mass in this example (blue box). Note that the black oil description is also used, showing the volume of type gas (purple arrows).

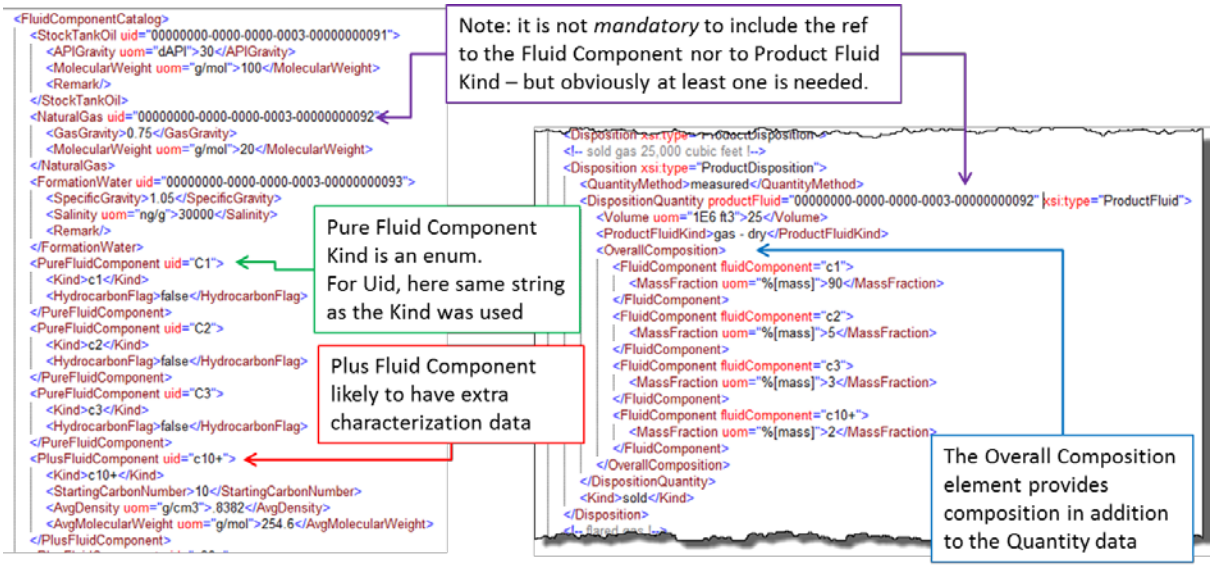


Figure 6–18. Fluids can be reported compositionally.

#6 Allocation to Layer Level

A requirement is to be able to allocate volumes into reservoir layers where required, in commingled systems. This is done in the worked example for well 03 which is completed in two reservoirs, the connections being represented by two wellbore completions. See Figure 6–1.

The pattern of reporting volumes does not change. You only need to include the level of detail that is needed in the reporting entities and hierarchies (as shown), and then to reference these as shown in **Figure 6**–19. Note that by way of illustration, the quantity method for these two entities is “estimated” (being downhole, unless we have metering downhole).

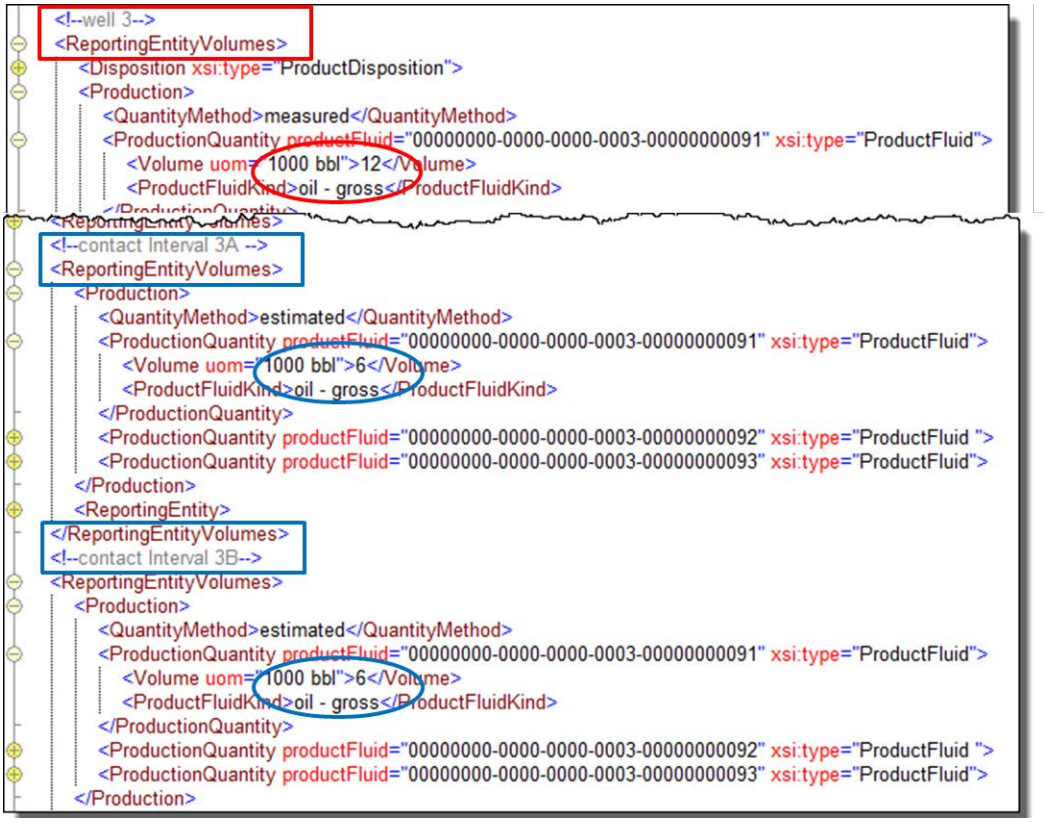


Figure 6–19. Allocation to individual layers.

#7 Split Period Due to Choke Change

A further requirement is to be able to report allocated volumes across sub-periods within the overall reporting period. For example, when you need to report a well being shut in for some of the period, and where its flow would be all allocated to the flowing period. The worked example shows an example of a choke change part way through the period. To see how this is done, see **Figure 6**–20.

Multiple sub-periods can be defined using the optional start time and duration elements within the reporting entity volumes. In this case, there are two reporting entity volumes for the entity concerned (well 01), one for each sub-period. The figure shows the start date, duration, and volumes for this entity for the first period (red boxes) and second period (blue boxes).

To see how the changing choke settings (in this example) are transferred for these same periods, see Section 6.4.2.

Figure 6–20. Splitting quantities allocated across a period.

#8 Deferred Production

Deferred production can be reported using the specific element for this, deferred production event. **Figure 6**–21 shows the worked example. A deferred production event sits under a reporting entity volumes, which associates it with a specific entity. Each event has a start and end time and a duration. The duration is mandatory and the times are optional because you may not always know exactly when a failure occurred.

With the deferred production quantity, an estimation method enumeration can be included, which lists the various ways in which deferred volumes can be calculated.

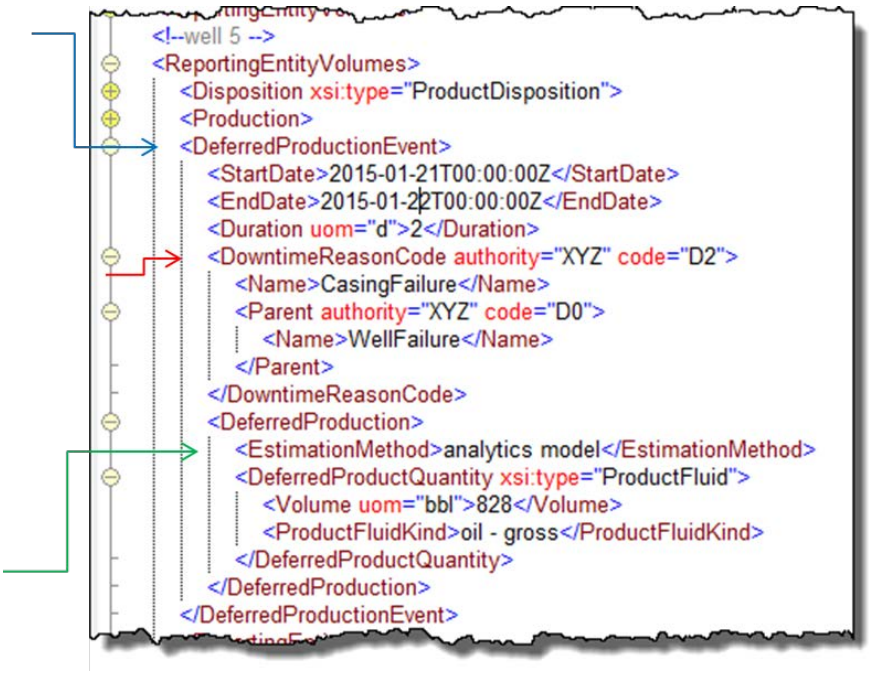


Figure 6–21. Deferred production.

There is a code for downtime reason. Because the codes are company specific, they are not part of the standard. **Figure 6**–22 shows the UML for the downtime reason code. These can be arranged in any desired hierarchy. In the example, two levels of code are shown. Use the authority element used to record which company (or standard) codes are being used.

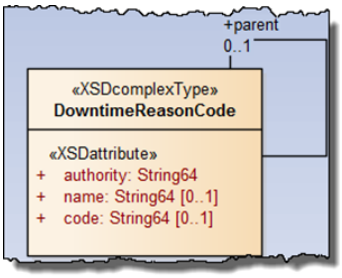


Figure 6–22. Downtime reason can be reported using a hierarchy of codes.

#9 Transfer or Terminal Lifting

A transfer or terminal lifting can happen at any time during the period being reported. There is a choice of using a standalone data object transfer each time this happens, or of incorporating the data within the periodic report (using asset production volumes).

- Section 6.4.3 describes the transfer as a standalone.
- Section 6.4.4 describes the terminal lifting as a standalone.
- Section 6.4.5 describes how these can be incorporated within the asset production volumes data object.

6.4 Event-driven Production Reporting

There are four kinds of event which are expected to give rise to a data transfer that is asynchronous to the periodic volume reporting described in Section 6.3. The worked example has instances of all four. They are:

- production well test
- well production parameters
- transfer
- terminal lifting

These are described in Sections 5.4 and 5.5. Their occurrence in the worked example month is shown in Figure 6–12. **Figure 6**–23 shows the example files for these transfers.

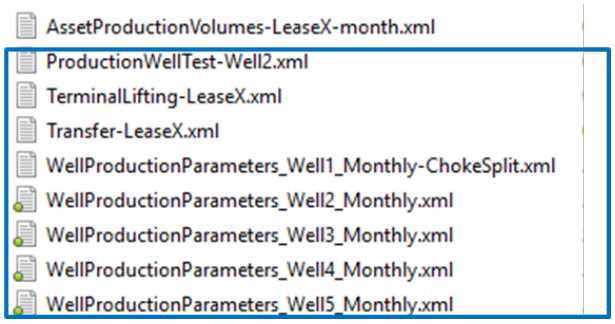


Figure 6–23. Event-driven transfer with example XML data objects in worked example root folder.

6.4.1 Production Well Test

Figure 6–24 shows the worked example production well test. Note these key points:

- Fluid component catalog and the references to fluid components work in the same way as other objects, such as asset production volumes as shown in Section 6.3.2 and Figure 6–15 (purple box).
- Well test method and validation can be included (orange box).
- Well test reports flow rates rather than quantities (green box).
- A range of operating parameters can be included (blue box).
- The reporting entity data object reference is *expected* to be to a well kind of reporting entity, but this match is not enforced (red box).

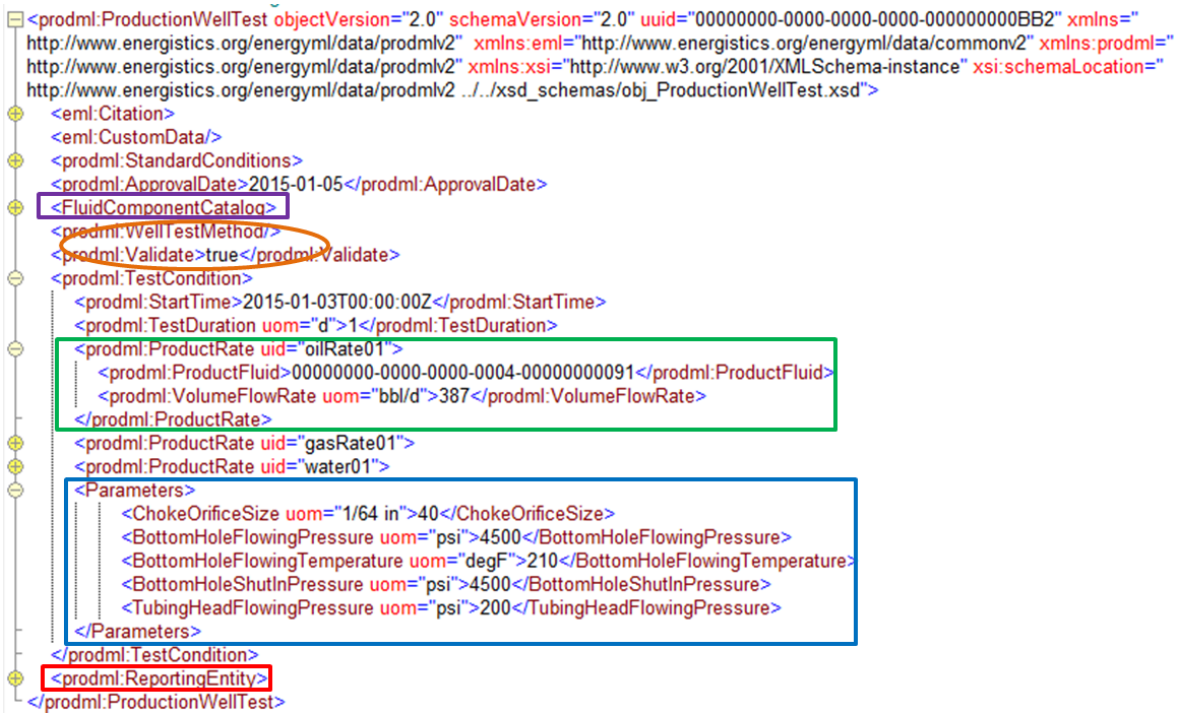


Figure 6–24. Production well test: event-driven transfer.

6.4.2 Well Production Parameters

Figure 6–25 shows the worked example well production parameters. Note these key points:

- Fluid component catalog and the references to fluid components work in the same way as other objects, such as asset production volumes as shown in Section 6.3.2 and Figure 6–15.
- A range of operating parameters can be included (purple arrow).
- The transfer can optionally be split into a number of sub-periods using multiple production well period elements. The example shows the choke change which resulted in the volume reporting for this well, as shown in Section 6.3.2 and Figure 6–20.
- Optionally well production parameters can report flow rates (red arrow).
- For production well test, the reporting entity data object reference is *expected* to be to a well kind of reporting entity with, but this match is not enforced.

Figure 6–25. Well production parameters event-driven or reported across any number of discrete periods.

6.4.3 Transfer

The transfer example in the worked example is shown in **Figure 6**–26. This is a standalone data object with the following highlights:

- Fluid component catalog and the references to fluid components work in the same way as other objects, such as asset production volumes as shown in Section 6.3.2 and Figure 6–15 (red arrows).
- Transfer has a source facility and a destination facility. No tanker is involved; the transfer happens via pipeline (green arrows). These are data object references to the reporting entities for these objects.
- As an alternative to using a standalone transfer data object, transfers can be embedded in the periodic asset production volumes as a special type of disposition; see Section 6.4.5 and Figure 6–28.

Figure 6–26. Transfer: standalone data object in the event-driven mode.

6.4.4 Terminal Lifting

The terminal lifting example in the worked example is shown in

Figure 6–27. This is a standalone data object with the following highlights:

- Fluid component catalog and the references to fluid components work in the same way as other objects, such as asset production volumes as shown in Section 6.3.2 and Figure 6–15 (red arrows).
- Terminal Lifting has a loading terminal and a destination terminal. A tanker is involved in the lifting from the former to the latter (green arrows). These are data object references to the reporting entities for these objects.
- A certificate number can be added for the reference to the document defining the lifting onto the tanker.
- Note that reporting entity kinds are available for “oil tanker” and “tanker truck”, for ship and land (truck) transport.
- As an alternative to using a standalone transfer data object, transfers can be embedded in the periodic asset production volumes as a special type of disposition; see Section 6.4.5 and Figure 6–28.

Figure 6–27. Terminal lifting: standalone data object in event-driven mode.

6.4.5 Transfer and Terminal Lifting as Dispositions within Periodic Asset Production Volumes Reporting

Transfer and terminal lifting data can be included as dispositions within the periodic asset production volumes reporting. This is in addition to their being able to be reported in an event-driven manner as standalone data objects for the reasons explained in Section 5.5. The standalone examples are shown in Sections 6.4.3 and 6.4.4. The embedded example of the same data is shown in Figure 6–28.

Note these key points about this way of reporting transfer and terminal lifting:

- Transfer and terminal lifting are available as specific types of disposition (red boxes). Note: All other types of dispositions than these two are typed “product disposition” with an enumeration to define the kind of disposition. See the example in Figure 6–14.
- The same elements that can be included in the standalone version can also be included in this embedded version. The example shows the elements *tanker* and *transfer direction* for illustration (green boxes).
- The UUID of the standalone data object is not included in the embedded version (because this data object has the UUID of the parent asset production volumes). The example here shows the remark element used to refer to the event-driven transfer by UUID (purple arrow).

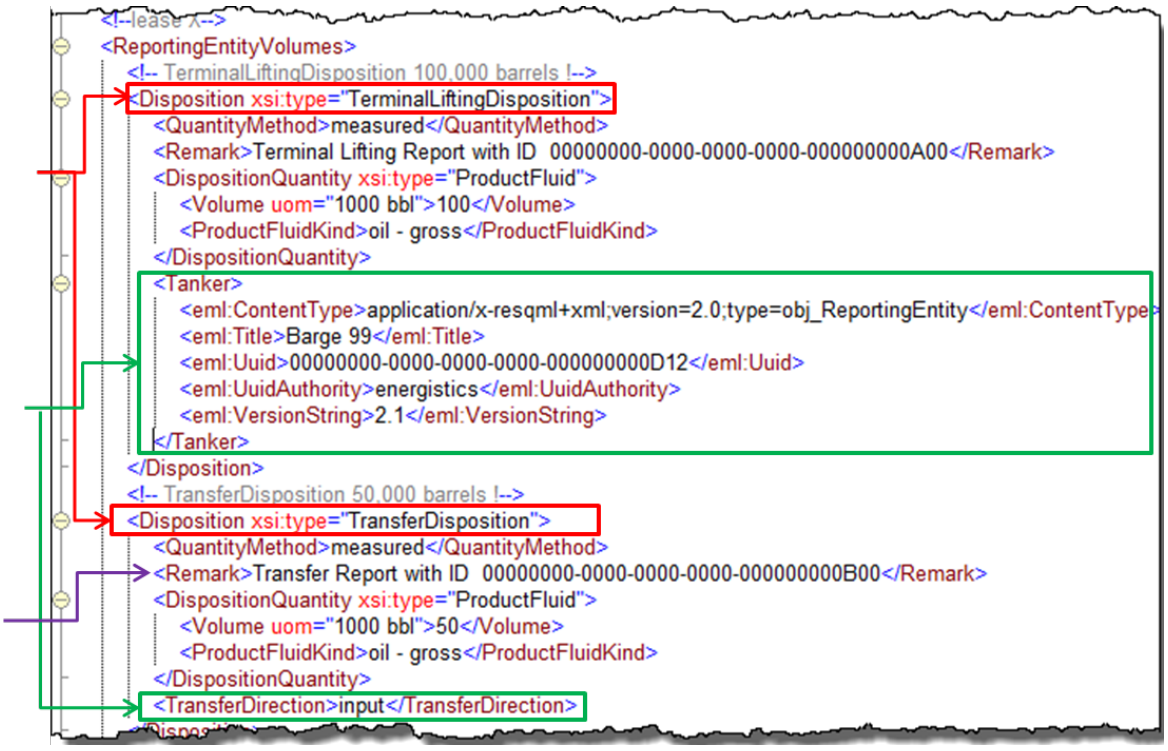


Figure 6–28. Transfer and terminal lifting optional inclusion in asset production volumes period report.

7 Appendix for Simple Product Volume

7.1 Use Cases

This appendix contains the detailed use cases that are listed in Section 4.2.

7.1.1 Use Case 1 & 2: Direct Exchange of Production Data Between Partners

Use Case 1: Receive from a partner

Use Case 2: Transmit to a partner

Use Case Name	Direct Exchange Of Production Data Between Partners
Version	1.0
Author	Terry Kite (Merrick)
Reviewer(s)	Bill Logsdon/Ashraf Wardeh (Oxy) and William Gilmore (Accenture)
Goal	Provide production data for a fixed duration necessary for monitoring, decision-making, forecasting and reporting, and financial record-keeping associated with operated properties to others with working and/or revenue interest in those properties.
Business Requirement	Discharge contractual obligations Assess ongoing participation in properties operated by others (OBO) Evaluate current operations of OBO properties Comply with financial and reporting obligations related to OBO properties Forecast future potential of OBO properties
Business Value	Better operational insight supporting improved decision making and/or requests for operational changes Improved estimate of future production potential (reserve estimates/reporting; marketing deliverability, etc.) Timelier data for booking related financial transactions, governmental/financial reporting, etc. reduce the cost and effort of both providing data and consuming data through a consistent standard (currently each company tends to have their own solutions so each party has to develop a separate solution for each partner in order to load their data
Summary Description	For producing properties that are subject to joint operating agreements, the Operating Partner is responsible for collecting operational information on those properties and determining well-level and facility-level production volumes by product (or flow stream). They are also typically responsible for providing this information to the other parties in the agreement (Non-operating Partners). The Non-operating Partner uses this data to update its own data store(s) that support making operational decisions, estimating reserves, reporting to outside parties and booking related financial transactions. Although typically transmitted on a regular on-going basis, the Non-operating Partner may from time to time request data that encompasses prior periods to verify and/or complete its own data store. Likewise, the operating partner may at times advise non-operating partners of prior period adjustments and re-issue previously transmitted data that has been amended.

Use Case Name	Direct Exchange Of Production Data Between Partners
Actors	Non-operating Partner; Operating Partner; non-working interest partners (e.g., royalty owners)
Triggers	Agreed upon periodic schedule (daily, weekly, monthly, etc.) Out-of-Schedule request by Non-operating partner
Pre-conditions	Non-operating Partner has a revenue and/or working interest in a producing property operated by the Operating Partner Operating Partner has collected and verified operating data and performed necessary calculations to determine gross well-level production data.
Primary or Typical Scenario	The data is transmitted from the operating partner to interested parties (working interest partner, royalty owner, inter-departmental use, etc.) on a regular basis encompassing all new data since the previous report (note: new well or facility may include historical data. For records representing the aggregation of data over a period of time, the duration of each record is typically fixed (full day, partial day, week, month, etc.) in a report, but there can be multiple reports each consisting of different data ranges and record durations.
Alternative Scenarios	Operating Partner amends production related information previously reported to the Non-operating Partner as part of the agreed upon periodic reporting process. The Operating Partner transmits the report with all modified data, including producing period. Non-operating partner wishes to refresh their production history on a property or properties in which they have an interest (to verify the accuracy of their data, to fill gaps in their data store, outages, etc.). They would request that the Operating Partner transmit a report over a specified date range using a specific record duration (day, week, month, etc.). A non-operating partner that has not been tracking operating information on a well may need a one-time issue of all historical data to establish a starting point. It may be necessary to distinguish between data exchanges which are intended to over-write old records or whether the transmission is of new data.
Post-conditions	Non-operating Partner has data necessary to assemble a production history for the property adequate for making operational decisions, estimating reserves, reporting to outside parties and booking related financial transactions.
Business Rules	

Use Case Name	Direct Exchange Of Production Data Between Partners
Data Requirements	Well identifier Production date and period Total produced volume of oil, water and gas Injection volumes (oil, water or gas); allocated and raw Pressures (tubing, casing, etc.) Temperature Sales volumes (oil, gas) Fuel volumes (gas) Flare/Vent volumes (gas) Downtime/Deferred Production Hours (minimum) Volume Deferred Comment (minimum Formatted reason code/description) Start date/time End date/time Operations comments Well tests (Oil/Water/Gas 24 hour rates and test date) Last modified date/time Overwrite/new data flag Extended Requirements: Well Type Well Status Artificial Lift History Well Maintenance History Facility measurement data Measurement point identifier (NB facility identification may not be a standard) Measurement point type (tank, meter, etc.) Inventory (tank volumes) Volume flowed (meters) Fluid characteristics (temperature, pressure, density, etc.) Pressures/temperatures (vessels/equipment)
Notes	
Definitions	

7.1.2 Use Case 3: Provide Historical Data Pre-Sale ("Divestiture")

Use Case Name	Data Room Pre Sale
Version	1.0
Author	Bill Logsdon/Ashraf Wardeh (Oxy)
Reviewer(s)	Joe Palatka (BP) Shaji John (Halliburton)
Goal	Reduce cost and effort and support automation of data room presale activities and generate added value for properties being sold, as well as for those purchasing properties
Business Requirement	General requirements Provide required data room production information in an easy to access format Support standard data exchange interfaces. Provide consistent, complete information needed to make purchase decisions
Business Value	Consistent format and contents of production information reduces amount of custom programming required for both buyer and seller Consistent format and contents of production information ensures a consistency, accuracy and completeness of data sets avoiding costly rework and potential delays. Making production information available virtually, ensures access to hard copy data by the divesting operator. Ability to use pre-existing solutions based on PRODML standards reduces time and cost for data room functions (both seller and prospective buyers) Ability to support common data exchange formats and interfaces – PRODML, oData, web services, ETL, will provide flexibility and efficiency in data access for potential buyers. Providing additional information allows prospective buyers to determine maximum value of properties allowing higher potential bids. Publish the meta information format.
Summary Description	During the data room pre-sale process, the divesting partner provides information on assets being offered for sale. Prospective acquiring parties analyze the available information and determine whether to make a bid on the properties and how much to bid. Typically multiple prospective acquiring parties will come into the data room to view data and documents and to ask questions. The number and length of visits varies depending on the size and complexity of the sale. Additional discussions and meetings outside the scope of the formal data room are common and may occur after the data room phase is completed. The end result of this process is for the prospective acquiring parties to submit their bids so the divesting party can determine whether to sell the property and who the acquiring party will be.

Use Case Name	Data Room Pre Sale
Actors	Divesting Party – organization selling properties Prospective Acquiring Parties – organizations considering bidding on a property
Triggers	Initiated when divesting party decides to offer properties for sale and to host a data room Initial event and update cycles determined by divesting party based on the length of the data room phase of the sales process Multiple data room events can be scheduled if the mix of assets being sold changes or other events require
Pre-conditions	Driven by divesting party decisions for data room phase of sales process
Primary or Typical Scenario	Prospective acquiring parties may be invited by divesting party or the data room may be open to the public. Typically a dedicated facility is made available containing documents, computers, and telecommunications hookups for prospective acquiring parties. Resources are made available to answer questions and required data is made available for all prospective acquiring parties. The Data room phase of the sale process will have a specific duration and prospective acquiring parties may make multiple visits.
Alternative Scenarios	Physical data room may be eliminated or minimized in exchange for virtual meetings and external links for data on properties being sold. In this case prospective acquiring parties will download data and analyze it locally. Questions will be handled via teleconferences.
Post-conditions	During the due diligence phase of the sales process (after the acquiring party is selected and the bid is finalized), the acquiring party may download current / updated information and may request information about the source of data or may request additional information. Both of these activities are outside the scope of this use case for PRODML.
Business Rules	

Use Case Name	Data Room Pre Sale
Data Requirements	Specific data requirements relevant for PRODML include: Monthly allocated production and injection volumes by product by well completion Monthly end inventory by product and facility Well status history by wellbore Producing method history Well test data by well completion Downtime and Deferred Production by well completion Gas composition and energy content (BTU) data if available Daily meter volume by product if available (optional) Hydrocarbon qualitative information, if available. Scope of allocated volumes includes the entire history of the properties. Scope for other data is based on availability of the data. Additional data requirements that are outside the scope of PRODML include Well master data (lat long, field, reservoir, operator, completion date) Facility data Maintenance history Cost information
Notes	
Definitions	Well completion – the zone / completion of the well where production or injection is occurring (typically assigned an API_NO14 is US properties).

7.1.3 Use Case 4: Obtain Historical Production Data Post-Sale ("Acquisition")

Use Case Name	Acquisition Post Sale
Version	1.0
Author	Bill Logsdon/Ashraf Wardeh (Oxy)
Reviewer(s)	
Goal	Reduce cost and effort of post-acquisition data loads and improve completeness and accuracy of loaded data.

Use Case Name	Acquisition Post Sale
Business Requirement	General requirements Provide needed production data in easy to access format Provide consistent, complete information needed to operate acquired properties Specific data requirements relevant for PRODML include: Monthly allocated production and injection volumes by well completion Complete well status, type, and Method Of Production (i.e., artificial lift type) history Well test data by well completion Downtime and lost production by well completion Daily Raw Injection Volumes Daily meter volume if available Gas composition data if available and supported by PRODML Analog data if available (Inferred Production, Beam & ESP settings, etc.) if supported by PRODML Booked Production Volumes (i.e., sold volumes) reporting by Financial Accounting systems. (Noting these are not revised; instead a prior period adjustment would be made in case of error: hence allocated and booked volumes can be different). Scope of allocated volumes includes the entire history of the properties. Scope for other data is based on availability of the data. Additional data requirements that are outside the scope of PRODML include Complete well master data, including wells, well bores, and well completions (lat long, field, reservoir, operator, completion date, date of production, location information, operating lease) Facility data Well and Facility Maintenance history Cost information
Business Value	Consistent format and contents of production information reduces amount of custom programming required for both buyer and seller Ability to use pre-existing solutions based on PRODML standards reduces time and cost for data room functions (both seller and prospective buyers) Fully incorporating PRODML data model into acquisitions process will yield more consistent and complete data, which will drive more operational success for acquiring companies and ultimately better sales prices for divesting companies
Summary Description	Once the acquisition is complete, the next step is for the acquiring company to take over the properties and start operating them. The immediate need is for minimum information to be able to perform closings for financial and production accounting systems. Additional reserves need to be calculated and booked and production and status history need to be loaded for future production accounting functions. The time frame for doing the initial loads is often very short as deadlines for taking over acquired assets tend to be very short. Once the initial data loads need to support accounting closings are completed, the rest of the data is loaded and validated as resources are available. Success in this second phase supports more efficient and successful operation of the acquired property.

Use Case Name	Acquisition Post Sale
Actors	Divesting Party – organization selling properties Prospective Acquiring Parties – organizations considering bidding on a property
Triggers	Initial load is triggers when the sale is closed and the final contracts are signed Secondary load is triggered when the initial load is completed and acquiring company has taken over operation of the properties
Pre-conditions	NA
Primary or Typical Scenario	Acquiring party is typically given all relevant data that can be provided by the divesting party (although quality of data may be inconsistent). The acquiring party determines critical data need to support initial closing process. Once the required data is located, they attempt to load the data and perform their initial accounting closings. If critical data is missing, requests will be made to the divesting party. Complete load may take several months and data gaps found during this phase are much less likely to be addressed by the divesting party.
Alternative Scenarios	Follow up data transfers may be requested when needed data is missing from the initial data load process.
Post-conditions	Validation of the acquired data is performed to determine completeness, consistency, and accuracy. In addition reformatting may be necessary to convert data into format used by acquiring company (PRODML can provide value here by reducing data conversions needed).
Business Rules	
Notes	
Definitions	Well completion – the zone / completion of the well where production or injection is occurring (typically assigned an API_NO14 is US properties).

7.1.4 Use Cases 5 and 6: Transfer/Receive Data from Third-Party Source

Use Case 5: Transmit monthly data to central data exchange

Use Case 6: Receive monthly data from a central data exchange

Standard	North American Production Reporting Standard
Version	0.1

Author	Peter Westwood (EnergySys)
Reviewer(s)	Barry Barksdale (PDS Energy) / Shaji John (Haliburton)
Goal	Ensure that when data is exchanged the sender is able to specify the rules for access to the data, such that the receiver may enforce privacy rules specified by the data owner.
Business Requirement	Maintain privacy over sensitive data at all times, notably after onward transmission. Allow configuration of the rules for data access to be transferred between two systems. The responsibility for ensuring all data is clearly marked with access conditions lies with the data owner.
Business Value	Contractual conditions typically specify who can see sensitive production numbers at various locations. Unauthorized disclosure of this data to a third party is at best embarrassing and at worse litigious.
Summary Description	In several of the use cases two parties agree to exchange information. The Operating Partner typically collects and sends data to the Non-Operating Partner. In some cases the data may be distributed, possibly via a third-party hub. When this occurs, it is important that the data access rights are carried with the data, so that the rules for data security can be asserted by the receiving system.
Use Case Scope	TBS
Primary or Typical Scenario	Real Examples that this may apply to include: Well Production data should indicate which interested parties are permitted to access the numbers for a given well Where a party receives information concerning the production, apportioned by NRI (or other such split), only the personal net allocation and possibly the total, as defined by the contract, should be viewable by a receiving party. … It is expected that many cases of this type of data permissions need to be supported.
Alternative Scenarios	Data where the supplied permission entitles are not found. The receiving system does not support the enforcement of the required permissions.
Post-conditions	

Business Rules	If no permission are set then the assumption is that the data is accessible to anyone If the any of the identifiers within the permission blocks is set, then the receiver is expected to only allow the identified permissions entity. Entities that may be permissioned include… Permissioned roles will be… Permission classes include Read or Read/Write
Data Requirements	
Notes	The data transfer can be considered as having an Envelope and a Content Body. The Envelope of the data, when present, will define the required permissioned entities allowed to view the content. Systems could exchange the defined capabilities for enforcement of access permissions. This would allow the refusal to transfer data where the privacy cannot be enforced. It is required that the data permissions be configured such that the rules for any contract can be described and managed from the schema. An optional permissions entity could be added to any business data transferred. While it is outside the scope of the standard to say how this is used, it must be enough to allow the receiver the ability to apply access controls.
Definitions	Content Body – any business data being transferred. Permitted Entity – Group identify for whom the access is being defined. Identity Map – mapping between the entities used in one or more systems.
Issues	The Parties exchanging data will need a way to identify the Permitted Entities. This means that they will need to have shared identifiers for both the names of items. This will be difficult to manage but is required to allow the secure transfer of permissions between the two parties.

7.2 Reporting Entity Kinds

The behavior of reporting entities according to kind is not enforced; for example, there is nothing in the schema to restrict well tests to being associated only to reporting entities whose kind is “well”.

The Category column in this table is not a schema feature; it is provided here for information and guidance only. Three categories are used for the reporting entity kinds: Asset, Geographic, and Organizational.

Kind	Description	Category
platform	A single platform.	Asset
tank	A single tank.	Asset

Kind	Description	Category
terminal	A physical object that is an industrial facility for the storage of oil and/or petrochemical products and from which these products are usually transported to end users or further storage facilities.	Asset
well	A single well, possibly with many wellbores (side tracks). Optionally, it can reference a WITSML well object.	Asset
wellbore	A single wellbore (side track) within a well. Optionally, it can reference a WITSML wellbore object.	Asset
Contact Interval	Represents the details of a single physical connection between well and reservoir, e.g., the perforation details, depth, and reservoir connected. Meaning: this is the physical nature of a connection from reservoir to wellbore. Optionally, it can reference a WITSML ContactInterval class within the wellboreCompletion object.	Asset
Wellbore Completion	Each wellbore completion represents a flowing connection between wellbore and reservoir. It contains contact Intervals, which reference the physical aspects of these connections detailed in downholeComponents. Optionally, it can reference a WITSML wellboreCompletion object.	Asset
Well Completion	The well completion data object represents a “flow” or “stream” from the well (e.g., from a wellhead port) that is associated with a set of wellbore completions. When there is more than one such wellbore completion, the flows from them commingle in the well (the wellbore completions may be located in multiple wellbores). The well completion represents this commingled flow. Optionally, it can reference a WITSML wellCompletion object.	Asset
flow meter	A single flow meter.	Asset
pipeline	A fluid conductor that consists of pipe, possibly also including pumps, valves, and control devices, intended for conveying liquids, gases, or finely divided solids.	Asset
gas plant	A facility that processes natural gas to achieve the recovery of natural gas liquids and/or removal of contaminants.	Asset
facility	A generic label for a facility that is not described by the other physical reporting entity kinds.	Asset
production processing facility	A single production processing facility.	Asset
FPSO	Floating production, storage and offloading facility.	Asset
oil tanker	An oil tanker, a vessel that could be a barge, or a sea-going ship.	Asset
tanker truck	A truck which carries oil.	Asset
country	A single country.	Geog
county	A single county.	Geog
field	A single field.	Geog
formation	A bed or deposit composed throughout of substantially the same kind of rock.	Geog

Kind	Description	Category
rock-fluid unit feature	A fluid phase plus one or more stratigraphic units. A unit may correspond to a pair of horizons that are not adjacent stratigraphically, e.g., a coarse zonation, and is often used to define the reservoir. Available to match reported production to the rock-fluid feature in RESQML. Optionally, it can reference a RESQML rock-fluid unit feature object.	Geog
state	A single state or province.	Geog
field - part	An area or zone that forms part of a field.	Geog
reservoir	A single reservoir.	Geog
company	A company name that is the name of the operator company.	Org
lease	A single lease.	Org
license	A regulatory agreement that gives the licensees exclusive rights to investigate, explore, and recover petroleum deposits within the geographical area and time period stated in the agreement.	Org
commercial entity	An organizational construct through which a group of organizations or facilities are aggregated as if it were a single organization.	Org

Part III: Fluid and PVT Analysis

Part III contains Chapters 8 through 12, which explain the set of PRODML data objects for fluid and PVT analysis.

Acknowledgement

Special thanks to the following companies and the individual contributors from those companies for their work in designing, developing and delivering the Fluid and PVT Analysis data model: Calsep, Chevron, ExxonMobil, KBC Advanced Technologies, Shell, Schlumberger and Weatherford.

8 Introduction to Fluid and PVT Analysis Data Objects

8.1 Overview of Fluid and PVT Analysis Reporting

PRODML includes a set of XML data objects that can be used to consistently capture and communicate fluid and pressure-volume-temperature (PVT) analysis data covering:

1. Fluid System and "rock-fluid feature" from which a sample was taken;
2. Sample acquisition, including the methods and conditions of the operation;
3. Sample Container in which a sample is transported;
4. Laboratory Analysis and results thereof;
5. Fluid System Characterization and property generation for upstream technical workflows.

The PVT data objects are designed to improve reliability and reduce costs for data exchanges between field personnel, laboratory personnel, subject matter experts, and end users including technical software applications. The standard is also designed to support the evolution of a single "document" for a fluid sample's lifecycle. Other uses include storing these data in a system of record.

These chapters do not specify how to perform these steps in the reservoir fluid's lifecycle; however, the standard has been designed to allow data to be initially incomplete and updated as additional data are developed. Also, final products—such as a fluid property table or an equation-of-state (EoS) model—remain connected to the earlier lifecycle stages of characterization, analysis, and acquisition and the reports and documents created at each stage.

For a quick overview or to be able to make a presentation to colleagues, see the slide set: Worked Example PVT.pptx which is provided in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files.

8.2 Business Case: Why is a PVT Data-Exchange Standard Important?

Data associated with reservoir fluids are diverse, detailed, and important. Good reservoir fluid characterization is critical for most business decisions about the development of oil and gas fields. Reliable reservoir fluid properties obtained during the field's lifecycle are critical factors in planning reservoir, facilities, and well developments to maximize return on capital.

However, fluids data are often based on samples, which are normally collected, analyzed, and modelled early in the reservoir's lifecycle. Because of the diversity and physical nature of reservoir fluids, the data are usually characterized by complex experiments that represent a best approximation of fluid behavior, along the entire production pathway. The quality of the results of these experiments depends on the test sample's handling and preparation, the experimental process used, and the test conditions used. To ensure that adequately representative properties are available for a specific technical workflow, it is vital to understand and accurately communicate this background process for fluid properties.

8.2.1 Data Exchange Challenges

The extended workflow—from sample collection and handling, to lab analysis and interpretation, to communicating results to end users—is a complex, data-intensive process with ample opportunity for errors, both in the actual workflow process (see a list of these challenges in Section 8.2.2) as well as the data exchange required as part of that process. The challenges associated with data exchange include:

- Chain of custody/time lags
- Multiple labs/multiple clients
- Incomplete, misinterpreted, or misunderstood information
- Sample results, while accurate, may need adjustments for use

- Multiple available data stores, vocabularies, and formats
- Multiple consumers of the data

These PRODML PVT data objects have been designed to address these challenges and improve the accuracy and completeness of the information transferred in these workflows.

8.2.2 Process Challenges

Standing (1981) declared "the greatest use of PVT data lies in calculating reserves of oil and gas in place in the reservoir and the effect of field operational methods on the recovery of oil and gas." In other words, PVT data is fundamental to our ability to understand and predict resource size and recovery.

Some of the challenges (per Standing):

- No assurance that samples obtained in one well are representative of the entire reservoir.
  - Gravity effects can create sample differences
  - Samples at the same structural elevation may have differences based on geologic activity over time
- How representative are samples captured in a well?
  - Fluid may be commingled (from multiple reservoir zones)
  - Fluid may be contaminated
  - Where two phases exist in contact in the same zone, flow rates are almost always different
- How complete is the samples collection data?
  - Collection points, times, etc.
- How depleted is the reservoir?
  - Prefer above 80% of original pressure
  - Below 40% will likely lead to inaccuracies
  - Samples should be re-run as reservoir depletes

References

Standing, M.B. 1981. *Volumetric and Phase Behavior of Oil Field Hydrocarbon Systems*, ninth edition. Richardson, Texas: Society of Petroleum Engineers of AIME.

8.2.3 Benefits of Use of PVT Data Objects

Use of the PVT data objects will improve both the accuracy and availability of usable fluid and PVT data within and between upstream companies. Significant improvements are expected in:

- **Simplifying the project management of a fluids analysis program** while increasing efficiencies in the communication between the customer and the laboratory personnel.
- **Simplifying the reporting of detailed laboratory results** by using a common, self-validating XML format. Improving the quality and consistency of fluid property data created for use in technical applications by both systematically capturing the pedigree of fluid data in the workflow all the way back to sample acquisition and the standardization of input data structures.
- **Increasing the quantity of fluid and PVT analysis data in systems of record** by simplifying the workflow used to load field and laboratory measurements. Also, enabling standard PVT data storage and search and allowing the data to be exchanged between databases, fluid characterization applications, and end-user applications.
- **Reducing mistakes in the communication of results** caused by differences in technical understanding of the fluid property data and the consistent handling of details, such as reference conditions and units of measure.
- **Making the field notes more available to laboratory analysts** so that the quality of the experiments and measurements more realistically represent real-life conditions, and making field and

laboratory notes more available to fluid property specialists seeking to characterize individual samples or to produce a system characterization.

As this standard is adopted, these improvements are expected to be realized in both service and software offerings from vendors and in operators' internal workflows. Opportunities for additional impacts have been identified for workflows attempting to automate field surveillance and performance prediction because the use of these fluid data standards will support the consistent expression of fluid properties across multiple and diverse applications and workflows.

It has been difficult to estimate the impact of more accurate and consistent fluid data on the operational bottom line, but the quantification of its impact should include:

- Less risk for estimates of initial rates, final recovery fractions, and development/depletion strategies.
- Better assessments of peak capacities used to size surface operating facilities.
- Improved accuracy of subsurface flow potential and wellbore tubular design.

Other benefits may be realized through one standard by its interaction with other standards (i.e., Metcalfe's Law). By delivering fluid and PVT analysis as useful data objects in PRODML, these can be readily accessed and used by other data objects in both RESQML and WITSML and by the flow network and completion/nodal analysis data objects in PRODML.

8.3 Scope of PVT Data Objects

Figure 8–1 shows a high-level workflow from which a set of functional requirements were developed.

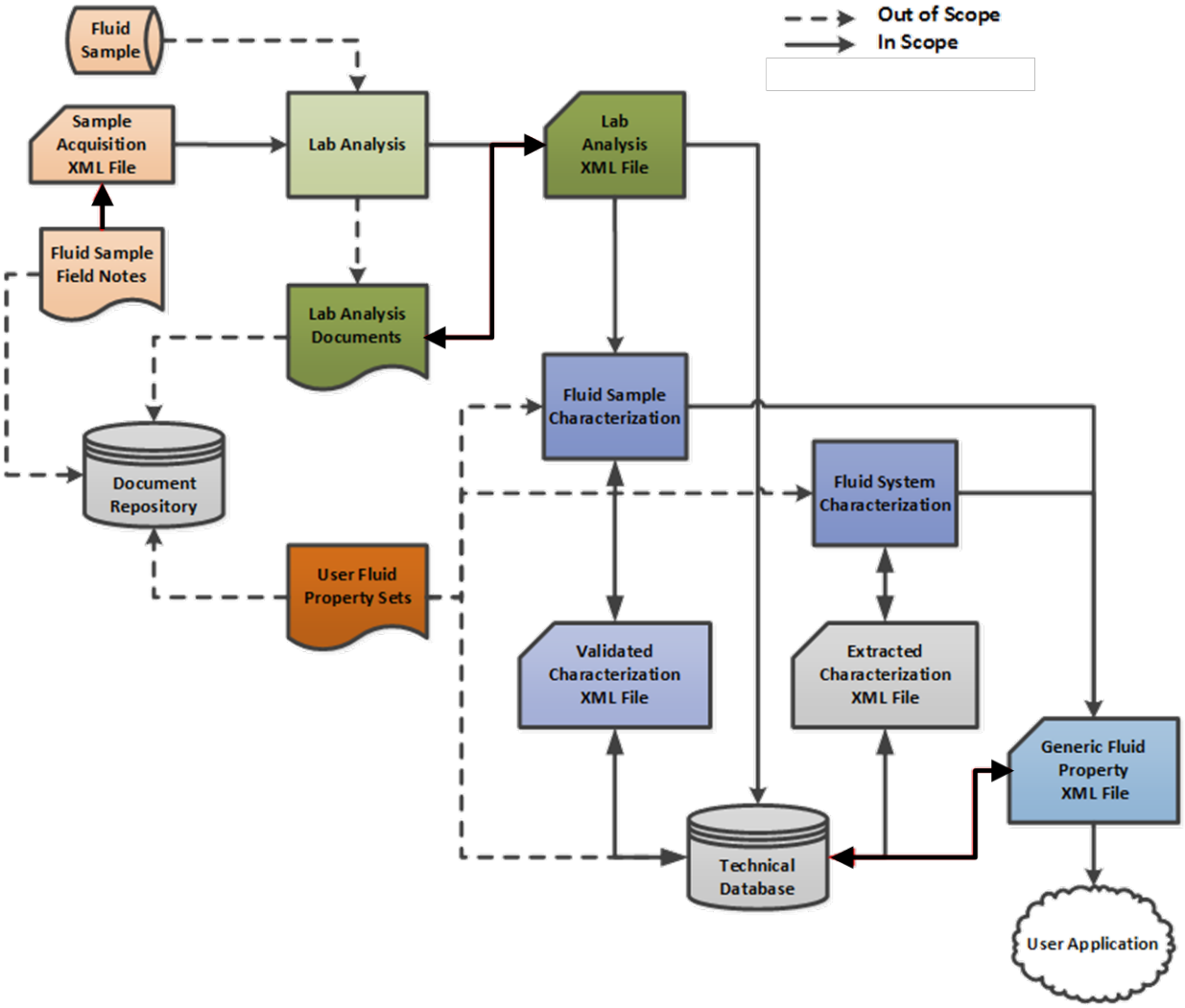


Figure 8–1. High-level fluid sample gathering and analysis workflow, which serves as basis for defining requirements for PRODML PVT data objects.

The areas of interest covered by these data objects include:

- Fluid sample acquisition
- Sample analysis (i.e., laboratory measurements)
- Characterization, including both the validation of laboratory measurements and the generation of calculated results

The solid lines show the data movement processes that have been addressed. The goal of these data objects is to provide complete and accurate XML-based machine readable documents that can be transferred between the different parties and software systems that exist today. The need for this capability extends from capturing fluid samples in the field to supplying data to these consumers:

- Systems of record (e.g., corporate/technical databases, etc.)
- Interpretation software (e.g., proprietary PVT analysis packages)
- End-user applications (e.g., nodal analysis, reservoir simulation, etc.)

Out of scope for this release (dashed lines in Figure 8–1):

- Sample tracking through material movement systems and storage
- Water chemistry and geochemical analysis
- Fluid sampling and analysis for custody transfer
- Project databases

8.4 Fluid and PVT Use Cases

The main use cases are listed and described in Chapter 9.

9 Fluid and PVT Analysis: Use Cases

This chapter provides an overview of the main use cases and explains how the set of PVT data objects support them. Use cases include:

- **Use Case 1: Sample Acquisition** defines the requirements to capture all the relevant information about a fluid sample and its capture, information which may subsequently be needed to understand, quality check, and apply results of laboratory analysis. For details, see Section 9.1.
- **Use Case 2: Laboratory Analysis** provides requirements for the laboratory fluid analysis program and the resulting experiment measurements, which involves many different participants and detailed, accurate communication. For details, see Section 9.2.
- **Use Case 3: Sample Characterization** is the process of building a numerical model of the laboratory measurements from an individual sample and determining the quality of the results received from laboratory fluid analysis; it involves a QC and verification of the samples and results. For details, see Section 9.3.
- **Use Case 4: System Characterization** is a process with a goal of building a numerical fluid model for data-specific operating scenarios and fluid systems and providing a fluid description to be used in engineering models; data from multiple samples may be used for conditions not investigated in the laboratory. For details, see Section 9.4.
- **Use Case 5: Data Storage** addresses the storage, search, and retrieval of information in systems of record for the lifecycle of the fluid sample data (i.e., from acquisition to utilization to disposal). For details, see Section 9.5.

9.1 Use Case 1: Sample Acquisition

The requirements for this use case are to capture all the relevant information about collecting a fluid sample, including everything that may be needed to understand, quality-control, and apply the results of laboratory analysis. **Figure 9**–1 highlights (red box) the sample acquisition portion of the workflow.

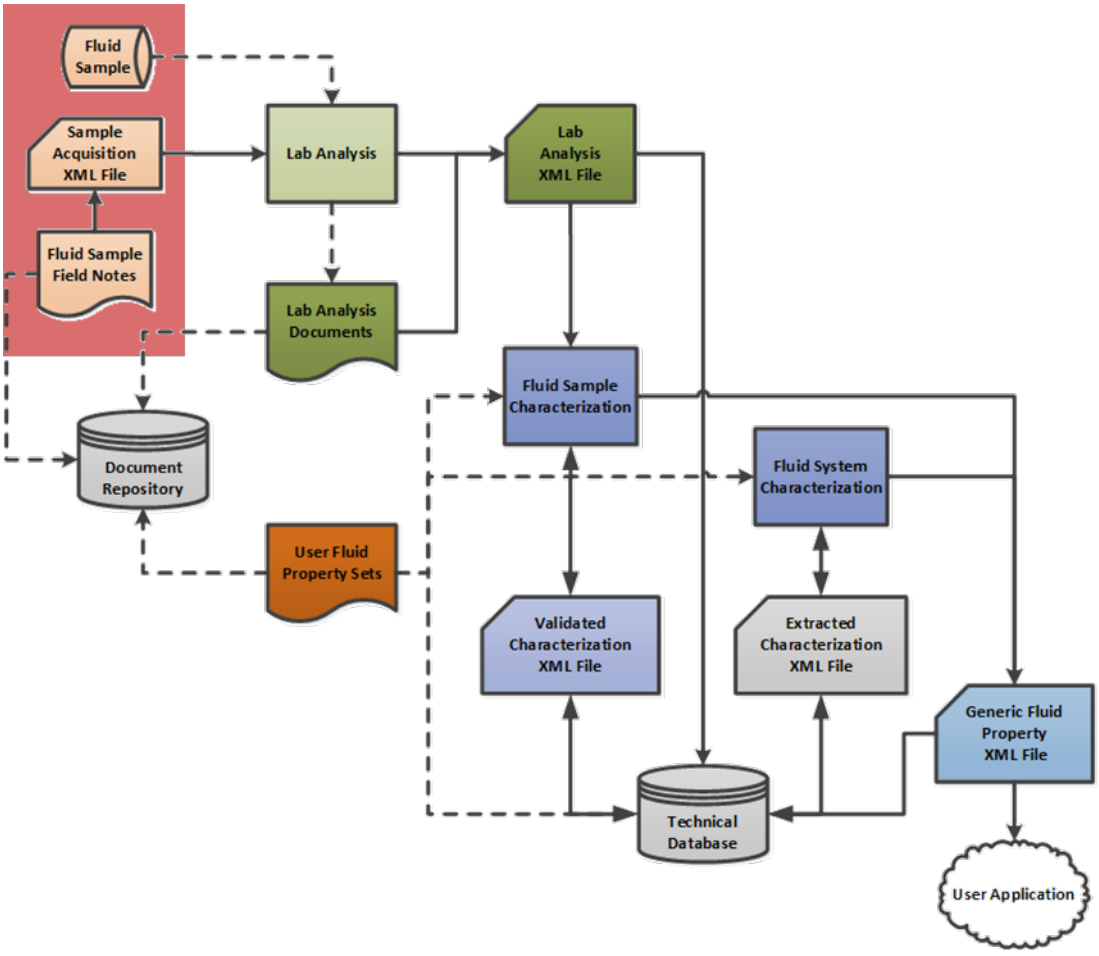


Figure 9–1. Sample acquisition use case.

9.1.1 How the Solution Supports this Use Case

The sample acquisition data object helps to answer questions like these:

- Which reservoir did this sample flow from?
- Which conditions was the sample acquired under? Where? When?
- What method was used to acquire it? Which particular tool was used?
- How much actual and usable sample volumes were collected?
- Were multiple samples taken from same location?
- What are the intended uses of the samples?
- Who handled the sample? What did they do? When?

This information is needed to assess the sample's representativeness, its potential for contamination, and its susceptibility for the removal of trace components (e.g., by adsorption onto surfaces of the sampling tool). This information is also needed when designing the laboratory analyses and selecting the best

samples for characterization. The ability to describe the physical configuration and conditions of the flow stream being sampled—either at the surface or downhole— was recognized as a very important part of this use case. For this reason, the sample acquisition data object contains specific references to other available data objects, such as WITSML's Well, Wellbore and Well Completion; and PRODML's Product Network port, Welltest, and WFT Run, allowing the context of the sampling event to be derived, when available, from their data. Additional context is preserved by enabling field notes and other relevant documents to be referenced from and "bundled" with the sample acquisition data object.

9.2 Use Case 2: Laboratory Analysis

This use case provides requirements for the laboratory fluid analysis program and the resulting experiment measurements, which involves many different participants and detailed, accurate communication.

Figure 9–2 shows the components used to define the laboratory analysis program and its measurements. Most of the information for this use case is described using the fluid analysis data object.

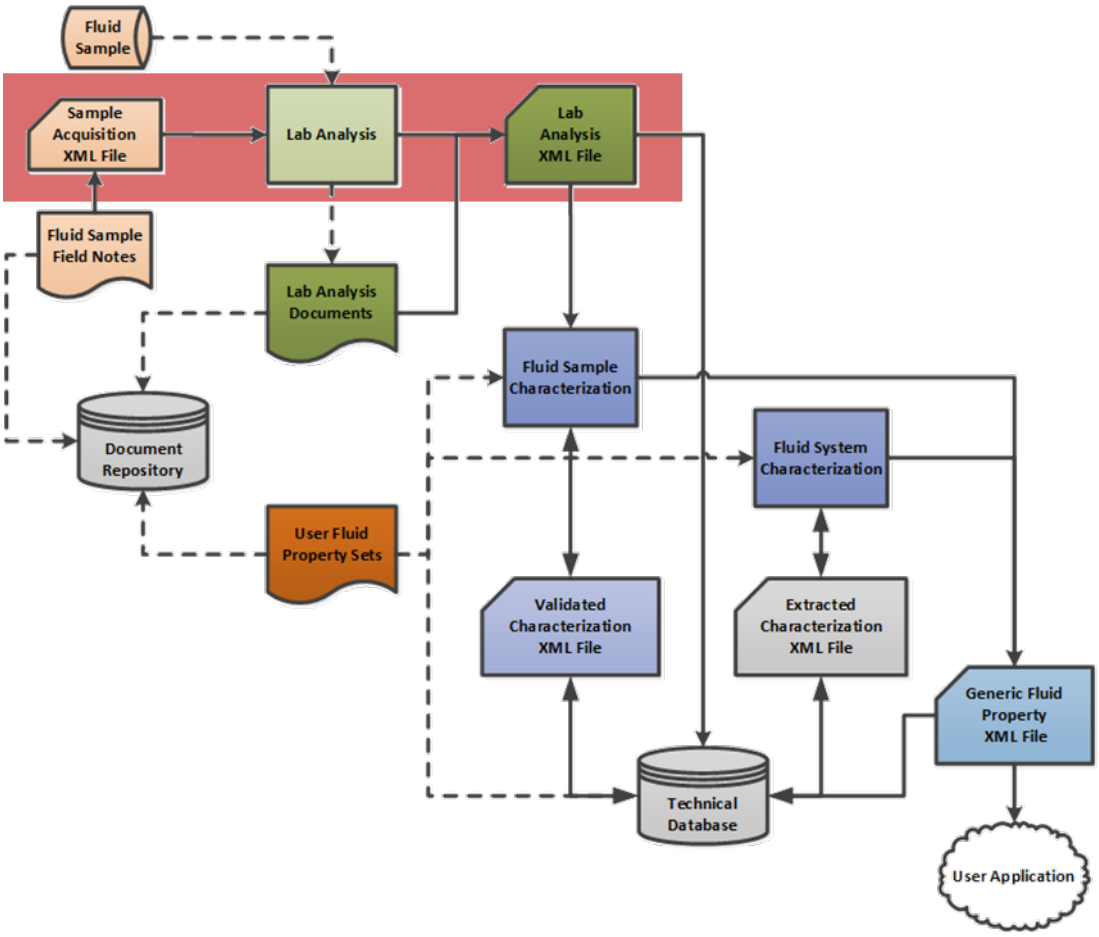


Figure 9–2. Laboratory analysis use case.

After the fluid samples arrive at the laboratory and based on the business needs identified by the customer and the information gathered during sample acquisition, a matrix of samples and laboratory analyses is usually created. For each sample employed, the chain of custody is updated to detail how much of which sample was used. Executing this plan, these tests are conducted and the resulting measurements are reported. **Figure 10**–6 shows the types of laboratory analysis tests that are supported. For a description of these tests, see Section 10.5.1.

Laboratory reports may be included as documents within the EPC (Energistics Packaging Convention) package

9.3 Use Case 3: Fluid Sample Characterization

Figure 9–3 shows the next step in the fluid and PVT analysis life cycle, sample characterization. The goal of this step is to develop a mathematical model that can faithfully reproduce the observed laboratory measurements for a single fluid sample. This model may then be used to calculate the properties of the fluid system (i.e., system characterization, the next use case) for historical field conditions or for various depletion strategies (such as EOR), which is the process of building a numerical model of the laboratory measurements from an individual sample and determining the quality of the results received from laboratory fluid analysis. This use case involves a QC and verification of the samples and results.

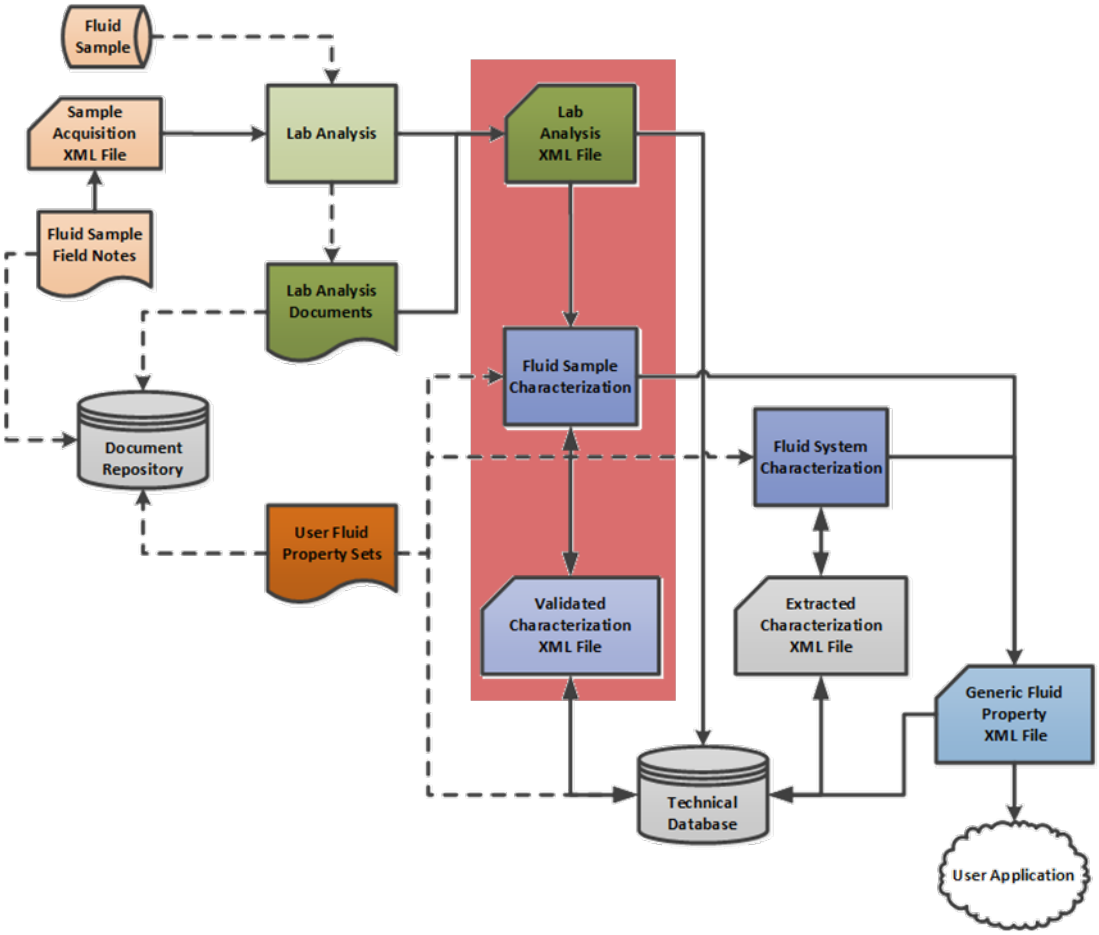


Figure 9–3. Sample characterization use case.

After the laboratory analysis is received, the usual first action is to complete a consistency and correctness check of the data. If the information coming from the laboratory is found to be non-coherent or not self-consistent, the laboratory may be requested to repeat some or all the experiments, if possible. In other cases, to increase the confidence in the results, some or all of the experiments may be repeated, either by the original laboratory or by another laboratory. Regardless of validity, all measurements on fluid samples may be potential candidates for characterization and may be described by the fluid analysis data object. For example, a sample that may have too much gas may form a basis for a future sensitivity analysis.

In the past, samples were characterized using traditional methods, such as direct input of the lab data superimposed with correlations/smoothing functions and possibly coupled with K-value charts or correlations. To convey this legacy data, the tabular format (see Section 10.6.2) could be used. Modern reports, however, are overwhelmingly analyzed using tools with built-in EoS engines for which the parametric format is appropriate. Either or both of these formats may be used as needed for a specific set

of results. The final goal of the PVT workflow, regardless of the points of origin, is to use it in the tools that help in engineering decisions.

Conducting the sample characterization soon after the PVT test makes it easier to repeat or add some of experiments if needed. However, in the current business environment or depending on the in-company processes, staff turnover may be an impediment to the technical level continuity issues, depending on how long it takes to complete the full cycle analysis. Many times, it is difficult to locate the correct data unless it has been properly categorized, indexed and stored. The fluid analysis data object is designed to carry to system identifiers throughout its lifecycle to make this task easier and more consistent.

9.4 Use Case 4: Fluid System Characterization

Fluid System characterization (**Figure 9**–4) is a process with a goal of building a numerical fluid model for data-specific operating scenarios and fluid systems and providing a fluid description to be used in engineering models; data from multiple samples may be used for conditions not investigated in the laboratory.

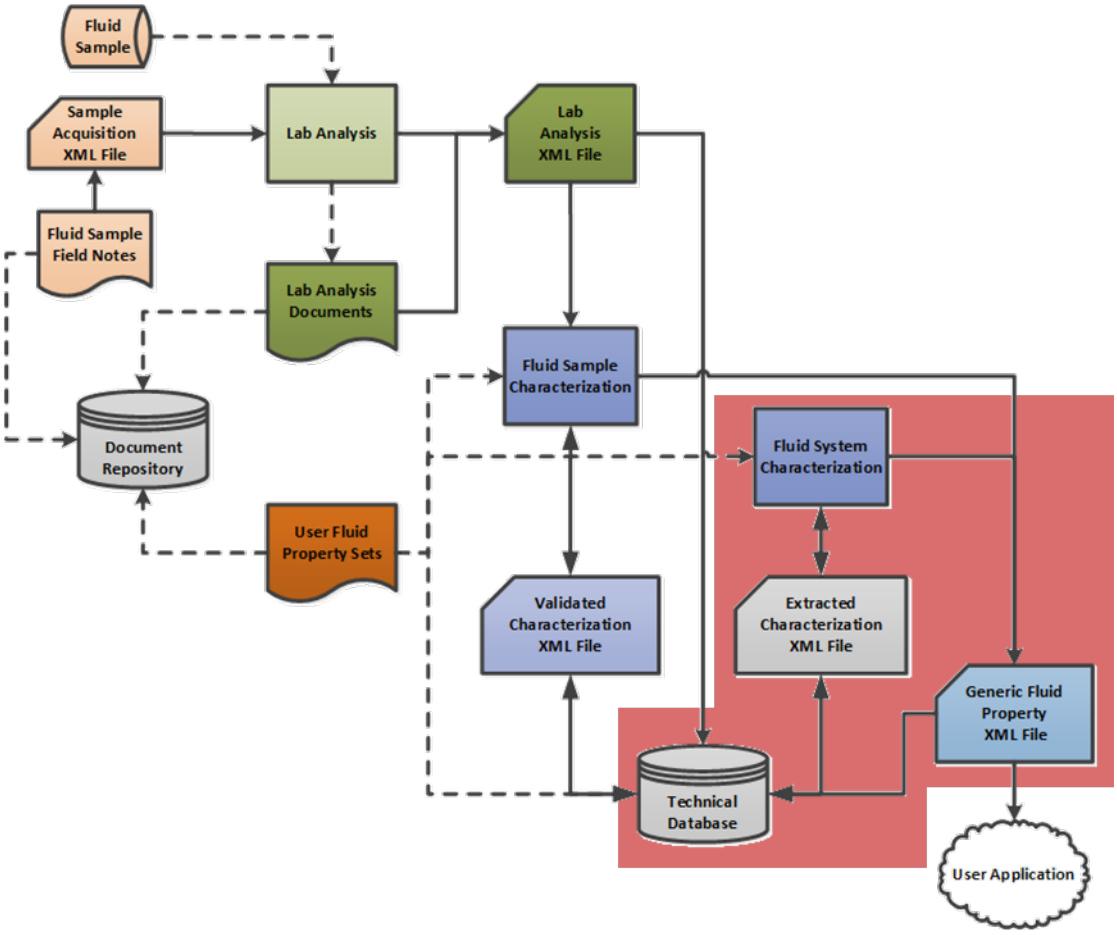


Figure 9–4. System characterization use case.

Many disciplines and their respective software tools need a fluid system description that adequately represents how the fluid is going to behave at different pressure and temperatures; for example: reservoir simulators, lift packages, process simulators, and others targeting various disciplines such as reservoir engineering, production and well engineering, facilities design, and operations. In the context of the current effort, fluid system characterization refers to developing and characterizing fluid components within the fluid system for the purpose of simultaneously evaluating the validity and applicability of earlier

fluid sample analyses and developing a mathematical model to represent holistic behavior of the reservoir fluid system.

In the fluid characterization data object, references may be made to the fluid components defined for the fluid system. This includes the use of specific oil, water and gas components, standard components used in sample compositional analysis, or new pseudo-components intended for use in reservoir or process simulators. The additional properties and characteristics (such as molecular weight, boiling point, etc.) for each fluid component used are added as data in the fluid characterization data object.

9.5 Use Case 5: Data Storage

Data storage (**Figure 9**–5**)** addresses the storage, search, and retrieval of information in systems of record for the lifecycle of the fluid sample data (i.e., from acquisition to utilization to disposal).

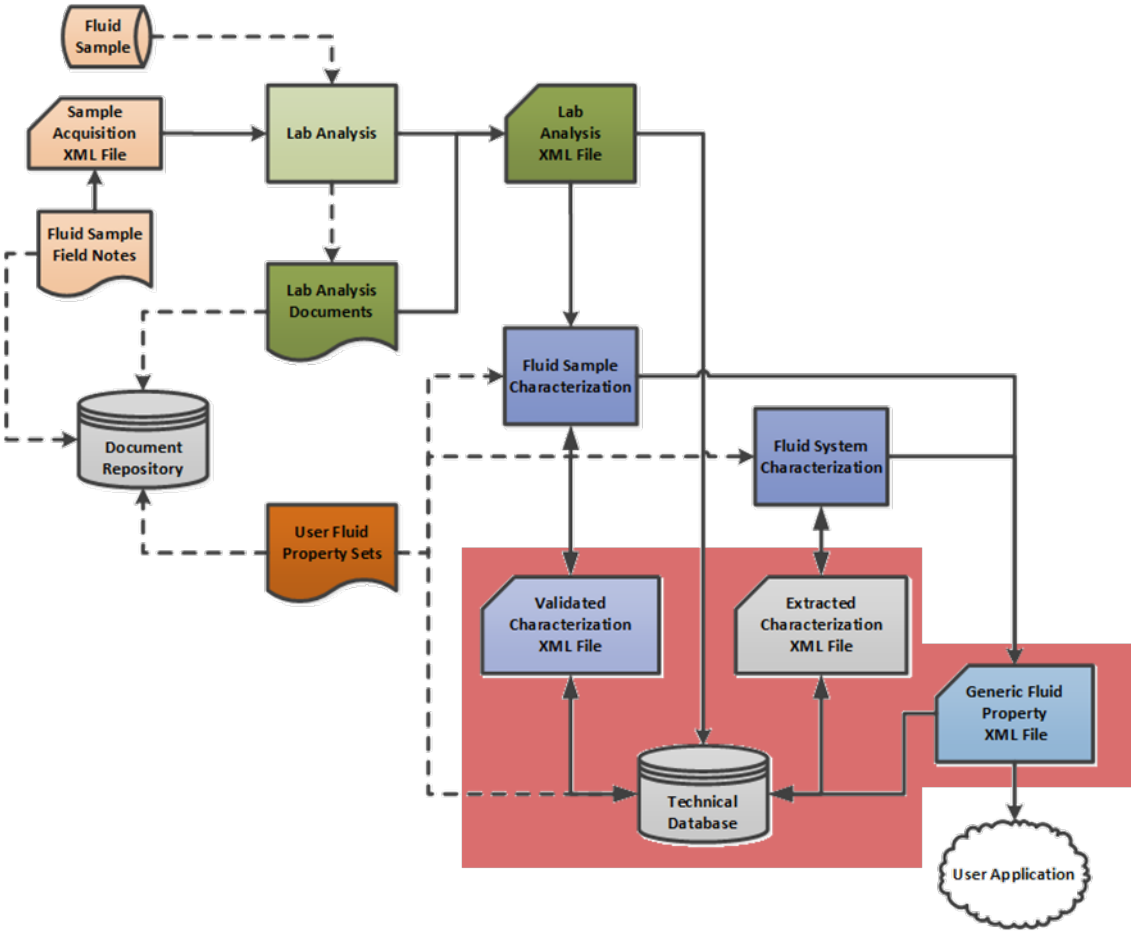


Figure 9–5. Data storage use case.

This use case is specifically intended to support the requirements from laboratories and operators for loading, searching, and exporting fluid analysis and characterization datasets from systems of records, such as in corporate databases.

The two general requirements for the data storage use case are:

- Accurate capture and organization of data to meet the needs of future workflows
- Preservation of context for data items that are used in multiple workflows.

These requirements significantly influenced the structure and behavior of the data objects that make up this standard—such as the separation of the fluid sample from distinct sample acquisition and laboratory

analysis objects—so that implementers of the standard are not burdened with the need to “make up” data not otherwise available. For example, the distinction between fluid analysis data and fluid characterization data more clearly defines the difference between measured and calculated data while better supporting the description of user-defined data because it can be handled without having to fake “laboratory tests”.

Another product of these requirements was the assignment of identifiers within each data object and allowing all data objects to reference other data objects by these identifiers. For example, this enables the data in the sample acquisition data object to remain connected to the fluid samples it produces and vice versa. These schemas are also designed to incorporate the operator’s system of record’s identification (e.g., completion, well, or facility identifiers) within the initially created data objects, which are then carried forward through the workflow, making it easier to load them into a system of record.

Because of the expanded scope of these specifications over earlier ones (Aydelotte et al. 2003), the scope of the corporate repositories needing to store these datasets may need to be expanded. Specifically, additional tables for sample acquisition, laboratory analysis tests, and fluid characterization may be needed unless these data are de-normalized into existing tables, which could potentially limit the system’s ability to export the original data. Also, a capability to managing the document attachments (field notes and laboratory reports, etc.) may be required. Definition of the scope or details of these potential changes are beyond the scope of any transfer standard and are the province of the owner of the relevant databases

10 Fluid and PVT Analysis: Data Model

This chapter describes the PVT data model, which is composed of the key data objects shown in **Figure 10**–1 and explained in the sections below.

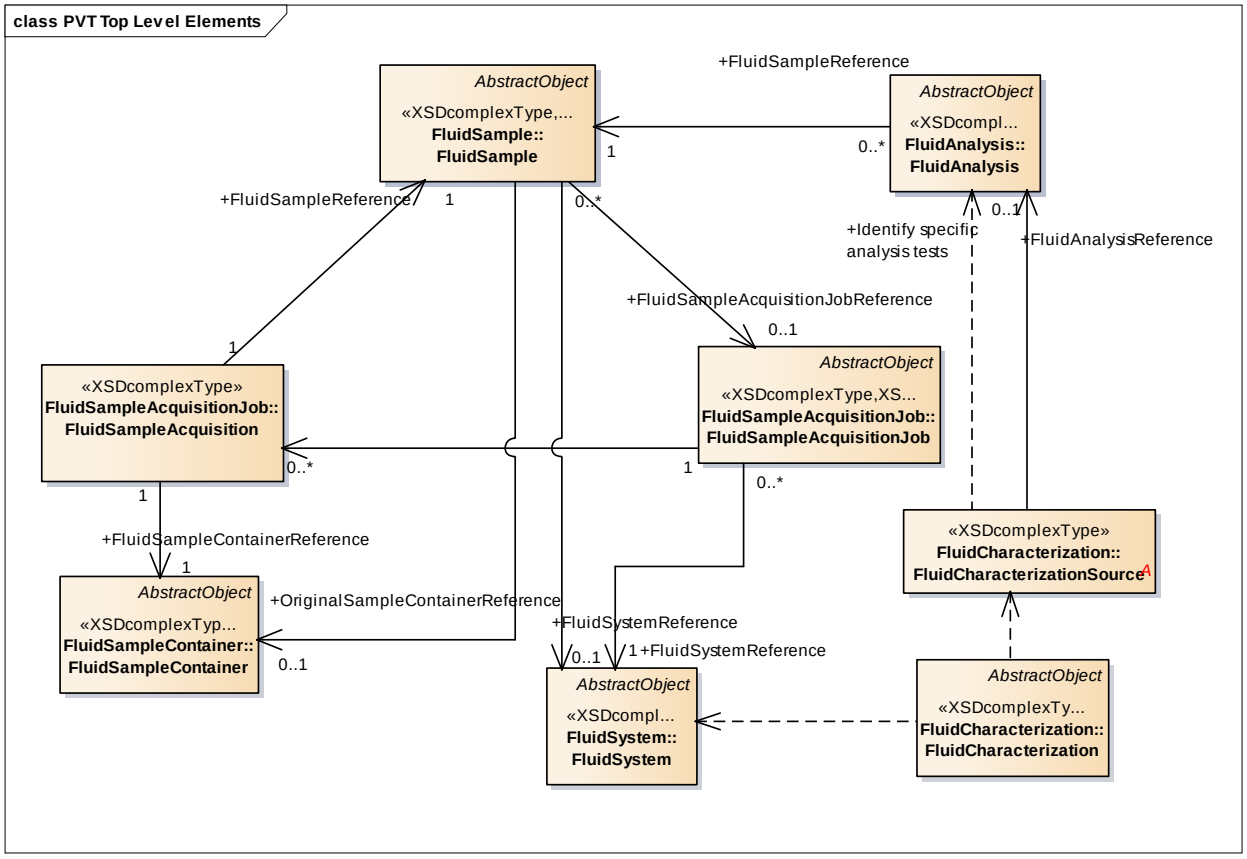


Figure 10–1. Key data objects of PRODML PVT capabilities.

Note that common to several objects, wherever fluid composition is required, is a system for describing the fluid composition in terms of a catalog of fluid components and their properties; this is described in Section 10.7.

The Standard Conditions element is used in many of the PVT objects. A choice is available – either to supply the temperature and pressure for all the volumes which follow, or to choose from a list of standards organizations' reference conditions. Note that the enum list of standard conditions is extensible, allowing for local measurement condition standards to be used. See **Figure 2**–5 which shows the Abstract Temperature Pressure class: this is the type for standard conditions in all PVT objects.

Use is also made within PVT of the Pressure Value type, allowing absolute or relative pressures. See Section 2.3 and **Figure 2**–4.

Chapter 11 provides an explanation of the examples (files) included with the PVT data object download. The objective is this: An understanding of the basic data model described in this chapter combined with the example files and related explanations provide good working knowledge of how PVT works and can be used.

10.1 Fluid System Object

The fluid system data object was created to designate each distinct subsurface accumulation of economically significant fluids. This data object primarily serves to identify the source of one or more fluid

samples and provides a connection to the geologic environment that contains it. Characteristics of the fluid system include the type of system (e.g., black oil, dry gas, etc.), the fluid phases present, and its lifecycle status (e.g., undeveloped, producing, etc.).

Within the fluid system, the products that make up the fluid system can be described. Hydrocarbon components for crude oil and natural gas with the GOR, and formation water, can be defined. For more detailed analysis and characterization, any number of fluid components may be defined in other data objects.

The fluid system can also contain references to multiple rock fluid unit features, which are RESQML objects describing the concept of a combination of a fluid kind in a reservoir zone. Samples can subsequently be associated with a specific rock fluid unit feature from which they are taken.

The main elements of the fluid system object are shown in **Figure 10**–2. The fluid system is referenced by several of the other objects, as shown in Figure 10–1.

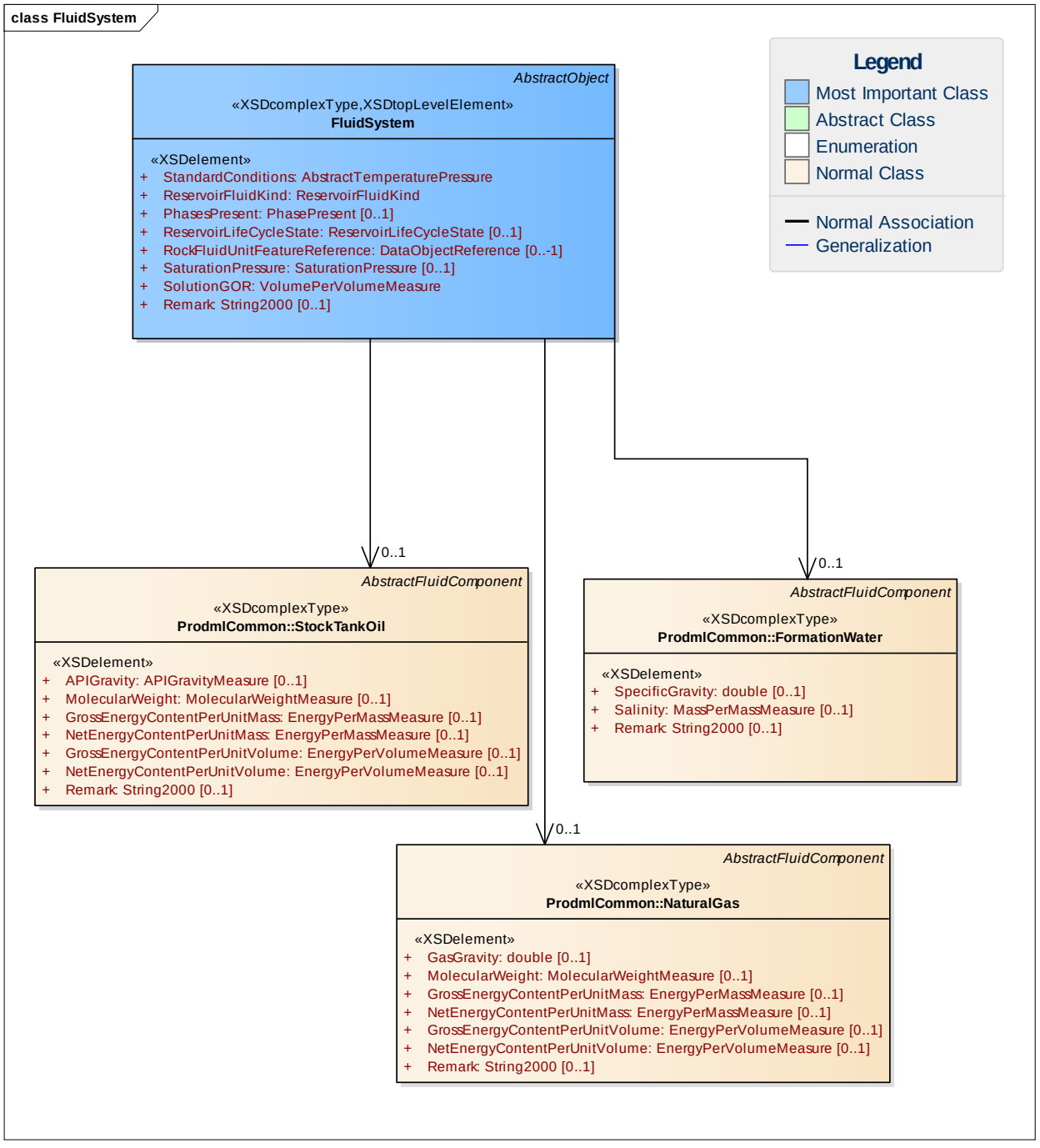


Figure 10–2. Fluid System Object (some details omitted).

10.2 Fluid Sample Acquisition Job Object

The fluid sample acquisition job data object is used to describe the method, equipment, time, place and operating conditions for each fluid sample acquired. The sample acquisition job represents the operation to collect one or more fluid samples. Fluid sample acquisition elements repeat, one per sample acquired, within one job. Fluid sample acquisitions can be made in five types of locations: surface facilities, separators, wellheads, downhole, or directly from the formation by wireline formation tester. Each type of location is defined with specific characteristics so that the important measurements for each type are

captured, such as measured depth for downhole samples and the operating conditions for separator samples. **Figure 10**–3 shows the fluid sample acquisition job data object.

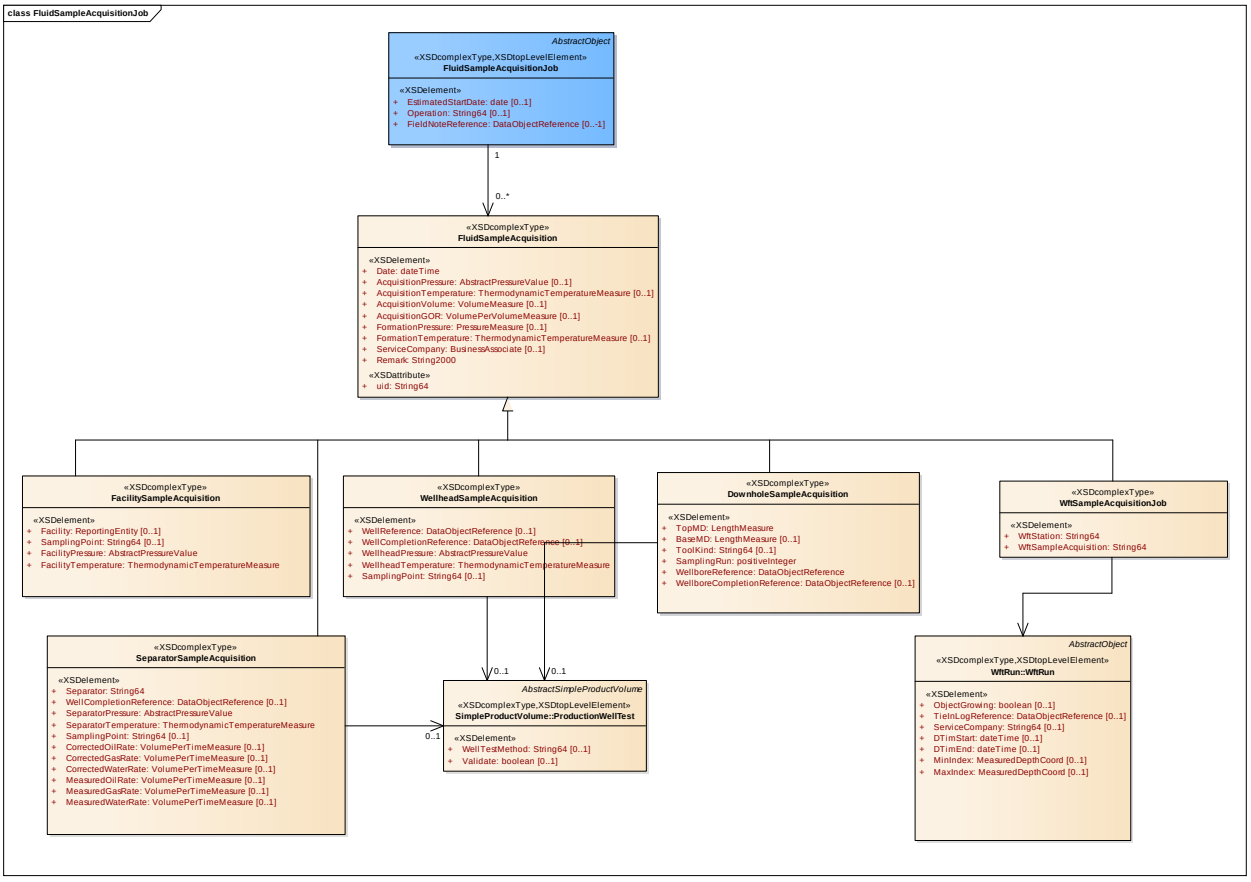


Figure 10–3. Fluid Sample Acquisition Job Object (some details omitted).

Using data objects defined elsewhere in PRODML, these sample acquisition locations may be linked to the specific points within a facility in addition to identified separators, wells, wellbores, and completions when these objects are known. To allow sample acquisition activities to be defined within the context of other well work, they may be included in the events ledger (Well Construction Management (CM) Ledger) introduced and described in the WITSML completion data object.

An important feature of the sample acquisition job data object is its ability to contain links to external documents (text, word processor or spreadsheet files) as URLs. Using the Energistics Packaging Conventions (EPC), these documents may also be packaged into a “zip”-like file for delivery with the data file. This feature is intended to provide mechanisms by which field notes and other relevant documents may be attached to the sample acquisition job data object for later use.

10.3 Fluid Sample Data Object

Initially in a sampling project, each fluid sample represents a small amount of fluid extracted from a parent fluid system, as described by the fluid sample acquisition within the fluid sample acquisition job. **Figure 10**–4 shows the main elements of the fluid sample data object, which are further explained below the figure.

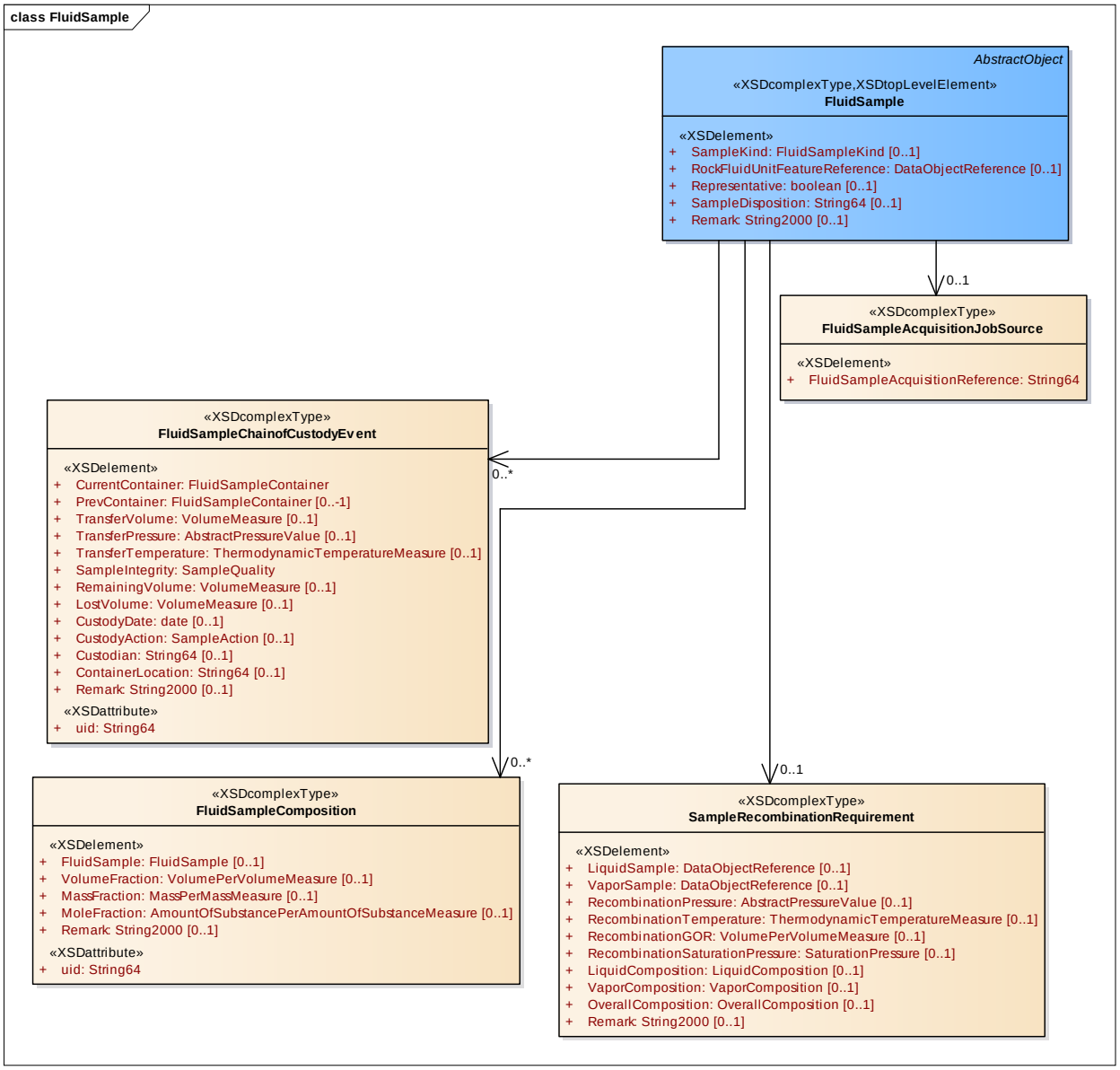


Figure 10–4. Fluid Sample Object (some details omitted).

Each fluid sample is assigned a name to identify it within the context of its lifecycle. Each fluid sample can have a description of its source geologic feature. Information such as the expected reservoir temperature, pressure, and gas-liquid ratio are also recorded for each sample (some of the data is within the associated fluid sample acquisition in the fluid sample acquisition job). Other characteristics of a fluid sample include qualitative descriptors for sample quality and representativeness of the fluid sample.

Because fluid samples may be combined with other fluid samples—either by design in the case of recombination samples or for specific laboratory investigations using additional fluids—a new fluid sample

can be described from the blending of other fluid samples. These synthesized samples are thereafter treated as regular fluid samples with additional data (the fluid sample composition) describing the identity and amounts of source fluid samples used. For a recombined liquid and vapor sample, the sample recombination requirement (i.e., the specification for the recombination) can be defined. Also, the disposition of a fluid sample may be recorded to describe what happened to the sample after analysis.

Often it is important to record the handling of the fluid sample before its analysis. To meet this requirement, a chain of custody may be transferred for each fluid sample. The chain of custody information includes the identity of the fluid sample containers in which the sample is stored, transfers of specific volumes from one container to another, the pressures, temperatures, locations and dates at which these transfers were conducted, and the identity of the custodians. As an essential element of chain of custody process, integrity checks, such as opening and closing pressures and temperatures for each fluid sample, may also be recorded for each transfer.

10.4 Fluid Sample Container Object

Each fluid sample may be contained within a fluid sample container at a point in time. These containers represent the internal tool chambers (fixed or removable) and external sample bottles in which fluid samples are stored and transported. The information recorded for these containers may include their serial numbers, owner, make, model and identity as well as its capacity, metallurgy, service compatibility, fluid integrity capability, pressure and temperature ratings, last inspection date and transport certification. **Figure 10**–5 shows the fluid sample container data object.

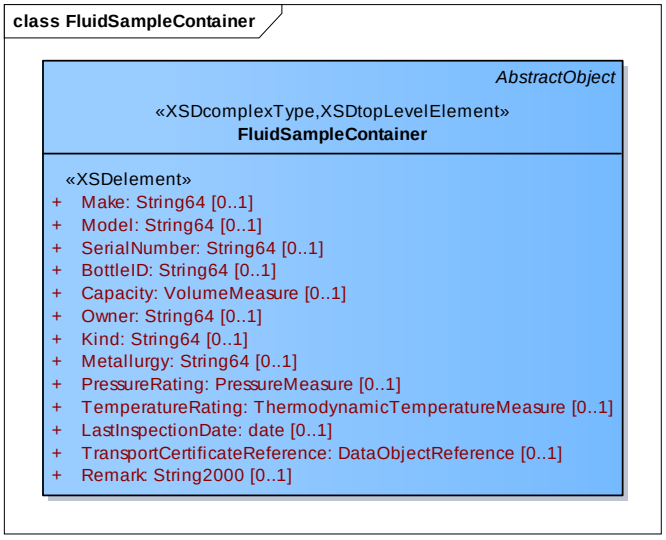


Figure 10–5. Fluid Sample Contained Object.

10.5 Fluid Analysis Object

The fluid analysis data object details the measurements made on fluid samples by a variety of laboratory tests. With the separation of fluid samples into hydrocarbon and water samples, fluid analysis has also been separated into hydrocarbon and water analysis. Hydrocarbon analysis has been described in greater detail than water analysis. For each analysis, the sample being tested is identified along with the date, purpose, analyst, and company conducting the test. As with sample acquisition, the laboratory report or reports resulting from the analysis may be linked or attached to the data file produced.

For hydrocarbon analysis, all of the routine/standard laboratory analyses have been addressed (these are listed later). The common structure of each test is to define each one with a “header” record containing contextual data and a series of “step” records that record the constant conditions and measurements. Typically, the header record is given the name of the test and a laboratory-assigned test number. Other information in the header includes the constant and/or reference conditions (such as temperature in isothermal tests) at which the test is conducted. The step record consists of the conditions changed in the

test (such as pressure) and the property measurements made at that step. For many test types, the volumes and compositions of the fluid phases present at each step may also be captured.

The same header and step record structure is used for water analysis, but only a limited set of properties have been defined. The properties for water include isothermal density, formation volume factor, viscosity, salinity, solution gas-water ratio and its corresponding gas and water (ionic) composition, and compressibility as functions of pressure. This part of the schema will be further developed in the future.

10.5.1 Types of Laboratory Analysis Supported

Figure 10–6 shows the types of laboratory analysis supported by this data object. The tests are then described in more detail in the sections following. For the use case context for these tests, see Section 9.2. The types of hydrocarbon analysis tests are shown in **Figure 10**–7.

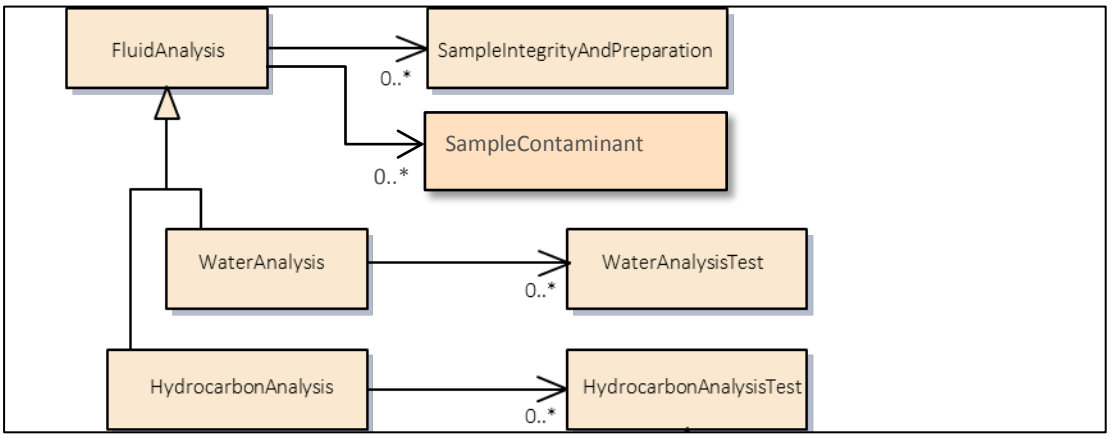


Figure 10–6. Types of fluid analysis tests.

10.5.2 Sample Integrity and Preparation

The initial description of a fluid sample is handled as a distinct test type. The data captured includes opening pressures, volumes and phases present, gas-oil ratios (GORs), densities and/or specific gravities. Contaminants or other adverse characteristics may be identified within the fluid sample, and the laboratory may conduct and record additional steps to make the fluid sample more representative of the in-situ reservoir fluid or stock tank oil. The purposes of these analyses are to establish sample validity and repeatability to ensure confidence in results of PVT testing analyses yet to be performed. Extended validation tests, such as saturation (bubble point) pressure, viscosity measurements, air content, water content, etc., are not described in this test but may be reported in other tests.

An important step in the preparation of some separator samples is the recombination of distinct oil and gas samples into a reservoir fluid sample. This involves the physical manipulation of separator oil and gas samples to duplicate the reservoir and producing conditions observed in the field. The sample produced by this test can then be subjected to a variety of PVT and flow assurance testing to describe the reservoir from which the samples were collected.

10.5.3 Sample Contaminant

A fluid analysis is conducted on one fluid sample. This sample may have been contaminated, e.g., by mud filtrate. This can be reported by the sample contaminant object. Multiple contaminants can be reported. Each reports the proportion of contaminant, its composition and other properties, and can optionally reference a sample of the contaminant itself (e.g., a mud filtrate sample which was taken for this purpose).

10.5.4 Hydrocarbon Analyses

The different types of hydrocarbon analysis can be seen in **Figure 10**–7 (at outline level). An instance of fluid analysis can contain any or all of the analyses shown.

Figure 10–8 shows the model for an example analysis (constant composition expansion) and also shows a concept common to many analysis kinds—the test step. This is a set of properties that repeat at different pressures or temperatures. To save space, the model diagrams for the remaining hydrocarbon analyses are not shown but can all be seen in the UML provided in the documentation.

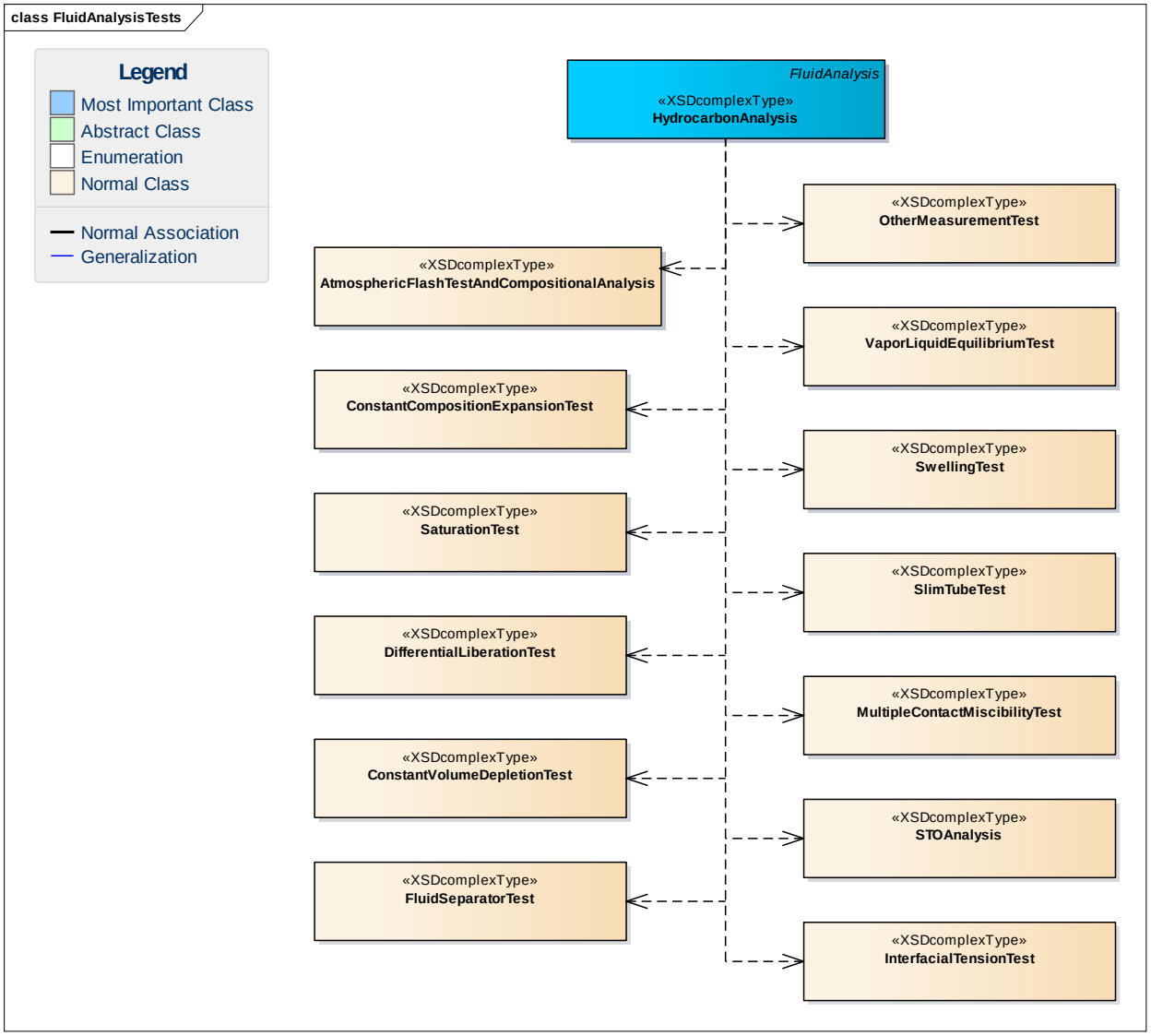


Figure 10–7. Hydrocarbon analysis types.

10.5.5 Atmospheric Flash and Compositional Analysis

An atmospheric flash takes the reservoir fluid to stock tank conditions, producing a dead oil sample and a gas sample. These samples are used to measure basic properties like the oil's API gravity, the gas-oil ratio, and the compositions of the oil and gas.

10.5.6 Constant Composition Expansion

A constant composition expansion (CCE) is an isothermal test that establishes the pressure/volume relationship for a fluid system as it may be depleted in a reservoir. Measurements of this test include saturation pressure, relative volume, single-phase densities, and other details depending on fluid type, for example, compressibility (oil), deviation factors (gas), and liquid dropout (condensates) as a fluid sample is progressively allowed to expand with decreasing pressure without removing any of the sample.

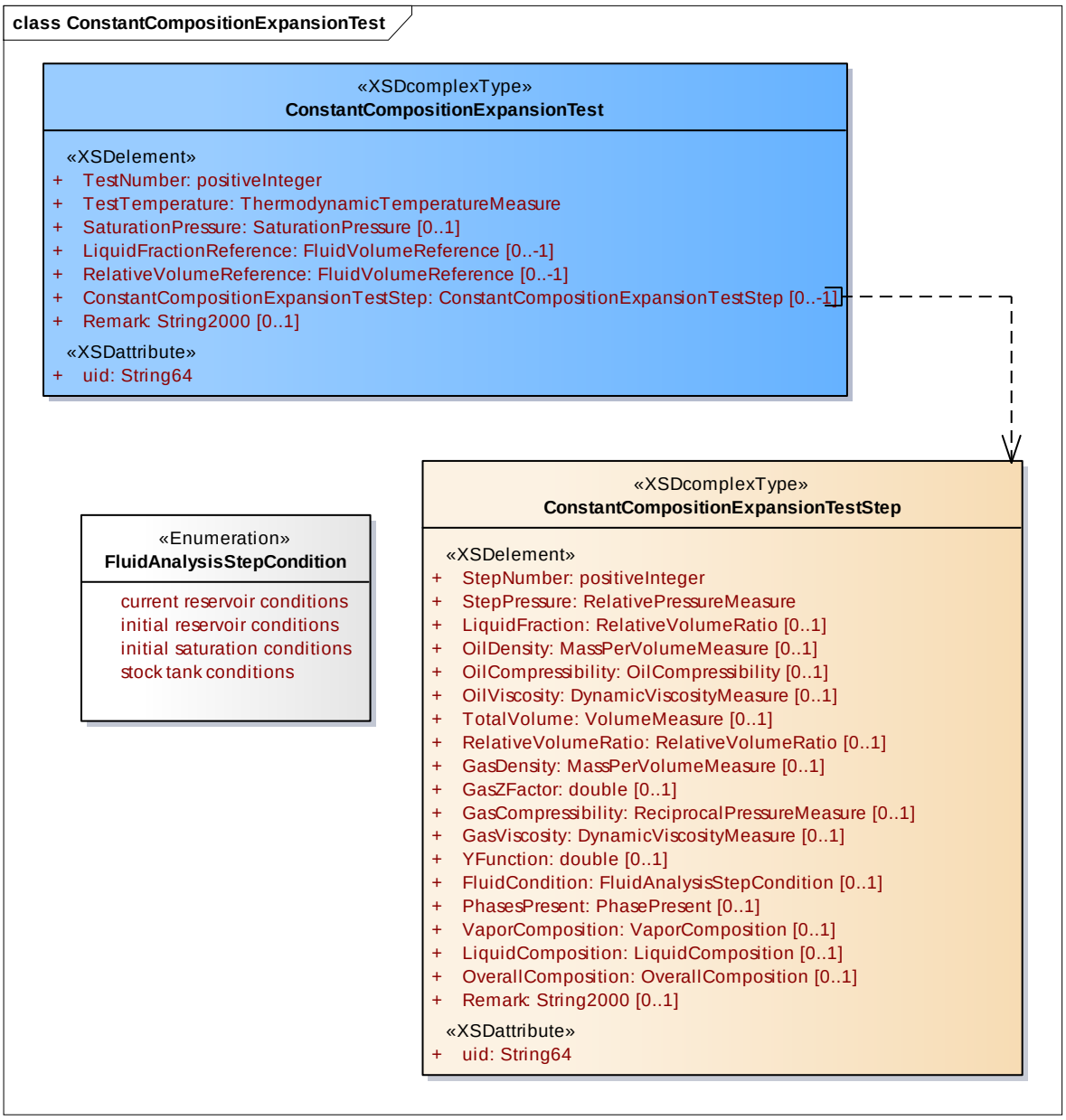


Figure 10–8. Example of a hydrocarbon analysis model (constant composition expansion).

10.5.7 Saturation Test

Within the context of PVT studies, a saturation test is a test in which bubble or dew point pressure of liquid or a gas system is determined at a constant temperature. The test involves determining pressure/volume relationship of a constant amount of reservoir at constant temperature. In most cases, a saturation point determination can be part of constant composition expansion (CCE) test. This test is considered the most basic PVT study on a live fluid sample.

10.5.8 Differential Liberation

Differential liberation is a test to simulate the depletion of a black-oil or volatile-oil reservoir system at a constant reservoir temperature. This test is designed to measure what happens to the reservoir fluid as it is depleted below the bubble point—at which point evolved gas begins to flow segregated from the oil. Measurements from this test include formation volume factors for oil and gas and the solution gas-oil

ratios and viscosities below saturation pressure. The measurements from this test, corrected to separator test results, are appropriate for material balance calculations in reservoir engineering models.

10.5.9 Constant Volume Depletion

Constant volume depletion (CVD) is a test to simulate the production of a retrograde gascondensate system through the depletion of the reservoir below the dew point. The test simulates what occurs when the liquid dropout stays in the reservoir and the surface production stream becomes leaner relative to the single-phase system. Deliverables from this test include well stream compositions below the dew point and the average liquid yields, Z-Factor, as the fluid system is produced to abandonment pressure.

10.5.10 Separator

A separator test mimics the processing of produced fluids through one or more stages of separation. Primarily for black-oil and volatile-oil fluid systems, the results of this test are used to optimize the recovery of hydrocarbon products by testing samples at a series of separation stages and settings. This test also measures key parameters, such as formation volume factors and densities, shrinkages, API gravity, and producing GOR that are used both in reservoir engineering calculations and facilities design.

10.5.11 Transport

This test is to assess the properties of the fluid relevant to transportation. A series of test steps for properties (such as viscosity, density, etc.) can be defined. Additionally, tabular data can be defined for the test, using the same tabular format as available for fluid characterization (see Section 10.6.2, page 92).

10.5.12 Vapor-Liquid Equilibrium

A vapor-liquid equilibria (VLE) test is a specialized PVT test that is used to represent phase equilibria in a gas injection enhanced oil recovery (EOR) process. In this test, a mixture of oil and injection gas is equilibrated in a fixed condition of pressure and temperature where two distinct vapor and liquid phases are present. Composition and pressure (generally at fixed reservoir temperature) of the system is designed in such a way that equilibria is established where maximum mass transfer occurs between the two phases (near critical condition/fluid). Properties of each phase (composition, density, viscosity) are also used to optimize an equation-of-state model to represent phase equilibria during gas injection EOR process.

10.5.13 Swelling Test

A swelling test is a PVT test that is used for characterization of gas injection EOR processes. In essence, a swelling test is a series of constant composition expansion tests, in each of which an increasing amount of particular injection gas is added to a reservoir oil sample at fixed temperature. In each expansion test, saturation pressure and physical and transport properties of the mixture (density and viscosity) can be determined. Swelling test data are used to optimize an equation-of-state model that properly represents phase behavior of reservoir oil and injection gas mixture at different mixing ratios.

10.5.14 Slim Tube Test

A slim tube test is used to study the miscibility condition of a reservoir oil sample when it comes in contact with a gas sample flowing inside a long tube filled with sand (porous media) in a near one-dimensional flow condition. In general, a number of slim tube tests are performed at constant temperature with varying pressures to determine the minimum pressure at which miscibility is achieved after injection of certain pore volume of gas. The minimum miscibility pressure (MMP) together with production information during slim tube tests may be used to optimize an equation-of-state model for use in reservoir simulations. Slim tube tests may also be used to optimize injection gas composition that ensures miscibility at condition of particular oil reservoir.

10.5.15 Multiple Contact Miscibility

A multiple contact miscibility test is a specialized PVT test that represents establishment of miscibility when a gas and oil sample come in contact under a fixed condition of pressure and temperature. The test involves multiple contact of a fixed amount of gas with multiple fresh charges of an oil sample, where intermediate components from the oil sample are stripped consecutively (forward contact). In case of a backward contact test, a fixed charge of an oil sample will come in contact with multiple charges of fresh rich gas, so that intermediate components in the gas phase are transferred to the oil phase. The process continues until miscibility is achieved. During each contact, the composition of each phase is measured. Such mass transfer process and compositional change may be shown using a pseudo-ternary diagram. Results of such study are generally used to optimize an equation-of-state (EoS) model. The optimized EoS model is generally used in a compositional reservoir simulator to properly capture the miscibility process in a reservoir during a gas injection EOR process.

10.5.16 STO Analysis

For representative samples of stock tank oil, an additional set of stock tank oil (STO) analysis tests may be conducted to provide measurements of viscosity (at temperature), pour point, cloud point, wax appearance temperature, paraffin content, asphaltene content and SARA, Reid vapor pressure, total acid number, total sulfur, molecular weight, water content, and the amounts of lead, nickel, vanadium and elemental sulfur present.

10.5.17 Interfacial Tension

Interfacial tension tests measure the interfacial tension between two phases. These are labelled the wetting phase and the non-wetting phase. A surfactant can also be defined. Each test step reports the interfacial tension and optionally, the surfactant concentration.

10.5.18 Water Analysis

In addition to the pressurized tests described above, a number of standardized lab tests can be performed on water samples. The subset of measurements supported at this time includes formation volume factors, viscosities, and a set of anion/cation concentrations that can be used for scale modeling and for water injection capabilities.

For each of these tests, a “header” is created for the appropriate test within the fluid analysis data object. This record describes the test requested and captures the identifiers and constant parameters needed, such as temperature in an isothermal test. Subsequently, when the test is performed, each test step is recorded as a child record within the “header” record. These test results are communicated in this tabular style to applications and/or data stores using this data. As many tests as needed may be described in one fluid analysis data object. In addition to the conditions and measurements of each test, an attachment mechanism allows for the laboratory to upload a copy of the report.

10.6 Fluid Characterization

The fluid characterization data object (**Figure 10**–9) describes the characteristics and properties of a fluid sample or fluid system under the conditions expected in a historical or future state. The characterization may be based on the measurements from one or more fluid samples, and usually (but not always) uses a specialized application that simulates fluid behavior. The typical workflow is to characterize individual fluid samples as soon as their data are delivered to the sample owner from a fluid analysis laboratory. Although not standardized, the typical approach includes iterative optimization of characteristics (and in some cases amount) of individual compositional components representing the fluid sample, and parameters of an EoS model (or other mathematical models) to simultaneously represent (simulate), and evaluate (consistency/QC check) experimental PVT data. In some cases, it is required to either represent multiple fluid samples with one characterization or to re-optimize existing characterization for different operating conditions.

Such a developed characterization may be used to produce a data set descriptive of the fluid behavior appropriate for use in upstream workflows, such as reservoir simulation, flow in tubulars, and flow through

surface facilities. Flow assurance data (e.g., solid deposit rates, multiphase flow regime maps) are not supported now, although some of the data used (e.g., asphaltenes) are included.

Two basic formats are available to represent fluid characterization results for delivery to consumer applications:

- Model (parametric)
- Tabular

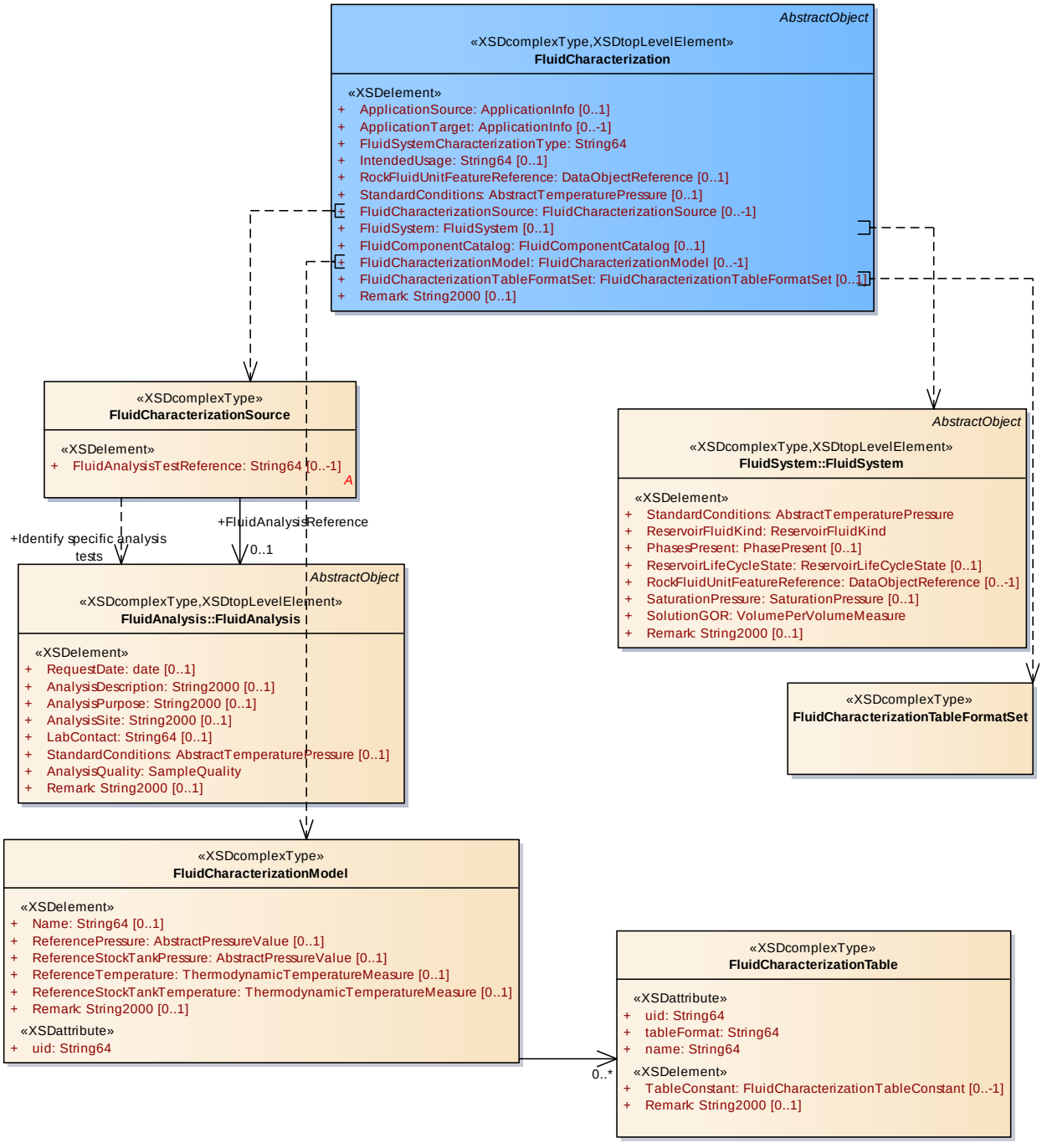


Figure 10–9. Fluid characterization high-level model.

The basic structure is shown in Figure 10–9. A fluid characterization has any number of fluid characterization sources. Each is a reference to a fluid analysis and to the specific tests in that analysis which were used for this characterization. The fluid system is also referenced. There are then a set of fluid characterization models, and these can either have model parametric representation (see below), and/or a set of tables of output properties. The format of these tables is contained in its own object.

A fluid characterization may also be represented as a combination of tabular and parametric approaches (e.g., in thermal compositional reservoir simulation applications), where part of fluid characterization (e.g., k-values) is represented in a parametric model (EoS), and another part of fluid characterization (e.g., component viscosities) is represented in a tabular format. It is important to note that the current schema is designed to represent only one fluid sample/system at a time.

With either representation, information is included to describe the geologic feature and the fluid system being characterized. Also provided are references to the specific fluid samples and laboratory tests that were used as the basis for the characterization. Additional information found in the fluid characterization data object are the identity (and version) of the application used for its characterization, and the use case for which the characterization was done.

10.6.1 Model-Parametric Output

In the model-parametric approach, parameters of a named EoS model determined during the characterization effort are provided to an application using the same model formulation. Typical information found in a parametric format includes:

- Identification and parameters of EoS models.
- Characteristics of each component used in characterization, such as critical properties, acentric factor, etc.
- Identification and parameters of viscosity models.
- Parameters of equations to represent thermal properties of components.
- Mixing parameters for density and viscosity models.

The models are arranged in a hierarchy, with the top level being compositional (i.e., with a set of fluid components each having fluid component properties), or correlation (i.e., a set of parameters and a model which fits the behavior of the whole fluid) (**Figure 10**–10). A fluid characterization model can have a model specification, and the specification is abstract down to the level of a known model which then has its required parameters (see below for more details).

Correlation models are currently all viscosity models and are divided into dead oil, bubble point, under-saturated oil and gas viscosity correlations.

Compositional models are available as EoS (phase behavior) and viscosity models.

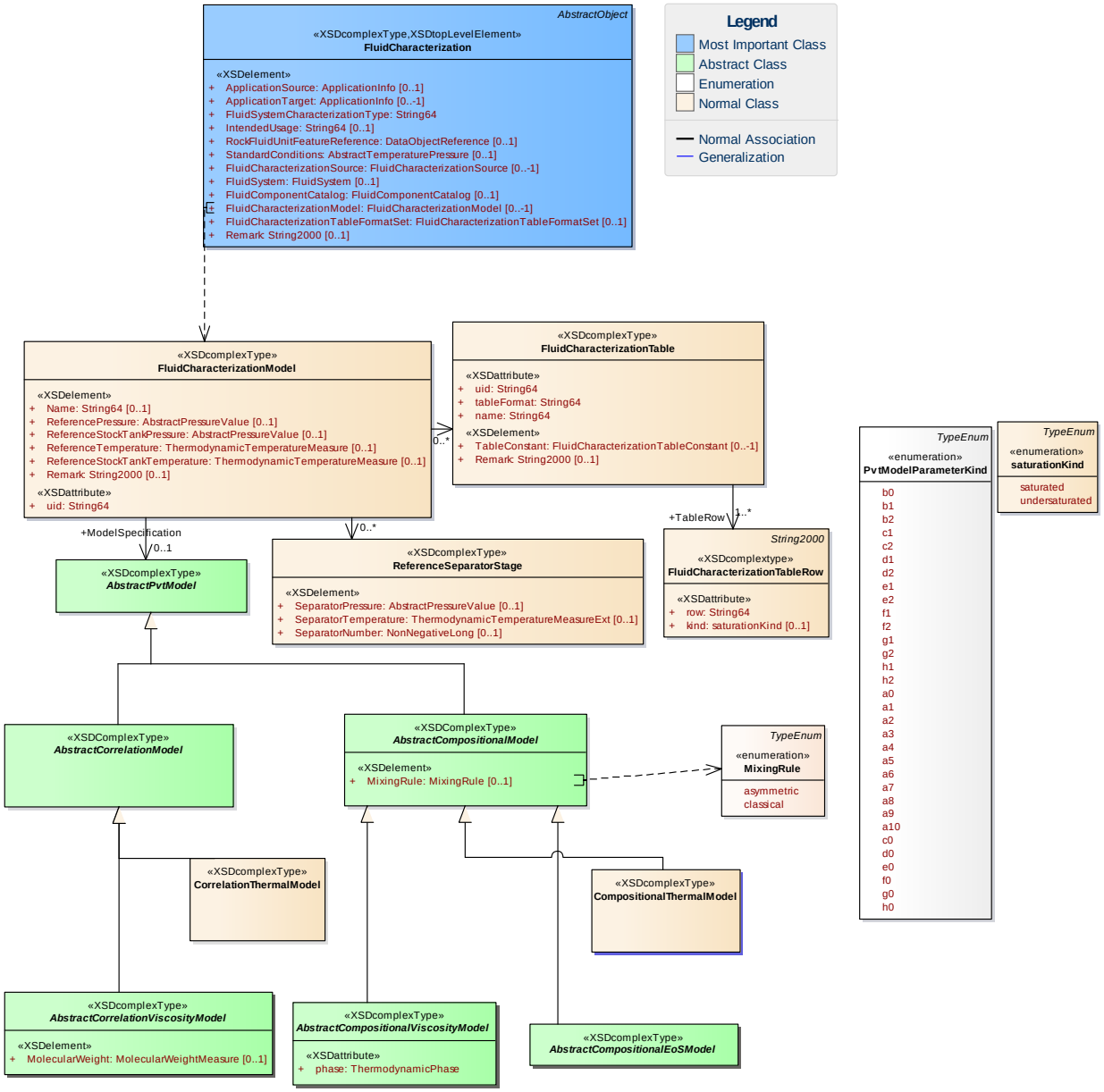


Figure 10–10. Top Level of PVT Model hierarchy.

Figure 10–11 shows more details of the models. All models (correlation and compositional) can have model-wide PVT model parameters. All models (correlation and compositional) can also have custom extensions, allowing extra parameters which specialist studies may have defined, to be added. Compositional models also can have a range of fluid component properties defined per fluid component, and a set of binary interaction coefficients, each relating one fluid component to another.

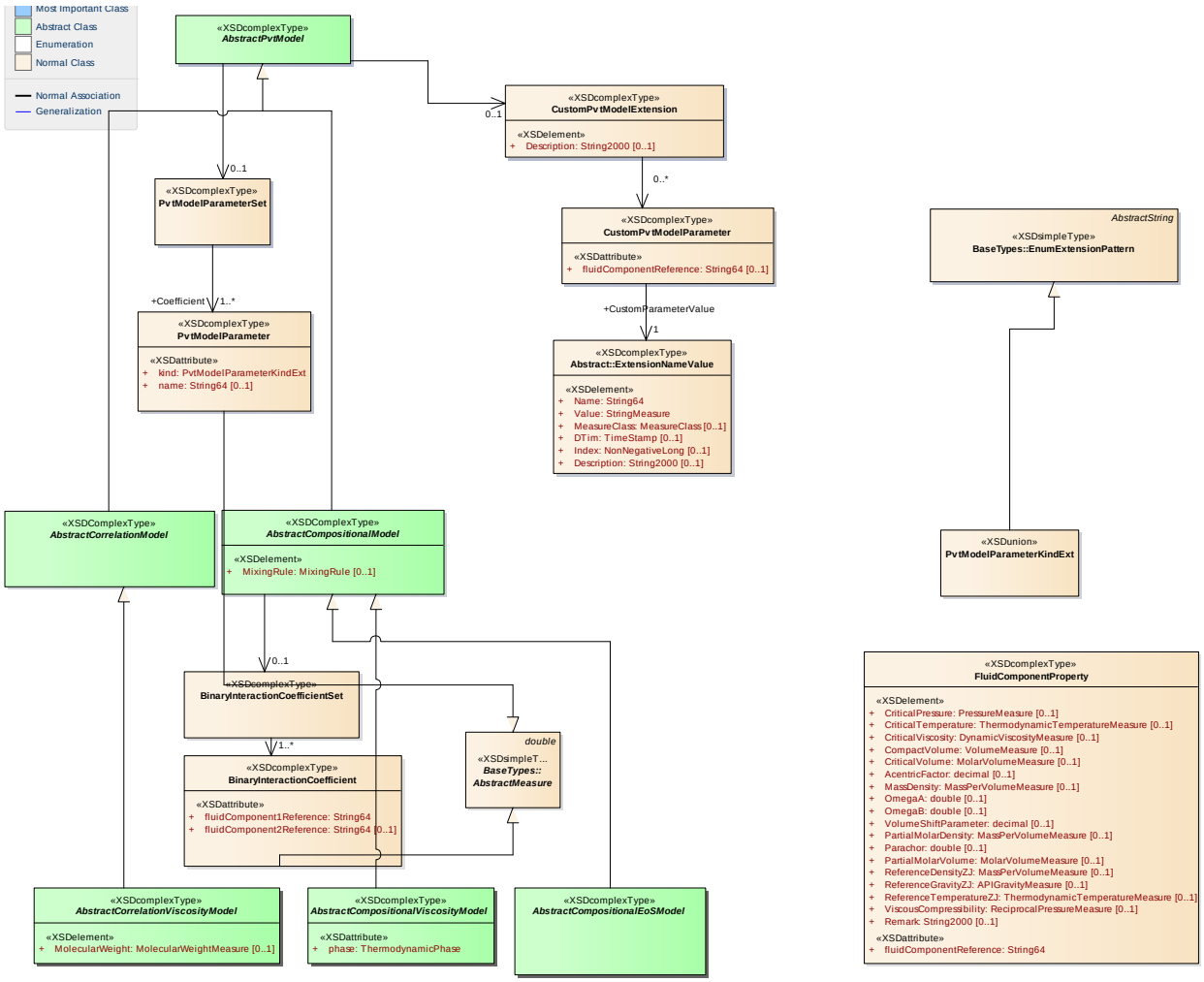


Figure 10–11. Details of PVT model types and parameters available.

For a list of the specific models (which are concrete classes that may be instantiated in the XML data transfer), see Section 12.2. The schema is designed so that each model inherits its correct parameter set, and as noted above, custom parameters for extended models can be added.

10.6.2 Tabular Output

In a tabular format, fluid properties (such as pressure, density, viscosity, etc.) are provided as rows of a table where a target application could use these fluid properties directly. The organization of each table is defined for each characterization file, and many table formats may be used in a single characterization. This organization gives the ordering, properties, and units of measure for each column, and allows constant values and delimiters to be assigned. Typical information found in a tabular format includes:

- Independent operating conditions such as pressure and temperature.
- Fluid properties such as oil and gas formation volume factor, solution gas or dissolved liquid, viscosity and compressibility.
- Slope of compressibility/formation volume factor vs. pressure line in the under-saturated region, if saturated fluid properties are provided.
- Tabular representation of k-values or component viscosities as a function of pressure and temperature.
- Miscibility tables in case of gas/solvent EOR fluid characterization.

The organization of the tabular model is shown in Figure 10–12. The table format defines null values and delimiters, plus a set of column headings which detail what comes in each column. The rows are then strings of delimited values (e.g., CSV) which contain the values in the columns, plus a saturated/under-saturated flag. Additionally, the table can contain table constants.

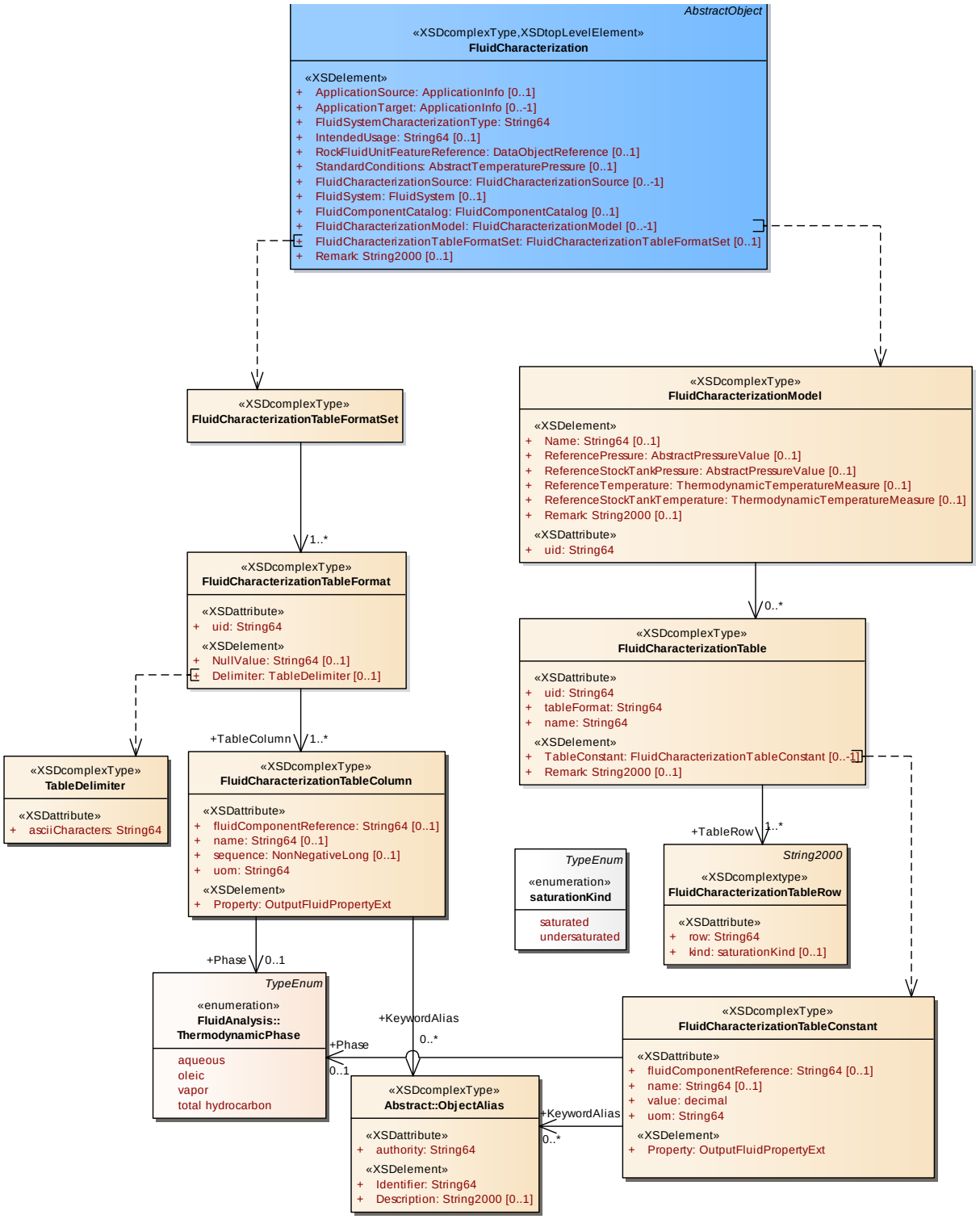


Figure 10–12. Tabular Output model.

10.7 Fluid Composition

Fluid composition refers to the mixture of fluid components in a hydrocarbon fluid. A fluid composition has a list of fluid components and the molar fraction of each.

Fluid components are the individual molecular types and are specified by referencing them inside the fluid component catalog. This method of describing fluid composition is used both in the fluid analysis and the fluid characterization objects. The roles of fluid composition, fluid component and fluid component catalog are shown in **Figure 10**–13.

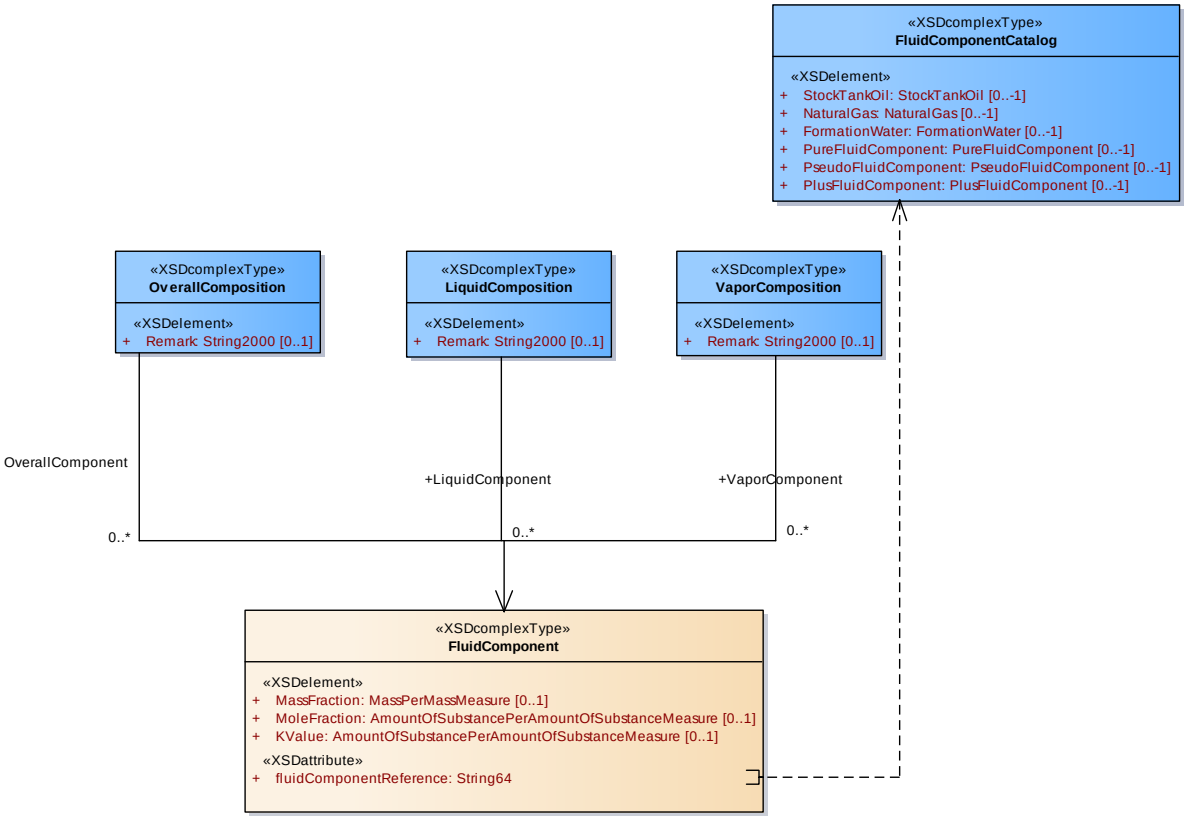


Figure 10–13. Fluid Composition model common to all PRODML

Within the fluid analysis and the fluid characterization objects, each instance of a component within a fluid composition refers to one of the members of the fluid component catalog in that same object, using the UID for reference. The inclusion of a fluid component catalog can be seen in Figure 10–9 showing fluid characterization. Examination of the hydrocarbon analysis class attributes (it is shown in outline in Figure 10–7) shows it too contains a fluid component catalog. What this means is that every characterization within a fluid characterization—and every hydrocarbon analysis within a fluid analysis—share the one set of possible fluid components within the parent analysis or characterization object. Note that every composition does not need to use *all* of the fluid components in the fluid component catalog, just that every component which is going to be used to describe a composition in a fluid analysis or fluid characterization must be included in the fluid component catalog.

An example of the use of a fluid composition within fluid analysis can be seen in Figure 10–8, which shows constant composition expansion. Each test step (a pressure-temperature step) contains vapor, liquid and overall compositions. Each of these is shown in Figure 10–13.

Examples of the use of referencing to a fluid component within the catalog within fluid characterization can be seen in Figure 10–11: in the fluid component property class, the attribute at the bottom of the list is the fluid component reference. This will contain the UID for the fluid component which is within the catalog for the parent fluid characterization object. This class contains the various parameters required for that

component. The binary interaction coefficient has the same referencing but for the two components whose binary interaction coefficient is being reported.

Figure 10–14 shows the fluid components in the catalog and includes:

- stock tank oil
- natural gas
- formation water
- pure fluid component
- plus fluid component
- pseudo fluid component

The first three are aimed at black oil descriptions and the second three at compositional descriptions; however, the two kinds may be mixed. Figure 10–14 shows the attributes of each type of fluid component.

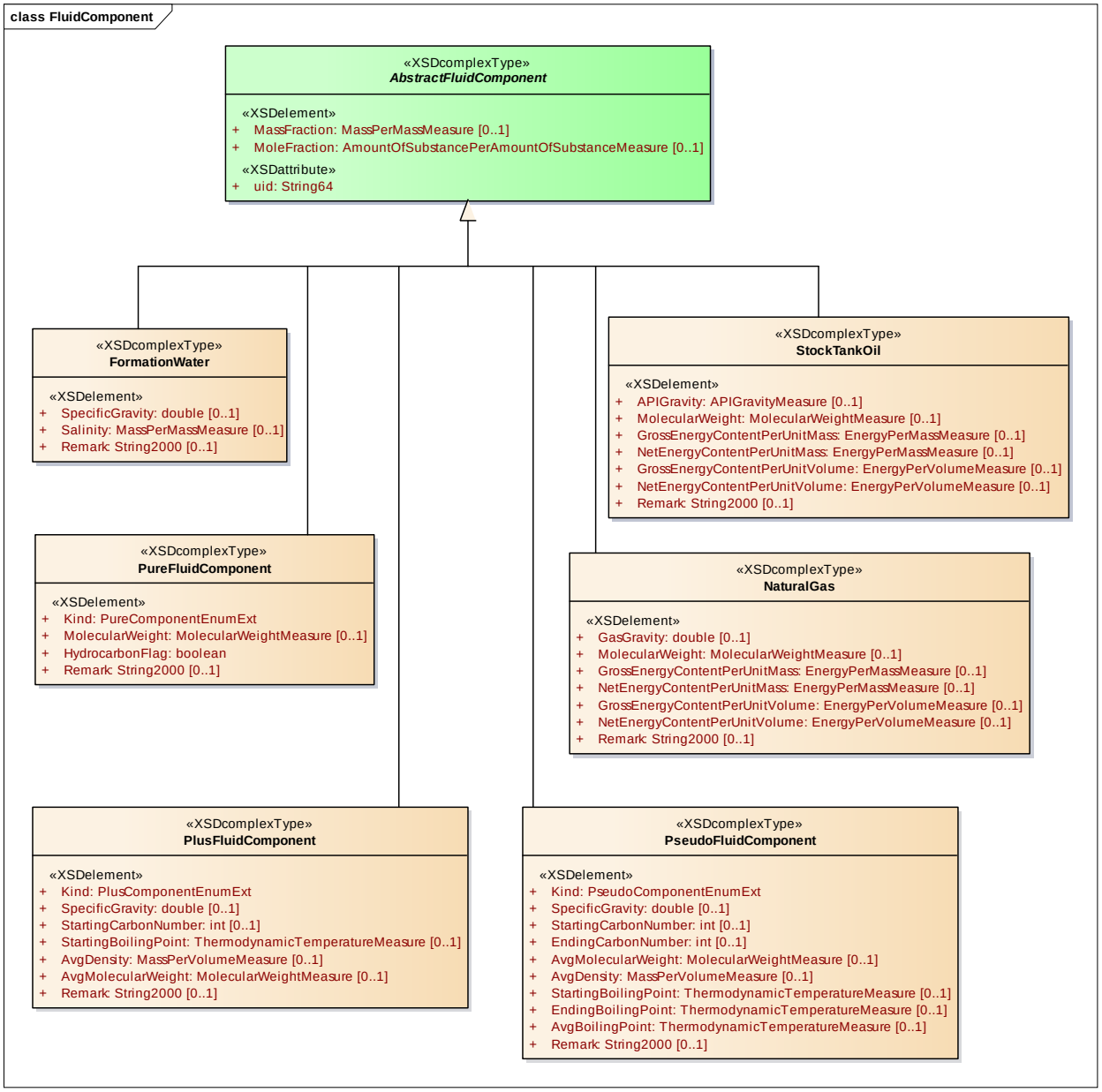


Figure 10–14. Types of fluid components supported for fluid composition.

Section 12.1 lists all the fluid components defined in the schema. However, the enumerated lists are extensible so user-defined components can be added.

Note that the Fluid Component Catalog and the Fluid Components are contained in the PRODML Common package, and are shared with the Simple Product Volume data objects, so are compatible with production volume reports created using PRODML v2.

11 Fluid and PVT Analysis: Worked Examples

To help explain how the PVT data objects have been designed to work, an example (all files as part of the PRODML download) is included and explained here.

The worked example is found in the folder energyml\data\prodml\v2.0\doc\examples\PVT\Worked Examples and all the files are called “Good Oil 4 xxx” where “xxx” is the name of the data object type and optionally a number.

11.1 End-to-End Worked Example

The main worked example supplied covers a complete workflow, from acquiring a sample to characterizing it. All the data objects, and multiple key types and sub-classes within them are used to show how this workflow comes together in a typical scenario. **Figure 11**–1 shows the worked example workflow.

In a fluid sample acquisition job, two samples are taken from a fluid system, filling two fluid sample containers. One sample is invalid and destroyed. The other sample is split by having a water sample removed. The hydrocarbon phase and the water phase are then sent for appropriate fluid analyses. The hydrocarbon is then characterized using compositional models, and tabular output is generated from black oil models,

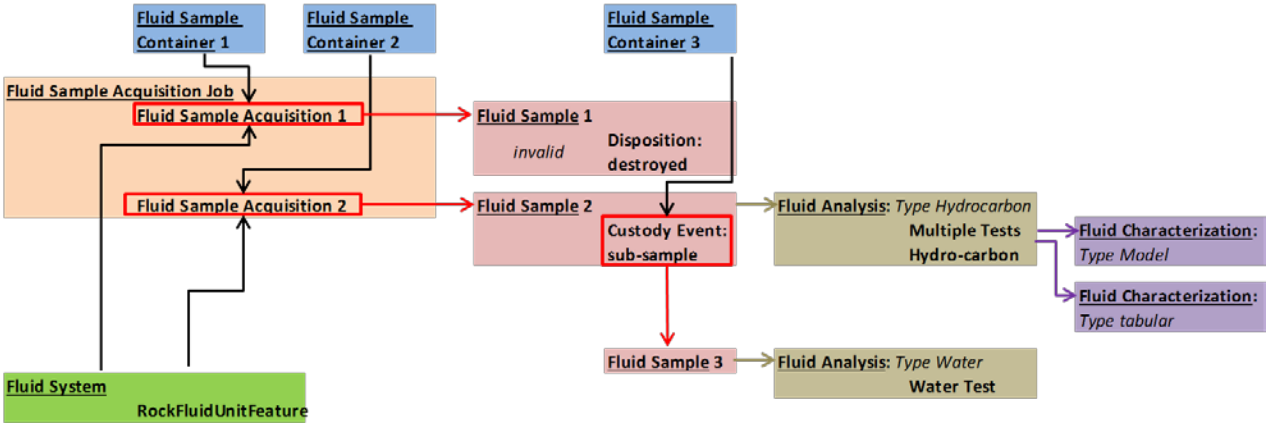


Figure 11–1. Overall workflow for PVT worked example. Top-level object types each given own color and types underlined. Red boxes indicate events that result in the creation of a new object.

11.1.1 Fluid System

The fluid system is referenced by many other objects (**Figure 11**–2). This is not a detailed object; the full model for a reservoir and its sub-divisions containing hydrocarbons is the domain of RESQML. The rock fluid unit feature in RESQML is the level of granularity which can be reproduced in the fluid system description. Multiples can be included and they may correspond to, e.g., layers within an overall reservoir comprising a fluid system.

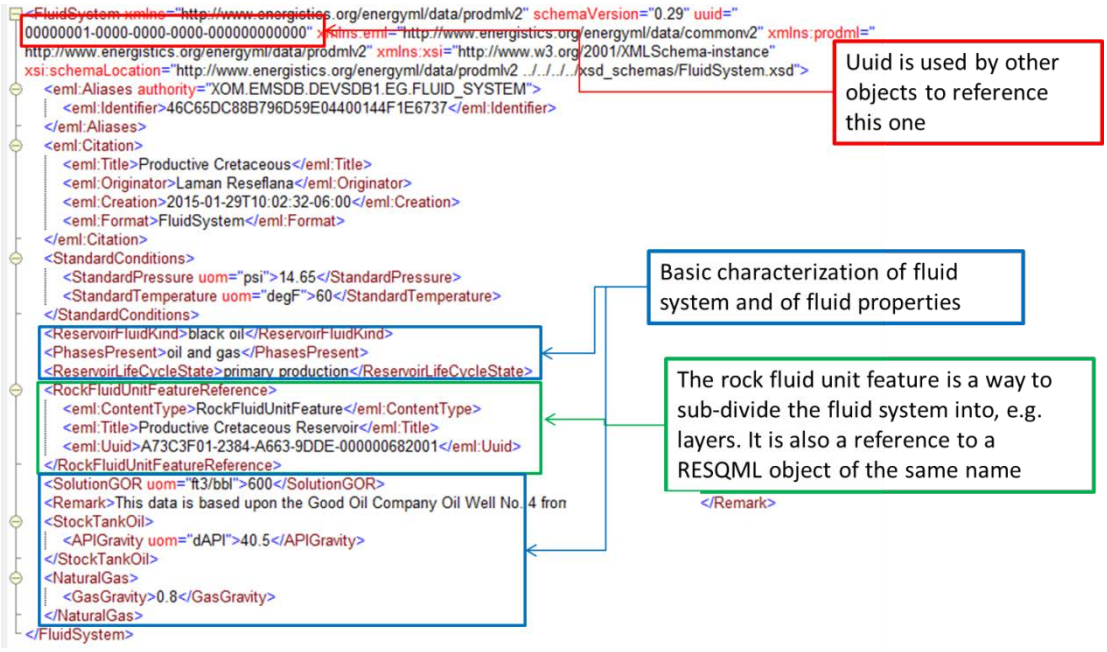


Figure 11–2. Fluid system describes the reservoir and fluid contained in it (high-level outline).

11.1.2 Fluid Sample Acquisition Job

This object describes the operation whereby one or more samples are acquired. The high level outline of the worked example is shown in **Figure 11**–3**.** The fluid sample acquisition element recurs, one for each sample acquired within the job.

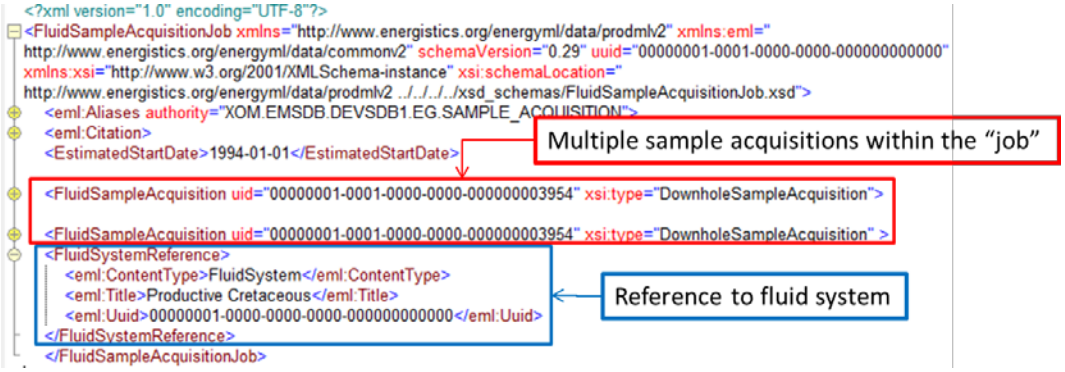


Figure 11–3. Fluid sample acquisition job describes the operation of sample acquisition.

As described in Section 10.2, there are 5 types of fluid sample acquisition; the worked example is for the downhole sample acquisition type. This represents the data from a downhole sampling tool in a flowing well. One of the fluid sample acquisitions is shown in **Figure 11**–4. Additional data can be added from the schema. Each type of sample acquisition has its own data and references. A well, wellbore and a well test are included in the examples and are referenced from this fluid sample acquisition.

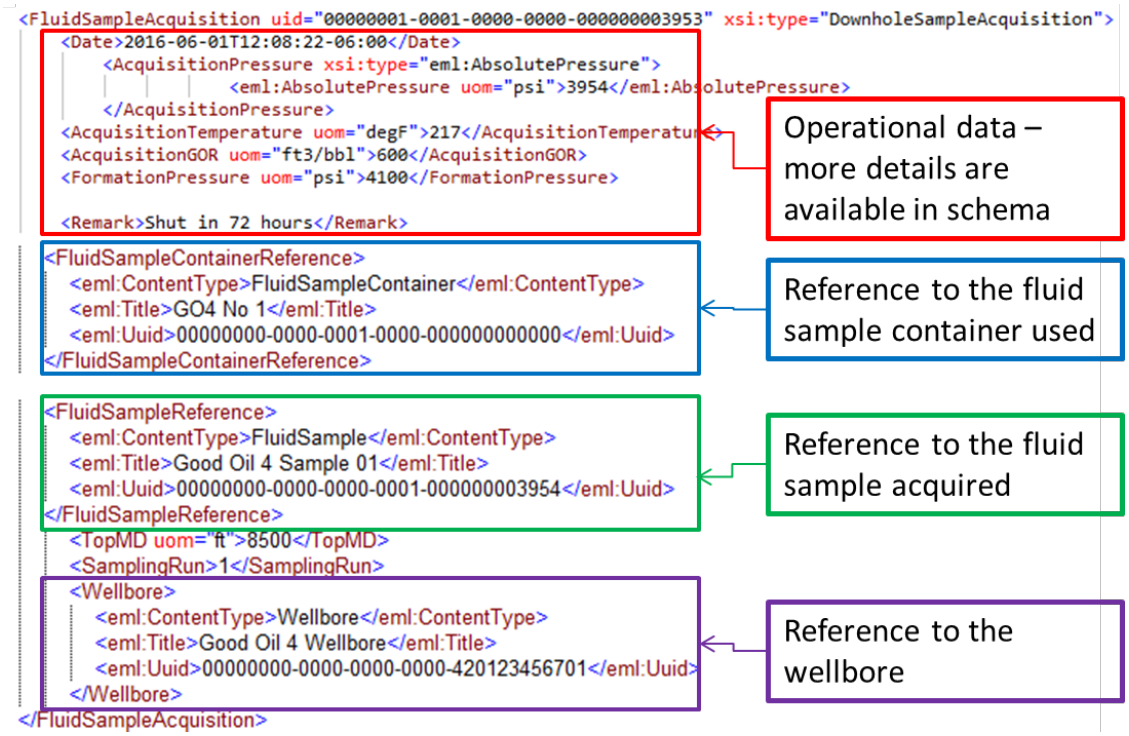


Figure 11–4. Fluid sample acquisition has the details of the operation to acquire each individual sample.

11.1.3 Fluid Sample Container

Three fluid sample containers are used in the example. Two for the original fluid sample acquisition and one for the water sample which is sub-sampled from the retained sample. An example is shown in **Figure 11**–5. These are relatively simple data objects, which are referenced from other objects as needed.

```
<FluidSampleContainer xmlns="http://www.energistics.org/energyml/data/prodmlv2" schemaVersion="0.29" uuid="
00000000-0000-0001-0000-000000000000" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:eml="
http://www.energistics.org/energyml/data/commonv2" xmlns:prodml="http://www.energistics.org/energyml/data/prodmlv2"
xsi:schemaLocation="http://www.energistics.org/energyml/data/prodmlv2 ../../../../xsd_schemas/FluidSampleContainer.xsd">
  <eml:Aliases authority="XOM.EMSDB.DEVSDB1.EG.SAMPLE_CONTAIN">
    <eml:Identifier>DEF564DD9E31D47BE040D70A2C4F2D67</eml:Identifier>
  </eml:Aliases>
  <eml:Citation>
    <eml:Title>GO4 No 1</eml:Title>
    <eml:Originator>Laman Reseflana</eml:Originator>
    <eml:Creation>2015-01-01T13:14:33-06:00</eml:Creation>
    <eml:Format>FluidSampleContainer</eml:Format>
  </eml:Citation>
  <Make>Downhole Bottles</Make>
  <Model>Brisbane</Model>
  <SerialNumber>12-435287-a7hs73m</SerialNumber>
  <BottleID>Bottle.01</BottleID>
  <Capacity uom="mL">750</Capacity>
  <Owner>Bill's Bottle and Brass Emporium</Owner>
  <Kind>single phase</Kind>
  <Metallurgy>unknown</Metallurgy>
  <PressureRating uom="psi">15000</PressureRating>
  <TemperatureRating uom="degF">392</TemperatureRating>
  <LastInspectionDate>1993-10-15</LastInspectionDate>
  <Remark>this is the first of two bottles.</Remark>
</FluidSampleContainer>
```

Figure 11–5. Fluid sample container example. It does not reference other objects; they reference it.

11.1.4 Fluid Sample

A total of three fluid samples exist in the worked example:

1. Sample is acquired, found to be invalid and destroyed.
2. Sample is acquired and retained, and used for hydrocarbon analysis.
3. Sample is sub-sampled from Sample 2, with a water sample being removed and used for water analysis.

Important information about the sample concerns its provenance: where it came from in the reservoir, and which operation created it. This aspect is shown in **Figure 11**–6.

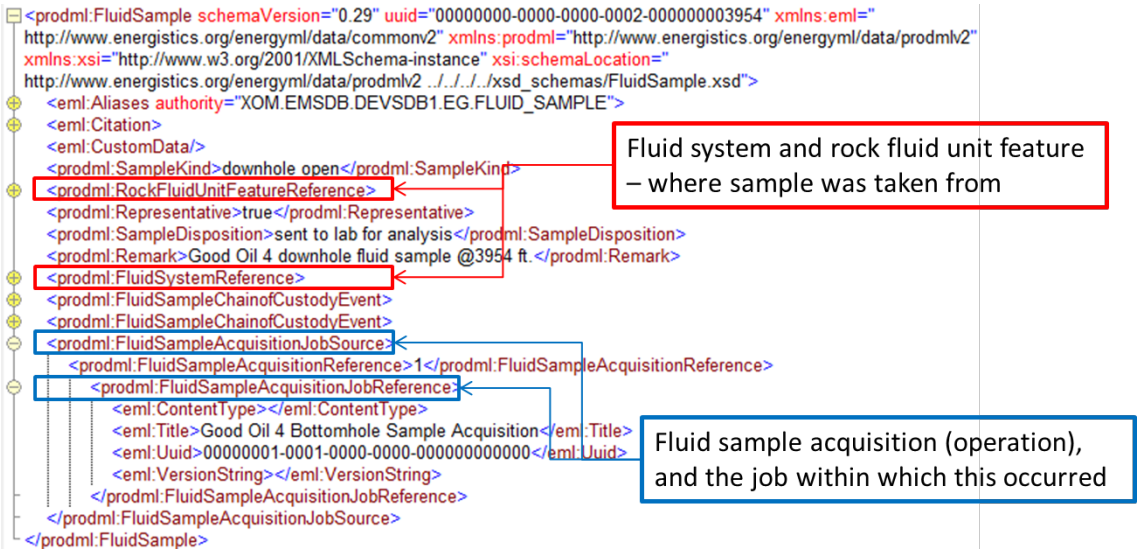


Figure 11–6. Fluid sample showing references to its provenance.

Note that the fluid sample container for a fluid sample is not a 1:1 correspondence because the sample may be moved between containers during its lifetime. The original fluid sample container is recorded with the fluid sample acquisition job. Any subsequent changes are recorded in the fluid sample chain of custody event element. Two such are recorded for Sample 2:

1. Custody transfer to a new custodian (a PVT Lab), also recording the current fluid sample container.
2. Sub-sample of dead fluid taken. This is into a new container and creates a new fluid sample. This event is shown in detail in **Figure 11**–7.

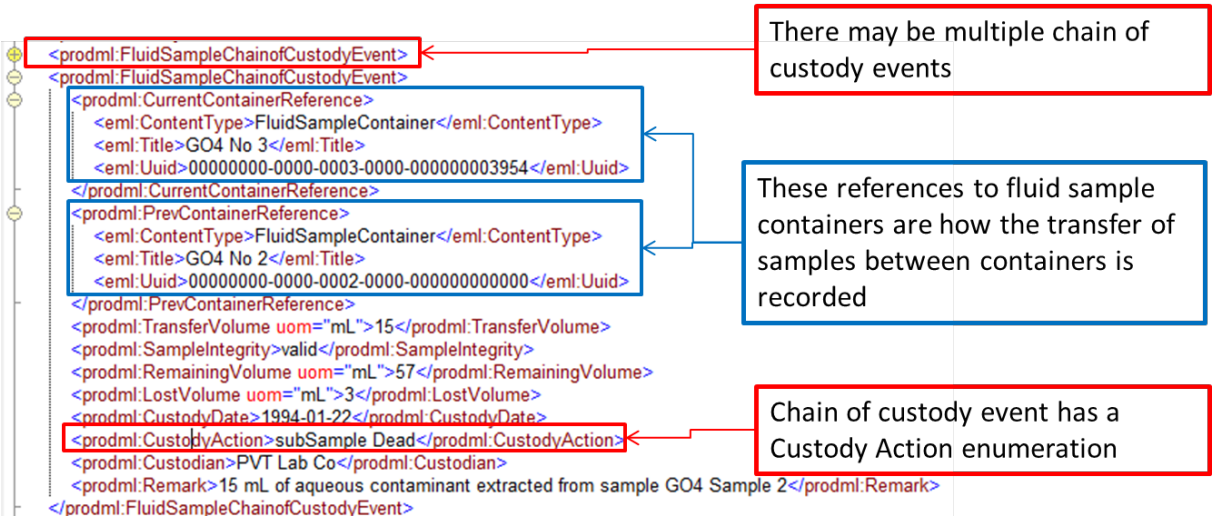


Figure 11–7. Fluid sample chain of custody event— showing a sub-sample of aqueous phase being taken.

Note that while not shown in the example, an important feature of the model is the ability to combine samples into a new sample, for example recombining separator oil and gas phase samples.

11.1.5 Fluid Analysis

11.1.5.1 Hydrocarbon Analysis

The hydrocarbon analysis example illustrates a number of key features of the fluid analysis:

1. Reference to fluid sample analyzed
2. Reference to an external report file
3. The fluid component catalog
4. Details of sample contaminants
5. Details of sample integrity and preparation
6. Details of the analysis tests themselves.

This high-level content is shown in **Figure 11**–8. Items 3 - 6 on this list are expanded below.

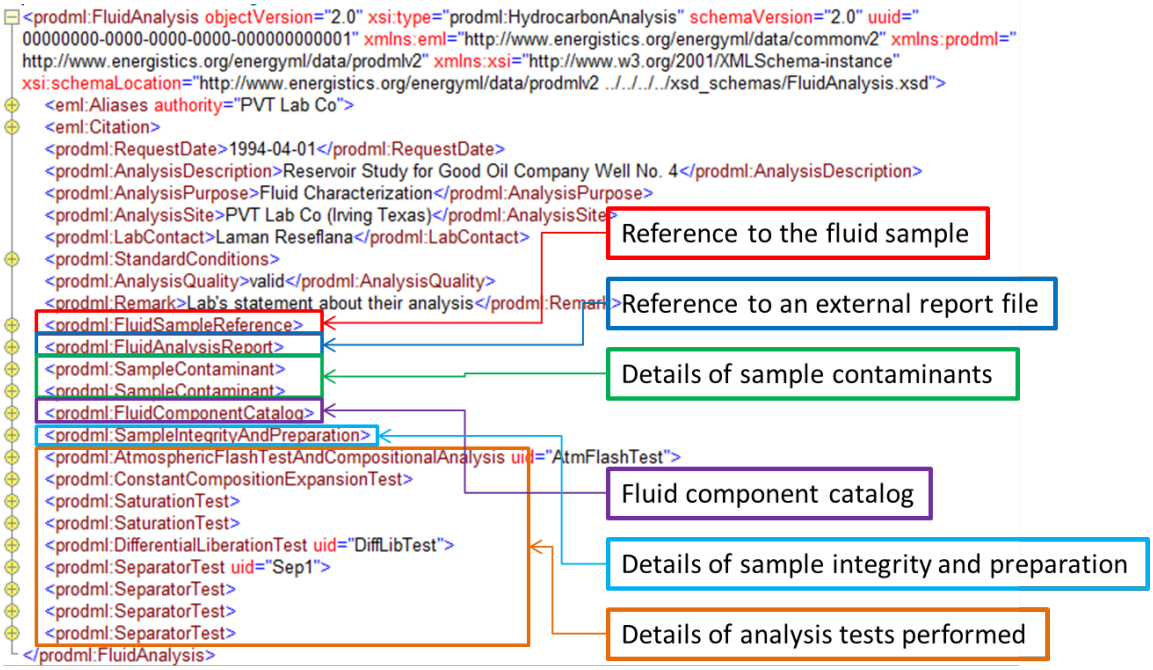
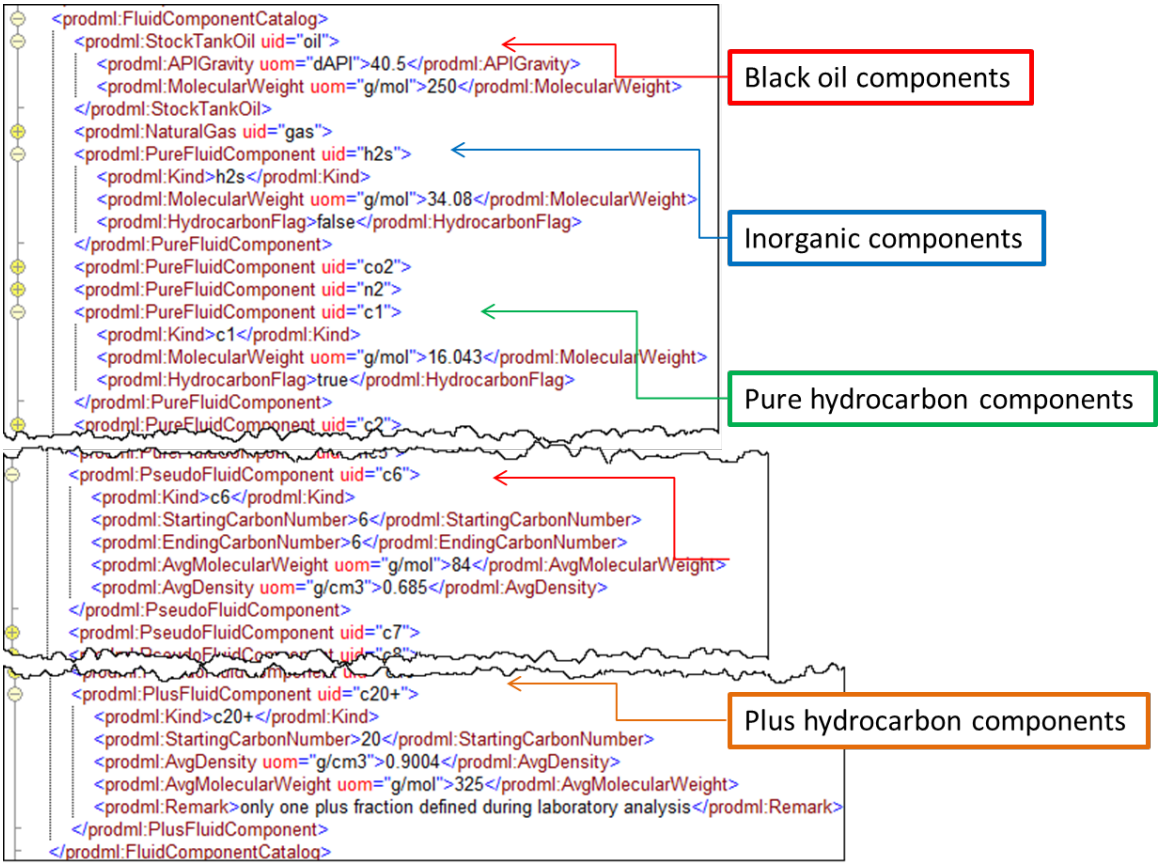


Figure 11–8. Fluid analysis—hydrocarbon, high-level content.

The fluid component catalog is unique to each fluid analysis object (also to each fluid characterization). The worked example contains a catalog containing all the kinds of fluid component, as listed and shown in **Figure 11**–9. Each analysis test then references items from the fluid catalog to report molar composition, etc. Note that the UID attribute is the means by which a fluid component is referenced from a test. See **Figure 11**–12 for an example. The UID only needs to be unique within this data object; it is not a UUID. Hence, any convenient naming convention can be used.




```
<prodml:FluidComponentCatalog>
  <prodml:StockTankOil uid="oil">
    <prodml:APIGravity uom="dAPI">40.5</prodml:APIGravity>
    <prodml:MolecularWeight uom="g/mol">250</prodml:MolecularWeight>
  </prodml:StockTankOil>
  <prodml:NaturalGas uid="gas">
  <prodml:PureFluidComponent uid="h2s">
    <prodml:Kind>h2s</prodml:Kind>
    <prodml:MolecularWeight uom="g/mol">34.08</prodml:MolecularWeight>
    <prodml:HydrocarbonFlag>false</prodml:HydrocarbonFlag>
  </prodml:PureFluidComponent>
  <prodml:PureFluidComponent uid="co2">
  <prodml:PureFluidComponent uid="n2">
  <prodml:PureFluidComponent uid="c1">
    <prodml:Kind>c1</prodml:Kind>
    <prodml:MolecularWeight uom="g/mol">16.043</prodml:MolecularWeight>
    <prodml:HydrocarbonFlag>true</prodml:HydrocarbonFlag>
  </prodml:PureFluidComponent>
  <prodml:PureFluidComponent uid="c2">
  <prodml:PseudoFluidComponent uid="c6">
    <prodml:Kind>c6</prodml:Kind>
    <prodml:StartingCarbonNumber>6</prodml:StartingCarbonNumber>
    <prodml:EndingCarbonNumber>6</prodml:EndingCarbonNumber>
    <prodml:AvgMolecularWeight uom="g/mol">84</prodml:AvgMolecularWeight>
    <prodml:AvgDensity uom="g/cm3">0.685</prodml:AvgDensity>
  </prodml:PseudoFluidComponent>
  <prodml:PseudoFluidComponent uid="c7">
  <prodml:PlusFluidComponent uid="c20+">
    <prodml:Kind>c20+</prodml:Kind>
    <prodml:StartingCarbonNumber>20</prodml:StartingCarbonNumber>
    <prodml:AvgDensity uom="g/cm3">0.9004</prodml:AvgDensity>
    <prodml:AvgMolecularWeight uom="g/mol">325</prodml:AvgMolecularWeight>
    <prodml:Remark>only one plus fraction defined during laboratory analysis</prodml:Remark>
  </prodml:PlusFluidComponent>
</prodml:FluidComponentCatalog>
```

Figure 11–9. Fluid analysis—fluid component catalog showing various kinds.

Fluid sample integrity and preparation is described in the element of that name, as shown in **Figure 11**–10.

```
<prodml:SampleIntegrityAndPreparation uid="sampleIP1">
    <prodml:OpeningDate>1994-01-01</prodml:OpeningDate>
    <prodml:InitialVolume uom="cm3">400</prodml:InitialVolume>
    <prodml:OpeningPressure xsi:type="eml:AbsolutePressure">
        <eml:AbsolutePressure uom="psi">1000</eml:AbsolutePressure>
    </prodml:OpeningPressure>
    <prodml:OpeningTemperature uom="degF">217</prodml:OpeningTemperature>
    <prodml:SaturationPressure kind="bubble point" uom="psi">2620</prodml:SaturationPressure>
    <prodml:SaturationTemperature kind="bubble point" uom="degF">220</prodml:SaturationTemperature>
    <prodml:FreeWaterVolume uom="cm3">0</prodml:FreeWaterVolume>
    <prodml:WaterContentInHydrocarbon uom="%[mass]">0.1</prodml:WaterContentInHydrocarbon>
    <prodml:OpeningRemark>no issues noticed</prodml:OpeningRemark>
    <prodml:SampleRestoration>
        <prodml:Date>1994-01-01</prodml:Date>
        <prodml:RestorationDuration uom="h">120</prodml:RestorationDuration>
        <prodml:RestorationPressure xsi:type="eml:AbsolutePressure">
            <eml:AbsolutePressure uom="psi">4300</eml:AbsolutePressure>
        </prodml:RestorationPressure>
        <prodml:RestorationTemperature uom="degF">220</prodml:RestorationTemperature>
        <prodml:MixingMechanism>rocking with mixing balls</prodml:MixingMechanism>
        <prodml:Remark>restoration succcesful.</prodml:Remark>
    </prodml:SampleRestoration>
</prodml:SampleIntegrityAndPreparation>
```

Figure 11–10. Fluid analysis—sample integrity and preparation.

The use of multiple sample contaminant elements, one of which includes a reference to a sample of the contaminant itself, is shown in **Figure 11**–11. Note that this “sample of contaminant” refers to the water sample which was sub-sampled from Sample 2, becoming Sample 3 (see Section 11.1.4).

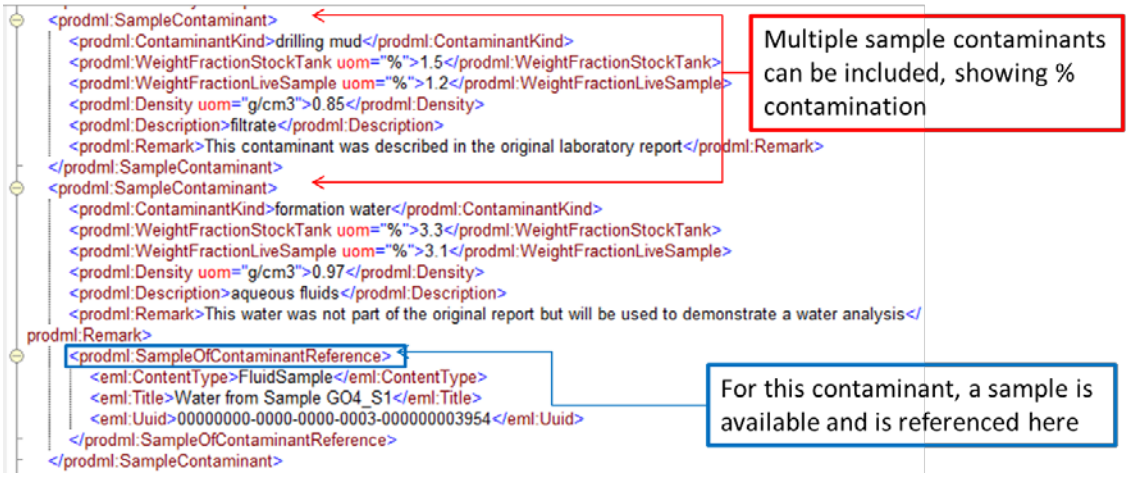


Figure 11–11. Fluid analysis—sample contaminant, showing reference to a sample of contaminant.

An example analysis test is shown in **Figure 11**–12. This example shows how the fluid components in the catalog are referenced using their UIDs. Each test also has a UID and this can be used to reference the test from elsewhere. This will be used in the fluid characterization to identify the tests that were used in a given characterization.

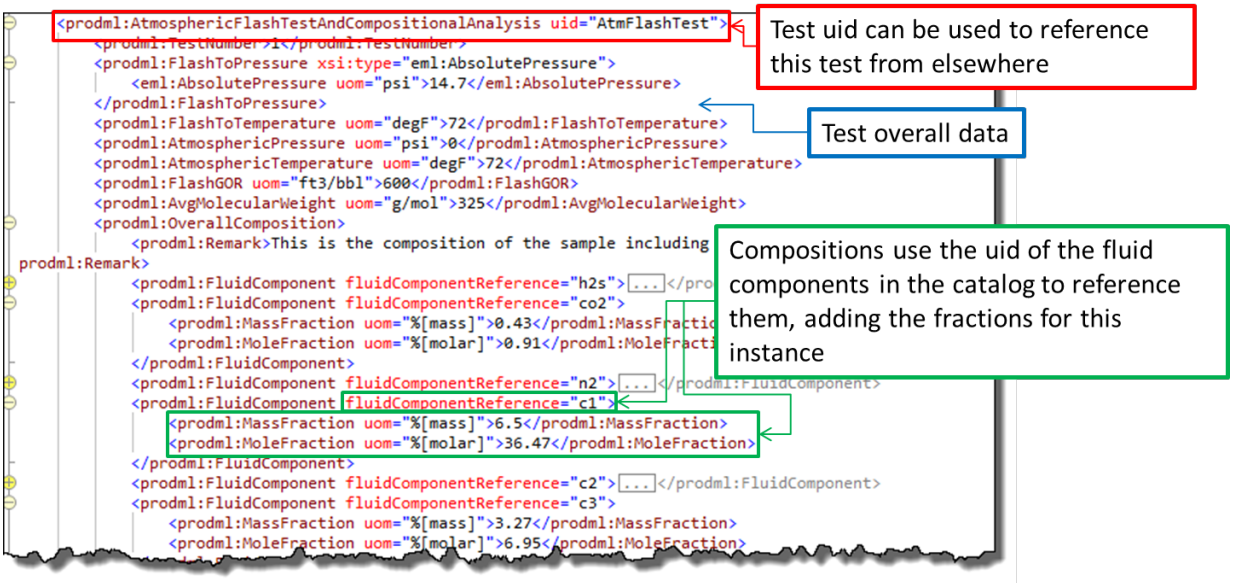


Figure 11–12. Fluid analysis—use of fluid components previously defined in fluid component catalog.

As shown in **Figure 11**–13, many tests have a pattern of common data followed by recurring test steps. This example figure shows the constant composition expansion test. In the worked example, the separator tests have multiple test steps, some of which report fluid composition of multiple phases, so combining aspects of the examples detailed here.

```
<prodml:ConstantCompositionExpansionTest>
  <prodml:TestNumber>2</prodml:TestNumber>
  <prodml:TestTemperature uom="degF">220</prodml:TestTemperature>
  <prodml:SaturationPressure kind="bubble point" uom="psi">2620</prodml:SaturationPressure>
  <prodml:RelativeVolumeReference uid="MSAT">
    <prodml:Kind>saturation-measured</prodml:Kind>
    <prodml:ReferenceVolume uom="mL">10</prodml:ReferenceVolume>
  </prodml:RelativeVolumeReference>
  <prodml:ConstantCompositionExpansionTestStep>
    <prodml:StepNumber>1</prodml:StepNumber>
    <prodml:StepPressure uom="psi">5000</prodml:StepPressure>
    <prodml:OilDensity uom="g/cm3">0.6763</prodml:OilDensity>
    <prodml:RelativeVolumeRatio fluidVolumeReference="MSAT" uom="bbl/bbl">0.9702</prodml:RelativeVolumeRatio>
    <prodml:PhasesPresent>oil</prodml:PhasesPresent>
  </prodml:ConstantCompositionExpansionTestStep>
  <prodml:ConstantCompositionExpansionTestStep>
    <prodml:StepNumber>2</prodml:StepNumber>
    <prodml:StepPressure uom="psi">4500</prodml:StepPressure>
    <prodml:OilDensity uom="g/cm3">0.6808</prodml:OilDensity>
    <prodml:OilCompressibility kind="average" uom="1/psi">0.000037</prodml:OilCompressibility>
    <prodml:RelativeVolumeRatio fluidVolumeReference="MSAT" uom="bbl/bbl">0.9639</prodml:RelativeVolumeRatio>
    <prodml:PhasesPresent>oil</prodml:PhasesPresent>
  </prodml:ConstantCompositionExpansionTestStep>
  <prodml:ConstantCompositionExpansionTestStep>
  <prodml:ConstantCompositionExpansionTestStep>
```

Figure 11–13. Fluid analysis—many tests have a pattern of common data followed by recurring test steps.

11.1.5.2 Water Analysis

Water analysis is a type of fluid analysis object (it is either hydrocarbon or water analysis type) (**Figure 11**–14). There is only one kind of water analysis test, and it can contain multiple test steps.

```
<prodml:FluidAnalysis xsi:type="prodml:WaterAnalysis" schemaVersion="0.29" uuid="
00000000-0AB1-A111-0001-000000003955" xmlns:eml="http://www.energistics.org/energyml/data/commonv2" xmlns:prodml="
http://www.energistics.org/energyml/data/prodmlv2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.energistics.org/energyml/data/prodmlv2 ../../../../xsd_schemas/FluidAnalysis.xsd">
  <eml:Aliases authority="PVT lab Co">
  <eml:Citation>
  <prodml:RequestDate>1994-04-01</prodml:RequestDate>
  <prodml:AnalysisDescription>Reservoir Study for Good Oil Company Well No. 4</prodml:AnalysisDescription>
  <prodml:AnalysisPurpose>Fluid Characterization</prodml:AnalysisPurpose>
  <prodml:AnalysisSite>PVT lab Co (Irving Texas)</prodml:AnalysisSite>
  <prodml:LabContact>Laman Reseflana</prodml:LabContact>
  <prodml:StandardConditions>
  <prodml:AnalysisQuality>valid</prodml:AnalysisQuality>
  <prodml:Remark>Lab's statement about their analysis</prodml:Remark>
  <prodml:FluidSampleReference>
  <prodml:FluidAnalysisReport>
  <prodml:WaterAnalysisTest>
    <prodml:LiquidGravity>1.02</prodml:LiquidGravity>
    <prodml:Salinity uom="ppm">80000</prodml:Salinity>
    <prodml:PH>6.9</prodml:PH>
    <prodml:Resistivity uom="ohm.m">0.0421</prodml:Resistivity>
    <prodml:WaterAnalysisTestStep>
      <prodml:StepPressure uom="psi">4000</prodml:StepPressure>
      <prodml:StepTemperature uom="degF">217</prodml:StepTemperature>
      <prodml:WaterFormationVolumeFactor uom="bbl/bbl">1.0372</prodml:WaterFormationVolumeFactor>
      <prodml:WaterIsothermalCompressiblity uom="1/psi">0.00000268</prodml:WaterIsothermalCompressiblity>
      <prodml:WaterDensity uom="lbm/ft3">63.612</prodml:WaterDensity>
      <prodml:WaterViscosity uom="cP">0.4251</prodml:WaterViscosity>
    </prodml:WaterAnalysisTestStep>
    <prodml:WaterAnalysisTestStep>
    <prodml:WaterAnalysisTestStep>
    <prodml:WaterAnalysisTestStep>
    <prodml:WaterAnalysisTestStep>
  </prodml:WaterAnalysisTest>
</prodml:FluidAnalysis>
```

Fluid analysis object type is water analysis

There is just one type of water analysis test

Multiple test steps can be reported

Figure 11–14. Fluid analysis—water analysis test.

11.1.6 Fluid Characterization

There are two ways in which a fluid can be characterized. Both can be mixed in the same fluid characterization data object. Here they are shown in two worked examples.

1. Fluid characterization using models (file: Good Oil 4.ModelCharacterization).
2. Fluid characterization using tables (file: Good Oil 4.TabularCharacterization).

The high level content of the model approach is shown in **Figure 11**–15. The tabular approach follows later and shares the same high-level data elements, but adds tabular data in place of (or in addition to) model data.

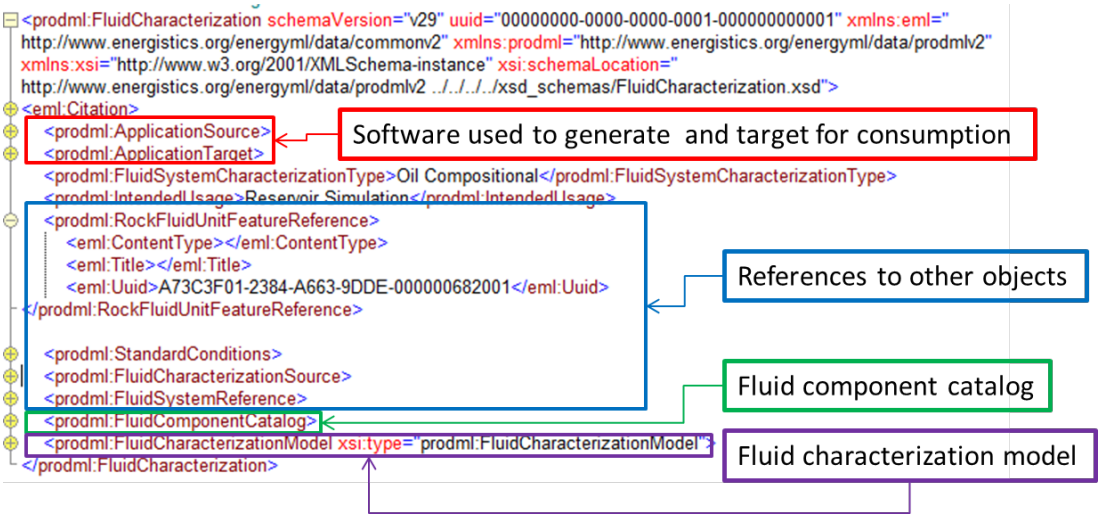


Figure 11–15. Fluid characterization—model, high-level contents.

The software used to generate the characterization and its intended target software for consumption of the data can be reported (**Figure 11**–16).

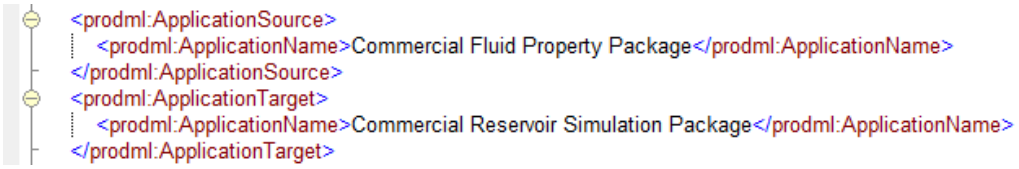


Figure 11–16. Software used to generate and target for consumption.

The source of the characterization means which fluid analysis tests results were used in this fluid characterization. This is done by referencing the UID of the tests concerned, with a reference to the parent fluid analysis data object, which can be seen in **Figure 11**–17**.**

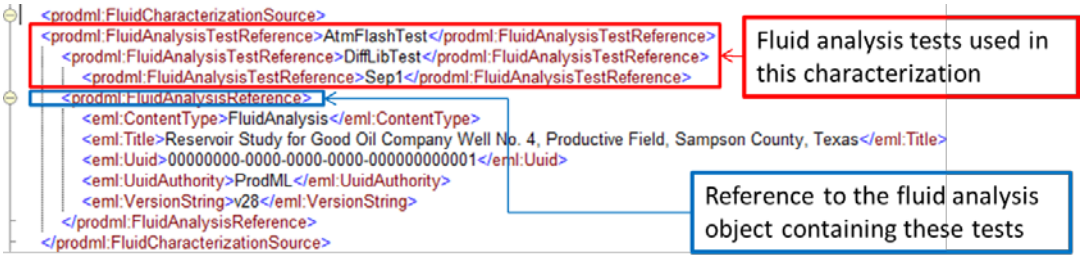


Figure 11–17. Fluid characterization source.

Standard conditions can be reported. Also possible, but not shown in the example, separation conditions (one or more stages plus stock tank conditions) can be reported, these being the conditions at which the fluid is characterized.

The fluid component catalog works the same way as for the fluid analysis example (see Section 11.1.5.1). The worked example does not use the same catalog as the analysis, instead having one with fewer components as would typically be used in software models.

The model specification contains the different kinds of parameter appropriate to the model. **Figure 11**–18 shows the worked example. Note that the model type (seen at the top, red box) is one of the models listed in Section 12.2. Each of these model types has its own set of parameters which the schema enforces. See Figure 10–10 and Figure 10–11 for the concept behind this. Details are in the schema.

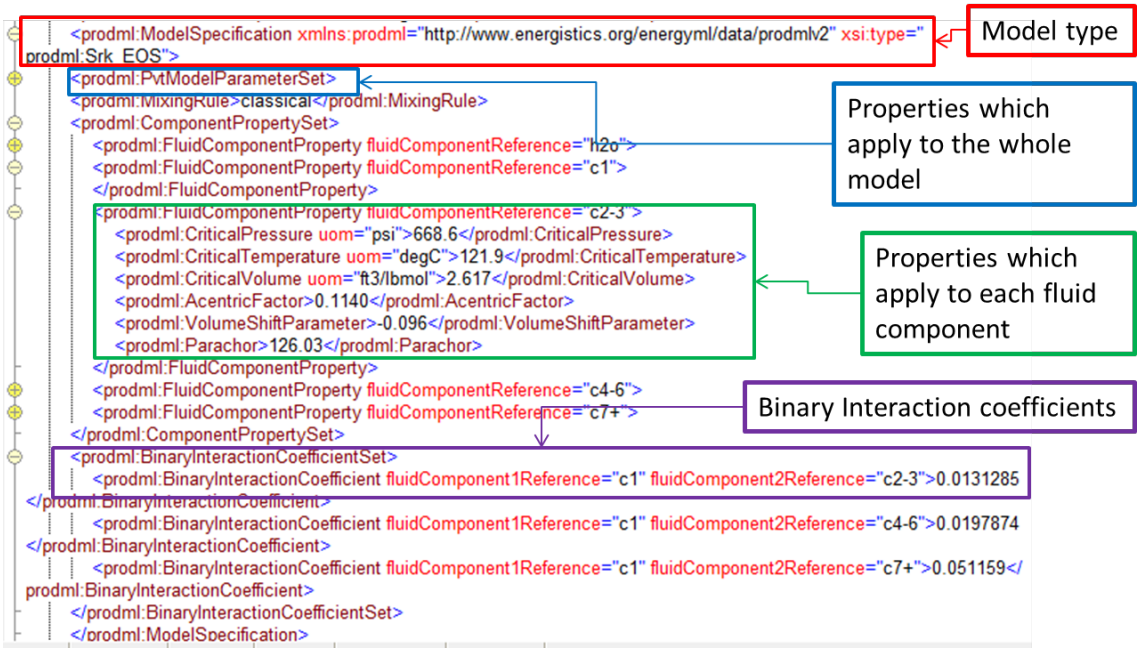


Figure 11–18. Fluid characterization – model specification.

The tabular fluid characterization uses the following two main elements:

1. One or more fluid characterization tables. These have table constants (per table). The data is then transferred in table rows within each table.
2. One or more fluid characterization table formats. This contains table column headings (representing what the contents of a column are).

The UID of the fluid characterization table format is used in the fluid characterization table to show which format applies to the table (**Figure 11**–19).

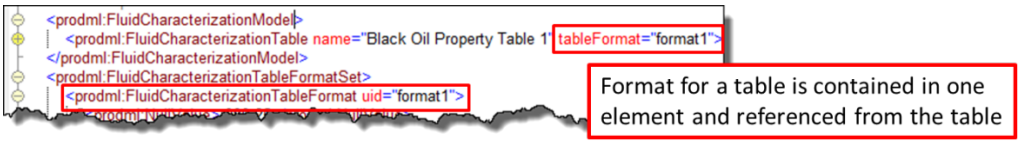


Figure 11–19. Table and table format.

The table format defines the columns that appear in the table. It also defines the column delimiter asci character (e.g., a comma), and the null value (e.g., -999.25). The attributes of each column are as follows.

Attribute	Description	Optionality
Name	User-defined name for this attribute	optional
Sequence	Integer to allow this column attribute to be indexed	optional
Property	The physical property that this value represents	mandatory
Uom	String to represent the unit of measure for the property values	mandatory

Attribute	Description	Optionality
fluidComponentReference	Reference to the fluid component to which this value relates	optional
Phase	Reference to the phase to which this value relates	optional
KeywordAlias	Used to apply a consumer product keyword to this value, with "authority" being the product name concerned	Optional (zero to many)

Property is an enumeration that can be extended and the properties are listed in Section 12.3. Phase is also an enumeration and is listed in Section 12.3. Phase and fluid component reference are provided so that properties may be output that relate to a specific phase or fluid component. Unit of measure is not a controlled list. Keyword alias is provided in case it is useful to map properties onto keywords in a specific software package.

An example extract from the worked example fluid characterization table format is shown in **Figure 11**–20.

```
<prodml:FluidCharacterizationTableFormat uid="format1">
  <prodml:NullValue>-999.25</prodml:NullValue>
  <prodml:Delimited asciiCharacters="10,35"/>
  <!-- the delimiters in this example are: [tab],[#] -->
  <prodml:TableColumn sequence="1"  name="Pressure" uom="psi">
    <prodml:Property xsi:type="prodml:OutputFluidPropertyExt">Pressure</prodml:Property>
  </prodml:TableColumn>
  <prodml:TableColumn sequence="2"   name="Bubble Point Pressure" fluidComponentReference="oil" uom="psi">
    <prodml:Property xsi:type="prodml:OutputFluidPropertyExt">Saturation Pressure</prodml:Property>
  </prodml:TableColumn>
  <prodml:TableColumn sequence="3"  name="Solution GOR" uom="SCF/STB">
    <prodml:Property xsi:type="prodml:OutputFluidPropertyExt">Oil-Gas Ratio</prodml:Property>
```

Figure 11–20. Table format.

The fluid characterization table then contains the actual values defined by the format. These are contained in table row elements where the values are separated by the ASCII character defined above.

In addition to the table rows, the table also has table constants. These have the same attributes as listed above for table columns except that sequence is not present (tables do not have a sequence). A value attribute is added and is mandatory.

The table has a UID so that it can be referenced from elsewhere, and a name.

An example table is shown in **Figure 11**–21.

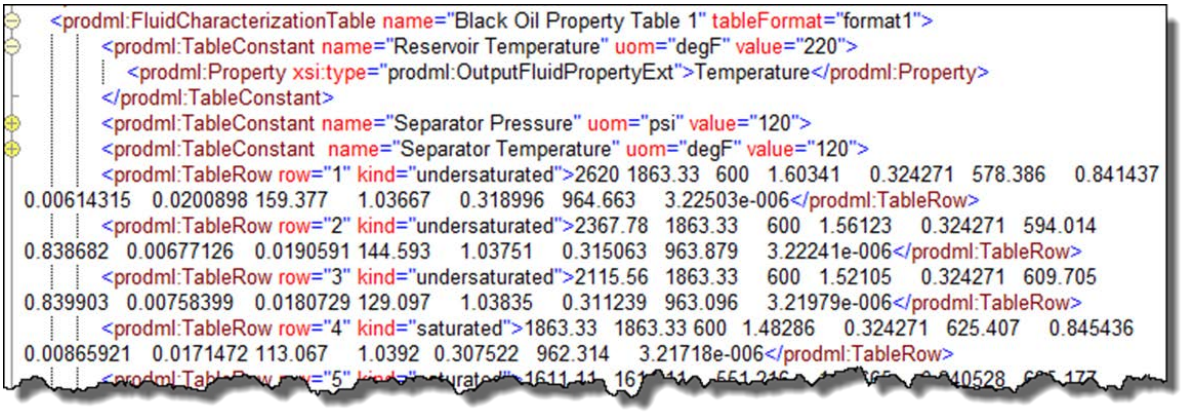


Figure 11–21. Fluid characterization table.

The table example contains some other illustrations, e.g. an extension to the enum list of properties, which can be seen by inspecting the file.

11.2 Other examples

Three other worked examples of lab analyses are included. These can be found in the Other Examples folder. Only the fluid analysis objects are included.

1. Volatile Oil
2. Gas Condensate
3. Oil Plus Swelling Tests

Data in the form of Excel files for the first two of these are provided.

12 Appendix for Fluid and PVT Analysis

This chapter contains appendix information for the previous main fluid and PVT analysis chapters.

12.1 Fluid Components Defined in the Model

12.1.1 Pure Fluid Components

The table below shows the enumeration of pure fluid components. The enumeration is extensible so that additional molecule types can be added.

Aliphatic	Cyclic and Aromatic	Inorganic
c1	methylcyclopentane	ar
c2	cyclohexane	co2
c3	methylcyclohexane	h2
n-c4	ethylcyclohexane	h2s
i-c4	benzene	he
n-c5	toluene	hg
i-c5	m-xylene	n2
neo-c5	o-xylene	
n-c6	p-xylene	
2-methylpentane	ethylbenzene	
3-methylpentane	1-2-4-trimethylbenzene	
2-dimethylbutane		
3-dimethylbutane		
n-c7		
2-methylhexane		
3-methylhexane		
n-c8		
2-methylheptane		
3-methylheptane		
n-c9		
n-c10		

12.1.2 Pseudo Fluid Components

The table below shows the enumeration of pseudo fluid components. The enumeration is extensible so that additional pseudo components can be added. Additional parameters are available with which to characterize all pseudo components.

Range of carbon numbers	Single carbon numbers	Single carbon numbers (cont.)
c2-c4+n2	c4	c20
	c5	c21
	c6	c22

Range of carbon numbers	Single carbon numbers	Single carbon numbers (cont.)
	c7	c23
	c8	c24
	c9	c25
	c10	c26
	c11	c27
	c12	c28
	c13	c29
	c14	c30
	c15	c31
	c16	c32
	c17	c33
	c18	c34
	c19	c35

12.1.3 Plus Fluid Components

The table below shows the enumeration of plus fluid components. The enumeration is extensible so that additional plus components can be added. Additional parameters are available with which to characterize all plus components.

Plus Carbon Number	Plus Carbon Number (cont.)	Plus Carbon Number (cont.)
c5+	c9+	c20+
c6+	c10+	c25+
c7+	c11+	c30+
c8+	c12+	c36+

12.2 PVT Models

12.2.1 Correlation PVT Models – Viscosity

For dead oil, bubble point and under-saturated conditions:

- Bergan-Sutton
- DeGhetto
- Dindoruk-Christman
- Petrosky-Farshad
- Standing

12.2.2 Compositional EoS

- Peng-Robinson 76
- Peng-Robinson 78
- SRK

12.2.3 Compositional – Viscosity

- C S Pedersen 84
- C S Pedersen 87
- Lohrenz-Bray-Clark
- Friction Theory

12.2.4 Compositional – Thermal

Currently, no specific models defined so a generic model together with custom model extensions can be used.

12.3 Fluid Properties that can be Output to Tables

The table below shows enumeration of fluid properties which can be output to tables. The enumeration is extensible so that additional fluid properties can be added.

Also listed are the enumeration values for phase.

Output Fluid Property	Phase
Compressibility	aqueous
Density	oleic
Derivative of Density w.r.t Pressure	total hydrocarbon
Derivative of Density w.r.t Temperature	vapor
Enthalpy	
Entropy	
Expansion Factor	
Formation Volume Factor	
Gas-Oil Interfacial Tension	
Gas-Water Interfacial Tension	
Index	
K value	
Misc Bank Critical Solvent Saturation	
Misc Bank Phase Density	
Misc Bank Phase Viscosity	
Miscibility Parameter (Alpha)	
Mixing Parameter Oil-Gas	
Oil-Gas Ratio	
Oil-Water Interfacial Tension	
Parachor	
Pressure	
P-T Cross Term	
Saturation Pressure	
Solution GOR	
Solvent Density	
Specific Heat,	
Temperature	

Output Fluid Property	Phase
Thermal Conductivity	
Viscosity	
Viscosity Compressibility	
Water vapor mass fraction in gas phase	
Z Factor	

Part IV: Distributed Temperature Sensing (DTS)

Part IV contains Chapters 13 through 16 which explain the set of PRODML data objects for distributed temperature sensing (DTS).

Acknowledgement

Thanks are due to the members of the PRODML SIG who joined the effort to develop the revised DTS data object.

In particular, the efforts donated by members from the following companies must be acknowledged: Shell, Chevron, AP Sensing, Perfomix, Schlumberger, Sristy Technologies, Tendeka, Teradata, and Weatherford. The provision of definitions of terminology by the SEAFOM Joint Industry Forum is also gratefully acknowledged.

13 Introduction to DTS

This section describes the data model in PRODML to cover the most common business scenarios and workflows related to DTS as identified by the industry at large.

13.1 Overview of DTS

Distributed Temperature Sensing (DTS) is a technology where one sensor can collect temperature data that is spatially distributed over many thousands of individual measurement points throughout a facility. DTS requires that the facilities being monitored—for example, wellbores or pipelines—are fitted with fiber optic cable for gathering temperature data along the entire length, instead of using individual gauges.

While still a relatively young technology in the oil and gas industry, DTS and other fiber technologies are offering a lot of promise for improved temperature monitoring and for quickly detecting production-related problems. Research is showing that, in addition to temperature sensing, fiber optic technology can also be used to detect sound, pressure, flow, and fluid composition. For more information about DTS, see Chapter 16.

13.2 The Business Case

The applications of DTS data in the oil & gas industry are growing rapidly. There has been an emergence of business workflows that depend on DTS data in order to make timely decisions. This is a radical change from the way DTS data was handled in the past, where a limited audience consumed the data.

In addition to the changes observed in the oil & gas industry there has been an increased application of DTS technology to other areas as well. It is important that the new standard can accommodate other scenarios and it could be extended beyond upstream oil and gas to other industries if required.

13.2.1 Business Drivers and Benefits

By having a common data schema that provides comprehensive coverage of DTS data and all its potential uses we provide a mechanism by which different vendors and operating companies can integrate solutions seamlessly.

The main goal for this PRODML capability is to ensure that all the business requirements outlined by the participating oil companies and oil services companies are covered. This in turn will allow different companies create a number of products (for data transport, storage, management, analysis, etc.) that can integrate seamlessly, allowing mix-and-match hardware and software.

13.2.2 Challenges with DTS Data

Extensive use of the previous PRODML schema for DTS revealed the following areas requiring enhancements, which were first made in version 1.3.

- Data schema provided definitions for describing a fiber installation but more detailed information would be desired.
- A clearer distinction was needed between ‘raw trace’ measurements obtained from the fiber versus derived ‘log-like’ curve data that is eventually used for interpretation and analysis.
- Provision of a mechanism to label data with ‘versions’ was needed, in order to accommodate diverse business workflows that involve the use of DTS data.
- It would be useful to include placeholders for annotations and contextual data to aid subject matter experts troubleshoot any issues related to the DTS measurements.

13.3 Scope and Use Cases

Sample use cases are listed and documented to illustrate the level of coverage of DTS in PRODML and to provide a few examples of how the data model could be used. From these examples it will be easy to extrapolate additional use cases that will apply to more specific situations. The use cases documented below focus on these areas:

- Describing a physical fiber installation, where we will highlight the different possibilities to accommodate multiple deployment scenarios that have occurred in real life. The option to capture how a physical installation has changed over time is also covered by this data schema.
- Capturing measurements from DTS instrumentation so they can be transferred from the instrument to other locations and ultimately be stored in a repository. This includes the 'raw' measurements obtained by the light box as well as 'derived' temperature log curves.
- Manipulation of DTS-derived temperature log curves (depth adjustments, for example) while maintaining a record of all the changes performed. The PRODML schema supports multiple forms of data manipulation and versioning so that most business workflows surrounding DTS can also be represented.

For detailed explanation of these use cases and related key concepts, see Chapter 14.

14 Use Cases and Key Concepts

This chapter provides an overview of use cases addressed with this data object, related key concepts, and data object organization.

14.1 Use Cases

The DTS data objects have been designed to address these primary use cases:

1. Represent an optical fiber installation;
2. Capture DTS measurements for transport and storage;
3. Manipulate DTS-derived temperature log curves.

Each is outlined in the following sections.

14.1.1 Use Case 1: Represent an Optical Fiber Installation

The Optical Path data object has been designed to address the following scenarios:

- **Installation of an optical path** that consists of multiple fiber segments joined via splices, connections and turnarounds. Optical fibers can adopt multiple configurations such as:
  - Straight fiber with a termination at the end.
  - 'J' configuration.
  - Dual-ended fiber that terminates back in the instrument box.

For diagrams of these and other geometries, see Figure 16–4.

Optical fiber installation is not restricted to wells. The data schema has been designed to accommodate other deployment scenarios where optical fiber could also be applied, such as pipelines.

Further note that the optical fiber installations are used in a similar manner for distributed acoustic and strain sensing (DAS and DSS) applications. For DAS PRODML (Part V. Distributed Acoustic Sensing) the same definition of installation of optical paths apply,

The data schema is extremely detailed, allowing great flexibility and fine granularity at the same time. It is possible easily to document things such as:

- Location of splices, and their type.
- Signal loss and reflectivity properties on a per-fiber-segment basis.
- Overstuffing (whereby the length of fiber is greater than that of the physical facility being measured).
- Type of material used in the optical fiber.
- Conveyance of the fiber, e.g. in a control line, in a permanent cable, deployed in a wireline logging mode, etc.
- Map the length along the optical path to specific **facility lengths**, so that the analyst knows which parts of a measurement pertain to the wellbore, pipeline etc. which they are analyzing.
- Denote locations in the fiber where **fiber defects** exist so that future troubleshooting of the measurements can take into account the presence of these defects.
- Store calibrations of the Instrument Box or whole system.
- Store **OTDR** (Optical Time Domain Reflectometry) a type of diagnostic test on the optical path) information, including the type of equipment and personnel used for taking the OTDR.
- Represent the **DTS Instrumentation Box** that has been installed, either permanently connected or as a temporary installation, to the optical fiber. Several details regarding the instrumentation can be represented through the data schema, ranging from make/model of the box, contact information on the person who installed it, configuration data, and calibration data, to diagnostics information. The DTS Instrument Box is sometimes known as a "*Lightbox*".

- Represent **DTS Installation** comprising one Optical Path and one Instrument Box. This is the “unit” which generates measurements. Various configurations can be represented this way, such as an Instrument Box that will be shared among multiple optical fibers in one or multiple physical locations (a “drive-by” instrument box).

14.1.2 Use Case 2: Capture DTS Measurements for Transport and Storage

The DTS data object clearly differentiates between the ‘raw’ measurements obtained by the instrument box (such as **stokes** and **anti-stokes** curves) and the final, derived, temperature value along the fiber that is recorded in the form of a temperature log curve for easier loading into different visualization and analysis tools.

Each measurement set has associated with it the DTS Installation which created it, so that traceability can be assured.

The data schema does not impose requirements as to what measurement curves are required so that each different installation can make use of the appropriate curves. However, the family of curves which can be used is limited to a set agreed by a group of major DTS suppliers. One family of curve names covers both Rayleigh and Brillouin methods of DTS measurement, with absent channels being omitted.

The structure used for representing the measurements was chosen in order to maintain a balance between the size of the resulting XML file and compatibility with current Energistics XML representations and libraries.

14.1.3 Use Case 3: Manipulation of DTS-derived Temperature Log Curves

A very important part of DTS data usage is the support for diverse business workflows that not only uses DTS data for analysis but also performs different data transformations to it for better supporting decision-making activities.

When modifying DTS data it is critical that one can always trace the origins of any transformation. The DTS Data object provides all the necessary fields so that any modifications done to the data (such as depth shift) can be transferred maintaining its association with the original reading obtained from the DTS Installed System. The data allows representation of scenarios where a single measurement from a DTS Installed System has undergone different transformations for various reasons, keeping all the transformations (by versioning) and having all those transformations reference the original measurement for full traceability.

In addition to data versioning the data object offers a number of flags that can be used to denote attributes such as:

- Whether the measurement is ‘bad’ or not.
- Use of keyword “tags” for easier search and retrieval.
- When having multiple versions of interpretations derived from one measurement, have a flag that shows which interpretation is the ‘approved’ one for business decisions
- Whether the measurement is empty or not, allowing tracking of how often the instrumentation generates readings where those readings are completely empty (because the fiber is disconnected, for example)

There are also placeholders for storing diagnostics information from the instrument box that can be used for troubleshooting any issues that may be found with the measurement itself.

14.2 Key Data Model Concepts

This section presents an overview of the DTS data object model, and then goes through each major element and concept in turn.

14.2.1 Data Model Overview

Figure 14–1 is a simple diagram of the data model for a DTS implementation. The colored boxes represent the minimum set of objects in the DTS data object that are needed to represent any DTS

deployment. The meaning and usage of the colored boxes is explained in the next section of the document.

Each of these boxes is covered by a top-level object in the data model:

- Fiber Optical path
- DTS Instrument box
- DTS Installed system
- DTS Measurement

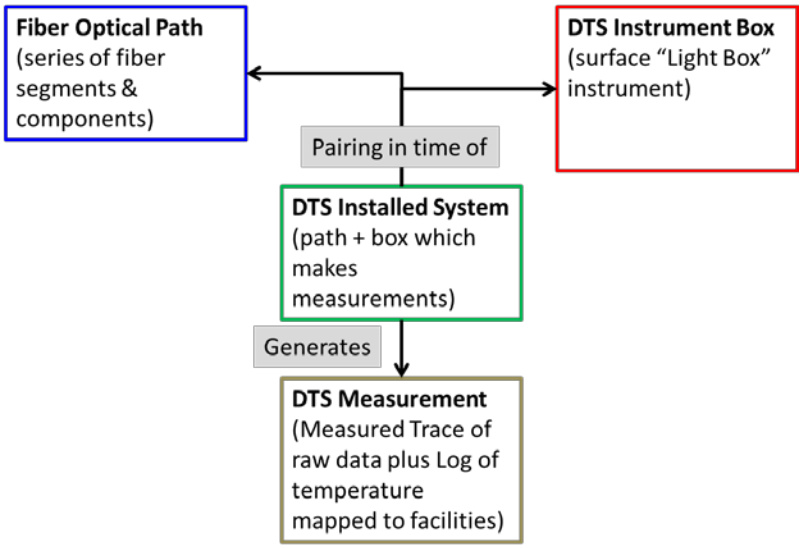


Figure 14–1. Overall representation of a DTS installation and measurements.

14.3 Defining the Optical Path

The fiber optical path is used by both the DTS and the DAS sub-domains of PRODML. This section 14.3 is therefore common to both.

For the UML diagrams of the Fiber Optical Path, see the UML model (the XMI file included with PRODML download).

14.3.1 Fiber Optical Path

An optical path is a set of continuous optical waveguides that acts as a linear sensor, used to record temperature or acoustic or dynamic strain events along its length, as shown for example in **Figure 14–2.** It can be composed of one or more optical path components and has one termination. An optical path component could be a fiber segment, a connector, a splice, or a turnaround. See Section 16.3 for some example physical fiber installations in wellbores which may make up the Optical Path.

In the PRODML distributed data objects, the optical path is represented by a Fiber Optical Path top-level object. This object contains one optical path object that represents both the collection of components used along the path and the connection network.

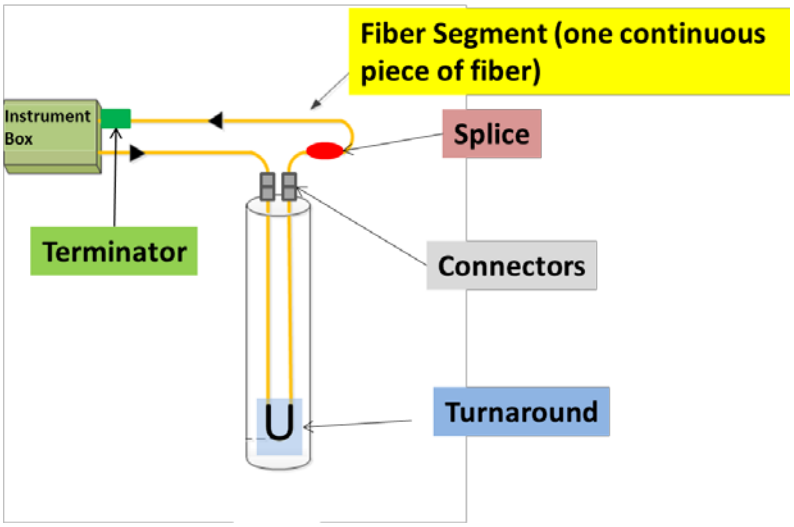


Figure 14–2. Example of a dual-ended optical path installation, showing different types of components.

In this example, we have one optical path that is deployed in a well in a dual-ended configuration.

To support the requirement to be able to track the changes in the path over time, the optical path is represented using the inventory and network pattern explained in the following sections.

- Optical Path **Inventory**. This is a list of *all* the components used in the Optical Path over the whole time being reported. (For more information, see Section 14.3.2 (below).)
- Optical Path **Network**. This is a representation of the connectivity of a set of components at a given time. (For more information, see Section 14.3.3 (page 120).)

14.3.2 Optical Path Inventory Representation

The optical path in **Figure 14–2** is composed of the following components, which are referred to as the Inventory of components in the optical path.

- 5 fiber segments, one from each of these locations:
  - the instrument box to the first connector (at the wellhead)
  - the first connector to the turnaround at the bottom of the well
  - the turnaround to the second connector
  - the second connector to the splice
  - the splice back to the terminator at the instrument box
- 2 connectors:
  - 1 splice
  - 1 turnaround
- 1 terminator of type ‘looped back to instrument box’ to represent that the optical path ends back into the instrument box in a dual-ended configuration. (The alternative type of terminator is ‘termination at cable’, where the optical path is single ended).

It is necessary to declare only those components and segments that are of relevance to the particular application or well design. If there is no interest in keeping track of splices and connectors, you can easily declare this optical path to consist of 1 fiber segment that happens to be the entire length of the optical path. Of course, doing so means that there is no mapping of possible anomalies to connections, etc., and no way to record different physical properties of each fiber segment.

In addition, OTDR (optical time domain reflectometry) tests can be reported for the optical path.

14.3.3 Optical Path Network Representation

The optical path network is represented by a PRODML Product Flow Model (a standard object for networks in PRODML) (**Figure 14**–3). This model uses the concept of units, which are "black boxes" that have ports, representing connection points. Each component in the inventory that is part of the optical path is represented by a unit. Connectivity is represented by nodes, which are virtual objects representing the connection between two ports.

Generally along the optical path, all the components have linear connectivity so units have 2 ports. One benefit of this approach is standardization with other network representations, e.g., the flow network of PRODML.

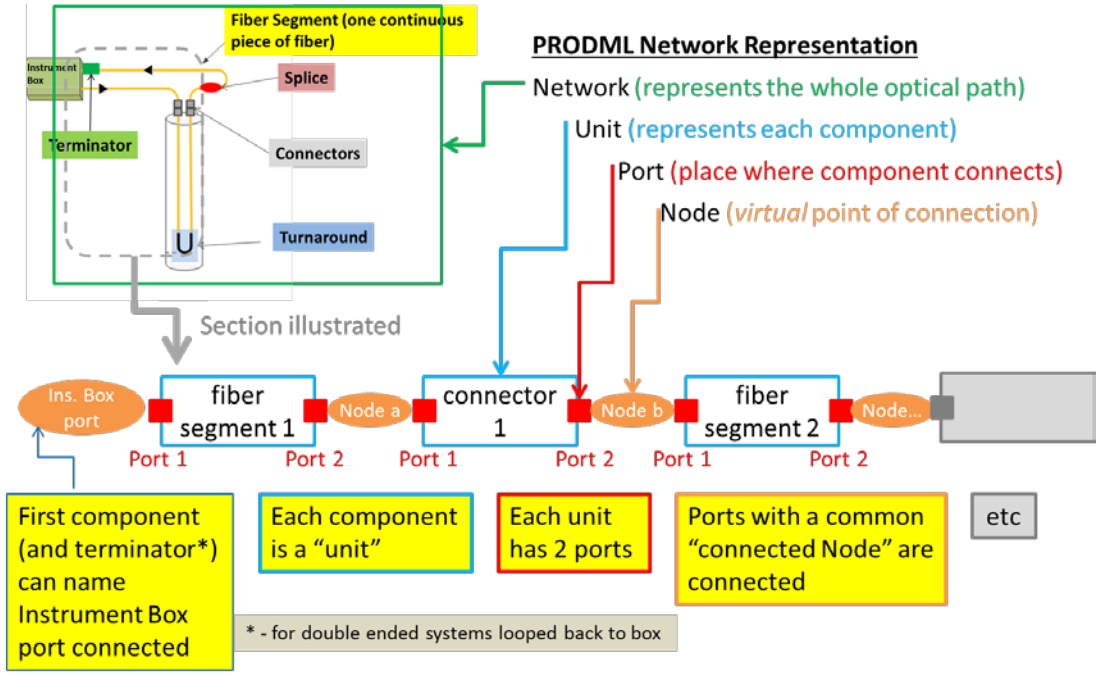


Figure 14–3. Representation of the optical path components in a PRODML Product Flow Model (network).

For the UML model that represents this network, see **Figure 14**–4. This diagram also shows the mandatory elements and notes as to which convention to follow.

Note, the connectedNode element for two ports on different units indicates these two units are connected. Example, "Port 2" of "fiber segment 1" and "Port 1" of "connector 1" both show connectedNode as "Node a". The first component can reference a specific named port on the instrument box if required to report this, and the same applies to the port on the terminator if "looped back to instrument box". This way the instrument box connections are reported.

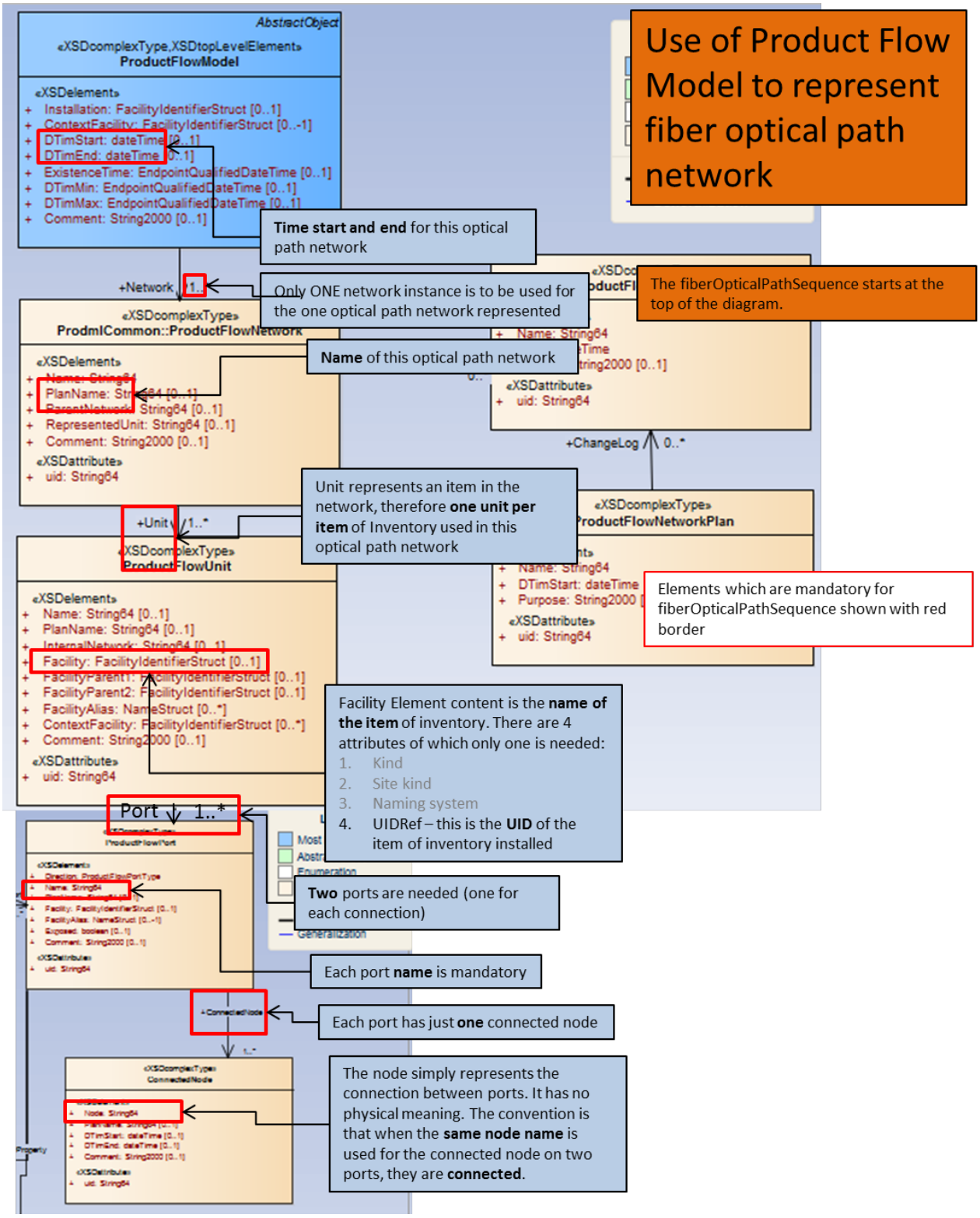


Figure 14–4. Product Flow Model which represents a network, showing mandatory elements (red border) and including explanatory notes (blue boxes).

14.3.4 Facility Mappings

The purpose of the Facility Mappings element within the Fiber Optical Path object is to represent the relationship between an optical path and the facility(s) (sometimes called "*assets*") where the optical fiber has been deployed. For locations where the fiber is used in one facility only, such as a well, the relationship between the facility and the fiber (optical path) is naturally 1 to 1.

However if a fiber (optical path) is long enough that it can cover multiple assets (for example, 2 wells and 1 pipeline, as shown in **Figure 14**–5) then you must represent that relationship between that one optical path and the 2 wells and the pipeline.

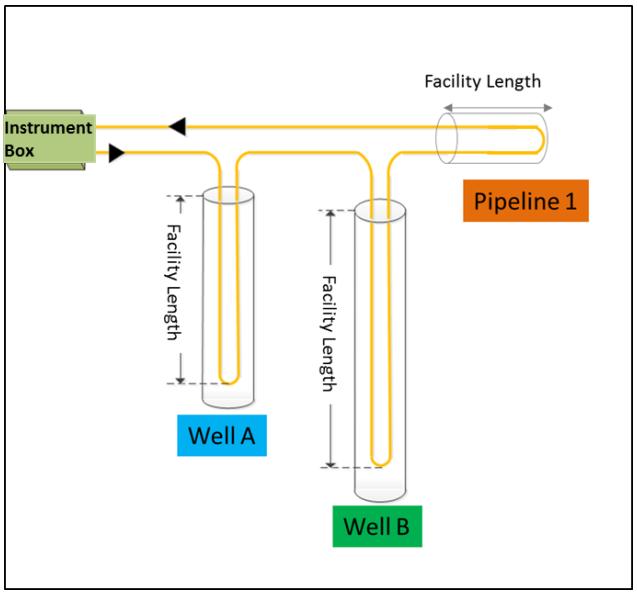


Figure 14–5. Example of an optical path that spans more than one well (asset)

When a PRODML distributed data measurement is obtained, we must be able to map the "optical path length" of the measurement (which gives values along the measured length of the fiber) to the "facility length" of the measurement (which gives distributed values in reference to lengths or depths within the facility, for example in a well, as shown in Figure 14–5).

The Facility Mapping element provides that translation between the actual distance along the optical path as measured from the instrument box and the actual distance of the measurement locations as measured from a reference datum defined for the facility. See the example in **Figure 14**–6 wherein the mapping is shown as the grey "facility lengths," which are formed from certain "optical path lengths" along the total length of the optical path.

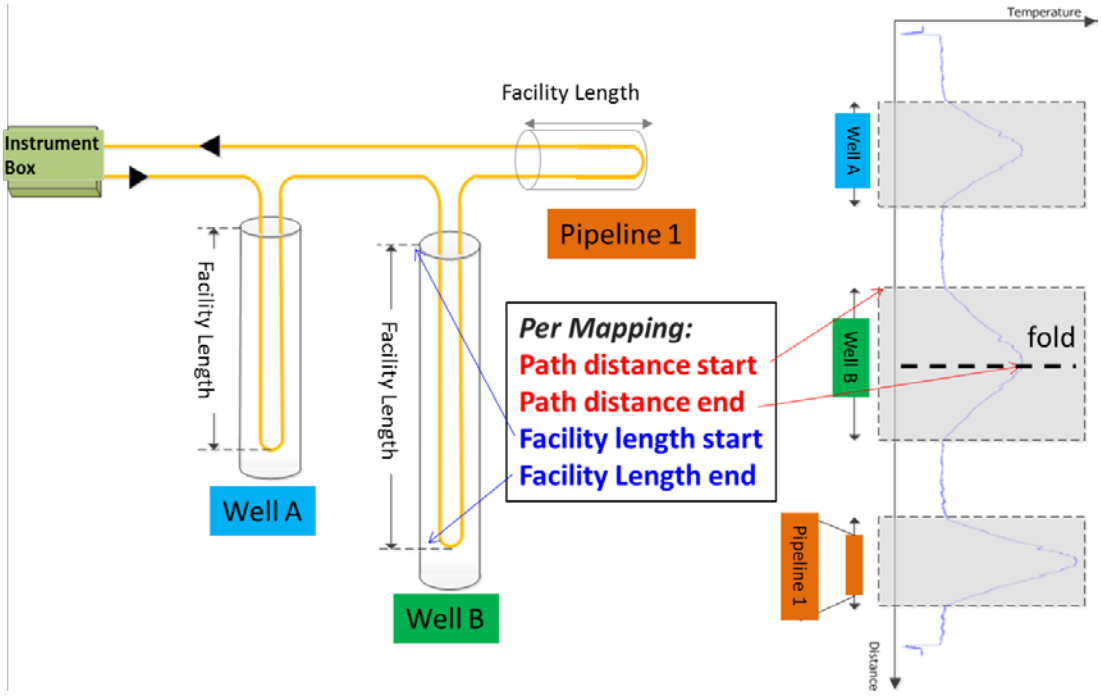


Figure 14–6. Mapping Optical Path Distance and Facility Distance.

With this arrangement, it is possible to receive a distributed measurement where all the “raw” values are indexed by the absolute distance along the optical path plus the mapping for measurements to be indexed against the distance along the facility.

14.3.5 Fiber Defects

The purpose of the Defects object is to facilitate troubleshooting and data analysis after the measurements have been collected. **Figure 14**–7 shows an example of a defect.

It is quite common to have areas in the optical path where the measurement obtained is different from what was obtained in surrounding areas. This difference in the measurement is not caused by a true change in the environment but rather by an artificial ‘artifact’ that is causing the light scatter to return abnormally. By documenting the areas where these anomalies are known to be, anybody analyzing these measurements can promptly discard the measurement changes as part of the physical installation rather than as a location of unexpected changes.

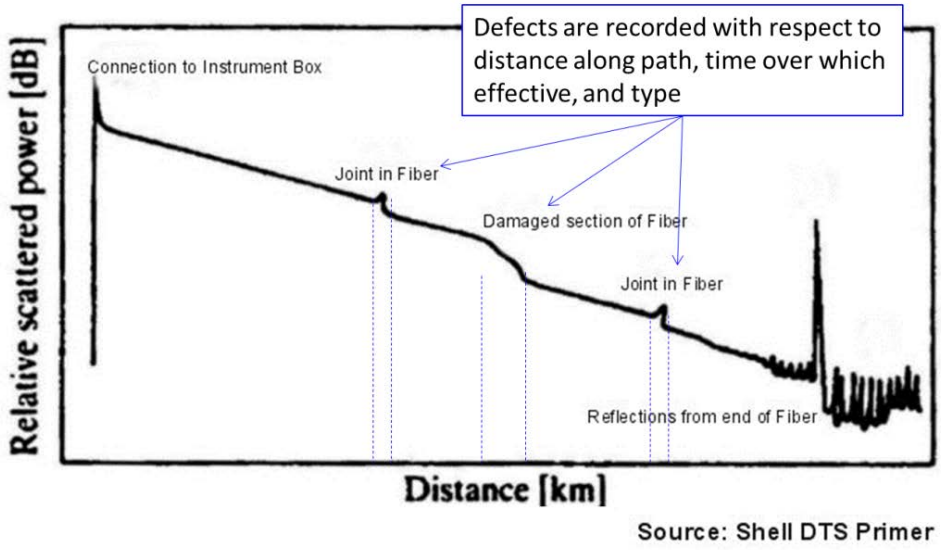


Figure 14–7. Examples of defects found on a distributed measurement trace.

14.3.6 Conveyance

Conveyance refers to the physical means by which fiber is installed in or carried into the facility. The categories of conveyance are listed in the following table:

Category of Conveyance	Description / Application
Permanent	Applies to permanent installations and contains details, such as fiber clamping or other method of attaching to tubulars. The fiber is encapsulated in a control line. However, it is not pumped down the control line, in contrast with the following method.
Fiber in Control Line	Applies when fiber is pumped into the facility inside a control line (small-diameter tube) and reports: • The size and type of control line. • Pumping details for operation in which fiber is pumped into control line.
Intervention	Applies to temporary installations (in a wellbore) and contains details such as the intervention method by which fiber is conveyed into the well (on wireline, coiled tubing, etc.).

14.3.7 Historical Information

Fiber Optical Path uses a common Energistics XML pattern that separates inventory of components from the implemented network configuration, as described in sections 14.3.2and 14.3.3.

Use of this pattern means it is possible to keep items in the optical path description that have been decommissioned, with the value that failure modes can be tracked.

For example, **Figure 14**–8 shows an optical path spanning two wells and one pipeline. At some point, it is assumed that the pipeline segment is decommissioned for some reason. This would give rise to:

- one set of inventory (including all the segments and components)
- two networks that include the pipeline and later another one without the pipeline components included

This way, distributed measurements through the whole life of the installed system can be associated with the right optical path and facilities.

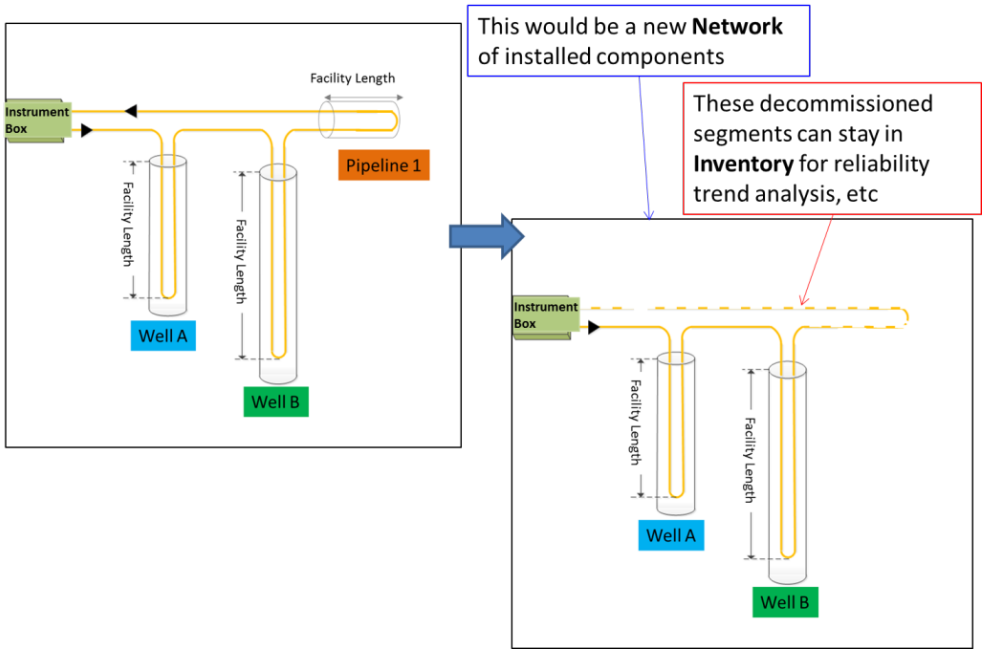


Figure 14–8. Keeping track of historical changes.

14.4 Defining the DTS Instrument Box

The DTS Instrument Box object represents all the hardware equipment located at the site which is responsible for generating DTS measurements. The DTS Instrument Box is usually located next to the facility element for which DTS surveys are taken (a well or a pipeline, for example). It consists of, among other devices, an optoelectronic instrument to which the optical fiber is attached. This contains a controllable light source, optical switches, photonic detection devices, and a reference temperature chamber. In some cases there is a computer or server connected to the instrument that will be responsible for capturing the measurements and initiating the data transmission.

Hardware setup can wildly vary from one deployment to the next, and will also depend on the vendor, make/model of the hardware used. The data schema was created to allow you to capture the relevant parameters of the installation without worrying about how the hardware is physically laid out. Attributes available for the instrumentation cover areas such as:

- Make/model of the instrument
- Number of channels
- Software Version
- Factory Calibration information
- Warmup and Startup times
- Oven location
- Reference temperature
- Calibration parameters

The PRODML data schema will also supports keeping track of inventory, including old equipment, as it is possible to specify decommissioning details to every instrument box.

14.5 Defining the Distributed Sensing Installed System

The installed system concept is used by both the DTS and the DAS sub-domains of PRODML. This Section 14.5 is therefore common to both.

14.5.1 Distributed Sensing Installed System Concept

NOTE: In this section, the DTS and DAS data models treat the Distributed Sensing Installed System slightly differently:

- DTS uses an intermediate object called DTS Installed System. This contains pointers (data object references) to the Optical Path and DTS Instrument Box data objects, which comprise the Installed System. Calibrations may also be attached to the Installed System. The DTS Measurement data object points to the DTS Installed System data object that generated the measurement.
- DAS does not use an intermediate data object; instead, a DAS Acquisition data object points to the Optical Path and DAS Instrument Box data objects that generated the acquired data.
- In both cases, the important point is: there is a 'pairing in time' of an optical path and the instrumentation.
  - For permanent deployments (where the instrument box is fixed and not reused with other optical paths), this pairing in time is permanent.
  - For a drive-by instrument box, an installed system is formed every time one instrument box is temporarily connected to an optical path in order to obtain measurements. As the instrument box is moved from one location to the next, it employs different calibration settings and therefore constitutes a different installation.

14.5.2 Distributed Sensing Installed System Examples

The installed system concept is highly relevant, because all distributed measurements are directly associated with one installed system, and through that installed system the data consumer determines the instrument box and optical path for which the measurement was taken. In **Figure 14**–9 to **Figure 14**–12, the installed system is represented by the curved boundary around the various combinations of (DTS or DAS) Instrument Box and Optical Path making the measurements at any particular time.

A typical distributed sensing (DTS or DAS) installation consists of an instrument box and a fiber (optical path). See Figure 14–9.

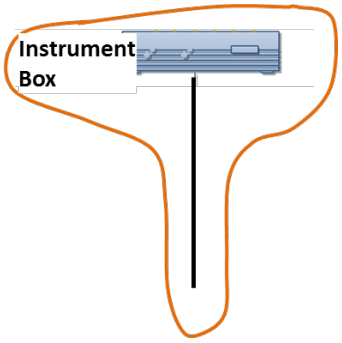


Figure 14–9. A simple single optical path and instrument box installation.

However, in real life there are multiple variations on how the equipment is installed and used. The first variation can be found when one instrument box is used to take measurements from more than one optical path. See **Figure 14**–10. Here, in the Optical Path Network object, it is possible to record the channel to which each optical path is connected at the instrument box.

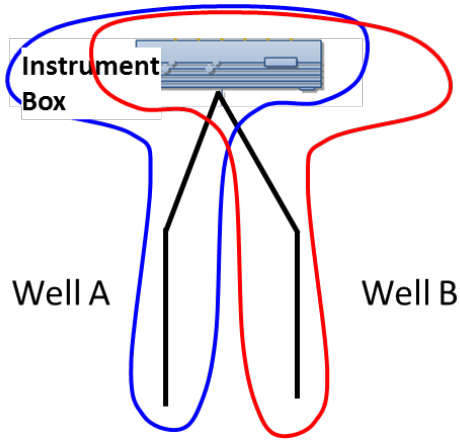


Figure 14–10. One instrument box being used with two optical paths comprises two installed systems (blue and red). In DTS each would be one installed system, both pointing to the same instrument box. In DAS, the DAS acquisitions would point to the appropriate optical path, and both would point to the same instrument box.

A second example is found when the same instrument box is moved around to different locations and used to take measurements from multiple optical paths. This is sometimes known as a 'drive-by' instrument box. See **Figure 14**–11.

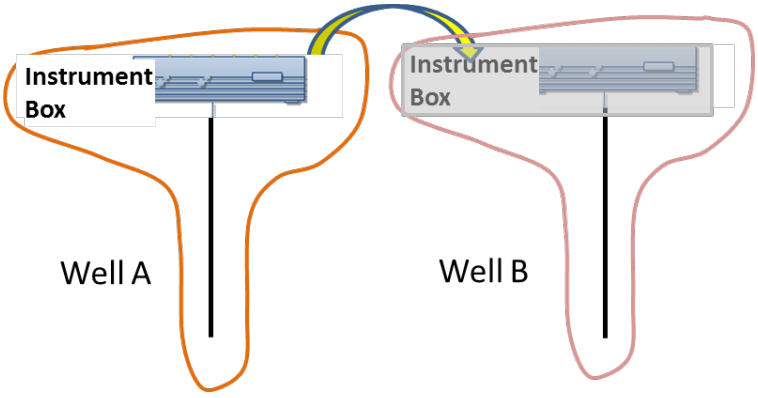


Figure 14–11. Drive-by instrument box: a box is taken from well to well and is connected only occasionally to each optical path. Each such combination is an installed system.

From the hardware point of view, the instrument box does not change, and neither does the optical path, so these two objects previously introduced are sufficient to describe these configurations. Things get more interesting when we start dealing with the measurements obtained, which depend on the situation. As an instrument box is moved from one optical path to another, it uses different calibration settings and may generate different diagnostics information. When analyzing a distributed measurement, it is very important that the data consumer knows exactly which instrument box took the measurement and to which optical path this measurement relates. Also, it is conceivable that at some point the instrument box is upgraded for a newer model and, therefore, the new measurements are now taken with a different instrument box for the same optical path.

The most extreme example of the temporary use of a distributed installed system is when an instrument box is used for a logging operation, in which case both the instrument box installation and the optical path are temporary. See **Figure 14**–12**.**

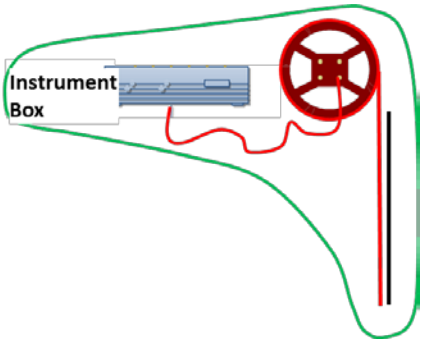


Figure 14–12. Instrument box and portable fiber used for temporary logging operation.

14.6 DTS Measurements

As mentioned previously, DTS measurements are obtained from a DTS installed system, and can contain both the 'raw' measurement and the interpreted temperature log curves (**Figure 14**–13). Both types are actually optional, depending on the situation and the requirements for that particular installation. DTS Measurement is another top-level object.

Every DTS Measurement object contains a mandatory header section that contains metadata about the measurement sets contained. Metadata includes elements such as:

- Timestamp, indicating when the measurement began.
- Timestamp, indicating when the measurement finished.
- Different flags and tag entries.
- Pointers to other measurements that may be related, e.g., to the parent measurement data for interpreted data.

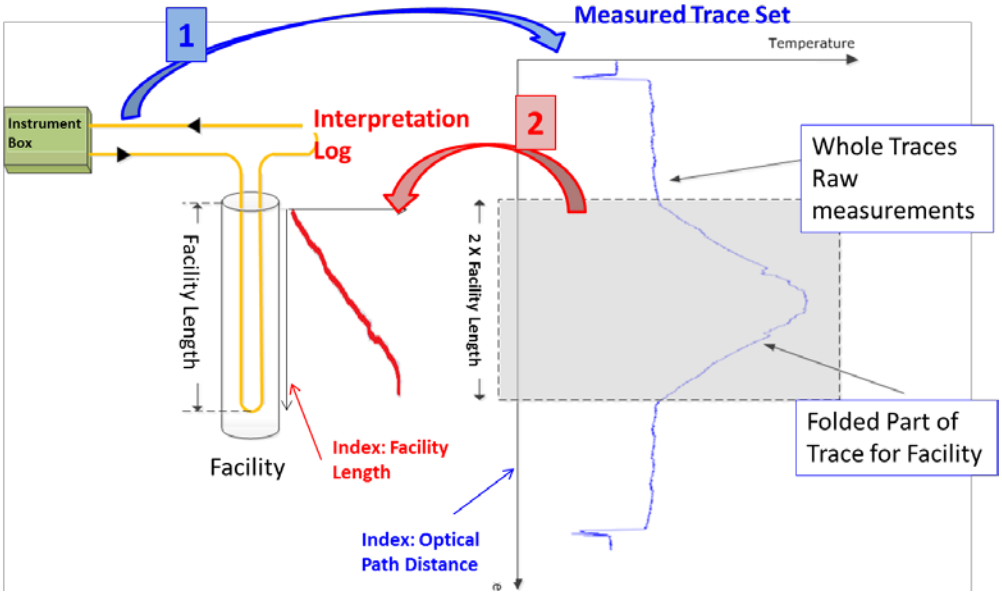


Figure 14–13. Relationship between the Measured Trace Set (blue) and the Interpreted Log (red).

The actual data (index of distance/length and the different measured or derived channels) are stored in WITSML objects called ChannelSets. For more information, refer to the *WITSML_Tech_USAGE_Guide_v2.0_Doc_v1.0.pdf* document which can be found in the WITSML download in folder *energyml\data\witsml\v2.0\doc*, and see Section 2.4. The WITSML resources can be obtained by following the links at http://www.energistics.org/drilling-completions-interventions/witsml-standards/current-standards.

Examples of WITSML ChannelSets are shown in the next chapter, as part of the worked examples.

There are two types of DTS measurement data:

- Measured Traces: the “raw” measurement, indexed along the optical path distance, and as measured by the instrument box.
- Interpreted Logs: the “adjusted” (i.e., calibrated, re-sampled, smoothed, etc.) temperature indexed against the facility length of the physical facility (well, pipe, etc.) being measured. This may be considered the “product” of DTS measurement plus analysis/processing.

The mnemonics which are expected to be used for these two types of DTS measurements are as here:

Measurement Trace		**Interpreted Log**	
Mnemonic	**Expected UoM**	**Mnemonic**	**Expected UoM**
FiberDistance (***index***)	m	FacilityDistance (***index***)	m
Antistokes	mW	AdjustedTemperature	degC
Stokes	mW		
ReverseAntiStokes	mW		
ReverseStokes	mW		
Rayleigh1	mW		
Rayleigh2	mW		
BrillouinFrequency	GHz		
Loss	dB/Km		
LossRatio	dB/Km		
CumulativeExcessLoss	dB		
FrequencyQualityMeasure	dimensionless		
MeasurementUncertainty	degC		
BrillouinAmplitude	mW		
OpticalPathTemperature *this is assumed to be adjusted to be correct temperature.*	degC		
UncalibratedTemperature1	degC		
UncalibratedTemperature2	degC		

Note that the index is mandatory for these types of data, but the other channels are not. In particular, the measurement trace set of mnemonics are suitable for either Raman- or Brillouin-type DTS systems, and any one system will certainly not generate all the measurement traces.

BUSINESS RULE: These mnemonics must be observed but are not enforced in the schemas.

15 DTS Worked Examples

This chapter reviews the XML code examples for implementing the DTS data object for each of the use cases described in Chapter 14, and which are provided in the PRODML installation in the folder *energyml\data\prodml\v2.0\doc\examples\DTS*.

15.1 Use Case 1: Represent an Optical Fiber Installation

This section contains examples of XML code for representing a typical common style of fiber installation (**Figure 15**–1). There are two fiber segments of 25m and 492.43m, giving a total distance of 517.43m. There is overstuffing of 1.9% in the downhole fiber segment, giving it a length along the facility of 483.25m. Two facility mappings are made: 1) the surface cable, which is one to one, and 2) the downhole cable mapped to the wellbore, which includes the effects of the offset to the wellhead and of the overstuffing.

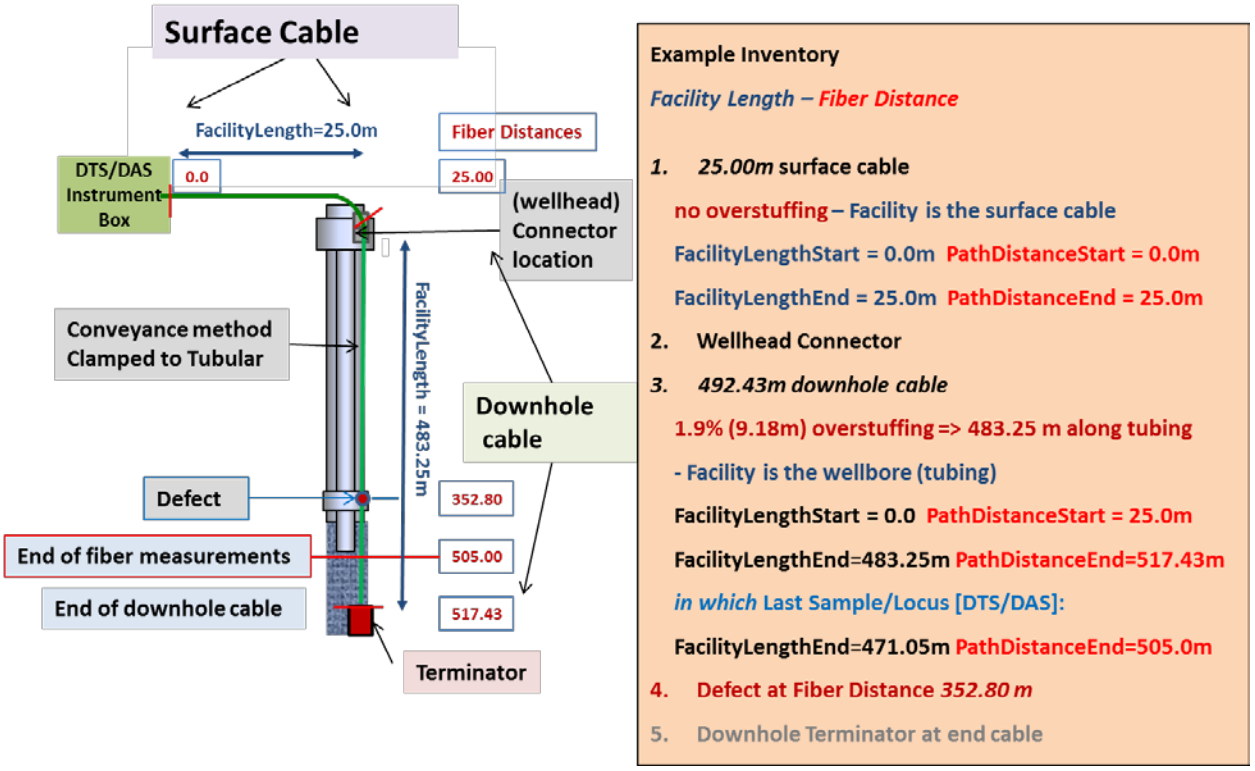


Figure 15–1. Worked example fiber optical path showing fiber segments, optical path lengths and facility lengths

For the optical path shown in Figure 15–1, the key sections are as shown below.

First, the inventory lists out the components comprising the optical path (**Figure 15**–2). Note that each component has a name and a type (and optionally additional data), and also a uid (red box) which is used to reference this component from the network (see below).

```
<Inventory>
    <Connection uid="20">
        <Name>Surface Connector</Name>
        <Type>connector</Type>
        <ConnectorType>dry mate</ConnectorType>
    </Connection>
    <Segment uid="10">
        <Name>Surface Fiber</Name>
        <Type>fiber</Type>
        <FiberLength uom="m">25</FiberLength>
        <OverStuffing uom="m">0</OverStuffing>
        <CableType>single-fiber-cable</CableType>
    </Segment>
    <Segment uid="30">
        <Name>Downhole Fiber</Name>
        <Type>fiber</Type>
        <FiberLength uom="m">492.430</FiberLength>
        <OverStuffing uom="m">9.182</OverStuffing>
        <FiberConveyance>
            <Cable xsi:type="PermanentCable">...</Cable>
        </FiberConveyance>
    </Segment>
    <Terminator uid="40">
        <Name>Main Terminator</Name>
        <TerminationType>termination at cable</TerminationType>
    </Terminator>
```

Figure 15–2. Inventory section of the Optical Path.

Next, the Network section shows how these components listed in Inventory are connected. (For details on how this is done, see Section 14.3.3, Figure 14–3.) The network contains Units which are a nominal representation of an item of inventory ("facility" is the Energistics terminology for general hardware) within the network. **Figure 15**–3 shows a network Unit (blue box), which uses a facility reference (uidRef) back to the uid of the component in Inventory (red box).

```
<Inventory>
    <Connection uid="20">...</Connection>
    <Segment uid="10">
        <Name>Surface Fiber</Name>
        <Type>fiber</Type>
        <FiberLength uom="m">25</FiberLength>
        <OverStuffing uom="m">0</OverStuffing>
        <CableType>single-fiber-cable</CableType>
    </Segment>
    <Segment uid="30">...</Segment>
    <Terminator uid="40">...</Terminator>
</Inventory>
<OpticalPathNetwork uid="OPN1">
    <ContextFacility>text</ContextFacility>
    <Network uid="N1">
        <Name>Current Setup of Well-01</Name>
        <Unit uid="OPNU10">
            <Facility uidRef="10">Surface Fiber Segment</Facility>
            <Port uid="1">
```

Figure 15–3. Network section of Optical Path showing reference to Inventory.

The unit has one or more Ports which are where connection occurs. Ports can be inlet or outlet (taking direction from the lightbox). Ports contain a Connected Node element, and where two ports share the same name and uid of Node, this defines their connection. See **Figure 15**–4, in which the outlet port (i.e., “far end”) of the surface fiber (in blue box), can be seen to be connected to the inlet port (i.e., “near end”) of the surface connector (in red box). Both have a connected node called Surface Connector 1 with uid CSC1, which defines the connection between the surface cable and this wellhead connector.

```
<Network uid="N1">
    <Name>Current Setup of Well-01</Name>
    <Unit uid="OPNU10">
        <Facility uidRef="10">Surface Fiber Segment</Facility>
        <Port uid="1">...</Port>
        <Port uid="2">
            <Direction>outlet</Direction>
            <Name>Connection to Surface Connector</Name>
            <ConnectedNode uid="CSC1">
                <Node>Surface Connector 1</Node>
            </ConnectedNode>
        </Port>
    </Unit>
    <Unit uid="OPNU20">
        <Facility uidRef="20">Surface Connector</Facility>
        <Port uid="3">
            <Direction>inlet</Direction>
            <Name>Connection from Surface Connector to Surface Fiber</Name>
            <ConnectedNode uid="CSC1">
                <Node>Surface Connector 1</Node>
            </ConnectedNode>
        </Port>
        <Port uid="4">...</Port>
    </Unit>
    <Unit uid="OPNU30">...</Unit>
    <Unit uid="OPNU40">...</Unit>
</Network>
```

Figure 15–4. Optical path network showing Node connecting Ports of two components.

In Figure 15–5, we see the full inventory and network of the worked example with the links and connections all shown.

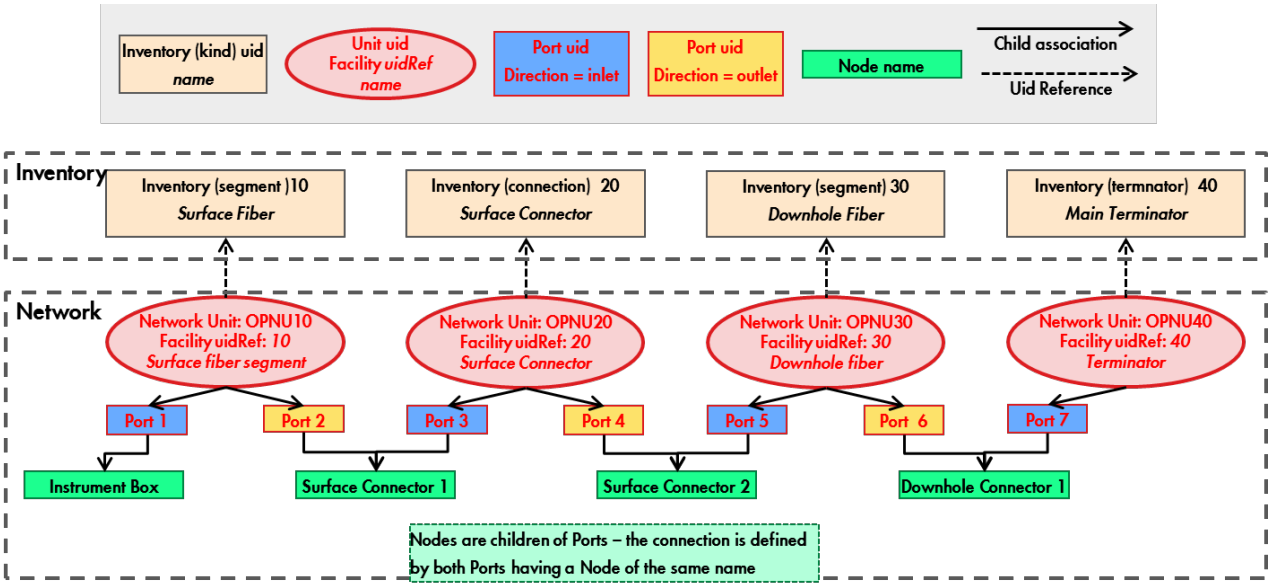


Figure 15–5. Worked example optical path network showing how connectivity is defined.

The facility mapping section is where the fiber optical path distances are mapped onto the facility lengths of physical equipment. Here, two mappings are shown: 1) a one-to-one mapping to the surface cable from the surface segment of the optical path and 2) a mapping to the wellbore, which includes the offset and the overstuffing, as shown in Figure 15–1. **Figure 15**–6 shows the two mappings. Note that the FiberFacilityWell type of mapping (the second one) includes a well datum and a data object reference to the wellbore concerned.

```
<FacilityMapping uid="FM1">
    <TimeStart>2005-07-20T01:00:00</TimeStart>
    <FiberFacilityMappingPart uid="FMP1">
        <OpticalPathDistanceStart uom="m">0.0</OpticalPathDistanceStart>
        <OpticalPathDistanceEnd uom="m">25.0</OpticalPathDistanceEnd>
        <FacilityLengthStart uom="m">0.0</FacilityLengthStart>
        <FacilityLengthEnd uom="m">25.0</FacilityLengthEnd>
        <Comment>No 'timeEnd' specified since this mapping is still valid. No overstuffing in surface cable</
Comment>
        <FiberFacility xsi:type="FiberFacilityGeneric">
            <FacilityName>Surface Cable 1</FacilityName>
            <FacilityKind>Surface Cable</FacilityKind>
        </FiberFacility>
    </FiberFacilityMappingPart>
    <FiberFacilityMappingPart uid="FMP2">
        <OpticalPathDistanceStart uom="m">25.0</OpticalPathDistanceStart>
        <OpticalPathDistanceEnd uom="m">517.43</OpticalPathDistanceEnd>
        <FacilityLengthStart uom="m">0.0</FacilityLengthStart>
        <FacilityLengthEnd uom="m">483.25</FacilityLengthEnd>
        <Comment>No 'timeEnd' specified since this mapping is still valid. The over stuffing of 9.182 m equally
distributed along the installed downhole cable</Comment>
        <FiberFacility xsi:type="FiberFacilityWell">
            <Name>Well-01 Fiber</Name>
            <WellDatum>kelly bushing</WellDatum>
            <WellboreReference>
                <eml:ContentType>wellbore</eml:ContentType>
                <eml:Title>Main wellbore of well Well-01</eml:Title>
                <eml:Uuid>4ab0fae6-dc0e-405a-b812-203a154c1243</eml:Uuid>
```

Figure 15–6. Facility Mapping section of the Optical Path.

Figure 15–7 shows some of the other sections of the Optical Path. Defects can be recorded (for the example, see Figure 15–1 and, for real-world defect examples, see Figure 14–7). OTDR surveys to verify the optical path quality can be referenced. The data for OTDR are not in an Energistics standard because OTDR is a generic fiber optical industry measurement with its own file formats; an external file is referenced. Other data can be added as seen, such as vendor, the facility identifier, etc.

```
<Defect defectID="OPTDEFECT1">
    <OpticalPathDistanceStart uom="m">352.80</OpticalPathDistanceStart>
    <DefectType>other</DefectType>
    <TimeStart>2005-07-20T01:00:00</TimeStart>
    <Comment>Consistent signal spike detected at this location</Comment>
</Defect>
<Otdr uid="OTDR1">
    <Name>Initial OTDR</Name>
    <DTimRun>2005-02-14T09:00:00</DTimRun>
    <DataInOTDRFile>myOTDR.dat</DataInOTDRFile>
    <OpticalPathDistanceStart uom="m">0</OpticalPathDistanceStart>
    <OpticalPathDistanceEnd uom="m">517.43</OpticalPathDistanceEnd>
    <Direction>forward</Direction>
    <Wavelength uom="nm">1550</Wavelength>
</Otdr>
<InstallingVendor>...</InstallingVendor>
<FacilityIdentifier xsi:type="FacilityIdentifier" uid="">
    <Name>Well-01</Name>
</FacilityIdentifier>
```

Figure 15–7. Defect, OTDR, and other sections of Optical Path.

15.1.1 Instrument Box

Declaring a DTS Instrument Box is also very straightforward (**Figure 15**–8**)**. Required elements include unique identifier, a name, and the type. All other attributes are optional.

```
<prodml:DtsInstrumentBox xmlns:prodml="http://www.energistics.org/energyml/data/prodmlv2" xmlns
http://www.energistics.org/energyml/data/commonv2" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
="2" uuid="dc0e381a-094a-4fd2-ab89-dce867e3b99d" xsi:schemaLocation="
http://www.energistics.org/energyml/data/prodmlv2 ../../../xsd_schemas/DtsInstrumentBox.xsd">
    <eml:Citation>...</eml:Citation>
    <prodml:SerialNumber></prodml:SerialNumber>
    <prodml:InternalOvenLocationNear uom="0.1 ft">1</prodml:InternalOvenLocationNear>
    <prodml:InternalOvenLocationFar uom="0.1 ft">2</prodml:InternalOvenLocationFar>
    <prodml:FacilityIdentifier uid="System 1">...</prodml:FacilityIdentifier>
    <prodml:InstrumentCalibration uid="Instrument Calibration1">
        <prodml:DTimCalibration>2002-05-30Z</prodml:DTimCalibration>
    </prodml:InstrumentCalibration>
    <prodml:Instrument>
        <prodml:Name>Instrument 1</prodml:Name>
        <prodml:Manufacturer>Nimbus 2000</prodml:Manufacturer>
        <prodml:ManufacturingDate>2001-07-22Z</prodml:ManufacturingDate>
        <prodml:Type>Turbo</prodml:Type>
    </prodml:Instrument>
</prodml:DtsInstrumentBox>
```

Figure 15–8. DTS Instrument Box example.

15.1.2 Installed System

Once both the optical path and instrument box are declared, they need to be 'tied together' as a DTS installed system. All measurements will be generated by an installed system, not an individual instrument box or an optical path. The DTS Installed System is a simple object to do this, as seen in **Figure 15**–9. It additionally allows for a calibration to be associated with the system of path + box as a combination.

```
<prodml:DtsInstalledSystem xmlns:prodml="http://www.energistics.org/energyml/data/pr
http://www.energistics.org/energyml/data/commonv2" xmlns:xsi="http://www.w3.org/2001
="2" uuid="dc0e381a-094a-4fd2-ab89-dce867e3b99d" xsi:schemaLocation="
http://www.energistics.org/energyml/data/prodmlv2 ../../../xsd_schemas/DtsInstalled
    <eml:Citation>...</eml:Citation>
    <prodml:DateMin>2001-12-17T09:30:47Z</prodml:DateMin>
    <prodml:OpticalPathLength uom="m">517.43</prodml:OpticalPathLength>
    <prodml:OpticalPathReference>
        <eml:ContentType>optical path reference</eml:ContentType>
        <eml:Title>OP1</eml:Title>
        <eml:Uuid>bfd164f4-f9e3-41c0-a74e-372b69cc2a09</eml:Uuid>
    </prodml:OpticalPathReference>
    <prodml:InstrumentBoxReference>
        <eml:ContentType>instrument box</eml:ContentType>
        <eml:Title>Sample Instrument Box</eml:Title>
        <eml:Uuid>dc0e381a-094a-4fd2-ab89-dce867e3b99d</eml:Uuid>
    </prodml:InstrumentBoxReference>
    <prodml:FacilityIdentifier uid="System 1">
        <prodml:Name>Sample Well</prodml:Name>
        <prodml:Installation>my platform</prodml:Installati
        <prodml:Kind>well</prodml:Kind>
        <prodml:ContextFacility>System 1</prodml:ContextFac
        <prodml:GeographicContext></prodml:GeographicContext>
    </prodml:FacilityIdentifier>
</prodml:DtsInstalledSystem>
```

Reference to the optical path which made this measurement

Reference to the instrument box which made this measurement

Figure 15–9. DTS Installed System.

A 'permanent' DTS installation (i.e., the lightbox is fixed and so is the optical path) will only contain a <dateMin> element with no <dateMax>, to indicate that this installation is still active.

After these three entities are declared, then captured measurements can be made and associated with the installed system above.

It is strongly recommended that the installed system object also includes calibration information so that any measurements obtained from the installed system can be compared against the calibration parameters to help determine if any further fine tuning is required for the lightbox.

15.2 Use Case 2: Capturing DTS measurements for Transport and Storage

As stated previously, there are two types of measurements that can be obtained from a DTS installed system: 1) the actual 'raw' DTS measurement along the length of the optical path and 2) the interpretation log, which has a calculated temperature along the length of the facility only. This section contains examples of XML for representing a DTS Measurement obtained from the installation described in the previous use case.

In the previous use case, we assume the well is 483.25m deep and the entire optical path is 517.43m long. Therefore, it is expected to have a measurement from distance 0 to distance 517.43m and an interpretation log from distance 25m to distance 470m (the sample closest to the last measurement at 505m, see Figure 15–1).

The 'raw' measurement traces with stokes, antistokes, and an uncalibrated temperature would look as shown in **Figure 15**–10. This is the example file DTS Measurement.xml. Note that there can be many traces per DTS Measurement xml. Each has a uid (red box). Optionally, a trace can have an attribute parentMeasurementID which contains the uid of another trace which may be the parent of this trace (in the sense that this trace may have been depth shifted or otherwise derived from the parent).

Figure 15–10. DTS Measurement with measurement traces data.

The DTS Measurement object does not contain the actual numerical data—this is instead in a WITSML object called Channel Set (for more details and references, see Section 14.6). An extract of the channel

set object corresponding to the DTS measured traces is shown in **Figure 15**–11. This shows the Index element, which describes the index, and then the three channels listed above. The first of these is shown expanded (antistokes). This is the example file *DtsMeasurement In ChannelSet.xml.*

```
<witsml:Index>
    <witsml:IndexType>measured depth</witsml:IndexType>
    <witsml:Uom>m</witsml:Uom>
    <witsml:Direction>increasing</witsml:Direction>
    <witsml:Mnemonic>pathDistance</witsml:Mnemonic>
</witsml:Index>
<witsml:Channel schemaVersion="2" uuid="86a1c354-b031-4a9b-863d-7945ac049d4f">
    <eml:Citation>...</eml:Citation>
    <witsml:Mnemonic>antistokes</witsml:Mnemonic>
    <witsml:DataType>double</witsml:DataType>
    <witsml:Uom>mW</witsml:Uom>
    <witsml:GrowingStatus>active</witsml:GrowingStatus>
    <witsml:TimeDepth>depth</witsml:TimeDepth>
    <witsml:ChannelClass schemaVersion="2" uuid="d8891798-a45f-41a3-ba06-3ef5e40ce2ee">
        <eml:Citation>...</eml:Citation>
        <eml:QuantityClass>power</eml:QuantityClass>
    </witsml:ChannelClass>
    <witsml:LoggingCompanyName>Company name</witsml:LoggingCompanyName>
    <witsml:Index>
        <witsml:IndexType>measured depth</witsml:IndexType>
        <witsml:Uom>m</witsml:Uom>
        <witsml:Direction>increasing</witsml:Direction>
        <witsml:Mnemonic>pathDistance</witsml:Mnemonic>
    </witsml:Index>
</witsml:Channel>
<witsml:Channel schemaVersion="2" uuid="6a7cbcde-57cd-4684-80d2-7000121fba14">...</witsml:Channel>
<witsml:Channel schemaVersion="2" uuid="7029c81e-2b73-4e47-b407-01f6b0529413">...</witsml:Channel>
<witsml:Data>
    <witsml:Data>
```

Figure 15–11. Channel Set for DTS Measurement traces.

The data then follows (as seen about to start at the bottom of Figure 15–11), as shown in Figure 15–12. The comma-separated values in the data section correspond to the channels defined previously in this object.

```
<witsml:ChannelSet schemaVersion="2" uuid="987587b3-18bd-4461-b4b5-50144548bf96" xsi:schemaLocation="
http://www.energistics.org/energyml/data/witsmlv2 ../../../../../witsml/v2.0/xsd_schemas/Log.xsd" xmlns:eml="
http://www.energistics.org/energyml/data/commonv2" xmlns:witsml="http://www.energistics.org/energyml/data/witsmlv2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
        <eml:Citation>...</eml:Citation>
        <witsml:Index>...</witsml:Index>
        <witsml:Channel schemaVersion="2" uuid="86a1c354-b031-4a9b-863d-7945ac049d4f">...</witsml:Channel>
        <witsml:Channel schemaVersion="2" uuid="6a7cbcde-57cd-4684-80d2-7000121fba14">...</witsml:Channel>
        <witsml:Channel schemaVersion="2" uuid="7029c81e-2b73-4e47-b407-01f6b0529413">...</witsml:Channel>
        <witsml:Data>
            <witsml:Data>
            [
            [5,0.696967324,3.514406317,25.68171947]
[10,0.322944731,3.308459525,25.48662041]
[15,0.790652363,0.164481871,25.49488728]
[20,0.739910877,4.487284368,25.3289979]
[25,0.249175057,4.481827007,25.47721579]
[30,0.280988987,2.256644365,25.86206061]
[35,0.106927337,3.018633943,26.3491024]
[40,0.875277243,1.44599566,26.77767283]
[45,0.701072275,4.464228736,27.47208933]
[50,0.323033111,4.26778354,27.77949983]
[55,0.459705236,3.525890013,28.84776963]
```

Figure 15–12. Data section of DTS Measurement for measurement traces.

Similarly, an interpreted log is shown in the example file DTS Measurement with Log.xml. This replaces the measurement traces with an interpreted log but is otherwise similar. The data runs from 5m to 470m of facility length (i.e., measured depth in this wellbore), see Figure 15–1. The Channel Set file is DTS Log Interpretation In ChannelSet.xml. See Figure 15–13 which shows part of the DTS measurement file. Note the two referenced Channel Sets (red and blue boxes). Note also the IntepretationData has a mandatory measurement reference attribute which contains the uid of the measurement trace from which this interpretation log was derived (green boxes).

```
<prodml:InterpretationLog >
    <prodml:InterpretationData measurementReference="Measurement 1" uid="ID1" >
        <prodml:FacilityMapping>a</prodml:FacilityMapping>
        <prodml:SamplingInterval uom="m">5</prodml:SamplingInterval>
        <prodml:BadFlag>false</prodml:BadFlag>
        <prodml:CreationStartTime>1967-08-13</prodml:CreationStartTime>
        <prodml:InterpretationProcessingType>temperature-shifted</prodml:InterpretationProcessingType>
        <prodml:IndexMnemonic>facilityDistance</prodml:IndexMnemonic>
        <prodml:PointCount>1500</prodml:PointCount>
        <prodml:ChannelSetReference>
            <eml:ContentType>Log Interpretation ChannelSet Data</eml:ContentType>
            <eml:Title>Log Interpretation ChannelSet Data</eml:Title>
            <eml:Uuid>d191e44c-b1b3-4dd8-8395-8b588f127a40</eml:Uuid>
        </prodml:ChannelSetReference>
    </prodml:InterpretationData>
</prodml:InterpretationLog>
<prodml:MeasurementTrace uid="Measurement 1">
    <prodml:TraceProcessingType>as acquired</prodml:TraceProcessingType>
    <prodml:SamplingInterval uom="m">5</prodml:SamplingInterval>
    <prodml:IndexMnemonic>fiberDistance</prodml:IndexMnemonic>
    <prodml:PointCount>94</prodml:PointCount>
    <prodml:FrequencyRayleigh1 uom="Hz">10</prodml:FrequencyRayleigh1>
    <prodml:FrequencyRayleigh2 uom="Hz">20</prodml:FrequencyRayleigh2>
    <prodml:ChannelSetReference>
        <eml:ContentType>Measurement Trace Channel Set for ID1</eml:ContentType>
        <eml:Title>Measurement Trace Channel Set for DTS1</eml:Title>
        <eml:Uuid>987587b3-18bd-4461-b4b5-50144548bf96</eml:Uuid>
    </prodml:ChannelSetReference>
</prodml:MeasurementTrace>
```

Figure 15–13. DTS measurement with measurement trace and interpretation log.

15.3 Use Case 3: Manipulation of DTS-derived Temperature Log Curves

This section contains example XML that represent situations when temperature logs are modified by end-users after DTS measurements or interpretation logs have been obtained and stored in a repository. These types of modifications are common in places where additional workflows are in place that enable subject matter experts to apply different mathematical algorithms to the data (for example, de-noising the temperature values) and need to store the results for sharing with other people. When changing values of interpreted logs, it's very important that the original log is never changed. It is also important that any modified log is also connected to the original log from which the calculations were performed (i.e., log 'versioning'). The PRODML schema can accommodate all these scenarios.

Furthermore, there will be cases where several versions of a temperature log have been generated, but there is one version that is 'preferred' for use towards business decisions and deeper analysis. The PRODML schema provides a special tag that allows you to flag one of the log versions as the preferred

one. See

```
    <prodml:InterpretationLog>
        <prodml:PreferredInterpretationReference>ID3</prodml:PreferredInterpretationReference>
        <prodml:InterpretationData uid="ID1" measurementReference="Measurement 1" >...</prodml:InterpretationData>
        <prodml:InterpretationData uid="ID2" parentInterpretationReference="ID1" measurementReference="Measurement 1">
        <prodml:InterpretationData uid="ID3" parentInterpretationReference="ID2" measurementReference="Measurement 1">
            <prodml:FacilityMapping>FMP2</prodml:FacilityMapping>
            <prodml:SamplingInterval uom="m">5</prodml:SamplingInterval>
            <prodml:BadFlag>false</prodml:BadFlag>
            <prodml:CreationStartTime>1967-08-13</prodml:CreationStartTime>
            <prodml:InterpretationProcessingType>denormalized</prodml:InterpretationProcessingType>
            <prodml:IndexMnemonic>facilityDistance</prodml:IndexMnemonic>
            <prodml:PointCount>94</prodml:PointCount>
            <prodml:ChannelSetReference>
                <eml:ContentType>Log Interpretation Channel Set Data 3</eml:ContentType>
                <eml:Title>Log Interpretation ChannelSet Data</eml:Title>
                <eml:Uuid>d191e44c-b1b3-4dd8-8395-8b588f127a40</eml:Uuid>
            </prodml:ChannelSetReference>
        </prodml:InterpretationData>
    </prodml:InterpretationLog>
    <prodml:MeasurementTrace uid="Measurement 1">...</prodml:MeasurementTrace>
</prodml:DtsMeasurement>
```

Figure 15–14. This shows the example file DTS Interpretation Log With Versions.xml.

This contains three versions of interpretation log data with its original measurement trace. The tag which identifies which is the preferred interpretation log (ID3) that should be used for business decisions is shown in a green box. All three interpretation data instances refer to the same measurement trace (red boxes). The example also shows how interpretation logs can show that they are derived from a parent interpretation log (blue box). The interpretation processing type (purple box) indicates what was processed to generate this instance. Here, ID1 was temperature shifted, ID2 took ID1 and averaged and ID3 took ID2 and denormalized. Note: in the example, all three interpretation logs point to the same Channel Set data.

```
    <prodml:InterpretationLog>
        <prodml:PreferredInterpretationReference>ID3</prodml:PreferredInterpretationReference>
        <prodml:InterpretationData uid="ID1" measurementReference="Measurement 1" >...</prodml:InterpretationData>
        <prodml:InterpretationData uid="ID2" parentInterpretationReference="ID1" measurementReference="Measurement 1">
        <prodml:InterpretationData uid="ID3" parentInterpretationReference="ID2" measurementReference="Measurement 1">
            <prodml:FacilityMapping>FMP2</prodml:FacilityMapping>
            <prodml:SamplingInterval uom="m">5</prodml:SamplingInterval>
            <prodml:BadFlag>false</prodml:BadFlag>
            <prodml:CreationStartTime>1967-08-13</prodml:CreationStartTime>
            <prodml:InterpretationProcessingType>denormalized</prodml:InterpretationProcessingType>
            <prodml:IndexMnemonic>facilityDistance</prodml:IndexMnemonic>
            <prodml:PointCount>94</prodml:PointCount>
            <prodml:ChannelSetReference>
                <eml:ContentType>Log Interpretation Channel Set Data 3</eml:ContentType>
                <eml:Title>Log Interpretation ChannelSet Data</eml:Title>
                <eml:Uuid>d191e44c-b1b3-4dd8-8395-8b588f127a40</eml:Uuid>
            </prodml:ChannelSetReference>
        </prodml:InterpretationData>
    </prodml:InterpretationLog>
    <prodml:MeasurementTrace uid="Measurement 1">...</prodml:MeasurementTrace>
</prodml:DtsMeasurement>
```

Figure 15–14. Multiple DTS interpretation logs in one transfer, showing preferred log, parentage, and processing type.

16 Appendix for DTS: Fiber Optic Technology and DTS Measurements

This appendix provides details on what DTS is, implications for oil and gas wells, and some of the challenges that the DTS data object was designed to address.

16.1 Principles

When light is transmitted along an optic fiber, transmitted light collides with atoms in the lattice structure of the fiber and causes them to emit bursts of light at frequencies slightly different from the transmitted radiation, which propagate back along the fiber and can be detected at its end. This is known as backscattering; this is shown in **Figure 16**–1.

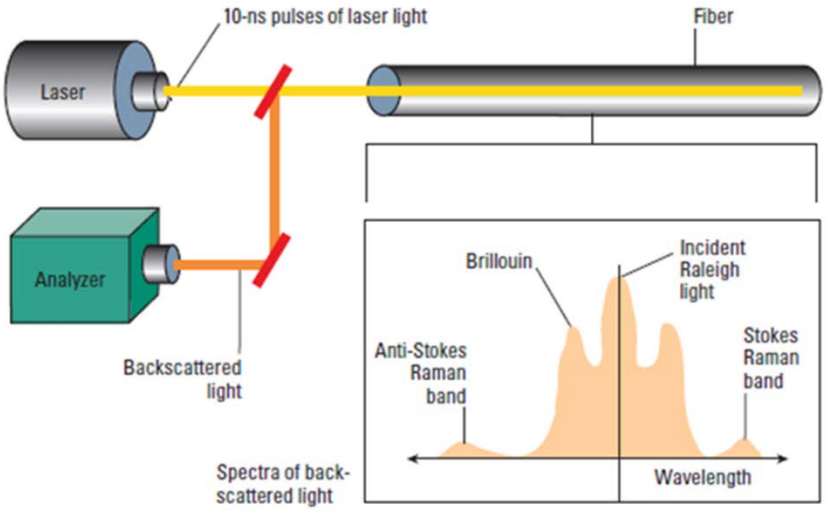


Figure 16–1. Principle of DTS Measurement.

Optic fiber thermometry depends upon the phenomenon that the frequency shifts occur to bands both less than and greater than the transmitted frequency. Furthermore, in the case of the Raman shifted bands the intensity of the lower frequency band is only weakly dependent on temperature, while the intensity of the higher frequency band is strongly dependent upon temperature (Anti-stokes and Stokes bands). See **Figure 16**–2. Less commonly, the Brillouin bands are used.

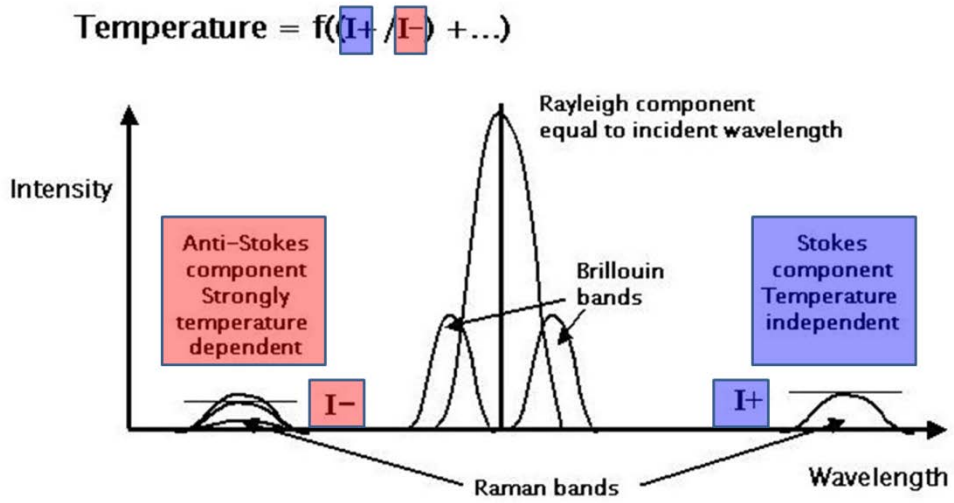


Figure 16–2. Backscatter spectrum, showing Rayleigh, Brillouin, and Raman bands.

The ratio of these intensities is related to the temperature of the optic fiber at the site where the backscattering occurs: i.e., all along the fiber. These intensities are averaged at regular Sampling Intervals. See **Figure 16**–3. The intensities of backscatter of individual pulses are very weak, necessitating statistical stacking of thousands of backscattered light pulses.

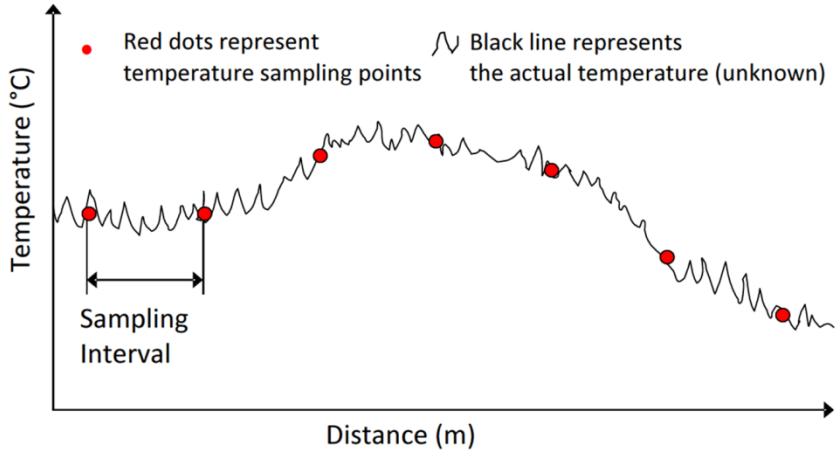


Figure 16–3. . Illustration of temperature data which is averaged over fixed Sample Intervals.

16.2 Equipment: Instrument Box and Fiber

Several organizations manufacture and / or operate DTS equipment on a service basis for the E&P industry. The optic fibers are mass-manufactured by a number of companies with a variety of materials and designs for different applications. The 'Instrument Boxes' comprise a laser light source, an oven which provides a reference temperature, and photo-electric sensors which capture the backscattered light, along with digital circuitry which both controls the laser transmission and analyses the frequency and intensity of the backscatter. Instrument boxes and fibers can readily be interchanged.

The transmission characteristics of optic fibers are dependent upon several factors in addition to temperature, which include their construction, physical deformation and chemical, particularly aqueous,

contamination. For these reasons, calibrations of the equipment are routinely necessary to monitor the transmission characteristics.

16.3 Installation in Wellbores

The location of the source of the backscattered light is determined in terms of its distance along the fiber by the two-way transmission time and the velocity of light in the transmitting medium. It is therefore necessary to know how length along the fiber correlates to measured depth along the wellbore in which the fiber is installed. The foregoing information includes knowledge of the positions in the well and along the fiber where connections through casing shoes, packers and other features occur. All these can be assumed to remain constant between initial installation and mechanical interventions in the wellbore.

There are several fiber configurations deployed in wellbores, illustrated in **Figure 16**–4 which enable various calibrations to be made to the temperature data.

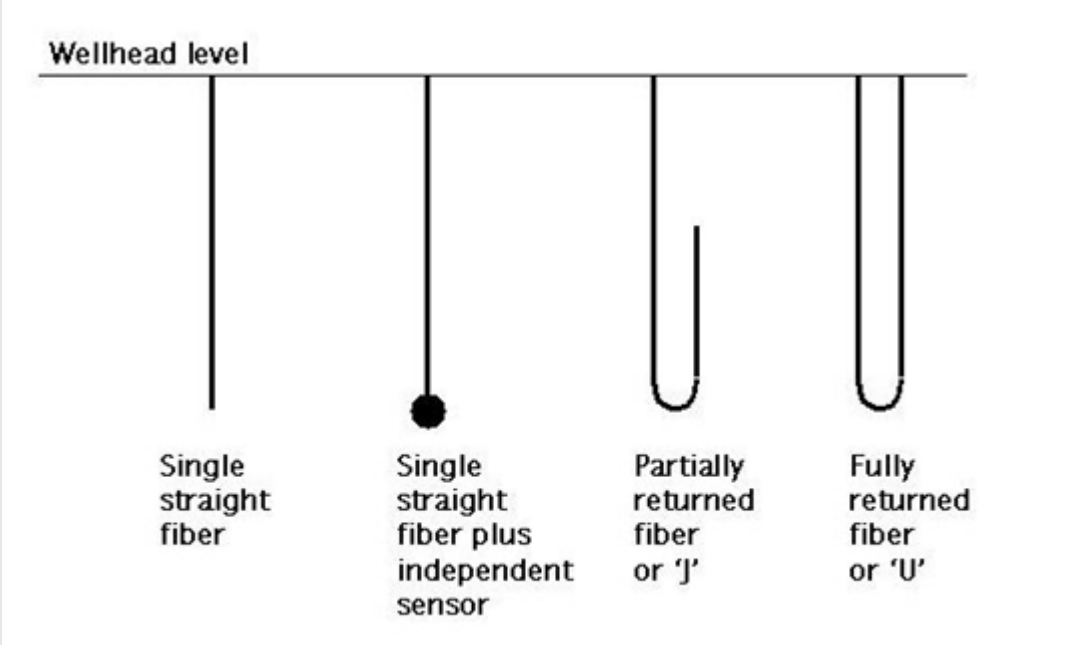


Figure 16–4. DTS fiber configuration patterns.

16.4 Calibrations

Two kinds of calibration are applied to DTS data. These are:

- Factors pertaining to the Instrument Box such as the oven reference temperature, the number of launches being stacked to get a temperature value at the required resolution.
- Factors related to the fiber, including the (practically linear) differential attenuation along the fiber, and the refractive index of the fiber.

Where the fiber may be installed in a well bore for several years, it is necessary to recalibrate these factors periodically in order to be able to compare reservoir temperatures over these periods.

The Optical Time Domain Reflectometry (OTDR) technique provides a routine means of examining the condition of the fiber. OTDR consists of observing the attenuation of backscattered light in the frequency band of the transmitted Rayleigh waves. An OTDR profile is illustrated in **Figure 16**–5.

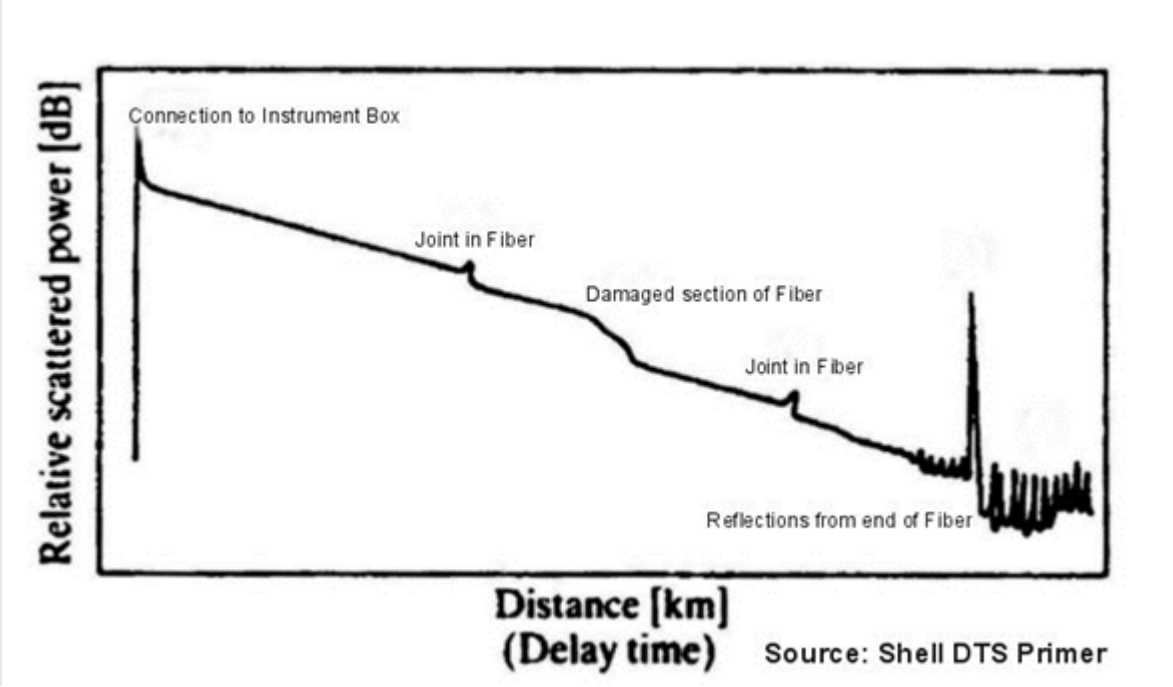


Figure 16–5. Optical Time Domain Reflectometry (OTDR) Profile.

16.5 Measurements

The measurements produced by the Instrument Box for a temperature profile typically comprise a set of records containing contextual data followed by the temperature records. The contextual data identify the well, the equipment, the date and time, and various calibration data. The temperature records contains the length along the fiber of the temperature sample, the computed temperature, and measurements of the Stokes and anti-Stokes intensities, and other measured curves as appropriate to the Instrument Box concerned.

Part V: Distributed Acoustic Sensing (DAS)

Part V contains Chapters 17 through 22 which explain the set of PRODML data objects for distributed acoustic sensing (DAS).

Acknowledgement

Special thanks to the following companies and the individual contributors from those companies for their work in designing, developing and delivering the DAS data object: Shell, Fotech, OptaSense, BP, Baker Hughes, Chevron, Pinnacle/Halliburton, Schlumberger, Silixa, Weatherford, Ziebel.

17 Introduction to DAS

This chapter serves as an overview and executive summary of both distributed acoustic sensing (DAS) and the DAS data objects.

17.1 Overview of DAS

Distributed acoustic sensing (DAS) is a fiber optic technology used for oil and gas surveillance applications, which include wellbore, pipeline and surface facility monitoring. DAS datasets are used as input to many production surveillance software applications. Examples of downhole applications include monitoring of flow, hydraulic fractures, and seismic surveys. Examples of surface applications include monitoring of pipelines, intrusion detection, and equipment vibration.

At a high level, here's how DAS works (**Figure 17**–1): Oil and gas assets, such as wells and pipelines, are outfitted with optical fibers (fiber optic cable). Acoustic sources create small vibrations of the fiber, and DAS interrogator systems measure tiny variations in the back-scattered light caused by these vibrations—which essentially turns the fiber itself into a sensitive distributed sensor that can measure acoustic, temperature, and strain variations over distances of many kilometers.

The data collected by the DAS interrogator unit is often referred to as raw data and is further processed; this processing often includes data reduction, filtering of frequency bands of interest, or transformations into spectrum data. Processed datasets that are commonly provided to customers are frequency band extracted (FBE) data and Fourier transformed spectrum datasets. Surveillance software applications derive information from both raw and processed datasets.

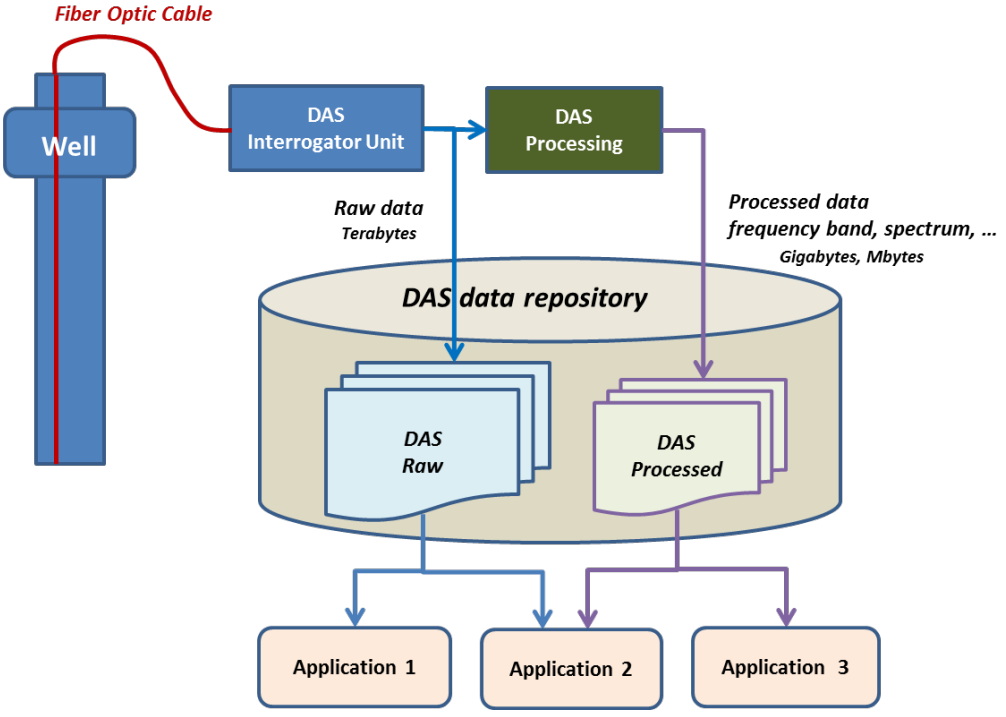


Figure 17–1. DAS Overview: Oil and gas assets are outfitted with optical fibers, which are vibrated by an acoustic source. DAS interrogator systems measure tiny variations in the back-scattered light, essentially turning the fiber itself into a sensitive distributed sensor that can measure acoustic, temperature, and strain variations over distances of many kilometers. DAS data acquisition raw and processed (frequency band and spectrum) data are used for input to many production surveillance software applications.

For a quick overview or to be able to make a presentation to colleagues, see the slide set: Worked Example DAS.pptx which is provided in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files.

17.2 The Business Case

Since 2009, DAS applications for oil and gas have seen rapid development by both service companies and operators. Several commercial applications for surveillance based on DAS measurements have been developed and are being offered on the market today, including a large range of wellbore applications such as noise monitoring, hydraulic fracturing and perforation monitoring, flow profiling for several well types, and vertical seismic profiling. Furthermore, several pipeline monitoring applications were under development when this document was being written.

17.2.1 Benefits of DAS-based Systems

In many cases, fiber-based DAS applications are replacing conventional downhole measurements performed with wireline tools or permanently installed downhole sensors because they often can provide data more frequently, with higher resolution, and over longer distances, in harsh environments, and sometimes at significantly lower cost, than conventional applications.

17.2.2 Challenges of DAS

The challenge for DAS is how to manage the huge volumes of data produced, which is typically terabytes for raw data and hundreds of gigabytes for frequency filtered data. Because of the large data volumes and lack of standards, industry workers spend lots of time manually reformatting data, which is costly and error prone. These challenges hamper uptake of the commercial DAS applications and development of new DAS applications.

For these reasons, a group of operators and service providers proposed to Energistics to define a shared industry exchange format for DAS data types. Previous work at Energistics has been done for distributed temperature sensing (DTS). Because of the similarities, the DAS design work is based on the DTS design.

17.3 Scope of DAS Data Objects

Data in scope for the current version of the DAS data object includes:

- A description of the DAS optical path and instrument box (leveraging the existing PRODML DTS (distributed temperature sensing) data object.
- Raw and processed DAS data; Processed data types included are frequency band extracted (FBE) and Fourier transformed spectrum data.

Out of scope are data formats for specific applications that derive end products from the raw, frequency band or spectrum DAS data; examples are seismic traces, leak detection events, and flow profiles. For many of these applications, industry standards already exist (SEG standards for seismic, LAS for well log data, PRODML and WITSML for DTS, etc.).

17.3.1 DAS Data Objects

The DAS data objects were developed to provide an industry-defined, vendor-neutral format for exchanging the large volumes of data associated with DAS. It builds on recent work to define data-exchange standards for data associated with distributed temperature sensing (DTS), which is also part of PRODML.

Like all Energistics standards, DAS is implemented in software using standards-based technology, such as XML and HDF5. Developers implement the DAS data object so that software can read and write the DAS format, thereby allowing software packages to “exchange” data. For more information on technology used, see Section 19.1.1.

17.4 DAS Use Cases and Workflow

The current version of the DAS data object supports exchanging data between the business process use cases that are listed below and described in Section 22.2.

- Define DAS acquisition requirements with client

- Represent fiber optic installation
- Configure DAS equipment for acquisition
- Perform DAS data acquisition
- Post-process DAS data

17.4.1 DAS Workflow

Figure 17–2 shows a very high-level DAS workflow. Assets such as wells and pipelines are fitted with fiber optic cables. DAS interrogator systems are connected to these fibers and collect huge volumes of raw acoustic data. The raw data is collected in the field, often some pre-processing takes place to extract relevant data and reduce the data volume. The raw and/or extracted datasets are then transferred to the office where they are further processed and stored. Oil and gas software uses both the raw and processed data as inputs for applications such as surveillance.

For a more detailed workflow, see Section 18.2.

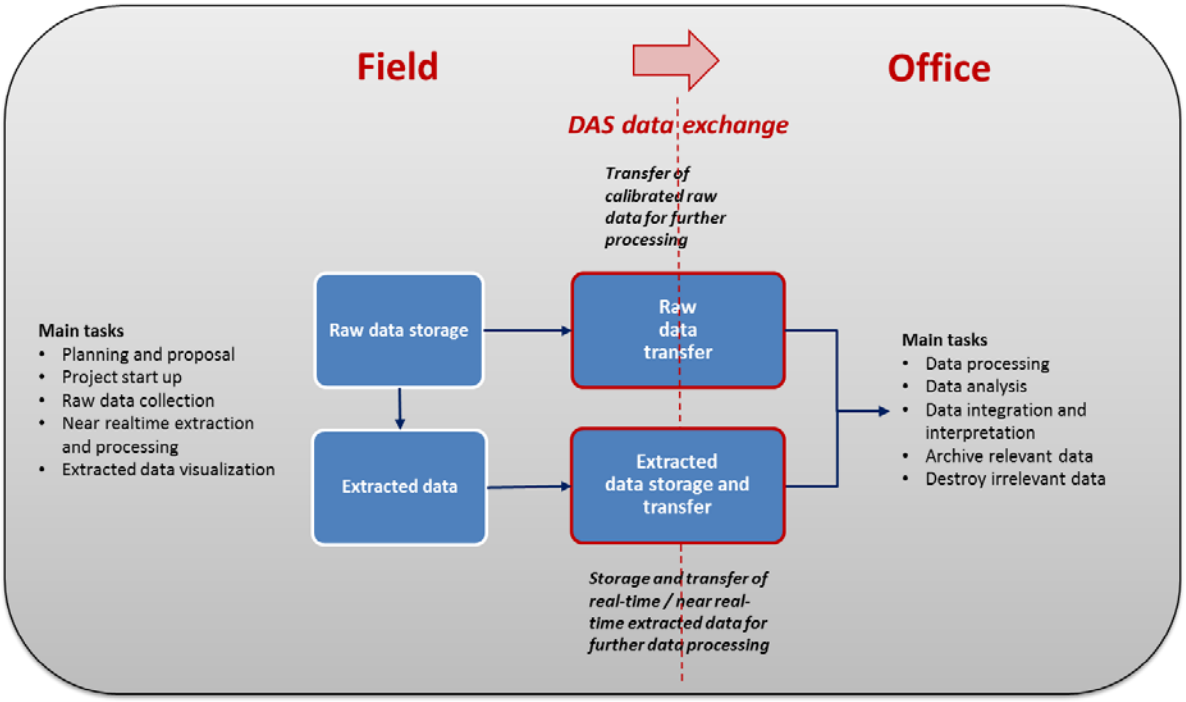


Figure 17–2. High-level DAS workflow, from the field to the office.

18 DAS: Key Concepts, Workflow, and Use Cases

This chapter provides:

- Useful concepts and terminology to help you understand DAS technology and processes
- An overview of the DAS workflow

Additional information:

- For business process use cases that are supported by the DAS data object, see Section 22.2.
- For information on the DAS data model, see Chapter 19.

18.1 DAS Measurement Concepts

18.1.1 Terminology

Because DAS is an emerging technology with new and specialized terminology, definitions are provided in this document.

SEAFOM (an international joint industry forum aimed at promoting the growth of fiber optic monitoring system installations in the upstream oil and gas industry) and the Energistics DAS work group have collaborated to define key terms for DAS. A few key terms are defined here for convenience. For a detailed list of definitions, see Chapter 22.

Term	Definition
DAS Job	A set of one or more DAS acquisitions acquired in a defined timeframe using a common optical path and DAS instrument box.
DAS Acquisition	Collection of DAS data acquired during the DAS survey.
DAS Instrument Box	The DAS measurement instrument. Sometimes called the “interrogator unit”.
Optical Path	A series of fibers, connectors, etc. that together form the path for the light pulse emitted from the interrogator unit.
Interrogation Rate (or Pulse Rate)	The rate at which the DAS instrument box interrogates the fiber sensor. For most instruments, this is informally known as the pulse rate.
Output Data Rate	The rate at which the measurement system provides output data for all ‘loci’ (spatial samples). This is typically equal to the interrogation rate/pulse rate or an integer fraction thereof.
Trace (or Scan)	Array of sensor values for all loci interrogated along the fiber for a single “pulse.”
Locus (plural Loci)	A particular location that indicates a spatial sample point along the sensing fiber at which a “time series” of acoustic measurements is made.
Measurement Start Time	The time at the beginning of a data “sample” in a “time series.” This is typically a GPS-locked time measurement.
Sample	A single measurement, i.e., the acoustic signal value at a single locus for a single trace.
Spatial Resolution	The ability of the measurement system to discriminate signals that are spatially separated. It should not be confused with Spatial Sampling Interval.
Spatial Sampling Interval	The separation between two consecutive “spatial sample”’ points on

Term	Definition
	the fiber at which the signal is measured. It should not be confused with “spatial resolution.”
DAS Raw Data	DAS data exchange format that describes unprocessed DAS data provided by the vendor to a customer.
DAS FBE Data	A category of “processed” data. DAS data exchange format for frequency band extracted (FBE) DAS data provided by the vendor to a customer. This DAS data type describes a dataset that contains one or more frequency bands filtered time series of a “raw” DAS dataset. For example, the RMS of a band passed filtered time series or the level of a frequency spectrum.
DAS Spectrum Data	A category of “processed” data. DAS data exchange format for frequency spectrum information extracted from a (sub) set of “raw” DAS data.

18.1.2 Principles: Pulse and Backscatter of Light

An optical time-domain reflectometer (OTDR) is an optoelectronic instrument used to characterize an optical fiber. An OTDR in its most basic form uses a pulse of optical radiation to interrogate a waveguide, typically an optical fiber.

As the pulse propagates along the fiber, discontinuities formed by tiny density fluctuations (frozen into the fabric of the fiber at the time of manufacture) cause a small fraction of the incident light to scatter. This process is usually referred to as Rayleigh scattering.

This scattered light is then recaptured by the fiber and guided back toward the starting end where it can be detected. By measuring properties of the scattered light as a function of time, properties of the waveguide or fiber as a function of distance can be inferred.

The simplest way of understanding it is as a one-dimensional (1D) RADAR, but at optical wavelengths.

All OTDR systems share the following properties:

- Detection time maps linearly to location along the fiber.
- Properties of the scattered light are a function of the local properties of the waveguide.

By tailoring the properties of the launched pulse and the processing applied, different information about the waveguide/fiber being interrogated can be inferred. In the case of coherent OTDR, the fiber is interrogated by a pulse of coherent light of a finite length.

As stated above, this pulse propagates along the fiber and a small fraction is scattered. The resulting intensity of the scattered light as a function of time and hence distance is determined by the coherent sum of scatter from typically millions of scatter sites. The distribution of the position and magnitude of the scatter sites are random and hence the intensity of the scattered field is also random as a function of distance.

From this point on, different DAS technologies may use different properties of the scattered light to infer the action of an acoustic disturbance. For example, systems may measure the absolute phase of the scattered light or they may measure the differential phase; however, for illustration of how the OTDR principle may form a DAS system and for simplicity, the following example describes an intensity-based, single-pulse DAS system.

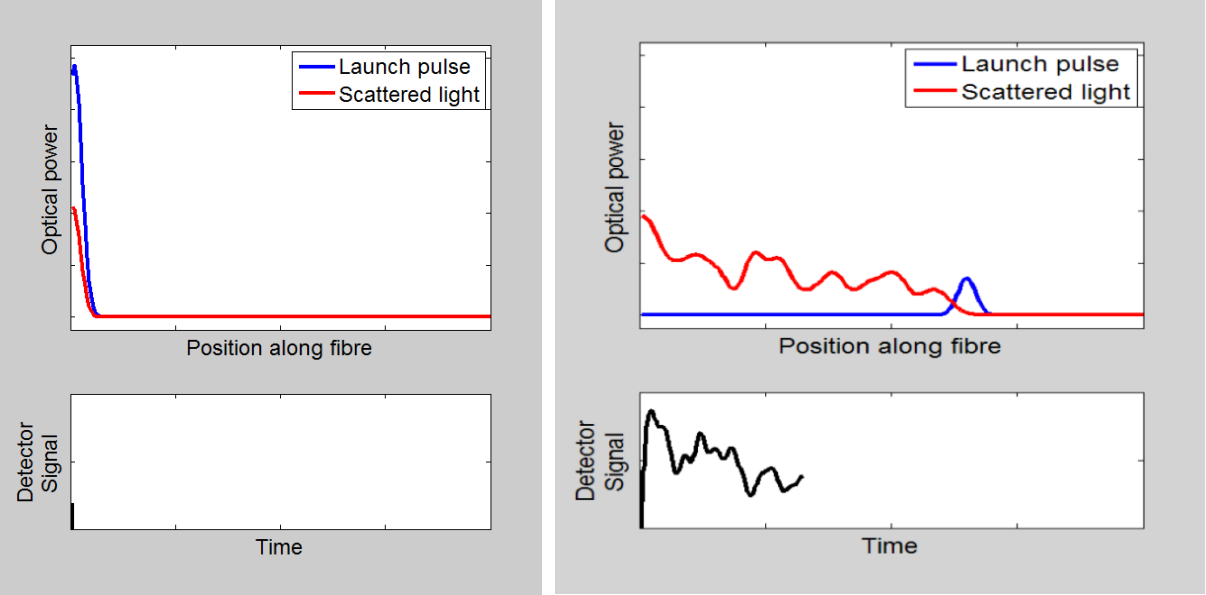
18.1.2.1 Example: Intensity-Based, Single-Pulse DAS system

Figure 18–1 (and the linked animation referenced below) shows the scattered intensity as a function of time from a fiber illuminated by single pulse of coherent light. The important thing to note for the application of DAS is that, if the fiber is unperturbed, the scatter pattern generated by a fiber remains constant on successive pulses.

For a simple but effective animation that better shows this concept, see the PowerPoint slides included in Worked Example DAS.pptx which is provided in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files (slide 9). To see the animation, run the presentation Slide Show mode.

In coherent OTDR, a launch pulse travels along fiber scattering from tiny imperfections, inherent to the fiber. The scattered light is detected and used to create a signal for each location along the fiber, known as a scatter pattern. Because the intensity of this signal at each location is determined by the coherent addition of millions of waves scattered from millions of independent scatter sites, the intensity varies randomly as a function of distance. However, upon successive measurements, this pattern nominally remains constant.

In Figure 18–1, the pair of plots on the left is a “time zero” when the pulse (blue) is sent down the fiber; the pair of images on the right is some time later as the pulse nears the end of the fiber, travelling from left to right, whilst the back scattered light from each point along the fiber is making its way back up the fiber towards the detector, from right to left (red).



(Images courtesy of OptaSense.)

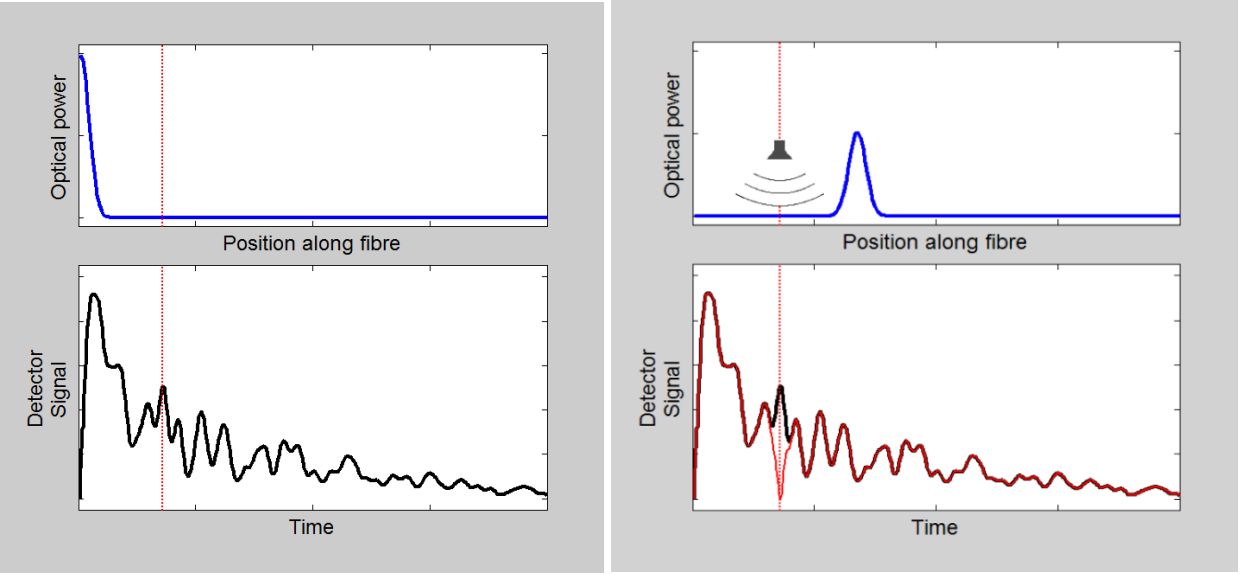
Figure 18–1. The first two plots (left) are a “time zero” when the pulse (blue) is sent down the fiber; the second pair (right) is some time later, as the pulse nears the end of the fiber, travelling from left to right, while the back scattered light from each point along the fiber is making its way back up the fiber towards the detector, from right to left (red). See the animation in the PowerPoint example; run it in Slide Show mode.

18.1.3 Measurements: Pulse Rate and Sample Locations

A typical DAS measurement cycle consists of the fiber being interrogated by a sequence of pulses at a repetition rate arranged such that the scatter from the distal end of the fiber has time to return along the length of the fiber and be detected before the next pulse being launched. Such an interrogation is often referred to as a *ping* in the same sense as that used in sonar.

In **Figure 18**–2 (and the linked animation referenced below), the coherent scatter pattern remains constant on successive pings until the fiber is disturbed. Vibration from an acoustic source modulates the fiber altering the physical length of the waveguide and in turn the relative positions of the scatter site. In turn, the coherent sum of the scatter sites in the disturbed region is also modified, thereby affecting the scatter pattern. The data gathered recording the scatter pattern over several pulses makes it possible, with correct processing, to determine a history of the nature of acoustic source as a function of position.

For a simple but effective animation that better shows this concept, see the PowerPoint slides included in the Worked Example DAS.pptx which is provided in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files (slide 11. To see the animation, run the presentation Slide Show mode.



(Images courtesy of OptaSense)

Figure 18–2. Vibration from an acoustic source modulates the fiber and hence the scatter pattern. The data gathered makes it possible to determine a history of the nature of acoustic source as a function of position. The left hand lower plot shows the back scattered intensity as a function of time (equivalent to distance along fiber) with no disturbance to the fiber (black). The right hand plots show that when a sound source is adjacent to the fiber at any location, the back scattered intensity from that location is altered (red line), so that it repeats the undisturbed signal for most of the fiber length, but diverges from it where the noise source is (red). By processing these variations all along the fiber length, the acoustic signal at each point can be obtained.

18.1.4 DAS Equipment and Data Types: Raw and Processed Data

Figure 18–3 shows basic DAS equipment configuration and usage and introduces some important concepts about the raw data and how it's represented in the DAS PRODML data schemas, which are explained in Chapter 19.

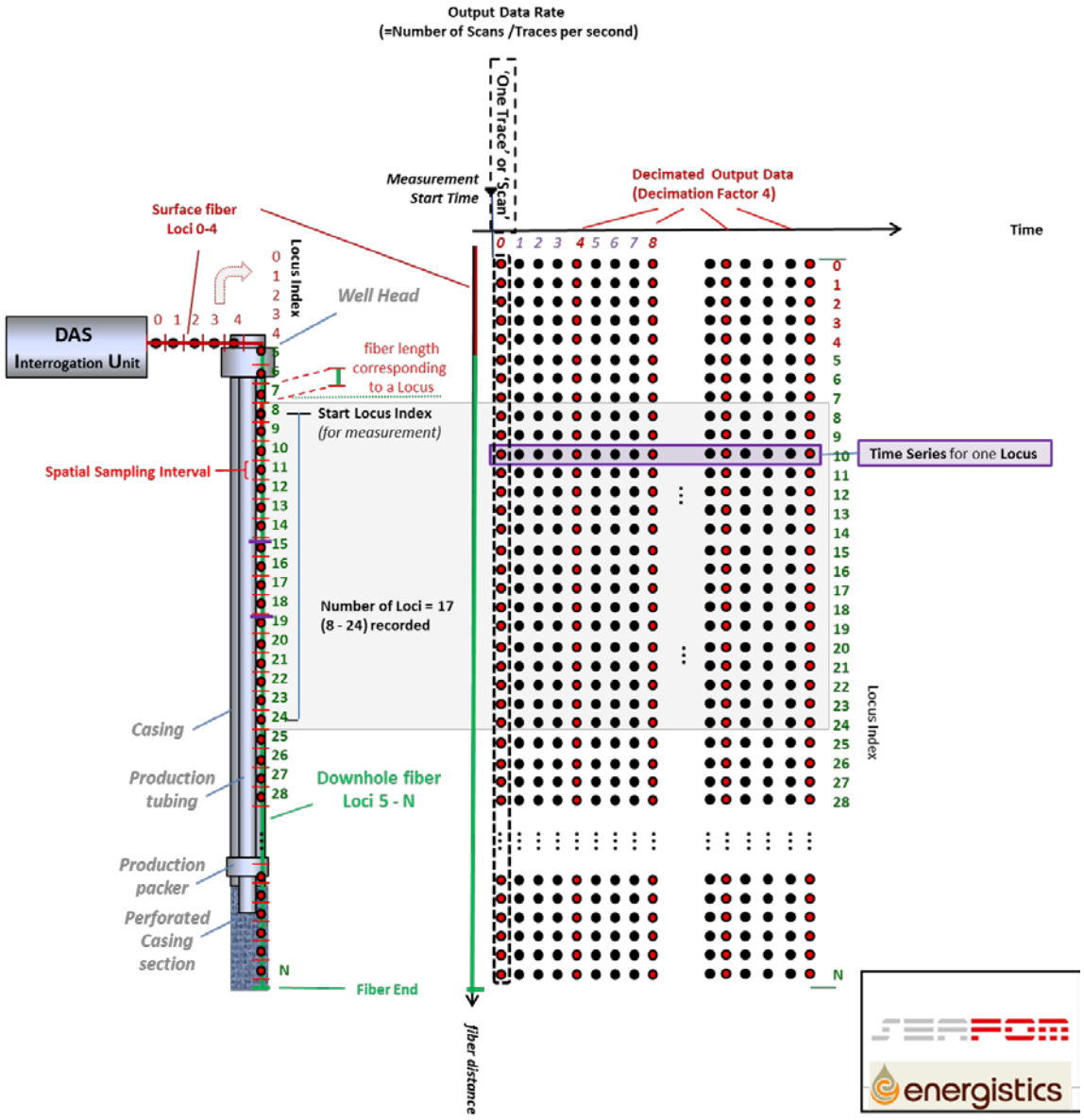


Figure 18–3. DAS installed system loci and raw data time series.

The DAS instrument box (or interrogation unit or IU) is connected to a sensing fiber. The sensing fiber consists of a surface part and a downhole part together presenting the full fiber optical path. The DAS IU sends light pulses in the fiber at a pre-configured rate (interrogation or pulse rate), and samples the backscattered light creating an ensemble of N loci samples along the fiber. Such an ensemble is often referred to as a "trace" or "scan," and is shown as columns of dots in the figure. The time at which a trace is collected is indicated by the measurement start time. A DAS IU interrogating the full fiber outputs a maximum of N x pulse rate "raw" DAS samples per second. Each locus sample is shown as a dot in Figure 18–3.

In many cases the DAS IU is configured such that it decimates the output by an integer factor *M* and only outputs every *M* th trace. The rate at which scans are output by the interrogator is called the "output data rate." The red dots in Figure 18–3 show such a decimation example (decimation factor 4).

Further, the DAS IUs often collect only measurements for a subset of the loci along the fiber. In Figure 18–3, the shaded grey box indicates such a scenario where only loci 8 to 24 are collected. In the figure, “start locus index” and “number of loci” describe which subset is collected. To ensure a unique numbering system, the first locus at the interrogator has by definition index ‘0’.

Measurement samples output for a single locus can form a time series of samples, represented by rows of dots in the figure.

Figure 18–4 introduces a number of additional concepts when processing raw data into frequency bands and spectra using Fourier transforms. An example of the raw data is shown in box ‘A’ containing a representation of a sine wave exciting a single channel (locus) along the fiber (left plot) and the corresponding waterfall DAS raw data plot (right plot). When we now apply filters to this raw data, one filter which includes and one that excludes the sine wave’s frequency, then we obtain two processed filtered images shown in box ‘B’ referred two as processed DAS Frequency Band Extracted (FBE) data. We can also calculate the full signal frequency spectrum for each DAS channel using a Fast Fourier Transform shown in box ‘C which is referred to as Spectrum data.

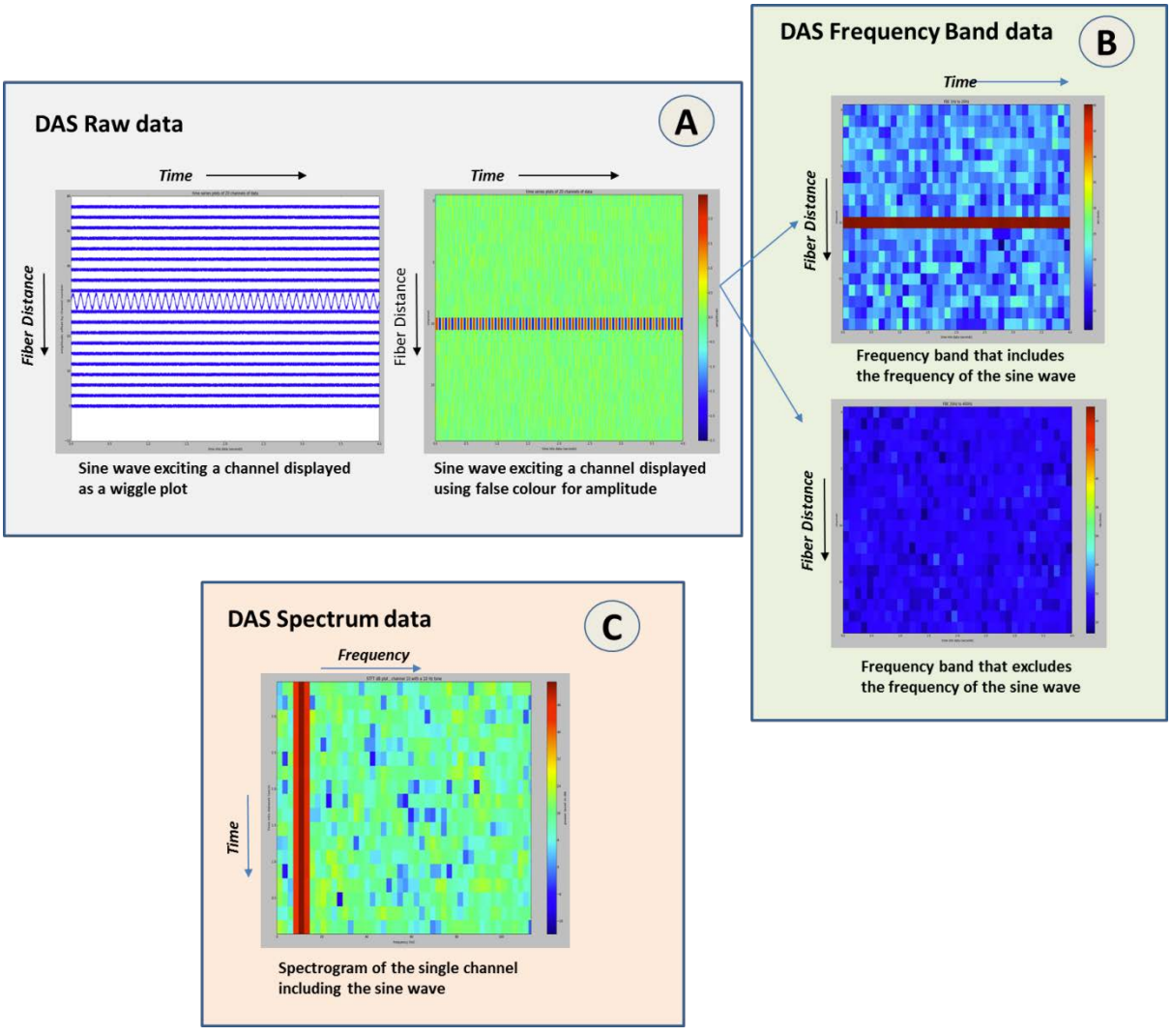


Figure 18–4. DAS raw and processed data types. (A) DAS raw data: a sine wave exciting a single locus along the fiber (left) and the resulting waterfall plot (right). (B) DAS processed FBE data: the top plot shows a filtered version of the raw data containing the sine wave frequency, whereas the bottom plot shows a filtered

version without the sine wave frequency (C) DAS processed spectrum data: frequency spectrum plotting as a function of time.

18.2 DAS Data Life Cycle

Figure 18–5 is a high-level view of the entire DAS workflow, from acquisition, through processing, use and storage.

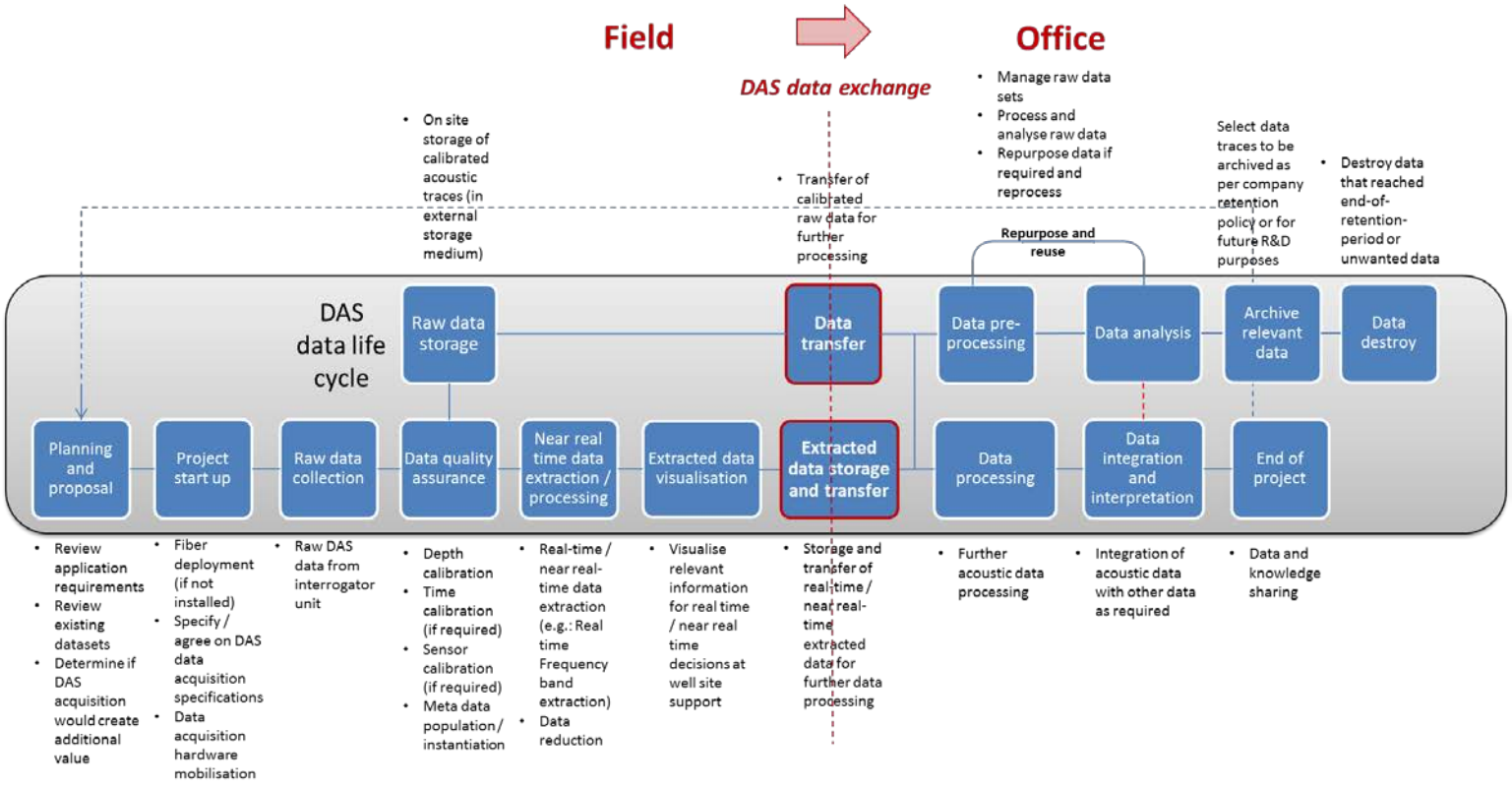


Figure 18–5. High-level DAS workflow, from the field to the office.

19 DAS: Data Model

This chapter explains the DAS data model, provides a brief overview of the technology used to implement DAS, and explains how the technology maps to the data model. Specifically, it:

- Explains the DAS equipment data model, including the XML data objects and options for transferring the fiber optical path and DAS instrument box.
- Explains the acquisition data model, which includes:
  - Specific optical path and instrument box for a particular acquisition
  - The acquisition array data and options for how it can be stored

For more information:

- For a walkthrough of the worked example included in the DAS download, see Chapter 20.
- For examples of other DAS data transfer configurations, see Chapter 19.5.

19.1 Data Model and Technology Overview

Figure 19–1 **i**s a high-level diagram of the set of DAS data objects, which includes definitions of the optical path and the DAS instrument box, and an acquisition data set. Each of these components is explained in more detail in the sections below.

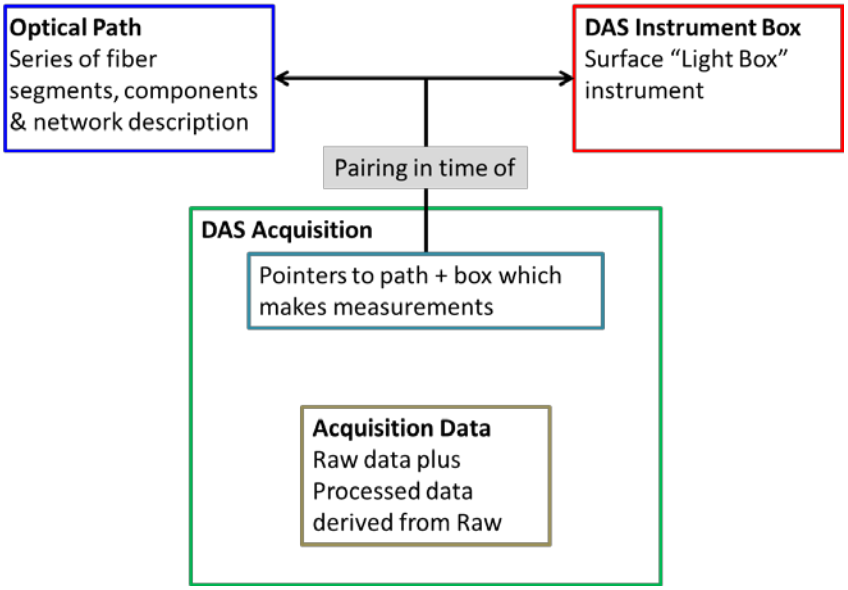


Figure 19–1. The basic DAS installed system is composed of an optical path and instrument box which produces the DAS acquisition data.

Object	Define Using this Schema	For More Information, See:
Optical Path	Fiber Optical Path	Section 14.3
DAS Instrument Box	DAS Instrument Box	Section 19.2
DAS Installed System *(to represent the path+box pairing)*	DAS Acquisition	Section 19.4
DAS Acquisition	DAS Acquisition	Section 19.5

19.1.1 Technology

Like all Energistics data objects, developers implement the DAS data object so that software can read and write the common format; they do this using the key technologies in the Energistics Common Technical Architecture (CTA). For more information, see the *CTA Overview Guide* and other CTA resources, which are included in the PRODML download.

The main technologies are briefly described here. For important information on DAS usage of these technologies, see Section 19.1.1.5.

19.1.1.1 UML for Data Modeling and XSD Generation

Energistics uses the Unified Modeling Language™ (UML®), implemented with Enterprise Architect (EA), a data modeling software tool, to design PRODML, including DAS. The DAS download includes an XMI file of the UML model; the XMI file can be implemented into any UML modeling tool.

For an overview of the DAS data object UML models, see Section 19.6.

19.1.1.2 XML

Each data object (for example, a DAS acquisition, optical path or DAS instrument box) is defined by an XSD file and is stored as an XML file. XML is used because of its portability and ability for humans to read and understand it. Common design patterns leverage the CTA, which promotes integration across all Energistics standards.

For more information, see the *CTA Overview Guide*.

19.1.1.3 HDF5

XML is not very efficient at handling large volumes of numerical or array data, so for this purpose Energistics uses the Hierarchical Data Format, version 5 (HDF5), which is a data model, a set of open file formats, and libraries designed to store and organize large amounts of numerical/array data for improved speed and efficiency of data processing.

DAS uses HDF5 to store both raw and processed measurement data. HDF5 is also part of the CTA, and its general use in Energistics is described in the *CTA Overview Guide*.

For more information about how to use HDF5 with DAS, see Section 19.5.

19.1.1.4 EPC

The Energistics Packaging Conventions (EPC) is a set of conventions that allows multiple files to be grouped together as a single package (or file), which makes it easier to exchange the many files that may make up a model. **Figure 19**–2 shows a conceptual model of how EPC works for DAS.

EPC is an implementation of the Open Packaging Conventions (OPC), a commonly used container file technology standard supported by two international standards organizations. Essentially, an EPC file is a .zip file. You may open it and look at its content using any .zip tool. EPC is also part of the CTA, and how it works is described in the *EPC Specification*.

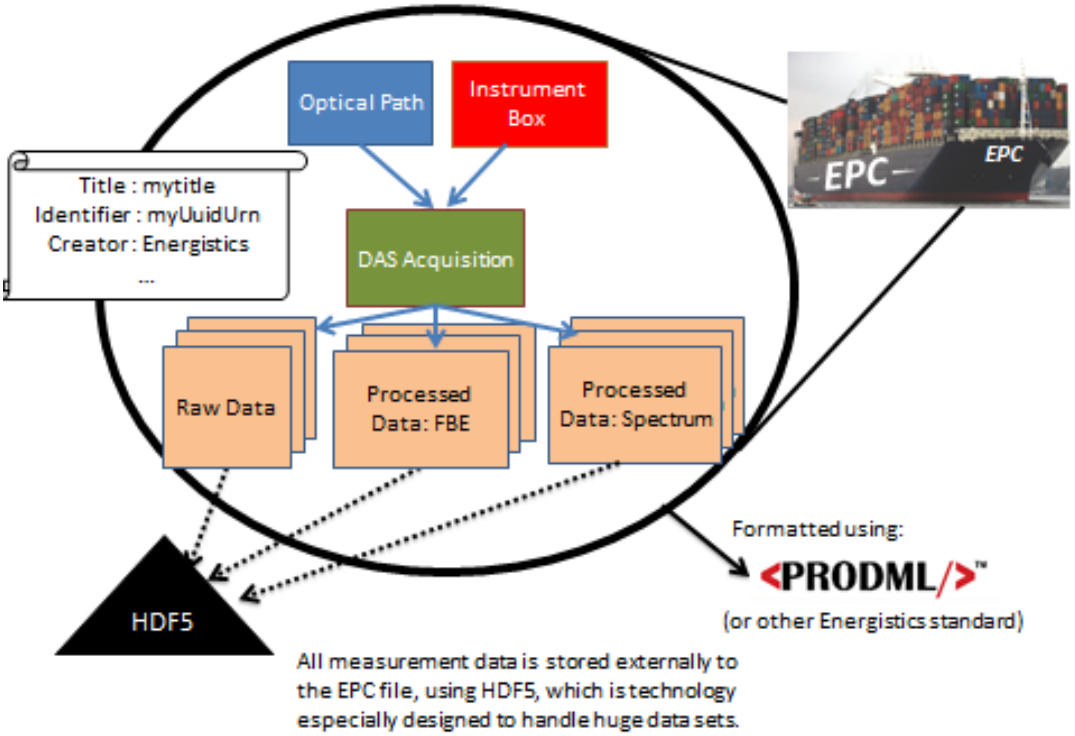


Figure 19–2. Diagram showing how EPC provides the technology to group together related files and exchange them as a single package. (Container ship photo from Wikipedia: http://fr.wikipedia.org/wiki/Fichier:CMA_CGM_Marco_Polo_arriving_Port_of_Hamburg_-_16._01._2014.jpg. **Licence :** Créative Commons paternité – partage à l'identique 3.0 (non transposée))

19.1.1.5 Important Usage of these Technologies for DAS

- **DAS data objects** use the following XSDs:
  - The optical path is represented using the PRODML Fiber Optical Path, which is shared with the DTS (Distributed Temperature Sensing) schemas (see Section 14.3).
  - All other DAS concepts use schemas developed specifically for DAS.
- **Metadata**. All of the metadata about the DAS Acquisition—but not the arrays of data—are stored in XML files. The metadata also is stored in the HDF5 files, along with the arrays of numbers. The advantages of having this data in the XML include:
  - XML has built-in schema validation, and all Energistics XML standards work on the basis that the standard is represented by XSD schemas. HDF5 does not include schema-based validation. This means that the DAS XML files can be checked for validity by simple schema validation built in to most XML tools and editors.
  - The XML files are small and can be opened in any editor, including a text editor, which may make it much easier to figure out what the content of the set of acquisition files is, compared to opening all the potentially huge HDF5 array files. All the metadata can sit in one small XML file, while the actual measurement values are in many large HDF5 files.

  The HDF5 files also contain the metadata. The metadata is duplicated from the XML files because it is possible that the HDF5 files— which can easily fill several hard drives with raw data—may become physically separated from the small XML file during transit. Having the metadata in each HDF5 file enables files to be assembled back together as a coherent set.

  For more information about HDF5 configuration options for DAS, see Section 19.5.3.

- **EPC.** The *EPC Specification* states that HDF5 files may be stored either inside or outside of the EPC file; however, DAS requires that HDF5 files be stored outside the EPC file.
- **Relationships between data object.** The XML content is also used to provide the links between the metadata and the data arrays, through Energistics data object reference, which gets implemented in the EPC file.

19.2 Defining the Optical Path

The Optical Path is a shared object between the DTS and DAS data models. It is described Section 14.3.

19.3 Defining the DAS Instrument Box

The DAS Instrument Box object is shown in **Figure 19**–3. This represents the hardware equipment located at the site that is responsible for generating DAS measurements. The DAS instrument box is usually located next to the facility element for which DAS surveys are taken (a well or a pipeline, for example). It consists of, among other devices:

- An optoelectronic instrument to which the optical fiber is attached. This instrument contains:
  - a controllable light source
  - optical switches
  - photonic detection devices
- In some cases, a computer or server is connected to the instrument that is responsible for capturing the measurements and initiating the data transmission.

Hardware setup can vary widely from one deployment to the next and also depends on the vendor and the make/model of the hardware used. The data schema was created to capture the relevant parameters of the installation without worrying about how the hardware is physically laid out. Attributes available for the instrumentation cover areas such as:

- Make/model of the instrument
- Number of channels
- Software version
- Factory calibration information
- Calibration parameters

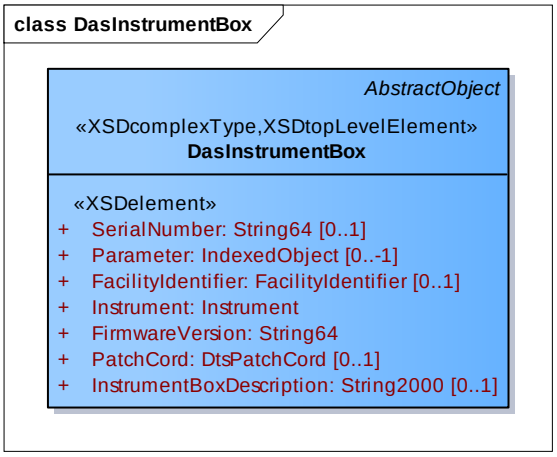


Figure 19–3. DAS Instrument Box model. Note that the Instrument element contains further generic details about the hardware.

19.4 Defining the DAS Installed System

A DAS acquisition system consists of a pairing of an optical path and a DAS instrument box, which are used to acquire DAS measurements. Because there are several ways in which an optical path and a DAS instrument box can be combined to make measurements, the DAS Acquisition object contains a reference to a path and a box from which the measurement was generated. This pairing of fiber optical path and DAS instrument box is known as a DAS Installed System. For examples of the combinations of Optical Path and Instrument Box which can be represented, see Section 14.5.

19.5 Defining the DAS Acquisition Data (Raw and Processed Data)

One DAS acquisition job may consist of many data arrays, including both raw and processed (FBE and spectra) data. Because DAS arrays are so large, HDF5 is used to store and manage the measurement data arrays.

NOTE: DAS arrays may be so large that they are split across multiple HDF5 files. XML is used to store all the metadata, and this metadata is repeated in the HDF5. Comments below about metadata refer to both formats.

19.5.1 High-Level Conceptual Model

Figure 19–4 shows the high-level conceptual model for how raw and processed data are related. The orange “Arrays” boxes in the diagram refer to data arrays. In this model, a DAS Acquisition may have multiple raw arrays (0..*) and one processed array set, but the processed array set may have multiple spectra and FBE (0..*) arrays.

There are 3 types of data: raw and two types of processed data: spectra and FBE.

Each type of data has two parts:

- the “data” array (the values of the measurements across time and across loci for raw and for FBE, and the values of Fourier transformed data across frequencies for the spectra)
- the “time” array (the times which are common to all loci)

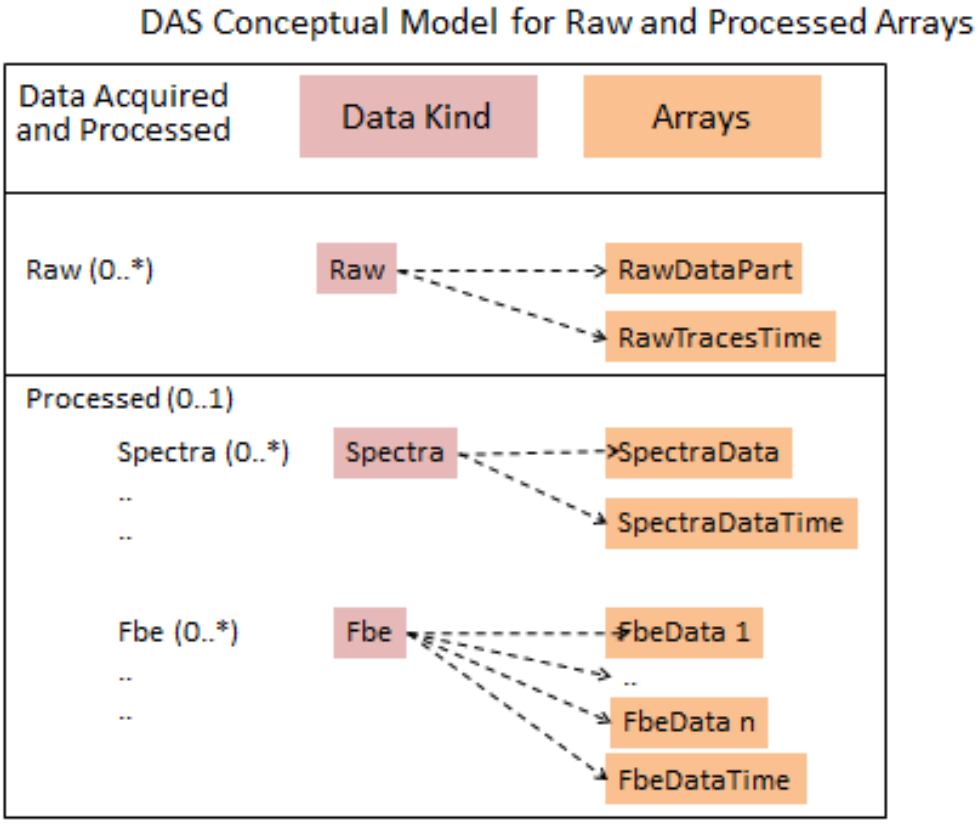


Figure 19–4. DAS data can consist of both raw and processed (spectra and FBE) arrays. This figure shows conceptually how different data arrays are related.

In addition to the array of values, attributes of the groups and arrays are used to provide necessary ancillary and metadata. The HDF5 file also contains Custom and Calibration data. Custom contains attributes for the whole acquisition or any data group (e.g., raw, FBE or spectra). The custom attribute can be used by individual vendors to specify vendor specific or customer specific requested parameters. Calibration is data to locate loci along the optical path, in relation to the facility being measured.

At the root level of the HDF5 file, attributes refer to attributes of the whole DAS acquisition, as seen in the blue class in Figure 19–8 (page 165), including the DAS time (time zone for the acquisition).

19.5.2 Overview of the HDF5 File

NOTE: The screen shots of HDF5, XML and EPC files in the following sections are taken from the worked example, which has:

- 1 raw array, which is split over 2 physical HDF5 files
- 1 processed FBE data set containing 2 bands of FBE data
- 1 processed spectrum

The cardinality for this data is listed here. Note that the pattern shown here remains the same but the specific cardinalities of arrays changes for other examples:

- Custom
- Calibration
- RAW group is 0..* (1 in this example)
- FBE groups is 0..* (1 in this example).

- FBE bands within an FBE group is 1..* (2 in this example) 1
- Spectra is 0..* (1 in this example)

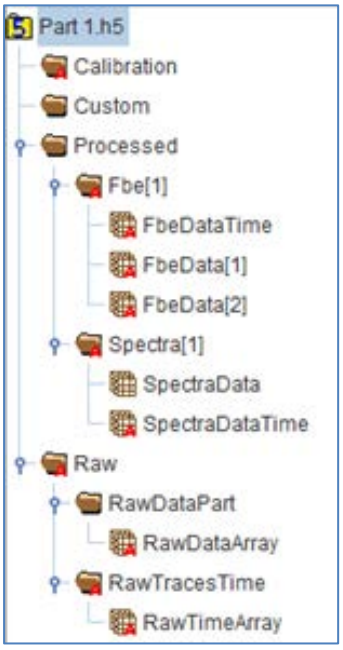


Figure 19–5. DAS example HDF5 file structure. Use of DAS requires data stored in HDF5 to contain the structure and naming conventions shown in the figure.

19.5.3 HDF5 File Array Configuration Options

DAS data arrays can be very large (especially raw arrays) and can exceed the capacity for individual HDF5 files (i.e., exceed the space available on a disk). For this reason, it is possible to split arrays across multiple physical HDF5 files. However, multiple arrays may be transferred within one DAS acquisition HDF5 file.

There are several options for how to store these combinations of DAS array data in HDF5 files, which are listed in the table below and shown in **Figure 19**–6 and **Figure 19**–7. Other combinations are of course possible.

Configuration Options	Description of Contents
Processed in separate file	• One or more files that contain a raw array split across them. • The processed data is in its own file.
Processed with own raw data	• One or more files that contain a raw array split across them. • Each file contains its sub-section of the raw array and that sub-section's corresponding processed array(s.
Hybrid of 1 and 2	• One or more files that contain a raw array split across them. • The processed array(s) for the whole of the raw array are in one of the files along with one sub-section of the raw array.
Raw only	• One or more files that contain a raw array split across them and no processed arrays. • Typically used when one company does the acquisition and another company does any subsequent processing.

Configuration Options	Description of Contents
Processed only	• One file containing the processed arrays, with no raw arrays. • Typically used when one company does the acquisition and another company does the processing.

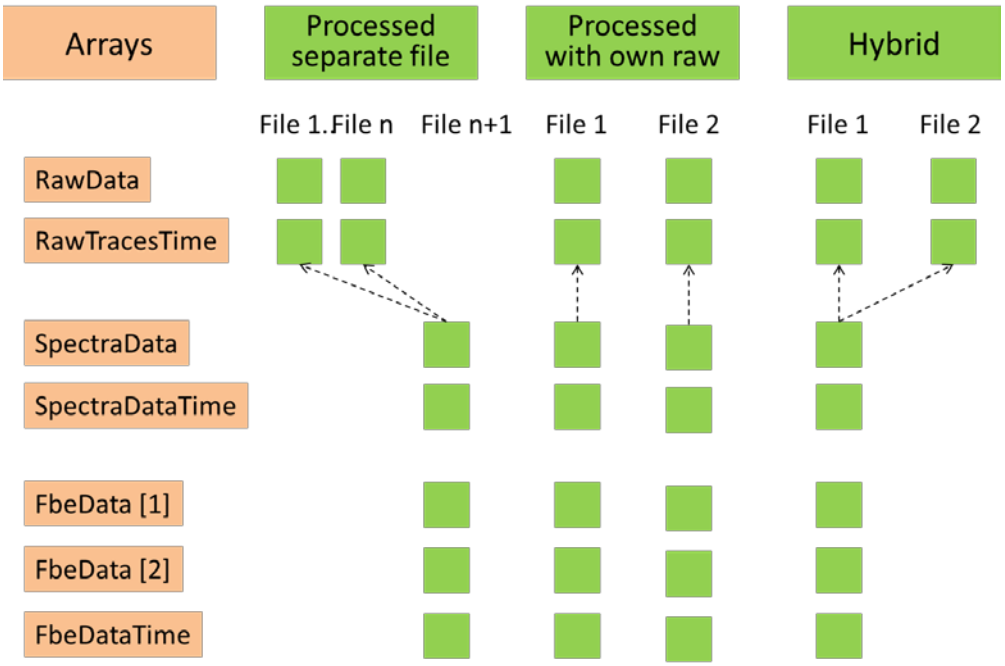


Figure 19–6. Possible HDF5 (.h5) file configurations for use with DAS (1 of 2).

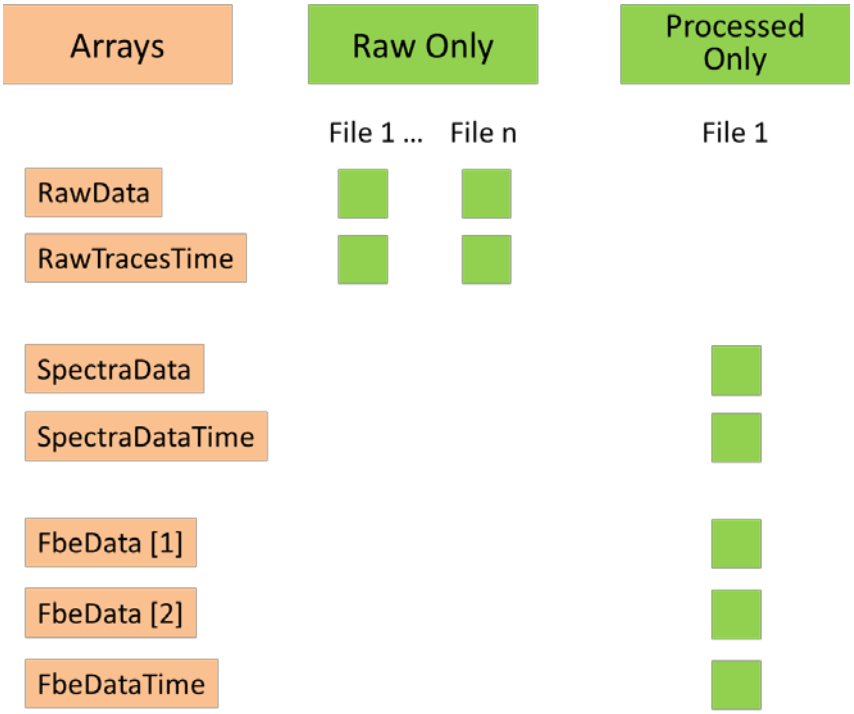


Figure 19–7. Possible HDF5 (.h5) file configurations for use with DAS (2 of 2).

19.5.4 HDF5 Structure and Naming Conventions (Required for DAS)

Figure 19–5 shows the required structure and naming conventions of an HDF5 file required by DAS.

HDF data groups are available for the following data types. The XML contains elements with the same names and attributes (but does not contain the array data):

Group	Description
Raw	Zero to many raw data arrays are allowed. The groups must be named “Raw” followed by consecutive numbers in square brackets, e.g., “Raw[1]”with the additional groups and data sets (arrays) shown in the figure.
Processed	One processed group is allowed, but that group may have zero to many FBE and spectra groups. These individual Groups must have the name “Fbe” or “Spectra” followed by consecutive numbers in square brackets, e.g., “Fbe[1]”. The Fbe[n] Group contains multiple arrays for the frequency bands and these must be named FbeData[m] with “m” identifying the band. There is one FbeDataTime array for the times common to all FbeData[ ] bands. The Spectra[n] Group contains two arrays, for data and for time, as shown.
DAS custom	Custom data is any service–provider-specific customization parameters, which service providers can provide as required. For more information, see Section 19.7.5. If you look at the UML diagram (Figure 19–8), you can see that any custom data can also be added to the individual data groups (raw, FBE, and spectra).
DAS calibration	Calibration data is a mapping of loci-to-fiber and facility distance along the optical path for the DAS acquisition. For more information, see Section 19.7.3.

19.5.5 Optional Arrays and Attributes in HDF5

- If an attribute is mandatory, it must have a value. All mandatory attributes musts have a valid entry (which means either with a valid attribute value or a defined NULL value).
- Non-mandatory attributes (marked in the attached Schema with [0..1]) or groups (marked with 0..*) should be omitted from the XML or h5 files if they are not used (i.e., empty).

This means, in practice, it is allowable *not* to include groups like raw, processed, spectra and Fbe. However, groups like Das Acquisition, DAS Instrument Box and attributes that are marked as mandatory in optional groups which are used (such as NumberOfLoci, RawDataTime, etc.) *must* always be properly populated if the associated group is used.

19.6 Overview of the DAS Acquisition UML Model

This section provides an overview of the DAS Acquisition data model using UML diagrams. For a complete description of all elements in the UML model/DAS XSDs, see the PRODML UML model or the *PRODML Technical Reference Guide*.

Figure 19–8 shows the content of the XML DAS Acquisition file. The data elements in this XML schema are repeated in the HDF5 file as HDF attributes, distributed through the file associated with the group or array concerned. For more information about the DAS acquisition, see Section 19.7.

Figure 19–9 shows the DAS data arrays that are used to store the DAS raw, DAS FBE, DAS spectra samples and their corresponding sample times. Because of the volume of data, these arrays are *only* stored in the HDF5 files. For more information about DAS data arrays, see Section 19.8.

Detailed definitions of individual attributes can be found in Section 22.1.

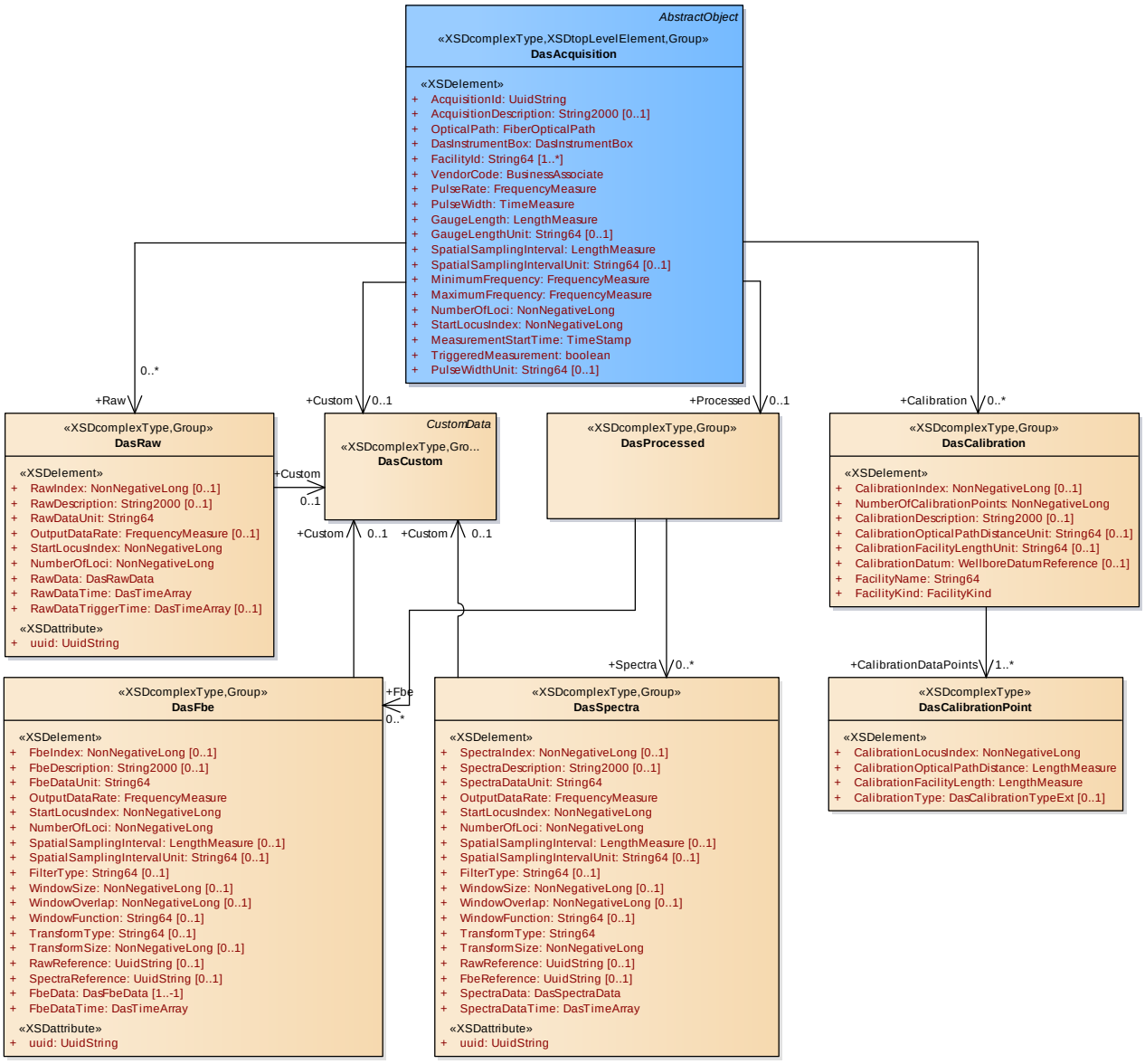


Figure 19–8. UML diagram of XML content of DAS Acquisition file (described in Section 19.7below).

19.7 DAS Acquisition Object Details

The DAS Acquisition object (see Figure 19–8) contains metadata about the DAS acquisition common to the various types of data acquired during the acquisition, which includes DAS measurement instrument data, fiber optical path, time zone, and core acquisition settings like pulse rate and gauge length, measurement start time, and whether or not this was a triggered measurement.

19.7.1 DAS Instrument Box

Note that the DAS Instrument Box data elements (DasInstrumentBox) are described in the DAS Instrument Box XML file and are not repeated in the DAS Acquisition XML or HDF5 files. A reference to the DAS Instrument Box data object is provided. This object is expected to be one of those within the EPC container (see Section 19.1.1.4 and example in Section 20.2).

19.7.2 Fiber Optic Path

Note that the Fiber Optical Path data elements are described in the Fiber Optical Path XML file and are not repeated in the DAS Acquisition XML or HDF5 files. A reference to the Fiber Optical Path data object is provided. This object is also expected to be one of those within the EPC container (see Section 19.1.1.4 and example in Section 20.2).

19.7.3 DAS Calibration

The DAS Calibration objects (DasCalibration) contain a mapping of loci-to-fiber for each facility along the optical path for the DAS acquisition. For example when a well is interrogated the optical path typically consists of a surface fiber and a downhole fiber in the well each described by their own Calibration object.

Each Calibration object contains details about the optical path segment, facility and datum and then has the actual calibration points. The facility type can be: *generic*, *pipeline*, or *well*. “Generic” can be used for things like connecting cables. For a well, the well datum can be used to state where the datum for measured depth originates.

The CalibrationType can be used to differentiate between different calibration points. The user is free to define their own calibration types, but three special types have been pre-defined:

- A so-called ‘**tap test**’ is a test where an operator uses a (spark-free!) hammer to create a vibration on a known location like the well head to provide the interpreter a known point for depth calibration.
- The ‘**last locus to end of fiber**’ indicates that this calibration point contains the fiber distance from the last locus to the end of the fiber. This is a useful calibration point for permanent downhole installations. The operator can use this measure to create exactly the same interrogation interval along the fiber after a DAS instrument or surface cabling has been changed out of. Note that for this calibration type OpticalPathLength doesn’t mean anything and should be set to a NULL value.
- **Locus calibration** is where a single locus along the optical path is mapped to a length along one of the facilities being measured.

The calibration points are provided in the XML as a set of repeating elements, and in the HDF5, as an array of DAS calibration data. Either consists of four elements: a locus index, the corresponding optical path distance and facility length, and a description of the calibration type. The user can provide as many calibration points as necessary per calibration object.

19.7.4 DAS Raw

The DAS Raw object (DasRaw) contains the attributes of raw data acquired by the DAS measurement instrument. This includes the unit in which the raw data was measured, the indices of the loci where the raw data was acquired along the fiber optical path (start locus and number of loci), and information about times and (optional) triggers. Note that the actual raw data samples, times and trigger times arrays are not present in the XML files but only in the HDF5 files because of their size. The XML files only contain references to locate the corresponding HDF files, which contain the actual raw samples, times, and (optional) trigger times.

The array types to store the data samples and corresponding times arrays in the HDF files are described in the Section 19.8.

19.7.5 DAS Custom

The DAS Custom object (DasCustom) contains service–provider-specific customization parameters. Service providers can define the contents of this data element as required. This data object has intentionally not been described in detail to allow for flexibility. DasCustom can be inserted as a separate data group under DasAcquisition, but also under the DasRaw, Das Fbe and Das Spectra. Note that these objects are optional and if used the service provider needs to provide a description of the data elements to the customer.

19.7.6 DAS Processed

The DAS Processed object (DasProcessed) contains data objects for processed data types and has no data attributes. Currently only two processed data types have been defined: the frequency band extracted (FBE) and spectra. In the future other processed data types may be added.

Note that a DasProcessed object is optional and only present if DAS FBE or DAS spectra data is exchanged.

Multiple FBE and spectra groups may be used with a DasProcessed object. Multiple groups of FBE bands could be exchanged, for example, one group with linearly distributed frequency bands (e.g. two bands: 0.01-10.00 Hz and 10.01-20.00 Hz) and another group with logarithmically distributed frequency bands (e.g. three bands: 0.1-1.0 Hz, 1.0-10.0 Hz and 10.0-100.0 Hz). In the HDF5 file these group should then be named Fbe[1] and group Fbe[2] with the number added in square brackets [n]. If there is only one FBE or Spectra group, then the brackets and number can be omitted. In the XML file this is not necessary. An example is found in Figure 19–5.

19.7.7 DAS FBE

The DAS FBE object (DasFbe) contains the attributes of FBE processed data. This includes the FBE data unit, location of the FBE data along the fiber optical path, information about times, (optional) filter related parameters. Note that the actual FBE data samples and times arrays are not present in the XML files but only in the HDF5 files because of their size. The XML files only contain references to locate the corresponding HDF files containing the actual FBE samples and times.

The array types to store the data samples and corresponding times arrays in the HDF files are described in Section 19.8.

19.7.8 DAS Spectra

The DAS Spectra object (DasSpectra) contains the attributes of spectra processed data. This includes the spectra data unit, location of the spectra data along the fiber optical path, information about times, (optional) filter related parameters, and UUIDs of the original raw from which the spectra file was processed and/or the UUID of the FBE files that were processed from the spectra files. Note that the actual spectrum data samples and times arrays are not present in the XML files but only in the HDF5 files because of their size. The XML files only contain references to locate the corresponding HDF files containing the actual spectrum samples and times.

The array types to store the data samples and corresponding times arrays in the HDF files are described in Section 19.8.

19.8 DAS Array

The DAS Array (DasArray) schema (Figure 19–9) defines the raw, FBE, spectrum sample arrays and the corresponding times arrays used in the HDF5 files.

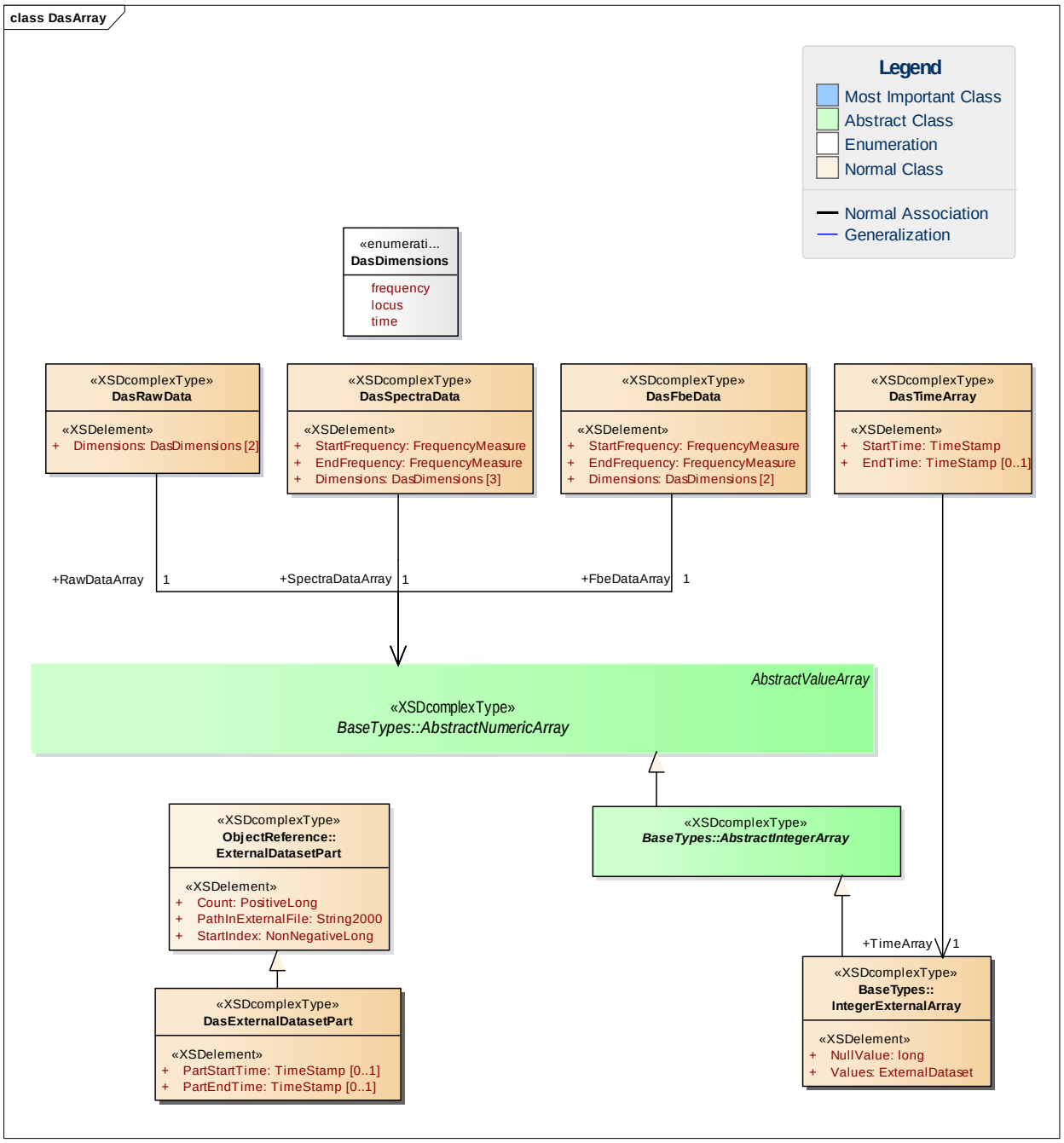


Figure 19–9. UML diagram of HDF5 DAS arrays for raw samples, FBE samples, spectra samples and corresponding times (described in Section 19.8 above)

Further the DasArray schema defines ExternalDataSetPart and derived class DasExternalDatasetPart which help the user to keep track of which HDF5 file contains which part of the DAS sample data. Use of these parts is necessary for very large acquisitions where the DAS data samples array may become so large that must be stored in multiple HDF5 files.

The raw, FBE and spectra arrays have a Dimensions attribute of type array that specifies the ordering. The array element with the highest index contains the fastest running index. For the indexes, the three options are 'locus', 'time' and 'frequency'. Typically the acquisition system will write raw data with locus as the fastest running index (e.g., dimensions array = {'time','locus'} because it acquires all active loci on the

fiber for each pulse and writes these out trace after trace. However, the ordering may differ in these cases:

- The raw data array is a 2D array containing data samples for each 'scan' (array size = number of scans x NumberOfLoci).
- The FBE data array is a frequency band filtered version of the raw data and also a 2D array (array size = number of scans x NumberOfLoci).
- The spectrum data array is a 3D array because each Discrete Fourier Transform (DFT) provides N points, with N equal to the TransformSize (array size = number of scans x NumberOfLoci x TransformSize).

In addition, the FBE and spectra arrays have StartFrequency and EndFrequency attributes:

- For the FBE data, these attributes contain the start and end frequency of the frequency band.
- In the spectra data set, these correspond to the minimum and maximum frequency in the spectra.

The **Times** arrays contain the 'scan' or 'trace' times at which the raw, FBE and spectrum arrays were acquired or processed.

Figure 18–3 shows traces/scans as vertical columns of dots.

Note that these Times arrays contain the times in ***Unix time*** (microseconds passed since 1 January 1970).

19.8.1 DAS Raw

Two-dimensional (loci and time) array containing raw data samples acquired by the DAS acquisition system.

19.8.2 DAS FBE Data

Two dimensional (loci and time) array containing processed frequency band extracted data samples. This processed data type is obtained by applying a frequency band filter to the raw data acquired by the DAS acquisition system. For each frequency band provided, a separate DAS FBE data (DASFbeData) array object is created.

19.8.3 DAS Spectra Data

Three-dimensional array (loci, time and transform) containing spectrum data samples. Spectrum data is processed data obtained by applying a mathematical transformation function to the DAS raw data acquired by the acquisition system. The array is 3D and contains *TransformSize* points for each locus and time for which the data is provided. For example, many service providers will provide Fourier transformed versions of the raw data to customers, but other transformation functions are also allowed.

19.8.4 DAS Time Array

The Times arrays contain the 'scan' or 'trace' times at which the raw, FBE and spectrum arrays were acquired or processed:

- For raw data, these are the times for which all loci in the 'scanned' fiber section were interrogated by a single pulse of the DAS measurement system.
- For the processed data, these are the times of the *first* sample in the time window used in the frequency filter or transformation function to calculate the FBE or spectrum data.

Figure 2-3 shows traces/scans as vertical columns of dots. Note that these Times arrays contain the times in *Unix time* (microseconds passed since 1 January 1970).

Because these UNIX times are difficult to decipher by a human user, the Times arrays contain a StartTime and EndTime in human-readable format, for example 2015-07-20T07:23.45.678000+00:00 corresponding to *UNIX time* 1437377025678000 in microseconds. StartTime and EndTime always refer to the *full* acquisition recording. If the recording is split over multiple files then DasExternalDatasetPart

attributes PartStartTime and PartEndTime are used to keep track of start and end times of individual recordings (see next paragraph).

19.8.5 DAS External Dataset Part

DAS Raw data recordings are often large in size which requires the recording to be split in parts stored across multiple HDF5 files. To keep track of the contents of each part PRODML classes ExternalDataSetPart and its subclass DasExternalDatasetPart are used.

ExternalDataSet attributes

- **Count**: the size of the partial array
- **PathInExternalFile**: the path to file
- **StartIndex**: start index of the time array in the full dataset at which this partial array starts

The following DasExternalDatasetPart attributes describe the start and end times of the **partial arrays** in human readable format:

- **PartStartTime**
- **PartEndTime**.

Note that DasTimeArray: StartTime and EndTime attributes in the DAS Time Array object always refer to the **full acquisition**.

19.8.5.1 Example

A DAS acquisition records 10 minutes of raw data starting at 1 May 2016 at noon. Because of the large number of loci (5000) and the high OutputDataRate of (1kHz) it is decided to split the data set in two 5 minute recordings and store the raw data in two HDF5 files called *part1.h5* and *part2.h5*.

The Das External Dataset Parts for these files in the XML and H5 will then contain:

Part1.h5

ExternalDataset

Count = NumberOfLoci x OutPutDataRate x time = 5000 x 1000 x 5 x 60 = 1 500 000 000
PathInExternalFile = Raw/RawDataPart
StartIndex = 0

DasExternalDatasetPart:

PartStartTime = 2016-05-01T12:00:00.000000+00:00

PartEndTime = 2016-05-01T12:04:59:999000+00:00

RawDataTime

StartTime =2016-05-01T**12:00:00.000000**+00:00

EndTime = 2016-05-01T**12:09:59:999000**+00:00

Part 2.h5

ExternalDataset

Count = NumberOfLoci x OutPutDataRate x rate = 5000 x 1000 x 5 x 60 = 1 500 000 000
PathInExternalFile = Raw/RawDataPart
StartIndex = 300 000 (the time index of the start of this partial dataset, within the whole array)

ExternalDatasetPart

PartStartTime = 2016-05-01T**12:05:00.000000**+00:00

PartEndTime = 2016-05-01T**12:09:59:999000**+00:00

RawDataTime

StartTime =2016-05-01T**12:00:00.000000**+00:00

20 DAS Worked Example

The DAS download includes an example DAS project, which consists of a set of files containing sample data. The data from this project is used in the examples that appear throughout this part of this guide and is commonly referred to as the "worked example."

This chapter:

- Lists the files that comprise the example
- Provides detailed descriptions and explanations of the worked example, including:
  - The EPC file
  - The XML files contained in the EPC file
  - The DAS acquisition file (which is stored outside the EPC file in HDF5 files)

20.1 Overview of the Example Files

The example file contains a simplified DAS project. To keep the example manageable, only 101 loci are included (compare to the thousands that might be in an actual DAS acquisition), and only 75 samples in time.

The example is composed of these files:

- An EPC file, named *example.epc*, which contains all the XML files comprising the acquisition (for details, see Section 20.2 below).
- Two HDF5 files, which are named *part1.h5* and *part2.h5*. These are external to the EPC "container" (**Figure 20**–1).

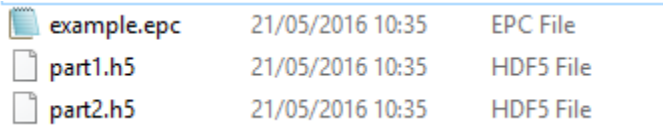


Figure 20–1. Worked example folder contents.

20.2 EPC "Container" File

For more information about EPC, see Section 19.1.1.4 (page 156) and the *Energistics Packaging Conventions (EPC) Specification* referenced in Section 2.4 (page 18).

The EPC file (which is actually a .zip file and can be opened by any zip file tool, e.g., Winzip, if its file extension is changed to .zip for this purpose) (**Figure 20**–2) contains all XML data that comprises a DAS acquisition, and includes:

- Optical Path
- Instrument Box
- DAS Acquisition
- References to the related HDF5 files (which the DAS standard requires to be stored outside the EPC file) using EPC External Part Reference files
- Proxy and relationship files

The files have the name of object type + uuid. This is not mandatory but makes it easier to follow by hand.

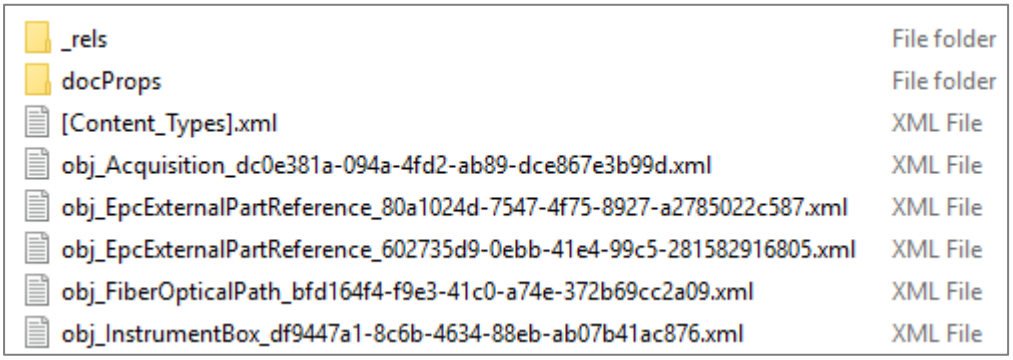


Figure 20–2. Content of the EPC file (which is a .zip file).

20.3 Relationships Between XML Metadata and Arrays

This section provides a detailed walk-through of how the relationships between the various files of the worked example are specified.

Figure 20–3 shows the conceptual data model and HDF5 storage for the arrays in the worked example and **Figure 20**–4 shows how this model is implemented in EPC and HDF5. Note, this is the option referred to as *hybrid* in Section 19.5.3.

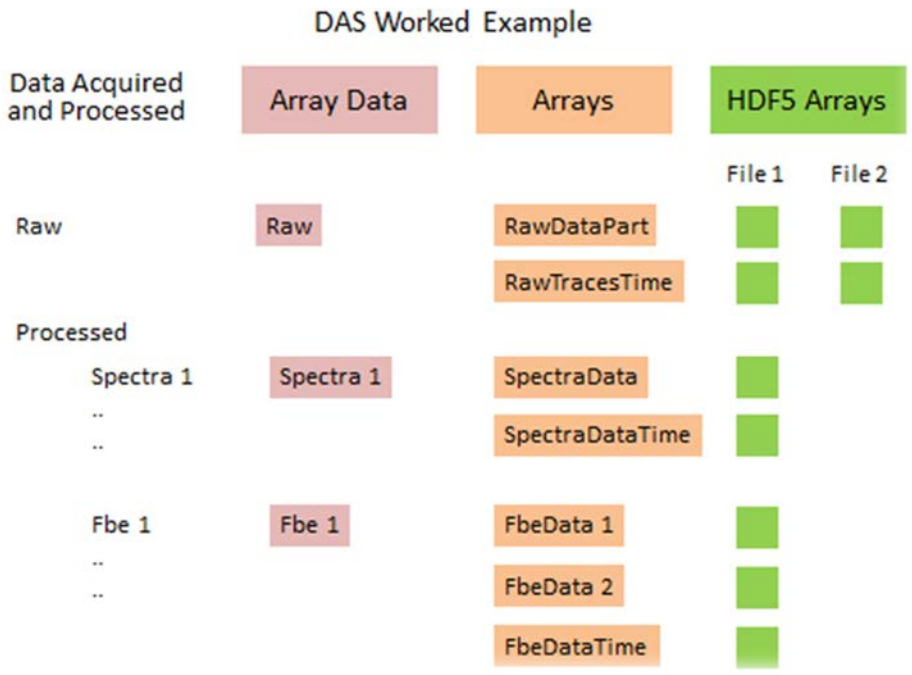


Figure 20–3. DAS worked example conceptual model with HDF5, which is a "hybrid" implementation (from the list in Section 19.5).

The XML files are all contained in the EPC file. The HDF5 files are transferred separately. Thus in the worked example, which shows the arrays split over 2 HDF5 files (see Figure 20–3), there are 3 files in total. See **Figure 20**–1 and the **top** of **Figure 20**–4 also shows these 3 files: one with extension *.epc* and the two HDF5 files with the *.h5* extension.

Opening the EPC file (i.e., unzipping it), shows 5 XML data files: the DAS Acquisition, Instrument Box, and Optical Path (blue colored arrows) and the two EPC external part reference files (red colored arrows) (see **Figure 20**–2 and the **middle** of **Figure 20**–4)).

Relationships to related files are stored in the rels folder. For each data file, there is a corresponding .rels file which stores the relationship of that file to the other files. These files have the extension .xml.rels.

The .rels files specify the relationships among ALL files—the XML files stored *internally* in the EPC file and the *externally* stored HDF5 files (see the bottom of **Figure 20**–4**).**

- Relationships to related files that are stored **internally** in the EPC file (such as the Instrument Box and Optical Path files in this example) are specified using an internal EPC mechanism with a TargetMode "Internal".
- Files that are stored **externally** to the EPC file (such as the 2 HDF5 files in this example) are specified using an EPC mechanism called *external part reference*. Each external file has a corresponding external part reference XML file.

The content of these is explained below.

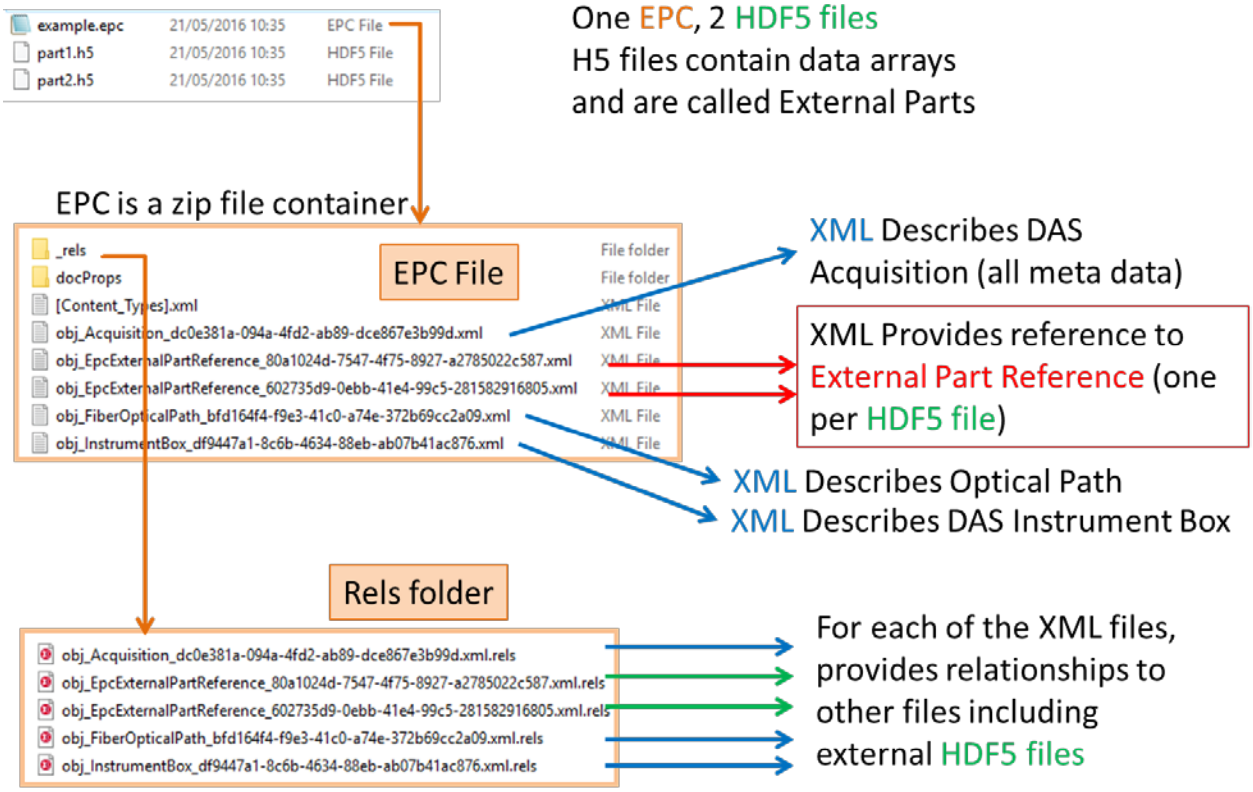


Figure 20–4. The 3 files comprising the worked example (*top*); the content of the EPC file (*middle*)—5 XML files; and the content of the Rels folder within the EPC file—one relationship file per XML data object (*bottom*).

In the main folder of the EPC file, the DAS Acquisition XML file contains the metadata as outlined previously. In the middle of **Figure 20**–5, extracts from this file are shown in a blue box. The UUID of the whole DAS Acquisition is an attribute of this file. The incorporation of the UUID into the file name itself, as shown, is also recommended (see the blue rectangles in the figure).

Similarly, the EpcExternalPartReference files (the red rectangle in the lower part of Figure 20–5) have their own UUID. These files act as proxies for the HDF5 files. These UUIDs are used within the DAS Acquisition XML to reference the external part (HDF5 data) concerned using the element

EpcExternalPartReference within the DAS Acquisition XML. This is shown by the red rectangle in the middle of Figure 20–5.

Because an EpcExternalPartReference refers to an HDF5 file which may contain multiple arrays, every array referenced from within the DAS Acquisition XML must have a specified (unique) path in the HDF5 file. This is shown by the green rectangle in the middle part of Figure 20–5.

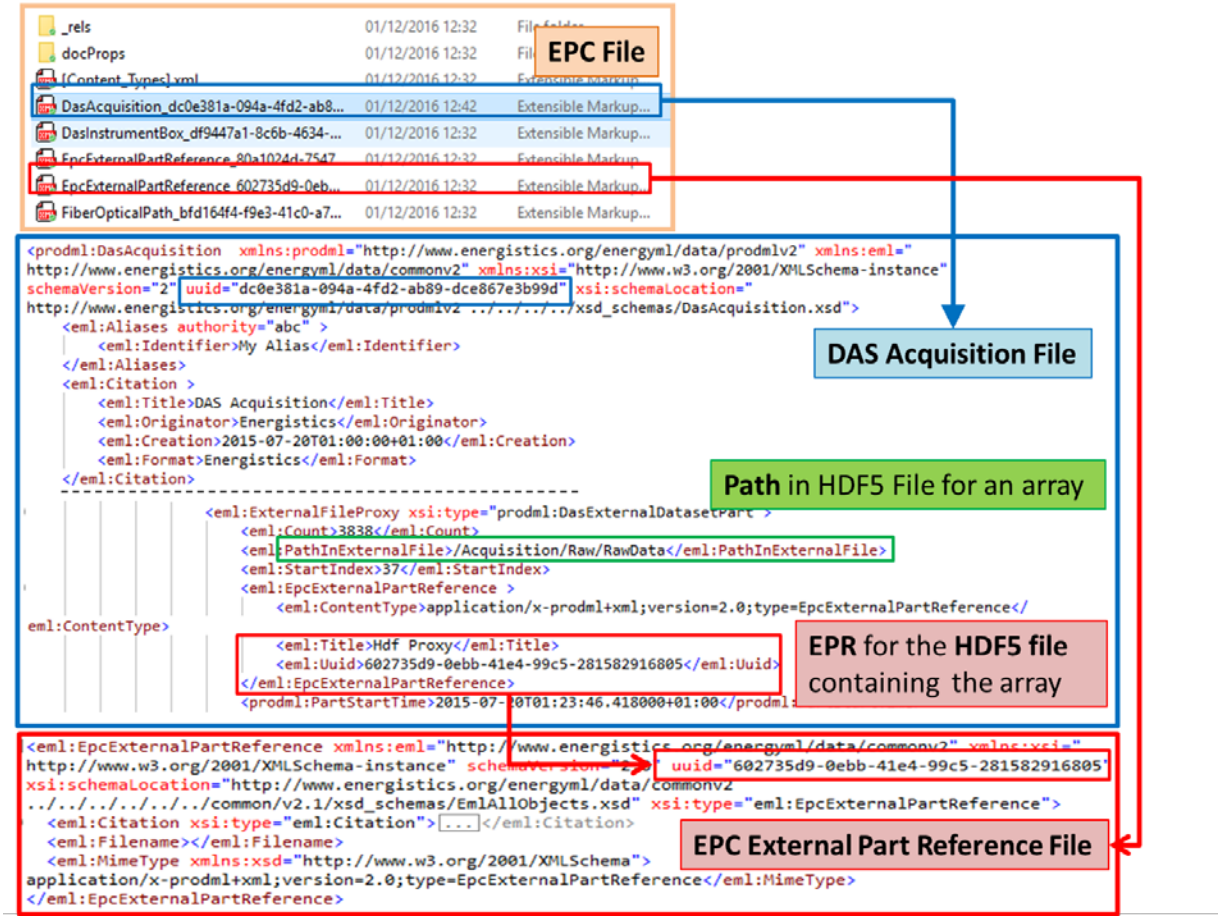


Figure 20–5. The 5 files in the EPC root folder (DAS Acquisition, DAS Instrument Box, Optical Path and two EpcExternalPartReference) (*top*); the DAS acquisition file showing an example of a reference to a single HDF5 array with the path to the H5 file (green) and UUID of EpcExternalPartReference (red) (*middle*); and how this UUID is located in the EpcExternalPartReference XML itself (*bottom*).

The rels folder contains one file (extension .xml.rels) per file contained in the root folder. See **Figure 20**–6, the top snippet, light brown border.

The rels file for the DasAcquisition lists all the relationships for this file. See Figure 20–6, the middle snippet, blue border. There are relationships to two files *internal* to the EPC (Optical Path and Instrument Box) (purple and brick red boxes), and to two *external* EpcExternalPartReference files (red boxes).

The rels file(s) for the EpcExternalPartReference files show a relationship back to the DasAcquisition (blue box showing its UUID), and a relationship to an *external* file (the .h5 file) (green box showing the file name itself). See Figure 20–6, the bottom snippet, red border.

By these means, the references all tie in with each other and the specific path to an array in a specific HDF5 file can be discovered.

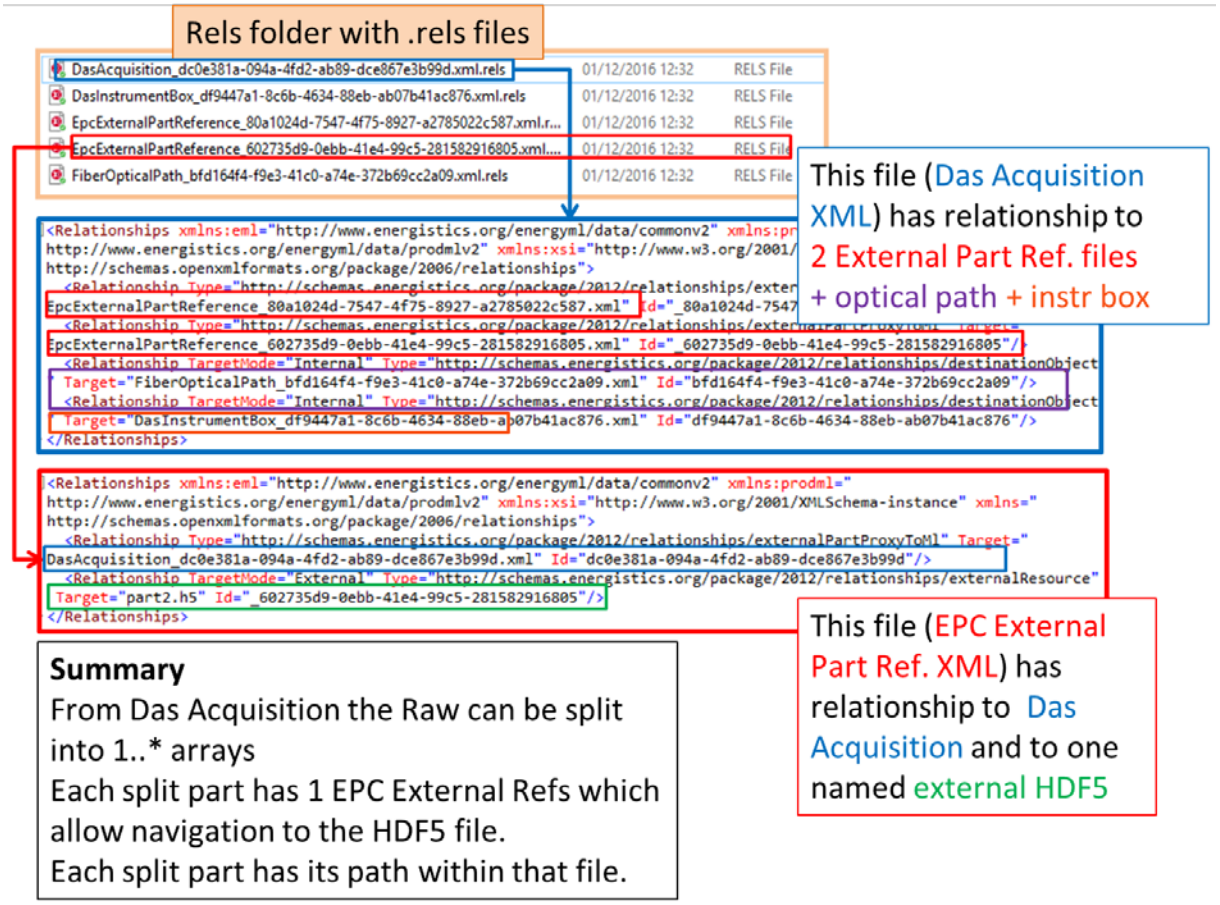


Figure 20–6. Showing the 5 files in the EPC rels folder (one for each data file comprising the set) (*top*); the relationships existing from the DASAcquisition with the two internal XML files, plus two EpcExternalPartReferences (*middle*); and the relationships existing from the EpcExternalPartReference to its parent DAS acquisition and to the physical H5 file (*bottom*).

When arrays are split over multiple HDF5 files (as they are in the worked example), then a single logical array in the DasAcquisition XML (e.g. for a Raw array) contains 2 ExternalFileProxy elements. **Figure 20**–7 shows the same worked example and shows how the Count and StartIndex elements are used to define the partitioning of the array across the two files. The paths for both arrays are shown here. To find the physical HDF5 file name, look in the rels file for the EpcExternalPartReference, as explained above.

The DasExternalFile proxy elements PartStartTime and PartEndTime show the start and end times of the partial Raw DAS data array stored in each HDF5 files.

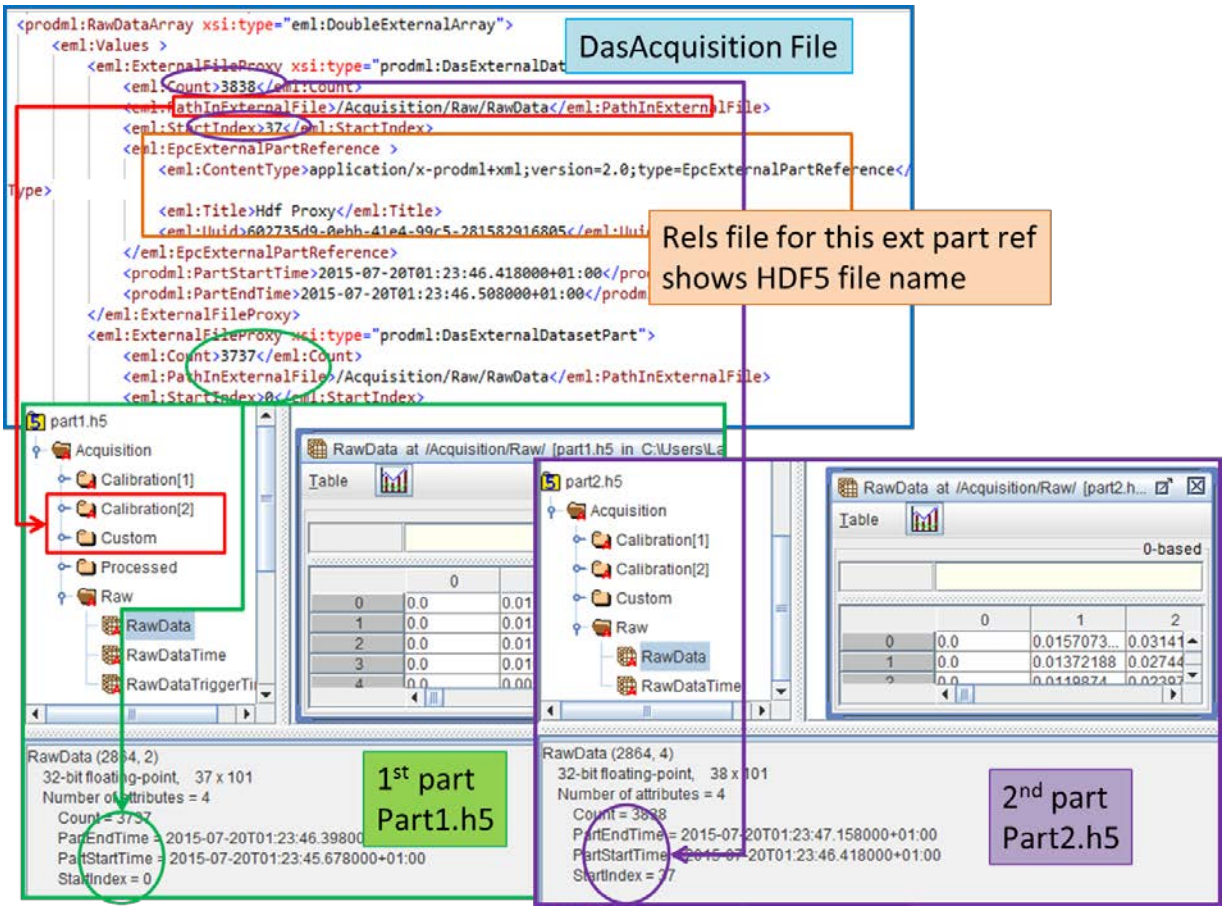


Figure 20–7. Shows how a single array (raw data in this case) is split across two physical files per the worked example and the figures above. The files are identified by green bubble/arrow/label (1st file, bottom left) and by a purple bubble/arrow/label (2nd file, bottom right).

Processed data, i.e., FBE band or spectra data, is stored and referenced the same way (**Figure 20**–8**)**.

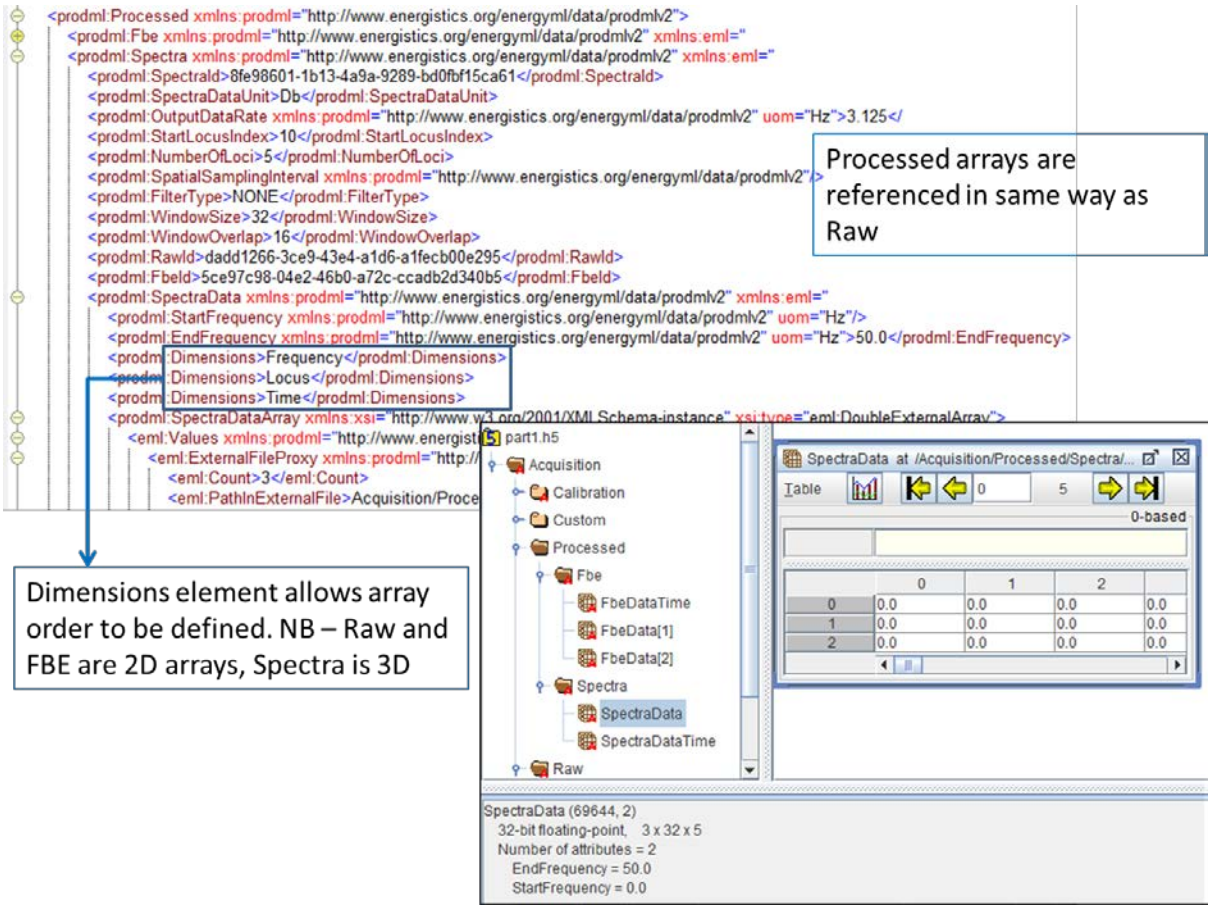


Figure 20–8. Spectra processed data referenced from the XML. In this case, each FBE band has a unique array name in the HDF5 file, and these can be seen in the PathInExternalFile elements in the snippets.

The above description has focused on how to navigate from the XML, via the data in the EPC, to the required arrays in the HDF5 files. The HDF5 files also contain identification data. The root level of the .h5 file contains the UUID of the DAS Acquisition (attribute named AcquisitionID). Every HDF5 file that is part of the sequence of HDF5 files has this same ID so that if an HDF5 file gets separated (e.g., a disk is misplaced); it can be associated with the right acquisition.

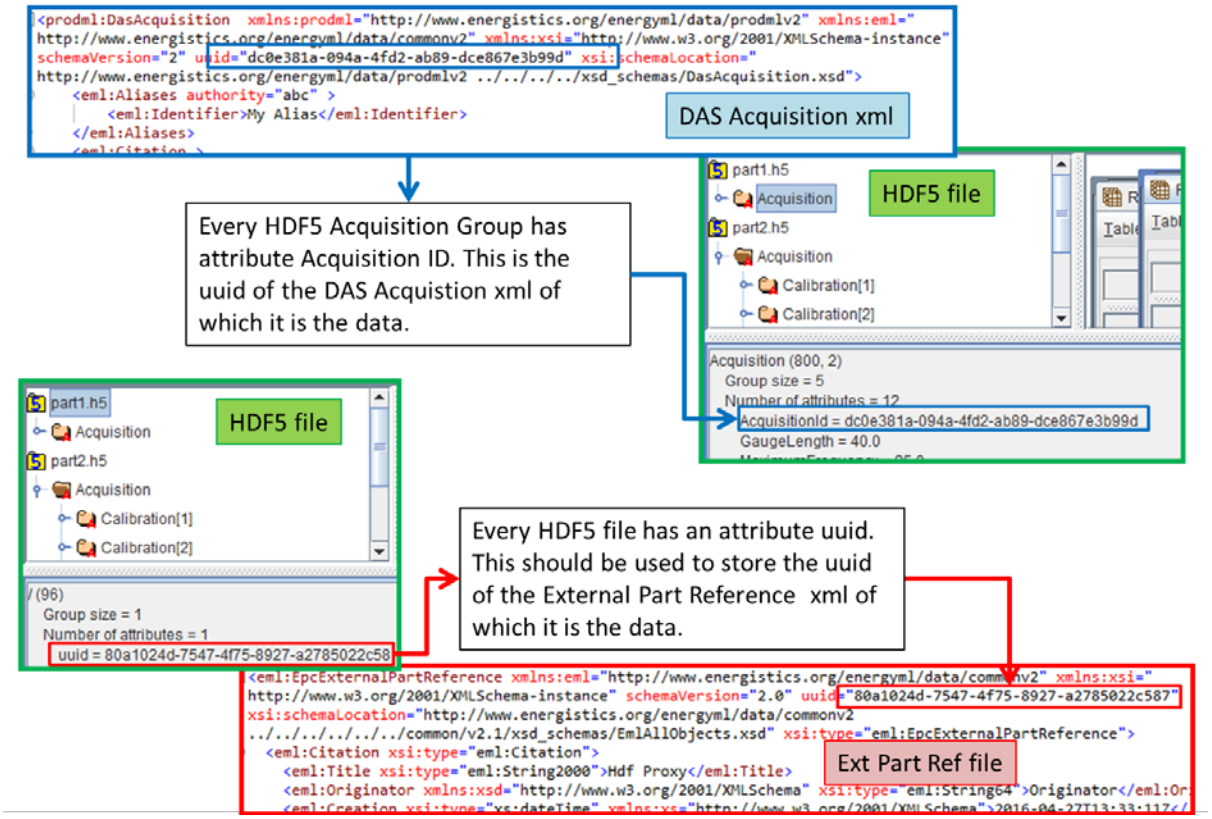


Figure 20–9. Keeping files associated in both directions: The external part reference mechanism is designed to navigate "forward" from DAS Acquisition xml to HDF5 arrays. Attributes and uuids are also used to be able to keep the files associated in both directions in case very large files on disks get separated from the set of disks/files.

20.4 Optical Path

For the optical path, the worked example is described in the DTS Section; se Section 15.1.

20.5 Instrument Box

Declaring a DAS instrument box is also very straightforward. Not many elements are required other than the unique identifier, a name, and the type. All other attributes are optional. Please see the schema for the details which may be added.

```
<prodml:DasInstrumentBox  xmlns:prodml="http://www.energistics.org/energyml/data/prodmlv2
http://www.energistics.org/energyml/data/commonv2" xmlns="http://www.energistics.org/ener
="http://www.w3.org/2001/XMLSchema-instance" uuid="df9447a1-8c6b-4634-88eb-ab07b41ac876"
xsi:schemaLocation="http://www.energistics.org/energyml/data/prodmlv2 ../../../../xsd_sch
    <eml:Citation>
        <eml:Title>Instrument Box</eml:Title>
        <eml:Originator>Fred Mertz, Field Tech</eml:Originator>
        <eml:Creation>2015-07-20T01:00:00.000000Z</eml:Creation>
        <eml:Format>Vendor:ApplicationName</eml:Format>
    </eml:Citation>
    <SerialNumber>12645A</SerialNumber>
    <Parameter index="1" name="Parameter1" uom="s" description="time"/>
    <Instrument>
        <Name>Instrument Box</Name>
    </Instrument>
    <FirmwareVersion>Firmware version 1</FirmwareVersion>
</prodml:DasInstrumentBox>
```

Figure 20–10. DAS Instrument Box example.

20.6 Acquisition Data

This section describes the equipment and optical path used for the worked example acquisition and the raw and processed measurement data.

The following sections illustrate the content for *part1.h5*, but the same general principles apply to all HDF5 files. The metadata is also present in the XML file. The relationships and navigation from the XML to the arrays in the .h5 files have been described above.

20.6.1 DAS Acquisition

The DAS Acquisition data object specifies the key instrument attributes for the full DAS acquisition including InstrumentBox and VendorFormat, MaximumFrequency, MinimumFrequency, PulseRate, PulseWidth, GaugeLength and the SpatialSampling interval along the fiber.

- The PulseRate is the number of pulses sent into the fiber per second.
- The GaugeLength is the distance between pair of pulses used in a dual-pulse or multi-pulse system.
- For each pulse, the backscatter is sampled along the fiber (by an AD converter that samples at a much higher rate than the PulseRate). The OpticalPath points to a metadata object that describes the (fiber) path along which the light pulses travel.

The StartLocusIndex is the first location (locus) along the fiber at which a recording is made. The distance between the recorded loci (locuses) is determined by the SpatialSamplingInterval. The number of samples made from and including the first locus is stored in the NumberOfLoci record. **Figure 20**–11 shows the concept.

Figure 20–11 also shows the metadata attributes for the example file. Some of the main attributes are explained here.

The DAS acquisition job name is indicated by the Title attribute. Further the Originator can be indicated.

The location of a locus on the fiber can be estimated by multiplying the locus number times the SpatialSampingInterval. The index of the first locus is 0 and hence locus 0 has a distance of 0. The first locus presents the measurement taken over first spatial sampling interval which starts at the DAS instrument connector/splice (fiber distance 0) and has a length of (fiber distance) SpatialSamplingInterval. Note that this is always an estimate because the fiber refractive index may vary slightly along fiber or vary more if the optical path consists of fibers with different refractive index values are spliced together. In addition, splices or certain optical components may cause delays, which must be corrected for by the end user in a distance through a calibration exercise.

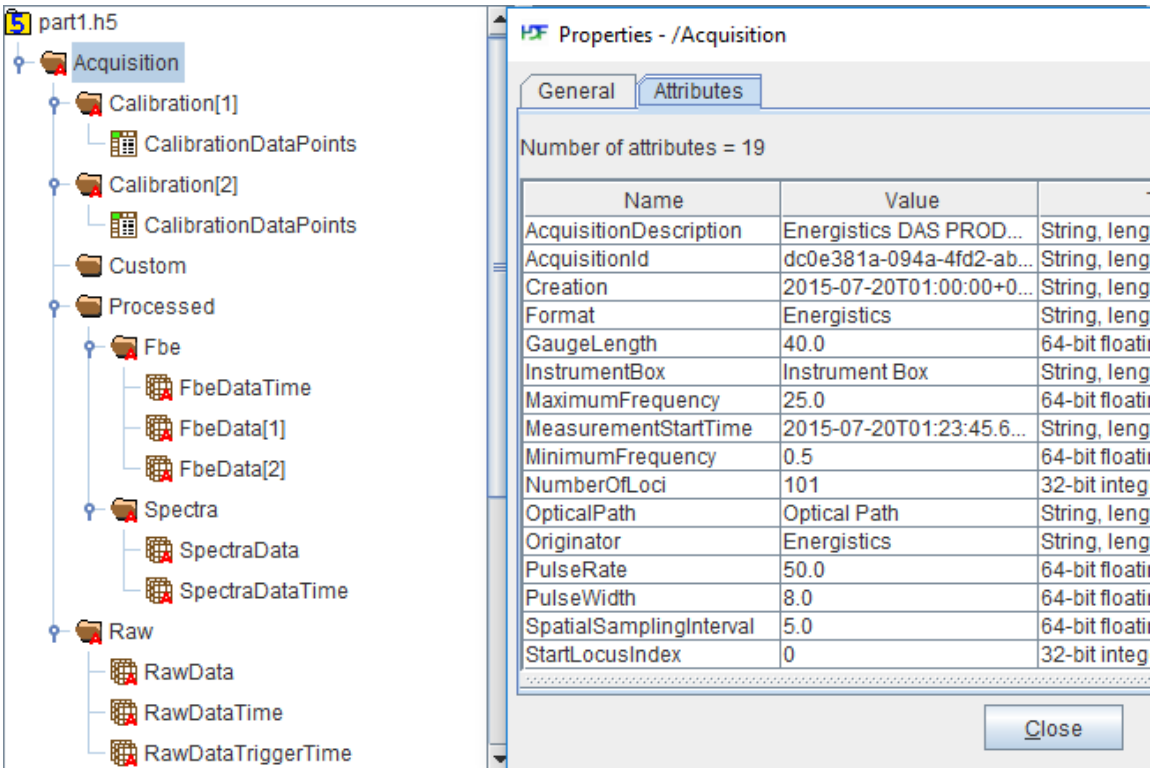


Figure 20–11. DAS acquisition attributes for *part1.h5*. In the left pane; note the file structure and naming conventions as explained in Section 19.5.4 above.

20.6.1.1 Acquisition Timing

The DASAcquisition data object specifies the following time elements for the acquisition:

- MeasurementStartTime (string): Specifies the UTC time of the first raw DAS sample in the acquisition.
- TimeZone (string): The time zone of the recording, this can be used to convert the UTC time into local time and vice versa.
- TriggeredMeasurement (Boolean): Flag set to TRUE if the recording is triggered.
- TriggeredTime (string): Specifies the trigger time at which the trigger was recorded. This time can be used to the first sample of the recording for the triggered event (for example, a microseismic event) in the DAS raw data recorded during the acquisition.

Figure 20–12 shows how the DAS acquisition time elements link the DAS raw data recording time elements described in the DAS raw times object. The triggered time indicates the start of the “triggered” record in the acquisition.

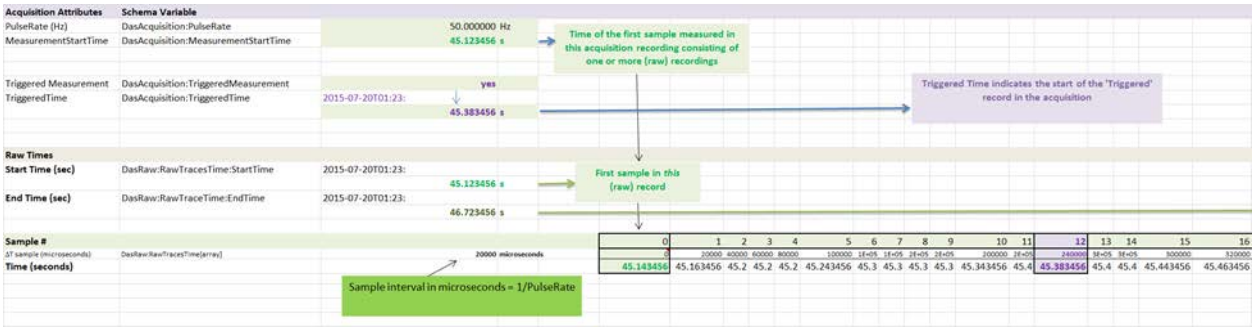


Figure 20–12. Link between DAS acquisition time, trigger time, and DAS raw time elements.

20.6.2 DAS Calibration

The (optional) Calibration group contains a mapping of loci index to optical path (fiber) distance and facility length. This group can repeat for each facility being measured. Consider Figure 15–1 in Chapter 15, which is the worked example optical path. In **Figure 20**–13, extra information is added to show which loci relate to which facility. For the purposes of the example, the surface cable itself is regarded as a "facility".

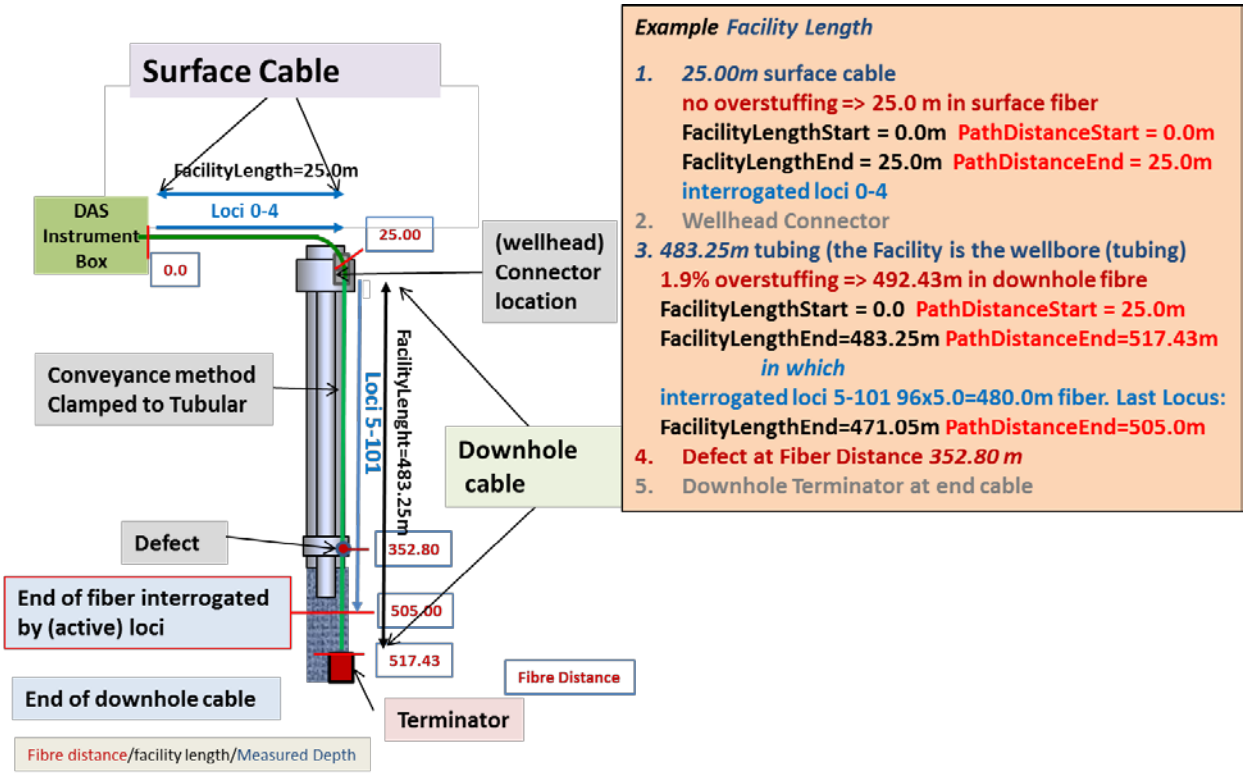


Figure 20–13. Worked example optical path, showing where the loci are, relative to the facilities being measured.

It is clear that there is a set of facility mappings for the surface cable facility and one for the wellbore. The worked example optical path contains these (see Section 15.1). In the DAS Acquisition file, we get to add the same information, plus calibration data, into the Calibration sections.

The mapping is stored as a Calibration for each facility. Each contains a set of DasCalibrationPoint structures. A CalibrationPoint structure contains four elements: LocusIndex, OpticalPathDistance , FacilityLength and CalibrationType.

Figure 20–14 shows the final data representation which needs to go into the calibration sections. To keep the files smaller, only a subsection of the 5.0 interval calibration points have been included.

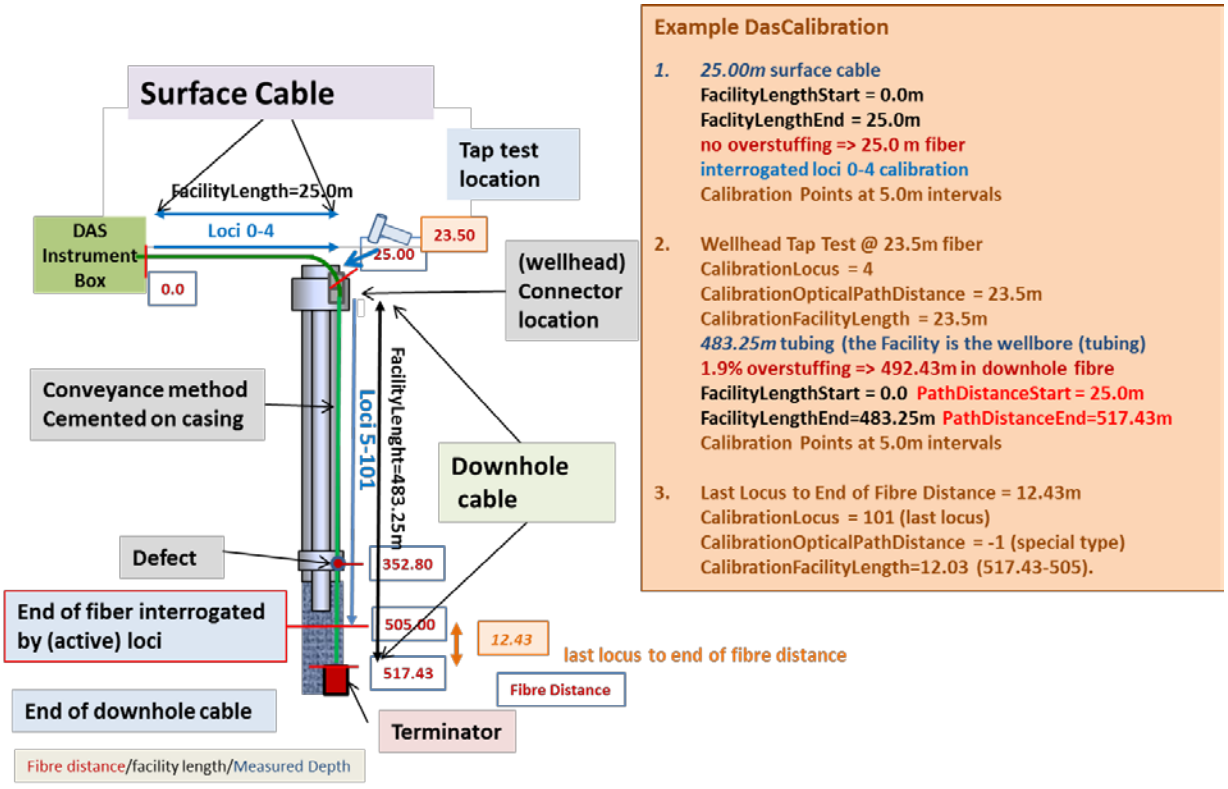


Figure 20–14. Worked example optical path showing the DAS calibration data.

The following tables shows how the DAS Calibration object can be used to provide a (very basic) depth calibration for the worked example. The first table shows the surface cable.

Calibration LocusIndex	Calibration OpticalPathDistance	Calibration FacilityLength	Calibration Type
0	5.000	5.000	locus calibration
1	10.000	10.000	locus calibration
4	23.50	23.50	tap test
4	25.000	25.000	locus calibration

The second shows the wellbore. The mapping now includes the effect of the offsetting optical path distance of the surface cable, and includes the over-stuffing effect (see figures and optical path XML file for more details).

Calibration LocusIndex	Calibration OpticalPathDistance	Calibration FacilityLength	Calibration Type
5	30.000	4.907	locus calibration
6	35.000	9.814	locus calibration
99	500.000	466.143	locus calibration
100	505.000	471.050	locus calibration

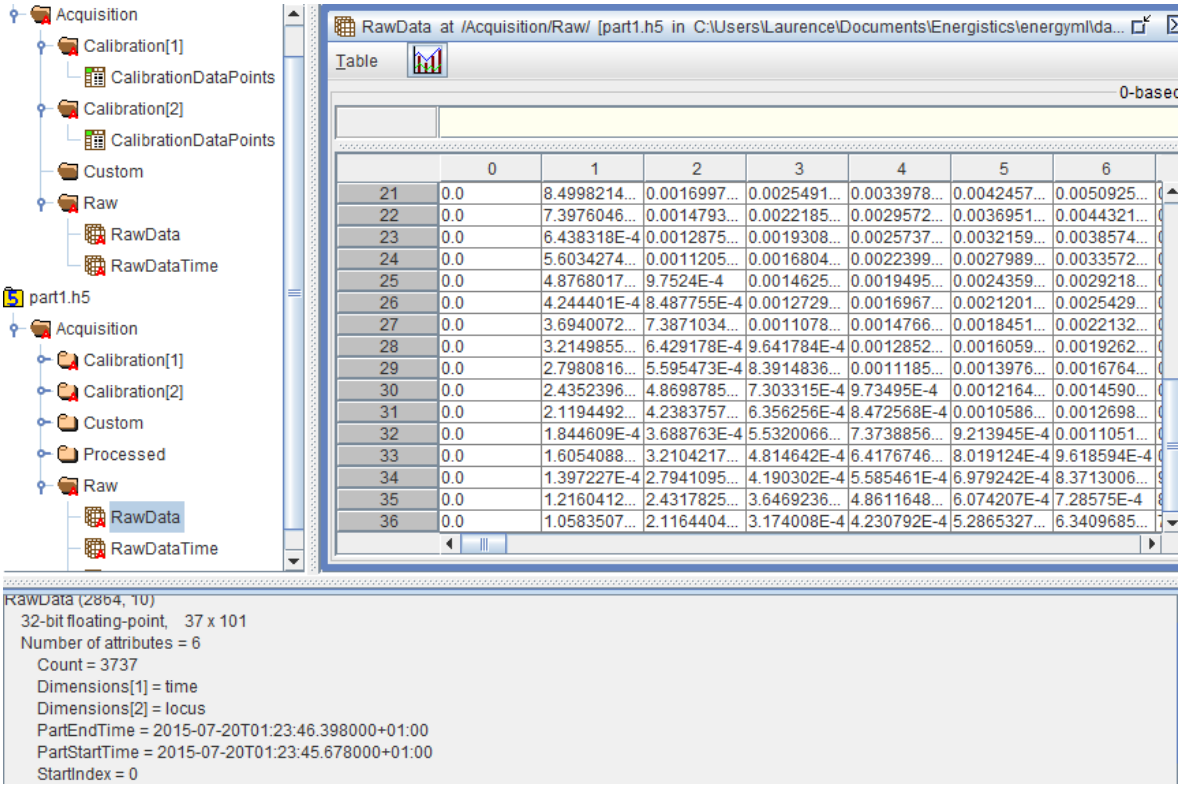
Calibration LocusIndex	Calibration OpticalPathDistance	Calibration FacilityLength	Calibration Type
100	12.43	NULL	last locus to end of fiber

These numbers and supporting data can be seen in both the Das acquisition xml and in the HDF5 file.

20.6.3 DAS Raw data

The Raw Group stores the raw sample measurements recorded by the DAS instrument. For each DAS pulse sent out, a measurement is obtained for all the loci interrogated by the instrument. This means that for each locus, a time-series is typically recorded at the instrument PulseRate frequency set in the Acquisition group, in case the raw output data is written at a different rate, then OutputDataRate should be present. If OutputDataRate has not been set, then it is assumed equal to the acquisition PulseRate.

The recorded data is stored in a 2D array named DasRawData. **Figure 20**–15 shows the stored values of some of the first 37 samples (numbered 0 to 36) of the times series for the first 7 loci (numbered 0 to 6) for the DAS raw data recorded in the example file. Note that the attributes include the *part start time* and *part end time*, for this array within the whole Raw data. The Dimension attributes specify the array ordering (time, locus), with locus the faster axis.



	0	1	2	3	4	5	6
21	0.0	8.4998214...	0.0016997...	0.0025491...	0.0033978...	0.0042457...	0.0050925...
22	0.0	7.3976046...	0.0014793...	0.0022185...	0.0029572...	0.0036951...	0.0044321...
23	0.0	6.438318E-4	0.0012875...	0.0019308...	0.0025737...	0.0032159...	0.0038574...
24	0.0	5.6034274...	0.0011205...	0.0016804...	0.0022399...	0.0027989...	0.0033572...
25	0.0	4.8768017...	9.7524E-4	0.0014625...	0.0019495...	0.0024359...	0.0029218...
26	0.0	4.244401E-4	8.487755E-4	0.0012729...	0.0016967...	0.0021201...	0.0025429...
27	0.0	3.6940072...	7.3871034...	0.0011078...	0.0014766...	0.0018451...	0.0022132...
28	0.0	3.2149855...	6.429178E-4	9.641784E-4	0.0012852...	0.0016059...	0.0019262...
29	0.0	2.7980816...	5.595473E-4	8.3914836...	0.0011185...	0.0013976...	0.0016764...
30	0.0	2.4352396...	4.8698785...	7.303315E-4	9.73495E-4	0.0012164...	0.0014590...
31	0.0	2.1194492...	4.2383757...	6.356256E-4	8.472568E-4	0.0010586...	0.0012698...
32	0.0	1.844609E-4	3.688763E-4	5.5320066...	7.3738856...	9.213945E-4	0.0011051...
33	0.0	1.6054088...	3.2104217...	4.814642E-4	6.4176746...	8.019124E-4	9.618594E-4
34	0.0	1.397227E-4	2.7941095...	4.190302E-4	5.585461E-4	6.979242E-4	8.3713006...
35	0.0	1.2160412...	2.4317825...	3.6469236...	4.8611648...	6.074207E-4	7.28575E-4
36	0.0	1.0583507...	2.1164404...	3.174008E-4	4.230792E-4	5.2865327...	6.3409685...

Figure 20–15. DAS raw data array values: 101 loci (horizontal 0-100), 37 samples (vertical 0-36) in h5 file *part1.h5*. (part2.h5 has the same loci and the samples continue in sequence, 38 samples (37-75).

20.6.3.1 Das Raw Times

A DAS Raw object has the following attributes specifying the recorded sample times in the example file (**Figure 20**–16):

- StartTime (string): Specifies the UTC time of the samples recorded in this h5 file RawDataPart 2D array (2D because for each loci vs. a series of samples is recorded).
- RawTracesTimes (array object): A 1D array that contains the times in Unix time (!) in microseconds for each sample in the current recording. The array object has two additional attributes describing the StartTime and EndTime of current recording in human readable form for convenience of the end user inspecting the h5 file: DasRawTraceTimes:StartTime and DASRawTraces:EndTime, which contain the start and end time of the full acquisition. For each partial h5 file, the start and end times are stored in PartStartTime and PartEndTime.

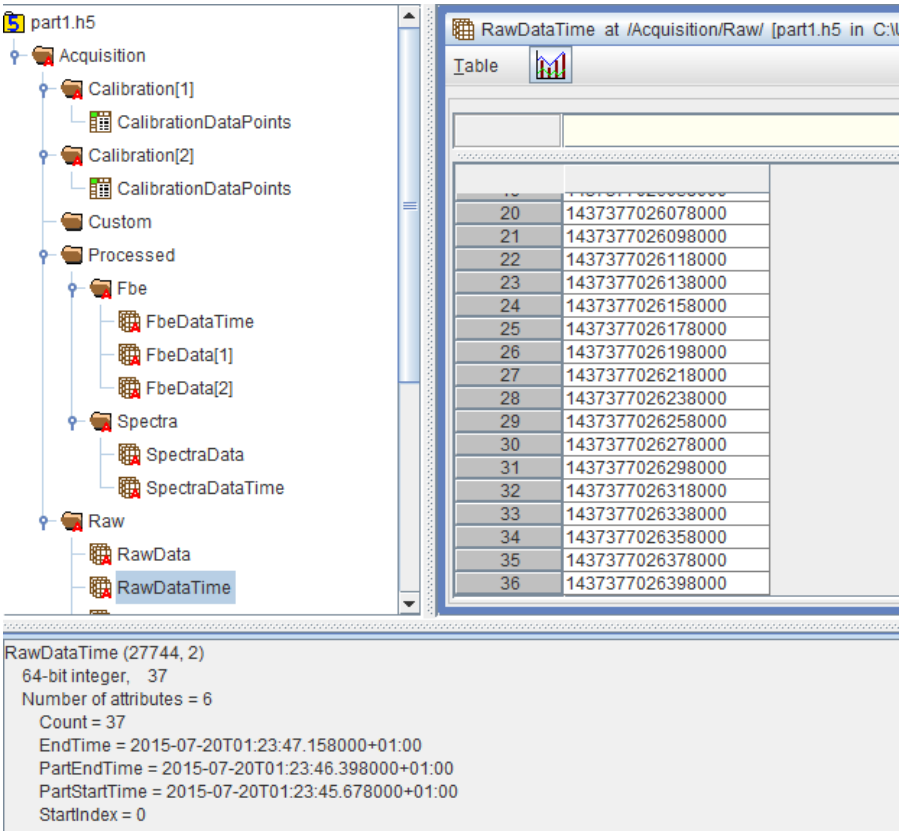


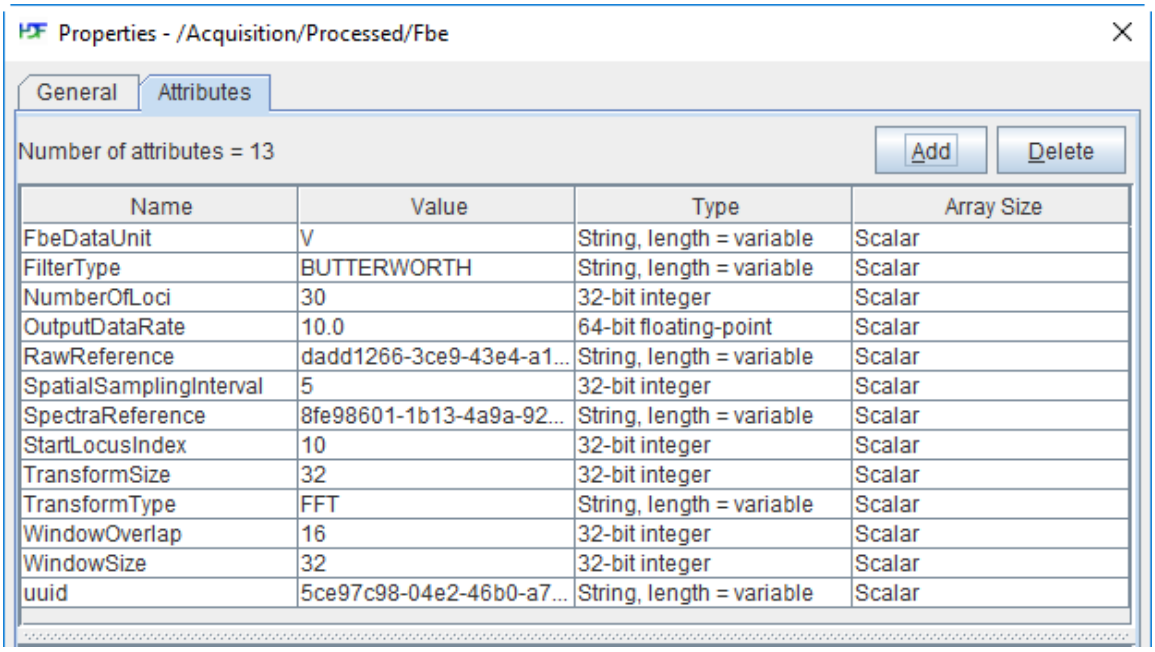
Figure 20–16. DAS raw trace times array and attributes in h5 file *part1.h5*.

20.6.4 DAS FBE data

The Processed DAS FBE data object stores so-called frequency band extracted (FBE) data values calculated from raw DAS data recordings. The FBE data values are typically calculating the energy in a specific frequency band in a filter-window over the raw DAS data recording for each locus. This calculation produces a 'trace' (1D array) that contains an energy value for each locus, each time the filter-window was applied.

Figure 20–17 shows the attributes for the DAS FBE data object in the example file. Key attributes stored are: the FilterType applied while creating the frequency bands, and the StartIndex and NumberOfLoci recorded. Note that FBE trace loci can be a subset of the DAS raw data loci recorded for the DAS raw data from which the FBE data was generated. The OutputDataRate determines the Number of FBE traces output per second in the Das FBE object and is equal to or smaller than the DAS instruments PulseRate used to collect the DAS raw data. For each frequency band, the calculated energy value are stored in an FbeData Array presented by a 2D array as shown in **Figure 20**–18. Each FbeData Array has a StartFrequency and EndFrequency attribute indicating in the FBE frequency band for which the data was calculated and stored. Figure 20–18 shows part of the first frequency band stored in FbeData[1] Array in the H5 *part1.h5* sample file. The frequency band covers 0.5-1.0Hz and contains energy data for

the first locus starting at index 10 (StartLocus) and covering 30 (NumberOfLoci) loci ranging with index numbers 10 to 39. There are 15 FBE traces (indexed 0 to 14).



Name	Value	Type	Array Size
FbeDataUnit	V	String, length = variable	Scalar
FilterType	BUTTERWORTH	String, length = variable	Scalar
NumberOfLoci	30	32-bit integer	Scalar
OutputDataRate	10.0	64-bit floating-point	Scalar
RawReference	dadd1266-3ce9-43e4-a1...	String, length = variable	Scalar
SpatialSamplingInterval	5	32-bit integer	Scalar
SpectraReference	8fe98601-1b13-4a9a-92...	String, length = variable	Scalar
StartLocusIndex	10	32-bit integer	Scalar
TransformSize	32	32-bit integer	Scalar
TransformType	FFT	String, length = variable	Scalar
WindowOverlap	16	32-bit integer	Scalar
WindowSize	32	32-bit integer	Scalar
uuid	5ce97c98-04e2-46b0-a7...	String, length = variable	Scalar

Figure 20–17. DAS FBE attributes h5 file *part1.h5*.

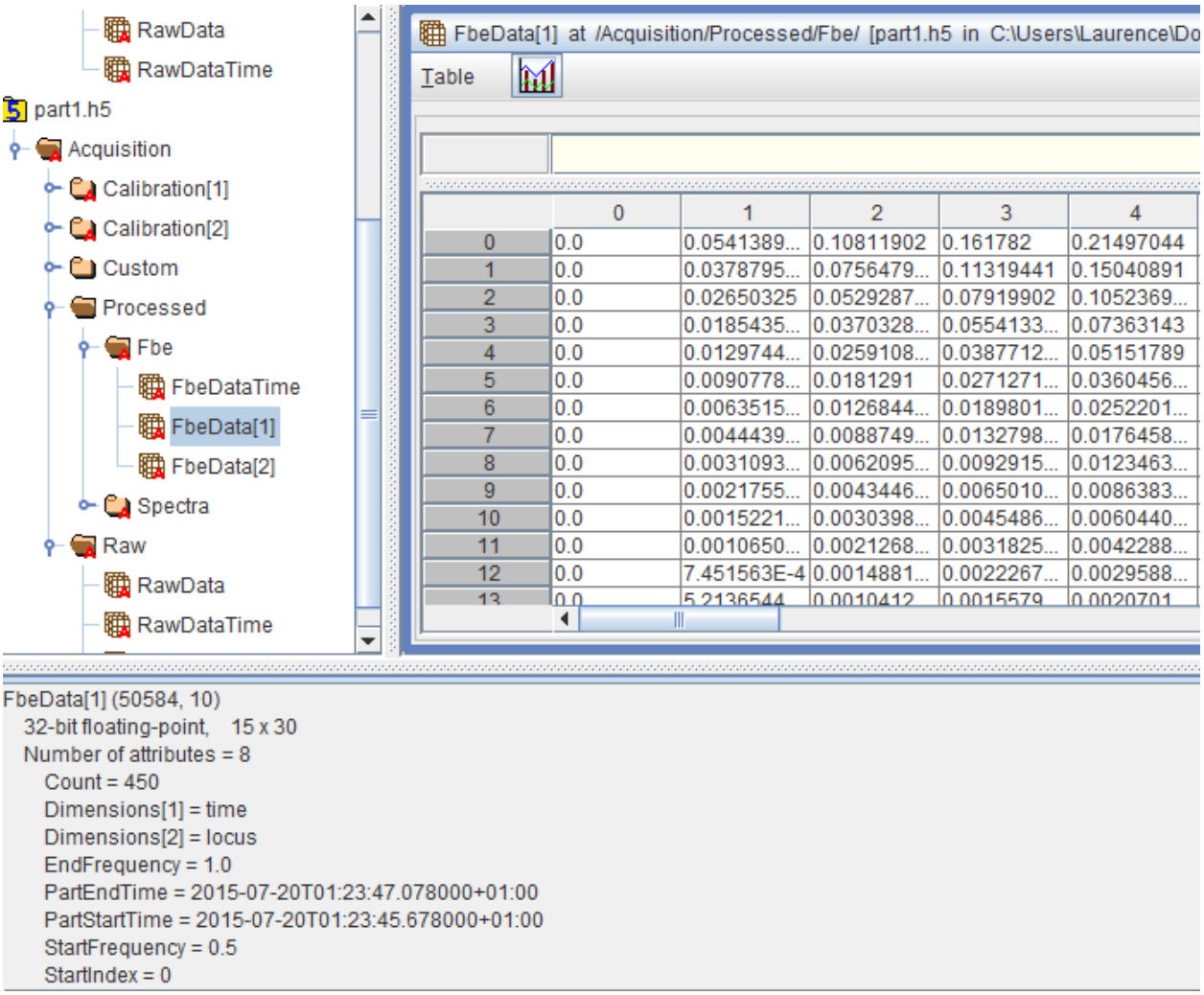


Figure 20–18. DAS FBE attributes and array values and for frequency band FbeData[1] in h5 file *part1.h5*.

20.6.4.1 DAS FBE 'trace' Times

An individual FBE trace's time is specified as the time of the first sample in the filter-window. The number of FBE traces output per second is determined by the OutputDataRate attribute (equal or smaller than the DAS instrument's 'raw' PulseRate).

A DasFbe object stores the time values for each FBE 'trace' in an FbeTracesTime array object. This is a 1D array that contains the times in microseconds for each FBE 'trace' in the current recording*. Note that the times in the 1D example array are 1/OutputDataRate microseconds apart. The FbeTracesTime array object has two additional attributes in human readable form for convenience of the end user inspecting the h5 file describing the StartTime and EndTime of the series of FBE traces recorded. StartTime corresponds to the first stored FBE trace time and EndTime corresponds to the last stored FBE trace time in this DAS FBE object. **Figure 20**–19 shows how the DAS FBE 'trace' times link together. **Figure 20**–20 shows the FBETraceTimes array attributes and the 14 FbeTraceTimes (one for each FBE trace) in microseconds in Unix time (!) in h5 file *part1.h5*.

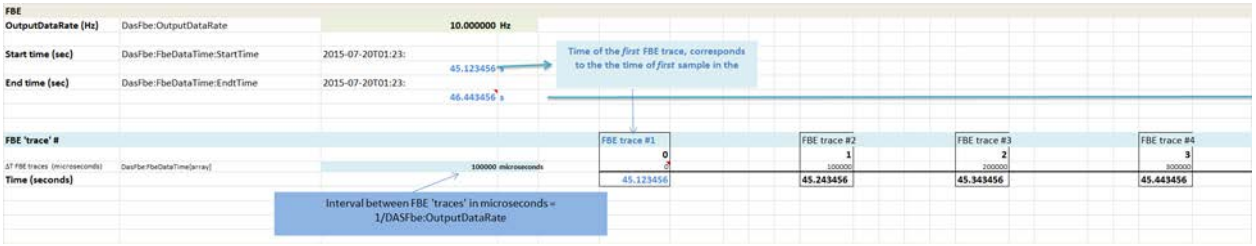


Figure 20–19. DAS FBE 'trace' times.

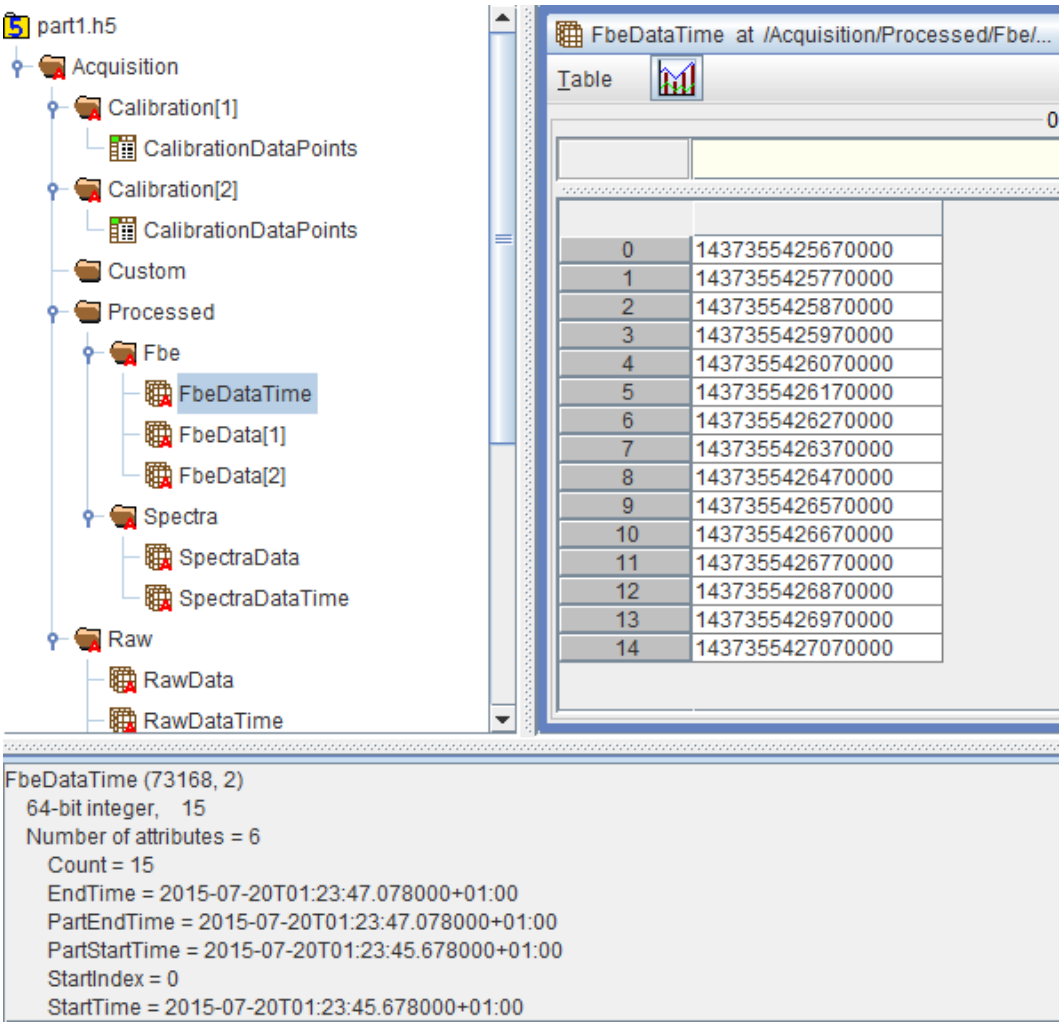


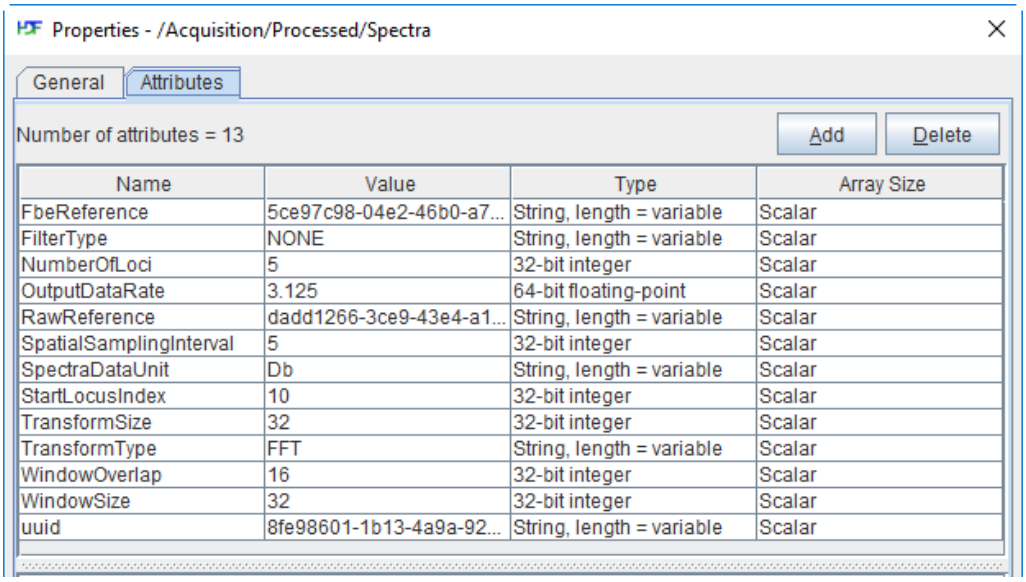
Figure 20–20. DAS raw FBE times array and attributes in h5 sample file *part1.h5*.

20.6.5 DAS Spectra data

The DAS Spectrum object stores one spectrum or multiple spectra calculated from the raw DAS data recordings. The spectra are typically calculated as N-point fast Fourier transforms (FFTs) along a window of length FilterWindowSize Raw DAS sample. The FilterWindowSize equals the FFT size N and is the number of output points from the discrete Fourier transform (DFT) calculation.

For each locus in the array of loci specified by StartLocus and NumberOfLoci, the FFT calculation produces FilterWindowSize data points. FFT calculations can be repeated by shifting the filter-window

over the raw DAS samples. The windows can be overlapping and the number of samples that overlap is specified by the FilterWindowOverlap attribute. This means that the spectrum values are stored in a 3D array [time:locus:FFT-value]. **Figure 20**–21 shows an example of 4 overlapping windows in the spectra object in the example file.



Name	Value	Type	Array Size
FbeReference	5ce97c98-04e2-46b0-a7...	String, length = variable	Scalar
FilterType	NONE	String, length = variable	Scalar
NumberOfLoci	5	32-bit integer	Scalar
OutputDataRate	3.125	64-bit floating-point	Scalar
RawReference	dadd1266-3ce9-43e4-a1...	String, length = variable	Scalar
SpatialSamplingInterval	5	32-bit integer	Scalar
SpectraDataUnit	Db	String, length = variable	Scalar
StartLocusIndex	10	32-bit integer	Scalar
TransformSize	32	32-bit integer	Scalar
TransformType	FFT	String, length = variable	Scalar
WindowOverlap	16	32-bit integer	Scalar
WindowSize	32	32-bit integer	Scalar
uuid	8fe98601-1b13-4a9a-92...	String, length = variable	Scalar

Figure 20–21. Spectra[1] attributes in H5 sample file *part1.h5*.

Figure 20–22 shows one of the FFT sub-arrays for one of the 5 loci (index 0 to 4 on the "page" axis), the 3 overlapping time windows (index 0 to 2 on the vertical axis) calculated for all spectra values (0 to 32) on the horizontal axis). The FFTs for the 3 windows were calculated from 75 samples stored in the two split h5 sample files part1.h5 (DAS raw samples 0-37) and part2.h5 (DAS raw samples 38-75).

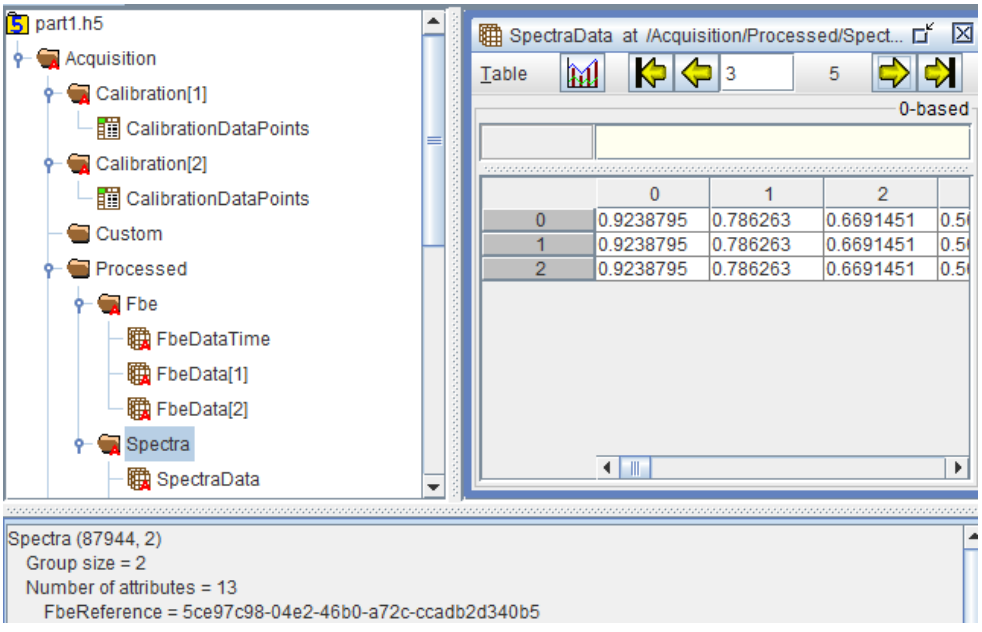


Figure 20–22. DAS FFT spectra for the spectra data array in h5 sample file *part1.h5*.

20.6.5.1 DAS Spectra 'trace' times

A DAS Spectrum object stores the start times of each FFT calculation window applied in the SpectraDataTimes array object. This is a 1D array that contains the times in microseconds in Unix time for each FFT window stored in the DAS spectrum object*. Note that the times in the 1D example array are 1/ OutputDataRate microseconds apart. The SpectraTimes array object has two additional attributes describing the times of the FBE traces recorded in human readable time format for convenience of the user: StartTime corresponds to the first stored FBE trace time and EndTime corresponding to the last stored FBE trace time in this DAS FBE object.

Figure 20–22 shows an example where spectra data is generated at an OutputDataRate of 3.125 Hz (note that this is 1/16th of the DAS instrument 'raw' PulseRate of 50 Hz, and corresponds to the number of raw samples the calculation filter-window is shifted when calculating the FFT for each locus. The sample times in microseconds are stored in the SpectraDataTime array; They are relative to the StartTime of the raw recording. **Figure 20**–23 shows the SpectraTimes array corresponding with the spectrum windows in Figure 20–22 in the example file.

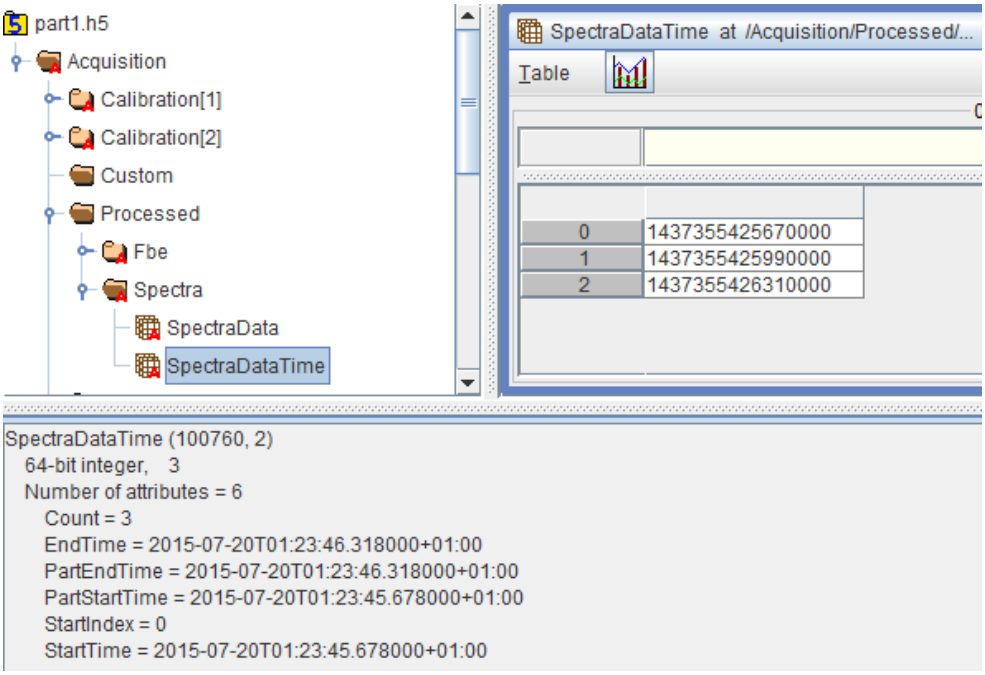


Figure 20–23. Spectra data time array and attributes in H5 sample file *part1.h5*.

21 Code Examples: Use Cases

For help in implementing the DAS PRODML standard, the download includes the example described in detail in Chapter 20, plus five additional examples of typical use cases for acquisition and processed data exchange between service providers and operators. NOTE: In the use case titles below, “transport medium” refers to a type of data transport media.

The five use cases presented are:

- **Use Case 1: Field Data Acquisition: raw, single transport medium.** The EPC package contains acquisition, optical path and instrument box XML files and one external part reference file pointing to a single HDF5 file containing all the raw DAS data for the acquisition.
- **Use Case 2: Field Data Acquisition and Processed Data; raw and frequency band; single transport media.** The EPC package contains acquisition, optical path, and instrument box XML files and one external part reference file pointing to a single HDF5 file containing all the raw and processed frequency band arrays.
- **Use Case 3: Field Data Acquisition and Processed Data: raw, frequency band and spectra data; single transport media.** The EPC package contains acquisition, optical path, and instrument box XML files and one external part reference file pointing to a single HDF5 file containing all the raw and processed frequency band and spectra arrays.
- **Use Case 4: Field Data Acquisition and Processed Data; raw, frequency band and spectra; multiple transport media.** The EPC package contains acquisition, optical path, and instrument box XML files and two external part reference files pointing to a first HDF5 file containing all the raw data acquired and a second HDF5 file containing all the processed frequency band and spectra arrays.
- **Use Case 5: Field Data Acquisition and Processed Data; raw, frequency band and spectra; multiple transport media.** The EPC package contains acquisition, optical path, and instrument box XML files and 3 external part reference files: 2 point to HDF5 files containing only raw data and the third to an HDF5 file that contains both raw data and the processed frequency band and spectra arrays.

22 Appendix for DAS

22.1 DAS Terminology

This appendix contains a detailed list and definitions of DAS terminology.

Name	Definition	Unit	Example
Decimated Output Data Rate	An integer fraction of the '**Output Data Rate'**.	Hz (1/s)	1Khz
Decimation Factor	The integer number by which the 'Output Data Rate' is decimated.	Integer	5
Fiber Distance	The 'Optical Path' distance from the connector of the measurement instrument to the desired acoustic sample point along the fiber that is the furthest from the measurement instrument for that particular test.	m*	1234m
Start Locus Index	The first 'Locus' acquired by the interrogator unit. Where 'Locus Index 0' is the acoustic sample point at the connector of the measurement instrument.	Integer	20
Frequency Response (of the Interrogator Unit)	The Nyquist frequency (or some fraction thereof) of the 'Interrogation Rate' or 'Pulse Rate'.	Hz (1/s)	20kHz
Gauge Length	This is a distance (length along the fiber) which the DAS Interrogator Unit manufacturer designs/implements by hardware/software to affect the Interrogator Unit spatial resolution.	m* or s	40m or 200ns
Locus* (plural: Loci) or 'Spatial Sample'	A particular location indicating a spatial sample point along the sensing fiber at which a 'Time Series' of acoustic measurements is made. In figure 1A a locus is identified as 'o' or 'o'.		
Locus Distance	The '**Fiber Distance'** to centre of the '**Locus'**.	m*	1250m
Locus Index	The index number of a 'Locus' represents a spatial sample, where such numbering of the spatial sample starts from the output of the Interrogator Unit (IU) starting with '0'. Successive loci are numbered as positive integers which increase along the length of the sensor fiber. The separation of consecutive loci along the sensor fiber is determined by the spatial sample interval. The dots in Figure 18-3 indicate the center of the particular spatial sample.	Integer	1

Name	Definition	Unit	Example
Measurement Start Time	The time at the beginning of a data 'Sample' in a 'Time Series'. This is typically a GPS locked time measurement.	Clock Time	2014-08-19 07:57:11.453 721 +- X
Number of Loci	The total number of 'Loci' (acoustic sample points) acquired by the measurement instrument in a single 'scan' of the fiber. The number of 'scans' per second is determined by the 'Interrogation Rate' or 'Pulse Rate'.	Integer	100
Number of Samples	The number of 'Samples' in a 'Time Series'.	Integer	12345
Optical Path	A series of fibers, connectors, etc. together forming the path for the light pulse emitted from the measurement instrument.		
Output Data Rate	The rate at which the measurement system provides output data for all 'Loci' (Spatial Samples). This is typically equal to the Interrogation Rate/Pulse Rate or an integer fraction thereof.	Hz (1/s)	5.0 kHz
Pulse	A single burst of laser light into the fiber		
Interrogation Rate/ Pulse Rate	The rate at which the Interrogator Unit interrogates the fiber sensor. For most interrogators this is informally known as the pulse rate.	Hz (1/s)	2.5 kHz
Pulse Width	The width of the 'Pulse' sent down the fiber.	s	10 ns
Trace / Scan	Array of sensor values for all loci interrogated along the fiber for a single 'pulse' (columns in Figure 18-3).		
Sample Number	The index of a particular 'Sample' within its parent 'Time Series'.	Integer	3
Sample Time	Time at which a particular 'Sample' was acquired	s	
Spatial Resolution	The ability of the measurement system to discriminate signals that are spatially separated. It should not be confused with Spatial Sampling Interval.	m*	5.0 m
Spatial Sampling Interval*	The separation between two consecutive 'Spatial Sample' points on the fiber at which the signal is measured. It should not be confused with 'Spatial Resolution'.	m*	1.0 m
Time Offset	The time off set relative to 'Measurement Start Time'.	µs	1234567
Time Series	A data set for a particular 'Locus' which is sampled at the Interrogation Rate/Pulse Rate (rows in figure 2.3).		
Time Series	The rate at which acoustic samples are acquired for a particular 'Locus'. This is equal	Hz (1/s)	5 kHz

Name	Definition	Unit	Example
Sampling Rate	to the 'Output Data Rate' or an integer fraction thereof ('Decimated Output Data Rate').		
Total Fiber Length	The maximum 'Fiber Distance' from the connector of the measurement instrument to the final end of the 'Optical Path'. This end is either a purposefully cut or terminated end of the fiber.	m*	5120 m
Trigger Accuracy	*(to be defined)*		
Triggered Measurement	Measurement for an acquisition which requires synchronization between a transmitting source (Tx) and a recording (Rx) measurement system. This needs to be recorded for every measurement regardless of what application it will serve.	Boolean	True, False
Triggered Time	The time of the trigger in a 'Triggered Measurement'	Clock Time	2014-08-19 07:57:11.453 721 +- X
DAS Acquisition	Collection of DAS data acquired during the DAS survey.		
Acquisition ID	A universally unique identifier (UUID) for the acquisition.		
Facility	Generic term for equipment being measured, e.g., a wellbore, a pipeline, etc.		
Facility Length	A distance along a physical Facility from the connector of the Interrogator Unit to the physical end of the Facility.	m	10 m
Optical Path	A series of fibers, connectors, etc. together forming the path for the light pulse emitted from the Interrogator Unit.		
Optical Path Distance	Fiber distance along the Optical Path measured from the connector of the Interrogator Unit.	m*	
Instrument Box / Interrogator Unit	The measurement instrument. Often referred to as Interrogator Unit or IU.		
Instrument Location	Latitude – Longitude – Mean Sea Level. If GPS antenna is not connected leave as N/A- N/A - N/A Or WGS84	(Latitude, Longitude, MSL pairs)	(35.466088,-84.415512,-85.9)
Time Zone	UTC time zone	String	
DAS Calibration	A mapping of loci to fiber distance to a physical location and/or depth along an Optical Path.		
Offset End Fiber	The Fiber Distance to the end of the fiber?	m*	5678 m

Name	Definition	Unit	Example
Distance			
Offset End Fiber Locus Index	The Index of the last Locus interrogated by the measurement instrument. The position of this Locus is determined by the Offset End Fiber Distance.		
Trigger Accuracy			
Vendor Code	A string describing the data acquisition provided by the vendor to the client. This is backed by a document that the vendor issues to its clients describing the data collected. For example data mode 1 for Vendor A might mean raw ADC single pulse and Mode 2 might be raw ADC single pulse, with a high pass filter. This must be vendor specific as it's commercially sensitive.	String	"OptaSense 010 DAS raw decimated"
Tap Test	A test to identify a certain physical location on the fiber in the field. This is often done by 'tapping' the wellhead, hence the name Tap Test.		
Tap Test Locus Index	The locus index of the Tap Test.		
Tap Test Fiber Distance	The Fiber Distance of the Tap Test.		
DAS Raw Data	DAS data exchange format for describes 'raw' (unprocessed) DAS data provided by the vendor to a customer.	System dependent	
DAS FBE Data	DAS data exchange format for Frequency Band Extracted (FBE) DAS data provided by the vendor to a customer. This DAS data type describes a dataset that contains one or more frequency bands filtered time series of a 'raw' DAS dataset. This could be for example the RMS of a band passed filtered time series or the level of a frequency spectrum.	System dependent	
DAS Spectrum Data	DAS data exchange format for Frequency Spectrum information extracted from a (sub) set of 'raw' DAS data.	System dependent	
DAS Job	A set one or more DAS acquisitions acquired in a defined timeframe using a common Optical Path and DAS Instrument Box.		
Start Frequency	Start an individual frequency band in a DAS FBE data set. This corresponds to the frequency of the 3dB point of the filter.	Hz	1.00 Hz
End Frequency	End of an individual frequency band in a DAS FBE data set. This corresponds to the frequency of the 3dB point of the filter.	Hz	9.99 Hz

Name	Definition	Unit	Example
Minimum Frequency	The minimum signal frequency a measurement instrument can provide as specified by the vendor.	Hz	0.1 Hz
Maximum Frequency	The maximum signal frequency a measurement instrument can provide as specified by the vendor. This is the Nyquist frequency (or some fraction thereof) of the Time Series.	Hz	10.0 kHz
Filter	Describes the mathematical operation applied to remove unwanted components of the signal pre-processing. Filters can be described in the time domain as a time series or in the frequency domain as a response per frequency 'sub'-band.		
Filter Vendor Technique	A string describing the filter applied by the vendor to data provided to the client. This is backed by a document that the vendor issues to its clients describing the data collected. Filters applied are part of a vendor's proprietary processing techniques.	String	"Vendor X Bandpass"
Filter Type	A string describing the type of filter applied by the vendor. Important frequency type filter classes are frequency response filters (low-pass, high-pass, band-pass, notch filters) and butterworth, chebyshev and bessel filters. The filter type and characteristics applied to the acquired or processed data is important information for end-user applications.	String	"Vendor X Bandpass"
Filter Window Size	The number of samples in the filter window applied.	Integer	128
Filter Window Overlap	The number of samples overlap between consecutive filter windows applied	Integer	64
Depth	Depths are commonly specified either as measured depth (MD) along the borehole or true vertical depth (TVD), which is the measured depth relative to a depth reference. Common datums used are Mean Sea Level (MSL) and Kelly Busing (KB).	m	1234.0 m (KB) or 1228.1 m (TVD)

22.2 Use Cases

These are high-level business process use cases. The DAS data object supports transfer of the data that is produced from these use cases, but not the entire workflow of the use case.

The DAS business process use case involves the following steps:

- Define DAS acquisition requirements with customer
- Represent fiber optic installation

- Configure DAS equipment for acquisition
- Perform DAS data acquisition
- Post-process DAS data

22.2.1 Actors

All DAS use cases have a set of potential possible actors, which are defined below and used in the use cases described in this section.

Actor	Description
DAS Field Engineer	Engineer configuring and operating the DAS equipment in the field.
DAS Project Engineer	Expert in DAS and its applications. Often the person who recommends acquisition settings, QCs the acquisition, and does the final data processing and transcribing of data to the required delivery medium.
DAS Instrumentation	The DAS acquisition system (interrogator unit).
Fiber Vendor	The vendor that supplied (and installed) the fiber.
DAS Vendor	The vendor that supplied the DAS instrumentation.
Customer Field Engineer	The engineer in the field representing the customer.
Customer Project Engineer	Usually the engineer responsible for the requirements and successful completion of the operation. Usually the recipient of the acquired measurements.
Third-Party System	IT system receiving DAS data.

22.2.2 Use Case: Define Acquisition Requirements with Customer

Requirements for a job that a Customer specifies to an acquisition company.

Use Case Name	DAS Project Engineer and Customer agree on the acquisition requirements of the DAS service
Version	1.1
Goal	Agreed definition of acquisition requirements and final deliverables including reports, data, and media formats.
Summary Description	Define the requirements and scope of the service that the DAS vendor shall provide. Agree on: ping rates, spatial resolution, spatial sampling, type of DAS data to provide (raw or reduced, e.g. frequency bands), when to acquire data, format and media of deliverables.
Actors	DAS Project Engineer, Customer Project Engineer
Triggers	Request for service
Pre-conditions	Suitable fiber optic downhole or surface installation compatible with the DAS interrogator unit to be provided.
Primary or Typical Scenario	Customer Project Engineer and DAS Project Engineer agree the requirements of the acquisition.
Alternative Scenarios	
Post-conditions	Acquisition configuration agreed. The requirements are reflected in the metadata that is transferred with the DAS data. Currently, no requirement exists in the standard to transfer only this data.

Business Rules	
Notes	
Definitions	See Section 22.1.

22.2.3 Use Case: Represent Fiber Optic Installation

Requirements for a job that a Customer specifies to an acquisition company.

Use Case Name	Customer provides fiber optic installation layout to DAS project engineer and agrees the optical path configuration for the DAS acquisition.
Version	1.1
Goal	Agree optical path configuration for the DAS acquisition.
Summary Description	In most services the DAS Vendor provides the DAS acquisition instrument and the Customer is responsible for providing the fiber optic cable installation in the field from which the data will be acquired. To properly operate the DAS instrument, the DAS Vendor needs to be informed about the details of the field fiber optic installation and cable properties (path, fiber length, optical connectors, fiber type single or multi-mode fiber etc.).
Actors	DAS Project Engineer, Customer Project Engineer
Triggers	
Pre-conditions	Suitable fiber optic downhole or surface installation compatible with the DAS interrogator unit to be provided.
Primary or Typical Scenario	Customer Project Engineer provides DAS Project Engineer with the configuration of the optical installation to which the DAS acquisition instrument will be connected.
Alternative Scenarios	For some DAS acquisitions the DAS vendor will provides the fibers in the field, for example by providing a flexible rod with integrated fibers that can be spooled into a well.
Post-conditions	Fiber optic path understood.
Business Rules	
Notes	
Definitions	See Section 22.1 (page 192).

22.2.4 Use Case: Configure DAS Equipment for Acquisition

Use Case Name	Configure DAS Equipment for Acquisition
Version(s)	1.1
Goal	DAS equipment configured, QCed and ready for acquisition to begin.

Summary Description	DAS Field Engineer tests fiber integrity and notes the positions of any losses and reflections. DAS Field engineer configures DAS Equipment with the required acquisition settings. DAS Field engineer performs initial depth calibration (facility mapping) DAS Project Engineer will perform QC DAS Project Engineer delivers QC report to Customer Field and Project Engineers. Ensure Instrument Box is configured to timestamp the data as per the customer's requirements, e.g. GPS time synced, or synced to other equipment in the operation.
Actors	DAS Field engineer, DAS expert, Customer Project Engineer, Customer Field Engineer, DAS Instrumentation
Triggers	Request from client to deploy to site and prepare for acquisition.
Pre-conditions	Fiber deployed / installed Fiber specification known (refractive index, type, optical path geometry, length, end depth) OTDR completed, results delivered with OTDR acquisition settings. Acquisition requirements agreed with operator Service company deployed to site. Calibrations performed.
Primary or Typical Scenario	Configure acquisition for a vertical seismic profiling service
Alternative Scenarios	Configure acquisition for alternative DAS borehole or surface applications. Typical examples of borehole applications are Hydraulic fracture monitoring, flow profiling, and well and casing integrity monitoring, Typical examples of surface applications are pipeline and equipment vibration monitoring.
Post-conditions	DAS Instrument Box Connected, configured and ready for acquisition. QC Setup report delivered to DAS Project Engineer, Customer Project and Field Engineer. These requirements are covered by the DAS instrument box and optical path data object that are part of this standard.
Business Rules	
Notes	
Definitions	See Section 22.1.

22.2.5 Use Case: Perform Data Acquisition

Use Case Name	Perform Data Acquisition and Provide Configuration and Measured Data
Version	1.1
Goal	Acquire the required DAS measurements Refine the depth calibration (facility mapping) Provide operation log for time event association
Summary Description	In its simplest form the DAS Field Engineer presses start record and stop record. DAS Project Engineer may perform timely QC. The Customer Project Engineer may request previews of the acquisition data.

Actors	DAS Field engineer, DAS Project Engineer, DAS Instrumentation, Customer Field Engineer, Customer Project Engineer
Triggers	Client requests acquisition to commence.
Pre-conditions	Configuration of DAS Instrumentation complete. Feedback from customer on setup QC report received.
Primary or Typical Scenario	Perform acquisition for a vertical seismic profiling service.
Alternative Scenarios	Perform acquisition for other borehole or surface monitoring services.
Post-conditions	DAS data has been acquired to 'field media' This data can be transferred using the standard.
Business Rules	
Notes	Field media may be a different media to the media the final acquired data is delivered on.
Definitions	See Section 22.1.

22.2.6 Use Case: Post-Process DAS Data

Use Case Name	Post Process and Deliver DAS Data
Version	1.1
Goal	Optionally post process the acquired data after acquisition and deliver to customer or customer's Third Party System.
Summary Description	Post processing may include: • Refining the depth calibration (facility mapping) • Re-decimation or extraction of different frequency bands • Updating or adding meta data
Actors	• DAS Project Engineer
Triggers	Acquisition complete
Pre-conditions	Acquisition complete and stored in the PRODML DAS format.
Primary or Typical Scenario	Refining the depth calibration facility mapping based on further analysis. Re-decimation or extraction of different frequency bands Updating or adding metadata Converting acquired data to other industry standard formats such as SEG Y.
Alternative Scenarios	
Post-conditions	Post Processed DAS data available for transfer using desired delivery media in PRODML DAS format. Derived formats such as SEGY for vertical seismic profiles may also be produced. Details of post processing activities should be documented accordingly in the metadata of the delivered data.
Business Rules	

Notes	For specific DAS acquisition services (e.g., vertical seismic profiles or pipeline monitoring) the client may only be interested in the DAS post processed derived product (e.g., SEGY for VSP or a list of events for pipeline monitoring) and may agree with the DAS service provider to store and retain the field or reduced processed data for an agreed period.
Definitions	See Section 22.1.

Part VI: Other PRODML Data Objects

PRODML has several data objects that for historical reasons have neither been fully documented nor widely adopted. The combination of new PRODML development and resources constraints has left time to provide only an introduction and overview of these data objects. Where available, links to other related resources are provided.

The data objects covered in this section are:

- Wireline Formation Tester (Chapter 23)
- Product Volume (Chapter 24)
- Product Flow Model (Chapter 25)
- Time Series and Time Series Statistic (Chapter 26)
- WellTest (Chapter 27)
- Production Operation (Chapter 28)

23 Wireline Formation Tester

Acknowledgement

Special thanks to the following companies and the individual contributors from those companies for their work in designing, developing and delivering the Wireline Formation Tester data object: Weatherford.

23.1 Overview of Wireline Formation Tester and WftRun Data Object

This data object deals with the data generated by wireline formation tester (WFT) wireline tools. Pictures and diagrams of such tools can be seen in the slide set referenced below. This type of tool and its analysis typically generates data as follows:

- Measured Data
  - Time series at specified flowing intervals/locations
- Data pre-processing
  - Smooth/reduce
  - Identify/tag test periods/kinds
- Analysis
  - Output is reservoir/fluid parameters for the tested intervals
- Fluid Samples may be associated with intervals

Figure 23–1 shows a schematic diagram of such a tool and its typical uses.

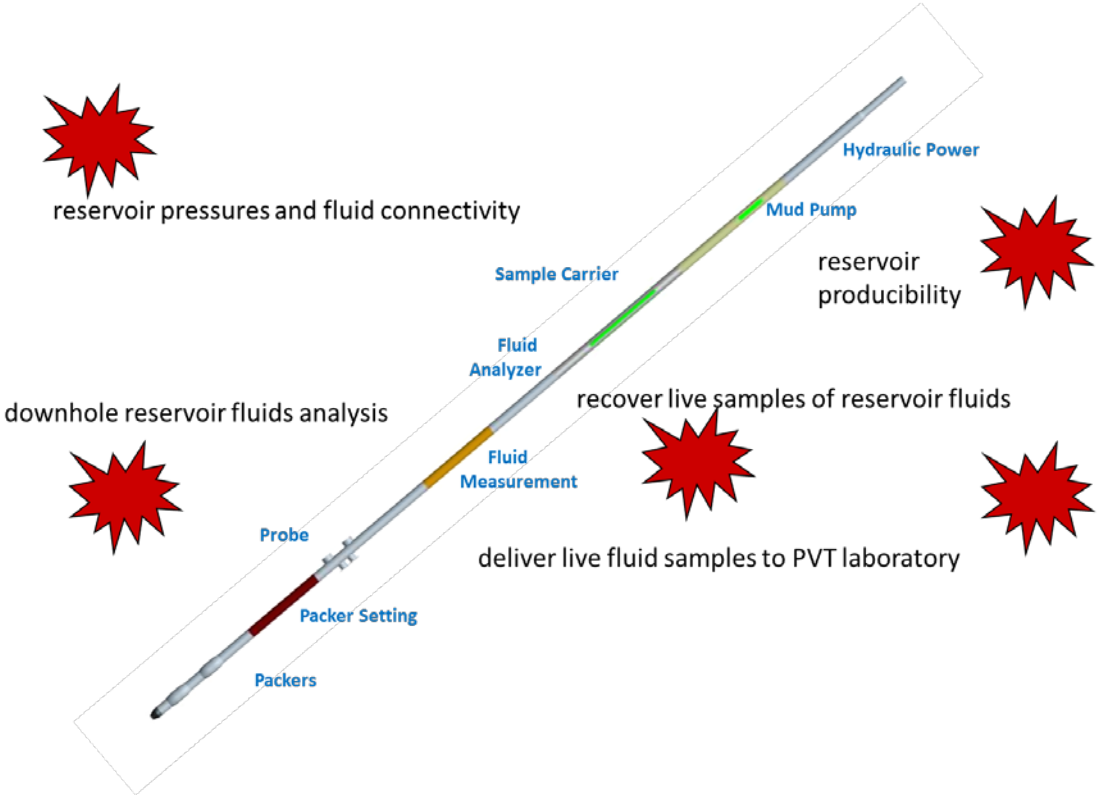


Figure 23–1. Schematic of typical wireline formation tester, with typical purposes for its use.

Figure 23–2 shows a diagrammatic wireline formation tester station with the kinds of data associated with it. A probe or packers are used to isolate a small section of wellbore and open it to flow into the tester

tool. Various sensors (pressure, temperature, fluid properties, etc.) then record the flowing and shut-in properties. Samples can be taken and conveyed back to surface in fluid sample containers. Results can be derived from the data collected, usually from analysis on specific Test periods within the time series.

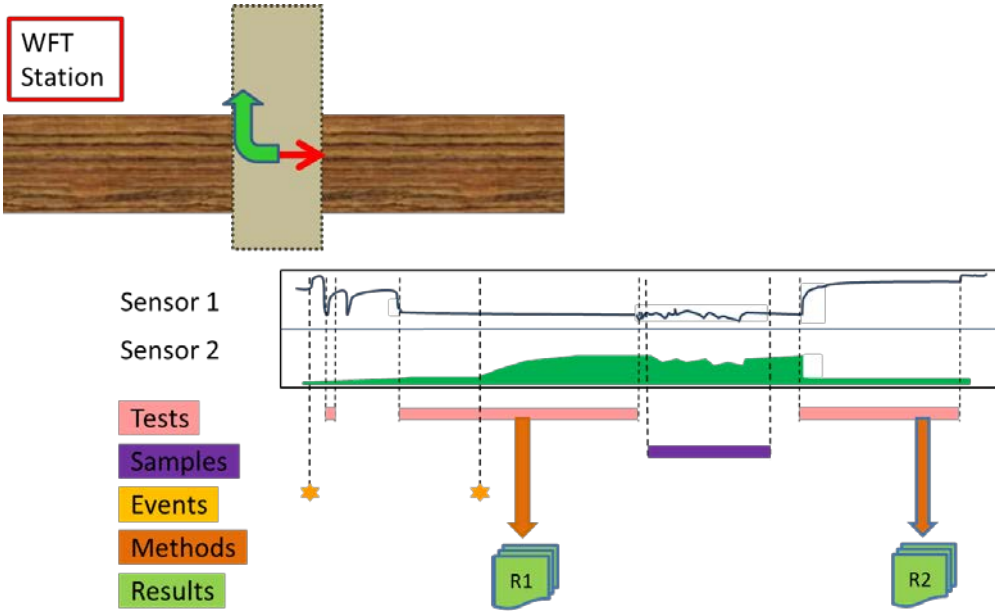


Figure 23–2. Illustrating a single wireline formation tester Station (arrow) with flow occurring from formation (green arrow), showing time series data divided into tests, sampling periods, events, from which results can be derived.

Figure 23–3 shows how these station results can then be combined at the level of the whole wellbore to yield property profiles (which have larger measurement scale than typical logs), and for certain properties (especially pressure), property gradients (for example, pressure/true vertical depth is indicative of fluid density).

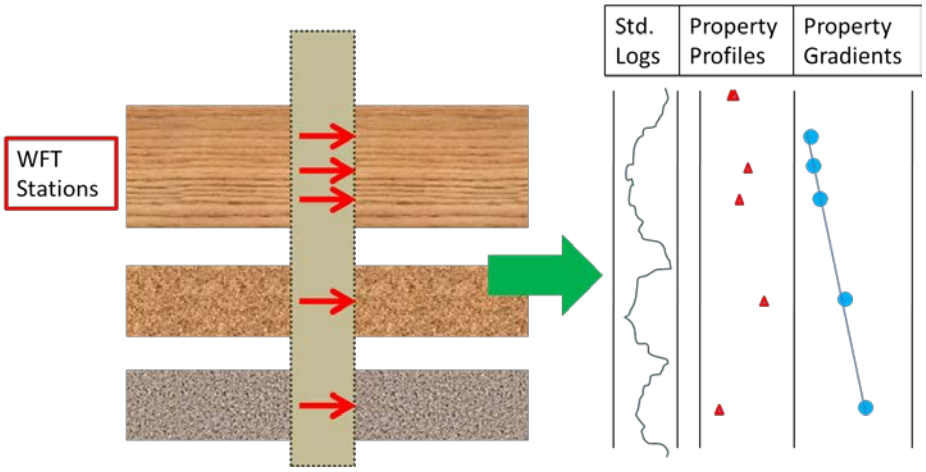


Figure 23–3. Illustrating multiple wireline formation tester Stations along the wellbore, generating property profiles and in some case cases, distinct property gradients.

The data object used for this purpose is WftRun “where “run” refers to a wireline run into the wellbore with the tool, collecting time series pressure transient data and/or fluid samples from multiple stations in the wellbore

The WftRun object can store the data about the overall Run, the stations, the division of the time series into tests, the sampling details, and the results at the level of test, station, or wellbore.

The time series data itself is not stored in the WftRun object, but in WITSML Log and Channel objects (see the mapping to other domain data objects shown in Figure 2–6).

23.2 WftRun Data Object Outline

A top-level class diagram for the WftRun can be seen in **Figure 23**–4. The major top-level elements of the data object are seen on the diagram and are as follows:

- WftRun – this describes the whole operation. It contains:
  - Wellbore ref
  - Time span
  - Depth span
  - Tie-in log refs
  - Stations
  - Results at wellbore level
- WftStation – this describes the data acquired at one Station, or flowing interval. It contains:
  - Tests
  - Samples
  - Events
  - Results at station level
- Test – to describe the events happening at a single Test within a Station (for example, a buildup test). It contains:
  - Kind of test
  - Test Data
  - Results at test level
- Sample Acquisition – to describe how a fluid sample was collected. It contains:
  - How the fluid sample was collected
  - Test Data (e.g. time series data) pertaining to the Fluid Sample acquisition.
  - Fluid Sample and Fluid Sample Container references – these relate the WftRun data object to the Fluid Sample and Fluid Sample Container data objects which comprise part of the Fluid and PVT Analysis section of PRODML. See Chapters 8 to 11 for details.
- Event – this is to describe events happening at one Station (example, tool being set or unset). It comprises a set of tags on a timeline. It contains:
  - Times
  - Durations
  - Event descriptions

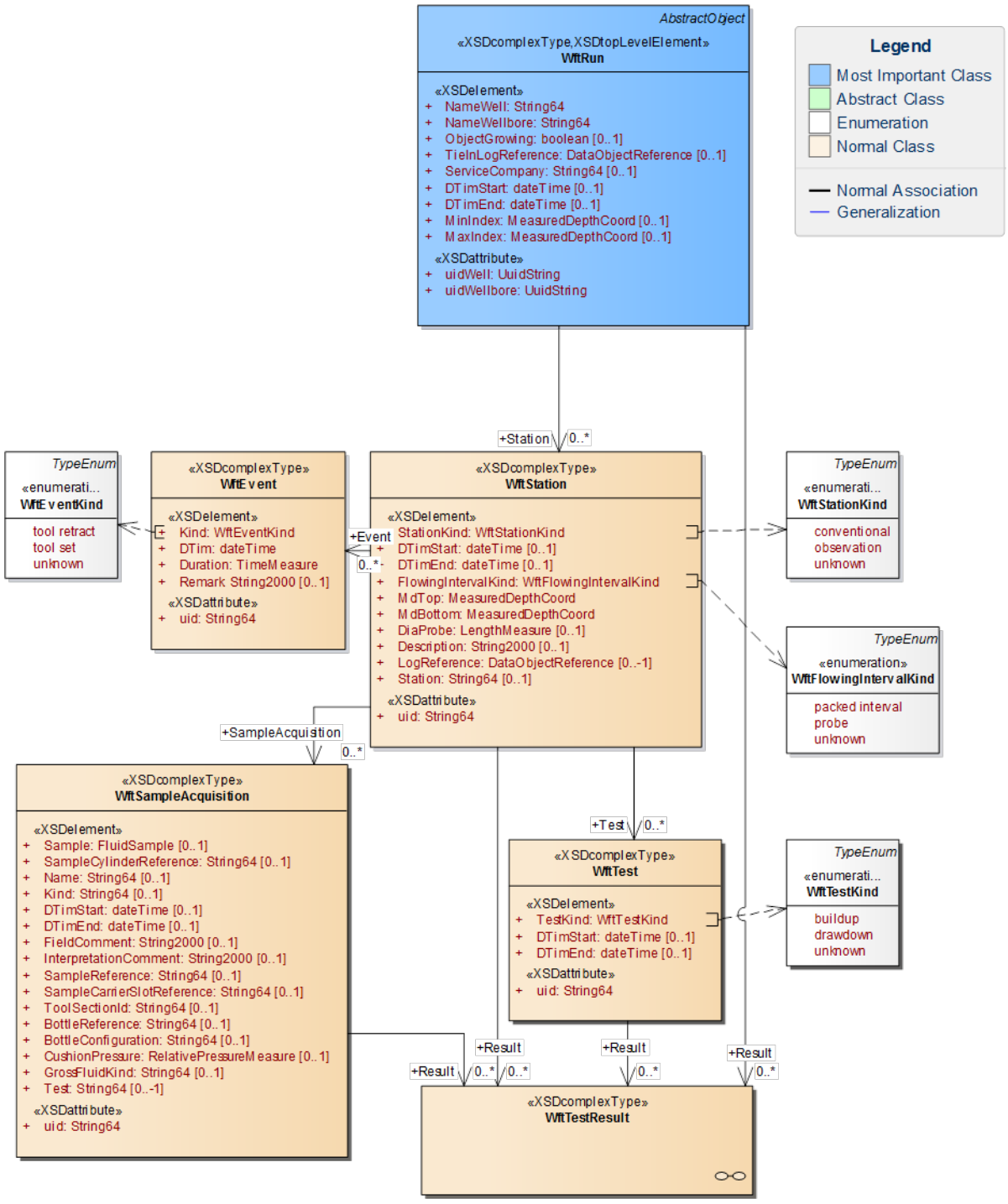


Figure 23–4. WftRun top-level class diagram.

The results from a wireline formation tester can be generated at multiple levels, and these are all available by the links to the WftTestResult class from:

- **WftRun top-level**: for results which apply to the whole operation or wellbore (e.g. fluid gradients).
- **Station**: for results which apply to the Station level such as a static reservoir pressure.
- **Test**: for results which apply to a single test such as permeability.

- **Sample**: for results derived from a sample.

See **Figure 23**–5 for the classes involved in the reporting of results from a WFT run. In this diagram, note the red rings around the links showing that the WftTestResult element is used from all four levels of data acquisition listed above. The diagram shows how each instance of WftTestData has a Role (currently, flow or pressure) and then one or more curve sections will be expected—these are references to the time series data in WITSML Channel objects. (For more information, see the *WITSML Technical Usage Guide*).

Parameters can also be attached to test results, and these can be input or output to the result (so for example, thickness *h* will typically be an input parameter to a buildup test, and permeability *k* will be an output).

It is also possible to link the test result to use as input, the output from another test result. This is useful where, for example, a pressure gradient is to be reported, and it is important to know which results at each station were used—for example “the pressure extrapolation of buildup 3 at station 5”. This allows an audit trail through the results to be preserved.

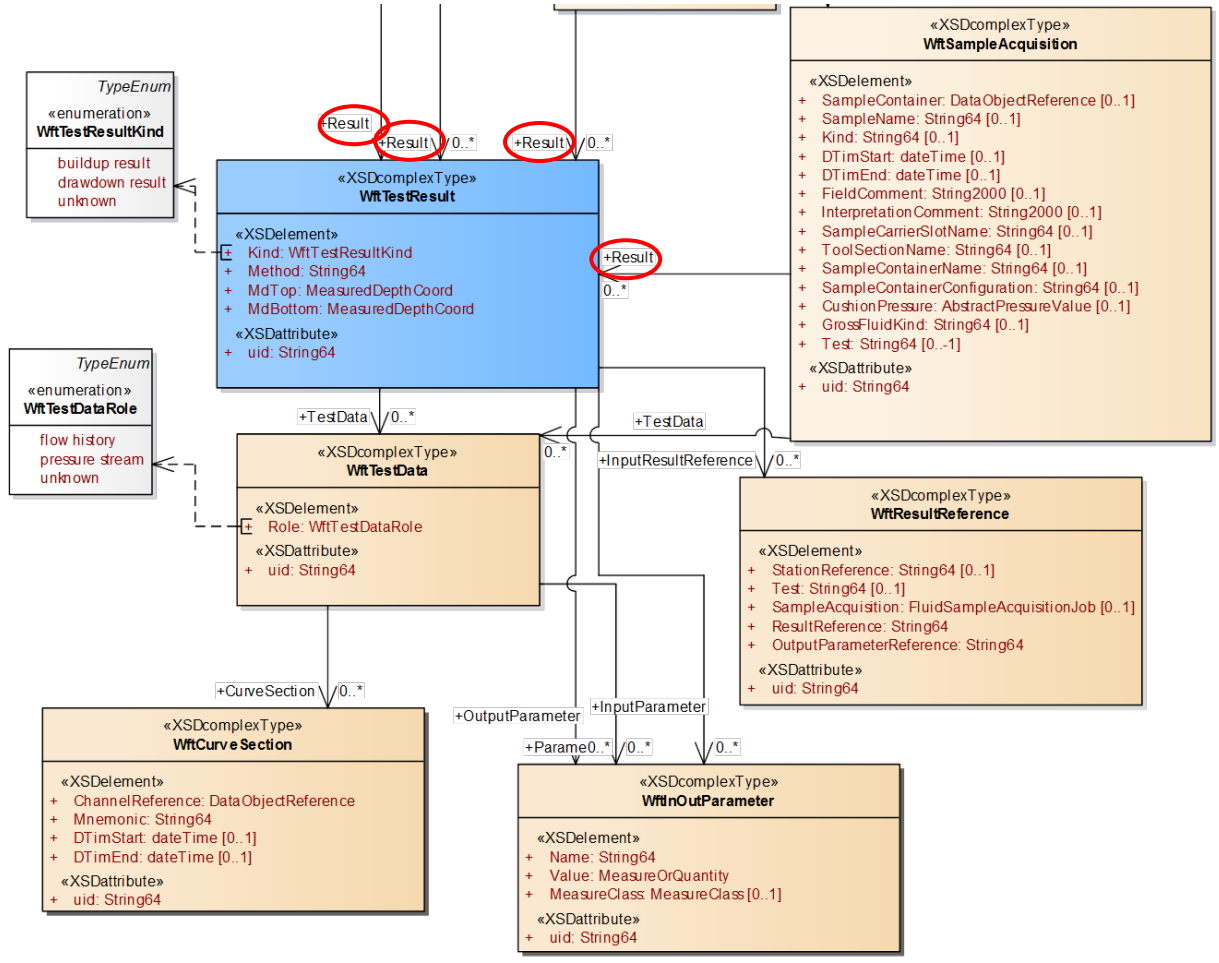


Figure 23–5. WftRun Results section.

For the further available documentation for this data object, see the slide set: *WFT Run Overview.pptx* which is provided in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files.

24 Product Volume

Acknowledgement

Special thanks to the following companies and the individual contributors from those companies for their work in designing, developing and delivering the Product Volume, Flow Network and WellTest data objects: the initial work was donated by ExxonMobil; this was then adapted for the NCS by, amongst others, EPIM, Statoil and TietoEnator. Saudi Aramco, Peformix (now Baker Hughes) and Atman Consulting then did various modifications and generated useful documentation.

24.1 Usage of Product Volume vs. Simple Product Volume Reporting

The Product Volume data object is used as the standard for reporting production on the Norwegian Continental Shelf (NCS). It has been used elsewhere for volume reporting, and also for production surveillance.

However, it has often been found to be overly complex for the task of volume reporting, which is a core requirement for PRODML in order to support production management. Product Volume is very flexible and can be used for many purposes. The drawback to having this flexibility is that it is somewhat complex to understand, and it sometimes can be found to offer multiple ways of achieving the same result. In the case of the NCS, by dint of a central regulatory authority defining exactly how Product Volume is to be used by reporting companies, the data standard has been successfully deployed and rolled out across a major oil province.

Because of the drawbacks mentioned above, and the desire to have a simple standard for volume reporting, the Simple Product Volume Reporting (SPVR) capability has been added to PRODML. It is believed that parties wishing to exchange volumes data without needing the additional data and flexibility allowed by Product Volume will chose to standardize on SPVR. See Chapters 3 to 7 for full details on SPVR.

The main additional functionality provided by Product Volume (PV) compared to SPVR is:

- PV works with the Product Flow Network (see Chapter 25) so that flows can be reported at precisely defined points within a network. Time Series (see Chapter 26) can also reference the Product Flow Network making detailed production surveillance data exchange possible. SPVR uses a list of Reporting Entities (e.g. well, platform) and then reports volumes against these. It is possible to arrange reporting entities in hierarchies, but there is no sense of a network. SPVR does not support time series data.
- PV has the concept of *flows* of fluids which can have function (e.g., production, gas lift) and which can be broken down into *products* which describe the composition of the fluid. SPVR only deals with products.
- PV also supports the transfer of Facility Parameters which are non-flow data of any facility. SPVR again only supports volume related flow, although it does have a Well Production Parameters data object for exchanging well parameters (the most common requirement).

24.2 Overview of Product Volume

The Product Volume data object can be used to report production flows or other parameters. For instance, it can be used to report the daily allocated volume of oil production for a well or group of wells. It could also be used to report other characteristics (pressure, temperature, flow rate, concentrations, etc.) associated with a specific wellhead. It uses a general hierarchy of:

Product Volume

Facility (wellhead, separator, flow line, choke, completion ...)

Parameter Set (block valve status, reciprocating speed, available room ...)

Parameter

Flow (production, injection, export, import, gas lift ...)

Product (oil, water, gas, CO2, oil-gas, cuttings ...)

Period (instant, day, month, year …)

temperature

pressure

flow rate

Parameter Sets allow time varying "usage" parameters to be defined for the facility. For example, a valve status may be toggled from "open" to "closed" to indicate that a well is offline.

Flows allow reporting for a flow of a product. For example, it may be used to specify the rate of oil produced for a specified well.

The relevant enumerations found in the *enumValuesProdml.xml* file are as follows:

- **Reporting Periods** (e.g. day, month, year, etc.) are given in the *ReportingPeriod* enumeration.
- **Facility Kinds** (e.g. wellhead, separator, flow line, choke, etc.) are given in the *FacilityParameter* enumeration.
- **Facility Parameters** (e.g. block valve status, reciprocating speed, etc.) are given in the *FacilityParameter* enumeration.
- **Flow Kinds** (e.g. production, injection, export, etc.) are given in the *ReportingFlow* enumeration.
- **Flow Qualifiers** (e.g. measured, allocated, etc.) are given in the *FlowQualifier* and *FlowSubQualifier* enumerations.
- **Product Kinds** (e.g. oil, water, gas, etc.) are given in the *ReportingProduct* enumeration.

The combination of Flow Kind and Flow Qualifier(s) are used to characterize the underlying nature of the flow. For example, the following combination is used to indicate a production flow that is measured.

```
<flow>
  <kind>production</kind>
  <qualifier>measured</qualifier>
     ...
<flow>
```

Similarly, the following combination is used to indicate an injection flow that is simulated.

```
<flow>
  <kind>injection</kind>
  <qualifier>simulated</qualifier>
     ...
<flow>
```

24.3 Product Volume Associated with Product Flow Model

A product flow network is not mandatory when using product volume. It is quite acceptable simply to use names on Facility elements under which the flows and products will be reported (see the hierarchy of product volume shown in Section 24.2).

However, where it is desired to exchange the detailed network model, typically so that the flow paths and the precise ports with volume data being reported can be defined, then a product flow model can be used (probably only at the outset and then whenever it changes, with product volume being used each time dynamic data is to be exchanged). Having used product flow model (refer to Chapter 25 for details on this), then product volume can now be associated with any given unit in the product flow model. The XML snippet in **Figure 24**–1 shows how to reference the associated unit and port from a Product Volume Report. Note that this snippet shows PRODML V1.x style; the content has not changed in v2.

The *unit* element provides the name and *uidRef* attribute that refers to the unique identifier for the flow unit **C** in the previous simple flow network example. Similarly, the *port* element provides the name and unique identifier for the port associated with a reported flow.

```
<productVolume>
  <name>Report for simple flow network example</name>
  <dTimStart>2012-01-01T00:00:00+03:00</dTimStart>
  <dTimEnd>2012-01-01T01:00:00+03:00</dTimEnd>
  <productFlowModel uidRef="PFM01">Simple Flow Network Example</productFlowModel>
  <!-- Well A -->
  <facility>...</facility>
  <!-- Well B -->
  <facility>...</facility>
  <!-- Production Header -->
  <facility>
    <name kind="manifold"
          siteKind="production header"
          uidRef="HDR01"
          namingSystem="aramco.com">Production Header</name>
    <unit uidRef="U03">C</unit>
    <!-- C-P outlet oil production flow -->
    <flow>
      <kind>production</kind>
      <port uidRef="5">C-P outlet</port>
      <direction>outlet</direction>
      <qualifier>measured</qualifier>
      <product>
        <kind>oil</kind>
        <period>
          <dTim>2012-02-01T00:00:00+03:00</dTim>
          <flowRateValue>
            <flowRate uom="bbl/hr">10.0</flowRate>
          </flowRateValue>
          <pres uom="psi">600</pres>
        </period>
        <period>
          <dTim>2012-02-01T01:00:00+03:00</dTim>
          <flowRateValue>
            <flowRate uom="bbl/hr">12.0</flowRate>
          </flowRateValue>
          <pres uom="psi">610</pres>
        </period>
      </product>
    </flow>
  </facility>
</productVolume>
```

Unit Reference

Port Reference

Flow rate and pressure at a specified time

Figure 24–1. Product Volume Report referencing Product Flow Model unit & port

24.4 Further Information on Product Volume

Product Volume has not been updated during the development of PRODML version 2 (other than to adopt the style of XML as defined in the CTA). This is primarily to keep the data model compatible with the usage of version 1.x Product Volume, as used primarily on the NCS. There is no exhaustive usage guide

for product volume, product flow model, and time series, as can work together as outlined in this chapter, Chapter 25, and Chapter 26. However, there are documentation sources that use version 1.x examples that are still applicable to version 2 because the data model has not changed.

- Online documentation for the NCS. This defines a standard of patterns for the usage of product volume for volume reporting to partners and regulators via the EPIM Reporting Hub. This is a very large scale adoption of PRODML where additional business rules have been added to define the norms for the usage of the base standards.
  - EPIM Reporting Hub introduction https://epim.no/erh/
  - EPIM Reporting Formats https://epim.no/erf/ . These are very similar schemas to Product Volume although in the earlier v1.x style. They have modified for use in the reporting in Norway but may be useful to review alongside the example linked to below.
  - Norwegian Regulator's introduction http://www.npd.no/en/Reporting/Production/Reporting-production-data-to-the-Norwegian-Petroleum-Directorate/
  - Worked example at http://www.npd.no/Global/Norsk/7-Rapportering/Produksjon/generell-felt.xml
- A PRODML companion document which was produced as part of an extension to PRODML further to a change request from a member organization. This extension added a number of features to make production surveillance easier to implement. The document deals with the interplay between product volume, product flow model and time series data objects to be able to use them for production surveillance. This document is entitled *PRODML Product Volume, Network Model & Time Series Usage Guide* and it can be found in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files. Extracts from that document have been included here. The document shows PRODML v1.x style schemas and examples but the data model has not changed in v2.x.

25 Product Flow Model

Acknowledgement

For Acknowledgements of detailed contributors, see Chapter 24.

25.1 Overview of Product Flow Model

The Product Flow Model data object can be used to define a directed graph of flow connections (**Figure 25**–1). The basic building block is a Unit which can be used to define the flow behavior of any facility (where the term facility represents any use of equipment to perform a function) such as a separator, a wellhead, a valve, a flow line. It uses a general hierarchy of:

Model (collection of networks)

Network (collection of connected units)

Unit (black box with ports)

Port (allows flow in or out)

Node (allows ports to connect)

The Network represents the internal behavior of the model or a unit in another network and is a collection of connected units. A Unit is essentially a black box that can represent anything (big or small). Ports allow flow in or out of a Unit. Nodes are used to connect ports.

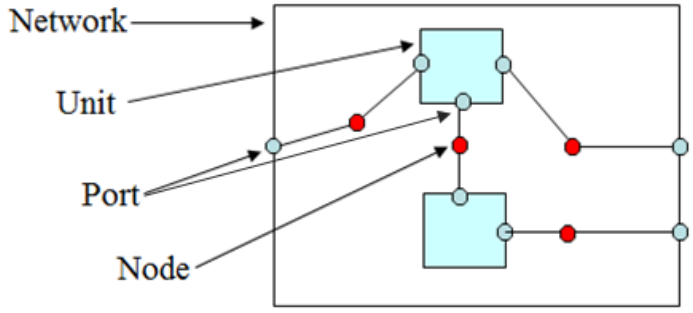


Figure 25–1. Product Flow Model data object.

In any given Product Flow Model, the following is assumed:

- Steady state fluid flow across nodes and ports. That is, pressure is constant across internally and externally connected ports and nodes.
- Conservation of mass across a node or port.
- Pressure can vary internally between ports on a unit.
- Connections between models should be one-to-one so that mass balance concerns are internal to each model.

A variety of models may be created and used for different systems (**Figure 25**–2). For instance, a production accounting system will have a different model than a production operations dashboard that is used to monitor real-time data from a facility. However, by using PRODML, these various models may be exchanged using the same standard format.

Note that Product Flow Model is also used in the Optical Path object (see Section 14.3) in distributed sensing (DTS and DAS), to show the connection of the components in a fiber optical path.

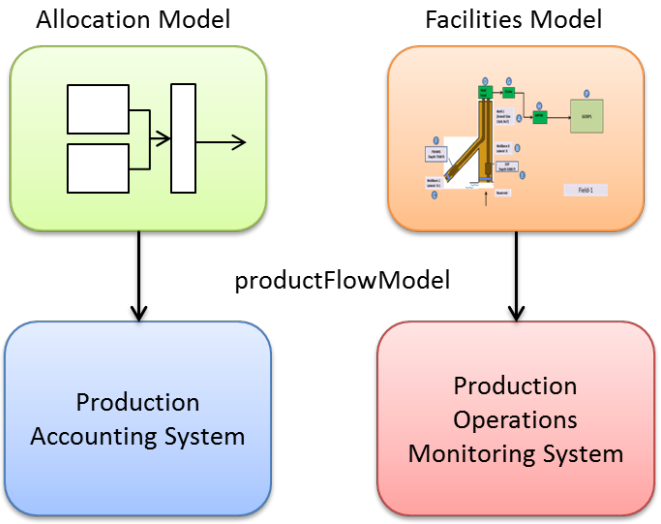


Figure 25–2. Various network models.

25.2 Simple Flow Network Example

The purpose of this example is to provide a very simple example of how to construct a PRODML Product Flow Model in order to introduce the overall concepts that will be used in a more complex example in the section

Figure 25–3 shows a very simple production network. This network consists of two producing oil wells connected to a pipeline. Although wells typically product a combination of oil, gas, and water, we will only consider the flow of oil for the purposes of this example.

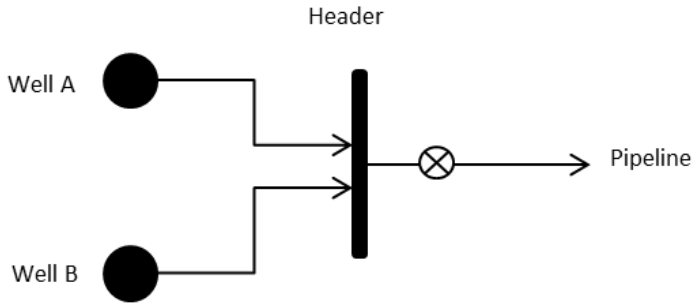


Figure 25–3. Simple oil production network.

25.2.1 Product Flow Model Construction

When constructing a Product Flow Model, it is important to differentiate between the “product” type measurements (e.g., oil rate, gas rate, pressure, etc.) which are reported at Ports, and the “facility” type measurements (e.g., motor speed, choke position, valve status, etc.) which can be measured and reported within a unit.

Information about a flow is best reported where it leaves or enters Units (facilities) which may modify the flow (e.g., separators, manifolds etc.). On the other hand facility parameters are internal to the “workings” of a facility.

IMPORTANT: The Product Flow Model assumes:

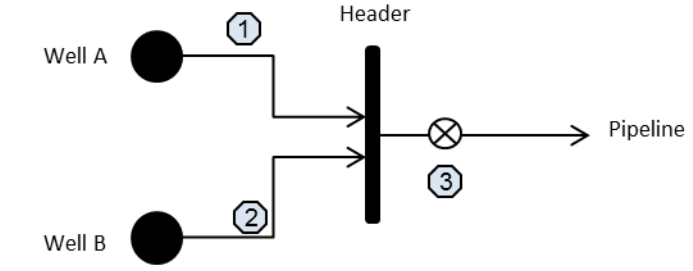
- Steady state fluid flow across nodes and ports (i.e. pressure is constant across internally and externally connected ports and nodes).
- Conservation of mass across a node or port.
- Pressure can vary internally between ports on a unit.
- Connections between models should be one-to-one so that mass balance concerns are internal to each model.

Construction of a PRODML flow network can be summarized in the following steps, which are further explained below:

1. Draw the real world diagram indicating measurement points.
2. Draw boundaries to include facility parameters and exclude flow measurements.
3. Make each boundary a unit in the flow network and identify the ports.

Step 1: Draw the Diagram

First, draw the real-world diagram indicating measurement points (**Figure 25**–4**)**.

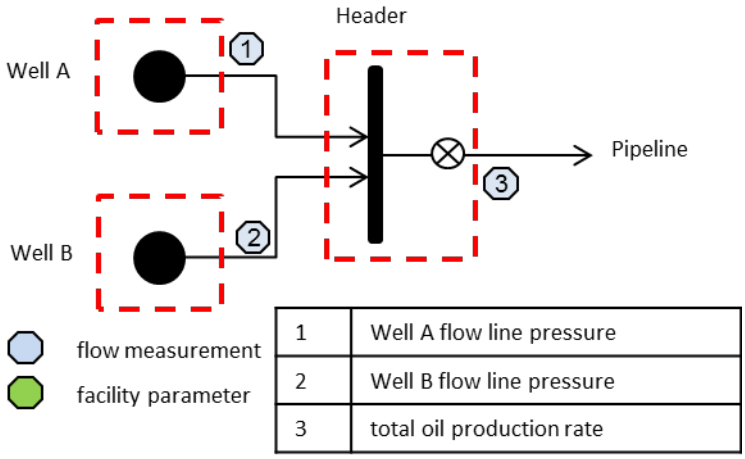


1	Well A flow line pressure
2	Well B flow line pressure
3	total oil production rate

Figure 25–4. Measurement points for a simple flow network.

Step 2: Draw Boundaries

The next step is to draw boundaries around items with no flow measurements (**Figure 25**–5).

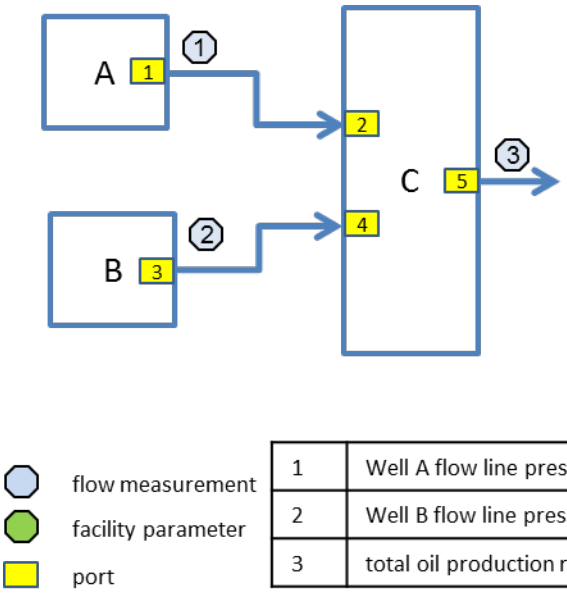


1	Well A flow line pressure
2	Well B flow line pressure
3	total oil production rate

Figure 25–5. Unit boundaries for a simple flow network.

Step 3: Make Each Boundary a Flow Unit

Finally, make each boundary a unit in the flow network and label each unit and node with a unique name (**Figure 25**–6**)**.



1	Well A flow line pressure
2	Well B flow line pressure
3	total oil production rate

Figure 25–6. Simple network units and nodes.

Table 1 summarizes the units of the flow network, assigned unique identifiers, and associated facility kinds.

Table 1 Simple Network Units					
Unit Name	**Unit ID**	**Facility ID**	**Facility Kind**	**Kind Qualifier**	**Measurements**
A	U01	W01	well		(outlet) pressure
B	U02	W02	well		(outlet) pressure
C	U03	HDR01	manifold	production header	(outlet) oil flow rate (outlet) pressure

Table 2 Simple Network Ports		
Port	**Direction**	**Connected Node**
1	outlet	A-C
2	inlet	A-C
3	outlet	B-C
4	inlet	B-C
5	outlet	C-P

Table 3 Simple Network Flow Properties				
Port	**Property**	**Flow**	**Product**	**Qualifier**
1	pressure	production	oil	measured
3	pressure	production	oil	measured
5	pressure	production	oil	measured
5	flow rate	production	oil	measured

Figure 25–7 illustrates the definition of port 5 in XML. Note that this snippet shows PRODML v1.x style; the content has not changed in v2.

```
<port uid="5">
  <name>C outlet</name>
  <direction>outlet</direction>
  <!-- Node C-P -->
  <connectedNode>
    <node>C-P</node>
    <dTimStart>2010-01-01T00:00:00.000Z</dTimStart>
  </connectedNode>
  <!-- oil production flow -->
  <expectedFlowProduct>
    <flow>production</flow>
    <product>oil</product>
    <qualifier>measured</qualifier>
  </expectedFlowProduct>
  <!-- oil flow rate (measured) to pipeline -->
  <expectedFlowProperty>
    <property>flow rate</property>
    <dTimStart>2012-01-01T00:00:00.000Z</dTimStart>
    <expectedFlowProduct>
      <flow>production</flow>
      <product>oil</product>
      <qualifier>measured</qualifier>
    </expectedFlowProduct>
  </expectedFlowProperty>
  <!-- (outlet) pressure -->
  <expectedFlowProperty>
    <property>pressure</property>
    <dTimStart>2012-01-01T00:00:00.000Z</dTimStart>
    <tagAlias namingSystem="osi.aramco.com">H01_OutletPressure</tagAlias>
    <expectedFlowProduct>
      <flow>production</flow>
      <product>oil</product>
      <qualifier>measured</qualifier>
    </expectedFlowProduct>
  </expectedFlowProperty>
</port>
```

Figure 25–7. Port definition example in XML

25.3 Further Information on Product Flow Network

Product Flow Network has not been updated during the development of PRODML version 2 (other than to adopt the style of XML as defined in the CTA). There is no exhaustive usage guide for product volume, product flow model and time series, as can work together as outlined in Chapter 24, this chapter and Chapter 26. There are however documentation sources which use version 1.x examples but which are still applicable to version 2 since the data model has not changed.

- A companion document as described in Section 24.3, entitled *PRODML Product Volume, Network Model & Time Series Usage Guide* and it can be found in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files. Extracts from that document have been included here. The document shows PRODML v1.x style schemas and examples but the data model has not changed in v2.x.
- The Optical Path usage of Product Flow Network is shown in Section 14.3.3.

26 Time Series

Acknowledgement

Special thanks to the following companies and the individual contributors from those companies for their work in designing, developing and delivering the Time Series data object: Shell, OSISoft, Petrotechnical Data Systems.

26.1 Time Series in Energistics

The time series data object (and companion time series statistic) are available for the exchange of time series data, and have been part of PRODML since v1.2. The PRODML Time Series data object is intended for use in support of smart fields or high-frequency historian type interactions.

It is likely that this data object will be replaced in time by usage of data objects associated with the Energistics Transfer Protocol (ETP) which has been developed for transfer of data channels streamed in real time. For backwards compatibility, it is included here.

26.2 Time Series Data Object Overview

The time series data object is intended for use in transferring time series of data, e.g. from a historian.

The Time Series data object describes a context free, time based series of measurement data for the purpose of targeted exchanges between consumers and providers of data services. This is intended for use in support of smart fields or high-frequency historian type interactions, not reporting. It provides a "flat" view of the data and uses a set of keyword-value pairs to define the business identity of the series, as described in the following generalized hierarchy.

Time Series Data

Meta Data

Keyword Name/Value Pairs (asset identifier, qualifier, product, flow …)

Units of Measure (psi, rpm, mA, m …)

Measure Class (electric current, thermodynamic temperature, …)

Time/Value Pairs

The keyword value pairs are used to characterize the underlying nature of the values. The key value may provide part of the unique identity of an instance of a concept or it may characterize the underlying concept. For example, the following keyword value pairs are used to indicate the measured production flow of oil.

```
<key keyword="flow">production</key>
<key keyword="product">oil</key>
<key keyword="qualifier">measured</key>
```

Keyword	KindQualifier
`asset identifier`	A formatted URI identifier of the asset (facility) related to the value. This captures the kind of asset as well as the unique identifier of the asset within a specified context (the authority). The identifier may define a hierarchy of assets. Refer to the *CTA Technical usage Guide* for more information on Energistics identifiers.
`qualifier`	A qualifier of the meaning of the value. This is used to distinguish between variations of an underlying meaning based on the method of creating the value (e.g., measured versus simulated). The values associated with this keyword must be from the list

Keyword	KindQualifier
	defined by type *FlowQualifier*.
`subqualifier`	A specialization of a qualifier. The values associated with this keyword must be from the list defined by type *FlowSubQualifier*.
`product`	The type of product that is represented by the value. This is generally used with things like volume or flow rate. It is generally meaningless for things like temperature or pressure. The values associated with this keyword must be from the list defined by type *ReportingProduct*.
`flow`	Defines the part of the flow network where the asset is located. This is most useful in situations (e.g., reporting) where detailed knowledge of the network configuration is not needed. Basically, this classifies different segments of the flow network based on its purpose within the context of the whole network. The values associated with this keyword must be from the list defined by type *ReportingFlow*.

26.3 Asset Identifier

In a Time Series Data object, the *key* element with the keyword of *asset identifier* element refers back to the Product Flow Model. The asset identifier key is an Energistics URI identifier of the asset (facility) related to the value. The identifier may define a hierarchy of assets. For more information on PRODML identifiers, see the *Energistics Identifier Spec*, which was included in the PRODML download.

The example in **Figure 26**–1 shows an asset identifier for the *manifold* facility (yellow highlight) assigned the unique identifier of *HDR01* (blue highlight) in the naming system *aramco.com* (green highlight).

```
<timeSeriesData>
  <name>Production Header outlet pressure</name>
  <key keyword="asset identifier">prodml://aramco.com/manifold(HDR01)</key>
  <key keyword="flow">production</key>
  <key keyword="product">oil</key>
  <key keyword="qualifier">measured</key>
  <unit>psi</unit>
  <measureClass>pressure</measureClass>
  <doubleValue dTim="2012-02-08T13:02:00+03:00">747.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:02:01+03:00">746.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:03:02+03:00">745.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:02:03+03:00">746.0003</doubleValue>
  <doubleValue dTim="2012-02-08T13:02:04+03:00">748.613</doubleValue>
```

Figure 26–1. Time series asset identifier example.

26.4 Value Status Attribute

In a Time Series Data object, the value *status* attribute is used as an indicator of the quality of the value.

IMPORTANT: If the *status* attribute is absent and the value is not set to *NaN*, then the data value can be assumed to be good with no restrictions.

The example in **Figure 26–2** shows several readings, each with a status of frozen, which indicates the sensor has been reading the same value for a specific period of time.

```
<timeSeriesData>
  <name>Production Header outlet pressure</name>
  <key keyword="asset identifier">prodml://aramco.com/manifold(HDR01)</key>
  <key keyword="flow">production</key>
  <key keyword="product">oil</key>
  <key keyword="qualifier">measured</key>
  <unit>psi</unit>
  <measureClass>pressure</measureClass>
  <doubleValue dTim="2012-02-08T13:00:00+03:00" status="frozen">747.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:01:00+03:00" status="frozen">747.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:02:00+03:00" status="frozen">747.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:03:00+03:00" status="frozen">747.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:04:01+03:00">746.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:05:02+03:00">745.7316</doubleValue>
  <doubleValue dTim="2012-02-08T13:06:03+03:00">746.0003</doubleValue>
  <doubleValue dTim="2012-02-08T13:07:04+03:00">748.613</doubleValue>
```

Figure 26–2. Value status example.

26.5 Time Series Statistic Object

This data object is a companion to the Time Series Data object. It has the same elements as time series data including the *keyword* concept, to identify a time series of data. Then, rather than the time series data itself, the time series statistic object has elements to define the minimum and maximum time values, between which the data statistics apply. This is followed by a set of statistical data applying to the time series data, as follows:

- Minimum
- Maximum
- Sum
- Mean
- Median
- Standard deviation
- Time at Threshold, which shows the time during which data sits within a range as follows:
  - Threshold minimum (optional): defines lower bound of the range.
  - Threshold maximum (optional): defines upper bound of the range.
  - Duration: the sum of the time over which data was within the range defined above.

26.6 Further Information on Time Series

Time Series has not been updated during the development of PRODML version 2 (other than to adopt the style of XML as defined in the CTA). There is no exhaustive usage guide for product volume, product flow model and time series, as can work together as outlined in Chapter 24, Chapter 25 and this chapter. There are however documentation sources which use version 1.x examples but which are still applicable to version 2 because the data model has not changed.

- A companion document as described in Section 24.3, entitled *PRODML Product Volume, Network Model & Time Series Usage Guide,* can be found in the folder: energyml\data\prodml\v2.0\doc in the Energistics downloaded files. Extracts from that document have been included here. The document shows PRODML v1.x style schemas and examples but the data model has not changed in v2.x.

27 WellTest

Acknowledgement

For Acknowledgements of detailed contributors, see Chapter 24.

27.1 WellTest Data Object Overview

The WellTest data object is a standalone object for exchanging well tests of certain types. It is a companion object to the product volume object described in Chapter 24. It has a set of identifying elements and common elements, and then one of three types of well test is defined. The three types are:

- Production test
- Injection test
- Fluid level test

The production and injection tests support inclusion of volumes produced, or flow rates during the test.

Note that pressure transient well testing is not supported by this WellTest data object, nor by other Energistics data models, except in the wireline formation testing context in the WftRun data object (see Chapter 23). The term ‘well test’ refers here just to steady-state conditions testing.

The data object also supports validation of the well test, allowing a series of validation steps to be recorded by step kind.

The schema is relatively simple, stand alone, and should be understandable by looking at the UML diagram or the schema in a development environment.

The patterns in the well test schema follow those in the product volume schema, and hence well test can be thought of as a companion data exchange model to product volume.

The well test data object is reasonably comprehensive and also flexible and, like the product volume object as described in Section 24.1, the flexibility comes at the price of some complexity. Because of this drawback mentioned above, and the desire to have a simple standard for volume reporting, the Simple Product Volume Reporting (SPVR) capability has been added to PRODML. It is believed that parties wanting to exchange production well test data without needing the additional data and flexibility allowed by well test will chose to standardize on SPVR. For full details on SPVR, see Chapters 3 to 7.

SPVR includes two objects suitable for well test data: Production WellTest and the Well Production Parameters. For details, see Sections 5.4, 6.4.1, and 6.4.2. These objects offer fewer options and functionality than well test, but as is the case for SPVR versus Product Volume (see Section 24.1), the intention is that the simple objects will be easier to understand and implement for most purposes.

28 Production Operation

Acknowledgement

For Acknowledgements of detailed contributors, see Chapter 24.

28.1 Production Operation Data Object Overview

The Production Operation Data object is a further companion to the product volume object described in Chapter 24. It enables the exchange of production operation data along the lines of a “morning report” for production operations. The volumes would be expected to be transferred using product volume.

Production operation has an offshore operation orientation, reflecting its origins in the Norwegian Continental Shelf reporting requirements.

Production operation has a set of identifying elements and common elements, and then has a repeating Installation Report element. This identifies the installation, so that one transfer can deal with a number of related installations, and then it contains the following categories of data:

- Count of crew, by crew type and beds available.
- Work hours performed day, month and year to date.
- Production operation HSE (see below).
- Production operation activity (see below).

Production operation HSE is a list of statistics concerning HSE such as number of incidents, time since last incident, and similar. It also contains a weather report with detailed weather attributes.

Production operation activity is a more detailed capability to report operational activity in a number of areas:

- Water quality
- Alarms
- Cargo ship operations (which includes volumes loaded, duplicating a part of product volume, but from the perspective of an operational report rather than a hydrocarbon accounting requirement).
- Marine operations (supply and standby vessels and related).
- Shutdowns (including time, duration, loss of oil and gas production).
- Lost production (which can also be referred to as *deferred production* since it is not “lost” in the usual sense of the word: in SPVR, deferred production was the term used; see Section 6.3.2, item 8). This element has a volume lost amount, and also an enum list of reason for loss. In SPVR this enum was not listed since it was understood that each company would have its own set of reason codes. The enum values reflect agreed values for NCS reporting.
- Comments on operations, which are typed by an enum list.

For the UML diagram of production operation activity, giving a good overview, see **Figure 28**–1.

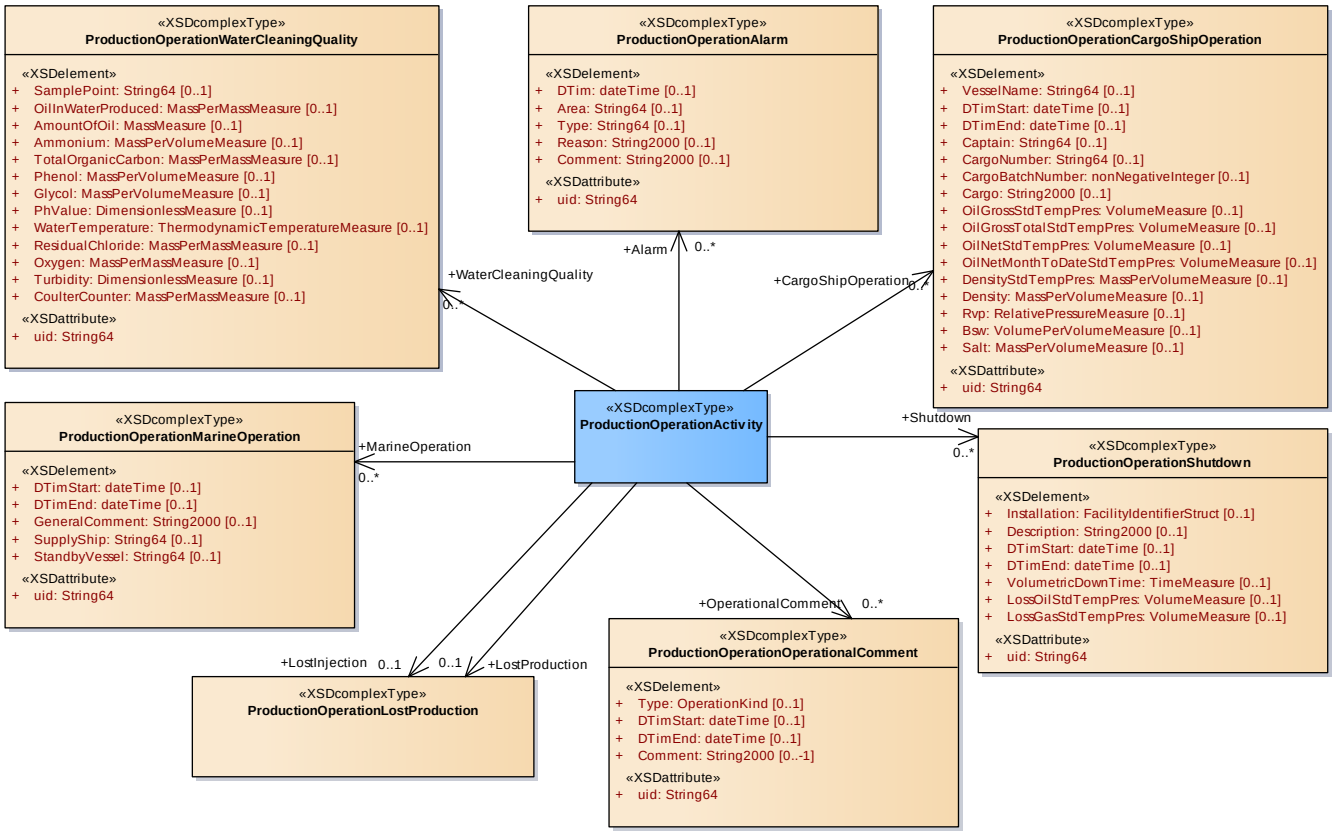


Figure 28–1. Model of production operations activities.