PRODML Product Volume, Network Model & Time Series Usage Guide

PRODML Overview	The PRODML standard facilitates data exchange among the many software applications used along the E&P subsurface workflow, which helps promote interoperability and data integrity among these applications and improve workflow efficiency.
Version of Standard	For PRODML v2.0, v2.1 and v2.2
Version of Document	1.1
Date published	June 17, 2019
Prepared by	Energistics and PRODML SIG
Abstract	This guide provides a domain/business overview of how PRODML can be used to model production networks and the flows.
Document type	Usage guide
Language	U.S. English
Keywords:	standards, energy, data, information, process, PRODML

ENERGISTICS
ADOPT > ADVANCE > ACCELERATE

Amendment History			
Standard Version	**Document Version**	**Date**	**Comment**
2.2	1.1	May 16, 2022	This document has NOT been updated specifically for v2.2. The concepts described are still applicable to the latest version of PRODML. NOTE: The Usage, Intellectual Property Rights, and Copyright statement on the previous page was updated slightly for Energistics under The Open Group.
2.0/2.1	1.1	June 17, 2019	This document was updated with the latest version of Energistics templates/branding. No content updates were made to this document for PRODML v2.1; the content for PRODML v2.0 applies to PRODML v2.1.
2.0	1.0	Dec 9, 2016	For publication with PRODML v2.0.

Table of Contents

1 Introduction

This document provides an overview of important concepts for PRODML v2.0, namely the product volume, network, and time series data models and related key concepts.

NOTE: While not all code examples have yet been updated to the v2.0 style, all concepts remain relevant.

1.1 Audience, Purpose and Scope

This guide is intended for information technology (IT) professionals who are implementing PRODML into software. This guide assumes that the reader has a good general understanding of PRODML and XML, and a basic understanding of the exploration and production (E&P) domain and workflows.

1.2 Resource Set

For a complete list of resources and documentation for PRODML, see the *PRODML Technical Usage Guide*, which is included in the PRODML download (zip file) available on the Energistics website.

For more information, see: [https://www.energistics.org/download-standards/.](https://www.energistics.org/download-standards/)

2 PRODML Concepts

PRODML is an industry initiative to provide open, non-proprietary, standard interfaces between software tools used to monitor, manage and optimize hydrocarbon production.

This guide focuses on three PRODML data objects:

- **Product Flow Model** – static model of flow connections used to create facility models.
- **Product Volume Report** – for reporting production flows (e.g., oil, gas, water, etc.) or other parameters (e.g., valve status, reciprocating speed, etc.).
- **Time Series** – for exchanging simple time based series of measurement data.

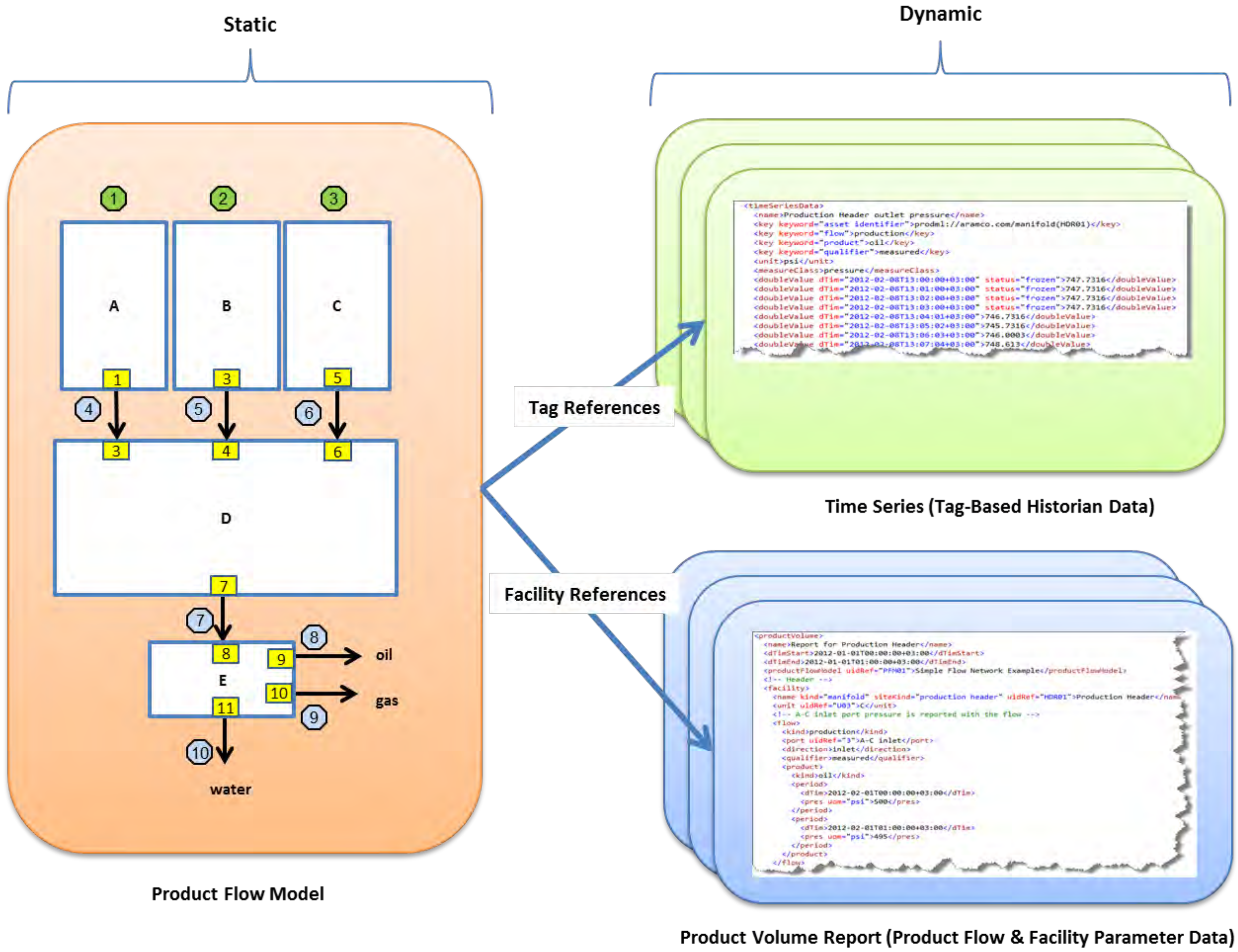


Figure 2–1. Product Flow Model, Product Volume Report, and Time Series.

2.1 Product Flow Model Data Object

The Product Flow Model data object can be used to define a directed graph of flow connections. 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 utilizes 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.

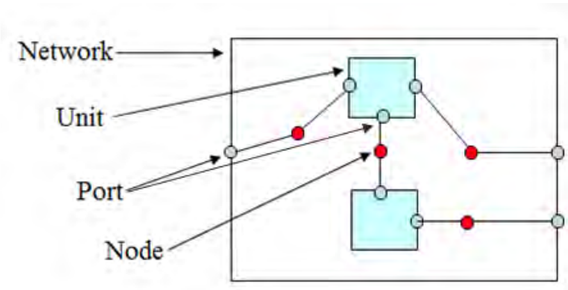


Figure 2–2. Product Flow Model data object.

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. 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 used the same standard format.

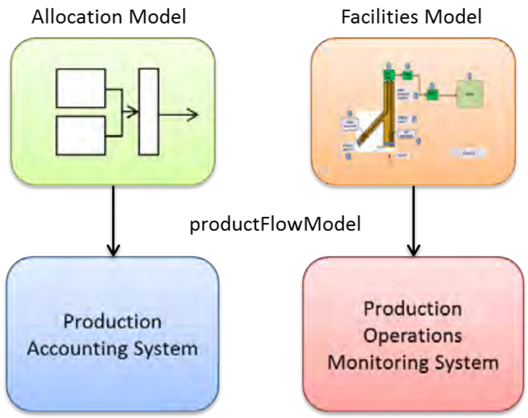


Figure 2–3. Various network models.

2.2 Product Volume Data Object

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 utilizes 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, CO_2, 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. The list of available values for parameters is given in the *FacilityParameter* enumeration in the *enumValuesProdml.xml* file.

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>
```

2.3 Time Series Data object

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 utilizes 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>
```

The available keyword values are given by the TimeSeriesKeyword enumeration values from the enumValuesProdml.xml file as well as the following table.

Keyword	KindQualifier
`asset identifier`	A PRODML 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 *PRODML Identifier Specification* for more information on PRODML 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 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*.

Keyword	KindQualifier
`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*.

2.4 Unique Identifiers

IMPORTANT: To identify data objects and resources, PRODML v2.0 uses UUIDs and URIs. For information, see the *Energistics Identifier Specification*, which is included in the PRODML download. The key concepts about relationships between objects explained in the examples in this section still apply.

Most entities, such as unit in the Product Flow Model and facility in the Product Flow Model and Product Volume Report, have specific attributes that are used for unique identification within a system. The unit's uid attribute is a unique identifier of the unit. A facility's unique identifier is conveyed in the uidRef attribute, as shown in the following snippet from a Product Flow Model. Refer to the PRODML Identifier Specification for more information on PRODML identifiers.

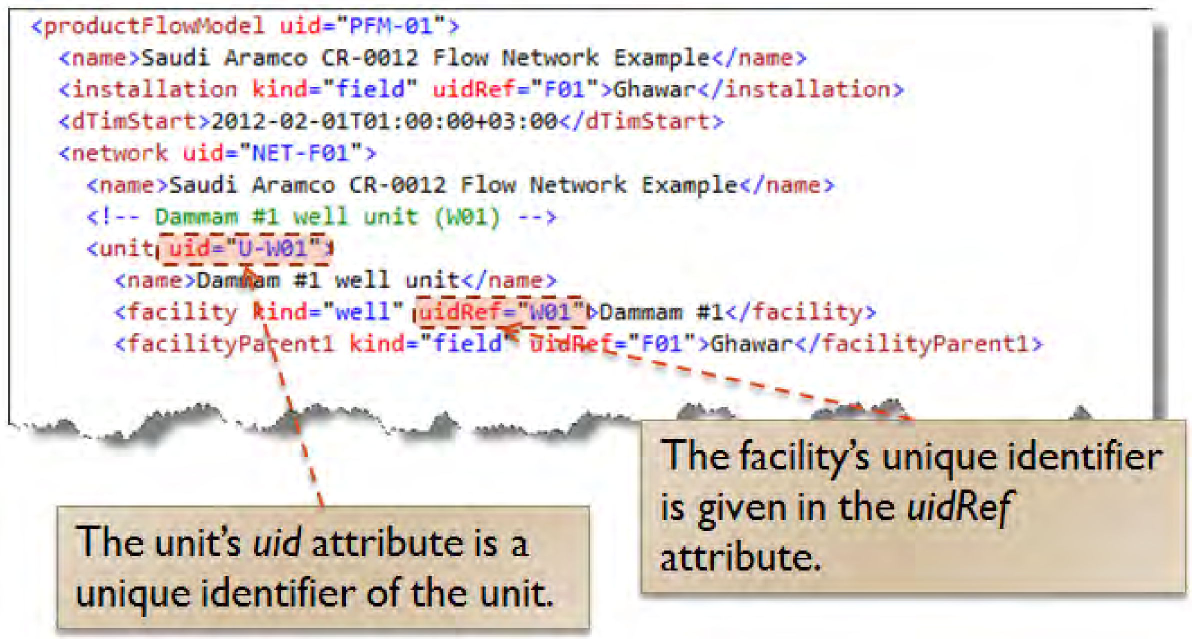


Figure 2–4. Unit & facility identifiers

There is an implicit relationship between the Product Flow Model and the Product Volume Report. In a Product Volume Report, several elements refer back to the Product Flow Model:

- The *productFlowModel* element is a pointer to the to the Product Flow Model for all Product Flow Units referenced within the Product Volume Report, as shown in Figure 2–5.
- The *unit* refers to the Product Flow Unit representing the facility, as shown in Figure 2–6.
- The *name* child element of the *facility* element refers to the associated Facility. In the Product Flow Model, the *facility* child element of the *unit* element is used to identify the facility for which this Product Flow Unit describes fluid flow connection behavior. This relationship is illustrated in Figure 2–7.
- The *port* element refers to the Product Flow Port to which the flow is assigned. This is the port on the Product Flow Unit that represents the facility, as shown in Figure 2–8.

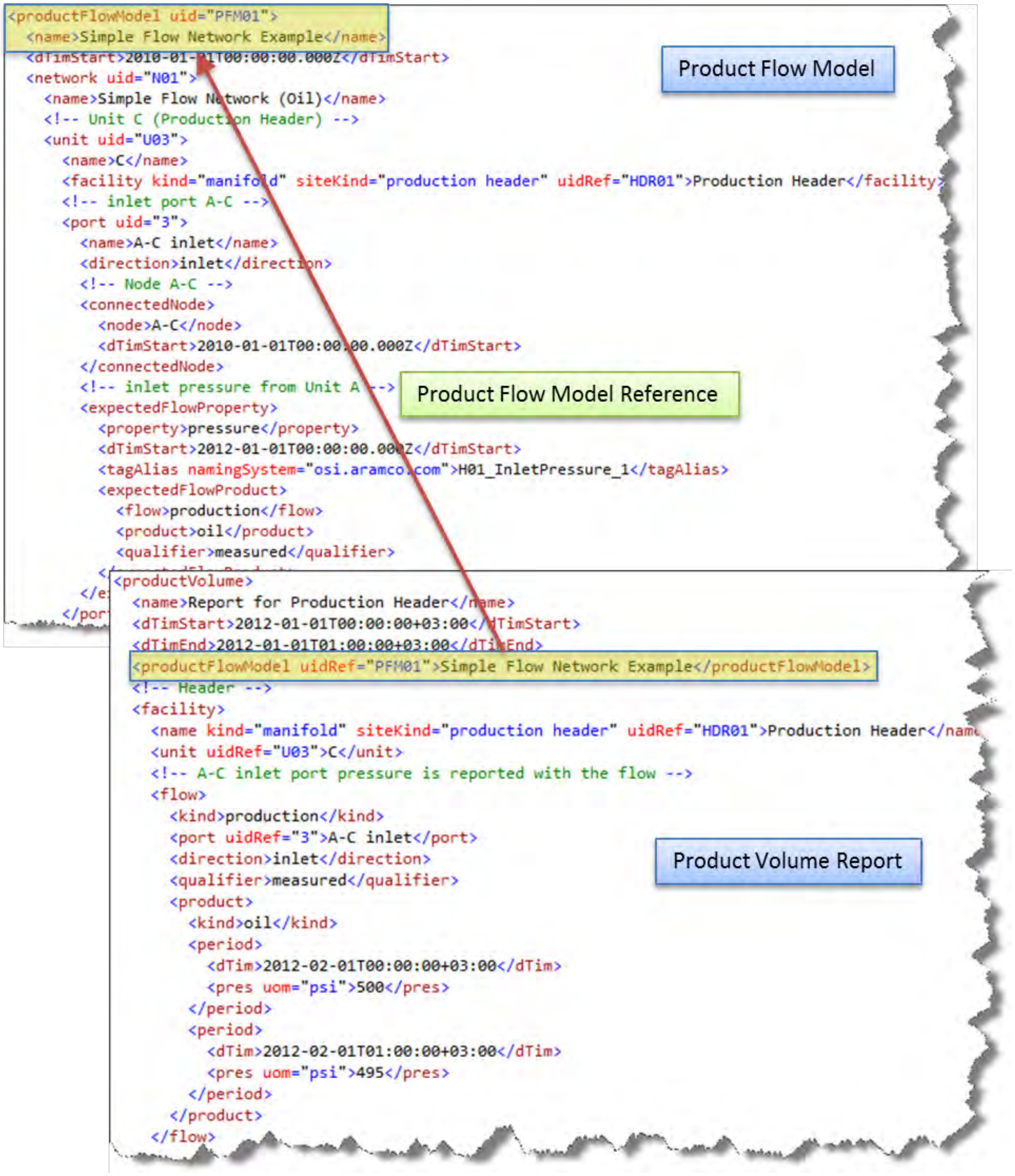


Figure 2–5. Product Flow Model reference in a Product Volume Report.

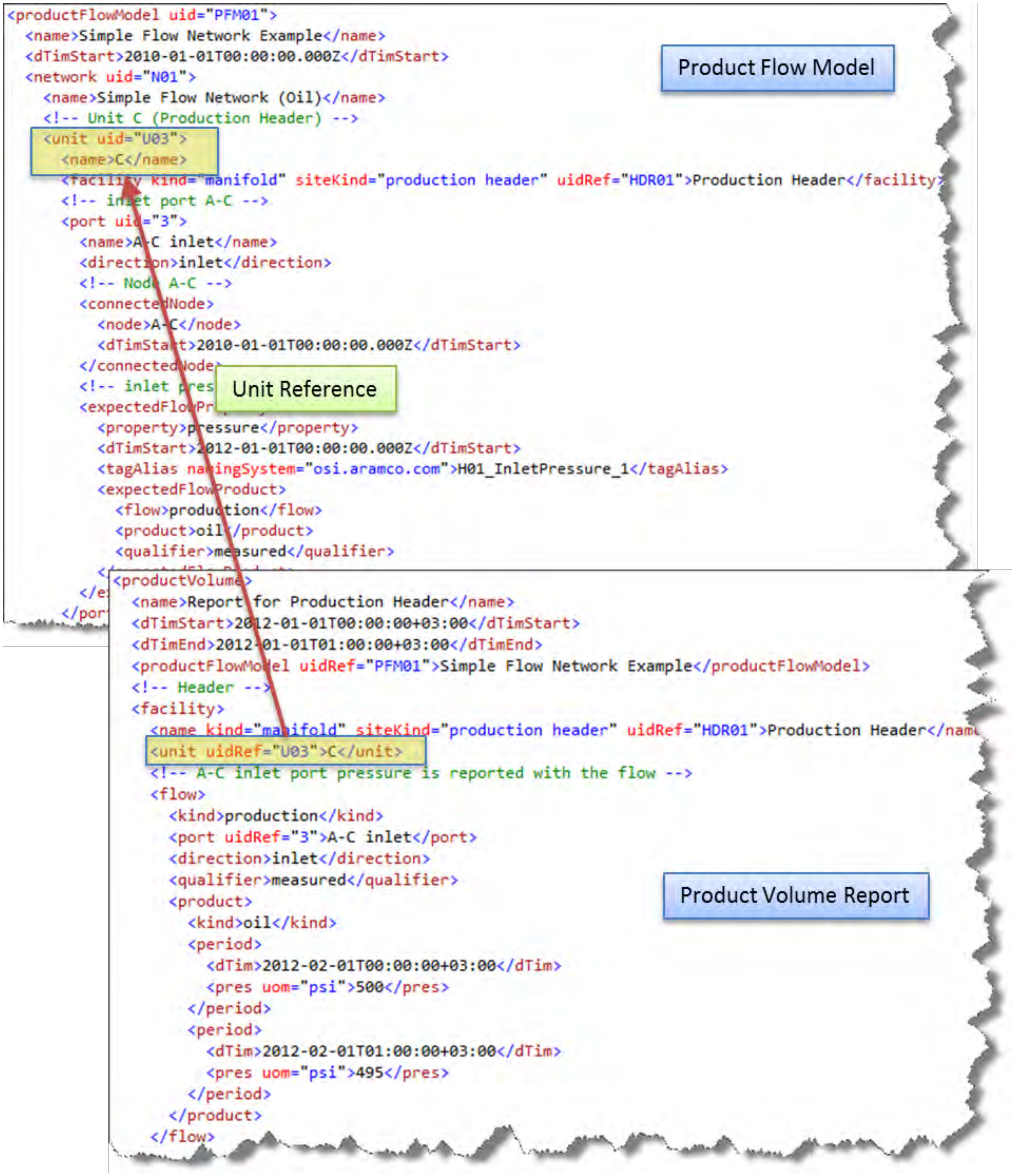


Figure 2–6. Unit Reference in a Product Volume Report.

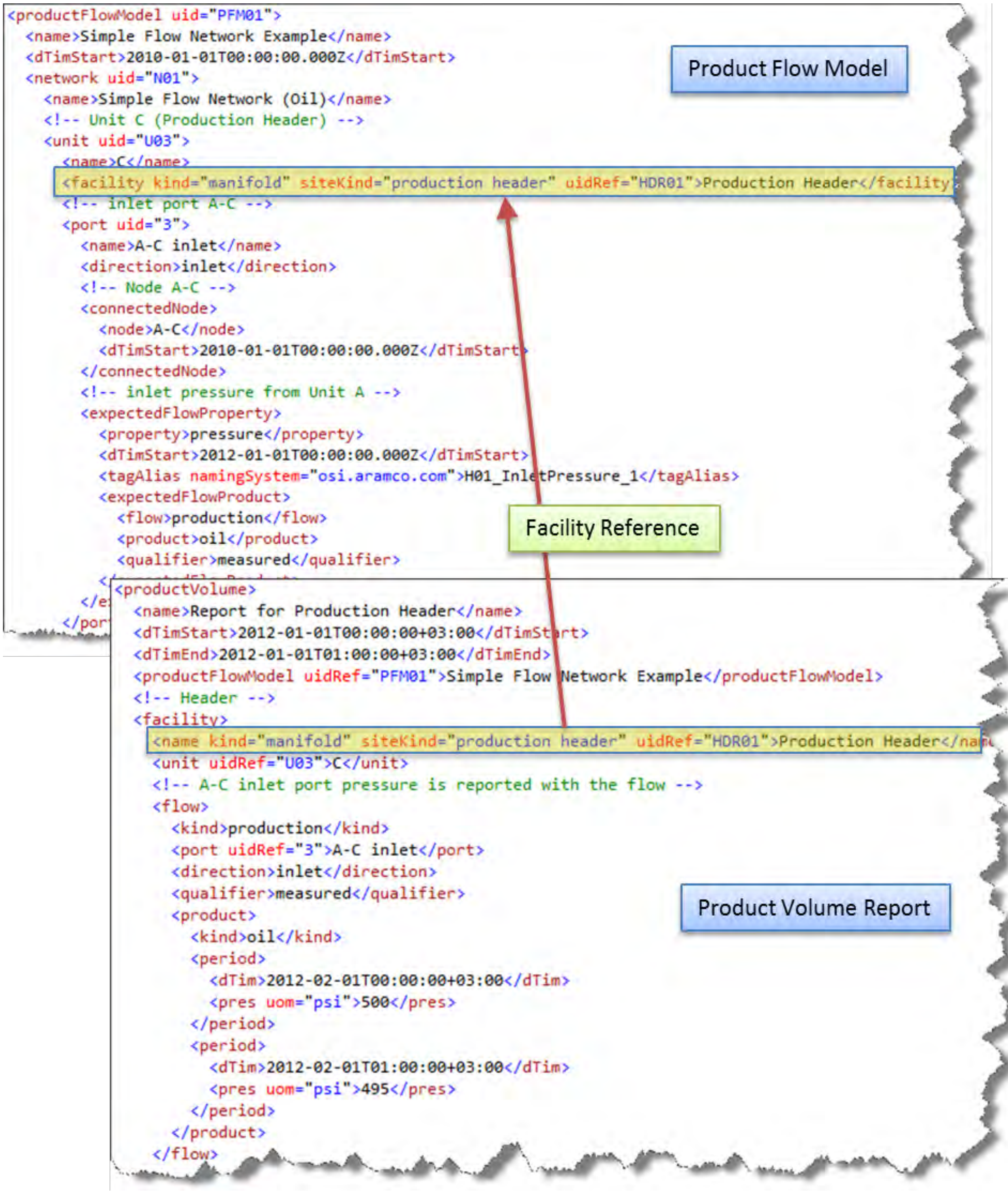


Figure 2–7. Facility Reference in a Product Volume Report.

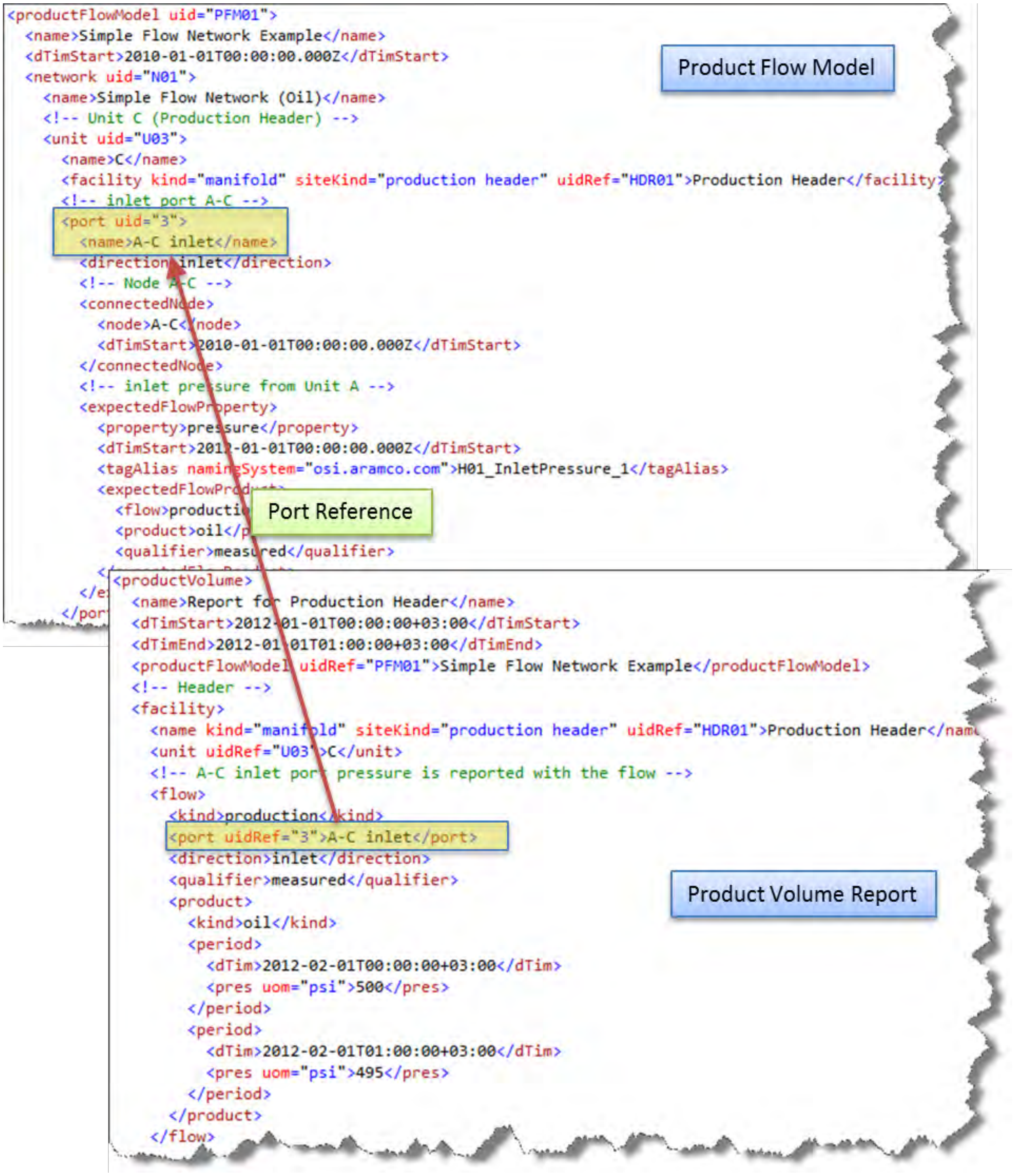


Figure 2–8. Port Reference from a Product Volume Report.

2.5 Facility Kinds

In PRODML, a facility is a general term used to represent any use of equipment to perform a function (e.g., separator, wellhead, valve, flow line, etc.) Facilities are categorized by using the *kind* and *siteKind* attributes of the *facilityIdentifierStruct* in the PRODML schemas.

- The *kind* attribute must be one of the standard values available in *ReportingFacility* enumeration found in the *enumValuesProdml.xml* file.

- The *siteKind* attribute is a custom sub-categorization of facility kind intended to allow implementers to provide a more specific or specialized description of the facility kind.

NOTE: The content of the siteKind attribute is a proprietary extension and therefore not guaranteed to provide interoperability between vendors.
The following table provides describes the requested kinds from CR_0012 and the proposed solution.

Requested Facility	**FacilityKind**	**Local Extension (siteKind)**
Venturi Flow Meter	flow meter	venturi flow meter
Multi Phase Flow Meter	flow meter	multi-phase flow meter
Micromotion flow meter	flow meter	micromotion flow meter
Test Trap	trap	test trap
Orifice Meter	flow meter	orifice meter
Gas Lift Valve Mandrel	gas lift valve mandrel	
Internal Control Valve	regulating valve	internal control value
Electric Submersible Pump	electric submersible pump	
Permanent Downhole Monitoring System	downhole monitoring system	permanent downhole monitoring system
Sub-Surface Safety Valve	regulating valve	sub-surface safety valve
Cathodic Protection Rectifier	rectifier	cathodic protection rectifier
Surface Pressure Meter	pressure meter	surface pressure meter
Remote Terminal Unit	remote terminal unit	
Test Header	manifold	test header
Test Line	pipeline	test line
Production Line	pipeline	production line
Production Header	manifold	production header
Sand Trap	trap	sand trap
Choke	choke	
Surface Pressure Transmitter	pressure meter	surface pressure transmitter
Surface Temperature Transmitter	temperature meter	surface temperature transmitter

2.6 Time Convention and its Use

Production optimization in its various forms depends upon the ability to automate the processing of information in a continuum of time. The main reason is that optimization must adapt to the inertia of the production process and operations. The basic data for this is a combination of more frequently changing activity (from sand face to marketplace changes in flow rates, pressures and temperatures, and less frequently changing activity (scheduled outages of wells and scheduled changes in export demand, especially for gas). In turn, the applications must perform the following time-based calculations:

- Accumulation based on past time (production volumes)
- Action based on current time (especially data quality and model management)
- Forecasting of tank levels and well flow rates based on past and current time
- Integration of schedules and calculations to produce useable forecasts
- Constraint of optimization based on forecasts
- Advice based on time-based statistics and forecasted alarms

PRODML supports both requests for data values at a single timestamp and requests multiple values between two timestamps, that is, over a range of time.

If a single timestamp is specified in a query, the response should contain the data value recorded either at or before the passed timestamp. PRODML requires that the returned data value is exactly equal to the data value at or before the passed timestamp without any interpolation. If the timestamp is current time at the time of the request, the response will be considered current data: the newest available values for all measurement points.

The ability exists to query a time interval (i.e. between a start-time and an end-time). The PRODML GDA *GetData* method provides a mechanism via the *DateTimeCriterion* parameter to allow a client to specify how to treat the definition of the interval boundaries. Options for defining the boundary values are given in the following table.

DateTimeCriterion	Description
`Inclusive`	The interval returned includes all data between and including the boundary points.
`Exclusive`	The interval returned includes all data between, but not including the boundary points.
`Extensive`	The interval returned includes all data between and including the first value outside of the boundary, if there is not a value at the boundary.

Each end-point condition can be specified separately.

3 WITSML Concepts

WITSML™ is an industry initiative to provide open, non-proprietary, standard interfaces for technology and software that monitor and manage wells, completions, and workovers.

IMPORTANT: Like PRODML, WITSML v2.0 has been released and uses the identifier conventions (UUIDs and URIs) as defined in the *Energistics Identifier Specification* (included in the PRODML download). For more information on WITSML, see the *WITSML Technical Usage Guide*, included in the WITSML download: https://www.energistics.org/download-standards/.

3.1 WITSML Wells and Wellbores

WITSML has adopted the following definitions:

- **Borehole -** A hole excavated in the Earth as a result of drilling or boring operations. The borehole may represent the hole of an entire wellbore (when no sidetracks are present), or a sidetrack extension. A borehole extends from an originating point (the surface location for the initial borehole or kickoff point for sidetracks) to a terminating (bottomhole) point.
- **Sidetrack** (or **lateral**) - A borehole that originates in another borehole as opposed to originating at the surface.
- **Wellbore -** A unique, oriented path from the bottom of a drilled borehole to the surface of the Earth. The path must not overlap or cross itself.
- **Well -** A unique surface location from which wellbores are drilled into the Earth for the purpose of either (1) finding or producing underground resources; or (2) providing services related to the production of underground resources.
- **Kickoff point** - The point where the centerline path of a wellbore representing a sidetrack splits from the centerline path of the originating (parent) wellbore. The two centerline paths coincide from this point to the surface.

3.2 Measured Depths

Measured depths represent the distance from a datum to a point along the centerline path of a wellbore. Each measured depth value must have a unit of measure and be related to one datum and one wellbore. The combination of wellbore, unit and datum *implicitly* represents one unique coordinate reference system (CRS). Any two values which represent the same CRS can be directly compared. Everything within a well is directly or indirectly located using measured depth values.

4 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 4–1 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.

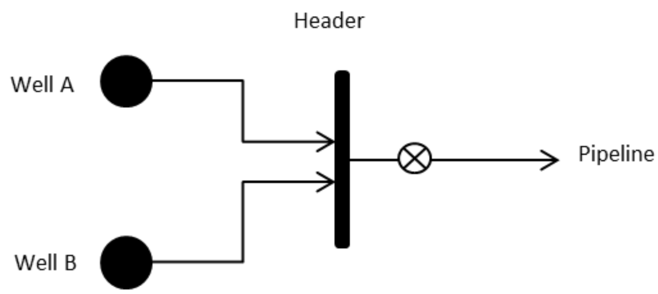


Figure 4–1. Simple oil production network.

4.1 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.**

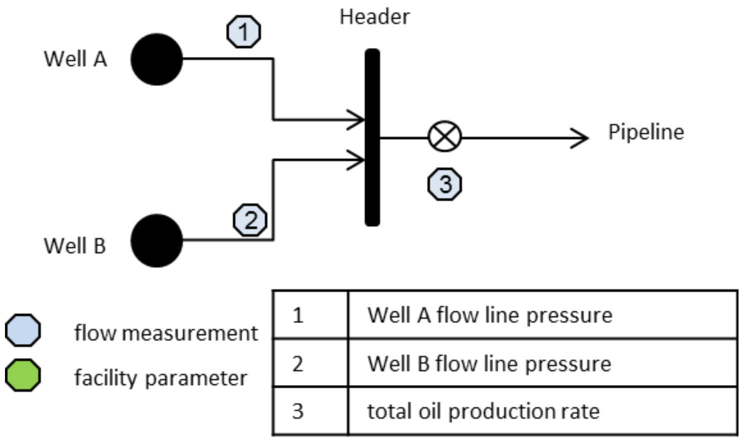
Construction of a PRODML flow network can be summarized in the following steps, which are further explained below:

1. Draw the real world diagram indicating measurement points.
2. Draw boundaries to include facility parameters and exclude flow measurements.

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, as shown in the following diagram.

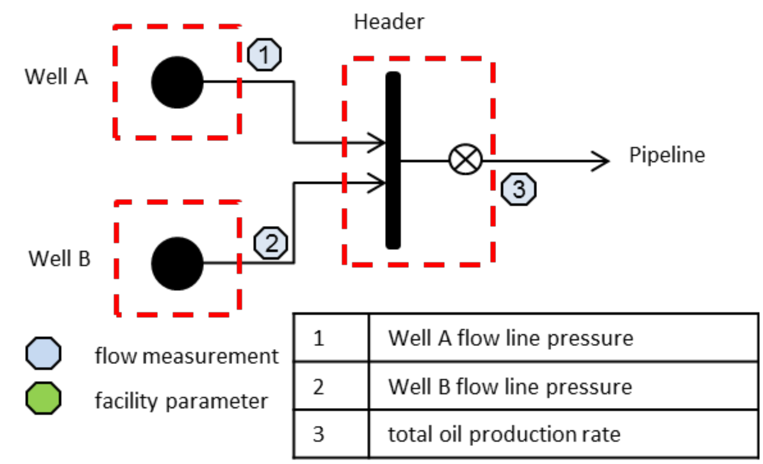


1	Well A flow line pressure
2	Well B flow line pressure
3	total oil production rate

Figure 4–2. Measurement points for a simple flow network.

Step 2: Draw Boundaries

The next step is to draw boundaries around items with no flow measurements, as shown in the following diagram.



1	Well A flow line pressure
2	Well B flow line pressure
3	total oil production rate

Figure 4–3. 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, as shown in Figure 4–4.

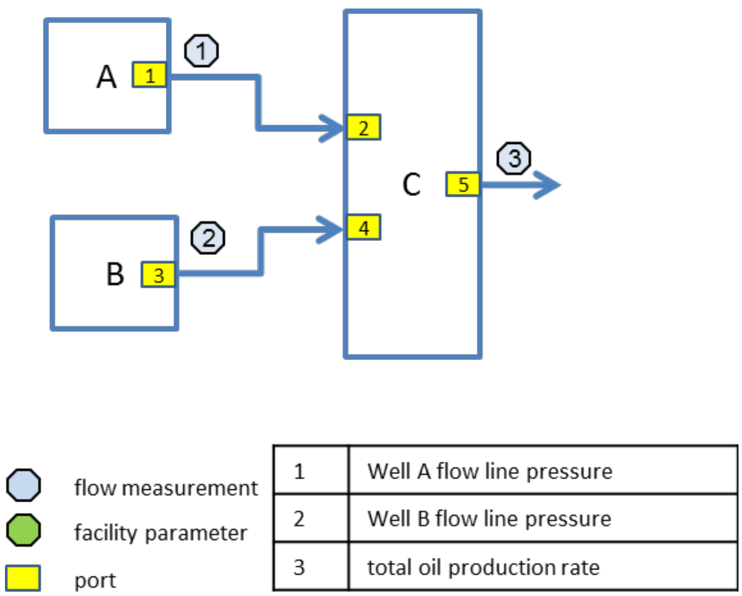


Figure 4–4. Simple network units and nodes.

The following tables summarize 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 4–5 illustrates the definition of port 5 in XML.

```xml
<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 4–5. Port definition example in XML.

4.2 Associated Product Volume Report

A Product Volume Report can now be associated with any given unit in the Product Flow Model. The XML snippet in Figure 4–6 shows how to reference the associated unit and port from a Product Volume Report.

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 4–6. Product Volume Report referencing Product Flow Model unit & port.

5 Retrieving Static Data

Some parameters, such as the measured depth of a piece of equipment, can be treated a static facility parameters. Static facility parameters are provided in the context of a Product Volume Report. A special category of Product Volume Report, where the kind element is specified as *static facility parameters*, is used for conveying static data.

5.1 Measured Depth Example

The measured depth of an electric submersible pump is treated a static facility parameters. The following diagram gives an example of such a scenario.

```xml
<unit uid="ESP01">
  <name>Dammam #1 ESP</name>
  <facility kind="pump" kindQualifier="electrical submersible pump" uidRef="ESP01">Dammam #1 ESP-01</facility>
  <facilityParent1 kind="wellbore" uidRef="B01">Dammam #1 B01</facilityParent1>
  <facilityParent2 kind="well" uidRef="W01">Dammam #1</facilityParent2>
  <contextFacility kind="field" uidRef="F01">Ghawar</contextFacility>
  <!-- measured depth -->
  <expectedProperty>
    <property>measured depth</property>
  </expectedProperty>
  <!-- motor current -->
  <expectedProperty>
```

Product Flow Model

```xml
<productVolume>
  <name>Static Facility Data</name>
  <kind>static facility parameters</kind>
  <facility>
    <name kind="pump" siteKind="electric submersible pump" uidRef="ESP01">Dammam #1 ESP01</nam
    <facilityParent1 kind="wellbore" uidRef="B01">Dammam #1 B01</facilityParent1>
    <facilityParent2 kind="well" uidRef="W01">Dammam #1</facilityParent2>
    <unit uidRef="U-ESP01">Dammam #1 ESP</unit>
    <contextFacility kind="field" uidRef="F01">Ghawar</contextFacility>
    <!-- measured depth -->
    <parameterSet>
      <name>measured depth</name>
      <measureClass>length</measureClass>
      <coordianteReferenceSystem>well datum(KB)</coordianteReferenceSystem>
      <parameter>
        <measureValue uom="m">2000</measureValue>
      </parameter>
    </parameterSet>
  </facility>
```

Product Volume

Figure 5–1. Measured depth example of static facility data.

If the Product Flow Model is being used, then *measured depth* is listed as an expected property of the facility. In this case, the facility is an *electric submersible pump*. However, the Product Flow Model is not required to be used since the measured depth of the electric submersible pump is always given in the Product Volume Report for the facility in the *parameterSet* structure as shown.

Of significant note is the *kind* element of Product Volume Report, which is specified as *static facility parameters*.

5.2 Annulus Size Example

This example shows how to report annulus size for a wellbore in inches (e.g., 9x13 or 9x7). The annulus is the space between the innermost casing and outside of the production tubing that was installed inside the casing.

```
<productVolume>
  <name>Static Facility Data</name>
  <kind>static facility parameters</kind>
  <facility>
    <name kind="wellbore" uidRef="B01">Dammam #1 wellbore</name>
    <facilityParent1 kind="well" uidRef="W01">Dammam #1</facilityParent1>
    <!-- inner diameter -->
    <parameterSet>
      <name>annulus inner diameter</name>
      <measureClass>length</measureClass>
      <parameter>
        <measureValue uom="in">9</measureValue>
      </parameter>
    </parameterSet>
    <!-- outer diameter -->
    <parameterSet>
      <name>annulus outer diameter</name>
      <measureClass>length</measureClass>
      <parameter>
        <measureValue uom="in">13</measureValue>
      </parameter>
    </parameterSet>
  </facility>
</productVolume>
```

Figure 5–2. Annulus size example of static facility data.

6 Product Flow Model Guidance

6.1 Tag Alias

The *tagAlias* element in the Product Flow Model allows for an alternative name for the sensor that will measure a property. . As shown in Figure 6–1, a tag alias *ESP01_MotorCurrent* is specified for the *motor current* parameter of an electric submersible pump.

Product Flow Model

```
<network uid="NET-01">
  <name>Example Network</name>
  <unit uid="ESP01">
    <name>Dammam #1 ESP</name>
    <facility kind="pump" siteKind="electric submersible pump" uidRef="ESP01">Damma
    <expectedProperty>
      <property>motor current</property>
      <qualifier>measured</qualifier>
      <tagAlias namingSystem="osi.aramco.com">ESP01_MotorCurrent</tagAlias>
      <deadband uom="mA">10</deadband>
      <maximumFrequency uom="s">3600</maximumFrequency>
    </expectedProperty>
```

Figure 6–1. Tag Alias usage in Product Flow Model.

6.2 Deadband

Deadband is the difference between two consecutive readings must exceed deadband value to be accepted. The *deadband* element in the Product Flow Model allows this information to be specified. As shown in Figure 6–2, a deadband value 10 milliamps is specified for the *motor current* parameter of an electric submersible pump.

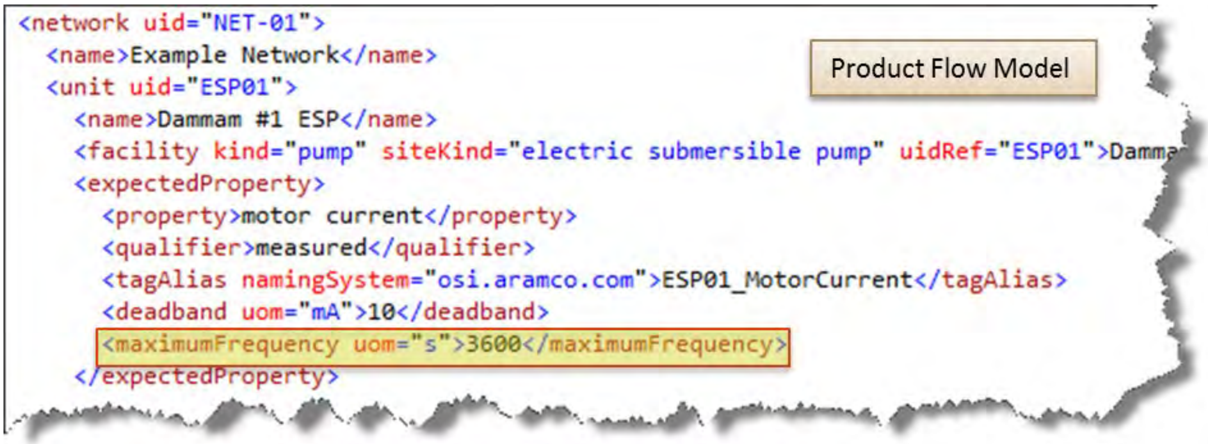
Product Flow Model

```
<network uid="NET-01">
  <name>Example Network</name>
  <unit uid="ESP01">
    <name>Dammam #1 ESP</name>
    <facility kind="pump" siteKind="electric submersible pump" uidRef="ESP01">Damma
    <expectedProperty>
      <property>motor current</property>
      <qualifier>measured</qualifier>
      <tagAlias namingSystem="osi.aramco.com">ESP01_MotorCurrent</tagAlias>
      <deadband uom="mA">10</deadband>
      <maximumFrequency uom="s">3600</maximumFrequency>
    </expectedProperty>
```

Figure 6–2. Deadband usage in Product Flow Model.

6.3 Maximum Frequency

Maximum Frequency is the maximum time difference from the last sent event before the next event is sent. The *maximumFrequency* element in the Product Flow Model allows this information to be specified. As shown in Figure 6–2, a maximum frequency value 3600 seconds is specified for the *motor current* parameter of an electric submersible pump.



```
<network uid="NET-01">
  <name>Example Network</name>
  <unit uid="ESP01">
    <name>Dammam #1 ESP</name>
    <facility kind="pump" siteKind="electric submersible pump" uidRef="ESP01">Damma
    <expectedProperty>
      <property>motor current</property>
      <qualifier>measured</qualifier>
      <tagAlias namingSystem="osi.aramco.com">ESP01_MotorCurrent</tagAlias>
      <deadband uom="mA">10</deadband>
      <maximumFrequency uom="s">3600</maximumFrequency>
    </expectedProperty>
```


Figure 6–3. Maximum Frequency usage in Product Flow Model.

7 Product Volume Guidance

7.1 Gas-Oil and Gas-Liquid Ratios

Figure 7–1 illustrates how to include gas oil ratio (*gor* element) and gas liquid ratio (*gasLiquidRatio* element) values in a Product Volume Report. In this scenario, a multi-phase flow meter is used to report the ratios within the measured flow of the combined oil, gas, and water product stream.

```
<productVolume>
  <name>Multi-phase flow meter #1 production</name>
  <dTimStart>2012-01-01T00:00:00Z</dTimStart>
  <facility>
    <name kind="flow meter" siteKind="multi-phase flow meter" uidRef="MPFM01">Multi-phase flow meter #1</name>
    <flow>
      <kind>production</kind>
      <qualifier>measured</qualifier>
      <product>
        <kind>oil and gas</kind>
        <period>
          <gor uom="%">75</gor>
          <gasLiquidRatio uom="%">70</gasLiquidRatio>
          <volumeValue>
            <volume uom="bbl">100.0</volume>
          </volumeValue>
        </period>
      </product>
    </flow>
  </facility>
</productVolume>
```

Figure 7–1. Gas-oil & gas-liquid ratio example.

7.2 Water Cut

Figure 7–2 illustrates how to declare water cut as an expected flow property in the Product Flow Model and report the values in a Product Volume Report. In this scenario, the *ReportingProperty* value of *volume concentration* is combined with the expected flow product of *water* to indicate the water cut reported from a multi-phase flow meter.

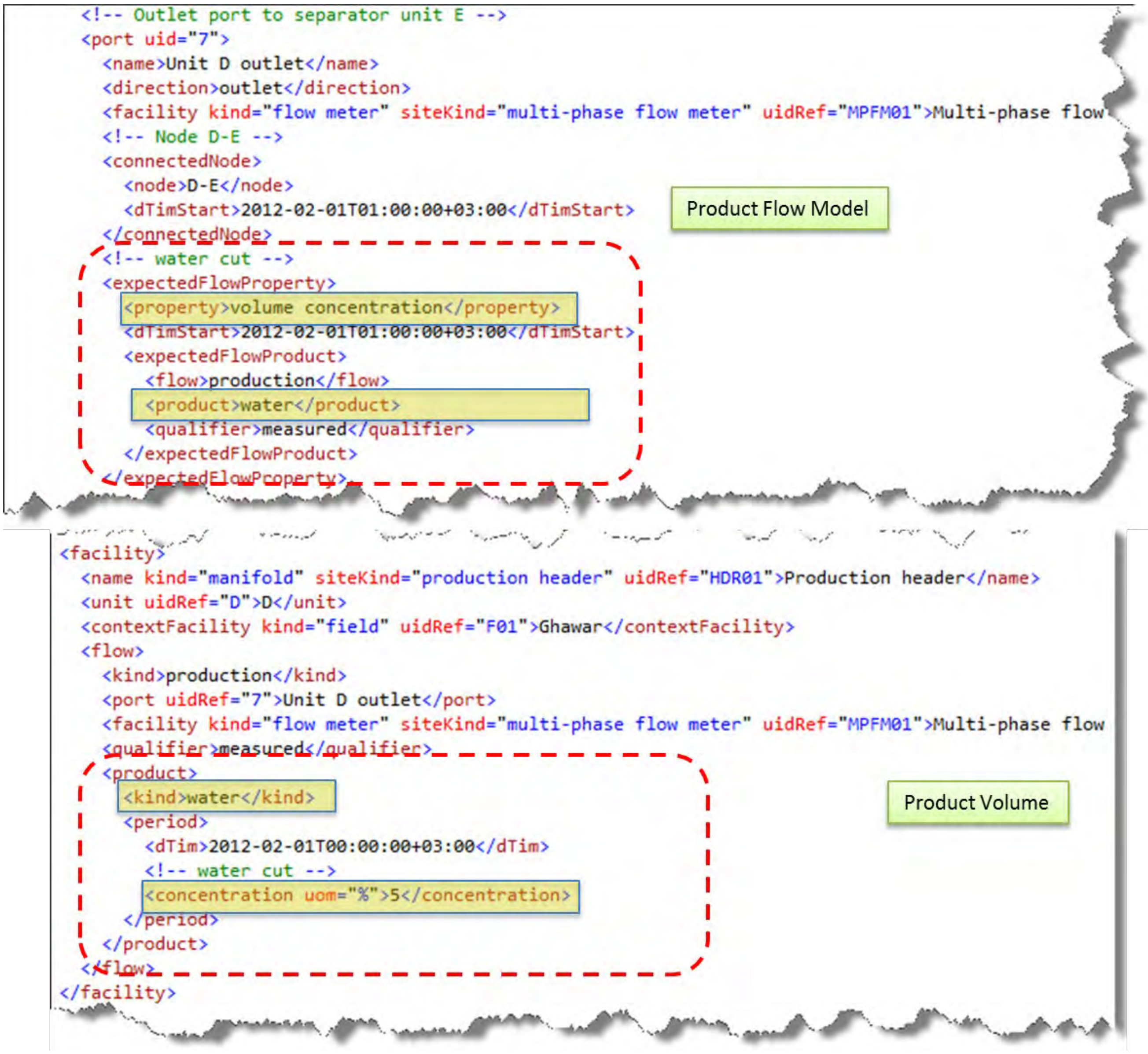


Figure 7–2. Water Cut Example.

7.3 Reporting at Standard Conditions (Temperature & Pressure)

Figure 7–3 illustrates how to specify standard conditions in the Product Flow Model and report the standard conditions in a Product Volume Report. The standardTempPres element in the Product Volume Report defines the default standard temperature and pressure to which all volumes, densities and flow rates in the report have been corrected. The default may be locally overridden for an individual value. If not specified, then the conditions must be presumed to be ambient conditions (i.e., uncorrected) unless otherwise specified at a local level.

Product Flow Model

```xml
<unit uid="U06">
  <name>F</name>
  <facility kind="flow meter" kindQualifier="multi-phase flow meter" uidRef=
  <port uid="10">
    <name>flow meter inlet port</name>
    <direction>inlet</direction>
    <!-- Node E-F -->
    <connectedNode>
      <node>E-F</node>
    </connectedNode>
    <!-- oil flow rate at standard conditions -->
    <expectedFlowProperty>
      <property>standard flow rate</property>
      <dTimStart>2012-02-01T01:00:00+03:00</dTimStart>
      <expectedFlowProduct>
        <flow>production</flow>
        <product>oil</product>
      </expectedFlowProduct>
    </expectedFlowProperty>
  </port>
```

Product Volume

```xml
<productVolume>
  <name>Production for multi-phase flow meter MPFM01</name>
  <standardTempPres>60 degF 1 atm</standardTempPres>
  <facility>
    <name kind="flow meter" kindQualifier="multi-phase flow meter" uidRef="MPFM01">Multi-phase flow met
    <unit uidRef="E">E</unit>
    <flow>
      <kind>production</kind>
      <port uidRef="10">flow meter inlet port</port>
      <qualifier>measured</qualifier>
      <!-- oil flow rate at standard conditions -->
      <product>
        <kind>oil</kind>
        <period>
          <dTim>2012-02-01T00:00:00+03:00</dTim>
          <flowRateValue>
            <flowRate uom="bbl/d">100</flowRate>
          </flowRateValue>
        </period>
      </product>
    </flow>
  </facility>
</productVolume>
```

Figure 7–3. Reporting at standard conditions.

8 Time Series Guidance

The PRODML Time Series data object is intended for use in support of smart fields or high-frequency historian type interactions, not reporting.

8.1 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 a PRODML formatted URI identifier of the asset (facility) related to the value. The identifier may define a hierarchy of assets. Refer to the *PRODML Identifier Specification* for more information on PRODML identifiers.

The example in Figure 8–1 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).

```xml
<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 8–1. Time series asset identifier example.

8.2 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 8–2 shows several readings with a status of frozen, which indicates the sensor has been reading the same value for a specific period of time.

```xml
<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 8–2. Value status example.

9 Using Child Facility Identifiers

The Product Flow Model and the Product Volume Report both provide for the ability to convey information about embedded facilities using the *prodmlRelativeIdentifier* construct (which is a Uniform Resource Identifier (URI)); This construct is defined in the *Energistics Identifier Specification* (which is included in the zip file when you download PRODML). The format follows the general pattern of "type(id)/type(id)" where "(id)" is optional (see the Identifier Spec for details). The advantage of this approach is to simplify the resulting Product Flow Model.

EXAMPLE: Consider a wellbore that contains an electric submersible pump (ESP) as shown in Figure 9–1.

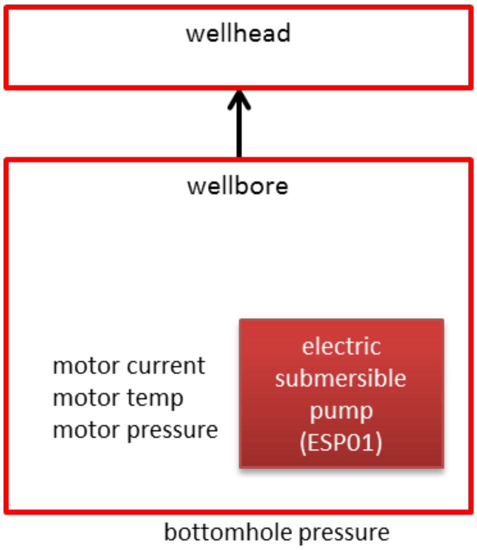


Figure 9–1. Wellbore with embedded equipment.

Let's also assume that we want to measure the ESP motor current, motor temperature, and motor pressure as well as the bottomhole pressure for the wellbore. One could construct a Product Flow Model identifying each entity (wellbore, ESP, wellhead, etc.) at a very granular level. However, this approach can quickly lead to a complex flow model with many levels of embedded models.

Alternatively, we can include the internal facility information as part of the wellbore to simplify our model, as shown in Figure 9–2.

```xml
<unit uid="U-B01">
  <name>Dammam #1 wellbore unit</name>
  <facility kind="wellbore" uidRef="B01">Dammam #1 wellbore</facility>
  <facilityParent1 kind="well" uidRef="W01">Dammam #1</facilityParent1>
  <contextFacility kind="field" uidRef="F01">Ghawar</contextFacility>
  <!-- ESP motor current -->
  <expectedProperty>
    <property>motor current</property>
    <childFacilityIdentifier>electric submersible pump(ESP01)</childFacilityIdentifier>
  </expectedProperty>
  <!-- bottomhole pressure -->
  <expectedProperty>
    <property>pressure</property>
    <childFacilityIdentifier>bottomhole</childFacilityIdentifier>
  </expectedProperty>
```

```xml
<productVolume>
  <name>Report for Dammam #1 wellbore and internal facility parameters</name>
  <facility>
    <name kind="wellbore" uidRef="B01">Dammam #1 wellbore</name>
    <facilityParent1 kind="well" uidRef="W01">Dammam #1</facilityParent1>
    <!-- ESP motor current -->
    <parameterSet>
      <name>motor current</name>
      <childFacilityIdentifier>electric submersible pump(ESP01)</childFacilityIdentifier>
      <parameter>
        <dTim>2012-02-01T01:00:00+03:00</dTim>
        <measureValue uom="A">60</measureValue>
      </parameter>
    </parameterSet>
    <!-- bottomhole pressure -->
    <parameterSet>
      <name>pressure</name>
      <childFacilityIdentifier>bottomhole</childFacilityIdentifier>
      <parameter>
        <dTim>2012-02-01T01:00:00+03:00</dTim>
        <measureValue uom="psi">743</measureValue>
      </parameter>
```


Figure 9–2. Child facility identifiers in Product Flow Model and Product Volume Reports.

10 Multi-Lateral Well Example

A multi-lateral well is one in which there is more than one horizontal or near horizontal lateral sidetrack drilled from a single main bore and connected back to that main bore. The WITSML wellbore object is best suited for providing detailed information about a multi-lateral well, as shown in Figure 10–1.

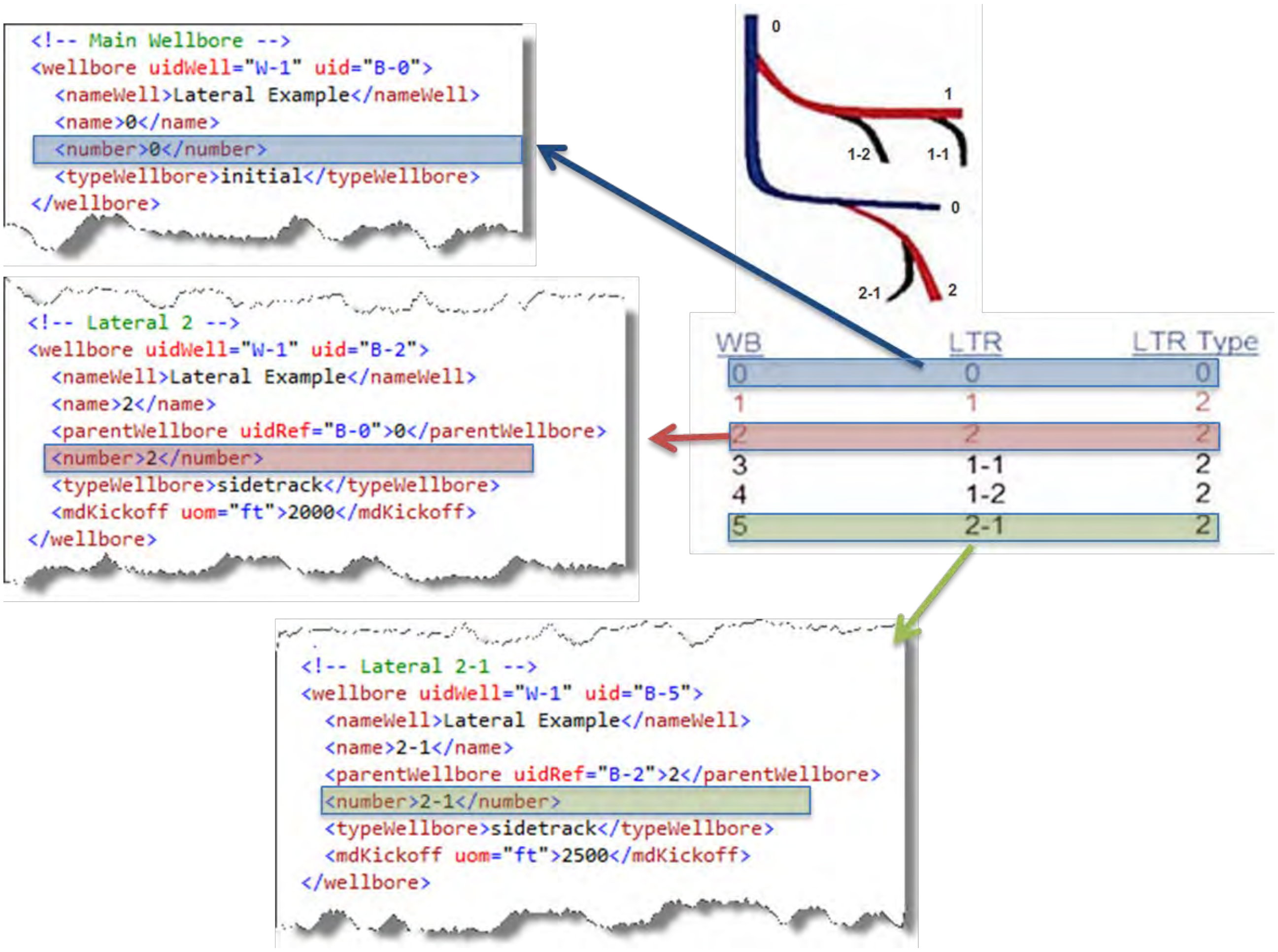


Figure 10–1. Multi-lateral well example.

Furthermore, a given wellbore may be references from the PRODML Product Flow Model by using the *uidRef* attribute of a unit's *facility* element.

```
<unit uid="B">
  <name>Lateral 2-1 unit</name>
  <facility kind="wellbore" uidRef="B-5">2-1</facility>
  <facilityParent1 kind="well" uidRef="W-1">Lateral Example</facili
  <port uid="3">
    <name>inlet port</name>
    <direction>inlet</direction>
```

11 Workflow Example (Saudi Aramco)

Figure 11–1 shows the high-level model of a production unit model of the well facility that is used throughout this example. This unit has three gas wells connected to a multi-phase flow meter via a production header. The product flows are then connected to a separator, which separates the mixed products into oil, gas, and water flows.

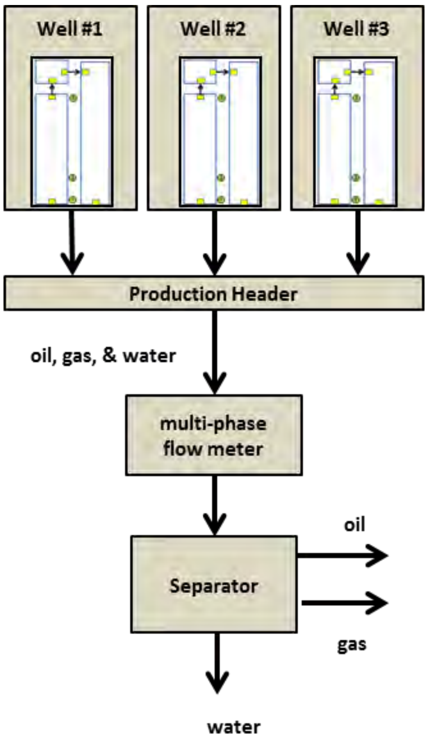


Figure 11–1. Example production unit.

Figure 11–2 shows the detailed model of one of the well facilities in the previous diagram. The well in this example has two wellbores – the original hole and a lateral. In many cases, the flow from a well is connected to a valve which directs the flow to either the test line or the production line. However, for this example, we only consider the case where the wells are connected directly to the production header.

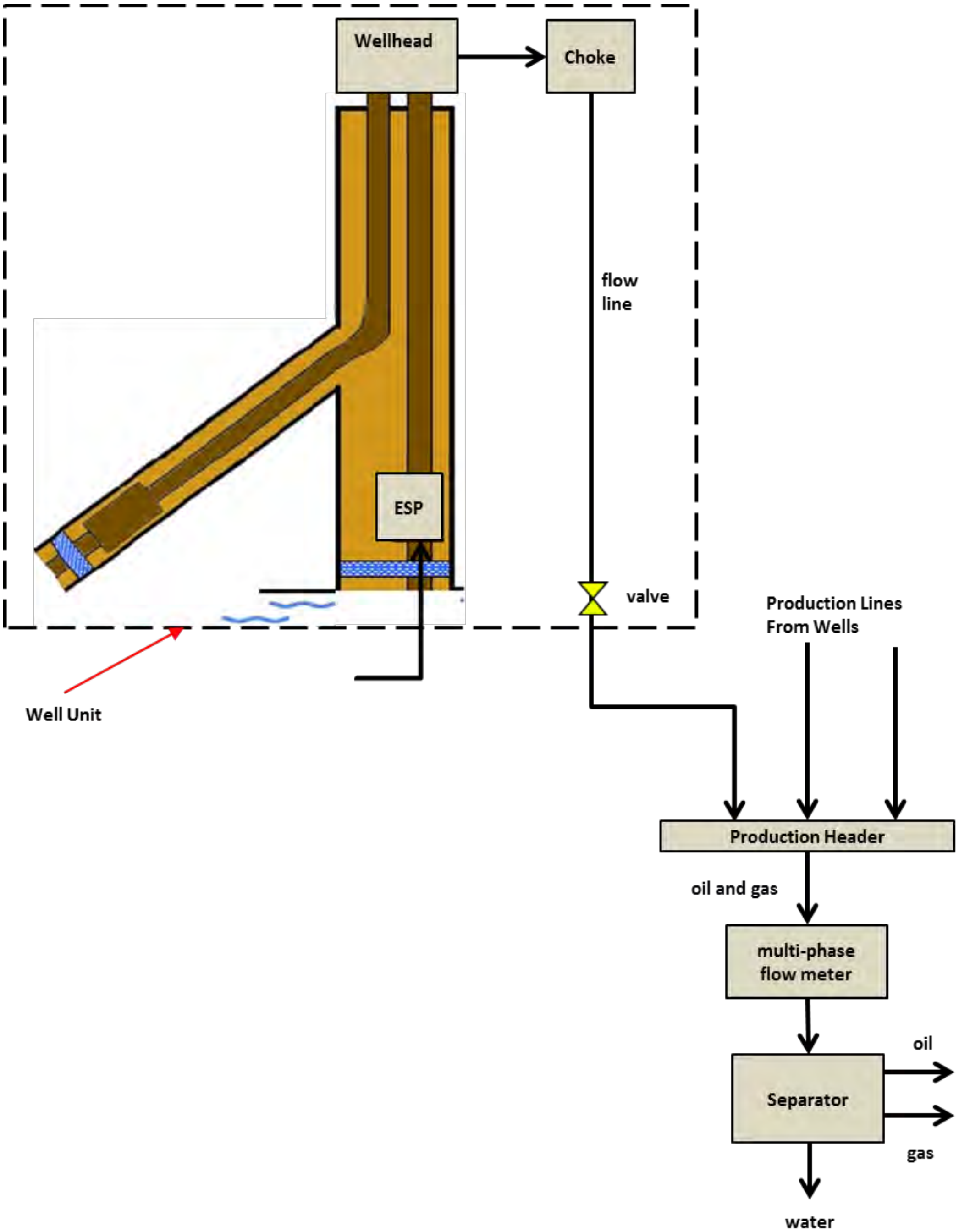


Figure 11–2. Facility model showing well detail.

11.1 High-Level System Architecture

The following diagram shows a possible system architecture where PRODML is used to integrate various systems.

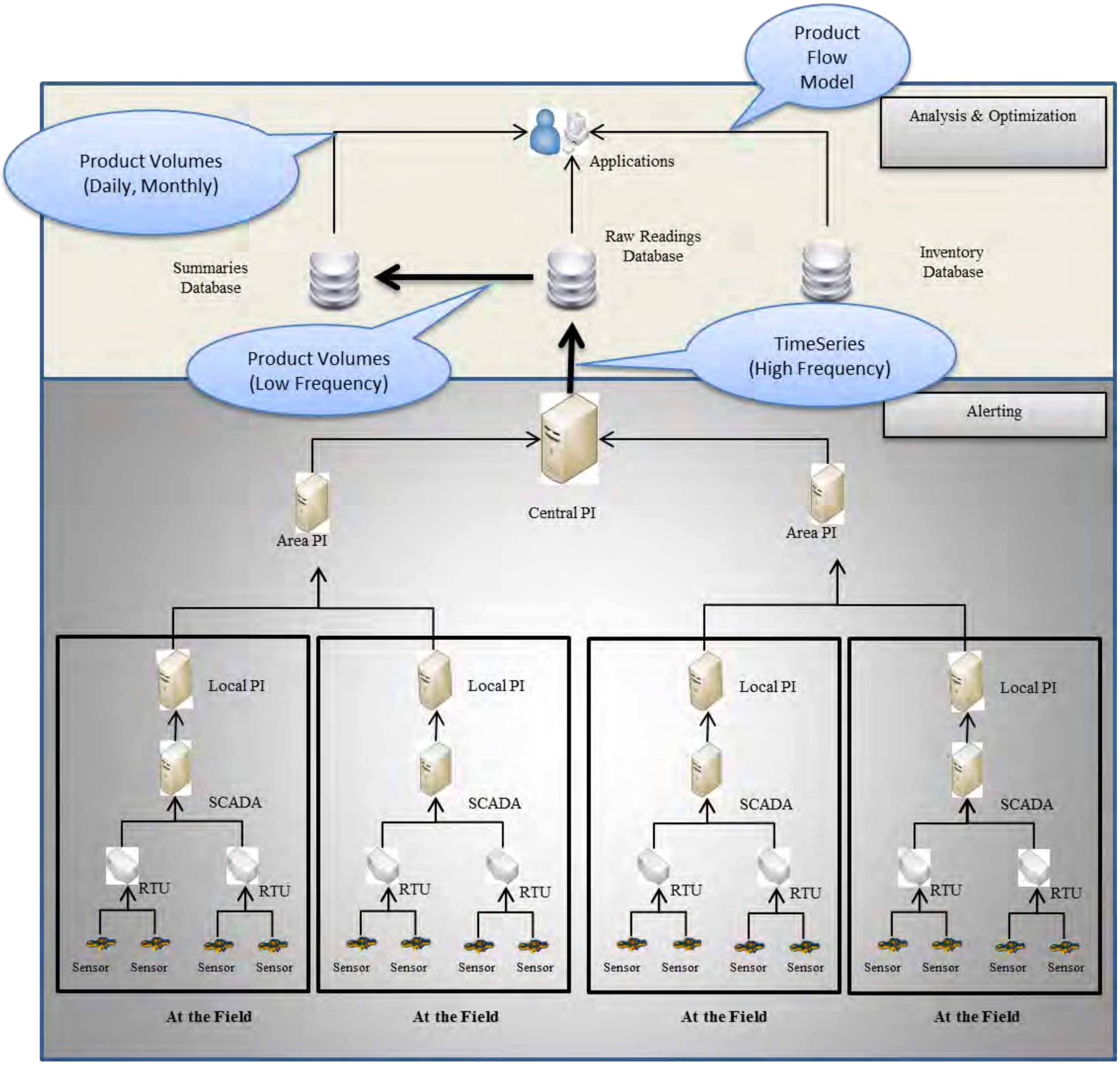
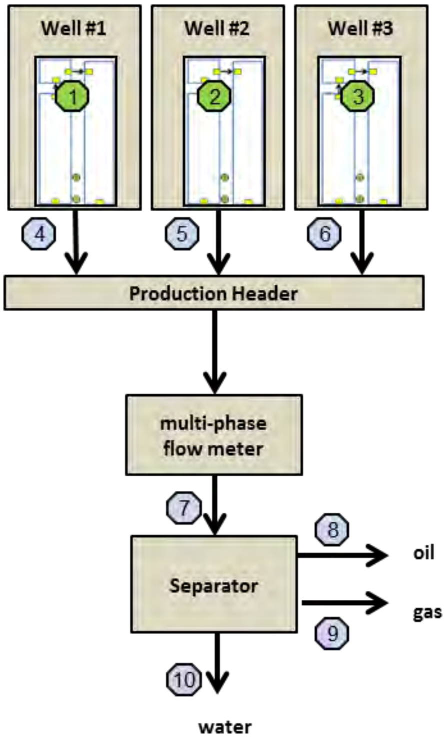


Figure 11–3. High-level system architecture example using PRODML.

11.2 Production Unit Flow Model Construction

The first step is to draw the real world diagram indicating measurement points, as shown in Figure 11–4.



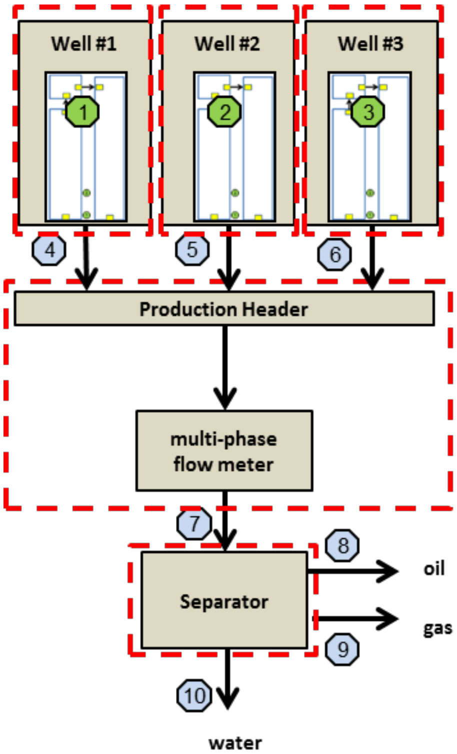
1	well #1 operating status
2	well #2 operating status
3	well #3 operating status
4	flow line pressure from well #1 (outlet)
5	flow line pressure from well #2 (outlet)
6	flow line pressure from well #3 (outlet)
7	flow line pressure gas rate oil rate water rate gas liquid ratio gas oil ratio water cut
8	total oil flow rate
9	total gas flow rate
10	total water flow rate

flow measurement

facility parameter

Figure 11–4. Production unit measurement points.

The next step is to draw boundaries around items with no flow measurements, as shown in Figure 11–5.



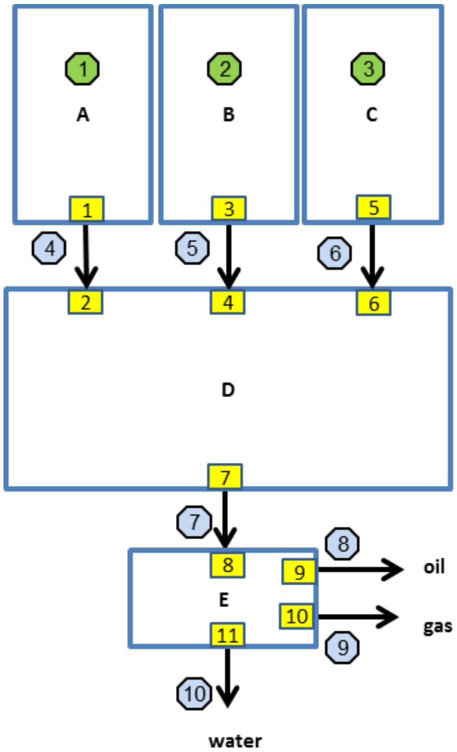
1	well #1 operating status
2	well #2 operating status
3	well #3 operating status
4	flow line pressure from well #1 (outlet)
5	flow line pressure from well #2 (outlet)
6	flow line pressure from well #3 (outlet)
7	flow line pressure gas rate oil rate water rate gas liquid ratio gas oil ratio water cut
8	total oil flow rate
9	total gas flow rate
10	total water flow rate

flow measurement

facility parameter

Figure 11–5. Production unit boundaries.

Finally, make each boundary a unit in the flow network, label each unit and port with a unique name or number, as shown in Figure 11–6



1	well #1 operating status
2	well #2 operating status
3	well #3 operating status
4	flow line pressure from well #1 (outlet)
5	flow line pressure from well #2 (outlet)
6	flow line pressure from well #3 (outlet)
7	flow line pressure gas rate oil rate water rate gas liquid ratio gas oil ratio water cut
8	total oil flow rate
9	total gas flow rate
10	total water flow rate

flow measurement
facility parameter
port

Figure 11–6. Production unit flow network.

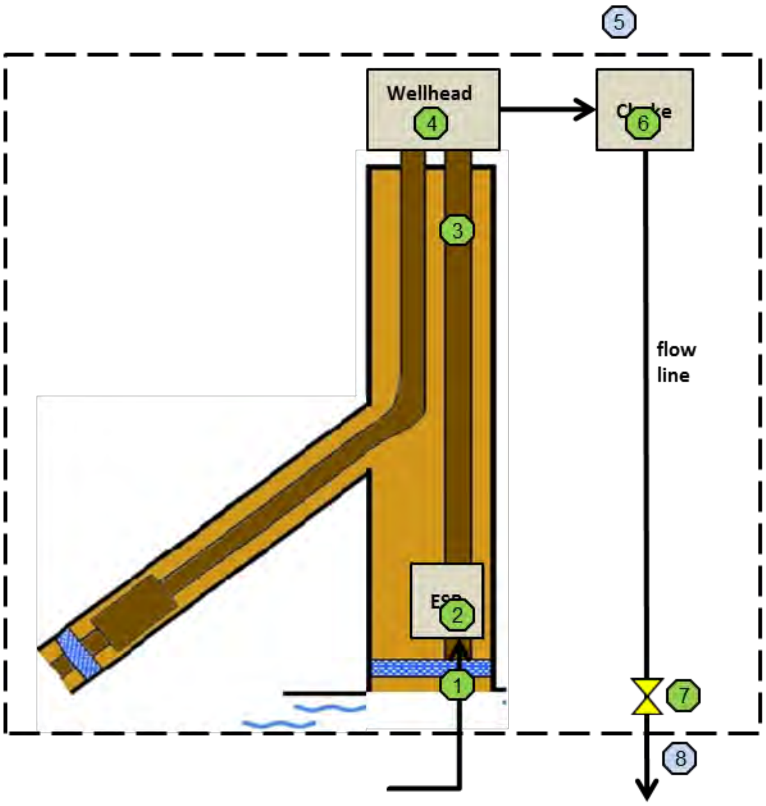
Table 4 Product Flow Model Units					
Unit	**Facility Name**	**Facility ID**	**Facility Kind**	**Site Kind**	**Internal Network**
A	Dammam #1	W01	well		NET-W01
B	Dammam #2	W02	well		NET-W02
C	Dammam #3	W03	well		NET-W03
D	Production header	HDR01	manifold	production header	
E	Separator #1	SEP01	separator		

Table 5 Product Flow Model Expected Flow Properties				
Port	**Property**	**Flow**	**Product**	**Qualifier**
1	pressure	production	oil and gas	measured
3	pressure	production	oil and gas	measured
5	pressure	production	oil and gas	measured
7	pressure	production	oil and gas	measured
7	temperature	production	oil and gas	measured
7	gas liquid ratio	production	oil and gas	measured
7	gas oil ratio	production	oil and gas	measured
7	flow rate	production	oil and gas	measured
7	flow rate	production	gas	measured
7	flow rate	production	oil	measured
7	flow rate	production	water	measured
7	volume concentration	production	water	measured
9	flow rate	production	oil	measured
10	flow rate	production	gas	measured
11	flow rate	production	water	measured

IMPORTANT: The product value of *oil and gas* refers to a combination of oil, water, and gas and is used to indicate all products in a flow.

11.3 Construction for a Well

The first step is to draw the real world diagram indicating measurement points, as shown in Figure 11–7.



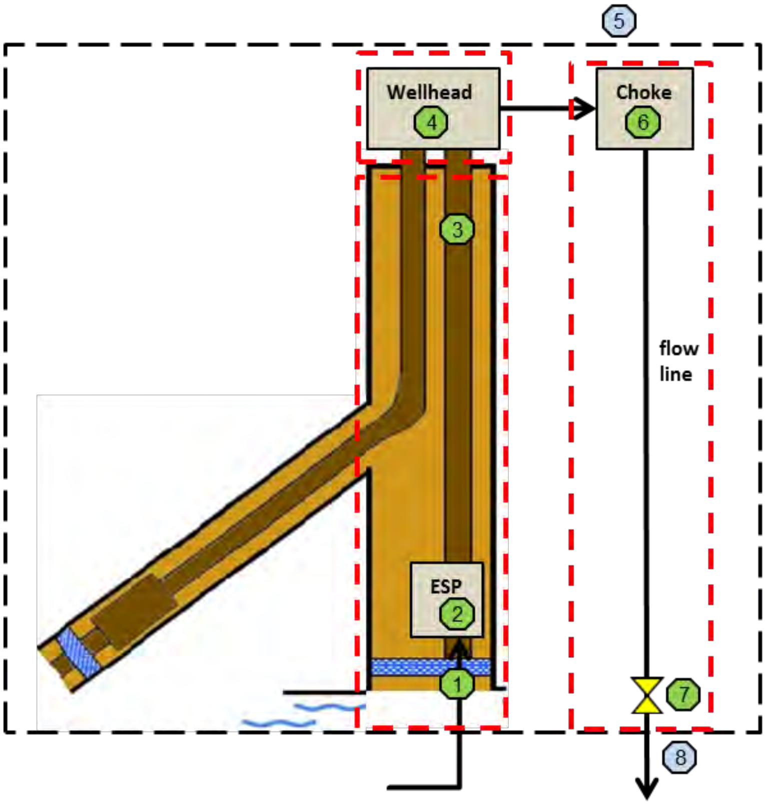
1	bottomhole pressure
2	ESP measured depth (static) ESP motor current ESP motor speed ESP motor voltage
3	annulus inner diameter (static) annulus outer diameter (static)
4	wellhead shut-in pressure
5	flow line pressure (inlet)
6	flow line choke setting
7	valve status
8	flow line pressure (outlet)

flow measurement

facility parameter

Figure 11–7. Internal flow model measurement points.

The next step is to draw boundaries around items with no flow measurements, as shown in Figure 11–8.



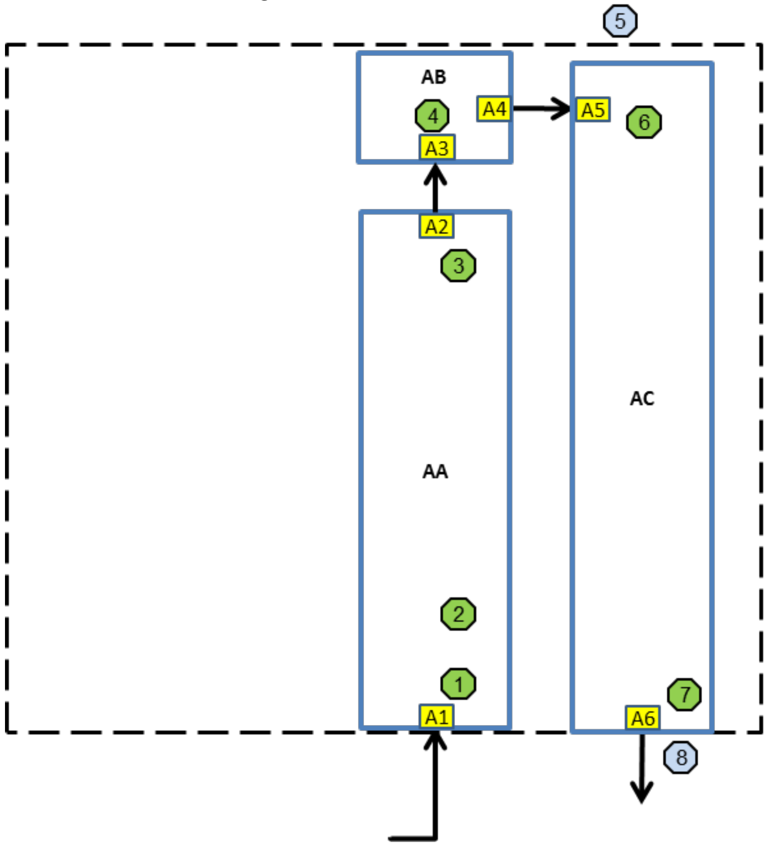
1	bottomhole pressure
2	ESP measured depth (static) ESP motor current ESP motor speed ESP motor voltage
3	annulus inner diameter (static) annulus outer diameter (static)
4	wellhead shut-in pressure
5	flow line pressure (inlet)
6	flow line choke setting
7	valve status
8	flow line pressure (outlet)

flow measurement

facility parameter

Figure 11–8. Internal flow model boundaries.

Finally, make each boundary a unit in the flow network, label each unit, node, and port with a unique name, as shown in Figure 11–9.



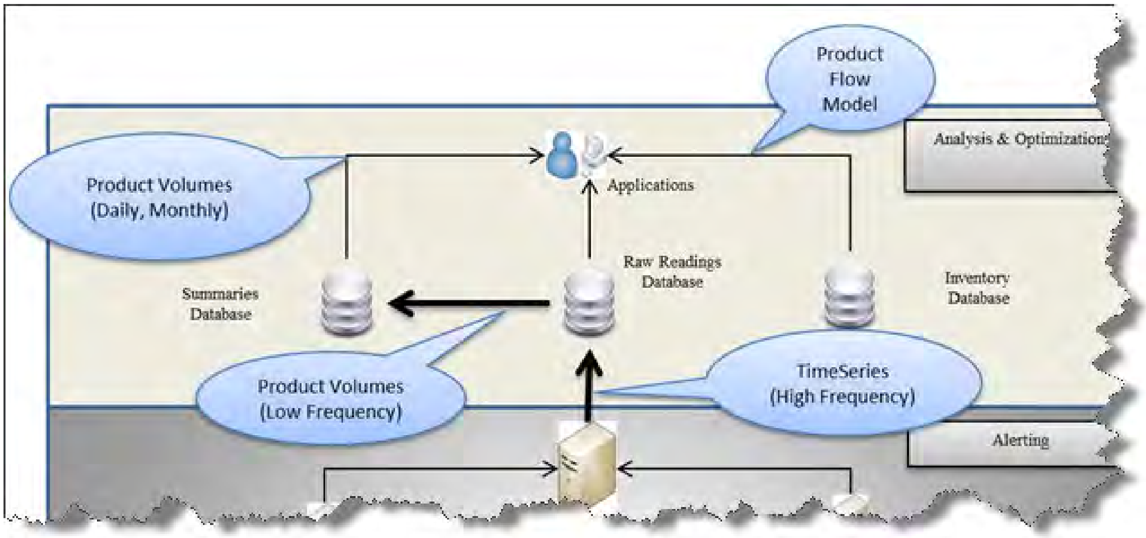
1	bottomhole pressure
2	ESP measured depth (static) ESP motor current ESP motor speed ESP motor voltage
3	annulus inner diameter (static) annulus outer diameter (static)
4	wellhead shut-in pressure
5	flow line pressure (inlet)
6	flow line choke setting
7	valve status
8	flow line pressure (outlet)

flow measurement
facility parameter
port

Figure 11–9. Internal flow model.

11.4 Retrieving Historian Data

The PRODML *timeSeriesData* data object is a best fit for retrieving tag-based, high-frequency data from the raw readings database.



Each measurement property in the flow model contains a *tagAlias* element which is used to retrieve a *timeSeriesData* data object. Refer to the *PRODML Identifier Specification* for more information on PRODML identifiers.

```
<expectedFlowProperty>
  <property>pressure</property>
  <dTimStart>2012-02-01T01:00:00+03:00</dTimStart>
  <tagAlias namingSystem="osi.aramco.com">SEP01_InletPressure</tagAlias>
  <expectedFlowProduct>
    <flow>production</flow>
    <product>oil and gas</product>
    <qualifier>measured</qualifier>
  </expectedFlowProduct>
</expectedFlowProperty>
```

Figure 11–10. Tag alias in flow model.

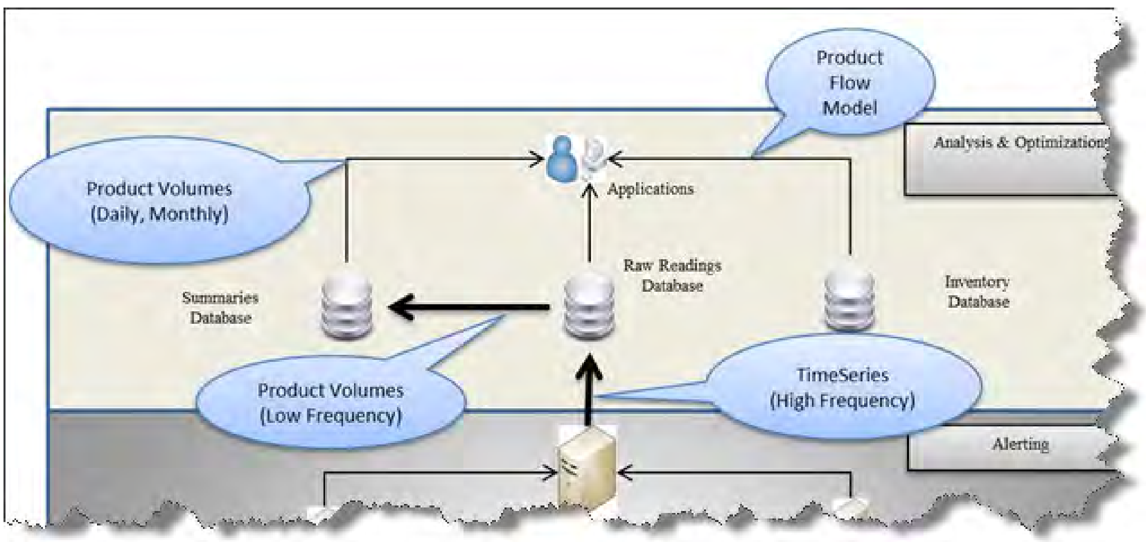
The corresponding *timeSeriesData* would be similar to the example in Figure 11–11.

```
<timeSeriesData>
  <name>SEP01_InletPressure</name>
  <key keyword="asset identifier">prodml://aramco.com/separator(SEP01)</key>
  <key keyword="flow">production</key>
  <key keyword="product">oil and gas</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">746.5214</doubleValue>
  <doubleValue dTim="2012-02-08T13:03:02+03:00">745.135</doubleValue>
  <doubleValue dTim="2012-02-08T13:02:03+03:00">744.0003</doubleValue>
  <doubleValue dTim="2012-02-08T13:02:04+03:00">742.613</doubleValue>
</timeSeriesData>
```

Figure 11–11. Historian time series example XML.

11.5 Retrieving Daily and Monthly Product Volume Reports

The PRODML productVolume data object is a best fit for retrieving historical production data from the summaries database.



The PRODML *productVolume* would be similar to that in Figure 11–12.

```
<productVolume>
  <name>Daily production summary example for separator</name>
  <periodKind>day</periodKind>
  <productFlowModel uidRef="PFM01">Saudi Aramco CR-0012 Flow Network Example</productFlowModel>
  <!-- separator -->
  <facility>
    <name kind="separator" uidRef="SEP01">Separator #1</name>
    <facilityParent1 kind="field" uidRef="F01">Ghawar</facilityParent1>
    <unit uidRef="U05">E</unit>
    <contextFacility kind="field" uidRef="F01">Ghawar</contextFacility>
    <!-- oil flow (outlet) -->
    <flow>
      <kind>production</kind>
      <port uidRef="9">F-PL01 outlet</port>
      <qualifier>derived</qualifier>
      <product>
        <kind>oil</kind>
        <period>
          <dTim>2012-02-01T00:00:00+03:00</dTim>
          <flowRateValue>
            <flowRate uom="m3/d">2250</flowRate>
          </flowRateValue>
        </period>
      </product>
    </flow>
    <!-- gas flow (outlet) -->
    <flow>
      <kind>production</kind>
      <port uidRef="10">F-PL02 outlet</port>
      <qualifier>derived</qualifier>
      <product>
        <kind>gas</kind>
        <period>
          <dTim>2012-02-01T00:00:00+03:00</dTim>
          <flowRateValue>
            <flowRate uom="m3/d">360</flowRate>
          </flowRateValue>
        </period>
      </product>
    </flow>
    <!-- water flow (outlet) -->
    <flow>
      <kind>production</kind>
```

Daily summary for oil

Daily summary for gas

Figure 11–12. Daily production example from summary database.