# RESQML Business Overview and Use Case Guide

For RESQML v2.1

| | |
|---|---|
| RESQML Overview | The RESQML 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 and flexibility. |
| Version of Standard | 2.1 |
| Abstract | An introduction to RESMQL for both domain/petro-technical professionals and software/IT professionals. Overview of the business value and a representative list of use cases supported by the current version of the standard |
| Prepared by | Energistics and the RESQML SIG |
| Date published | 11 November 2016 |
| Document type | Business overview guide |
| Keywords: | standards, energy, data, information, process, reservoir model, shared earth model |

energistics®
Energy Standards

| Document Information | |
|---|---|
| **DOCUMENT VERSION** | 1.0 |
| **DATE** | 11 November 2016 |
| **Language** | US English |

| Amendment History | | | |
|---|---|---|---|
| **Std Version/ Doc Version** | **Date** | **Comment** | **By** |
| 2.1 / 1.0 | 11 Nov 2016 | For RESQML v2.1. | Energistics and the REQML SIG |

# Table of Contents

# 1 Introduction

## 1.1 What is RESQML?

RESQML is an XML- and HDF5-based data-exchange standard that facilitates reliable, automated exchange of data among software packages used in subsurface workflows. RESQML consists of a set of XML schemas (XSD files) and other standards-based technology, which developers implement into software packages. Software that has implemented RESQML can read and write the standard format (**Figure 1-1**).

RESQML has been developed by a global consortium of operators, service companies, software vendors, and government agencies under the umbrella of Energistics.

### 1.1.1 **Subsurface Workflow Challenges**

The exploration and production (E&P) subsurface workflow is lengthy, iterative, and complex. It involves many people from different disciplines, sometimes different companies, and use of many different software packages for complex analysis, interpretation, modeling, and simulation.

This multi-discipline, multi-company, multi-software environment is iterative and requires users to move data back and forth between different software packages. Many of these packages use different data formats—often proprietary and incompatible.

This inherently complex process and inability to easily exchange data means E&P companies and their people face challenges that include: knowledge loss, rigid workflows, difficulty characterizing and sharing uncertainty, data loss, and productivity loss. (For more information, see Section 2.1.1 (page 9).

### 1.1.2 **How RESQML Helps Address These Challenges**

RESQML-compliant software can read and write this standard, common format, eliminating data incompatibility and the need for reformatting. Figure 1-1 below is a high-level overview of how RESQML works and the workflows it supports. The newest capabilities (see Section 1.5.2 (page **Error! Bookmark not defined.**)) help RESQML deliver these benefits:

- Delivers a "knowledge hierarchy" to organize data and transform it into knowledge.
- Increases workflow flexibility, for example, with partial model transfers that allow you to update/transfer only data that has changed.
- Supports traceability, with universally unique identifiers for each top-level data object and key metadata for data sources, updates, dates of change, etc.
- Supports uncertainty management through increased ability to run more scenarios and realizations and reliably update models.
- Defines a rich set of subsurface data objects and enables transfer of detailed models and a variety of model types.
- Improves efficiency for both petro-technical and IT professionals.

For more information on the challenges of subsurface workflows and how RESQML helps address them, see Chapter 2 (page 9).

For a list of supported use cases, see Chapter 3 (page 13).

**RESQML Supported Workflows**

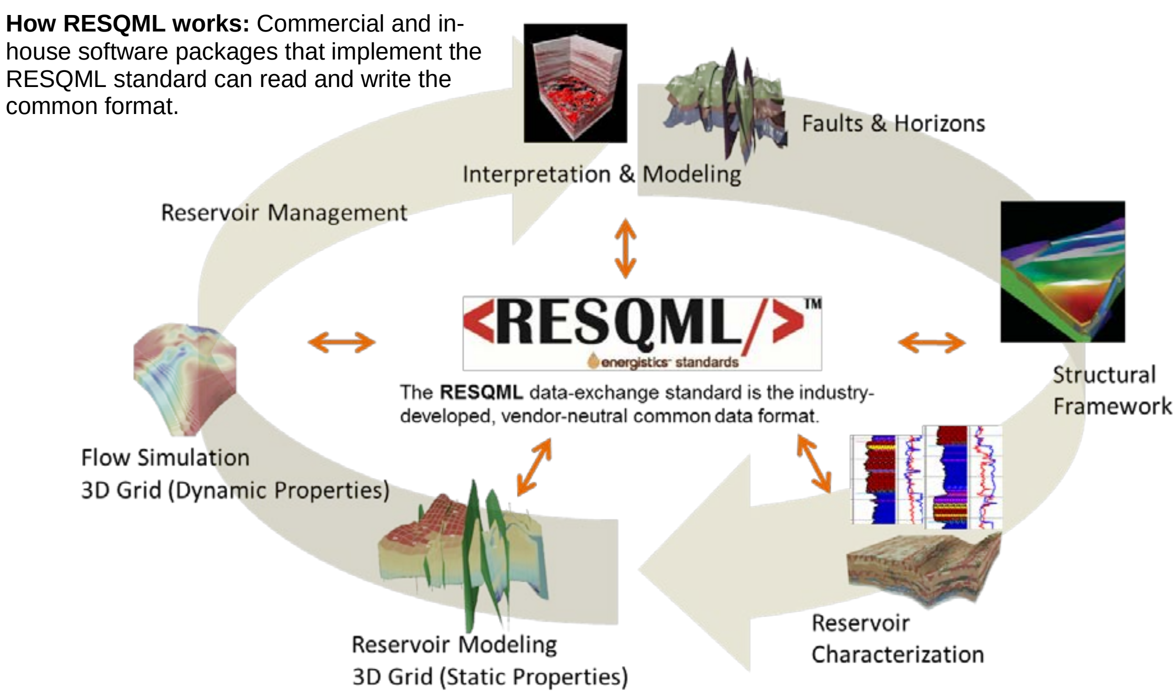


**Figure 1–1**

**Figure 1–2. Implementing RESQML in software used in the E&P subsurface workflows streamlines data flow among the many different software packages used. The latest version supports more workflows and more flexible workflows. New capabilities provide a rich set of data objects, a well-defined knowledge hierarchy throughout the model, methods for specifying and transferring relationships among data objects, and the ability to group all of the information into a single package, with associated files for floating point array data.**

## 1.2 Audience, Purpose, and Scope

This document is intended for petro-technical/GG&E domain professionals and information technology (IT) professionals—programmers, developers, architects and others—who are implementing RESQML into a software package.

This guide is a high-level introduction to RESQML—what it can do and how it works. It is intended to help people begin to understand RESQML, its benefit potential to the E&P business and workflows (how it can help solve the challenges described above).

Specifically, this guide includes business benefits of using RESMQL (Chapter 2 (page 9)) and a representative list of workflow use cases that it supports (Chapter 3 (page 13)).

## 1.3 Documentation Conventions

Documentation for RESQML observes the conventions listed in the following table.

| | Document/Resource | Description |
|---|---|---|
| 1. | Mandatory Behavior | Mandatory behaviors are specified as business rules as shown in the example below.<br>BUSINESS RULE: Array length is the number of cells in the grid or the blocked well. |
| 2. | Document Hyperlinks: Internal | Though no special text-formatting convention is used:<br>All section, page and figure numbers in this and all RESQML and |

| | Document/Resource | Description |
|---|---|---|
| | | Energistics documents are hyperlinks. The table of contents is also hyperlinked. |

### 1.3.1 RESQML Jargon

The notion of "transfer" is referring to the exchange of data between any two RESQML-enabled software packages. Frequently we also refer to "exporting" or "writing" data to the RESQML format and "importing" or "reading" data.

In this context, the terms "software," "software package," "application" and others are used interchangeably—no special distinction is made between these terms. Software that writes or exports RESQML data is informally referred to as a "writer", and a package that is reading or importing RESQML data is referred to as the "reader."

## 1.4 Resource Set

For the complete list of resources available for RESQML, including resources for the Energistics Common Technical Architecture, see the *RESQML Technical Usage Guide*.

## 1.5 What's New in RESQML v2.1?

RESQML v2.1 includes:

- Design improvements including:
  - Improved data storage for time series and multiple realizations
  - Simplified treatment of patch indexing
- Incorporation of previous "add-on" data objects (added to the v2.0 data model in v2.0.1) into the main design packages:
  - Property series and streamlines have now been integrated into RESQML
  - Activity model is now integrated into the Energistics CTA

### 1.5.1 Features of RESQML v2.0+

RESQML v2.0 (published in September 2014) added significant capabilities to the previous commercial version (RESQML, v1.1). For new RESQML users, the v2.0 changes are summarized here. The overarching design objectives were to:

- Improve the richness of existing domain objects.
- Add new domain objects to the data model to support the more sophisticated capabilities of the software packages used in E&P workflows.
- Add relationships among data objects.
- Support flexible workflows, for example, by supporting partial model updates.

In support of these design objectives, RESQML includes these capabilities:

- A subsurface knowledge hierarchy based on "features, interpretation, representations and properties" that describes subsurface features, multiple interpretations of those features, multiple representations of those interpretations, and the properties that may be "attached to" the representations.
- A standard methodology for the representation of subsurface scenarios and uncertainty.
- Support for structural, stratigraphic, and reservoir models:
  - At the interpretation level: definition of the involved geologic features with their relationships, i.e., contacts and their characteristics, within the context of the model.
  - At the representation level: topology and geometry of the model, either surface or volume. Ability to define contacts using nodes of associated geologic features.

- Associations between structural and stratigraphic features with the 3D reservoir grid.
  - Achieved using subrepresentations of the grid, which are defined by selecting particular elements of the grid such as the layers, pillars, cells, cell faces, etc.
- Structured and unstructured grids or a mixture of both. New grid capabilities include:
  - Block-centered grids defined by cell-centered geometry and properties, and their associated connectivity, to support both simple grids and current unstructured simulators.
  - Polyhedral and perpendicular-bisector (PEBI) grids to provide the visualization and geometry required by current fluid flow and geomechanical reservoir simulators.
  - Corner-point grids as used by reservoir simulators.
  - Tetrahedral grids as used by geomechanical simulators.
  - Ability to use parametric lines for pillar grid definitions.
  - Simplified grid descriptions for simple grid types, e.g., rectilinear or tartan grids.
  - Higher order extensions for grid cell geometry and for grid properties.
- Use of wellbore information from WITSML, the Energistics exchange standard for well and log information, which includes:
  - Support for wellbores, trajectories, logs, markers, and blocked wellbores.
  - Use of HDF5 storage.
  - Reference to existing WITSML data.
- Support for multiple topologies, geometries, and properties for multiple time steps and multiple model realizations.
  - These capabilities are shared across the schema and are not restricted to specific data objects.
- Ability to transfer only those parts of a model that have changed (partial model transfer), which enables the workflow flexibility to test alternative scenarios and characterize the range of uncertainty in structural and stratigraphic frameworks, as well as reservoir rock and fluid properties.
- Ability to group a variety of data objects and files from multiple sources into a single "package" for transfer using the new Energistics Packaging Convention (EPC), which is an implementation of the Open Packaging Conventions (OPC) standard.
- Ability to combine data referenced to different coordinate reference systems (CRS), for example, a RESQML reservoir model and WITSML wellbores, trajectories, and markers. The RESQML package supports a single global CRS and one or more local CRSs.

### 1.5.2 Moving from RESQML v2.0.1 to v2.1

RESQML v2.1 is considered a “breaking change” from the previous version (v2.0.1), so no transforms for conversion are available.

# 2 Business Overview

This chapter explains the business value of RESQML—identifying the business and domain problems it helps solve.

For a representative list of supported use cases, see Chapter 3 (page 13).

## 2.1 Why is RESQML Important to E&P?

RESQML delivers an industry-defined common data format, which helps ensure data integrity, capture and preserve asset team knowledge, better manage uncertainty, and improve subsurface workflow flexibility.

### 2.1.1 The Problems: Knowledge Loss, Rigid Workflows, and More

The exploration and production (E&P) subsurface workflow is lengthy, iterative, and complex. It involves many people from different disciplines, sometimes different companies, and use of many different software packages for complex analysis, interpretation, modeling, and simulation.

This multi-discipline, multi-company, multi-software environment is iterative and requires users to move data back and forth between different software packages. Many of these packages use different data formats—often proprietary and incompatible. This inherently complex process results in these business challenges:

- **Knowledge Loss.** Models don't just contain data, they include significant asset team knowledge—the decisions on picks, relationships between the objects, literally all of the geoscience interpretation process. These types of decisions and knowledge are often the most time consuming and crucial part of developing a coherent model. However, typically this knowledge is not explicitly captured by software nor transferred in data exchange, so extensive communication (offline of the model) between sender and receiver is required for receivers to ensure they understand the model, or receivers must simply try to figure it out for themselves.
- **Rigid Workflows.** Typical industry data transfers are of fairly complete proprietary models, or of the basic data components, with no support for relationships among components or to update or transfer only those parts of a model that have changed. These rigid workflows can inhibit a team's ability to iterate on multiple scenarios.
  - *No Traceability.* Iterative data-exchange among many software packages also means that asset teams typically have no traceability regarding the source of the data they received. For example, what software was used to pick this key horizon or fault in a structural model?
- **Difficulty Characterizing and Sharing Uncertainty.** By its very nature, subsurface modeling is rife with uncertainty; in fact, a key part of the modeling process is trying to quantify that uncertainty. But data and knowledge loss only add to uncertainty.
- **Data Loss.** When transferring data back and forth among these software packages, incompatible data formats require data reformatting (often manually), which decreases worker productivity and results in data loss and errors, ultimately reducing the reliability and integrity of modeling and analysis.
  - *Inability to Integrate Next-Generation Technology into Subsurface Workflow*s. The market has introduced several "next generation" reservoir simulators, for example, with increased integration between geomechanical calculations and flow simulation, especially useful for unconventional resource development. However, the inability to reliably transfer and use this new vital data and integrate it into inflexible workflows creates barriers to uptake and use of these new technologies.
- **Productivity Loss.** These issues of data and knowledge loss and uncertainty force petro-technical professionals to spend precious time determining, researching, and resolving errors—time they could be spending on actual subsurface domain work.

### 2.1.2 The Solution: RESQML

Software packages that implement RESQML can read and write "comprehensive" subsurface models—that is, the model data and the relationships among the data that define how it "goes together" as a model—to this vendor-neutral, industry-defined, common data format. A "comprehensive" subsurface model could contain the structural and stratigraphic frameworks along with 3D grids and associated reservoir properties. It might also contain wells and seismic coordinate references. Conversely, to increase efficiency and support flexible workflows, RESQML also makes it possible to logically transfer only parts of models and associated data.

RESQML-enabled software continues to process and save data in the software's native environment. And, when users need to move data and models to the next software package in their workflow, they can choose to export all of the information to the RESQML format. That next software package may be a tool used by another discipline in the workflow or by a partner company in a joint venture. If that software is RESQML-enabled, it can read the RESQML format and process the data in its native environment.

RESQML supports the workflows shown in **Figure 1-1**; the following section provides a more detailed example of how it works.

### 2.1.3 RESQML Workflow: A Simple Example

**Figure 2-1** is an example of a very simple subsurface workflow using RESQML-enabled software. When users need to move data to the next software application in their workflow, they choose to write (export) data to the RESQML format. In this example, User A using Software A writes the data to the RESQML format, which is transported in an EPC file (for more information, see the *Energistics Packaging Conventions (EPC) Specification*). That next software application may be a tool used by another discipline in the workflow or by a partner company in a joint venture. If that software application is RESQML-enabled, it can read (import) the EPC file containing the RESMQL data and process the data in its native environment.

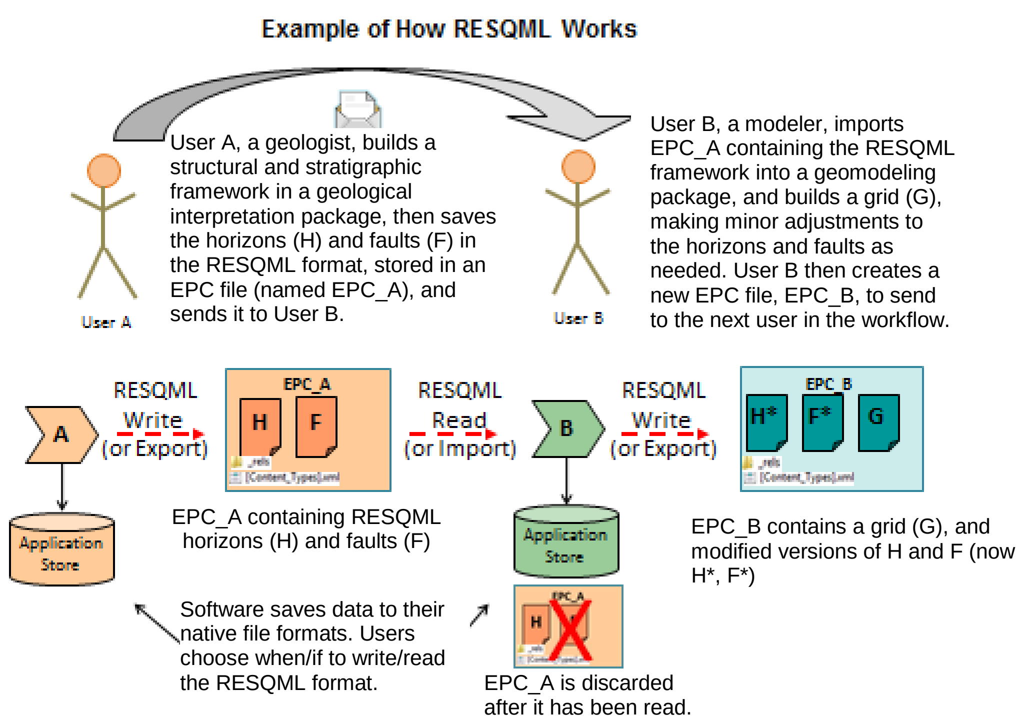


**Figure 2–1. A user writes (exports) a file to the RESQML format, which is stored in an EPC file and may be read (imported) by other RESQML-enabled software.**

#### *2.1.3.1 What's in a RESQML Data Exchange?*

When software applications exchange data using RESQML, that data exchange is composed of:

- One EPC file (a container file technology) that groups together in a single file all the XML data objects for the exchange. The Energistics Packaging Conventions (EPC) file is essentially a zip file.
- One (or possibly more) related HDF5 files that contain the floating point data for the XML data objects. HDF version 5 is a standard designed for the flexible and efficient I/O for high volume and complex data—particularly when compared to XML.

For more information, see the *RESQML Technical Usage Guide*.

### 2.1.4 Benefits of RESQML

The first version of RESQML began addressing some of these challenges; new and expanded capabilities in the latest version provide even more value. Implementation of the RESQML standard includes the benefits listed below.

For more information on the concepts discussed here, see the *RESQML Technical Usage Guide* (For a link to this document, see Section 1.4 (page 7).).

- **Delivers a "knowledge hierarchy" to organize data and transform it into knowledge**. RESQML has expanded the concept of "data" to now include the relationships between data objects. Specifically, RESQML can now transfer abstract subsurface features, human interpretations of those features, the data representations of those interpretations, and the properties indexed onto those representations—which provides the ability to transfer knowledge about a reservoir model, not simply its elemental components.

  RESQML also includes well-defined mechanisms to explicitly specify the relationship between and among different data objects to create organizations, which themselves may be grouped together to define an earth model (**Figure 2-2** (page 12)).

  This feature/interpretation/representation/properties knowledge hierarchy (informally referred to as "FIRP) is a key organizing concept in RESQML
- **Increases workflow flexibility.** The new data model and organization now makes it possible to transfer only those parts of a model that have changed. As team members continue the work to acquire data and knowledge and further develop subsurface models, they can more easily share the updates, which will allow better integration of activities across local and extended teams. This capability also makes it easier for asset teams to run multiple scenarios and realizations, because only data updates need to be transferred.
- **Supports traceability.** RESQML enables traceability of key information in models, such as by whom, when, and with which software it was last updated. Traceability also supports workflows and regulatory requirements such as newer, more stringent reserve reporting requirements that requires operators to archive and be able to reproduce models used to generate and document reservoir models.
- **Supports uncertainty management.** The ability to run more scenarios and realizations and then quickly and reliably transfer the results and interpretations between applications helps identify uncertainty and risk. The consistent RESQML data format also enables use of more data in new ways, for example, real-time visualization of the subsurface during drilling operations.
- **Defines a rich set of subsurface data objects and enables transfer of detailed models and more model types.** The current version of the standard expands the number of data objects defined and the richness of these descriptions (for a list of data objects, see the *RESQML Technical Usage Guide*). This coverage and capability is especially important in the reservoir simulation and geomechanical finite element domains, where industry standards have been lacking for all unstructured grid types, despite the number of applications that now exist and use these grids.
- **Improves model completeness.** RESQML helps to ensure that more data can be transferred and in a coherent format. RESQML's modular design lets users focus on specific steps in a workflow and transfer only data that is necessary to update a model or to document the data sets being used in

analysis, interpretation, and modeling. New technologies allow RESQML to handle very large grids and data sets quickly and efficiently.

- **Improves efficiency.** The ability to exchange data in a consistent, reliable format means petro-technical professionals have more time to focus on domain problems instead of trying to manually map data between applications. Software developers can eliminate the need for custom, one-off connectors between applications, allowing them to focus on enhancements for domain functionality. When the data flows easily from one stage or process to another, workflows become more efficient.

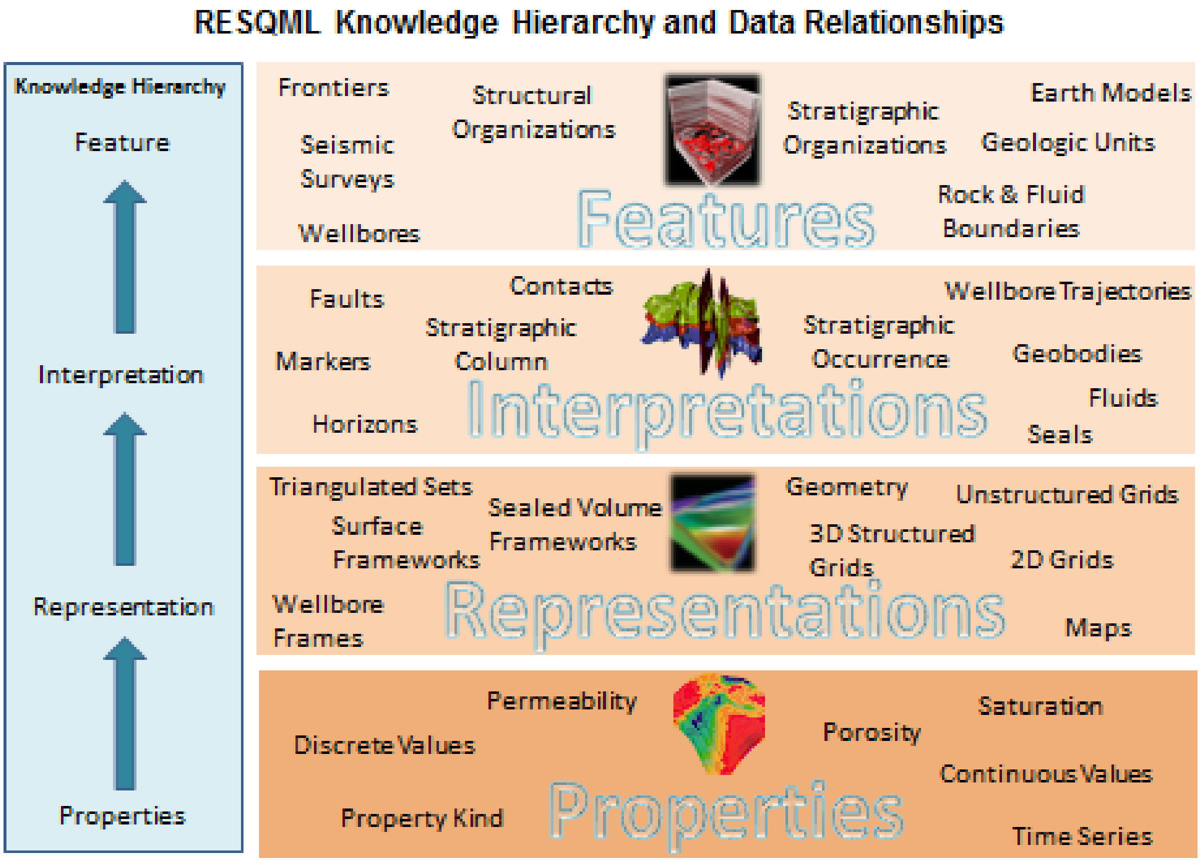


Figure 2–2. The feature/interpretation/representation/properties knowledge hierarchy is a key organizing concept in RESQML. For more information, see the *RESQML Technical Usage Guide*.

# 3 Use Cases

This chapter includes a representative list of use cases supported by RESQML. It highlights just a few of the high-level, representative use cases now supported by RESQML. That is, RESQML supports more than what is listed here.

This chapter is for use by both petro-technical and information technology (IT) professionals who want to use RESQML use cases to understand the business/domain application of RESQML.

NOTE: Additional use cases may be listed with content that explains related data objects. For more information, see the *RESQML Technical Usage Guide* (For a link to this document, see Section 1.4.1 (page **Error! Bookmark not defined.**).)

## 3.1 Use Case Defined

Use cases are tools used, often in software or systems engineering, to define from a user perspective the tasks that users want to perform and the goals they expect to accomplish. Developers employ these use cases as guides to develop solutions (software, systems or standards) that meet user requirements.

Many different terms exist for use case—for example, *user scenarios* or *stories, task scenarios*, or *user vignettes*.

### 3.1.1 **Benefits of Use Cases**

Use cases provide a structured format that:

- Clearly defines the tasks and goals of users, ensuring that RESQML provides the functionality that users need.
- Helps prioritize RESQML development based on business/user needs.
- Helps developers to clearly define and develop RESQML.
- Delivers a way for testing RESQML. Users should be able to accomplish the defined task or goal as described in the use case.

## 3.2 Full Model Transfer

Full model transfer refers to the ability to exchange all the "pieces" that make up a model and the relationships between those pieces, so that the model may be fully and accurately reconstructed by the software package receiving the data transfer. This use case also includes the notion of being able to transfer or move the model as "one thing" and be sure that all the pieces that comprise that model are included.

A model may range from a simple set of horizon and fault maps to a complete earth model, with a detailed description of stratigraphic and structural relationships.

### 3.2.1 **Example Using Alwyn Field Data**

This example using the Alwyn dataset, which is derived from field data from a North Sea field, helps illustrate some of RESQML key concepts. (The Alwyn dataset was donated to Energistics by Total. It provides an actual, robust dataset, which RESQML has been using for testing and development since v1.0. )

The horizons and faults are the fundamental data objects. The horizons appear as 2D grid representations, a horizon interpretation, and a genetic boundary feature. The faults are triangulated set representations, a feature interpretation, and a tectonic boundary feature. The combination of three corresponding objects within the model (feature/interpretation/representation) is an example of the RESQML knowledge hierarchy.

The horizons and faults are organized into a structural organization. As a representation, we have a non-sealed surface framework, which references the 2D grid and triangulated horizon and fault representations but also includes point sets, polylines, and polyline sets to describe their intersections, contacts, and boundaries. This information together represents a structural organization interpretation, which interprets an organization feature. Again we have an application of the RESQML knowledge hierarchy, but now it is used to transfer the relationships between the fundamental objects.

At the utility level, a RESQML dataset expresses all of these relationships within XML files, but the physical locations of the point data of the representations are stored in an external HDF5 file. A local 3D coordinate reference system is used to position the model with respect to global 1D + 2D coordinate reference systems. An EPC file is used to store this external relationship and to bundle together all of the data objects that constitute the model description.

Finally, this dataset includes a seismic survey. It consists of a seismic lattice feature, which characterizes the inline-crossline specifications of the survey. Seismic surveys do not require a distinct representation. Instead, seismic inline-crossline “coordinates” are added to the specification of the structural representations so that the relationships between the geologic description and the seismic survey may be stored and transferred as part of the RESQML model.

## 3.3 Partial Model Transfer

The first use case (Section 3.2 above) gives a sense of the richness of description that may be included in a RESQML model. The use cases in this section provide examples of workflow flexibility and partial model transfer.

As its name implies, partial model transfer refers to the ability to exchange pieces of data in the context of a model, but without having to exchange the entire model, while still retaining the correct context. An infinite number of scenarios and workflows exist for partial model transfer; a few are described here.

### 3.3.1 Structural Framework

Consider an example of the Alwyn model that consists of only the horizons and faults, which have previously been transferred to another 3D geologic modeling software package, simply for the purpose of visualization. Now that additional work has been done, we are ready to transfer the structural organization (described in the first example) to the same software package. However, we do not need to transfer the horizons and faults again, because they have not changed. With RESQML, we can simply reference the previously transferred horizons and faults by their universally unique identifiers (UUIDs) and associated metadata; these key pieces of data allow a RESQML-enabled application to recognize the data it has previously received.

### 3.3.2 Add/Update Properties

The reservoir simulation context provides us with another example of partial model transfer. Consider a static reservoir model constructed from well and seismic data. It consists of horizons, faults, and a structural framework, together with an initial fluid interpretation, which together constitute an earth model interpretation. A 3D faulted grid, discretized wells, and initial fluid contacts are constructed in the geologic modeling application and are then transferred to a reservoir flow simulator. This simulator calculates pressures and saturations and the reservoir engineer contrasts the predictions against the actual field performance. To improve the match against history, the engineer needs to increase volumes in some portions of the field and reduce them in others. She also needs to introduce several flow baffles as transmissibility modifiers to capture barriers that are apparent in the dynamic response of the field. The engineer now wishes to reconcile these changes with the geologic description and needs to transfer the pressures, saturations, pore volume multipliers, and transmissibility multipliers back to the geologic modeling application. There is no need to transfer the grid itself, or the static properties that described the initial conditions for the flow simulation.

## 3.4 Merging Data from Two or More Packages into Another Package

If I merge data from different software packages into one software package, and the source data is in the same CRS, then the merged data should be geometrically consistent with respect to each and the external world.

For example, I should be able to accurately compare the measured depth (MD) of a well with a surface location on a mountain to the MD of well with a subsea surface location, and also understand the different surface locations with respect to one another.

## 3.5 Traceability

### 3.5.1 Basic Traceability

When I import a model--either a full model transfer or only specific data objects (partial model) I want to be able to determine some basic information about the data objects and how it was created. For example: Are these new data objects or updates to data objects that were previously transferred? Who created it? When? What software/version was used to create it? What RESQML version was used to create it? When was it last updated? By whom?

### 3.5.2 Advanced Traceability: Transfer Activities Linked to Data Objects

For given data objects (e.g., grids) I want to be able to specify activities (e.g. tasks) that change the data object.

## 3.6 Time-to-Depth Conversion

This use case shows how RESQML may share work product between the reservoir modeling and geophysical domains. Consider a fairly complex 3D reservoir grid together with the structural framework, horizons, faults and contacts that went into its construction. The seismic survey from which this reservoir model was originally derived has been re-processed. However, before the survey is re-interpreted, the geophysicist would like to visualize the current reservoir modeling work product, which also includes geologic interpretation and dynamic data calibration. However, while the reservoir model has been constructed in depth, the geophysical application only works in two-way-time.

Although RESQML itself cannot perform a time-to-depth conversion, it can transfer the results of such a conversion. Specifically, RESQML can associate new 3D grid, structural framework, horizon, fault and contact representations with the existing representations by redefining the geometry of the 3D points, which provide the spatial location of the representations. Where the first set of representations was in depth, the new representations may be in two-way-time. None of the complex grid design or structural framework description needs to be duplicated, because these new representations retain the knowledge of the supporting representations from which they were derived. Once a representation exists in time, it may be transferred to the geophysical application.

This capability may also be used to transfer related representations into multiple coordinate reference systems. Although RESQML does not support coordinate transformations, it can transfer the results of coordinate transformations calculated within subsurface applications.

## 3.7 Transfer Unstructured Reservoir Modeling Grids

The current “next generation” reservoir simulators have moved to a geometry-free grid description. A traditional corner point grid describes the 8 nodes on the corners of each cell to specify a computational grid. Geometry is fundamental and the relationships between cells need to be inferred from the juxtaposition of cells. Now, a RESQML grid may instead consist of a list of cells, with their volumes and depths stored as properties, and a list of grid connections, with their connection areas, flow transmissibilities or fluid flux. In addition, we may choose to add geometric information to the nodes of these cells, which provides us with the ability to mix list-based geometry-free descriptions for the simulator and the geometric representation of a grid for visualization or for integration with other geologic and geophysical data. Before these latest capabilities were available in the current version of RESQML, only proprietary formats have been available for the list-based descriptions, and no data-transfer formats have been available to support a mixture of geometry and list-based representations.

## 3.8 Model Import Flow Simulator

| Use Case Name | RESQML Model Import to Flow Simulation |
|---|---|
| Version | 1.0+ |
| Goal | A Reservoir Engineer wishes to understand the fluid flow, pressure and energy distribution in a reservoir as it evolves with time. As a first step, the Reservoir Engineer wishes to incorporate a geological property model, developed by a geologist using geological property modeling software, into a separate dynamic flow simulation software. |
| Summary Description | A geologist exports a static reservoir description that was developed in geological property modeling software into a RESQML model. A reservoir engineer imports the static reservoir model into dynamic flow simulation software. |
| Actors | Reservoir Engineer<br>Geologist<br>Geological Modeling Software<br>Flow Simulation Software |
| Triggers | An asset team needs to either<br>(a) create a field development plan,<br>(b) compare different possible locations to drill new wells in a reservoir, or<br>(c) to optimize the operating policies for an existing developed field. |
| Pre-conditions | 1. A gridded, static, geological model must have been created and exist in a geological modeling software.<br>2. The geological modeling software and the flow simulation software must be capable of exporting and importing RESQML models. |
| Primary or Typical Scenario | 1. The geologist starts the geological modeling software and loads his geological model for the reservoir.<br>2. The geologist exports the geological model to a RESQML model.<br>3. The geologist informs the reservoir engineer of the file system location of the RESQML model.<br>4. The reservoir engineer starts the flow simulation software.<br>5. The reservoir engineer tells the flow simulation software to import the RESQML model from the file system location given to him by the geologist.<br>6. From the RESQML model, the flow simulation software should obtain the following:<br>(a) a structured grid of hexahedral cells (corner point) stored in some efficient format<br>(b) a set of properties defined for each cells, such as porosity, permeabilities, saturations, together with the name of each property, its dimensionality and units of measure.<br>(c) the location of well trajectories such that their intersection with grid cells may be calculated<br>(d) the location of fault surfaces with respect to grid cell faces,  and other discontinuities in the grid<br>The flow simulation software must be able to re-create the static model description that the geological software had, including an exact duplication of the I,J,K numbering of the structured grid and properties on each grid cell. |

| Use Case Name | RESQML Model Import to Flow Simulation |
|---|---|
| | 7. The flow simulation software may also find the following information useful if it included in the RESQML model.<br>(a) properties (such as seal/non-seal or transmissibility multipliers) on fault surfaces<br>(b) perforation intervals, in measured depth, associated with trajectories<br>(c) well logs, in measured depth, associated with well trajectories<br>(d) geological unit names associated with grid K layers<br>(e) 2d surfaces, such as horizons, seabed surface, or land surface.<br>8. If the set of cell properties do not have "well known" names, then the reservoir engineer must match the name given in the RESQML model to one of the property names used in the flow simulation software. |
| **Alternative Scenarios** | 1. An error is encountered while exporting the RESQML model<br>2. An error is encountered while importing the RESQML model<br>3. The geologist uses the geological modeling software to upscale the geological model to a lower resolution model before exporting to RESQML. The RESQML model may contain both the fine scale geological model and one or more upscaled models with different grid resolutions and associated property sets.<br>4. The geologist exports several "realization sets" of grids and properties, for different geological conceptual models, or different geostatistical realizations. |
| **Post-conditions** | 1. The flow simulation software has all the components of the static geological model that it needs to simulate the flow in the reservoir. Other necessary data, such as fluid properties, rock-fluid interactions, and production operating conditions are entered separately. |
| **Business Rules** | If REQML does not mandate grid property names (with a careful set of definitions) then a company, business unit, or asset team must have a set of agreed upon names so that the reservoir engineer can understand how to map the RESMQL model names to the flow simulation software property names. |
| **Notes** | Alternative Scenarios (3) and (4) could be written up as separate use cases. I have included them here as a reminder. |
| **Definitions** | Geological Modeling Software is software used to construct a static, gridded description of a porous media reservoir located deep in the earth. The inputs to construct the model are typically seismic and well log data and geological assumptions.<br>Flow Simulation Software calculates the time-varying fluid saturation and pressure within a reservoir, given a static, gridded description of a porous media reservoir, a fluid property description, well locations, and time-varying well perforation and operating conditions. |

## 3.9 Model Export to Flow Simulator

| Use Case Name | RESQML Model Export From Flow Simulation |
|---|---|
| Version | 1.0+ |
| Goal | Having completed a dynamic flow history match in a dynamic flow simulation software, a Reservoir Engineer wishes to export forecast/predicted reservoir saturations back to a geologist (and his geological property modeling software) for well planning purposes. |
| Summary Description | A geologist exports a static reservoir description that was developed in geological property modeling software into a RESQML model. A reservoir engineer imports the static reservoir model into dynamic flow simulation software. |
| Actors | Reservoir Engineer<br>Geologist<br>Geological Modeling Software<br>Flow Simulation Software |
| Triggers | An asset team needs to plan a location to drill new wells in a "brown field" reservoir with a number of years of production history. The well planning must be done in the geological modeling software, but needs to import the hydrocarbon saturations calculated in the flow simulation software. |
| Pre-conditions | 1. A RESQML model containing a gridded static reservoir model was exported from the geological modeling software and imported into the flow simulation software.<br>2. The flow simulation software was used to perform a "history match" such that the simulation predicted production from the reservoir "matched" the observed field production. |
| Primary or Typical Scenario | 1. The reservoir engineer starts the flow simulation software for the history matched reservoir.<br>2. The reservoir engineer tells the flow simulation software to export the predicted fluid saturations and pressures on the reservoir grid, at some simulation time, to a RESQML model.<br>a) export an entire, new, RESQML model, or<br>b) append the properties to an existing RESQML model (the one previously imported and forming the static model used in the history match.<br>3. The geologist starts the geological modeling software and opens the RESQML model, then imports the needed parts of the model.<br>4. The geologist uses the geological modeling software, and the imported fluid saturations and pressures to plan a new well. |
| Alternative Scenarios | 1. An error is encountered while exporting the RESQML model<br>2. An error is encountered while importing the RESQML model |
| Post-conditions | 1. The geological modeling software can display and use the fluid saturations and pressures, predicted by the flow simulation software, in its well planning tool. |
| Business Rules | If REQML does not mandate grid property names (with a careful set of definitions) then a company, business unit, or asset team must have a set of agreed upon names so that the reservoir engineer can understand how to map the RESMQL model names to the flow simulation software property names. |
| Notes | Steps (2.a) and (2.b) could be written up as separate use cases. |
| Definitions | Geological Modeling Software is software used to construct a static, gridded description of a porous media reservoir located deep in the earth. The inputs to construct the model are |

| Use Case Name | RESQML Model Export From Flow Simulation |
|---|---|
| | typically seismic and well log data and geological assumptions.<br>Flow Simulation Software calculates the time-varying fluid saturation and pressure within a reservoir, given a static, gridded description of a porous media reservoir, a fluid property description, well locations, and time-varying well perforation and operating conditions. |

## 3.10 Horizon Attribute Extraction

| Use Case Name | Horizon Attribute Extraction |
|---|---|
| Version | v1.0+ |
| Goal | Take a horizon interpretation and extract an attribute (e.g. Curvature) from a seismic volume, at each note on the surface. |
| Summary Description | Transfer a horizon from an interpretation application into an attribute calculation application, extract attributes onto the horizon and then move the horizon back into the interpretation application for display. |
| Actors | Geophysicist |
| Triggers | Need to determine the next location to drill - where it has been determined that a particular seismic attribute (e.g. Curvature) is a good indicator of pay – unfortunately the interpolation application cannot calculate attributes. |
| Pre-conditions | An interpreted horizon.<br>The interpretation and attribute extraction application must be capable of exporting and importing RESQML horizon and horizon property models. |
| Primary or Typical Scenario | The geophysicist interprets a horizon in the interpretation application.<br>The geophysicist exports the horizon into a RESQML model.<br>The geophysicist (same or different one) loads the exported horizon into the attribute extraction application.<br>The geophysicist extracts the required attributes from a seismic volume onto the imported horizon – creating a new horizon property.<br>The geophysicist exports the newly created horizon property into a RESQML model.<br>The newly created horizon property is imported back into the original interpretation application and associated with the initial horizon.<br>The new property is draped over the Horizon and a drill location is determined. |
| Alternative Scenarios | The horizon and its new property are viewed in the attribute extraction application and not exported back into the interpretation application. |
| Post-conditions | A user is able to view the horizon and the extracted attribute horizon property together. |
| Business Rules | Horizons will need unique identifiers such that when the horizon property is imported back into the original application it is known which horizon the property should be associated with.<br>We need well defined names and units for horizon properties such that the same attribute is treated the same in both applications. |
| Notes | Curvature is just one example of a seismic attribute that is extracted onto a horizon. |
| Definitions | |

## 3.11 Transfer of Well Data and Reservoir Relationships

| **Use Case Name** | **Transfer of Well Data and Reservoir Relationships** |
|---|---|
| **Version** | v2.0 |
| **Author** | RESQML/Mike King |
| **Goal** | Preservation of well trajectory geometry, properties along wells (logs), and relationships with other reservoir and well objects, e.g., markers and blocked wells, WITSML wells and datum information. As a RESQML representation, the well data objects should allow expressions of feature/interpretation relationships. As a physical object, the well data objects should maximize re-use or access to WITSML and minimize duplication. |
| **Summary Description** | Wells are critical to the development and description of reservoirs, and their usage is heavily inter-disciplinary and multi-disciplinary. RESQML provides the capability to express and preserve the relationships between reservoir and wells. However, to minimize the amount of duplication between RESQML and WITSML, the RESQML “well” focuses in on the wellbore itself, and only accesses information about the well through a reference to WITSML.<br>The RESQML wellbore consists of a wellbore trajectory representation, to which we may associate the geometry of a wellbore, and a related wellbore frame, which provides the topological support for markers and properties. A special case of a wellbore frame is the blocked wellbore, in which the intersections between the geometry of the wellbore trajectory and a reservoir modeling grid (or grids) are used to specify the topological support for properties.<br>The goal is for the writing software to write wellbore representations that allow the following goals:<br>Goal 1: The reading software will be able to reproduce the geometry of a wellbore trajectory, irrespective of the choice of parameterization of that trajectory.<br>Goal 2: The reading software will be able to associate a wellbore framework with a wellbore trajectory, to provide the topological support for properties and relationships.<br>Goal 3: The reading software will be able to identity related WITSML information, if any. |
| **Actors** | Writer: Application which creates or has imported wellbore trajectory geometry, wellbore frame properties and markers. The writer is expected to export references to imported objects and/or export the data from imported objects, and to export the additional information created by the application.<br>Reader: Application which imports the RESQML wellbore trajectory representation and the wellbore frame representation is expected to use this information for the goals listed above. |
| **Triggers** | Trigger 1: Model is exported (writer)<br>Trigger 2: Model is imported (reader) and mapped into the new application’s internal data model |
| **Pre-conditions** | The writing application contains a wellbore trajectory, and/or, properties along that trajectory, and/or, markers along that trajectory.<br>The writing application may contain one or several property values for each interval or node of the wellbore trajectory frame. |
| **Primary or Typical Scenario** | Scenario 1: Wellbore trajectories are imported from a WITSML web-service, and then markers and properties along that trajectory are used to create a 3D detailed geologic model. |
| **Alternative Scenarios** | Scenario 2: Properties are predicted along a hypothetical wellbore trajectory in one application, and that trajectory and properties are exported for well targeting purposes. |
| **Post-conditions** | 1/ The reader is able to export the original and additional data it has created using their application-specific data formats, for use in their current workflows.<br>2/ The reader is able to export the original and additional data it has created using RESQML, for use in multi-vendor workflows. |

| Use Case Name | Transfer of Well Data and Reservoir Relationships |
|---|---|
| Business Rules | Wellbore trajectories should be preserved as unique objects when used in a workflow loop, e.g., geometry imported, and then when geometry, properties and a reservoir model are exported, the wellbore trajectory geometry should share a UUID with the original geometry. |
| Notes | Currently WITSML and RESQML do not share a mechanism for referencing of data objects. This has necessitated some duplication of information between the two data standards. It is expected that as these standards converge that additional optimization of the well/wellbore data objects will occur. |
| Definitions | RESQML data objects referenced by this use case:<br>Wellbore trajectory representation, wellbore frame representation, MD datum<br>Additional objects:<br>WITSML well trajectories |

## 3.12 Transfer of Blocked Wells

| Use Case Name | Transfer of blocked wells |
|---|---|
| Version | v1.1+ |
| Author | RESQML/Laurent Deny |
| Goal | Preservation of blocked wells corresponding to the intersection between a grid and one or several wells when transferring this grid from one user (writer) to another user (reader). This information and the corresponding upscaled-log properties should remain available for other purposes and does not need to be reconstructed by the reader. |
| Summary Description | Transferring properties values from wells logs to grid cells is a mandatory step in every grid-based property modeling and flow simulation. This process usually involved two steps. The first step is a discretization of the well trajectory inside the grid that identifies the grid cells intersected by a well. The second step is to compute for each cell several upscaled property values from well log segments corresponding to the trajectory associated to the cell. Because different software may perform these two tasks differently and produce different output by using different methods or numerical sensitivities, it is important to transfer this information to maintain consistency along the workflow.<br>Given that each well-grid intersection is affecting only a small number of cells inside the grid, it would be very wasteful to use an entire grid property to store each upscaled log for each well.<br>The goal is for the writing software to write blocked well representations that allows the following goals:<br>Goal 1: The reading software will be able to identify the grid cells corresponding to a given well trajectory intersection, irrespective of the grid complexity or cell numbering scheme.<br>Goal 2: The reading software will be able to associate the blocked well to an actual or pseudo-well.<br>Goal 3: The reading software will be able to associate multiple property values to each of these individual grid cells.<br>Goal 4: The reading software will be able to access the corresponding information efficiently without reading through the entire grid. |
| Actors | Writer: Application which creates the original grid geometry and grid numbering scheme. The writer is expected to provide the grid information and blocked wells with well identifiers that will associate blocked wells with actual or pseudo-wells.<br>Reader: Application which imports the RESQML Grid and blocked wells, and is expected to use this information for the goals listed above. |

| **Use Case Name** | **Transfer of blocked wells** |
|---|---|
| **Triggers** | Trigger 1: Model is exported (writer)<br>Trigger 2: Model is imported (reader) and mapped into the new application's internal data model |
| **Pre-conditions** | The writing application contains a discretized well trajectory that can be represented as a list of grid cells. Each of these cell based trajectories is associated to an actual or pseudo-well.<br>The writing application may contain one or several upscaled property values for each cell of the discretized well trajectory. |
| **Primary or Typical Scenario** | Scenario 1: Blocked well are created by a geologic modeling package and a simulator application and its pre-processor will use the blocked well trajectory cells information. |
| **Alternative Scenarios** | Scenario 2: Blocked wells and upscaled well properties are created by a geologic modeling package and a second modeling package will use the upscaled well property values as data to perform subsequent property modeling.<br>Scenario 3: Blocked wells and simulation results are exported by a simulator and a geologic modeling package will perform quality control by transferring back the simulation result onto the well as a well log. This quality control process may involve several grids at different resolution. |
| **Post-conditions** | 1/ The reader is able to export the original and additional data it has created using their application specific data formats, for use in their current workflows.<br>2/ The reader is able to export the original and additional data it has created using RESQML, for use in multi-vendor workflows. |
| **Business Rules** | Block well cell based trajectory and property information should be preserved and utilized within the reader application. |
| **Notes** | The limited availability of well information in this release is reducing the amount of information that can be exchanged for blocked wells. The need for well unique identifiers, more precise trajectory information and information about the used upscaling methods have been identified and should be addressed in subsequent releases. |
| **Definitions** | RESQML data objects referenced by this Use Case:<br>Grid, Blockedwells, Properties<br>Additional objects:<br>Well trajectories |

## 3.13 Transfer of Complex Reverse-Fault Reservoir Grids

Support for the transfer of complex reverse-fault reservoir grids from one gridding vendor to a second, using an "exploded" I, J, or K representation. In the second vendor's application, determine the well intersections of a well that passes through the region of the reverse fault. If the second vendor is a simulation application, calculate the full list of inter-cell transmissibilities across the fault assuming unit permeability and fluid properties. In both cases, export these derived quantities in the native format of the second application.