RESQML Use Case Guide

For RESQML V 1.1

---

**RESQML Overview**

An industry-developed, vendor-neutral format that 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:**
**Abstract:**

*Use Case Guide for RESQML v1.1*

This guide contains use cases for RESQML. Use cases serve as an important design tool to ensure that the specification serves the business/domain needs of the user community.

This use cases are updated and maintained with the RESQML specification.

**Prepared by:**

Energistics and the RESQML SIG

**Date published:**

17 October 2011

**Document type:**

User guide

**Keywords:**

standards, energy, data, information, process, reservoir model, shared earth model
Document Information

<table>
<thead>
<tr>
<th>DOCUMENT VERSION:</th>
<th>1.1</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE:</td>
<td>17 October 2011</td>
</tr>
<tr>
<td>Technical</td>
<td>e.g., Color: R: 210 G:124, B50</td>
</tr>
<tr>
<td>Language</td>
<td>US English</td>
</tr>
</tbody>
</table>

Acknowledgements
This version of the RESQML standard and documentation were developed by the Energistics RESQML Special Interest Group (SIG), which includes these members: BP, Chevron, Computer Modeling Group (CMG), Dynamic Graphics (DGI), Halliburton/Landmark, IFP Energies Nouvelles (French Institute of Petroleum), Paradigm, Roxar, Schlumberger, Shell, Total, Transform Software.

Usage, Intellectual Property Rights, and Copyright
The material described in this document was developed by and is the intellectual property of Energistics. Energistics develops material for open, public use so that the material is accessible and can be of maximum value to everyone.

Use of the material in this document is governed by the Energistics Intellectual Property Policy document and the Product Licensing Agreement, both of which can be found on the Energistics website, [http://www.energistics.org/legal-policies](http://www.energistics.org/legal-policies).

All Energistics published materials are freely available for public comment and use. Anyone may copy and share the materials but must always acknowledge Energistics as the source. No one may restrict use or dissemination of Energistics materials in any way.

Trademarks
Energistics™, POSC®, Epicentre®, WITSML™, PRODML™, Upstream Standards. Bottom Line Results.™, The Energy Standards Resource Centre™ and their logos are trademarks or registered trademarks of Energistics. Access, receipt, and/or use of these documents and all Energistics materials are generally available to the public and are specifically governed by the Energistics Product Licensing Agreement ([http://www.energistics.org/product-license-agreement](http://www.energistics.org/product-license-agreement)).
<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Comment</th>
<th>By</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>31 Dec 2010</td>
<td>First publication for version 1.0 of the RESQML standard.</td>
<td>Energistics and RESQML SIG</td>
</tr>
<tr>
<td>1.1</td>
<td>17 Oct 2011</td>
<td>Updated as necessary for version 1.1 of RESQML.</td>
<td>Energistics and RESQML SIG</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **Introduction** .......................................................................................................................................................... 1  
   1.1 Audience, Purpose, and Scope .............................................................................................................................. 1  
   1.2 Document Set and Resources ............................................................................................................................... 1  

2. **What are Use Cases and Why are they Important?** ................................................................................................. 2  
   2.1 Use Case Defined ..................................................................................................................................................... 2  
   2.2 Benefits of Use Cases ............................................................................................................................................ 2  
   2.3 Use Case Notation .................................................................................................................................................. 2  

3. **RESQML Use Cases** ............................................................................................................................................... 3  
   3.1 Extended Use Case and Testing with Alwyn North Field Data .............................................................................. 3  
      3.1.1 Resources .......................................................................................................................................................... 3  
      3.1.2 Use Case Overview ........................................................................................................................................... 3  
      3.1.3 Use Case Workflow ............................................................................................................................................ 4  
   3.2 Standard Use Cases ................................................................................................................................................. 5  
      Use Case 1: Exporting/Importing Consistent Model from Package A to Package B .............................................. 8  
      Use Case 2: Move Features, Such as Horizons or Faults, Independently (without a grid) .................................. 9  
      Use Case 3: Merging Data from Two or More Packages into another Package .................................................. 10  
      Use Case 4: Traceability ........................................................................................................................................... 11  
      Use Case 5: Move RESQML Model .......................................................................................................................... 15  
      Use Case 6: Model Import to Flow Simulator .......................................................................................................... 17  
      Use Case 7: Model Export from Flow Simulator ...................................................................................................... 20  
      Use Case 8: Horizon Attribute Extraction ................................................................................................................ 22  
      Use Case 9: Transfer of Blocked Wells ...................................................................................................................... 24  
      Use Case 10: Transfer of Complex Reverse-fault Reservoir Grids .......................................................................... 26
1. Introduction

RESQML is an XML-based data exchange standard that helps address the data-incompatibility and data-integrity challenges faced by petro-technical professionals when using the multiple software technologies required along the entire subsurface workflow, for analysis, interpretation, modeling, and simulation.

For a complete introduction to RESQML, see the RESQML Overview Guide.

This document contains use cases for RESQML. The standard may support any number of use cases; those contained herein were used for development and testing purposes and are provided to aid users and implementers.

1.1 Audience, Purpose, and Scope

This guide:

- 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.
- Briefly defines use cases and describes their purpose.
- Lists and describes RESQML use cases.

To ensure you are reading the latest version of this document, visit the Energistics web site, http://www.energistics.org/current-resqml-standards.

1.2 Document Set and Resources

The RESQML documentation set includes the following, which can be found at http://www.energistics.org/current-resqml-standards.

<table>
<thead>
<tr>
<th>Document</th>
<th>Purpose/Audience</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESQML Overview Guide</td>
<td>Overview of workflow and business processes that RESQML was designed to facilitate. A useful introduction to RESQML for both petro-technical and IT technical professionals.</td>
</tr>
<tr>
<td>RESQML Usage Guide</td>
<td>Overview of key technical concepts for implementing the RESQML standard in a software package. For IT professionals implementing RESQML.</td>
</tr>
<tr>
<td>RESQML Use Case Guide</td>
<td>Lists RESQML use cases. For use by both petro-technical and IT professionals to understand the business/domain application of RESQML.</td>
</tr>
</tbody>
</table>
2. What are Use Cases and Why are they Important?

In support of RESQML development, implementation, and testing, and based on input and review from RESQML SIG (special interest group) members, a series of use cases has been developed. These use cases will continue to be developed and new ones created in support of enhancements and continued development of RESQML.

2.1 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.

2.2 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.
- Provides a way to prioritize RESQML development based on business/user needs.
- Helps developers to clearly define and develop RESQML.
- Provides a means for testing RESQML. Users should be able to accomplish the defined task or goal as described in the use case.

2.3 Use Case Notation

Some of the use cases include document flow diagrams, which show how the documents or objects are exchanged in the use case. The following figure explains the notations used in these diagrams.

![Use Case Notations](image)

Figure 1 Notation for use case document flow diagrams
3. **RESQML Use Cases**

This chapter describes an extended use and test case developed using actual data from the Alwyn North Field and the use cases developed based on oil company requirements by the RESQML SIG.

### 3.1 Extended Use Case and Testing with Alwyn North Field Data

The RESQML SIG has developed an extended use case and testing data sets that use actual field data from the Alwyn North field. This data, which was contributed by Total, is an ongoing invaluable resource for conducting further development and real-world testing to help ensure the rigor and reliability of RESQML.

Several companies that have implemented RESQML in their software packages then tested the use case with data from the Alwyn North field. Initially developed as a demonstration for an industry conference, the use case continues to serve as an ongoing test case and results have been presented at other events.

#### 3.1.1 Resources

A number of resources are available documenting this extended use case.

- **SPE Paper 143846-MS**: Using RESQML for Shared Earth Model Data Exchanges between Commercial Modelling Applications and In-House Developments, Demonstrated on Actual Subsurface Data, presented at SPE Digital Energy Conference and Exhibition, 19-21 April 2011, The Woodlands, Texas, USA. [link to abstract at onepetro.org.]
- **The 73rd EAGE Conference & Exhibition (Vienna ’11) Presentation**: Complete Reservoir Characterization Workflow Involving Several Vendors by Using the RESQML v1.0 Standard, [link to abstract at earthdoc.org].
- **Energistics Presentation**: Update presented to Energistics Western Europe Region Meeting participants on May 31, 2011. [link to presentation on energistics.org]

#### 3.1.2 Use Case Overview

Based on the Alwyn North field data, several of the companies participating in the RESQML SIG developed a use case that covers most of the workflow steps and exchange points that RESQML is working to support. The use case has these objectives:

- Transfer interpretation results from an oil company’s in-house interpretation package through a representative sub-surface workflow using software from several different vendors.
- At each step of the data transfer (especially the final one), ensure consistency with the original interpretation by transferring the data back into the originating software for a visual check.

The interpretation results (e.g., horizons, faults and well formation markers) from the oil company’s software, was first transferred as RESQML features to a structural application and then to diverse stratigraphic and reservoir applications from multiple vendors.

At the end of these operations, the final checked resqml document may be archived, either using pure XML or as XML and HDF (less storage space). (Recommendation: Store HDF libraries also at the same time to ensure that HDF data can be read in the future.)

Figure 5 shows the workflow for the use case data exchange. The next section provides details about the types of software packages and data exchange.
3.1.3 Use Case Workflow

The following steps describe the data exchanges in the use case shown in Figure 5.

For a detailed explanation of the data exchange (including the specific details of working with the Alwyn data, such as specific horizon and fault names, and detailed graphics of all models, etc.), see SPE Paper 14386-MS referenced above.

In each step, the user saves the work in the software’s native file format, then saves (or exports) a RESQML document, which is then read (or imported) by the next software package in the workflow. The scenario assumes a different user for each software package in the workflow.

Also, as a quality check, at each stage of the data exchange, the RESQML document is read by a QC application to compare and ensure consistency with the original interpretation document.

1. **Seismic Interpretation (D1).** Oil company in-house seismic interpretation package (D1) is used to identify major horizons and faults which are then saved as a RESQML document. RESQML metadata provides traceability for these interpretations. An EPSG CRS and an area of interest are also defined in the same RESQML document for geolocalization purpose. The exported RESQML file contains:
   - Picked horizons as 2D grid points.
   - Picked horizons as well formation markers along well trajectories to get a good fit between horizon picking and well markers.
   - Major faults represented as faults sticks.

2. **Structural Modeling (D2).** The user imports the D1 RESQML document and completes the following activities, after which the user saves the work as a RESQML document, D2:
   - Models the faults as triangulated meshes.
   - Using an interactive process, sets up the fault network.
   - Issues an initial horizon model, which, in the future, will be used to define the future fluid flow limits in the 3D reservoir grid.
3. **3D Grid Builder (D3).** The user imports the D2 RESQML document. At this point, the user needs to add more data that is not yet supported by the RESQML schema, including some new layers, and the main axis and lateral extension of the domain.

Package D3 exports a RESQML document, D3, containing explicit 3D grid representations with only geometry and new layers it created. Future versions of the RESQML schema will handle the reservoir framework information, stratigraphic organization, and geological environment that, for the purpose of this demonstration were handled with additional documentation.

Note: For the original demonstration of this use case, Package D2 also generated an alternate grid document, which demonstrates how different models can be realized concurrently from diverse vendor products. For more details, see the SPE paper.

4. **Static Property Population and Upscaling (D4).** The user imports the D3 RESQML document and uses the reservoir grid to define litho-stratigraphic unit limits, The user determines the lithotype model, defines and populates the litho units by static properties, and upscales them to the cells of the 3D grid created by the 3D grid builder, then exports the grids, upscaled static properties and blocked wells to RESQML document D4.

Today, wells must be imported independently using another method or product (WITSML, for example). Similar to Step 3, workarounds for objects not yet supported by the RESQML schema (for example, proper dissociation of property population in the litho-stratigraphic units and upscaling in 3D reservoir grid cells) were used.

5. **Dynamic Reservoir Fluid Flow Simulation (D5).** The user imports the D4 RESQML document (which contains the grid, upscaled static properties and blocked wells), combines with in-place fluid and production data from other sources, and creates a flow model and simulates for 10 years.

Output flow properties are generated monthly for 10 years, resulting in 41 different property kinds for a total of 1,249 discrete grid property value arrays, which are appended to the D4 document to create RESQML document, D5.

Note: The RESQML option to use HDF file format was used, so the property array data is actually stored in a compressed HDF file, which is about 30% smaller than the uncompressed raw simulation data file. (For more information about HDF, see the RESQML Usage Guide.)

### 3.2 Standard Use Cases

The following table lists RESQML use cases, which were originally developed based on oil company requirements for development of RESQML v1.0. However, the RESQML SIG continues to evolve these use cases to support continued enhancement and development of RESQML.

<table>
<thead>
<tr>
<th>RESQML Use Cases</th>
<th>Link to Page with More Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>Title and Description</td>
</tr>
<tr>
<td>1</td>
<td><strong>Export/Import Consistent Model from Package A to Package B</strong> I want to transfer from Package A to Package B the 3D grid along with all its properties and input data used to create it (for example, horizons and faults) in one exchange. When I open the model in Package B, it should be “geometrically consistent” (regardless of elevation or depth) with the model in Package A. “Geometrically consistent” refers to structures retaining the same size, shape, orientation and position/spatial relationships. Example of how this might be used: between partners in a joint venture (JV).</td>
</tr>
<tr>
<td>2</td>
<td><strong>Move Features, Such as Horizons or Faults, Independently (without a grid)</strong> I want to transfer objects independently, such as horizons or faults, without a 3D grid.</td>
</tr>
<tr>
<td>No.</td>
<td>Title and Description</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td><strong>Merging Data from Two or More Packages into Another Package</strong>&lt;br&gt; 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.&lt;br&gt;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 a well with a subsea surface location, and also understand the different surface locations with respect to one another.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Traceability</strong>&lt;br&gt;When I open a RESQML model, I want to be able to determine some basic information about how it was created. For example: 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?</td>
</tr>
<tr>
<td>5</td>
<td><strong>Move RESQML Model</strong>&lt;br&gt;I want to easily transfer a RESQML model among users, networks or file systems for use in the same package, for example, from a European-based office/network to a US-based office/network.&lt;br&gt;Ideally, I would like to move it as a “single thing” or single file. That “single thing” may actually be made up of multiple files, but I don’t want to have to know that or see that.&lt;br&gt;If a “single thing” is not possible, then as few files as possible, and there must be a single, easily distinguishable “main file” or entry point that when opened correctly associates or integrates other files.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Model Import to Flow Simulator</strong> (See also Use Case 7)&lt;br&gt;A reservoir engineer (RE) wants to understand the fluid flow, pressure and energy distribution in a reservoir as it evolves with time. As a first step, the RE wants to incorporate a geological property model, developed by a geologist using geological property modeling software, to a separate dynamic flow simulation software package.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Model Export from Flow Simulator</strong> (See also Use Case 6)&lt;br&gt;Having completed a dynamic flow history match in a dynamic flow simulation software (Use Case 7), a reservoir engineer (RE) wants to export forecast/predicted reservoir saturations back to a geologist and the geological property modeling software for well planning purposes and model QC/validation.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Horizon Attribute Extraction</strong>&lt;br&gt;Take a horizon interpretation and extract an attribute (e.g. Curvature) from a seismic volume, at each node on the surface.&lt;br&gt;Transfer a horizon from an interpretation software package into an attribute calculation software package, extract attributes onto the horizon and then move the horizon back into the interpretation software package for display.</td>
</tr>
<tr>
<td>9</td>
<td><strong>Transfer of Blocked Wells</strong>&lt;br&gt;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</td>
</tr>
<tr>
<td>No.</td>
<td>Title and Description</td>
</tr>
<tr>
<td>-----</td>
<td>----------------------</td>
</tr>
</tbody>
</table>
| 10  | **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. | 26 |
Use Case 1: Exporting/Importing Consistent Model from Package A to Package B

I want to transfer from Package A to Package B the 3D grid along with all its properties and input data used to create it (for example, horizons and faults) in one exchange.

When I open the model in Package B, it should be “geometrically consistent” (regardless of elevation or depth) with the model in Package A.

“Geometrically consistent” refers to structures retaining the same size, shape, orientation and position/spatial relationships. Example of how this might be used: between partners in a joint venture (JV).
Use Case 2: Move Features, Such as Horizons or Faults, Independently (without a grid)

I want to transfer objects independently, such as horizons or faults, without a 3D grid.

(The previous RESCUE format required that features, such as horizons or faults, be transferred with an associated 3D grid.)

Figure 4 Use Case 2: Move Features Independently
Use Case 3: 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.

Figure 5 Use Case 3: Merging Data
Use Case 4: Traceability

When I open a RESQML model, I want to be able to determine some basic information about how it was created. For example: 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?

Traceability in RESQML: GUIDs and Dublin Core Elements

Traceability capabilities in RESQML are accomplished by two main technologies: global unique identifiers and Dublin Core elements.

- Global unique identifiers (GUID) give unique IDs to each RESQML document, and each data object in a document, including every grid, horizon and fault.
- Dublin Core elements are library cataloguing standard data elements, a subset of which has been adapted for use in RESQML. The elements are used to capture key data (as described in the use case above) for any item with a GUID (documents and data-objects).

For information about how Dublin Core elements have been tailored and implemented for RESQML, see the RESQML Usage Guide.

For more information about the Dublin Core Metadata Initiative, see www.dcmi.org.

Using Dublin Core to Help Manage Different Versions of the Same Representation

Dublin Core information can also be used during the modeling process itself—when RESQML models are frequently updated and exchanged between software packages—to help users determine the latest version of a model.

Overview

The current version of the RESQML data model is not designed to exchange partial information between models; instead, it focuses on maintaining the consistency of the whole model.

In this use case, Software A and B represent applications used in consecutive stages of sub-surface workflow. When new information becomes available for a given model in Software A, users of Software B will typically have to reload a new RESQML document containing the entire updated model. If users of Software B already modified the model, they will have to either:

- Discard their modified model and manually add their modifications to the updated/newly imported model, or
- Manually resolve the conflict between these two versions of the same model.

How it Works

Dublin Core information can help tremendously in the process of conflict resolution. The two key pieces of information available for each individual representation, property and other entities are:

- The unique identifier (GUID) used to indicate if an entity present in a RESQML file is already available in the current session of Software B, indicating the potential conflict between two versions of the same entity.
- The "modified" time inside the Dublin Core elements (or the "created" time if the entity has not yet been modified) indicating that the version in New File A is more recent than the edition (or version) of the entity in the Software B model.

After a conflict has been detected, Software B can extract other Dublin Core information to help users select one version, either manually or following a common strategy (for example, always overwrite, ignore, keep the most recent one, import as a copy, and so forth).

In the above process, Software B is “kept live” between the initial import of the model and the import after modification. However the same resolution process can also occur in two different sessions of Software B, for example, if Software B saves the model in another RESQML document or in its own persistence.
mechanism (e.g., database). IMPORTANT! For this approach to work, the persistence mechanism MUST retains the GUID and Dublin Core information.

**Conflict Resolution: Frequent Exchange in One Session**

Figure 6 shows an example workflow for resolving data conflicts resulting from frequent update and exchange of data models during one session.

![Conflict Resolution: Frequent Exchange in One Session](image)

**Figure 6 Conflict resolution: frequent exchange in a single session.**

The scenario is:

2. Package B imports document Resqml1, modifies H
3. Package B exports document Resqml2: F + H*, (where H* = edited H)
4. When Package A imports document Resqml2, it should have enough information to know:
   - F has not been modified, and does not have to be reloaded
   - H* is an edited version of H; there is enough information to let a user decide what to how to reconcile the two versions.
Conflict Resolution: Exchange in Multiple Sessions with Persistent Data Store

Figure 7 shows an example workflow for resolving data conflicts resulting from updates and exchange during multiple sessions with persistent data store.

Figure 7 Conflict resolution: exchange with alternative persistence.

The scenario is similar to the single-session scenario:

1. Package A exports document resqml1,
2. Package A saves (persists) H and F in alternative storage (for example, a database), then Package A Session1 is closed.
3. Package B imports resqml1 and edits are made to H, creating H*.
5. Package A begins a new session, Session 2, and reloads H and F from alternative storage.
6. When Package A imports resqml 2 into Session 2, Session2 has enough information to know without using resqml1 that:
   - F has not been modified, and does not have to be reloaded.
   - H* is an edited version of H; there is enough information to let a user decide how to reconcile the two versions.
Conflict Resolution Example
Figure 8 shows how RESQML with Dublin Core elements can be implemented in a software package to help resolve data conflicts that arise from frequent update and exchange of data models—a common occurrence in the sub-surface workflow.

Figure 8 Conflict resolution example of how RESQML and Dublin Core elements can be used in software packages to resolve data conflicts.

When the same object is present in a RESQML file and session, a user can decide what to do based on the differences (red text):

- Replace the content of the session by the content of the file.
- Create a new copy with the content of the file.
- Do not load the content of the file.
- Load only the new properties.
Use Case 5: Move RESQML Model

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Move RESQML Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0+</td>
</tr>
</tbody>
</table>
| Goal            | • An entire RESQML model (all files that make up the model) can be moved, for example, from one network drive/office to another network drive/office, as a “single thing,” for example, a single folder or zip file.  
• The recipient can open the model in its new location using the same software with which it was created and view the same complete model. Should be able to read and write the RESQML file.  
• There should be no conflicts with any existing files in the destination location. |
| Summary Description | Users need to be able to move RESQML models between office locations or networks as a “single thing,” or a small number of files and be sure that all necessary files are transferred.  
The user does not want to know what all the files are, only that all necessary files have been moved so that the recipient can open the model using the same software with which it was created and view the same complete model as in the original location. |
| Actors          | Two or more users (User A and User B) from any disciplines, such as geologists or reservoir engineers. |
| Triggers        | A user needs for a colleague or partner to view the same model, but they do not have access to a common network location. |
| Pre-conditions  | • An existing RESQML-compliant model.  
• A means to transfer the file (e.g., copy across a network, ftp, email) |
| Primary or Typical Scenario | A user needs for a colleague or partner to view the same model, but they do not have access to a common network location.  
The users may be co-workers in the same company but different locations, or partners in different company. |
| Alternative Scenarios |  |
| Post-conditions | • An entire RESQML model (all files that make up the model) can be moved, for example, from one network drive/office to another network drive/office, as a “single thing,” for example, a single folder or zip file.  
• It gives me an error message to indicate if/what data is missing  
• The recipient should be able to open the model in its new location and view the same complete model.  
• Should work regardless of operating system  
• See Notes below |
<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Move RESQML Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business Rules</td>
<td>Ideally, the model should:</td>
</tr>
<tr>
<td></td>
<td>• Have as few files as possible</td>
</tr>
<tr>
<td></td>
<td>• If folders/hierarchy required, you must be able to rename the top folder and still maintain a consistent model, perhaps with a RESQML file extension.</td>
</tr>
<tr>
<td></td>
<td>• Files should be as small as possible.</td>
</tr>
<tr>
<td></td>
<td>• No absolute path names needed</td>
</tr>
<tr>
<td></td>
<td>• No Unix/Windows file name clashes</td>
</tr>
<tr>
<td></td>
<td>• A designated extension and/or icon so that users can easily identify the file and launch it/open it.</td>
</tr>
</tbody>
</table>

**Definitions**

**Moving Models in RESQML**

In RESQML’s predecessor standard, RESCUE, a model was composed of many small files, so trying to move it—and ensuring you had all of the files or knowing which one was the main file to be opened—often posed some serious challenges.

In RESQML v1.0 a RESQML document consists of a maximum of two files: one XML file (extension = .resqml) and an optional HDF5 file (extension = .h5). So moving a model means moving one or two files, by traditional methods such as copying them or sending them as attachments to an email.

Currently, wells must be moved separately either using WITSML or a user’s current method. Integration of detailed well objects (wells, wellbores, trajectories, etc.) will be addressed in a future version of RESQML by leveraging the WITSML specification. RESQML has been developed to be compatible with WITSML.

**Moving Models for the Alwyn North Use/Test Case**

In the development and testing of the extended use case based on Alwyn North field data (see Section 3.1, page 3), the members of the RESQML SIG had ample opportunity to test the concept of “moving models.”

The data exchange and testing occurred between five different companies in North America and Europe, using the Energistics’ FTP site, where testers would upload and download data models—each consisting of one resqml file and one HDF5 file—for exchange.

For each of the exchanges, each participant did the following:

- Downloaded the XML and HDF5 file,
- Performed the tasks required for that part of the workflow then saved the files with new names (though new names were not required),
- Uploaded the two files back to the FTP site, for the next application in the workflow to use.
# Use Case 6: Model Import to Flow Simulator

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>RESQML Model Import to Flow Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>1.0+</td>
</tr>
<tr>
<td>Goal</td>
<td>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.</td>
</tr>
<tr>
<td>Summary Description</td>
<td>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.</td>
</tr>
</tbody>
</table>
| Actors        | Reservoir Engineer  
Geologist  
Geological Modeling Software  
Flow Simulation Software |
| Triggers      | An asset team needs to either  
(a) create a field development plan,  
(b) compare different possible locations to drill new wells in a reservoir, or  
(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.  
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.  
2. The geologist exports the geological model to a RESQML model.  
3. The geologist informs the reservoir engineer of the file system location of the RESQML model.  
4. The reservoir engineer starts the flow simulation software.  
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.  
6. From the RESQML model, the flow simulation software should obtain the following:  
(a) a structured grid of hexahedral cells (corner point) stored in some efficient format |
<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>RESQML Model Import to Flow Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(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.</td>
</tr>
<tr>
<td></td>
<td>(c) the location of well trajectories such that their intersection with grid cells may be calculated</td>
</tr>
<tr>
<td></td>
<td>(d) the location of fault surfaces with respect to grid cell faces, and other discontinuities in the grid</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>7. The flow simulation software may also find the following information useful if it included in the RESQML model.</td>
</tr>
<tr>
<td></td>
<td>(a) properties (such as seal/non-seal or transmissibility multipliers) on fault surfaces</td>
</tr>
<tr>
<td></td>
<td>(b) perforation intervals, in measured depth, associated with trajectories</td>
</tr>
<tr>
<td></td>
<td>(c) well logs, in measured depth, associated with well trajectories</td>
</tr>
<tr>
<td></td>
<td>(d) geological unit names associated with grid K layers</td>
</tr>
<tr>
<td></td>
<td>(e) 2d surfaces, such as horizons, seabed surface, or land surface.</td>
</tr>
<tr>
<td></td>
<td>8. If the set of cell properties do not have &quot;well known&quot; 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.</td>
</tr>
</tbody>
</table>

| Alternative Scenarios | 1. An error is encountered while exporting the RESQML model |
|                      | 2. An error is encountered while importing the RESQML model |
|                      | 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. |
|                      | 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 RESQML 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 |
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.

Figure 9 Use Case 6: Model Import to Flow Simulator

Use Case 6: Model Import to Flow Simulator

A creates a 3D grid and associated faults which B reads in. B creates new properties for that grid (see Use Case 7).
## Use Case 7: Model Export from Flow Simulator

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>RESQML Model Export From Flow Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Version</strong></td>
<td>1.0+</td>
</tr>
<tr>
<td><strong>Goal</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Summary Description</strong></td>
<td>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.</td>
</tr>
</tbody>
</table>
| **Actors**    | Reservoir Engineer  
Geologist  
Geological Modeling Software  
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.  
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.  
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.  
a) export an entire, new, RESQML model, or  
b) append the properties to an existing RESQML model (the one previously imported and forming the static model used in the history match.  
3. The geologist starts the geological modeling software and opens the RESQML model, then imports the needed parts of the model.  
4. The geologist uses the geological modeling software, and the imported fluid saturations and pressures to plan a new well. |
<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>RESQML Model Export From Flow Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alternative Scenarios</td>
<td>1. An error is encountered while exporting the RESQML model 2. An error is encountered while importing the RESQML model</td>
</tr>
<tr>
<td>Post-conditions</td>
<td>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.</td>
</tr>
<tr>
<td>Business Rules</td>
<td>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.</td>
</tr>
<tr>
<td>Notes</td>
<td>Steps (2.a) and (2.b) could be written up as separate use cases.</td>
</tr>
</tbody>
</table>
| 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.  
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. |

**Use Case 7: Model Export from Flow Simulator**

Use Case 7: Model Export from Flow Simulator

Figure 10 Use Case 7: Model Export from Flow Simulator

B creates time-recurrent properties for 3D grid (which was previously imported from Package A, see Use Case 6) and exports those properties back to A.
Use Case 8: Horizon Attribute Extraction

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Horizon Attribute Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>V1.0+</td>
</tr>
</tbody>
</table>

**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**
1. An interpreted horizon.
2. The interpretation and attribute extraction application must be capable of exporting and importing RESQML horizon and horizon property models.

**Primary or Typical Scenario**
1. The geophysicist interprets a horizon in the interpretation application.
2. The geophysicist exports the horizon into a RESQML model.
3. The geophysicist (same or different one) loads the exported horizon into the attribute extraction application.
4. The geophysicist extracts the required attributes from a seismic volume onto the imported horizon – creating a new horizon property.
5. The geophysicist exports the newly created horizon property into a RESQML model.
6. The newly created horizon property is imported back into the original interpretation application and associated with the initial horizon.
7. The new property is draped over the Horizon and a drill location is determined.

**Alternative Scenarios**
1. 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**
1. 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.
2. 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**
Use Case 8: Horizon Attribute Extraction

Figure 11 Use Case 8: Horizon Attribute Extraction

Reads horizons (exported from A) and produces attribute extraction (P) for that horizon. Then attributes are transferred back into interpretation package (A).
Use Case 9: Transfer of Blocked Wells

<table>
<thead>
<tr>
<th>Use Case Name</th>
<th>Transfer of blocked wells</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>V1.1+</td>
</tr>
<tr>
<td>Author</td>
<td>RESQML/Laurent Deny</td>
</tr>
</tbody>
</table>

**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.

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.

The goal is for the writing software to write blocked well representations that allows the following goals:

**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.

**Goal 2:** The reading software will be able to associate the blocked well to an actual or pseudo-well.

**Goal 3:** The reading software will be able to associate multiple property values to each of these individual grid cells.

**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.

Reader: Application which imports the RESQML Grid and blocked wells, and is expected to use this information for the goals listed above.

**Triggers**
Trigger 1: Model is exported (writer)
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.

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. |
| 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. |
| 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: |
| Grid, Blockedwells, Properties |
| Additional objects: |
| Well trajectories |
Use Case 10: 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.