*User manual*

# Implementing a Reportnet/SEIS node

## A manual for the member states


**Prepared by**

**Søren Roug**



**June 2009**

**Version 0.3**

European Environment Agency

# Version management

| No | Date | Changes | Author |
|---|---|---|---|
| 0.1 | 25.05.2008 | Initial version | Søren Roug |
| 0.2 | 16.06.2008 | Specification of types | Søren Roug |
| 0.3 | 26.06.2009 | Added advanced concepts and SOER C proposal | Søren Roug |

# Contents

1. INTRODUCTION ..... 4

2. IMPLEMENTING REPORTNET/SEIS ..... 5

2.1. Type 1 Reportnet/SEIS node ..... 5
2.2. Type 2 Reportnet/SEIS node ..... 7
2.3. Type 3 Reportnet/SEIS node ..... 10
2.4. Type 4 Reportnet/SEIS node ..... 11

3. ADVANCED CONCEPTS ..... 12

3.1. Object types ..... 13
3.2. More types ..... 14
3.3. Super-types ..... 15
3.4. Rolling your own objects ..... 16

4. SOER C REPORTING ..... 17

4.1. Proposal to implement option 2 as Reportnet/SEIS ..... 17
4.2. What could it look like? ..... 18
4.3. Vocabularies ..... 19

5. FURTHER READING ..... 19

# 1. Introduction

SEIS is about getting access to and sharing data and information that is handled in information systems and in organisations all over Europe, both at local, intraregional, national as well as the European level. Its guiding principles are:

- information is managed as close as possible to its source
- information is provided once and shared with others for many purposes
- information should be readily accessible to end-users

What does this mean for Reportnet? Let's first examine the core premises of Reportnet:

- Data reporting has a slow periodical cycle. Reportnet is only set up to handle deliveries that happens once a month or less often than that. Most dataflows are yearly.
- Dataflows have deadlines. The legal instruments set a reporting cycle with deadlines. It must therefore be possible to determine whether the reporter met the deadline or not.
- History is important. Once a delivery has been made, it must always be possible in the future to fetch the same historical data on the same URL[1].
- Preparation of data is a manual process, hence QA work after delivery can also be a manual process.
- Few dataflows have GIS data. Most are information on national level. E.g. compliance reporting and statistics.
- Due to historical reasons, deliveries come in many formats such as spreadsheets or other types of office documents.

This basically means that Reportnet explicitly excludes always current data, as you'd see in map-services, and it also excludes light-weight frequent deliveries, as you'd see in ground level ozone.

Given these premises above, Reportnet has built a number of services around the process of delivering. These are:

- Notifications can be sent out by email when a delivery is made.
- There are tools that can convert between different formats, especially XML can be converted to various office formats for post-processing.
- When the file format is XML there are tools to automatise some of the QA work.
- The requesters can provide feedback to the reporters via a bulletin board mechanism.
- The requesters can ask for a redelivery.

[1] Especially DG-ENV is adamant about persistence. Once a delivery has been made it may never be deleted or changed.

# 2. Implementing Reportnet/SEIS

At EEA we have discussed what effect SEIS would have on Reportnet with regards to:

a) How to find the datasets amongst all the other documents on the Internet

b) How to describe what the dataset contains so it is possible to decide before download whether it is relevant or not.

c) How to ensure it is not obsolete. E.g. a newer edition is available

d) How to understand what format the dataset is in, and

e) How to convert the dataset to the needed format.

... and we drew up four scenarios that progressively tightened the requirements for the providers. The following sections will describe each type in order and what the provider will have to do to implement it.

Before we go into the scenarios, we need to establish the following definitions:

1. A *delivery envelope* is a container that holds the *set of files* that constitutes the delivery. The metadata is on the envelope; not the individual files.
2. If a delivery/file is tagged with the triple: ROD obligation, country, reporting period, it is easy to handle, as it constitutes an *answer* to a reporting obligation. But the downside is that many datasets from NGOs don't have obligations in ROD.
3. QA reports - including the automatic - are the data requester's feedback on the deliveries. Therefore it shouldn't be the providers' task to develop automatic QA scripts for their own reports.
4. The Content Registry (CR) mentioned below is an object oriented search engine.

## 2.1. Type 1 Reportnet/SEIS node

This type is the simplest possible. It is a normal website containing datasets that might be interesting for the production of environmental information.

The type is relevant if we expect to find and use information produced by NGOs and individuals, who have no incentives nor means to implement the requirements made in the next types.

These websites are unharvestable by CR. The first problem is that CR can't crawl the entire Internet to find the website. The second is that even if CR is given the address of the website, the system can't get information from the content of the files, as datasets tend to be only numbers.

We must assume that CR knows nothing about the data until someone adds metadata. Therefore;

– *Users* find the file with Google, surfing the web or the *owner* of the dataset "registers" it at EEA.

EEA has set up a website called http://qaw.eionet.europa.eu (short for Quality Assessment Workbench) where people can enter additional metadata for a dataset they have found. It is possible to enter the three ROD parameters (obligation, country, reporting period) and this will then make the delivery visible on ROD. QAW even provides a plugin for the webbrowser, so all you have to do is view the dataset in your browser and then press a button to register it with QAW.

It does have a few requirements:

1. The file must have a persistent URL because registration is a form of bookmarking to make it possible to fetch the file at any time in the future.
2. The file must have a last-changed timestamp. This is because other Reportnet functionality needs it. For instance, obligations have deadlines. If it looks like the file constantly changes, then it looks like the country delivered after the deadline.

### 2.1.1. Type 1 example

Asking the users to take the trouble to add metadata may sound complex, but Reportnet already does it. There is a mechanism on CDR, where the data provider can add the necessary metadata to a file stored on his own system. It is called adding a referral.

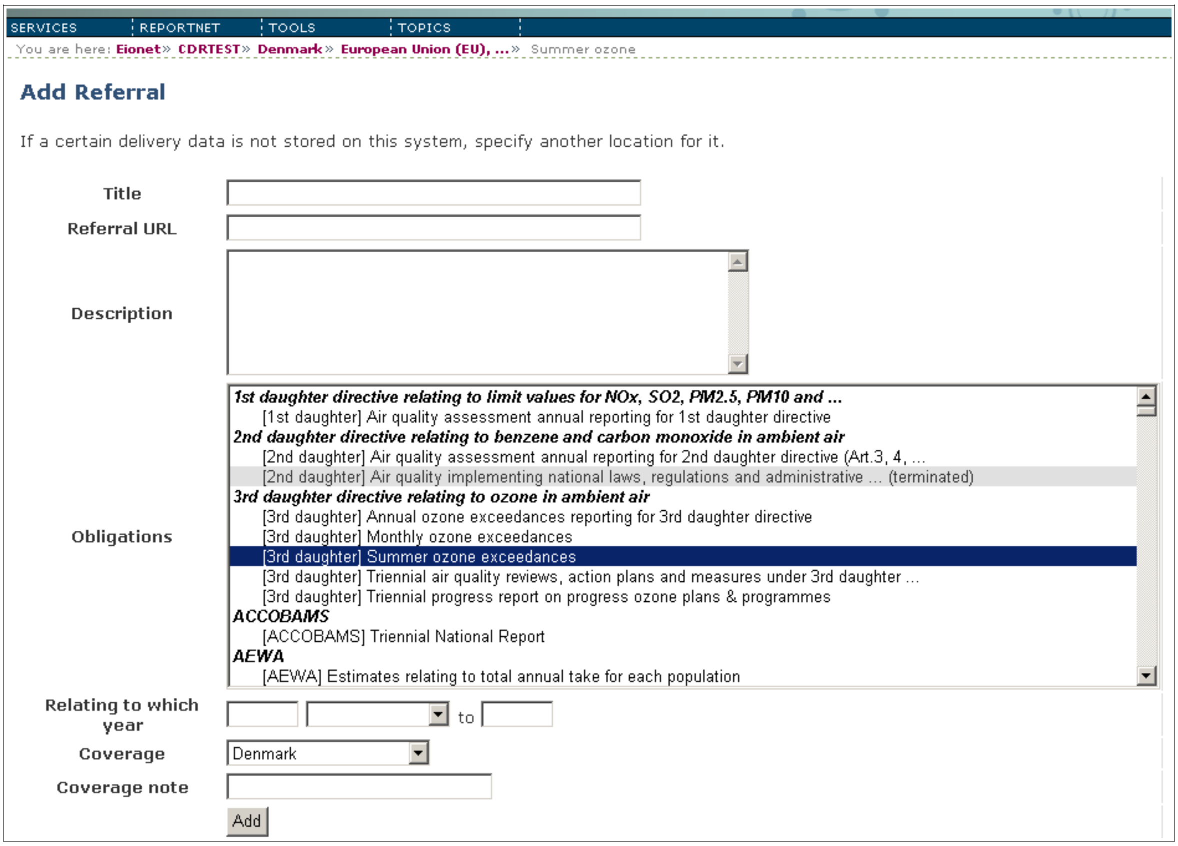



*Illustration 1: Filling out metadata for a referral to a SEIS dataset*

As you can see, the metadata being asked for is prearranged with the requesters. Obviously on CDR it can only be used for deliveries related to an obligation and for users at national focal points. On QAW we are able to register a larger selection of files.

## 2.2. Type 2 Reportnet/SEIS node

Type 2 is set up to match a situation, where EEA can make some requirements to the information provider, but without requiring a total reorganisation of an existing website. One such example would be Norway's www.environment.no.

- ✔ It is a static website
- ✔ The metadata is available for CR
- ✔ Not only ROD deliveries

The main difference is that the website is a *Reportnet/SEIS* node. This means the node is collaborating with Reportnet. The way the collaboration works is, that at a fixed URL, the node provides a manifest file of the relevant files existing on its own website. This file must be downloadable over the Internet and will be periodically harvested by Reportnet's search engine. When a file occurs on the manifest, it is then discovered by Reportnet. The manifest doesn't have to be a static file. It can be a script that extracts data from a database.

### 2.2.1. Making a manifest

A manifest file is very close in concept to an RSS[2] feed. The main difference is that RSS feeds always contain information about announcements, whereas a manifest can contain information about any type of data object. The manifest must be in RDF[3] format. RDF is an XML format where every element has a global meaning.

A declaration of one delivery with one file in it looks like this:

```
<?xml version="1.0" encoding="utf-8" ?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:cr="http://cr.eionet.europa.eu/ontologies/contreg.rdf#"
  xmlns:rod="http://rod.eionet.europa.eu/schema.rdf#">

  <rod:Delivery rdf:about="http://cdr.eionet.europa.eu/hu/colrzxujg/envrzxuuw">
    <dc:title>Bathing water report, 2006</dc:title>
    <rod:released>2007-11-15T16:17:27Z</rod:released>
    <rod:locality rdf:resource="http://rod.eionet.eu.int/spatial/17"/>
    <rod:period>2006P1Y</rod:period>
    <rod:obligation rdf:resource="http://rod.eionet.eu.int/obligations/21"/>
    <rod:hasFile rdf:resource="http://www.miljostatus.no/datakat/18548?xml"/>
  </rod:Delivery>

  <rod:File rdf:about="http://www.miljostatus.no/datakat/18548?xml">
    <dc:title>Locations of BWD stations</dc:title>
    <cr:mediaType>text/xml</cr:mediaType>
    <cr:xmlSchema>http://89.191.28.227/InternetReportServer?%2fMiljoInfoRapporter
%2fTabellMatrise_SL</cr:xmlSchema>
  </rod:File>
</rdf:RDF>
```

There are a few special elements:

[2] See http://en.wikipedia.org/wiki/RSS.

[3] RSS 1.0 is actually RDF, but since a generic RDF parser is more difficult to develop than a specific RSS parser, the community has in later versions of RSS stopped using the RDF principles for a simpler format.

- The <rod:Delivery> declares that the object described by the meta data is a Reportnet delivery. In the Advanced concepts chapter we'll discuss other object types.
- The <rod:released> is the release date of the envelope. It is used to verify that the country delivered before the obligation's deadline.
- <rod:locality> is a reference to the spatial coverage of the delivery. Reportnet doesn't use country codes. This is mainly because it was expected that regional organisations would deliver reports for a particular area such as the Black Sea. Therefore Reportnet uses a different code list, but most codes correspond to a country. The spatial code is a URL-like string that is globally unique, ensuring that there won't be clashes with national codes.
- <rod:period> is the period the delivery covers. You can e.g. deliver 2005 data in 2008. The format is the ISO 8601 date format. If the reporting is a one-off, then enter the date of the deadline. If it is periodical enter the date of the first day in the period, and the system will be able to look up in ROD how many months it covers. Otherwise use the format for time intervals. A period from 1 April 2008 and three months is then: 2008-04-01P3M.
- <rod:obligation> is a reference to the obligation identifier. It is a globally unique URL ensuring you can create your own national obligations without clashes. You can get the list of ROD obligations as RSS, RDF or via XML-RPC. Read this document to see how: [http://rod.eionet.europa.eu/text.jsv?mode=R](http://rod.eionet.europa.eu/text.jsv?mode=R).
- <rod:hasFile> tells CR that the URL pointed to is a file that is part of the delivery. You can have more than one <rod:hasFile> element inside the <rod:Delivery> element.

  Placing your dataset on your own website does not absolve you from providing it in the format agreed with the requester. If your system generates the dataset on-demand, <rod:hasFile> must point to the generation that delivers the agreed format for the declared period.

- <rod:File> provides metadata about the file contained in the delivery. Since it is a separate object, it is listed one the same level as <rod:Delivery>. The declaration registers the file with QAW, and CR will periodically check for its continued existence. CR might also attempt a download for the purpose of quality assessment.

  Let's take a closer look at the declaration:

```
  <rod:File rdf:about="http://www.miljostatus.no/datakat/18548?xml">
    <dc:title>Locations of BWD stations</dc:title>
    <cr:mediaType>text/xml</cr:mediaType>
    <cr:xmlSchema>http://89.191.28.227/InternetReportServer?%2fMiljoInfoRapporter
%2fTabellMatrise_SL</cr:xmlSchema>
  </rod:File>
```

None of these in the example above are required. They just make it easier to find the file in the search engine for a user.

- <dc:title> provides a title for the file. You can use all the Dublin Core elements, but title is the most useful. It is a good idea if there is no title, to provide the name of the file in its place. Filenames often have meaningful values.
- <cr:mediaType> is a hint to the user about what format the file is in.
- <cr:xmlSchema> is the XML schema identifier or DTD System identifer of the XML file. If it is not provided CR will attempt to get it from inside the file.

### 2.2.2. Type 2 example – how to link Norway's data to SEIS

To make the concept more clear we offer an example of the requirements Reportnet would impose on a member country in type 2 – which is judged to be closest to a normal SEIS node.

Johnny Auestad gave us a pointer to [http://www.environment.no/](http://www.environment.no/). If you go to this website, you will see it is full of data on Norway. How do we make this data available for SEIS? Discoverability is the key.

While the Norwegian website is pretty easy to navigate, if you want data for all of Europe, you have to look at 30+ websites with different navigational structure. So much time wasted that you'll never get to do the actual work.

For the example, we'll use the Contaminated soil data. Somewhere in the middle under "Hazardous chemicals", there is a Contaminated soil link. Click on it. Then click on Contaminated soil on the right side under Maps and data. At the bottom of this page there is a See meta-data (in Norwegian) As you can see, the Norwegians have done a great job filling out the meta-data, and there is even a link to EEA's Reporting obligation listed as: Rapporteringsforpliktelse. Unfortunately, all the metadata is not easy to find in Google, as it is shown for a *human* audience.

What we need is a machine-readable description of the meta-data for CR. It could look like this:

```
<rod:Delivery rdf:about="http://www.miljostatus.no/datakat/18548">
  <dc:title>Miljøgifter : Forurenset grunn</dc:title>
  <dc:language>no</dc:language>
  <rod:released>2008-05-14T08:49:29Z</rod:released>
  <rod:locality rdf:resource="http://rod.eionet.europa.eu/spatial/28"/>
  <rod:period>1989/2007</rod:period>
  <rod:obligation rdf:resource="http://rod.eionet.europa.eu/obligations/33"/>
  <rod:hasFile rdf:resource="http://62.92.43.137/ReportS?...Format=XML"/>
</rod:Delivery>

<rod:File rdf:about="http://62.92.43.137/ReportS?...Format=XML">
  <cr:mediaType>text/xml</cr:mediaType>
  <dcterms:isFormatOf rdf:resource="http://www.miljostatus.no/files/1698"/>
</rod:File>

<rod:File rdf:about="http://62.92.43.137/InternetReportS?...Format=Excel">
  <cr:mediaType>application/ms-excel</cr:mediaType>
  <dcterms:isFormatOf rdf:resource="http://www.miljostatus.no/files/1698"/>
</rod:File>

<rod:File rdf:about="http://62.92.43.137/InternetReportS?...Format=CSV">
  <cr:mediaType>text/plain</cr:mediaType>
  <dcterms:isFormatOf rdf:resource="http://www.miljostatus.no/files/1698"/>
</rod:File>
```

Norway's system doesn't really have the concept of an envelope, i.e. a folder on a website that shows the delivery meta data in a human readable way containing the files of the delivery. So we must fake it. CR needs a globally unique URL as the key to attach the metadata to. We choose "http://www.miljostatus.no/datakat/" plus an internal database key of 18548.

The data is provided in three formats. Excel, CSV and XML. It is possible to provide that information to CR by using the <dcterms:isFormatOf> element and point to the original format. If this is not available, then point to a non-existent unique URL. The user can then use CR to discover that the dataset is available in three formats and download the one that fits the purpose best.

What to do in case there is no ROD obligation to link to the delivery? The recommended way is to create a national obligation all deliveries will refer to. That way it is possible to find all older and newer deliveries of the same sort. If this is not possible, then do make the data available in the manifest anyway but without the <rod:obligation>. The QAW application makes it possible to categorise it in several other ways, making it possible for someone to find all datasets related to e.g. "soil".

## 2.3. Type 3 Reportnet/SEIS node

Type 3 is where the SEIS node is a static website, but the participant wants to use some of the other Reportnet modules for further integration.

### 2.3.1. Handling on-demand automatic QA

The QA service is a mechanism the reporters can use to test the delivery before they actually deliver or as a service for the website visitor to check the quality of the data. It is a set of rules defined by the data requester and deployed as a central web service at EEA. It only works for XML files. You call the QA service with the schema identifier of the XML file to check if there are any QA scripts for that file format. There can be more than one.

For each script you can supply a link the user can click on to send the file to the QA service for analysis. The result is a webpage body which can be shown to the user.

See the QueryService User Manual for more information.

### 2.3.2. Implementing persistent automatic QA

To avoid generating too heavy a load on EEA's QA service we recommend that when you make the final version of your data available on your website, you also call the QA service and save the result as a webpage linked from an overview page of your delivery. Because if the dataset doesn't change, it can be assumed the QA assessment won't change either.

If you do so, you should link to this page in your manifest file when providing meta data for the file:

```
<rod:File rdf:about="http://62.92.43.137/InternetReportS?...Format=XML">
  <cr:mediaType>text/xml</cr:mediaType>
  <cr:hasFeedback rdf:resource="http://62.92.43.137/InternetReportS?...q=QA"/>
</rod:File>
```

If the QAW application doesn't see the <cr:hasFeedback> property on a file it has been configured to do QA for, then it will do the QA, store the feedback report at its own location and add the <cr:hasFeedback> property to CR's database.

You can provide meta data on the feedback report, such as title, who wrote it and so on. That also is entered into the manifest:

```
<cr:Feedback rdf:about="http://62.92.43.137/InternetReportS?...q=QA">
  <dc:title>Review of Contaminated soil report</dc:title>
  <dc:creator>John Shuttleworth</dc:creator>
</cr:Feedback>
```

### 2.3.3. Sending a notification

When you make a delivery available for download (or make it unavailable), you can send a notification to the Unified Notification Service. Anybody subscribed to the channel will then receive the notification. The effect is to speed up the workflow. The data requesters will discover it immediately rather than when the search engines see it.

To set up notification, see the UNS User Guide. The notification can be sent with XML-RPC.

### 2.3.4. Calling the conversion service

Reportnet can help with conversion of XML to more user-friendly formats. The available conversions are keyed to the schema or DTD system id. See the XMLCONV User Manual for more information.

## 2.4. Type 4 Reportnet/SEIS node

Type 4 is intended to be a narrow extension of the existing Reportnet system. It is an expansion with distributed repositories with the same capabilities of CDR. What are the distinguishing features?

### 2.4.1. Handling manual QA feedback

CDR has a bulletin board system to facilitate communication between the data requester and the data provider. The purpose is to provide review of the data and optionally reject or accept the delivery. This is an advanced feature, that we can't expect all Reportnet/SEIS nodes to implement. *If* the Reportnet/SEIS node doesn't have its own feedback functionality, Reportnet has a fallback mechanism. The reviewer will store his feedback on QAW, and via CR make a reference between the delivery and the feedback. QAW will also send a notification via UNS when a new feedback is posted.

If you want to implement the feedback feature on your own Reportnet/SEIS node, you create a webform the reviewer can write his text in. The form should allow attachment of files. These files are not part of the delivery, but of the feedback.

Optionally you can provide a button for the reviewer that will revoke the release. It should make the delivery unavailable for unauthenticated visitors and remove the record from the manifest file.

Since the reviewer is typically an EEA or topic centre employee, it is recommended to set up authentication to use the Eionet site directory.

When a feedback has been posted for the delivery or a file in the delivery, then the manifest must reflect this. There is no difference between an automatic and a manual assessment. The same RDF element is used:

```
<rod:Delivery rdf:about="http://www.miljostatus.no/datakat/18548">
  <cr:mediaType>text/xml</cr:mediaType>
  <cr:hasFeedback rdf:resource="http://62.92.43.137/FeedbackReportS?...q=18548"/>
</rod:Delivery>
```

The page with the feedback report must have a persistent unique URL. If there are attachments to the feedback report, there is a <cr:hasAttachment> property, which works the same way as the <rod:hasFile> does for deliveries.

### 2.4.2. Using CR to get information

Since some information about a delivery is stored at the provider's location and other information at QAW, Reportnet uses CR to link it all together. The SEIS node can call CR with XML-RPC to retrieve whatever information is available for the delivery or file by giving the URL as the argument. What will come back is a table with three columns called Source, Predicate and Object.

- Source contains the URL of the location the information was retrieved from. You will see the URL of your own manifest file for some of the records, making it possible for you to discard the records you already have
- Predicate is the name of the element used in the RDF with the namespace part expanded. E.g. "cr:mediaType" is expanded to "http://cr.eionet.europa.eu/ontologies/contreg.rdf#mediaType".
- Object is the information.

For more information see the CR User Manual.

### 2.4.3. Using webquestionnaires

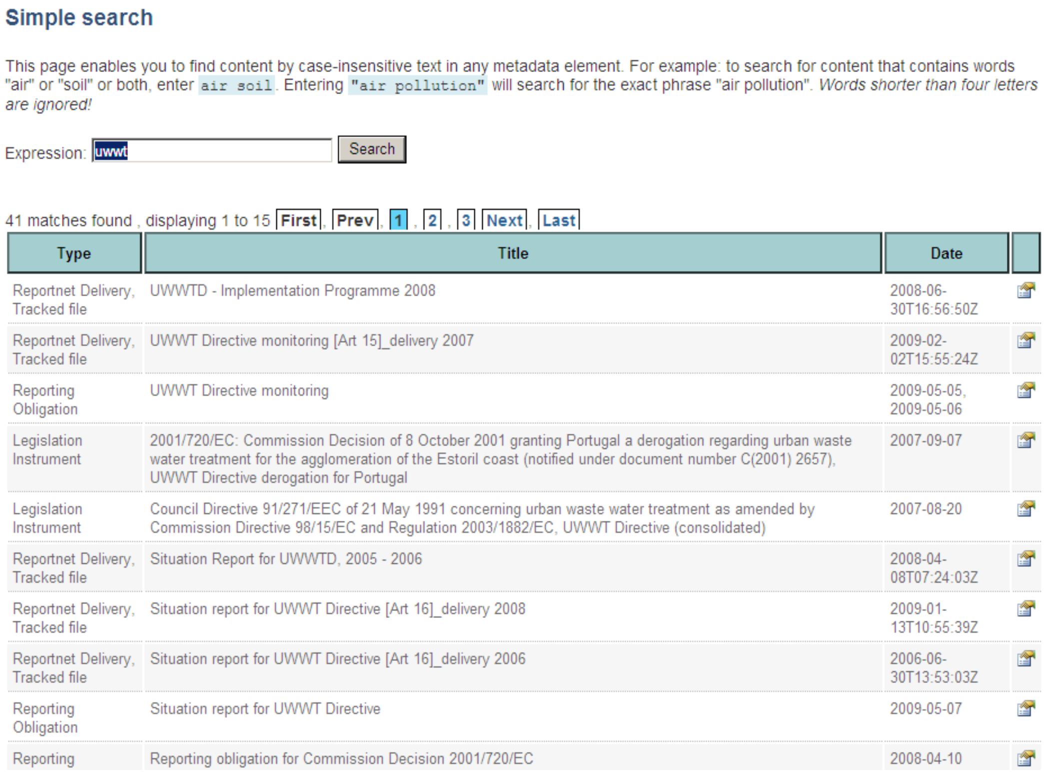
The interface between the data provider and the requester is the file that is transferred. But Reportnet also has a module to support drafting the delivery. It is the webforms module, and is a tool to edit XML files with forms. The node can ask the module if there are any forms available for a given XML format. It is mentioned here because it is a very visible part of CDR, but currently it is too specific to CDR to be of use by a Reportnet/SEIS node unless it is a direct clone of the CDR software.

# 3. Advanced concepts

You have now seen the practical use of manifests and declaration of objects. It is time for some theory.

## 3.1. Object types

The search engine in Reportnet is object oriented. On the simplest level, this means that when you search on a keyword, you can see what type of object you've found.



### Simple search

This page enables you to find content by case-insensitive text in any metadata element. For example: to search for content that contains words "air" or "soil" or both, enter `air soil`. Entering `"air pollution"` will search for the exact phrase "air pollution". *Words shorter than four letters are ignored!*

Expression: uwwt [Search]

41 matches found , displaying 1 to 15 First, Prev, 1 , 2 , 3 Next, Last

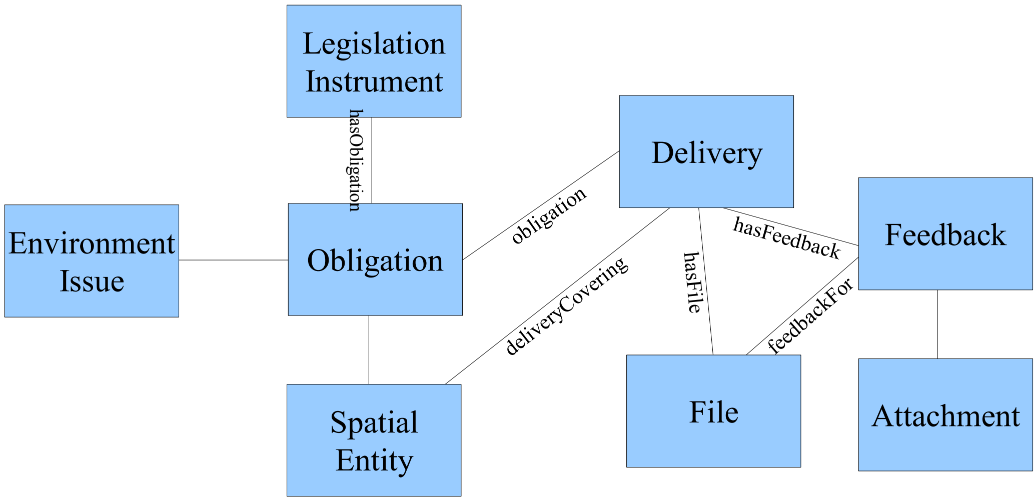
| Type | Title | Date | |
|---|---|---|---|
| Reportnet Delivery, Tracked file | UWWTD - Implementation Programme 2008 | 2008-06-30T16:56:50Z | |
| Reportnet Delivery, Tracked file | UWWT Directive monitoring [Art 15]_delivery 2007 | 2009-02-02T15:55:24Z | |
| Reporting Obligation | UWWT Directive monitoring | 2009-05-05, 2009-05-06 | |
| Legislation Instrument | 2001/720/EC: Commission Decision of 8 October 2001 granting Portugal a derogation regarding urban waste water treatment for the agglomeration of the Estoril coast (notified under document number C(2001) 2657), UWWT Directive derogation for Portugal | 2007-09-07 | |
| Legislation Instrument | Council Directive 91/271/EEC of 21 May 1991 concerning urban waste water treatment as amended by Commission Directive 98/15/EC and Regulation 2003/1882/EC, UWWT Directive (consolidated) | 2007-08-20 | |
| Reportnet Delivery, Tracked file | Situation Report for UWWTD, 2005 - 2006 | 2008-04-08T07:24:03Z | |
| Reportnet Delivery, Tracked file | Situation report for UWWT Directive [Art 16]_delivery 2008 | 2009-01-13T10:55:39Z | |
| Reportnet Delivery, Tracked file | Situation report for UWWT Directive [Art 16]_delivery 2006 | 2006-06-30T13:53:03Z | |
| Reporting Obligation | Situation report for UWWT Directive | 2009-05-07 | |
| Reporting | Reporting obligation for Commission Decision 2001/720/EC | 2008-04-10 | |

*Illustration 2: Search on 'uwwt' delivers hits of various types*

In chapter 2 we introduced <rod:Delivery>, <rod:File>, <cr:Feedback> and <cr:Attachment>. These are object types and they are used for the 'Type' column in Illustration 2. The <rod:Delivery> shows up as "Reportnet Delivery" because CR shows a friendly label for it.

Using the same mechanism, we have created object types for obligations, legal instruments, dams, EPER facilities, emissions and activities and created relationships between the objects. You have seen these as <rod:obligation>, <rod:hasFile>, <rod:locality> and <cr:hasFeedback>. These point to other objects with their rdf:resource attributes.

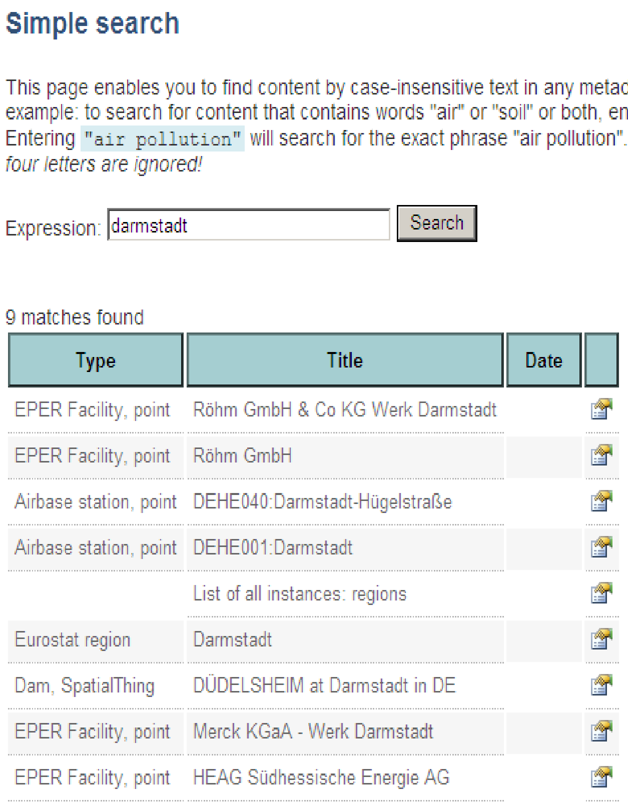
The effect is that we're slowly building a unified object model of all Reportnet's deliveries as objects. The illustration below shows the data model for ROD. You can recognise some of the types from the previous chapter.




## 3.2. More types

Content Registry even looks for objects *inside* the files of the deliveries. Therefore, a delivery of EPER facilities in *one* XML file from Germany creates about 1700 facilities, with 1800 activities and 4000 emissions. To get the facilities out of the XML file and into the CR database, we wrote a XSL stylesheet, that in practice converts the XML file into a manifest file and you can now search on EPER objects in CR. By just changing the stylesheet we can map the old EPER files into the E-PRTR data model, if we choose to do so, and then let CR reharvest the files.

When this happens for all Reportnet deliveries, you can search for very specific data, such as all objects related to Darmstadt like in Illustration 3 or finding everything that either generates or measures a specific pollutant in an area, which is determined by lat/long coordinates or NUTS region.



## Simple search

This page enables you to find content by case-insensitive text in any metadata element. For example: to search for content that contains words "air" or "soil" or both, enter `air soil`. Entering `"air pollution"` will search for the exact phrase "air pollution". *Words shorter than four letters are ignored!*

Expression: darmstadt Search

9 matches found

| Type | Title | Date | |
|---|---|---|---|
| EPER Facility, point | Röhm GmbH & Co KG Werk Darmstadt | | |
| EPER Facility, point | Röhm GmbH | | |
| Airbase station, point | DEHE040:Darmstadt-Hügelstraße | | |
| Airbase station, point | DEHE001:Darmstadt | | |
| | List of all instances: regions | | |
| Eurostat region | Darmstadt | | |
| Dam, SpatialThing | DÜDELSHEIM at Darmstadt in DE | | |
| EPER Facility, point | Merck KGaA - Werk Darmstadt | | |
| EPER Facility, point | HEAG Südhessische Energie AG | | |

*Illustration 3: Finding objects inside XML files*

## 3.3. Super-types

Using object-oriented techniques, it is possible to declare that all objects with lat/long coordinates shall be treated as points on a map of Europe. It requires, obviously, that the objects *have* lat/long coordinates. We have done it with EPER facilities, airbase stations and dams and a few others sofar.

We have created an object called WGS84 “point” and declared that “point” has four properties; latitude, longitude, altitude and label.

We then declare in EPER’s data model that an <eper:Facility> is a sub-class of <wgs84:Point> and subsequently that <eper:latitude> and <eper:longitude> are sub-properties of <wgs84:lat> and <wgs84:long> respectively. This modelling is done in a declarative language called RDF schema.

What it means, is that systems don’t need to know about the various object types. If you want something shown on a map, then query CR for objects of the “point” type. Likewise, we can introduce other abstract types such as “Station”, “Measurement”, “Event”, “Contact”, “Site”, “National legislation” etc.

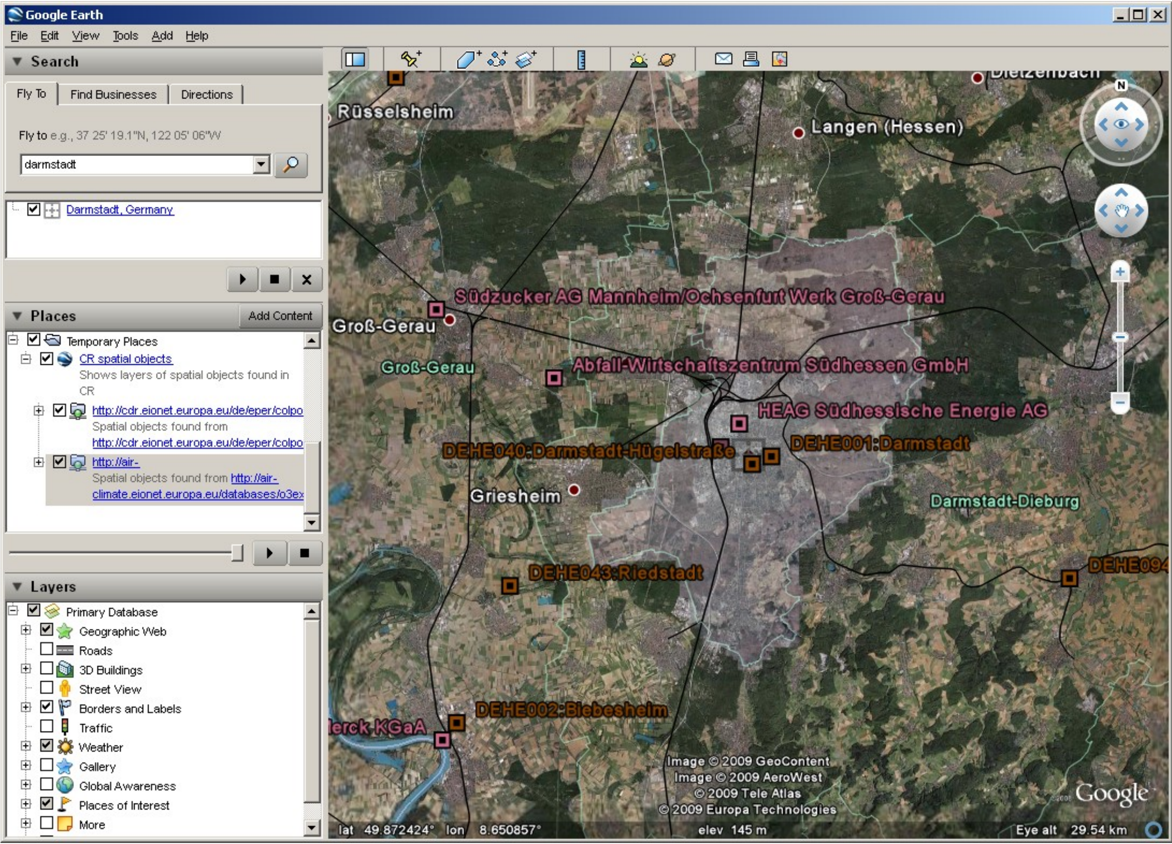



*Illustration 4: Known points around Darmstadt*

## 3.4. Rolling your own objects

It is possible to introduce your own obligations into the system as the list of obligations is just another dataset that can be amended.

To set up your own obligation, you first have to create the legislative instrument providing the foundation for the reporting obligation. A legislative instrument is a directive, national law, bylaw or other agreement describing the issue.

You should first check if ROD already has the instrument in its database. If it does, then your obligation should just refer to it, otherwise you have to create a record for it.

The rdf:about should point to an address where it is possible to get authoritive information about the instrument, including whether it is still in operation.

Here is an example of a legal instrument in the manifest file:

```
<rod:Instrument rdf:about="http://ejustice.just.fgov.be/LOI/2007000543">
  <dc:title>Loi du 15 avril 1994 relative à la protection de la population et de
l'environnement contre les dangers résultant des rayonnements ionisants et
relative à l'Agence fédérale de Contrôle nucléaire</dc:title>
  <dcterms:issued>2007-06-08</dcterms:issued>
</rod:Instrument>
```

Only the dc:title is needed for Reportnet, and it is used to organise the obligations into groups. But any attribute can be added, and you should add as much Dublin Core metadata as possible.

Next task is to create a record for the obligation in the manifest file.

```
<rod:Obligation rdf:about="http://www.ejustice.just.fgov.be/obligations/118">
  <dc:title>Liste des valeurs des différents rayonnements</dc:title>
  <rod:instrument rdf:resource="http://ejustice.just.fgov.be/LOI/2007000543"/>
  <rod:terminated>0</rod:terminated>
  <rod:nextdeadline>2011-12-31</rod:nextdeadline>
</rod:Obligation>
```

Finally, you need to tell the CR search engine where it can harvest the information about the obligations and instruments. These obligations will then appear on the list of obligations to choose from in CDR.

Ok, so it might be of limited value to create obligations for the purpose of reporting on national obligations to CDR. But ROD contains obligations that are "owned" by other international organisations. If we could get some of them to maintain a ROD cointaining their own obligations and legal instruments (following SEIS principles), then Reportnet can handle it.

### 3.4.1. Alternative regions

What might be more interesting for a member state is that you can create you own regions. The declaration follows the same principles as for obligations.

# 4. SOER C reporting

As discussed internally at EEA the countries have two main options:

Option 1: Their respective country web site residing on an Agency web server, but content maintained by the country.

Option 2: The country maintains its country web site on their national systems and provide feeds to the SOER Part C portal.

Option 3: A combination of these two options might be possible as well.

The SOER part C portal will support the upload via a web interface into which countries can directly upload their content either via a web form template (option one) or/and through "RSS/RDF feeds" (option two).

This chapter will discuss option two.

## 4.1. Proposal to implement option 2 as Reportnet/SEIS

We propose to reuse the expertise we've gained from Reportnet to harvest SOER part C content. What it means in practical terms is that the participants must create a feed (a.k.a. manifest) containing the metadata for the SoE reports. It is possible to provide several feeds, in case participants own more than one relevant website.

The first task is to create the object type. We can call it an SOER message. We need to decide what metadata is needed for a presentation on the SOER part C portal. Essentially we are talking about *category types* to navigate by and filter on.

These types are not agreed yet. There is a proposal to use DPSIR, topic, message type, country and publish date as types. If consensus is achieved, then those types are to be included in the feed as properties to the SOER message objects.

An example of the resulting RDF for one SOER message:

```
<soer:Message rdf:about="http://www.change.ie/en/whats-happening/National-Level">
  <dc:title>Ireland has committed to reducing greenhouse gas emissions</dc:title>
  <dc:date>2009-06-26</dc:date>
  <dc:description>Under the EU "burden sharing"** agreement in meeting the
targets set by Kyoto, Ireland has committed to restrict greenhouse gas emissions
growth.</dc:description>
  <soer:country>Ireland</soer:country>
  <soer:type>Future trends</soer:type>
  <soer:dpsir>Driver</soer:dpsir>
  <soer:topic>Climate change</soer:topic>
  <soer:topic>Policy instruments</soer:topic>
</soer:Message>
```

## 4.2. What could it look like?

The software development group has made a demonstration page to investigate various ideas. Content will be divided into a number of “facets” — matching the types above — which will then enable a user of Part C information to organise the information they need online according to their particular interests.

For example, if the screen is divided into three main columns, the right and left columns could show the facets which could be ticked off by the user, and then an automatic search would display the results in the centre column. As a concrete example, facet boxes could be ticked for “Slovakia”, “water” and “success stories”, and the results of the search would be displayed in the centre. See Illustration 5.

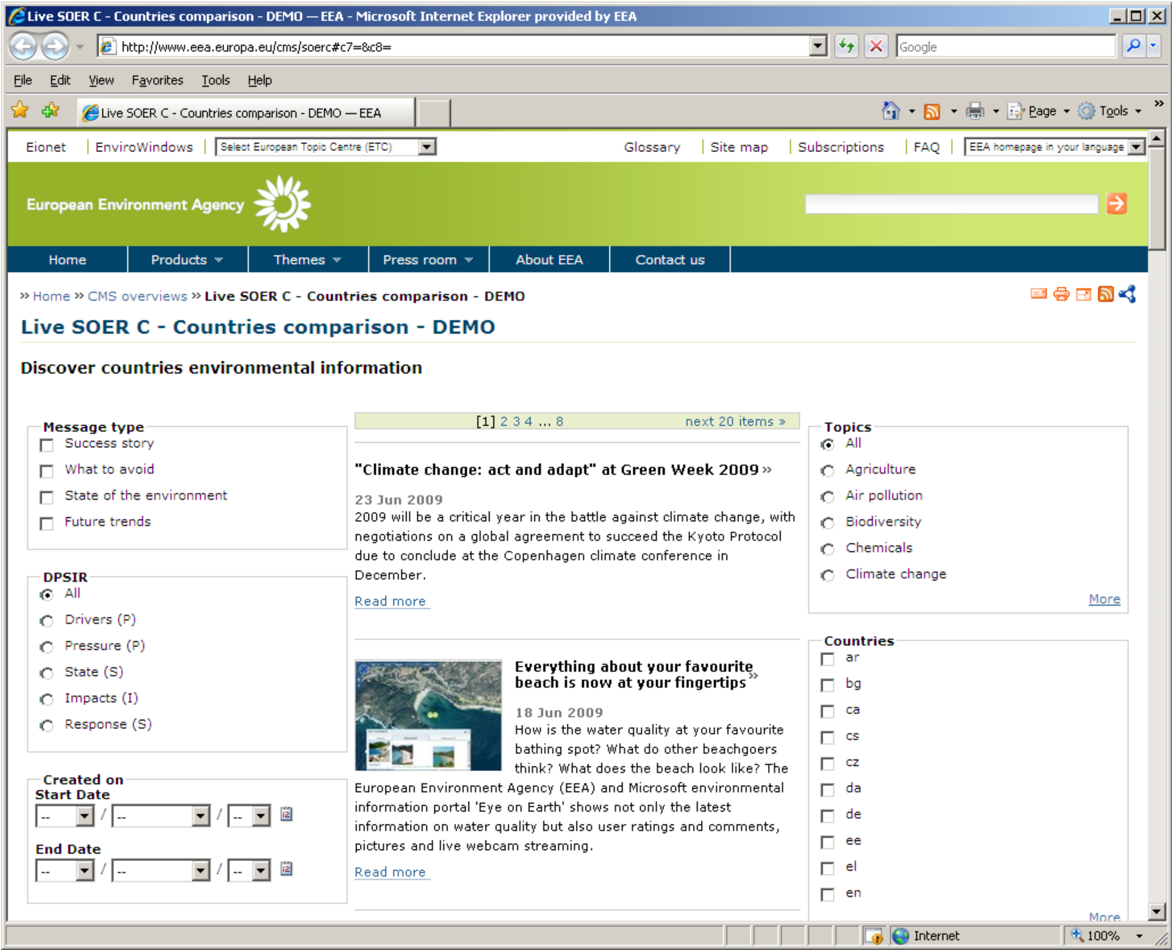



*Illustration 5: Demo of SOER C portal (with fictive content)*

## 4.3. Vocabularies

Once the facets have been agreed, the next task is to create vocabularies for them. A vocabulary describes a set of values allowed for the facet. It is important that everyone use the same terms otherwise grouping is going to break down.

Vocabularies can also set up relations between themselves. For instance, if there is a linkage between the DPSIR facet of "Response" and the topic facet of "Climate change adaptation", then the portal can grey out "Climate change adaptation" when the user deselects "Response".

# 5. Further reading

1. ISO-8601: http://en.wikipedia.org/wiki/ISO_8601
2. RDF-Primer: http://www.w3.org/TR/rdf-primer/
3. Linked data tutorial: http://www4.wiwiss.fu-berlin.de/bizer/pub/LinkedDataTutorial/