



---

# **Sensor Web Enablement**

## *Installation Guide for Sensor Observation Service with Generalizer, Diagram and KML extension Version 2.0*

---

**Document Change Control**

<b><i>Revision Number</i></b>	<b><i>Date Of Issue</i></b>	<b><i>Author(s)</i></b>	<b><i>Brief Description Of Changes</i></b>
0.1	2010-11-22	Carsten Hollmann	initial version
0.2	2012-03-21	Carsten Hollmann	Update of used tags
1.0			final review

**Editor**

Carsten Hollmann

52°North

Initiative for Geospatial Open Source Software GmbH

Martin-Luther-King-Weg 24

D-48155 Münster

Email: [c.hollmann@52north.org](mailto:c.hollmann@52north.org)

## Licence

This document is part of 52°North

Copyright (C) 2008 52°North

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA.

For more information, contact:

52°North

Martin-Luther-King-Weg 24

48155 Münster

Germany

<http://www.52north.org>

[This page has been intentionally left blank.]

## Table of Contents

<b>1 Introduction.....</b>	<b>7</b>
1.1 Scope.....	7
1.2 What you are doing following the installation procedure? .....	7
1.3 Characteristics of the Sensor Observation Service with Generalizer, Diagram and KML extension.....	7
1.4 Supported operations of the Sensor Observation Service with Generalizer, Diagram and KML extension.....	7
<b>2 Requirements.....</b>	<b>8</b>
<b>3 Installation Procedure.....</b>	<b>8</b>
3.1 Get the programmms.....	8
3.1.1 Configure Maven.....	8
3.2 Get the sources.....	8
3.3 Folder structures.....	9
3.4 Configure the properties.....	10
3.4.1 Configure build.properties in 52n-sos-dao-postgis-generalizer-diagram-kml.....	10
3.5 Configure the pom.xml files.....	11
3.5.1 Configure the main pom.xml in the standard SOS project.....	11
3.5.2 Configure the 52n-sos-service pom.xml in the standard SOS project.....	11
3.6 Build the web application.....	12
3.6.1 Configuration of the generalization.....	12
3.6.2 Deploy the web application.....	15
3.7 Tests.....	16
3.7.1 Example for GetCapabilities request.....	16
3.7.1.1 Capabilities-Request via HTTP GET.....	16
3.7.1.2 Capabilities Request via HTTP POST: .....	16
3.7.2 Other tests.....	17
<b>4 Extensions.....</b>	<b>18</b>
4.1 Generalization.....	18
4.2 Diagram and KML response in GetObservation request.....	19

---

4.2.1 Diagram in GetObservation request.....	19
4.2.2 KML in GetObservation request.....	21
4.3 Diagram pre-rendering.....	24
4.3.1 Diagram pre-rendering frequency.....	24
4.3.2 Replacement of invalid characters.....	24
<b>5 Appendix.....</b>	<b>26</b>
5.1 Samples of generalized time series.....	26
5.2 Additional properties in build.properties for the example requests.....	27
5.3 References.....	28

## 1 Introduction

### 1.1 Scope

This document describes the installation process of the Sensor Observation Service. Since the installation process is quite similar to the installation of the standard Sensor Observation Service, this is only an addition to the installation manual of the standard Sensor Observation Service. Please refer also to this manual if any questions occur during the installation process. A copy of this document can be found in the standard Sensor Observation Service repository.

### 1.2 What you are doing following the installation procedure?

The base Sensor Observation Service version is the 52°North SOS 3.2.0 release. You will download the source release from the 52°North download page and add some additional packages containing files for the SOS as a Java servlet

The `52n-sos-dao-postgis-generalizer-diagram-kml` extension is a replacement of the `52n-sos-dao-postgis` module. It brings own data source adapters for the SOS web service with PostgreSQL and PostGIS. The `52n-sos-generalizer` module is an extension for generalizing time series. The `52n-sos-coding-diagram` module is an extension diagram rendering and the `52n-sos-coding-kml` module is an extension for generating KML files from the data. It is necessary to check out the four repositories. Executing the installation steps [build] will deploy this service in your Apache Jakarta Tomcat web container as a web application (webapp).

### 1.3 Characteristics of the Sensor Observation Service with Generalizer, Diagram and KML extension

This Version of the SOS is based on the 52°North implementation of the SOS. Major functionality of the standard SOS is supported as well. In addition some more features are available, e.g.:

- Generalize the data
- Query KML in GetObservation request
- Query Diagrams in GetObservation request
- ...

### 1.4 Supported operations of the Sensor Observation Service with Generalizer, Diagram and KML extension

The SOS supports the SOS core profile with the following operations:

- GetCapabilities
- DescribeSensor
- GetObservation

Additionally, all other operation are supported.

## 2 Requirements

- Windows 2000 or higher [tested with Windows XP SP2]
- Sun JRE/JDK 1.6 [1.6.0\_XX]
- Apache Jakarta Tomcat 5.5 or higher [6.0.20]
- SVN-Client
- Maven 2.X [2.2.1, 3.0.3]
- 52N SOS 3.2.0 release
- 52N SOS Postgis/Generalizer/Diagram/KML module
- 52N SOS Generalizer module
- 52N SOS Diagram coding module
- 52N SOS KML coding module

## 3 Installation Procedure

### 3.1 Get the programms

- Download Apache Jakarta Tomcat from:

<http://jakarta.apache.org/tomcat>

Follow the installation instructions given on the Apache website to install the Apache Jakarta Tomcat.

- Download Apache Maven from:

<http://maven.apache.org/>

Follow the installation instructions given on the Apache website to install the Apache Maven.

#### 3.1.1 Configure Maven

- See chapter 3.1.1 in the how2install\_SOS document.

### 3.2 Get the sources

- Check out the **SOS 3.2.0 release** sources from 52°North download page:  
<http://52north.org/downloads/sensor-web/sos/52n-sensorweb-sos-320/download>
- Check out the **52n-sos-dao-postgis-generalizer-diagram-kml** sources from 52°North SVN-Repository:

Host: <https://52north.org/svn/>  
Repository: [swe/main/SOS/Extensions/PostGISGeneralizerDiagramKML/tags/](https://52north.org/svn/swe/main/SOS/Extensions/PostGISGeneralizerDiagramKML/tags/)  
Tag: `52n-sos-dao-postgis-generalizer-diagram-kml_20120321`



- Check out the **52n-sos-generalizer** sources from 52°North SVN-Repository:  
Host: `https://svn.52north.org/svn`  
Repository: `/swe/main/SOS/Extensions/Generalization/tags/`  
Tag: `52n-sos-generalizer_20120213`
- Check out the **52n-sos-coding-diagram** sources from 52°North SVN-Repository:  
Host: `https://svn.52north.org/svn`  
Repository: `/swe/main/SOS/Extensions/Diagram/tags/`  
Tag: `52n-sos-coding-diagram_20120321`
- Check out the **52n-sos-coding-kml** sources from 52°North SVN-Repository:  
Host: `https://svn.52north.org/svn`  
Repository: `/swe/main/SOS/Extensions/KML/tags/`  
Tag: `52n-sos-coding-kml_20120321`

### 3.3 Folder structures

The folder structure of the **52n-sos-dao-postgis-generalizer-diagram-kml** module is as follows:

- `conf`: configuration files; contains also the capabilities skeleton
- `src`: source files
- `webapp`: webapp folder with a SOS TestClient and examples

The folder structure of the **52n-sos-generalizer** module is as follows:

- `conf`: configuration files; contains also the capabilities skeleton
- `src`: source files
- `schemas`: XML-schemas for configuration files

The folder structure of the **52n-sos-coding-diagram** module is as follows:

- `conf`: configuration files
- `src`: source files (maven structure)

The folder structure of the **52n-sos-coding-kml** module is as follows:

- `conf`: configuration files
- `src`: source files (maven structure)

For the folder structure of the **standard SOS** see the `how2install_SOS` document chapter 3.3 in the standard SOS `/doc` folder.

### 3.4 Configure the properties

Although you have four repositories in your workspace, you will only need to configure the `52n-sos-dao-postgis-generalizer-diagram-kml` project. They will automatically pass the configuration to the standard SOS and compile it while building the SOS with Generalizer, Diagram and KML. You will find the configuration files in the `/conf` folders of your `52n-sos-dao-postgis-generalizer-diagram-kml` project.

#### 3.4.1 Configure build.properties in 52n-sos-dao-postgis-generalizer-diagram-kml

- Open the `build.properties` file in the `/conf` folder or the `52n-sos-dao-postgis-generalizer-diagram-kml` project and edit the following properties. The explanation contains a hint, if the property must be changed (MANDATORY) in order to allow the correct installation of the SOS or may be changed (OPTIONAL) to alter some default values. Properties marked as ADVANCED should not be changed. A description of the additional properties is listed below:

Table 1: SOS build.properties

Property	Explanation
<b>The following module specific properties</b>	
<b>conf.sos.path</b>	Path to the point where the configuration files are located. (ADVANCED)
<b>conf.sos.files</b>	Map with identifier and configuration file name. The id and file name are separated by '.' and the definitions by ';'. By default the following mapping is set (ADVANCED): <ul style="list-style-type: none"> <li>• GeneralizerConfig GeneralizerConfig.xml</li> <li>• Diagram diagram.config</li> <li>• KML kml.config</li> </ul>
<b>conf.sos.useGeneralizer</b>	Enables the generalization of requested SOS data (default := true). (ADVANCED)
<b>conf.sos.prebuild.diagram</b>	Indicate whether the SOS should pre render diagrams which can be accessed directly by a link. (default=false) For more information see chapter X.Y. (ADVANCED)
<b>conf.sos.prebuild.diagram.path</b>	Path where the pre rendered diagrams should be stored (default=[WEBAPP_BASE_PATH]/diagram)(ADVANCED)
<b>conf.sos.prebuild.diagram.period</b>	Indicates for how many days the diagrams should be pre rendered from actual time to past. (default=7). (ADVANCED)
<b>A default SOS property with additional functionality for diagram pre-rendering.</b>	
<b>conf.sos.capabilitiesCacheUpdateInterval</b>	Capabilities Cache Update Interval in minutes (0 = no automatic update). Here also used for starting the diagram pre-rendering. (ADVANCED)
<b>The following properties in the file are the default SOS properties. For detailed information see the how2install_SOS document in the doc folder of the default SOS-directory.</b>	

- Save changes

There are some additional properties in the `build.properties` file where you can define the parameters (e.g. offering, procedure, etc) for the examples requests. For more details see section 5.2.

### 3.5 Configure the pom.xml files

#### 3.5.1 Configure the main pom.xml in the standard SOS project

- Change in modules

```
<module>52n-sos-dao-postgis</module>
```

to

```
<module>../52n-sos-dao-postgis-generalizer-diagram-kml</module>
```

and **add**

```
<module>../52n-sos-generalizer</module>
<module>../52n-sos-coding-diagram</module>
<module>../52n-sos-coding-kml</module>
```

The modules section should look like this:

```
<modules>
  <module>52n-sos-ogc</module>
  <module>52n-sos-core</module>
  <module>52n-sos-coding</module>
  <module>52n-sos-service</module>
  <module>../52n-sos-dao-postgis-generalizer-diagram-kml</module>
  <module>../52n-sos-generalizer</module>
  <module>../52n-sos-coding-diagram</module>
  <module>../52n-sos-coding-kml</module>
</modules>
```

- Save pom.xml

#### 3.5.2 Configure the 52n-sos-service pom.xml in the standard SOS project

- Change in dependencies

- `<artifactId>52n-sos-dao-postgis</artifactId>`

- to

- `<artifactId>52n-sos-dao-postgis-generalizer-diagram-kml</artifactId>`

- and change the `<version>` to 1.0.0.

- and **add**

```
<dependency>
  <groupId>${pom.groupId}</groupId>
  <artifactId>52n-sos-generalizer</artifactId>
  <version>1.0</version>
</dependency>
<dependency>
  <groupId>org.n52.sensorweb.sos</groupId>
  <artifactId>52n-sos-coding-diagram</artifactId>
  <version>1.0</version>
</dependency>
<dependency>
  <groupId>org.n52.sensorweb.sos</groupId>
  <artifactId>52n-sos-coding-kml</artifactId>
  <version>1.0</version>
```

```
</dependency>
```

- The dependencies section should look like this:

```
<dependencies>
  <dependency>
    <groupId>${project.groupId}</groupId>
    <artifactId>52n-sos-dao-postgis-generalizer-diagram-
kml</artifactId>
    <version>1.0.0</version>
  </dependency>
  <dependency>
    <groupId>${pom.groupId}</groupId>
    <artifactId>52n-sos-generalizer</artifactId>
    <version>1.0</version>
  </dependency>
  <dependency>
    <groupId>org.n52.sensorweb.sos</groupId>
    <artifactId>52n-sos-coding-diagram</artifactId>
    <version>1.0</version>
  </dependency>
  <dependency>
    <groupId>org.n52.sensorweb.sos</groupId>
    <artifactId>52n-sos-coding-kml</artifactId>
    <version>1.0</version>
  </dependency>
</dependencies>
```

- Save pom.xml

## 3.6 Build the web application

### 3.6.1 Configuration of the generalization

The SOS is capable of generalizing time series to reduce the amount of data encoded in the O&M-response document. Therefore you need the generalizer extension. This extension contains several implementations of different generalization methods. The abstract class `AbstractGeneralizer` inside the `org.n52.sos.generalizer` package can be extended to implement own generalization methods. Generalizer implementations will be configured by using the `GeneralizerConfig.xml` file found in the `conf` folder of the `52n-sos-generalizer` project. At the time of delivery four generalization methods are supported:

- **Skipping every second value**

*This generalizer skips automatically every second value. The last value will be included anyway. No further configurations need to be made.*

- **Selecting every xx value**

*This generalizer takes only every xx value. The last value will be included anyway. The xx value has to be defined by using the `SELECT_VALUE_NUMBER` property.*

- **Skipping values within a certain tolerance**

*This generalizer calculates a line between different points and estimates the distance of the points in between to this line. If they are below a certain value,*

*these points will not be taken. This value is defined by using a percentage from the span of the requested interval's values. It has to be set by using the `TOLERANCE_PERCENT_VALUE` property. Additionally a maximum number of points to process can be defined by using the `MAX_ENTRIES` property. If this amount of point is reached an exception will be thrown. If this value is not defined it will be set to unlimited automatically.*

- **Skipping values within a certain tolerance (request time period dependant)**

*This generalizer acts like the one before. Only the percentage value can be set time dependant. Specific values can be set for different request time periods. To set these values use the `TOLERANCE_PERCENT_VALUE#`. The values are defined as follows: `Percent/Hours_from-Hours_till`, e.g. `0/0-24`. That means no tolerance will be applied for the period from 0 to 24 hours. Overlapping of percent/time values is not allowed. All periods from 0 hours to UNL need to be covered. The `MAX_ENTRIES` can be set as well as described above.*

To select and configure the volitional means of generalization edit the `GeneralizerConfig.xml` file in the `conf` folder of the `52n-sos-generalizer` project. First edit the `Generalizer`-tag and the `DecimalPlaces`-tag. Then define the `GeneralizerToUse`-Attribute. A description of the XML-schema can be found in figure 3.

Take a look at 4.1.7 for some sample graphics that show the effect of generalization.

Figure 1: screenshot GeneralizerConfig.xml file

```

<?xml version="1.0" encoding="UTF-8"?>
<GeneralizerConfiguration xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://localhost:8080/schemas/52n/generalizer.xsd" GeneralizerToUse="5"
DecimalPlaces="1">
  <Generalizer>
    <ID>1</ID>
    <Name>SkipEverySecond-Generalizer</Name>
    <Description>This generalizer skips every second value.</Description>
    <Procedure>urn:ogc:generalizationMethod:IFGI:SkipEverySecond</Procedure>
    <Class>org.n52.sos.generalizer.implementation.SkipSecondValueGeneralizer</Class>
    <Properties />
  </Generalizer>

  <Generalizer>
    <ID>2</ID>
    <Name>SelectValue-Generalizer</Name>
    <Description>This generalizer selects only each defined value.</Description>
    <Procedure>urn:ogc:generalizationMethod:IFGI:SelectValue</Procedure>
    <Class>org.n52.sos.generalizer.implementation.SelectValueGeneralizer</Class>
    <Properties>
      <Value key="SELECT_VALUE_NUMBER">4</Value>
    </Properties>
  </Generalizer>

  <Generalizer>
    <ID>3</ID>
    <Name>Tolerance-Generalizer</Name>
    <Description>This generalizer skips values that are within a predefined tolerance.</Description>
    <Procedure>urn:ogc:generalizationMethod:IFGI:Tolerance</Procedure>
    <Class>org.n52.sos.generalizer.implementation.ToleranceGeneralizer</Class>
    <Properties>
      <Value key="TOLERANCE_PERCENT_VALUE">2.0</Value>
    </Properties>
  </Generalizer>

  <Generalizer>
    <ID>4</ID>
    <Name>Tolerance-Generalizer time dependant</Name>
    <Description>This generalizer skips values that are within a predefined tolerance depending on the defined
tolerance for a certain time period.</Description>
    <Procedure>urn:ogc:generalizationMethod:IFGI:ToleranceTime</Procedure>
    <Class>org.n52.sos.generalizer.implementation.ToleranceTimeGeneralizer</Class>
    <Properties>
      <!--MAX_ENTRIES is the value for the maximum points the generalizer will handle, otherwise an
exception will be thrown; -1 is unlimited-->
      <Value key="MAX_ENTRIES">-1</Value>
      <!-- PERCENT/HOURS_FROM-HOURS_TILL -->
      <Value key="TOLERANCE_PERCENT_TIME_VALUE1">0/0-24</Value>
      <Value key="TOLERANCE_PERCENT_TIME_VALUE2">0.5/24-48</Value>
      <Value key="TOLERANCE_PERCENT_TIME_VALUE3">1.0/48-72</Value>
      <Value key="TOLERANCE_PERCENT_TIME_VALUE4">1.5/72-168</Value>
      <Value key="TOLERANCE_PERCENT_TIME_VALUE5">2.0/168-768</Value>
      <Value key="TOLERANCE_PERCENT_TIME_VALUE6">3.0/768-5440</Value>
      <Value key="TOLERANCE_PERCENT_TIME_VALUE7">5.0/5440-UNL</Value>
    </Properties>
  </Generalizer>

```

Figure 2: screenshot GeneralizerConfig.xml file (2)

```

<Generalizer>
  <ID>5</ID>
  <Name>Douglas-Peucker-Generalizer</Name>
  <Description>This generalizer skips non-characteristic values depending on a given
tolerance.</Description>
  <Procedure>urn:ogc:generalizationMethod:IFGI:douglas-peucker-1</Procedure>
  <Class>org.n52.sos.generalizer.implementation.DouglasPeuckerGeneralizer</Class>
  <Properties>
    <!-- MAX_ENTRIES is the value for the maximum points the generalizer will handle, otherwise an
exception will be thrown; -1 is unlimited -->
    <Value key="MAX_ENTRIES">-1</Value>
    <!-- estimated reduction rate for this use case, where REDUCTION_RATE = 3 means the time series is
reduced to 1/3 of it's size; -1 means there is no paper empirical value -->
    <Value key="REDUCTION_RATE">-1</Value>
    <!-- absolute tolerance value -->
    <Value key="TOLERANCE_VALUE">0.1</Value>
  </Properties>
</Generalizer>
</GeneralizerConfiguration>

```

### 3.6.2 Deploy the web application

- Make sure that your Tomcat is started.
- Open a command line and assume to the project's folder, e.g. C:\SOS
- Type the following command line expression:
  - `mvn -Pwith-deploy install`
- The SOS with extensions is now available.

### 3.7 Tests

Manual tests can be run with own requests. Sample requests can be found in the `xml` folder. Sending of those requests is described in the following paragraph:

#### 3.7.1 Example for GetCapabilities request

The GetCapabilities-request is the only request which is possible to be sent via HTTP GET- and HTTP POST. The SOS webclient (see below) is attached to enable POST requests to the SOS.

##### 3.7.1.1 Capabilities-Request via HTTP GET

- Open your browser. Type in  
`http://your.sos.location:port/webapp_name/sos?REQUEST=GetCapabilities&SERVICE=SOS&ACCEPTVERSIONS=1.0.0`
- The capabilities response document is now shown in your browser.
- You can extend the request above through appending the optional parameters SECTIONS and ACCEPTFORMATS. Look at the SOS specification for more details.

##### 3.7.1.2 Capabilities Request via HTTP POST:

- Open your Internet Browser (e.g. Mozilla Firefox). You can find the SOS Test Client under `http://your.sos.location:port/webapp_name/testClient-v2.html`. (Make sure that your tomcat is running)

**SOS TestClient** Version 2 - [old version](#)

For more information about the 52° North Sensor Observation Service visit <http://52north.org/sos>.

Service URL:

Request Examples:

You can change the examples in the folder [project-directory]/52n-sos-service/src/main/webapp/examples/.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <GetCapabilities xmlns="http://www.opengis.net/sos/1.0"
3   xmlns:ows="http://www.opengis.net/ows/1.1"
4   xmlns:ogc="http://www.opengis.net/ogc"
5   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
6   xsi:schemaLocation="http://www.opengis.net/sos/1.0
7   http://schemas.opengis.net/sos/1.0/sosGetCapabilities.xsd"
8   service="SOS">
9
10 <ows:AcceptVersions>
11   <ows:Version>1.0.0</ows:Version>
12 </ows:AcceptVersions>
13
14 <ows:Sections>
15   <ows:Section>OperationsMetadata</ows:Section>
16   <ows:Section>ServiceIdentification</ows:Section>
17   <ows:Section>ServiceProvider</ows:Section>
18   <ows:Section>Filter_Capabilities</ows:Section>
19   <ows:Section>Contents</ows:Section>
20 </ows:Sections>
21
22 </GetCapabilities>
23

```

This TestClient was successfully tested in Firefox 3.5.2, Safari 4.0.3, Opera 9.64 and InternetExplorer 8.0.6001.18702 and should work properly in Firefox 1.0 or higher, Safari 1.2 or higher, Opera 8 or higher and InternetExplorer 5 or higher. The editor is based on CodeMirror (<http://marijn.haverbeke.nl/codemirror/>).

Figure 3: Screenshot of the SOS test client 2.0 for capabilities request



- Chose a request from the drop down list named 'Request Examples' or create your own request and paste your request in the test box.
- Click the „Send“-Button. The capabilities response xml-document is now shown in your browser.

### 3.7.2 Other tests

- You can try the other XML request files in the `xml` folder analogous to the HTTP Post based GetCapabilities request (e.g. copy and paste the content of the `GetObs_BBOX.xml` file).

## 4 Extensions

### 4.1 Generalization

With the generalization extension the user can query a generalized time series directly from the SOS. The generalizers are listed in the Capabilities document as a procedure and you can information about the generalize,r if you do a DescribeSensor request with the generalizer id as parameter. In a GetObservation request you can chose some valid combination to query a generalized time series.

In the following list the additional combinations are listed.

Procedure	Description
No parameter	No procedure in request.
<procedure>procedureID</procedure> (...)	One or many procedure(s) as procedure parameter in request.
<procedure>generalizerID</procedure> (...)	One or many generalizer(s) as procedure parameter in request.
<procedure>procedureID,generalizerID</procedure>	One procedure with one generalizer as procedure parameter in request.
<procedure>procedureID_1,generalizerID_1</procedure> <procedure>procedureID_1,generalizerID_2</procedure> ... or <procedure>procedureID_1,generalizerID_1</procedure> <procedure>procedureID_2,generalizerID_1</procedure> ... or <procedure>procedureID_1,generalizerID_1</procedure> <procedure>procedureID_2,generalizerID_2</procedure>	One procedure with different generalizers or Many procedures with the same generalizer or Many procedures with different generalizers as procedure parameter in request.
<procedure>procedureID_1,generalizerID_1</procedure> <procedure>procedureID_1</procedure> ... or <procedure>procedureID_1,generalizerID_1</procedure> <procedure>procedureID_2</procedure> ...	Combinations of procedure and generalizer and procedures as procedure parameter in request.

All other combinations are not supported and the response is an exception.

The default implemented generalizers are:

Name	description
SkipEverySecond-Generalizer	This generalizer skips every second value.
SelectValue-Generalizer	This generalizer selects only each defined value.
Tolerance-Generalizer	This generalizer skips values that are within a predefined tolerance.
Tolerance-Generalizer time dependant	This generalizer skips values that are within a predefined tolerance depending on the defined tolerance for a certain time period.
Douglas-Peucker-Generalizer	This generalizer skips non-characteristic values depending on a given tolerance.

## 4.2 Diagram and KML response in GetObservation request

The SOS supports KML files or diagrams as response of a GetObservation request. Difference to the SOS 1.0 Specification, the featureOfInterest parameter is mandatory in a KML or DIAGRAM request and the observableProperty parameter is restricted to only one value. Here is a short description which parameters are used:

- Build a GetObservation with the following parameters
  - offering (MANDATORY)
  - observedProperty (**only one**) (MANDATORY)
  - featureOfInterest id (**only one**) (MANDATORY)
  - procedure id (**only one**) (MANDATORY)
  - responseFormat (MANDATORY)
  - eventTime with TimePeriod (OPTIONAL)
- send the request to the SOS (via TestClient-v2 section 3.7).
- Example requests:

### 4.2.1 Diagram in GetObservation request

The diagram rendering is excluded into a separate module and renders a diagram from the requested values. The 52n-sos-coding-diagram module has a separate configuration file (52n-sos-coding-diagram/conf/diagram.config) where some properties can be set to modify the styling of the returned diagram. Here is a list of the properties:

Property	Description
SOS_NO_DATA_VALUE	The SOS no data value. Used to eliminate no data values from the diagram view.
DIAGRAM_WIDTH	The width of the diagram
DIAGRAM_HEIGHT	The height of the diagram

Property	Description
<b>SHOW_TITLE</b>	Indicates whether a title should be set. By default the title is the featureOfInterest id.
<b>SHOW_HORIZONTAL_GRID_LINES</b>	Indicate whether horizontal grid lines should be plotted
<b>SHOW_VERTICAL_GRID_LINES</b>	Indicate whether vertical grid lines should be plotted
<b>LINE_COLOR</b>	Define the color of the line by color name, e.g. RED, BLUE, GREEN, or by RGB value definition, e.g. (123,1,89).
<b>COPYRIGHT_TEXT</b>	Definition of the copyright text which should be plotted.
<b>X_AXIS_LABEL</b>	The label name of the x axis.
<b>NO_DATA_MESSAGE</b>	The text which should be plotted if no data are available.

During the Maven build process this file is copied and is then located in the [TOMCAT\_HOME]/webapps/[SOS\_NAME]/WEB-INF/conf folder. It is mandatory to define this file in the SOS build.properties file or in the dssos.config file of the deployed webapp.

Add the following definition 'Diagram diagram.config' to the following properties (by default set in the SOS build.properties file):

- `conf.sos.files` in build.properties (before the SOS is deployed)
- `CONFIGURATION_FILES` in dssos.config (in the deployed SOS)

To get a diagram as response from the SOS you have to set the following value in the parameter `<responseFormat>`:

- **image/jpeg**

Here is an example request of a diagram request and the returned diagram.

```

<?xml version="1.0" encoding="UTF-8"?>
<GetObservation xmlns="http://www.opengis.net/sos/1.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sos/1.0
  http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
  service="SOS" version="1.0.0">
  <offering>WASSERTEMPERATUR_ROHDATEN</offering>
  <procedure>Temperatur-Stoer-Sperwerk_Bp_5970040</procedure>
  <observedProperty>Temperatur</observedProperty>
  <featureOfInterest>
    <ObjectID>Stoer-Sperwerk_Bp_5970040</ObjectID>
  </featureOfInterest>
  <responseFormat>image/jpeg</responseFormat>
</GetObservation>

```

Listing 1: GetObservation request example for Diagram response.

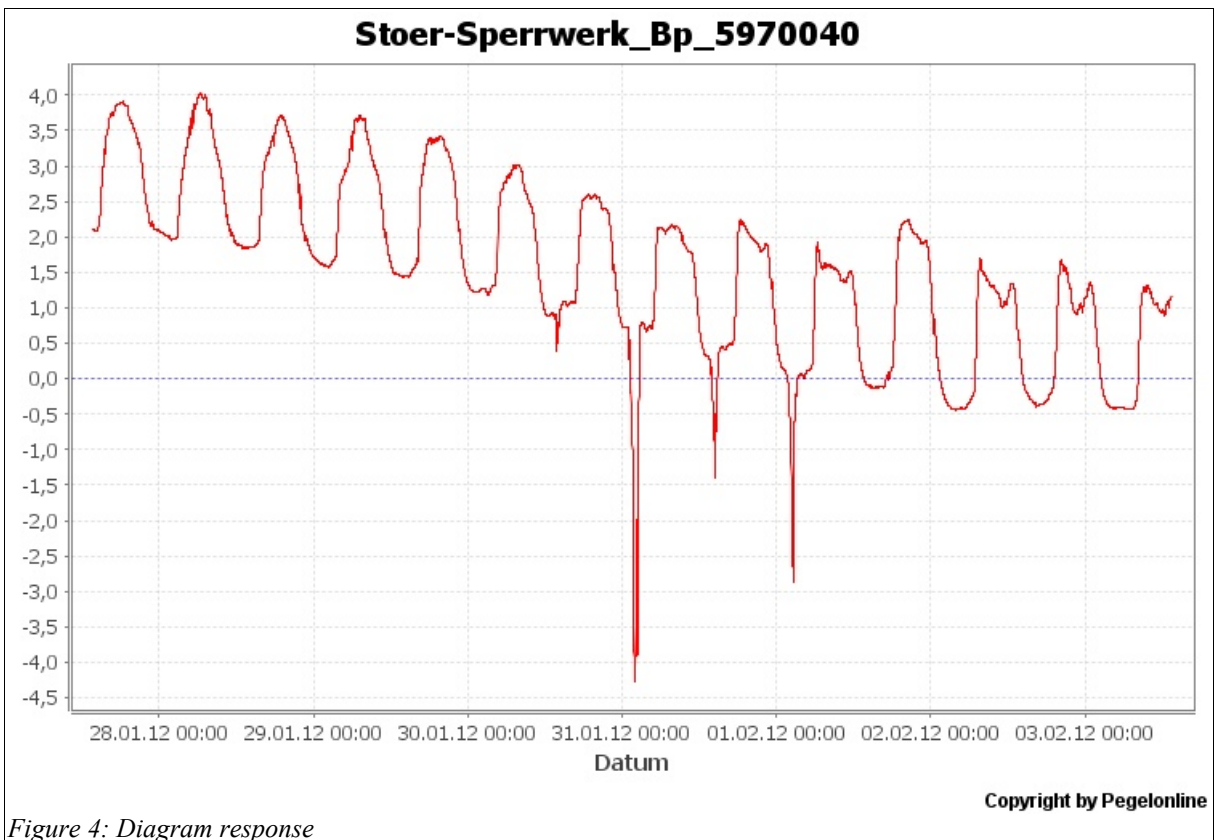


Figure 4: Diagram response

#### 4.2.2 KML in GetObservation request

The KML coding is excluded into a separate module and generates a KML file from the request. The 52n-sos-coding-kml module has a separate configuration file (52n-sos-coding-kml/conf/kml.config). It contains the definition of the used XSLT transformation file and some indicators how the diagram can be accessed (link in the KML file):

Property	Description
<b>XSL_10</b>	Path to the XSLT file.
<b>QUERY_DIAGRAM</b>	Indicates whether a HTTP-GET request is set (true) or a direct link (false) to get the diagram (default=true)
<b>QUERY_PATH</b>	The URL to the SOS for the HTTP-GET request. Mandatory if QUERY_DIAGRAM=true.
<b>DIAGRAM_DOWNLOAD_BASE_PATH</b>	Base path to the pre rendered diagrams. Mandatory if QUERY_DIAGRAM=false.
<b>DIAGRAM_DAYS</b>	The number of days which should be queried from actual time to the past for the diagram. Only set if request did not contain a eventTime parameter. Used if QUERY_DIAGRAM=true.
<b>DIAGRAM_WIDTH</b>	The width of the diagram
<b>DIAGRAM_HEIGHT</b>	The height of the diagram
<b>KML_RESPONSE_FORMAT</b>	KML response format definition. Valid are: <ul style="list-style-type: none"> <li>• application/vnd.google-earth.kml+xml (DEFAULT)</li> <li>• text/plain</li> </ul>
<b>DIAGRAM_RESPONSE_FORMAT</b>	Diagram response format definition, set in the HTTP-GET URL of KML response.

During the Maven build process this file is copied and is then located in the [TOMCAT\_HOME]/webapps/[SOS\_NAME]/WEB-INF/conf folder. It is mandatory to define this file in the SOS build.properties file or in the dssos.config file of the deployed webapp.

Add the following definition 'KML kml.config' to the following properties (by default set in the SOS build.properties file):

- `conf.sos.files` in build.properties (before the SOS is deployed)
- `CONFIGURATION_FILES` in dssos.config (in the deployed SOS)

To get a diagram as response from the SOS you have to set the following value in the parameter `<responseFormat>`:

- **application/vnd.google-earth.kml+xml**

Here is an example request of a kml request and the returned kml file.

```
<?xml version="1.0" encoding="UTF-8"?>
<GetObservation xmlns="http://www.opengis.net/sos/1.0"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/sos/1.0
  http://schemas.opengis.net/sos/1.0.0/sosGetObservation.xsd"
  service="SOS" version="1.0.0">
  <offering>WASSERTEMPERATUR_ROHDATEN</offering>
  <procedure> Temperatur-Stoer-Sperrwerk_Bp_5970040</procedure>
  <observedProperty>Temperatur</observedProperty>
  <featureOfInterest>
    <ObjectID>Stoer-Sperrwerk_Bp_5970040</ObjectID>
  </featureOfInterest>
  <responseFormat>application/vnd.google-earth.kml+xml</responseFormat>
</GetObservation>
```

*Listing 2: GetObservation request example for KML response.*

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns:sa="http://www.opengis.net/sampling/1.0"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:om="http://www.opengis.net/om/1.0"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:swe="http://www.opengis.net/swe/1.0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://earth.google.com/kml/2.1">
  <Document>
  <open>0</open>
  <Style id="FOI_STYLE_Stoer-Sperrwerk_Bp_5970040">
  <BalloonStyle>
  <bgColor>ffffff</bgColor>
  <text><b><font color="#CC0000" size="+3">${name}</font></b><br/><br/><font
  face="Courier">${description}</font><br/><br/></text>
  </BalloonStyle>
  </Style>
  <Placemark>
  <name>Stör-Sperrwerk Bp - Kilometer: 50</name>
  <description>Feature Of Interest: 'Stör-Sperrwerk Bp - Kilometer:
  50' (id := 'Stoer-Sperrwerk_Bp_5970040')</description>
  <styleUrl>#FOI_STYLE_Stoer-Sperrwerk_Bp_5970040</styleUrl>
  <Point>
  <coordinates>9.40234781963614, 3.82793502621518</coordinates>
  </Point>
  </Placemark>
  </Document>
</kml>
```

*Listing 3: KML response*

Here is an HTTP-GET example to query a GetObservation request with KML response which can be used directly in Google Maps:

```
http://maps.google.de/maps?f=q&hl=de&q=[SOS_URL]/sos?service=SOS
%26REQUEST=GetObservation%26version=1.0.0%26offering=WASSERSTAND_ROHDATEN
%26procedure=Wasserstand-Koeln_2730010%26observedProperty=Wasserstand
%26eventTime=2012-01-10T11:16:26.111/2012-01-
17T11:16:26.111%26featureOfInterest=Koeln_2730010%26responseFormat=text/xml
;subtype=%22kml/2.1%22
```

### 4.3 Diagram pre-rendering

This SOS has an additional functionality to pre-render diagrams for a defined period from the actual time.

To enable pre rendering you have to set the properties in the build.properties file (before deployment) or in the dssos.config file of the deployed service. After changes in the dssos.config file it is required to reload the SOS.

Here are the properties from the dssos.config file, the corresponding properties from the build.properties file are described in chapter 3.4.1.

- *PREBUILD\_DIAGRAM*
  - enables pre rendering of diagrams
- *PREBUILD\_DIAGRAM\_PATH*
  - path where the diagrams are stored
  - if the definition contains the string 'basePath', this string will be replaced with the webapps base path
  - The sub structure from this path is OFFERING\_NAME/OBSERVABLE\_PROPERTY\_ID/FEATURE\_OF\_INTEREST\_ID\_\_PROCEDURE\_ID.jpg
- *PREBUILD\_DIAGRAM\_PERIOD*
  - The period for which the diagrams should be pre-rendered in days, actual time minus defined days

#### 4.3.1 Diagram pre-rendering frequency

The frequency of the the diagram pre-rendering depends on the SOS capabilities cache update interval. The capabilities cache updater is a thread that periodically starts requesting the database and update the SOS internal cache. For diagram pre-rendering the capabilities cache update thread starts a new thread that pre-renders the diagrams. If the diagram pre-rendering thread is not finished until the capabilities cache update thread is started again, no new diagram pre-rendering thread will be started. When the next time the capabilities cache update thread runs and the diagram pre-rendering thread is finished, a new diagram pre-rendering thread will be started.

The capabilities cache update interval can be set in the build.properties file (see chapter 3.4.1).

#### 4.3.2 Replacement of invalid characters

The pre-rendered diagrams are stored in the described structure. In some cases the offering,



observableProperty, procedure or featureOfInterest can contain characters which are not allowed in a path, filename or URL. Then the pre-rendered diagram can not be stored in the file system or the URL is invalid. To prevent this, the not allowed characters will be replaced in the path/filename/URL with '\_' depending on the operating system.

Here are the characters which are replaced with '\_':

\ / : " \* ? < > ; # % = @

Special german characters are replaced as defined in the following:

- **ä -> ae**
- **ö -> oe**
- **ü -> ue**
- **Ä -> AE**
- **Ö -> OE**
- **Ü -> UE**
- **ß -> ss**

If you generate a direct link to the pre-rendered diagrams automatically in a program make sure that the characters mentioned before in offering, observableProperty, procedure and featureOfInterest are replaced with the '\_'.

Here is an example for a procedure id:

<b>Original</b>	<b>replaced</b>
urn:ogc:object:feature:Sensor:IFGI:ifgi-sensor-1	urn_ogc_object_feature_Sensor_IFGI_ifgi-sensor-1

## 5 Appendix

### 5.1 Samples of generalized time series

The sample images (figures 5-9) are taken from the ThinSweClient application. All generalized data is derived from the original data shown in figure 5.

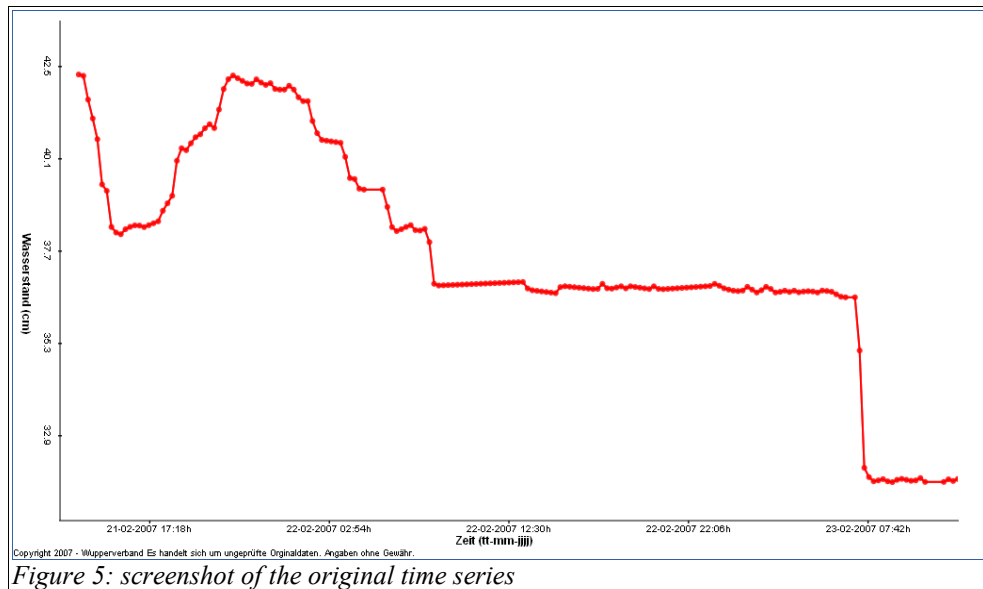


Figure 5: screenshot of the original time series

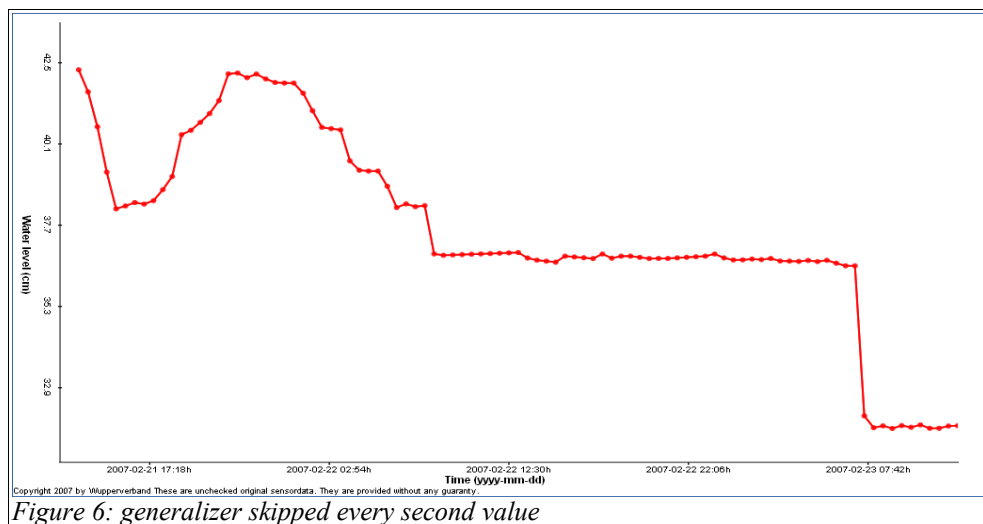
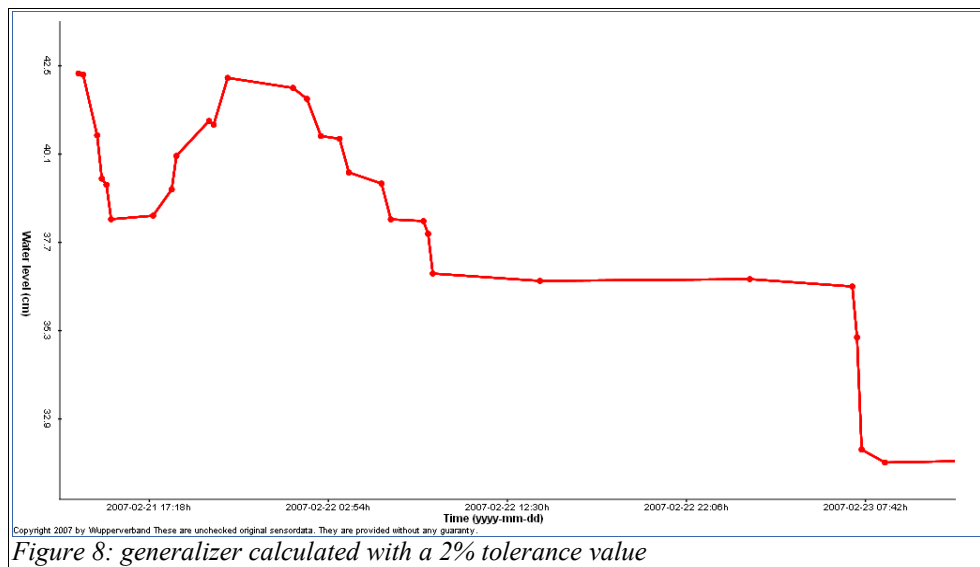
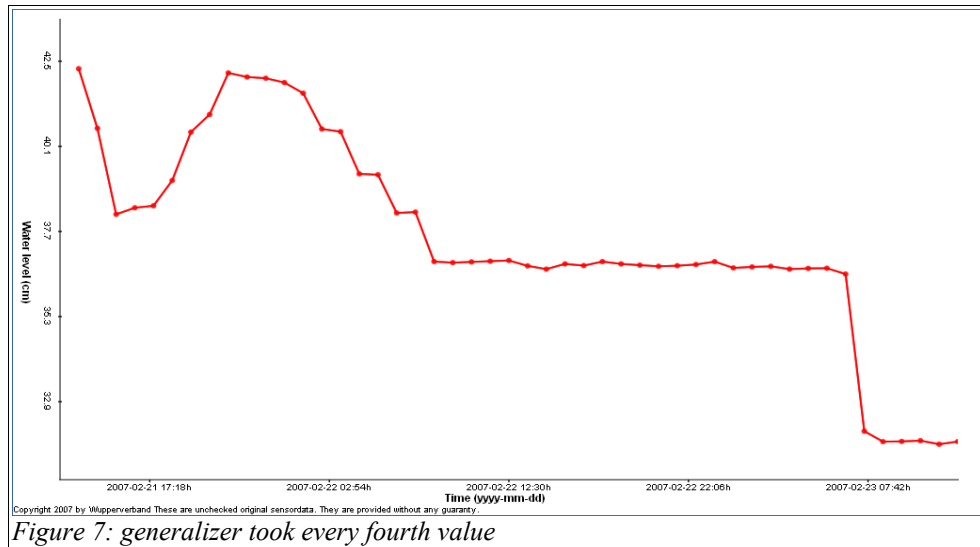


Figure 6: generalizer skipped every second value



## 5.2 Additional properties in build.properties for the example requests

This chapter describes some additional properties in the `build.properties` file where you can define the parameters (e.g. offering, procedure, etc) for the examples requests. These properties will be automatically set to the examples during the build process.

Property	Description
<code>sos.examples.def.offering</code>	The offering id for the requests.
<code>sos.examples.def.observableProperty</code>	The observableProperty id for the requests.
<code>sos.examples.def.procedure</code>	The procedure id for the requests.
<code>sos.examples.def.featureOfInterest</code>	The featureOfInterest id for the requests.
<code>sos.examples.def.generalizer</code>	The generalizer id for the requests.
<code>sos.examples.def.time.min</code>	The begin time for temporal filter.

Property	Description
<code>sos.examples.def.time.max</code>	The end time for temporal filter
<code>sos.examples.def.googleRequest.time.min</code>	The begin time for the temporal filter in the 'Station in Google-Maps via KML-Request' link in the testClient. <b>!!! It is required to define a time stamp without time zone/offset !!!</b>
<code>sos.examples.def.googleRequest.time.max</code>	The end time for the temporal filter in the 'Station in Google-Maps via KML-Request' link in the testClient. <b>!!! It is required to define a time stamp without time zone/offset !!!</b>
<code>sos.exmaples.def.srs</code>	The EPSG number for spatial filter request.
<code>sos.examples.def.bbox.lower</code>	Lower corner of BBOX in spatial filter requests (GetObservation, GetFeatureOfInterest)
<code>sos.examples.def.bbox.upper</code>	Upper corner of BBOX in spatial filter requests (GetObservation, GetFeatureOfInterest)
<p><i>If the default properties contains special sign like</i></p> <p><code>\ / : " * ? &lt; &gt; ; # % = @</code></p> <p><i>replace these characters with a '_' in the following properties.</i></p>	
<code>sos.examples.def.direktLink.offering</code>	The offering id for the direct diagram link.
<code>sos.examples.def.direktLink.procedure</code>	The observableProperty id for the direct diagram link.
<code>sos.examples.def.direktLink.observableProperty</code>	The procedure id for the direct diagram link.
<code>sos.examples.def.direktLink.featureOfInterest</code>	The featureOfInterest id for the direct diagram link.

### 5.3 References

- Botts, M. (2005): Sensor Model Language (SensorML) Implementation Specification, Best Practice Paper, Version 1.0.0, OGC document 05-086