Atkins logo Blue2718

**Development of an SqlServer data connector for SOS web service**

Technical Report

Development of an SqlServer data connector for SOS web service

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Project ref: 1010175 | | DOCUMENT REF: SOS final report | | | |
| Version | Remarks | Author | Reviewer | Approved | Date |
| 1 |  | ARH | JBM |  | 2010-09-23 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

Table of Contents

[1. Introduction and Scope 1](#_Toc277149626)

[1.1 Introduction 1](#_Toc277149627)

[1.2 Scope 1](#_Toc277149628)

[1.3 Background Information 1](#_Toc277149629)

[2. Development Environment Setup 2](#_Toc277149630)

[3. Experiences 5](#_Toc277149631)

[3.1 Challenges 5](#_Toc277149632)

# Introduction and Scope

## Introduction

This document describes the work carried out based on EEA’s SensorWeb TRAC ticket [#11: “Develop dataconnector for SqlServer”](https://svn.eionet.europa.eu/projects/SensorWeb/ticket/11).

As described under the ticket in TRAC the aim of the work has been to develop an additional data connector for the SOS web service. Currently, the SOS web service is only capable of retrieving data from a PostGIS database. With the new data connector however, it would also be able to retrieve data from Microsoft SqlServer.

## Scope

There are currently a large number of service operations available in the SOS web service. The scope of this task has been to make a minor subset of these operations run with SqlServer as a proto type or prove-of-concept. The names of the service operations in this subset are given in the screen dumb below. In the SOS TestClient They are the topmost 8 in the drop down list a prefixed with “EEA\_ “.

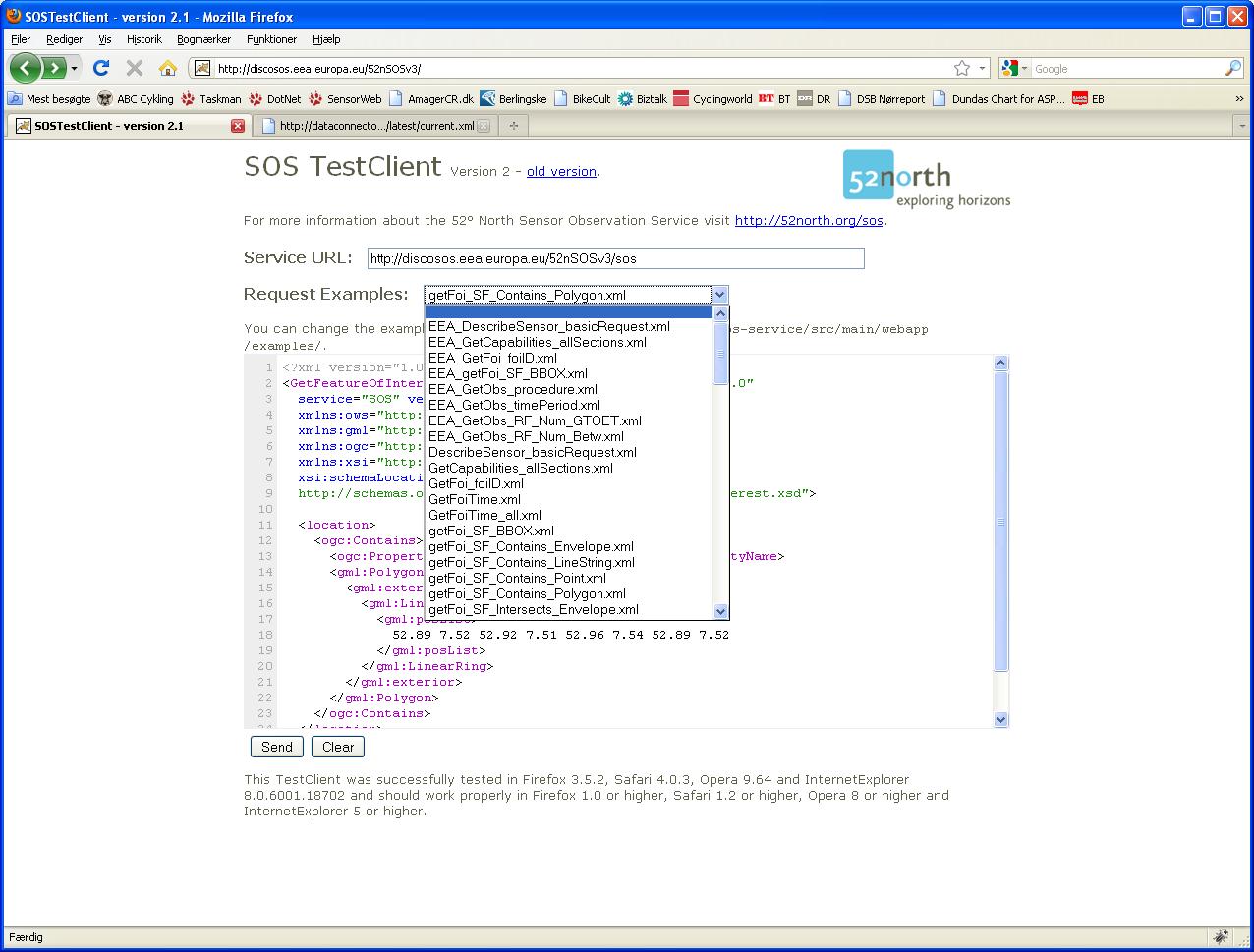
## Background Information

The initial work of setting up a functional development environment has widely been based on details from the how-to document from 52North: “How to check out the 52N Sensor Observation Service (SOS)” [version: 2010-05-26]. The document is available from the following URL:

<https://svn.52north.org/cgi-bin/viewvc.cgi/main/SOS/Service/trunk/SOS/52n-sos/doc/howto/maven-sos.pdf?view=co&root=swe>

Additional background information regarding the SOS web service can be on the wiki of 52North:

<https://wiki.52north.org/bin/view/Sensornet/SensorObservationService>



# Development Environment Setup

**Start here:**

Use the latest 'Maven-sos.pdf'-setup guide (date: 2010-05-26) from 52N [attached to this mail].

In **Section 2** pleaseInstall

· Maven 2.2.1

· Eclipse 3.4.1

· Subversive SVN Connectors 2.0.4

· Tomcat 6.0.18

· Java 1.6.0 JDK

Follow the configuration steps in **Section 3**.

On a Windows system the folder ${user.home}/.m2/  should be translated into C:\Documents and Settings\{username}\.m2    [I.e. the C:\Documents and Settings\{username} is also the plave where your "My Documents" folder is located].

**Section 4** is on how to check out SOS from SVN from inside Eclipse.

In **Section 5** when you enter the subsection "Developer Configuration" you may run into an Eclipse problem. If the button 'Enable dependency management' is not in the context menu when you right the project, you must re-add the button as described here

> The [Enable Nested Modules]-option is missing in new Eclipse versions,

> but can be re-added by doing the steps listed in this link:

> <http://stackoverflow.com/questions/2609938/what-happened-to-the-enable-nested-modules-option-in-m2-eclipse>

After enabling this feature for the project it should start to download all dependencies and build all code.

**New development:**

Now I suppose the next from here is to add a new folder ('eea-sos-dao-mssql') at the same level as '52n-sos-dao-postgis'.

Then you have to do changes in two files.

- parent pom.xml in the SOS directory: in section <modules> change the <module>52n-sos-dao-postgis</module> to <module>eea-sos-dao-mssql</module>

- pom.xml in the folder 52n-sos-service: go to profile section with profile 'with-deploy' ad change the dependency from <artifactId>52n-sos-dao-postgis</artifactId> to <artifactId>eea-sos-dao-mssql</artifactId>

Then you can deploy the SOS with your DAO module type 'mvn -Pwith-deploy install' .

When you run into an error like this

> \*svn: 'C:\A\SensorWeb\workspace\SOS\eea-sos-dao-mssql' is not under

> version control\*

Here is the solution, which disables the automatic build number generation.

- open the SOS - pom.xml and move to build-section

- comment the whole plugin with the atifactId 'buildnumber-maven-plugin'

by using <!-- -->

To be able to develop new code in we will want to be able to use break points in the code and step through the code line-by-line.

According to Carsten at 52N this is setup the following way.

Here are the steps to debug the SOS with Eclipse. It is not comfortable but we have not found a better solution for debugging with Maven.

We use 'remote debugging' and here are the settings:

- Tomcat 6.0.xx :

- Tomcat Properties (in Win start tomcat6W.exe) and select tab 'Java'

- Add to Java options:

-Xdebug

-Xnoagent

-Xrunjdwp:transport=dt\_socket,server=y,suspend=n,address=11111

- Restart Tomcat

- Deploy SOS

- Eclipse:

- Run → Debug Configurations → ‚Remote Java Applications

- Connect tab

- Project: select the SOS

- Host: the SOS host (e.g. localhost)

- Port: 11111

- Apply or Debug

You can set the break points and debug the SOS with sending an Request or reloading the SOS if there are starting/configuration problems.

After changes to SOS code:

- stop Tomcat

- delete the SOS war file and the SOS directory in Tomcat webapps

- start Tomcat

- start remote debugging

Undeployment does not work because of Tomcat blocks some Jars in the SOS directory. The directory is not deleted and if you try to deploy the SOS an existing directory error occurs.

# Experiences

## Challenges

Working on code that is not very thoroughly documented

It has been a challenge to work with another organisations working code. The result has been reoccurring disturbances where an Update from SVN trunk lead to code that would compile.