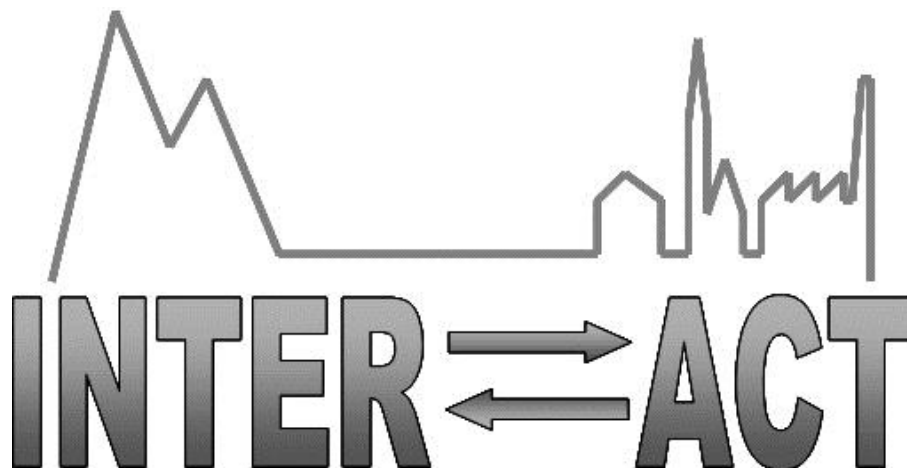


Combination of CP & CSA



D7.3- ScanDB V0 (Software)

Project No.262693– INTERACT

FP7-INFRASTRUCTURES-2010-1

Start date of project: 2011/01/01
Due date of deliverable: 31/12/2012

Duration: 48 months
Actual Submission date: 04/01/2013

Lead partner for deliverable: ITU
Author: Philippe Bonnet

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	Confidential, only for members of the Consortium (including the Commission Services)	

Table of Contents

Publishable Executive Summary	3
1. Introduction	4
2. Installation Requirements	4
3. ScanDB V0 Installation	4
4. Main Features	5
5. Conclusion	6
Appendix A. Technical Documentation	8

Publishable Executive Summary

ScanDB is a tool to help scientists to store time-series coming from different sources and with different properties. Scientist usually collect data from many different sources that may vary in position, source, weather conditions, etc. Each source is likely to have different properties such as air temperature, wind direction or snow depth. Typically, this data is usually gathered in spreadsheets and self-made databases which are then used to summarize and analyze it. Since there is no standard way of describing how to work with scientific data, each scientist defines his/her own workflow. This makes it hard for many scientists to work together and share data. ScanDB V0 aims to help scientists in this matter, allowing them to import data sets into a repository, annotating changes and keeping track of how data is updated with time. In this way, scientists can work directly with the data stored in the repository, avoiding "information specialists" and other middlemen, and increasing information awareness for other scientists. Also, by increasing provenance information, scientists can better determine the quality and lineage of the data in the repository. The ScanDB V0 software has been designed and is now available at <http://www.itu.dk/people/phbo/ScanDBV0.msi>

1. Introduction

ScanDB is a tool to help scientists to store time-series coming from different sources and with different properties. Scientist usually collect data from many different sources that may vary in position, source, weather conditions, etc. Each source is likely to have different properties such as air temperature, wind direction or snow depth. Typically, this data is usually gathered in spreadsheets and self-made databases which are then used to summarize and analyze it (see D7.1 for details).

Since there is no standard way of describing how to work with scientific data, each scientist defines his/her own workflow. This makes it hard for many scientists to work together and share data. ScanDB V0 aims to help scientists in this matter, allowing them to import data sets into a repository, annotating changes and keeping track of how data is updated with time. In this way, scientists can work directly with the data stored in the repository, avoiding "information specialists" and other middlemen, and increasing information awareness for other scientists. Also, by increasing provenance information, scientists can better determine the quality and lineage of the data in the repository. The design of ScanDB V0 targets items 1 and 2 of the functional requirements listed in D7.1.

The ScanDB V0 software has been designed and is now available at <http://www.itu.dk/people/phbo/ScanDBV0.msi> This report will guide INTERACT members to get started with ScanDB V0.

2. Installation Requirements

The requirements for installing ScanDB V0 are the following:

- .NET 4 or superior. (It is normally installed from Windows 7 onwards)
- SQL Express

If these are not installed, please follow the manuals provided with them and proceed with a standard installation.

3. ScanDB V0 Installation

There are two ways to install ScanDB:

1. The first way is to use directly the GIT repository where ScanDB is stored and open the project with Visual Studio. This option is not recommended for non-technical users. The advantage of this option is always counting on the most updated version of ScanDB V0. Versions match at the point ScanDB is delivered.
2. The second way is to download the pre-compiled version of ScanDB V0 (also known as executable) and run it. The ScanDB V0 executable can be found at: <http://www.itu.dk/people/phbo/ScanDBV0.msi>

Whichever way the user chooses, the steps to get SCANDB up and running are as follows:

1. Go to C:/Program Files/ScanDBV0/ScanDBV0 and click EcoManager.exe.config
(Please note that the PATH to ScanDBV0 might vary, but still should be very similar to this one)
When opening the file, we would see something like this:

```
<appSettings>
<add key="CMDSQLPath" value="C:\Program Files (x86)\Microsoft SQL
Server\100\Tools\Binn"/>
<add key="TMPPath" value="D:\temp"/>
<add key="ConnectionString" value="Data Source=(local)\SQL2008;Initial Catalog=DMU;
Integrated Security=True;Persist Security Info=False;Pooling=True;Min Pool Size=2;
Max Pool Size=1000;Load Balance Timeout=120;Packet Size=4096;Connection Timeout=300"/>
</appSettings>
```

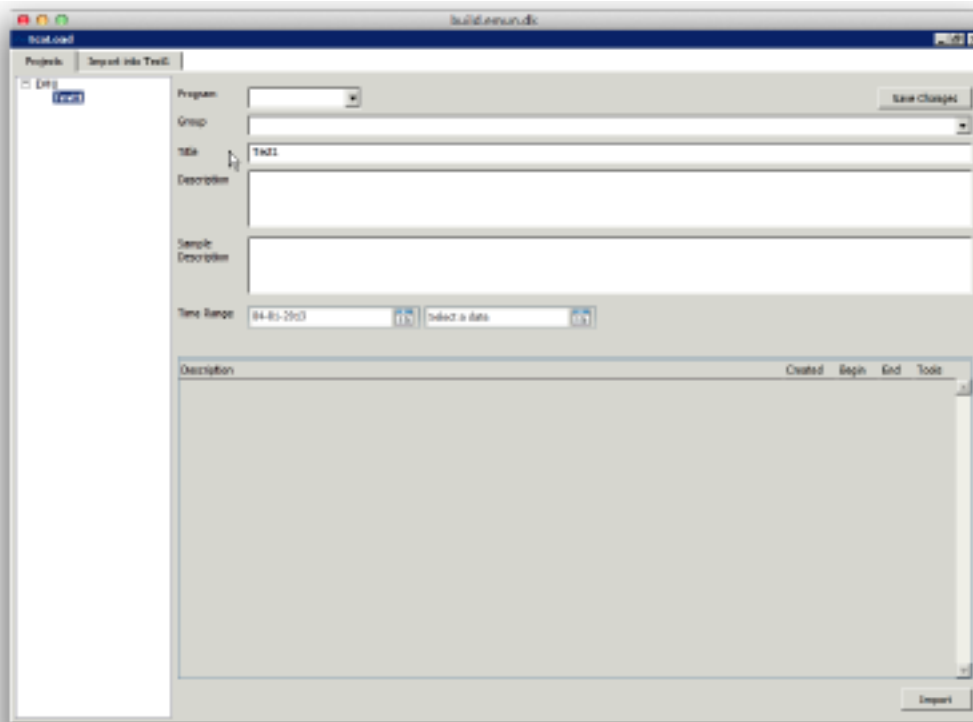
Now we should:

- Change CMDSQLPath = Look for SQLCMD and provide path to it. Typical path is provided)
- Create C:/Temp folder and update the temporary TMPPath
- Create the SCANDB Database. For this you can use any tool (e.g. the one provided with SQL Express). Use the script for creating the database. It can be found at C:/\$INSTALATIONDIR/SCANDBV0/Scripts/CreateDatabase.sql. Apply the script.
- Change ConnectionString to match your database instance

At this point SCANDBV0 is installed and fully functional. Look for it in Program Files and start ScanDB.

4. Main Features

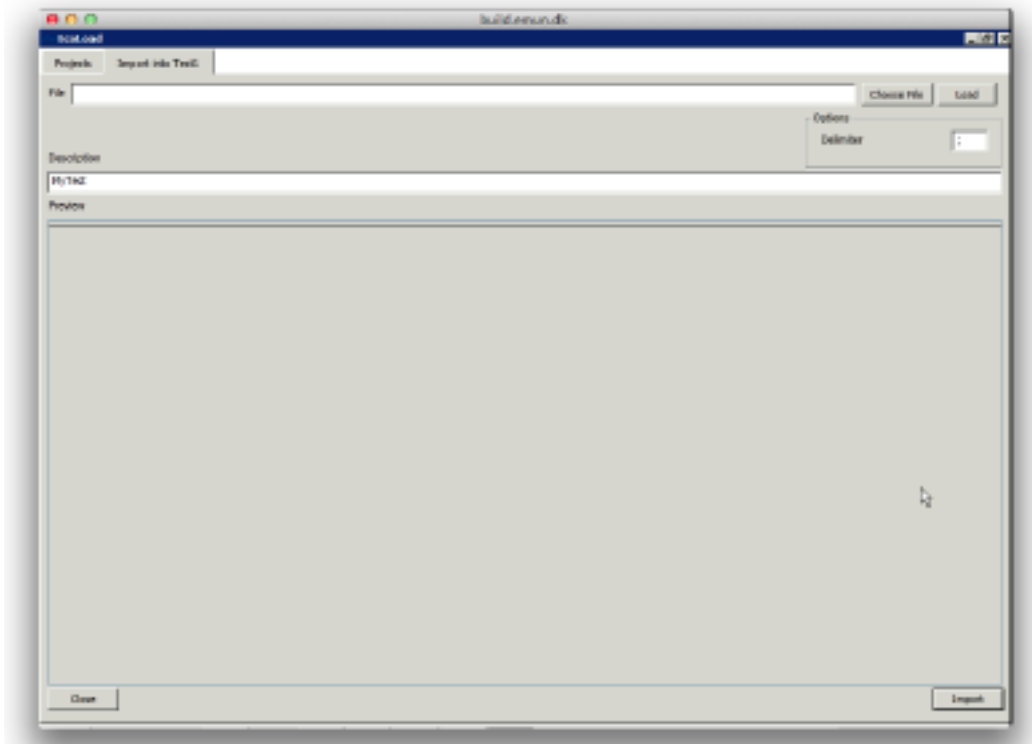
Once ScanDB is started, we should get an interface like this:



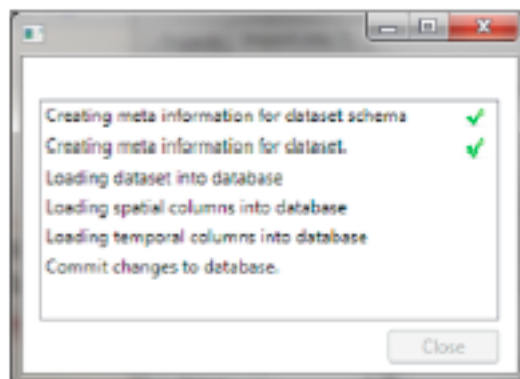
How to create a dataset group. Right click on the left menu and select create dataset. Fill out the information of the dataset.

How to remove a dataset group. Right click on the dataset that should be deleted and select delete.

How to export a dataset. Double click on a dataset and select export. Select a location to export it.
How to import a dataset. Select a dataset and click import on the lower right corner. A window like this will appear:



Select a file to import and click load. Change the types, according to the data in the dataset (e.g. integer, text, etc). Finally click import. The dataset has been imported.



How to update a dataset. Select a dataset and import it again with the added/removed rows. Datatypes will be remembered from the last import.

5. Conclusion

ScanDBV0 is a tool to help scientists to store time-series coming from different sources and with different properties. This tutorial contains the basics to get started with it, going from the installation to the most common tasks the tool is going to be used for.

Appendix A. Technical Documentation

The abstract database design uses XML, temporal and spatial types. The chosen database must therefore have some support for these features. More specifically we look for built-in support for:

- XML data type with indexes and query engine support.
 - Create C:/Temp folder and update the temporary TMPPath
 - Spatial data types, primarily geographical (spherical) data type.
 - Temporal data type with timezone support. Timezone is important if measurements are retrieved in GMT format and must be translated appropriately to the timezone where the user is located.
- All three data types should have some index support to speed up queries. Besides these specific requirements, the DBMSs of course should be mature and production ready.

We have chosen Microsoft SQL Server as the database platform. The small size of the database does not require support for advance partitioning as the Oracle and IBM alternative does. We do however use an ORM framework that supports multiple databases and make it possible to switch later. The abstract schema closely follows how the logical database design would be. We only have to define what data types each attributes should have, foreign keys and set up the necessary relationships. Figure 10 shows the data types that were chosen. The cascade options have been created for all tables. If a data set is deleted, the database maintains consistency without ScanDB or other has to maintain it. The XML attributes are created inside table Tables with the attribute name Storage. The support tables Annotations, SpatialInfo and Temporal* all have the key TableId, RowNr and TemporalGroup to uniquely identify a row in the XML storage. All the identifiers are 32bit signed integers. The natural limit on the number of rows available in the metadata and storage is approximately 2 billion. If it becomes a problem, the selected tables may be extended to 64bit or larger integers. This database implementation now acts as the repository and further work is client side and interacting with the repository and creating the necessary structures.

