

Software Maintenance and Evolution

Manual

Robbert-Jan Pijpker
s1982761

Joost Koehoorn
s2041014

November 2014

1 Deployment

This section shows how to deploy SolidTA and what dependencies it uses in order to be deployed:

1.1 Dependencies

SolidTA has the following dependencies:

- Python 2.6
 - psyco
 - py2exe
 - pysvn
 - PyOpenGL
 - pypng
 - pysqlite
 - wxPython
 - pywin32
 - *note*: The installers of all these packages are available in `.\Dependencies`
- NSIS installer (<http://nsis.sourceforge.net>)
- SolidSX (<http://www.solidsourceit.com/products/SolidSX-source-code-dependency-analysis.html>)
- SolidTFSImporter
 - Microsoft Visual Studio 2010
 - .NET 4.0

1.2 Create installer

After installing all necessary dependencies (except for SolidTFSImporter, see the next section for information on how to use this), running the following two commands inside `SolidSTA\deploy` will create the installer `SolidSTA\deploy\SolidSTA_install.exe` for SolidTA:

```
.\deploy_Win32.bat  
.\create_Win32_Installer.bat
```

1.3 Install SolidTFSImporter

Open the Solution files in Visual Studio and go to menu “Build → Build Solution”. To copy `SolidTFSImporter.exe` into the support directory of SolidTA, run `.\copy_SolidTFSImporter.bat Release`—or change `Release` to `Debug` depending on your build settings—in the folder `SolidSTA\deploy`. This is done automatically when deploying all of SolidTA and is thus only necessary during development.

SolidTFSImporter has a dependency on both `CSharpMetrics` (for file-by-file analyzation of simple complexity metrics) and `CSharpDeps` for dependency analysis. These projects are available in a different VS solution in `.\CSharpMetrics` and have to be build first in order for SolidTFSImporter to build. The dll-files of these projects have to be added to the SolidTFSImporter project’s references.

2 Development

Running SolidTA is possible without having to deploy it every single time. In `.\SolidSTA`, you can use the command `C:\Python26\python.exe -u SolidSTA.py` to start SolidTA.

3 Usage with SolidSX

After running SolidTA the main screen is shown. By choosing either “File → New” or “File → Open”, a new analysis project can be created, or an existing analysis project can be opened. For new projects, the file tree and file versions have to be downloaded first by means of the buttons in the lower left. Downloading files is not necessary for dependency analysis and may be skipped.

Now the dependencies should first be extracted and saved into a local database. To do so, got to “Tools → Compute metrics...” and locate “TFS C# dependencies” in the center column. Now click extract and fill out the dialog as shown in Figure 1.

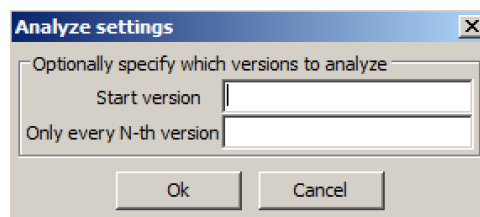


Figure 1: Analyzation settings dialog

Both values are optional and may be left blank. Specifying that only every N -th commit is analyzed is advisable to obtain results in a timely matter, because then a lot of intermediary commits are not analyzed individually. This process may take several hours, depending on the number of projects in the repository and number of commits. Note that doubling N will decrease the processing time needed by half, so choose wisely to achieve acceptable processing times.

Once this process has completed SolidSX can be opened by navigating to the tab **Dependencies** and pressing the button **Start SolidSX**.

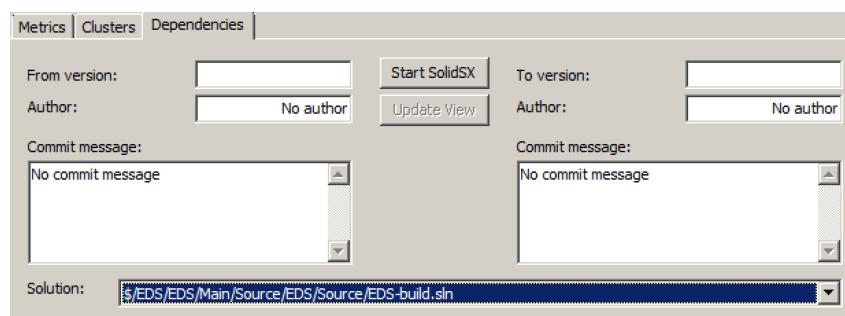


Figure 2: Specifying a range of versions to show only the edges which have been changed during that period.

After SolidSX has been started, the fields **From version** and **To version** in SolidTA can be used to filter the dependencies in SolidSX, after which **Update View** has to be pressed to update SolidSX. The dropdown menu in the bottom can be used to switch between different Solution files (see Figure 2). Selecting a different Solution file will immediately update SolidSX, not further action is necessary.

SolidSX offers extensive adjustment abilities and allows for various filtering options to narrow down the view. Besides the classical class/method hierarchy there is also an overview showing

the solution/project/file breakdown and then the dependencies between files may be observed. To accomplish this, under “Node selection” in the right sidebar switch to Node hierarchy 2.

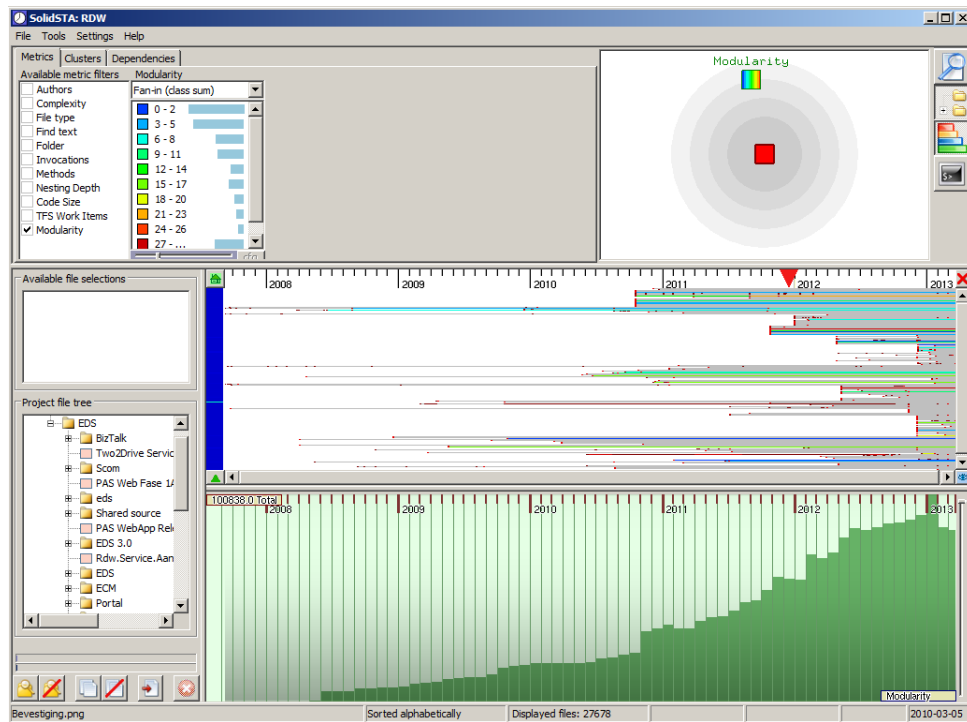


Figure 3: Example of SolidTA usage with SolidSX

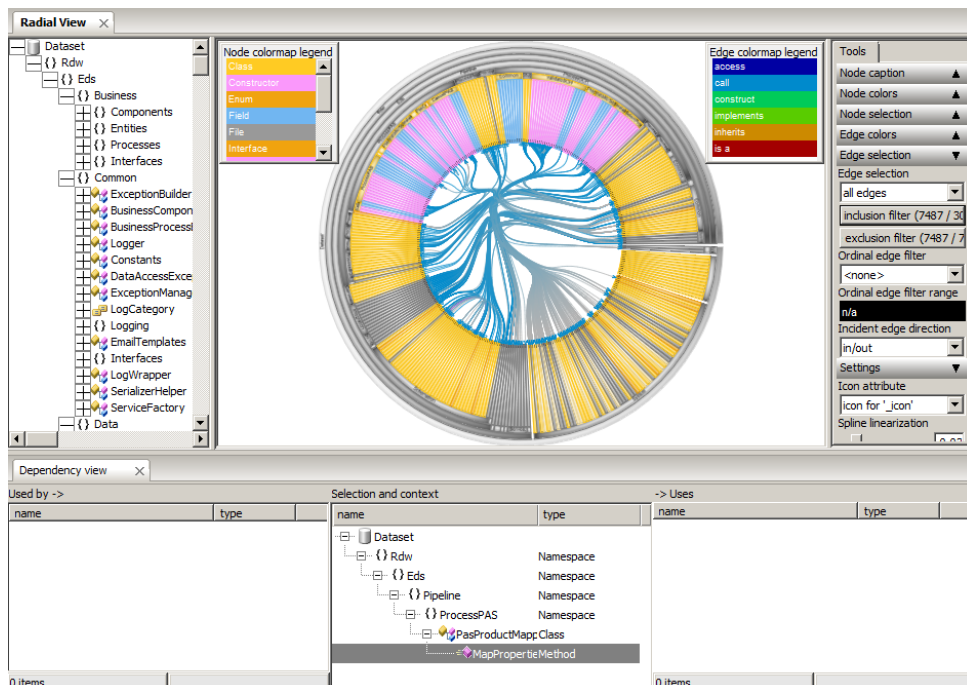


Figure 4: Example of SolidSX showing only call relations