SCIDIP-ES Gap Identification Service Installation Quick Start

Prerequisites

- JAVA SE Development Kit (at least version 7)
- Apache Tomcat Server (at least version 7)
- Working instance of opensource Virtuoso[1], (at least version 6.0) - only if the persistence layer that will be used is based on Virtuoso
- Source code for Gap Identification Service is available at: http://sourceforge.net/p/digitalpreserve/code/HEAD/tree/SCIDIP-ES/software/services/GapIdentificationService/tags/gapidentificationservice-2.5/

Customise configurations for local environment

After deploying GapIdentificationService webApp, the information about the persistent layer that will be used can be configured. The configuration file can be found at $TOMCAT_HOME/webapps/gapidentificationservice-webapps/WEB-INF/classes/eu/scidipes/impl/gapidentificationservice/server/config.properties. As described in the installation manual of Gap Identification Service there are two different persistence layers that can be used; a mainmemory layer and a persistent layer based on virtuoso triplestore. Below we describe the required updates for the different persistence layers. Note that there aren't any functional differences in the web application when using any of the persistence layers.

Using the main-memory persistence layer.

1. Update the value of the property `runmode` to `sesamesail` as shown below. This will set the persistent layer to the main memory store.

```
eu.scidipes.impl.gapidentificationservice.runmode=sesamesail
```

2. Afterwards the user can define the data that will be initially imported in the Gap Identification Service knowledge base. More specifically the user can define a comma-separated list of urls containing RDF data for modules and profiles in the properties `modules` and `profiles` respectively.

```
eu.scidipes.impl.gapidentificationservice.mainmemory.modules=http://athena.ics.forth.gr:9090/Applications/GapManager/data/ESA/ModulesEdited.rdf.xml,http://athena.ics.forth.gr:9090/Applications/GapManager/data/ESA/ModulesPackages.rdf.xml

eu.scidipes.impl.gapidentificationservice.mainmemory.profiles=http://athena.ics.forth.gr:9090/Applications/GapManager/data/ESA/Profiles.rdf.xml
```

[1] http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/

Using the virtuoso persistence layer.

1. Update the value of the property `runmode` to `virtuoso` as shown below. This will set the persistent layer to the virtuoso triplestore.

```
eu.scidipes.impl.gapidentificationservice.runmode=virtuoso
```

2. Afterwards the user must define the details for connecting to a working virtuoso triplestore. In particular the following information should be defined: host and port of the virtuoso running instance, the valid credentials for connecting to the triplestore and the name of the graphspace under which the GapIS data will be stored/retrieved.

```
eu.scidipes.impl.gapidentificationservice.virtuoso.url=http://
localhost/
eu.scidipes.impl.gapidentificationservice.virtuoso.port=1111
eu.scidipes.impl.gapidentificationservice.virtuoso.graph=http:
//www.scidip-es.eu/gapisgraph
eu.scidipes.impl.gapidentificationservice.virtuoso.username=vi
rtuoso_user
eu.scidipes.impl.gapidentificationservice.virtuoso.password=vi
rtuoso_pass
```

Harvesting data from the SCIDIP-ES registries

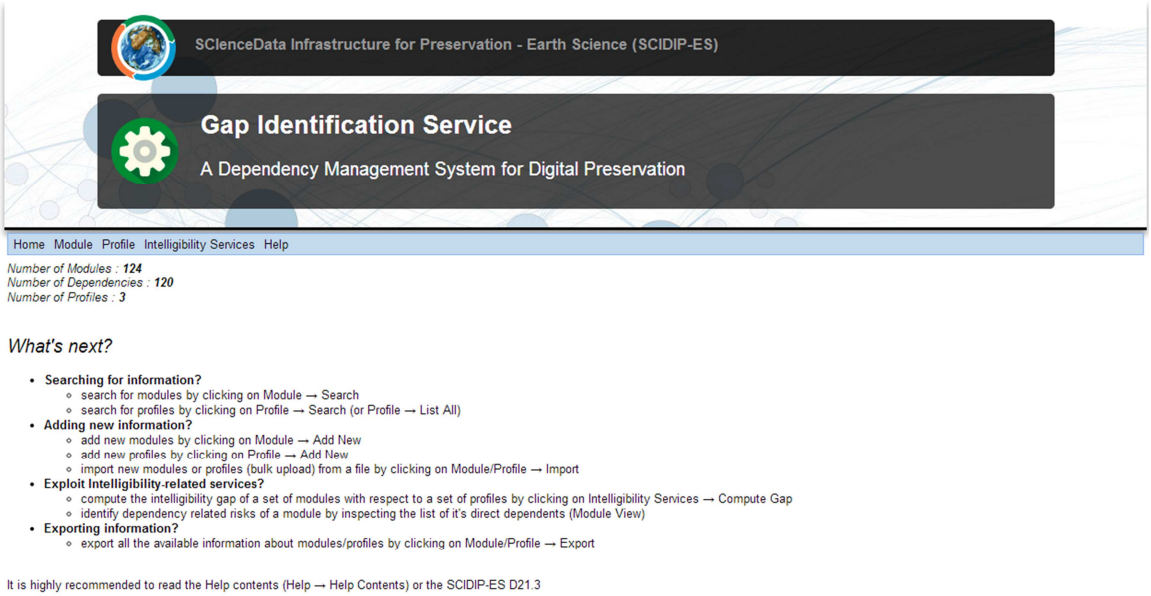
The user can alternatively add data coming from the SCIDIP-ES registries. However this requires extracting the available RepInfoLabels from the registries and then import them in the knowledge base of GapIdentificationService webApp. To do so execute the application found at http://sourceforge.net/p/digitalpreserve/code/HEAD/tree/SCIDIP-ES/software/services/GapIdentificationService/trunk/RegistryExporterForGapIS/. This application will produce an RDF file containing modules that can be afterwards imported in the GapIS webApp (through the menu Modules → Import → from RDF/XML) .

Test the installation

After deploying the GapIdentificationService webApp on tomcat the user can visit the homepage at

http://*<tomcat_host>*:*<tomcat_port>*/gapidentificationservice-webApp/

The home page should be displayed:



In the home page, just below the menu bar the total number of modules, dependencies and profiles that have been imported in the knowledge base will be reported, indicating that the data have been successfully loaded.