
PANACEA tutorial

WP 3

Author	Marc Poch
Affiliation	UPF
Status	internal report
Version	1.3
Release	draft
Date	2011-04-29
History:	

Abstract: This document is a short tutorial for PANACEA users.

Contents

1. Introduction.....	2
2. Users.....	2
2.1. PANACEA users.....	2

2.1.1.	Find web services (the registry)	3
2.1.2.	Run web services	3
2.1.3.	Design and run workflows.....	3
2.1.4.	Find and share already designed workflows	3
2.2.	PANACEA Service Providers.....	3
2.2.1.	Deploy web services	3
2.2.2.	Run web services	3
2.2.3.	Register and annotate web services.	3
2.2.4.	Design and run workflows.....	3
3.	Travelling Object.....	3
4.	Common interfaces	4
5.	FAQs	4
5.1.	Soaplab FAQs.....	4
5.1.1.	General	4
5.1.2.	Installation / ACD files development.....	4
5.1.3.	Deployment and configuration	4
5.2.	Taverna FAQs.....	5
5.2.1.	General	5
5.2.2.	Designing workflows	6
5.2.3.	Running workflows.....	6
5.3.	PANACEA registry (Biocatalogue instance) FAQs	7
5.3.1.	Registering web services	7
5.3.2.	Searching web services.....	7

1. Introduction

This document is a tutorial for PANACEA users. It can be used as a guideline for users to access all the necessary information and material to start working in PANACEA.

2. Users

PANACEA users can be divided in two groups: “PANACEA Users” and “PANACEA Service Providers”.

2.1. PANACEA users

These users are the ones willing to obtain language resources, analysis, etc. They want to make use of the tools but they don't have or don't need a deep knowledge of the technologies behind the web services. They make use of the tools because of their functionalities. Their basic tasks would be:

2.1.1. Find web services (the registry)

The first point of entry to the platform is the registry. It is where users are going to find the web services providing their desired functionalities. PANACEA registry can be found on <http://registry.elda.org>.

2.1.2. Run web services.

Once an interesting web service is found the user may want to try it before using it in a workflow or larger experiment. Or simply the desired task requires only one web service so a simple execution is required. Since most of the web services in PANACEA are deployed using Soaplab the best way to test them is using the “Spinnet” web client. Spinnet web client is a simple web form and it can be accessed following the link “Spinnet:” on the registry or using something similar to <http://{host}/soaplab2-axis/>. For example, “hunalign” web service can be found on the registry and it’s wsdl is: <http://www.cngl.ie/panacea-soaplab2-axis/services/panacea.hunalign?wsdl>. The url of the spinet of that web service provider is <http://www.cngl.ie/panacea-soaplab2-axis>.

2.1.3. Design and run workflows.

Once some web service have been tested and fulfill the desired expectations of the user they can be combined in a workflow. “Taverna” is the tool used in PANACEA to create and run workflows. It can be downloaded from <http://www.taverna.org.uk/> where manuals and tutorials can be found.

Tips and interesting links: see **5.2 Taverna FAQs** below.

2.1.4. Find and share already designed workflows

Users may want to share their web services or find already designed ones. PANACEA has a social network portal to share, find and discuss workflows: <http://myexperiment.elda.org>.

2.2. PANACEA Service Providers

PANACEA Service Providers are researchers, institutions, companies, etc. who want to make their tools available for everyone.

2.2.1. Deploy web services

PANACEA has chosen Soaplab to be the software to deploy web services. It can be used to deploy command line applications as web services. Other technologies like Apache Axis or apache CXF¹.

Tips and links: see **5.1 Soaplab FAQs** below.

2.2.2. Run web services

(PANACEA users section)

2.2.3. Register and annotate web services.

Service Providers must be registered to the Registry to operate. All web services should be registered and annotated (categorized and tagged) so users can find them.

Tips and links: see **5.3 PANACEA registry (Biocatalogue instance) FAQs** below.

2.2.4. Design and run workflows.

(PANACEA users section)

3. Travelling Object

Interoperability between web services is granted by using a common data format. This format is called “Travelling object” and is defined in [D.3.1](#) Section 6.1. *Travelling object. Corpus and data format*.

¹ <http://axis.apache.org/axis2/java/core/> and <http://cxf.apache.org/>

4. Common interfaces

Common interfaces are used to make web services easily interchangeable. All web services of a kind share the same interface at some level. In PANACEA, this means they have the same mandatory parameters. The Common Interface documentation can be found in <http://www.panacea-lr.eu/en/info-for-professionals/documents/>.

5. FAQs

These FAQs can be found also on PANACEA website:

<http://panacea-lr.eu/en/info-for-professionals/faqs/>

5.1. Soaplab FAQs

5.1.1. General

- **I have a command line application. How can I make it a web service?**

There are different solutions. We're using Soaplab in PANACEA.

- **What is Soaplab?**

It's a software tool to automatically turn command line applications into web services.

- **Where can I get Soaplab?**

Soaplab software and documentation can be found in <http://soaplab.sourceforge.net/soaplab2/>

- **Which Soaplab version is PANACEA using?**

The latest! Now, soaplab2-2.3.1

- **Which web application container / server do I need to use to deploy soaplab web services?**

Apache Tomcat. In PANACEA we are using Tomcat 6. <http://tomcat.apache.org>

5.1.2. Installation / ACD files development

- **When I use *ant* commands I get timeouts when connecting to other sites?**

Probably you're behind a **proxy**. Use **ant -autoproxy** instead of just **ant** in all your commands.

- **How can I create my ACD files?**

Follow the instructions in <http://soaplab.sourceforge.net/soaplab2/MetadataGuide.html>

- **Where do I put my ACD files describing my command line applications?**

In **src/etc/acd/sowa** folder in your soaplab2 installation folder.

- **What kind of input parameter should I use for large data?**

In PANACEA, we use **infile** parameters with "**comment: default_for_direct**". This way the input data can be provided with a **URL** (link) or direct data (a stream of characters). Direct data in Spinet web client can be provided directly typing in a text box or by uploading a file with "Browse..." button.

5.1.3. Deployment and configuration

- **How can I create the necessary XML metadata files from ACD files?**

Use **ant gen** to create metadata from ALL ACD files. In PANACEA we use **ant gensowa** to only generate XML from files in the SOWA directory (**src/etc/acd/sowa**).

- **Which protocol should I use: *axis1* or *jaxws*?**

In PANACEA, we use axis1 because Taverna workflow editor is not 100% compatible with jaxws.

- **How can I deploy web services?**

Use **ant jaxdeploy** or **ant axis1deploy**. In PANACEA, we use **ant axis1deploy** because axis1 is compatible with Taverna.

- Should I use typed interface?

Yes. More applications will be able to call your web services.

<http://soaplab.sourceforge.net/soaplab2/TypedServices.html>

- How can I configure soaplab to create web services with typed interface?

Set `typedinterface.enable = true` in your `build.properties` file.

- How can I create the WAR file without automatically deploying it?

Use `ant jaxwar` or `ant axis1war` instead of `ant jaxdeploy` or `ant axis1deploy`.

Running the soaplab services

- How can I test my soaplab web services?

An easy way to test them is by using the Spinet web client: <http://localhost:8080/soaplab2/>

or `http://{your_host}:{your_port}/soaplab2`

- How can I run soaplab web services? Which web services clients can I use?

You can use **Taverna** workflow editor (<http://www.taverna.org.uk/>), **spinet** web client, **perl** programs

like: <http://www.ebi.ac.uk/Tools/webservices/soaplab/guide> and there's **more** information in

<http://soaplab.sourceforge.net/soaplab2/ClientsOverview.html>.

- What is *detailed_status* output?

It indicates the exit status of your command line program. If a program ends correctly its exit status is 0 otherwise its value is 1. Soaplab captures this value and uses `detailed_status` to show it. If exit status is 1, soaplab will end with an error message.

5.2. Taverna FAQs

5.2.1. General

- Which taverna version are PANACEA members using?

Taverna 2.2.0

- Where can I find documentation about Taverna workbench, local processors, etc.?

<http://www.mygrid.org.uk/dev/wiki/display/taverna/Documentation+and+Videos>

5.2.2. Designing workflows

- How can I import soaplab web services from a service provider?

Use soaplab option after "Import new services" button. If the soaplab server you want to use is like `http://{host}/soaplab2-axis`. Then the address for Taverna is `http://{host}/soaplab2-axis/services/`. Don't forget the final slash (/).

- I'm trying to add a soaplab server and I get an error message?

Be sure you add the final slash to the address! `http://{host}/soaplab2-axis/services/` .

Verify the spinet web client is working.

- What should I use to connect the inputs and outputs from different soaplab processors: *output* or *output_url*?

In PANACEA, we recommend `output_url` to deal with big files.

- Can I inspect PANACEA catalogue from taverna?

Yes! Go to your Taverna *Preferences* and configure the Biocatalogue instance you want to use.

- How can get specific information from an xml data file? Is there a local processor to do that?

There are two options:

1. Use the "Xpath from text" local processor which will require a little xpath knowledge.

<http://www.mygrid.org.uk/dev/wiki/display/taverna/XPath+From+Text>

2. The “**Xpath service**” from service templates. Really easy to use and no xpath knowledge needed. You may need to update Taverna to be able to use this feature (Advanced menu -> updates). <http://www.mygrid.org.uk/dev/wiki/display/taverna/XPath>

- I’m working with some processors which outputs are lists. This is creating a problem of undesired nested lists. How can I solve this?

You can use “**flatten list**” local processor to reduce the list one level.

<http://www.mygrid.org.uk/dev/wiki/display/taverna/Flatten+list>

- I’m using nested workflows with list inputs. The output is creating several nested list with a lot of repeated outputs! What can I do?

The outputs you’re seeing are the result of processing all possible combination of inputs (cross product). You can get more information here:

<http://www.mygrid.org.uk/dev/wiki/display/taverna/List+handling>

If you want to pair up items from input lists you can follow these steps:

<http://www.mygrid.org.uk/dev/wiki/display/taverna/List+handling#Listhandling-Pairingupvalueswithdotproduct>

-I run a slow process or a big file without problems in Spinet but I get a timeout error in Taverna. What can I do?

Taverna has timeout limit. If the answer from the web service takes too long Taverna thinks something went wrong. To avoid this you must use “polling”. Follow these instructions:

http://www.mygrid.org.uk/usermanual1.7/soaplab_processor.html#soaplab_polling

-When there’s an error in one web service the data is lost. What can I do?

You should use “retries” to try to execute one web service more times in case it fails.

http://www.youtube.com/watch?v=-jWQH9_u370

Is there any way to execute task in parallel?

Yes! You can configure web services to be executed in parallel. You must be careful since machines have limited resources. Follow the examples and read the following document:

<http://taverna.knowledgeblog.org/2010/12/13/parallel-service-invocations/>

5.2.3. Running workflows

- **How can I improve Taverna performance (not web services performance!)?**

Deactivate “**in-memory storage**” and “Enable provenance capture” option in “**data and provenance**” on Taverna Preferences.

You can also assign more memory to Taverna following these instructions: <http://www.mygrid.org.uk/dev/wiki/display/taverna/Increase+memory+allocation>. We’re using the following configuration: `exec java -Xmx800m -XX:MaxPermSize=280m.`

- **Can I inspect intermediate results?**

Yes if you have that option activated on “**data and provenance**” on Taverna Preference. To inspect the results just click on the processor you are interested on the results view.

- **How can I count the execution time?**

In the results view click on the “**Progress report**” tab you’ll find detailed information about each processor / web service: iterations, **Average time / iteration**, etc.

5.3. PANACEA registry (Biocatalogue instance) FAQs

5.3.1. Registering web services

- How should I register my soaplab2 web services: SOAP(wsdl) or soaplab server?

You should use **soaplab server**. The registry will automatically gather all your web services and register them.

- Is it really necessary to categorize all my web services?

No, but it's recommended. Categorization helps users to find your services.

- Is it important to use tags?

Yes. Tags are a very versatile and useful tool to annotate web services. With tags users can find web services without being experts. For example, there's no need to know the category.

5.3.2. Searching web services

- How can I search for web services?

The Biocatalogue instance has tags, categories, providers, and more parameters to search web services.

The "Search" field on the web site can be used with wildcards to make more flexible queries i.e. asterisk (*) or question mark (?). For example **crawl*** would match **crawler** and **crawling**.

- Is it possible to know if a service is up and running?

Yes! Next to every service there's a "ball" indicating if the service is available. Move your mouse over it to get extra information like "last checked".