

**SIXTH FRAMEWORK PROGRAMME
INFORMATION SOCIETY TECHNOLOGIES**

IST-FP6-004559 SODIUM

“Service Oriented Development In a Unified fraMework”



SODIUM Platform Installation instructions

Identifier

Date:	January 2007
Author(s):	C. Pautasso (ETHZ), M. Pantazoglou (NKUA), R.Brown (ATC), Andreas Limyr (SINTEF)
Version:	1.0
Status:	Final
Abstract:	This document contains installation instructions for the SODIUM platform tools (incl. short descriptions of the components)



Revision History

Version Identifier	Revision Outline	Revision Author
0.1	First draft circulated to SODIUM members.	C. Pautasso (ETHZ)
0.2	Added input for the installation and configuration of the USQL Engine	M. Pantazoglou (NKUA)
0.3	Added input for the installation and configuration of the Composition Repository	R.Brown (ATC)
0.4	Added input for the Composition Studio	A. Limyr (SINTEF)
1.0	Final version	R.Brown (ATC)



Executive Summary

This document provides installation instructions for the SODIUM Platform in terms of its components:

- SODIUM Service Composition Suite (Studio)
- SODIUM Runtime environment:
 - SODIUM USQL engine
 - SODIUM USCL engine
 - SODIUM Composition Repository

In addition to installation instructions, each of these components is briefly described (in terms of functionality overview) in separate chapters of this document.

For a detailed presentation of the functionality of the various tools please refer to the report SODIUM Platform: Overview of functionality.



Table of contents

1	PROTOTYPE DESCRIPTION	7
1.1	Overview.....	7
1.2	SODIUM Service Composition Suite (Studio).....	7
1.2.1	Component Description	7
1.2.2	Installation instructions	7
1.3	Runtime environment.....	9
1.3.1	Overview.....	9
1.3.2	SODIUM Composition Repository	9
1.3.3	USQL Engine.....	12
1.3.4	USCL Engine.....	16
2	REFERENCES	18



List of Abbreviations

USCL	Unified Service Composition Language
USQL	Unified Service Query Language
VSCL	Visual Service Composition Language
WSDL	Web Service Definition Language



List of Figures

Figure 1-The USQL Engine’s Axis welcome page	14
Figure 2 -The USQL Engine’s offered Web Services	15
Figure 3 - Installing the SODIUM USCL Engine.....	16
Figure 4 - A successful installation of the USCL SODIUM Engine.....	17



1 PROTOTYPE DESCRIPTION

1.1 Overview

To install the Sodium platform the user will need to install and setup the development and the runtime environments.

The development environment is centred on Eclipse IDE (Integrated Development Environment) which allows the user to build, test and execute compositions. The tools accessed from within Eclipse are the Service Composition Suite (Studio), the USQL Dialog and the USCL Engine. The Composition Repository can be used in conjunction for the storage and organisation of compositions (if needed, optional).

The Runtime environment consists of the USCL Engine, the USQL Engine and the Composition Repository. The Composition Repository (if required, optional) is the place where compositions are stored and organised (e.g. a user would retrieve the compositions from it for deployment onto the USCL Engine.) The USCL Engine is integrated with the USQL Engine to allow compositions at runtime to discovery Services (regardless of its type e.g. web, grid or p2p).

1.2 SODIUM Service Composition Suite (Studio)

1.2.1 Component Description

The SODIUM Service composition studio is a development environment for modelling service compositions using a graphical editor. The modelling language is called Visual Service Composition Language (VSCL) and is based on UML 2 activity diagrams. The SODIUM Composition Studio consists of four main components:

- **The Visual Editor** for defining compositions of heterogeneous services in a unified way by using the SODIUM Visual Composition Language (VSCL). The Visual Editor implements the graphical VSCL language [D7].
- **The Language Translator.** This component transforms a VSCL model to USCL [D6] to produce a corresponding executable specification.
- **USQL Dialog.** This component is a dialog where the user can integrate with the USQL Editor to define, view and save USQL queries [D8] associated with tasks in the VSCL composition.
- **The Visual Composition Graph Analyser** which performs a syntactic check of the composition.

1.2.2 Installation instructions

It is possible to download all you need from <http://set.sintef.no/CS/update/workingCS.zip>. This is an “all-in-one” package containing everything you need to get started with the Composition Studio. Just unzip the file, find and execute the eclipse.exe file inside the eclipse folder and you are on your way.

To manually install the SODIUM Service Composition Studio, download these archive files and unpack them in an appropriate place. Just remember to synchronize the unpacking with



the tree structure provided by the Eclipse installation. (See to that there is only one feature folder and one plugins folder under the eclipse folder.)

- Eclipse 3.1.2
 - Windows
<http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.1.2-200601181600/eclipse-SDK-3.1.2-win32.zip>
 - Linux
<http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.1.2-200601181600/eclipse-SDK-3.1.2-linux-gtk.tar.gz>
 - Mac OSX
<http://download3.eclipse.org/eclipse/downloads/drops/R-3.1.2-200601181600/download.php?dropFile=eclipse-SDK-3.1.2-macosx-carbon.tar.gz>
- EMF SDK 2.1.2
 - All platforms
<http://www.eclipse.org/downloads/download.php?file=/tools/emf/downloads/drops/2.1.2/R200601191349/emf-sdo-xsd-SDK-2.1.2.zip>
- UML2 1.1.1
 - All platforms
<http://www.eclipse.org/downloads/download.php?file=/modeling/mdt/uml2-uml/downloads/drops/1.1.1/R200509282050/uml2-SDK-1.1.1.zip>
- GEF 3.1.1
 - All platforms
<http://www.eclipse.org/downloads/download.php?file=/tools/gef/downloads/drops/R-3.1.1-200509301327/GEF-ALL-3.1.1.zip>
- Web Standard Tools 1.0.2
 - Windows
<http://www.eclipse.org/downloads/download.php?file=/webtools/downloads/drops/R1.0/R-1.0.2-200604280245/wtp-all-in-one-sdk-R-1.0.2-200604280245-win32.zip>
 - Linux
<http://www.eclipse.org/downloads/download.php?file=/webtools/downloads/drops/R1.0/R-1.0.2-200604280245/wtp-all-in-one-sdk-R-1.0.2-200604280245-linux-gtk.tar.gz>
 - Mac OSX
<http://www.eclipse.org/downloads/download.php?file=/webtools/downloads/drops/R1.0/R-1.0.2-200604280245/wtp-all-in-one-sdk-R-1.0.2-200604280245-macosx-carbon.tar.gz>

After installing these dependencies it is possible to start Eclipse and do the rest of the download with help of the update manager. Using the update manager is also a good idea for those of you who downloaded the “all-in-one” zip file.

Step 1. Start Eclipse (and go to the workbench if not already there)

Step 2. Go to Help → Software Updates → Find and Install...



- Step 3. You should now see the same install dialog as the picture below. With use of the Import sites... functionality it is possible to import what you need. Paste <http://set.sintef.no/CS/update/bookmarks.xml> into the file name field and finish of the dialog.
- Step 5. Select Composition Studio, Jena and USQL Dialog update site (if not already selected) and unselect anything else.
- Step 6. Go to the next page.
- Step 7. Select Composition Studio, Jena and USQL Dialog update site
- Step 8. Go to the next page. Accept the terms of the license agreement and then finish.
- Step 9. After installation the workbench will ask to be restarted, agree to this.
- Step 10. Congratulations! You should now have completed your install of Composition Studio.

1.3 Runtime environment

1.3.1 Overview

The Runtime environment forms part of the overall SODIUM platform and is composed of the following components:

- the SODIUM Composition Repository,
- the USQL Query Engine and
- the USCL Execution Engine

Each prototype component of the SODIUM Runtime environment is briefly described in the following sections. More information on their specifications can be found in [D10].

1.3.2 SODIUM Composition Repository

1.3.2.1 Component Description

As described in [D10], the SODIUM Composition Repository enables users to store and manage the compositions that will be created by the use of the SODIUM composition suite, comprising of a storage area for compositions and a web enabled user interface for the management of this storage.

Composition Repository downloads:

All in one download: <http://www.atc.gr/sodium/tools/compositionRepository.zip>

Individual downloads:

Database scripts: <http://www.atc.gr/sodium/tools/scripts.zip>

Web services interface (optional): <http://www.atc.gr/sodium/tools/webservices.zip>

Web application: http://www.atc.gr/sodium/tools/web_app.zip



Required Software is as follows:

MySQL 4.1 Community Edition

URL: <http://dev.mysql.com/downloads/mysql/4.1.html>

Windows Downloads: Windows Installer (x86) 38.9MB

MySQL Connector/J – 3.1

URL: <http://dev.mysql.com/downloads/>

Java SDK 1.5 or greater

URL: java.sun.com

Tomcat 4.1.31 (Windows installer)

URL: <http://tomcat.apache.org>

Axis 1.2 or greater (Optional)

URL <http://ws.apache.org/axis/>

1.3.2.2 Installation Instructions

1.3.2.2.1 Installation of the Composition Repository Database

A) MySql Installation

Install MySQL using the installation wizard selecting “innodb” option.

MySql Installation Option - Choose “Complete” installation

MySql Registration - Skip Sign Up

MySql Server Instance Configuration Choose the following options when prompted

- 1) Maintenance Option - Reconfigure Instance
- 2) Configuration Type - Detailed Configuration
- 3) Server Type – Server Machine
- 4) Database Usage – Multifunctional Database
- 5) InnoDB Tablespace settings – press “Next”
- 6) Concurrent connections – Choose the appropriate setting
- 7) Network Options – Enable TCP/IP (default enabled)
- 8) Default character set – Standard CharacterSet
- 9) Windows options – Include BIN Directory in Windows Path
- 10) Security Settings – Modify the root password to your desired one

B) Create Database User “sodiumrepo”

1. Open the MySQL Command Line Client.
2. Enter the root password.



3. Enter the following commands:

- a. create database sodiumrepo;
- b. grant all on sodiumrepo.* to sodiumrepo@'%' identified by 'sodiumrepo';

Note: The password for the user “sodiumrepo” is “sodiumrepo”.

C) Create database schema

1. Open the file “scripts.zip” and extract the files to the hard disk.
2. Go to the command prompt (Start->Run and enter cmd).
3. Move to the directory where you extracted the scripts.
4. Enter the following commands `mysql -u sodiumrepo -p < schema.sql`

D) Create the default data

1. Open the file “scripts.zip” and extract the files to the hard disk.
2. Go to the command prompt (Start->Run and enter cmd).
3. Move to the directory where you extracted the scripts.
4. Enter the following commands `mysql -u sodiumrepo -p < data.sql`

1.3.2.2.2 Installation of the Composition Repository Application Server

A) Java SDK

Run the installation program for the Java SDK and follow the instructions.

B) Install Tomcat

Run the installation program for Tomcat and follow the instructions

C) Install MySQL Connector/J

Copy the `mysql-connector-java-3.1.13-bin.jar` file to the directory: `tomcat_installation_directory\common\lib`

D) Install the Composition Repository Application

Unzip the file `web_app.zip` and copy the directory “sodiumrepo” to the tomcat directory `tomcat_installation_directory\webapps`.

If the database server is on a different computer the database connection needs to be changed. Go to the directory `tomcat_installation_directory\webapps\sodiumrepo\WEB-INF` and open the file “site.properties” in an editor (Notepad). Find the following line:

```
sodiumrepo.url=jdbc:mysql://localhost:3306/sodiumrepo
```

“localhost” needs to be changed to the name or ip of the database server (MySQL).

Note: To access the Composition Repository from a browser the URL would be `http://servername_or_ip:port/sodiumrepo`. The default user is “user1” with password “user1”.



E) Install Axis (Web Services Interface) – Optional

Installation instruction included in the download. Hint: Copy the “axis” directory in “webapps” directory of the download to the webapps directory of tomcat and follow the instructions included in the download.

F) Install the Web Services interface – Optional (Requires step E)

Unzip the zip webservices.zip and copy the file to the directory tomcat_installation_directory\webapps\axis.

Now set the location of Composition Repository files. Go to the directory tomcat_installation_directory\webapps\axis and open the file “SodiumRepoWS.jws”. Find the following line:

```
static String FileDir = "C:\jakarta-tomcat-4.1.31\webapps\sodiumrepo\files";
```

Change the highlighted part to the installation directory of tomcat. Note the double backslashes are required when specifying the path.

If the database server is on a different computer the database connection needs to be changed. Go to the directory tomcat_installation_directory\webapps\axis and open the file “SodiumRepoWS.jws”. Find the following line:

```
static String DBURL = "jdbc:mysql://localhost:3306/sodiumrepo";
```

“localhost” needs to be changed to the name or ip of the database server.

1.3.3 USQL Engine

1.3.3.1 Component Description

The USQL Engine is the component responsible for discovering heterogeneous Web, P2P, and Grid services against heterogeneous registry mechanisms. More specifically, the USQL Engine framework is a Java-based product. It is being developed and tested with the use of *J2SDK 1.5.0 Update 5* [Java], but it is backwards compatible with any version starting from 1.4.x and forward.

The USQL Engine exposes its functionality through a set of web service interfaces and thus needs to be deployed to an appropriate web container. During development and testing, the current version was deployed on *Apache Tomcat 5.0.30* [Tomcat]. In any case, it should also be easily deployed in previous versions of Tomcat, like the 4.1.x series.

To summarize, the following versions are recommended for the successful installation of the USQL Engine as a web service:

- Java 2 SDK 1.5.0 Update 5
- Apache Tomcat 5.0.30

1.3.3.2 Installation Instructions

The process of installing the USQL Engine is straightforward. Provided that the previously outlined software has been installed and is running on your system, follow the steps below to complete the installation of the USQL Engine:



1. Download the “all-in-one” zip file of the USQL Engine from the following link: <http://www.di.uoa.gr/~michaelp/USQLEngine/core/usqlengine.zip>. This file contains everything that is required by the USQL Engine.
2. Unzip the contents of the zip file to directory \$TOMCAT_HOME/webapps.

The current version of the USQL Engine you just installed also contains the following plug-ins:

- *UDDI2Agent* for accessing uddi v2 compliant service registries
- *WSDLHandler* for handling WSDL 1.1 documents
- *WSDLSemanticsHandler* for handling WSDL-S documents
- *WSDLWSQoSHandler* for handling WS-QoS documents

A *JXTAAgent* for discovering services in JXTA networks is currently under development and will be available soon.

1.3.3.2.1 USQL Engine Web Services

The USQL Engine is shipped with the following web services:

- **ServiceDiscoverer** (URL = `http://<your_server >/usqlengine/servicediscoverer`) for applying service discovery
- **RegistryManager** (URL = `http://<your_server >/usqlengine/registrymanager`) for configuring the registry information that is required by the USQL Engine, so that it can access and query the various registries and networks
- **DomainAdministrator** (URL = `http://<your_server >/usqlengine/domainadministrator`) for managing the domain ontologies supported by the USQL Engine

In order to meet the respective security requirements, the *RegistryManager* and *DomainAdministrator* web services require HTTP authentication. You can create user accounts in Tomcat to enable access to these web services. To do so, you need to edit the file \$TOMCAT_HOME/conf/tomcat-users.xml and add one or more users which will play the roles of registry manager and/or domain administrator. An example is given below:

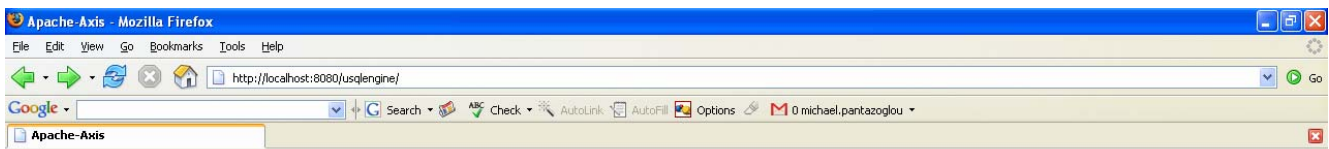
```
<user username="nkua" password="nkua"  
      roles="wsUSQLEngineRegistryMan,wsUSQLEngineDomainAdmin"/>
```

The bolded role names have been specified in the USQL Engine’s web configuration file, therefore they should not be renamed.

Restart Tomcat, open a web browser and enter the following url:

`http://<your_server >/usqlengine`

in order to verify that the USQL Engine was successfully installed. You should see the Axis welcome page (see **Error! Reference source not found.**):



Apache-Axis

Language: [en] [ja]

Hello! Welcome to Apache-Axis.

What do you want to do today?

- [Validation](#) - Validate the local installation's configuration
see below if this does not work.
- [List](#) - View the list of deployed Web services
- [Call](#) - Call a local endpoint that list's the caller's http headers (or see its [WSDL](#)).
- [Visit](#) - Visit the Apache-Axis Home Page
- [Administer Axis](#) - [disabled by default for security reasons]
- [SOAPMonitor](#) - [disabled by default for security reasons]

To enable the disabled features, uncomment the appropriate declarations in WEB-INF/web.xml in the web application and restart it.

Validating Axis

If the "happyaxis" validation page displays an exception instead of a status page, the likely cause is that you have multiple XML parsers in your classpath. Clean up your classpath by eliminating extraneous parsers.

If you have problems getting Axis to work, consult the [Axis Wiki](#) and then try the Axis user mailing list.



Figure 1-The USQL Engine's Axis welcome page

By clicking to the **List** link, you should see the 3 USQL Engine web services along with links to their respective WSDL descriptions:

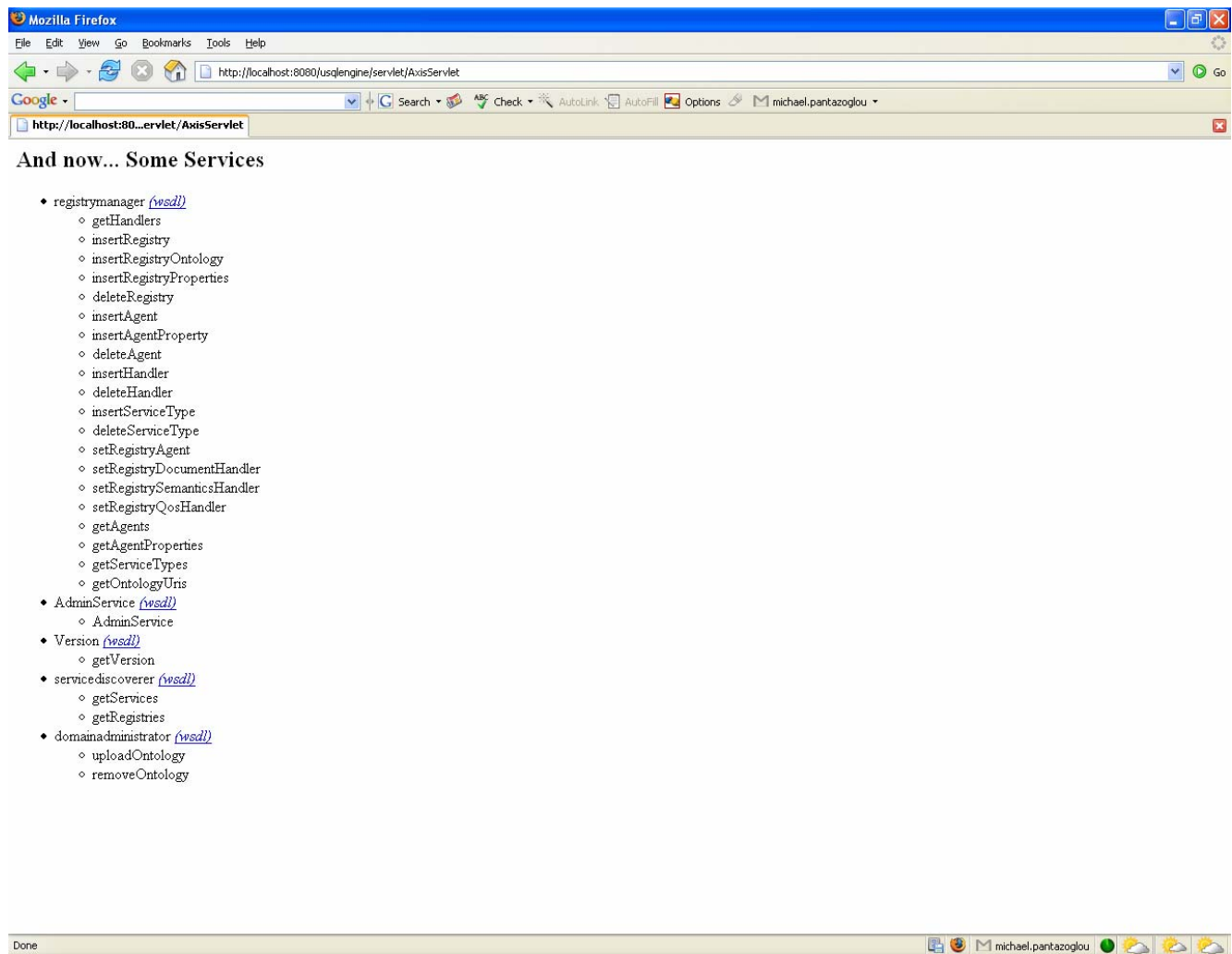


Figure 2 -The USQL Engine's offered Web Services

That's it! The installation is complete, so you should now proceed with configuring the USQL Engine.

1.3.3.3 Configuration

For simplicity purposes, in the following we will make use of a virtual variable `$USQL_ENGINE` when referring to path `$TOMCAT_HOME/webapps/usqlengine`.

Open the properties file of the USQL Engine, which is located in `$USQL_ENGINE/WEB-INF/classes/usqlengine.properties`. Edit the following properties in order to meet your environment settings:

- **edu.nkua.usql.engine.core.USQLEngine.datastoreConnectionUrl**

This should be set to:

`jdbc:hsqldb:file:$USQL_ENGINE/WEB-INF/datastore/usqlenginedb`

- **edu.nkua.usql.engine.core.USQLEngine.ontologyRepository**

This should be set to:

`$USQL_ENGINE/WEB-INF/ontologies`



Open the server configuration file `$USQL_ENGINE/WEB-INF/server-config.wsdd` and set the following parameter to be in accordance with your environment:

```
<parameter name="attachments.Directory"
  value="$USQL_ENGINE/WEB-INF/attachments"/>
```

The script file `usqlenginedb.script` in folder `$USQL_ENGINE/WEB-INF/datastore` contains the SQL statements that comprise the USQL Engine database. These statements populate the database with registry-specific information. Currently, the script contains statements that render accessible by the USQL Engine the test registry we have set in our premises. Also, the ontology repository contains a sample ontology named `healthcare.owl`. By making use of the *RegistryManager* and *DomainAdministrator* web services, you can add or remove registries and domain ontologies respectively from the USQL Engine.

1.3.4 USCL Engine

1.3.4.1 Component Description

The USCL Engine is the component responsible for the execution and publishing of the service compositions expressed utilising the Unified Service Composition Language (USCL) developed in the context of the project

1.3.4.2 Installation Instructions

The USCL engine is installed by pointing Eclipse to this update site (Use the Help, Software Updates, Find and Install, Search for new Features, Next, New Remote Site...)

<http://people.inf.ethz.ch/pautasso/update/sodium/>

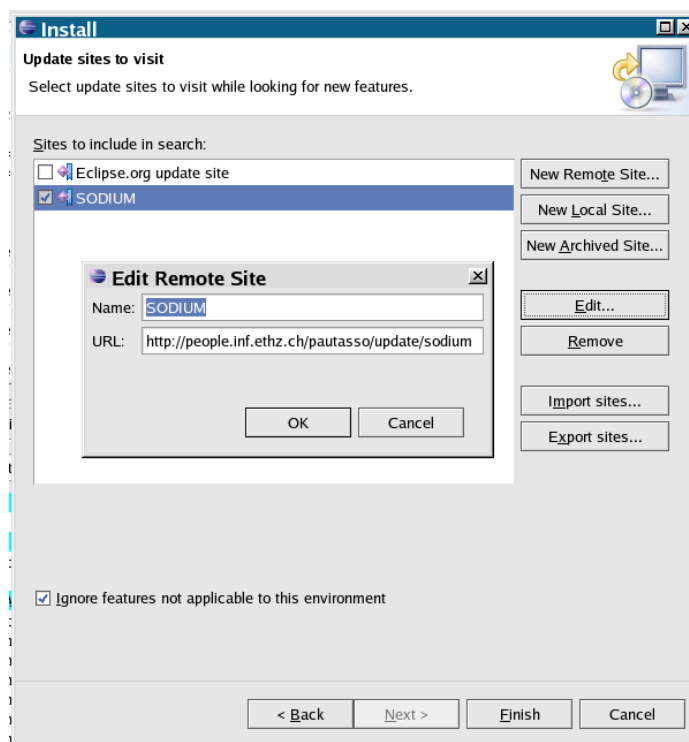


Figure 3 - Installing the SODIUM USCL Engine



After the "SODIUM USCL Engine" feature has been installed, the Eclipse workbench should be restarted. The engine can be controlled through the "Kernel Console" view under the "Other category"

Opening such view for the first time will trigger the creation of a new project in the workspace called "uscl", dragging and dropping USCL files into this project will automatically trigger the deployment of the contained processes into the USCL engine.

To test the successful deployment of a USCL composition, the: "show templates" can be typed in the console to list the name of the deployed processes. Processes can be started by typing the "start <processName>" command.

The following screenshot illustrates what should happen with a successful installation.

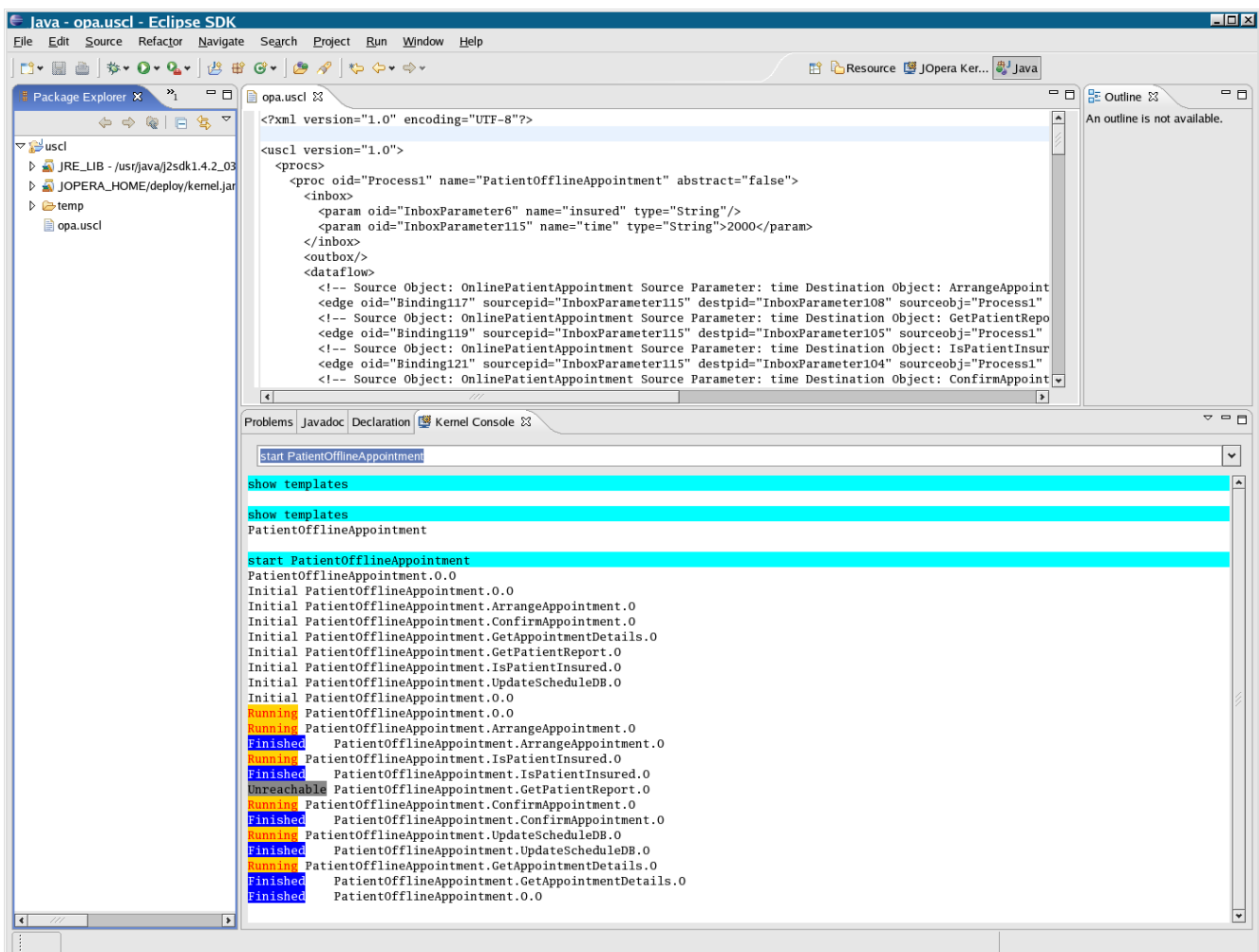


Figure 4 - A successful installation of the USCL SODIUM Engine

The USCL Engine also provides an embedded Web server (on port 8080), which is used to make the compositions accessible as a Web service. These are listed under:

<http://localhost:8080/wsdl?list>



2 REFERENCES

- [D3] SODIUM project Deliverable D3- Requirements Specification for Service Composition and Architecture of the SODIUM Platform Part II
- [D6] SODIUM project Deliverable D6: SODIUM Unified Service Composition Language Specification (USCL), Cesare Pautasso, Thomas Heinis, Gustavo Alonso, ETHz, version 1.5.
- [D7] SODIUM project Deliverable D7: SODIUM Visual Service Composition Language Specification (VSCL), Hjørdis Hoff, Roy Grønmo, David Skogan, SINTEF, version 0.9.
- [D8] SODIUM project Deliverable D8: Specification of the Unified Service Query Language (USQL), A. Tsalgatidou, M. Pantazoglou , G. Athanasopoulos,, NKUA, version 1.0, June 2005
- [D9] SODIUM project Deliverable D9: Detailed Specification of the SODIUM Composition Suite, David Skogan, Hjørdis Hoff, Roy Grønmo, Andreas Limyr, Tor Neple, SINTEF, version 1.2.
- [D10] SODIUM project Deliverable D10 - Detailed Specifications of SODIUM Runtime Environment
- [D18] SODIUM project Deliverable D18- SODIUM platform Overview of functionality
- [Java] Java 2 SDK, available at <http://java.sun.com/>
- [Tomcat] Apache Tomcat, <http://tomcat.apache.org/>