



## PEtALS Webconsole User Guide

*This document describes the installation and usage of the web monitoring and administration application of PEtALS*

PEtALS Team

*Maxime Carpentier* <[maxime.carpentier@ebmwebsourcing.com](mailto:maxime.carpentier@ebmwebsourcing.com)>

*Adrien LOUIS* <[adrien.louis@ebmwebsourcing.com](mailto:adrien.louis@ebmwebsourcing.com)>

- January 2009 -



(CC) EBM WebSourcing - This work is licensed under the Creative Commons Attribution-NonCommercial-ShareAlike License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>



# Table of Contents

Introduction .....	3
1. Pre-requisites .....	4
2. WebConsole usage .....	5
3. Installation .....	6
3.1. Installing the application .....	6
3.2. Configuring the database server. ....	6
3.3. Configuring the artifact installation mode .....	6
3.4. Accessing the application .....	7
4. Data Collector Configuration .....	8
4.1. Starting the DataCollector .....	8
4.2. Configuring the DataCollector .....	8
4.3. Accessing the WebConsole .....	9
5. PEtALS Administration .....	10
5.1. Start / stop Components, service assemblies and shared libraries .....	10
5.2. Start / stop Monitoring .....	10
6. Monitoring .....	12
6.1. Various statistics .....	12
6.2. Entity details .....	13
6.3. System information and queues .....	13
6.4. Messages Exchanges .....	14
6.5. Endpoints .....	15
7. Service Test .....	17
8. Proxy Service .....	20
8.1. The Proxy Menu .....	20
8.2. Provide WSDL File .....	20
8.3. Select the service .....	21
8.4. Validate the informations .....	22
8.5. Select the server .....	23
8.6. Start the SA deployed .....	23
8.7. Go to your service .....	25

# Introduction

This document describes the PEtALS WebConsole functionalities.

# Chapter 1. Pre-requisites

Before installing the webconsole, make sure you have installed :

- A servlet container like [Tomcat](#)
- A PEtALS server up and running. **The webconsole will NOT work with any PEtALS version older than 2.1.**
- Web navigator [Firefox](#) 1.5 or later with Javascript activated (InternetExplorer is not fully supported).

To make the jUDDI part of the webconsole work, you must have jUDDI installed and running. If you haven't installed jUDDI yet, please follow the instructions [here](#)

## Chapter 2. WebConsole usage

The webconsole allows the management, monitoring and test of PEtALS servers.

- **Administration view** : This view manage Components, Service Assemblies and Shared Libraries life cycles.
- **Monitoring view** : Statistics on servers, components, services and messages.
- **Test view** : This view allows to manually send messages to services.

# Chapter 3. Installation

## 3.1. Installing the application

The webconsole is packaged as a war, that you have to deploy in a servlet container.

For example, Tomcat users just have to copy the war to the "webapps" directory and wait for Tomcat to deploy the application.

**Note :** If you deploy your console on a machine, and the petals node you try to connect is on another machine, you must be sure that the `conf/topology.xml` configuration file of your petals server contains the IP address of the server instead of "localhost". If it is not the case, correct it to allow the web console to connect to this server.

**Note :** The PEtALS Quickstart distribution *doesn't need any installation step*, the webconsole is directly embedded in PEtALS and can be accessed via <http://localhost:7878>

## 3.2. Configuring the database server.

The WebConsole stores information in an embedded HSQL database server pre-configured. You can also configure the WebConsole to use any external database.

- Copy the JDBC driver library for your database in the `lib` directory of your servlet container. For example, Tomcat users have to copy the jar file in the `$catalina_base/lib` directory.
- Edit the `WEB-INF/classes/webconsole.cfg.xml` configuration file and change the `database` section.

Example of configuration for mysql server (the database petalsDatabase must exist):

```
<database>
  <database-mode>standalone</database-mode>
  <database-url>jdbc:mysql://localhost:3306/petalsDatabase</database-url>
  <database-user>user</database-user>
  <database-password>password</database-password>
  <database-driver>org.gjt.mm.mysql.Driver</database-driver>
  <hibernate-dialect>org.hibernate.dialect.MySQLDialect</hibernate-dialect>
</database>
```

The property "database-mode" can be "embedded" to use the HSQL database server, or "standalone" to use an external database server.

The property "hibernate-dialect" is not mandatory.

Once the properties are correctly set, restart your servlet container.

## 3.3. Configuring the artifact installation mode

The Administration view of the WebConsole allows the user to install Components, SharedLibraries and ServiceAssemblies on any connected PEtALS server.

There is 2 ways to install an artifact :

- The user uploads from its web browser an artifact on the WebConsole. Then the PEtALS server connects itself to the WebConsole http server to download this artifact.
- The WebConsole is configured to show the user an artifacts FTP repository. When the user wants to install an artifact, the PEtALS server connects itself to the ftp server to download this artifact.

The default artifact installation mode is http upload. If the WebConsole runs on a multi-IP machine, you have to configure the IP address that PEtALS server will use to download the artifacts.

If you do not configure it, the IP address used is the one that the user set on its http request in its web browser.

If the user use "localhost" in the http request (because the web browser and the WebConsole are on the same machine), the PEtALS server will use the first IP address defined in the machine System Information.

To define a specific IP address, edit the `WEB-INF/classes/webconsole.cfg.xml` configuration file and change the `artifact-installation` section.

```
<artifact-installation>
  <download-uri>http://[host]:[port]/petals-webconsole</download-uri>
</artifact-installation>
```



### Caution

The FTP artifact repository mode is not available in this version

## 3.4. Accessing the application

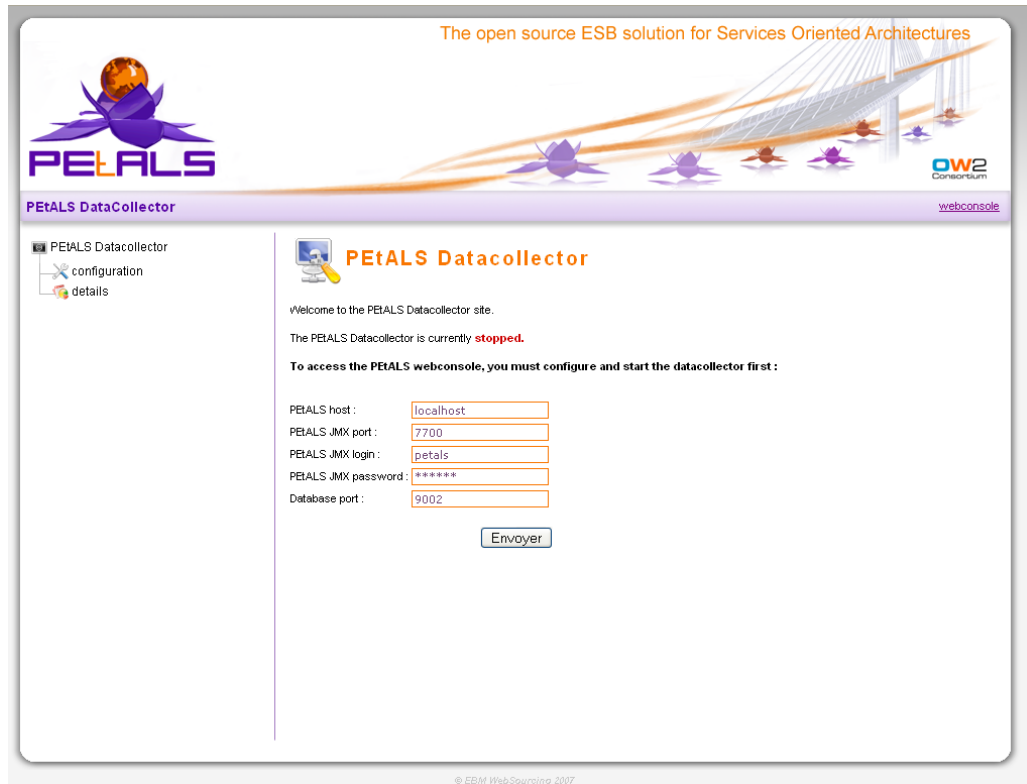
Once deployed, the webconsole can be accessed in your web browser through the address [http://\[host:port\]/petals-webconsole/](http://[host:port]/petals-webconsole/)

# Chapter 4. Data Collector Configuration

## 4.1. Starting the DataCollector

The PEtALS WebConsole uses a DataCollector to retrieve monitoring information from all the PEtALS server connected.

The first time you access the WebConsole, you need to configure and start the datacollector. The DataCollector is configured by referencing information of a JMX PEtALS server. Then the DataCollector automatically discovers the other PEtALS servers that are connected to this one.

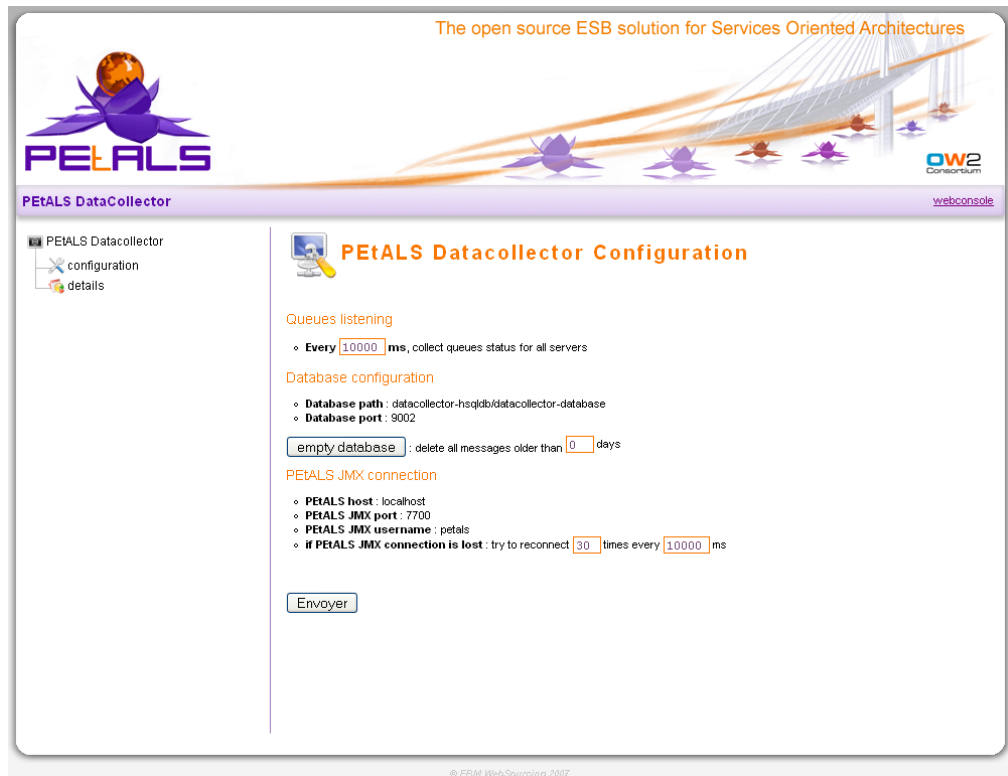


## 4.2. Configuring the DataCollector

Once started, the datacollector can be configured and administrated through the left menu : click on the left menu to access the configuration panel or the details panel

- **Queues listening** : each X seconds, the datacollector check all queues status and save them to its database if there's a non null value
- **Database configuration** : if there's too many messages saved in the collector database, you can clear all or a part of them thanks the "empty database" button
- **PEtALS JMX connection** : if the datacollector lost its connection to PEtALS, you can make it try to reconnect X time every X ms. Default is 30 times every 10 seconds





### 4.3. Accessing the WebConsole

Once the datacollector started, you can access the webconsole : click on the right link "webconsole" in the menubar to go to the console login page.

For now there's only 3 possible accounts:

- **petals/petals** : this user can access all pages without any restriction
- **admin/admin** : this user can only access administration and jUDDI pages
- **monitoring/monitoring** : this user can only access monitoring pages

# Chapter 5. PEtALS Administration


## 5.1. Start / stop Components, service assemblies and shared libraries

The first thing you may want to do is to install a component, a service assembly or a service unit : the administration part of the webconsole allows you to do so in a few clicks. Just click on the selected action in the left menu to open the right panel.




## 5.2. Start / stop Monitoring

In the left menu, clicking on the domain, the subdomain or the server opens a panel with details about the selected element. On the server panel, you can choose to activate or deactivate monitoring. If you want the messages in PEtALS to be monitored in real time, you must activate the monitoring.



The open source ESB solution for Services Oriented Architectures



[PEtALS Admin](#) | [JUDDI Admin](#) | [Monitoring](#)
logged in as **admin** (admin) | [log out](#) | [datacollector](#)

- PEtALS (standalone)
  - subdomain1
    - server 0
      - admin components
      - admin service assemblies
      - admin shared libraries

## PEtALS Administration - Server

server 0

**Components :**

- petals-sample-clock (Stopped)
- petals-sample-client (Started)

**Monitor the server**

- Monitoring status : **true** - Message content shown : **true**
- Start monitoring server 0 **with** messages content
- Start monitoring server 0 **without** messages content
- Stop monitoring server 0

**Server properties**

server name	0
server version	Petals JBI Container - version: 2.1
server host	localhost
JMX RMI port	3000
JMX JNDI port	7700
current state	STARTED
monitoring status	true

© EBM WebSourcing 2007

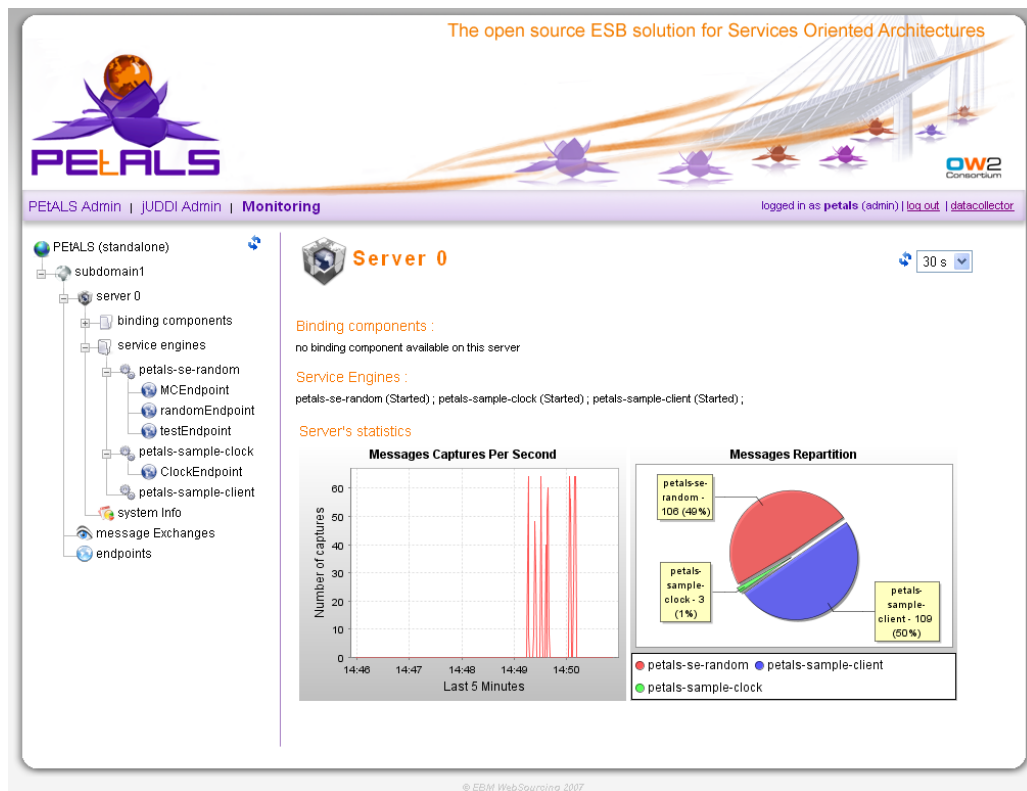
# Chapter 6. Monitoring

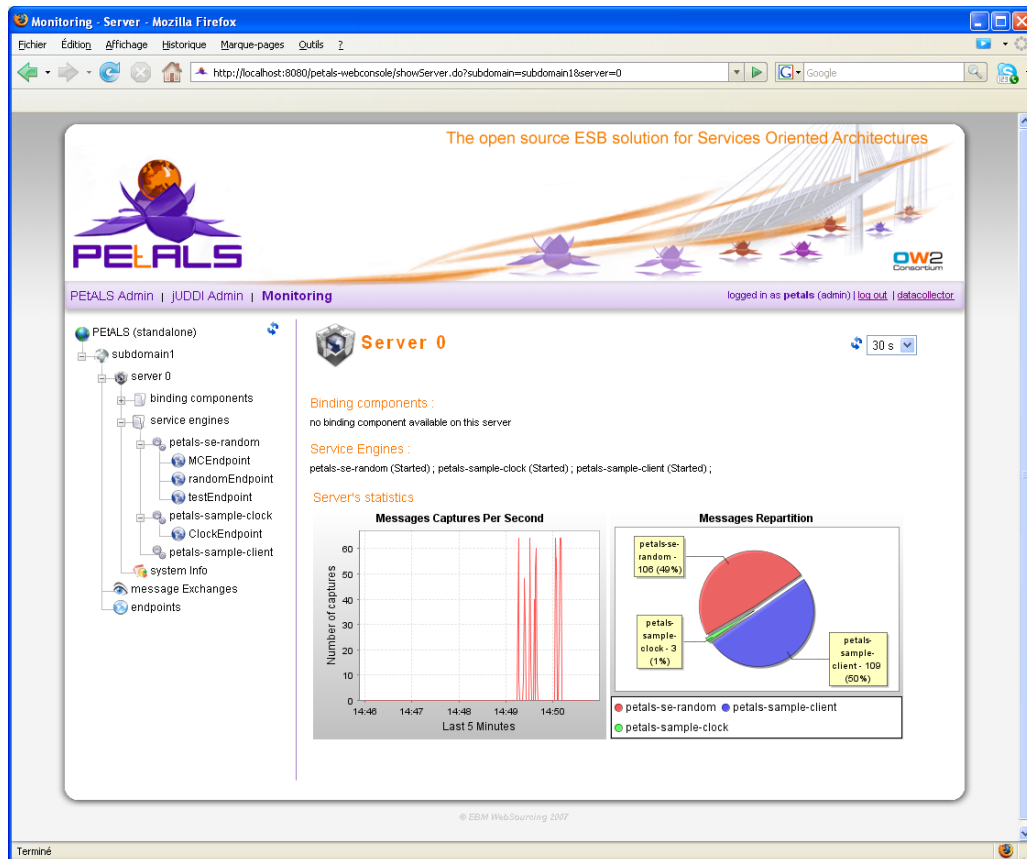
## 6.1. Various statistics

The monitoring part offers various statistics on every subdomain, servers, components, endpoints...just click on an item in the left menu to display charts about it.

usually, there's two charts displayed

- **Messages Captures Per Second** : Each time a message go through a component, a capture of this message is saved. For example, each **InOnly** message exchange will have **6** captures (in, out, done counted for each component)
- **Message Repartition** : Message Exchanges repartition, depending on the selected element. for example, a domain will have the message repartition chart by server.





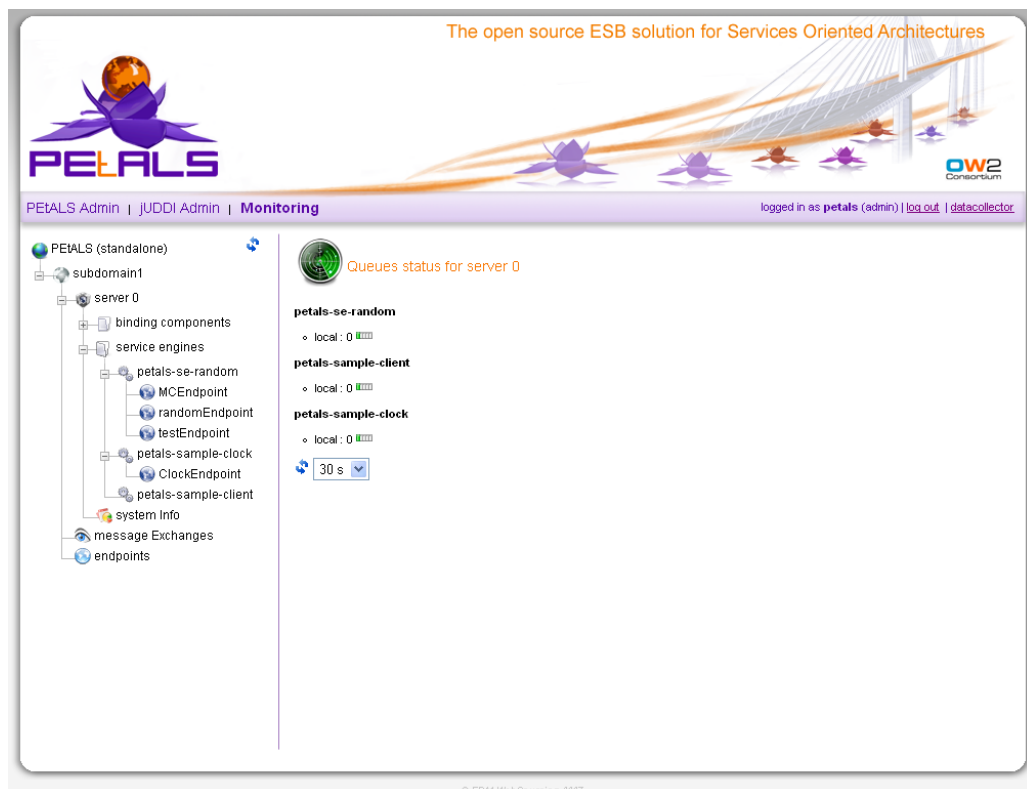
## 6.2. Entity details

Appart from message traffic, monitoring pages offers several information about the selected entity

- component : name, description, state, type, objectName, *list of endpoints* and *queues status*
- endpoint : interface, service, endpoint, component, server and *associated wsdl*

## 6.3. System information and queues

This section provides several about the system PEtALS is running on and about the different PEtALS queues.



## 6.4. Messages Exchanges

If you want to see what's transiting in PETALS, you're in the good section.

The messages Exchanges part of the webconsole allows you to :

- list all monitored messages exchanges

- see detailed information for each message exchange

Just click on the message Exchange item in the left menu and browse the message list. You can then select a message and get details about this message like the operation, the pattern, the content, ...

The open source ESB solution for Services Oriented Architectures

**PEtALS**

PETALS Admin | JUDDI Admin | **Monitoring** logged in as **petals** (admin) | [log out](#) | [datacollector](#)

PEtALS (standalone)

- subdomain1
  - server 0
    - binding components
    - service engines
      - petals-se-random
      - MCEndpoint
      - randomEndpoint
      - testEndpoint
      - petals-sample-clock
      - ClockEndpoint
      - petals-sample-client
    - system Info
    - message Exchanges
    - endpoints

**Message Exchanges list**

id	involves	status	start time	mep
petals:uid:B67611A5B5FCD97D3619681106230540106	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:11.015	InOnly
petals:uid:B67611A5B5FCD97D3619681042636983105	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.953	InOnly
petals:uid:B67611A5B5FCD97D3619681038578368104	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.953	InOnly
petals:uid:B67611A5B5FCD97D3619680746382570103	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.656	InOnly
petals:uid:B67611A5B5FCD97D3619680618404853102	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.531	InOnly
petals:uid:B67611A5B5FCD97D3619680482446512101	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.390	InOnly
petals:uid:B67611A5B5FCD97D3619680354447841100	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.265	InOnly
petals:uid:B67611A5B5FCD97D361968022646481699	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.140	InOnly
petals:uid:B67611A5B5FCD97D361968012248037198	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:10.031	InOnly
petals:uid:B67611A5B5FCD97D361967997815811797	petals-se-random petals-sample-client	done	15 févr. 2008 14:50:09.890	InOnly

0 1 2 3 4 5 6 7 8 9 10

© EBM WebSourcing 2007

The open source ESB solution for Services Oriented Architectures

**PEtALS**

PETALS Admin | JUDDI Admin | **Monitoring** logged in as **petals** (admin) | [log out](#) | [datacollector](#)

PEtALS (standalone)

- subdomain1
  - server 0
    - binding components
    - service engines
      - petals-se-random
      - MCEndpoint
      - randomEndpoint
      - testEndpoint
      - petals-sample-clock
      - ClockEndpoint
      - petals-sample-client
    - system Info
    - message Exchanges
    - endpoints

**Message Exchange details**

id	petals:uid:63E78A6D7A48095102240989462141133
start time	14 févr. 2008 16:08:18.437
duration	31 ms
status	done
mep	InOut
server / service / endpoint / operation	0 / (http://petals.ow2.org/ClockService / ClockEndpoint / time


duration: 0 ms

message type	in
start time	14 févr. 2008 16:08:18.437
duration	0 ms
content	<text>hi</text>


© EBM WebSourcing 2007

## 6.5. Endpoints


The endpoints section displays several informations about all endpoints for a subdomain




The open source ESB solution for Services Oriented Architectures







[PETALS Admin](#) | [JUDDI Admin](#) | **Monitoring**
logged in as **petals** (admin) | [log out](#) | [datacollector](#)

 PETALS (standalone)
 

- subdomain1
  - server 0
    - binding components
    - service engines
      - petals-se-random
        - MCEndpoint
        - randomEndpoint
        - testEndpoint
      - petals-sample-clock
        - ClockEndpoint
      - petals-sample-client
    - system Info
    - message Exchanges
    - endpoints

 **Endpoints**

interface	service	endpoint	details
(http://petals.ow2.org)Clock	(http://petals.ow2.org)ClockService	ClockEndpoint	
(http://random.com/1.0)testInterface	(http://random.com/1.0)randomService	MCEndpoint	
(http://random.com/1.0)randomInterface	(http://random.com/1.0)randomService	randomEndpoint	
(http://random.com/1.0)randomInterface	(http://random.com/1.0)testService	testEndpoint	

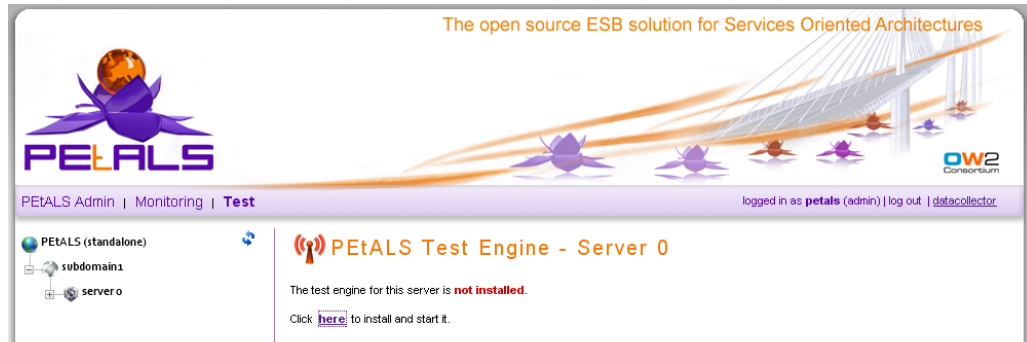
© EBM WebSourcing 2007



# Chapter 7. Service Test

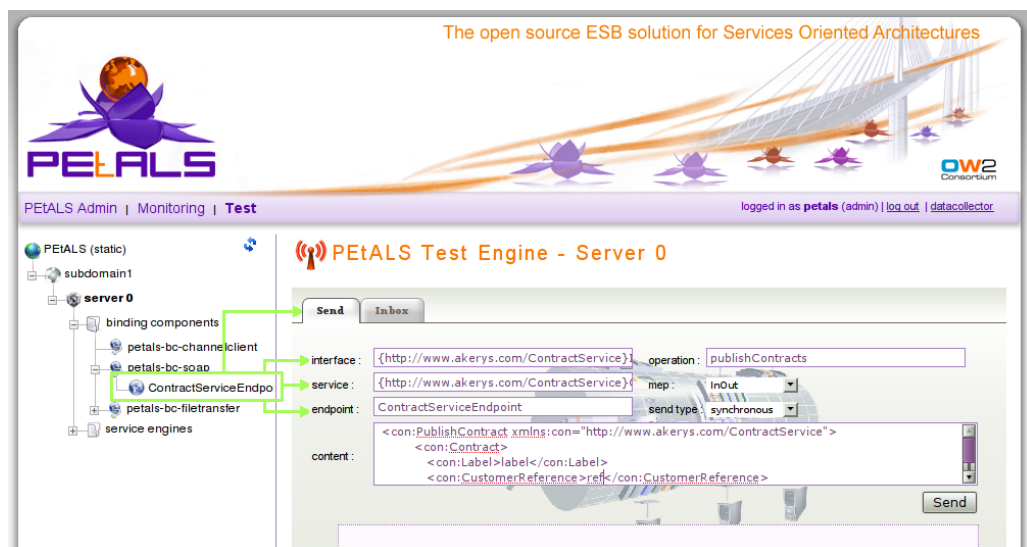
The Test view of the WebConsole allows the user to test services that are deployed on a PEtALS server.

Before the user can test a service, the WebConsole install a specific component on the PEtALS server to be able to send test message. If the component (petals-bc-channelclient) is not installed, the WebConsole proposes you to install it at the beginning of the Test tab.

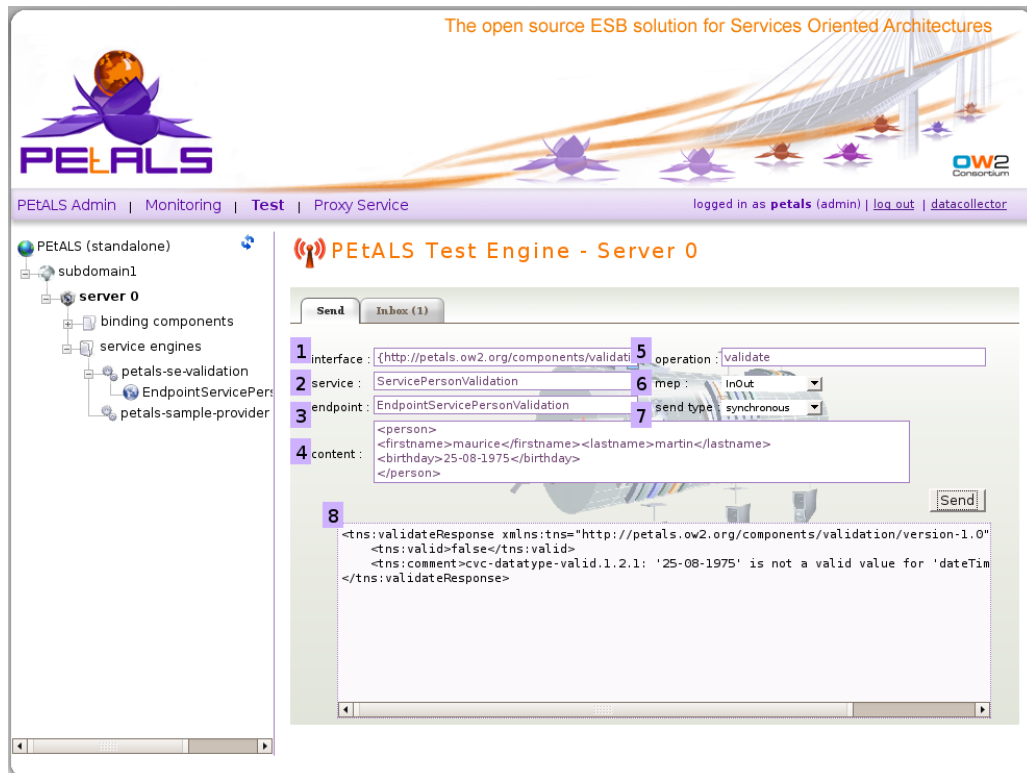


Once the test-component installed, you can select a service to test, and send an XML message.

Specific JBI information is also need to send a message (Exchange Pattern, ...)



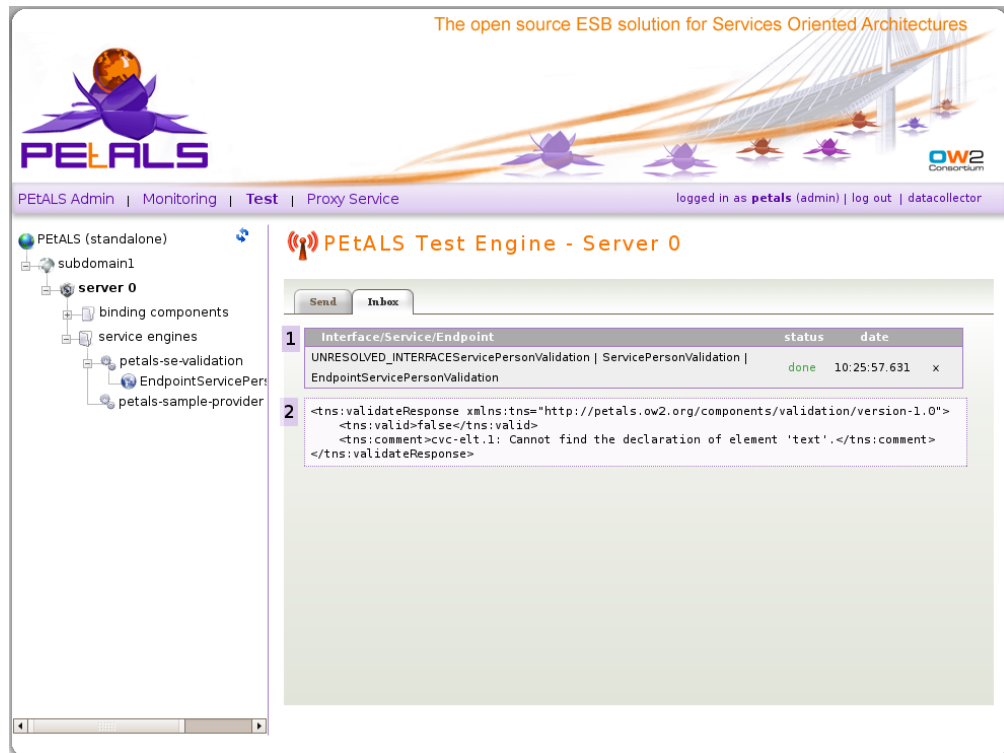
You can click on BC/SE endpoint to auto-complete some fields of in the "Send" tab, like Interface, Service and Endpoint.



This first tab allow you to test a service for Binding Component/ or Service Engine.

Description of fields :

1. This first field represent the interface of your service where several operation are define.
2. The second field of this tab reference the service which you need to perform the test.
3. Third field is the endpoint where your service is located (bounded).
4. This field represent the "content", the message that you send to the selected service.
5. The operation field represent de operation you want to test, provided by the interface.
6. The mep is the "Message Exchange Pattern"
7. This field is the send type how the message is send, it can take two value (synchronous/asynchronous).
8. This area represent the retrieval response area, allow the display of the message response of your test.



This second tab allow you to check the history of the response messages sent for several tests.

Description of fields :

- The first box represent one response with the Interface/Service/Endpoint used for this test, the status of the response (done, error ...) and the date when the response was received.
- This other box is the content of the reponse, returned for the test.

# Chapter 8. Proxy Service

We assume that you are already on the PEtALS web console.

## 8.1. The Proxy Menu



In the first step click on the « Proxy Service » menu for begin the proxyfication.

## 8.2. Provide WSDL File

You arrive at this window:

The open source ESB solution for Services Oriented Architectures

**PETALS**

PETALS Admin | Monitoring | Test | **Proxy Service** | logged in as **petals (admin)** | [log out](#) | [datacollector](#)

You can create and deploy web service proxy by providing the WSDL, the address of the SU to provide and consume, and the name of your proxy (SA).

WSDL File:  [Parcourir...](#)

1

2

During the second step, you must provide a « wsdl file » (seek it with (1), in the new window tree just appear). After you find the « wsdl file » click on « submit » (2)

### 8.3. Select the service

The open source ESB solution for Services Oriented Architectures

**PETALS**

PETALS Admin | Monitoring | Test | **Proxy Service** | logged in as **petals (admin)** | [log out](#) | [datacollector](#)

Service: <http://webservices.amazon.com/Endpoint/AWSECommerceServicePort>

Proxy name (SA):

WSDL:

Timeout:

SU provide address:

SU consume address:

Endpoint type:

Soap: ☐

Rest: ☐

Select the server where you want to deploy the SA

PETALS.subdomain1.server0

In this new window you can find all services found in your wsdl file, and several parameters related to the proxy generation. Here select the service (in the list) which will be proxyfied.

## 8.4. Validate the informations

The open source ESB solution for Services Oriented Architectures

**PETALS**

PETALS Admin | Monitoring | Test | Proxy Service

logged in as **petals** (admin) | [log out](#) | [datacollector](#)

Service: {http://webservices.amazon.com/}AWSECommerceServicePort

Endpoint: [AWSECommerceServicePort](#)

Proxy name (SA):  1

WSDL:  2

Timeout:  3

SU provide address:  4

SU consume address:  5

Endpoint type:

Soap: ☒ 6

Rest: ☐

Submit

Select the server where you want to deploy the SA

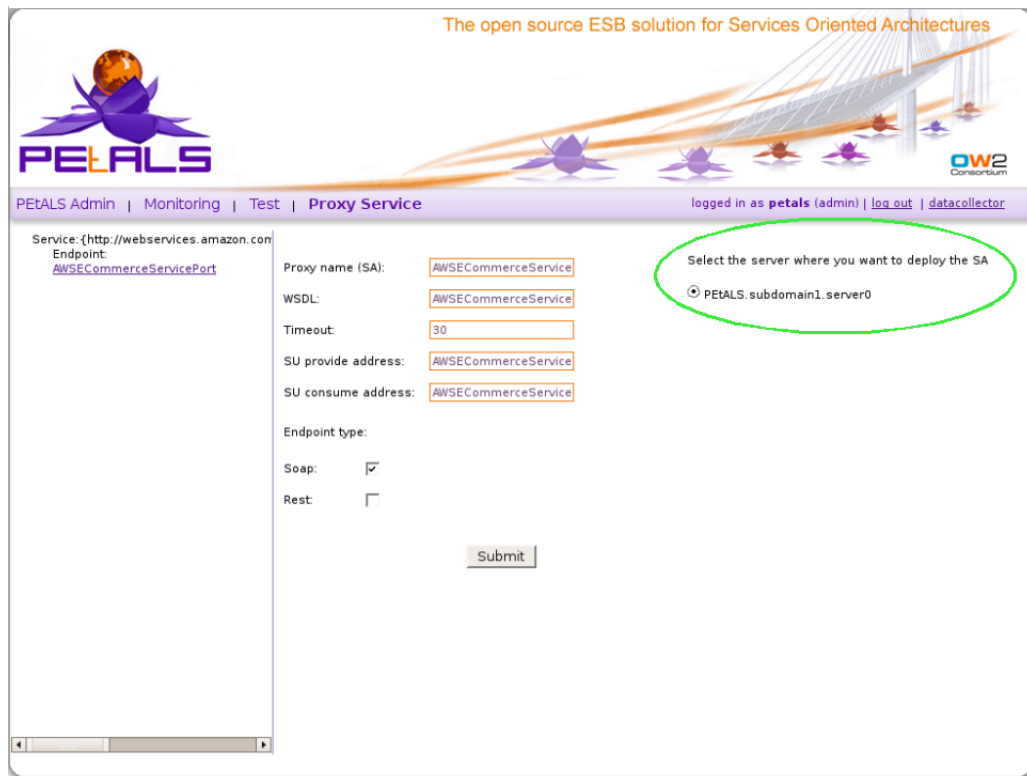
PETALS.subdomain1.server0

After this, all fields are filled with many informations for the proxy:

- 1 : The field Proxy name (SA) represents the name of service assembly of the proxyfied service.
- 2 : WSDL represents the wsdl you provided.
- 3 : Timeout represents the timeout delay when sending messages.
- 4 : SU provide address represents the service address where the service consumer will query.
- 5 : SU consume address represents the service address where the service provider will be accessible by the address of the web service.
- 6 : Soap et Rest are the differents types of the selected service in the list.

(You can modify the values of these differents fields, but we advise you to keep the default values).

## 8.5. Select the server



The open source ESB solution for Services Oriented Architectures

**PETALS**

PETALS Admin | Monitoring | Test | **Proxy Service** | logged in as **petals (admin)** | [log out](#) | [datacollector](#)

Service: {http://webservices.amazon.com/}AWSECommerceServicePort

Proxy name (SA):

WSDL:

Timeout:

SU provide address:

SU consume address:

Endpoint type:

Soap: ☒

Rest: ☐

Submit

Select the server where you want to deploy the SA

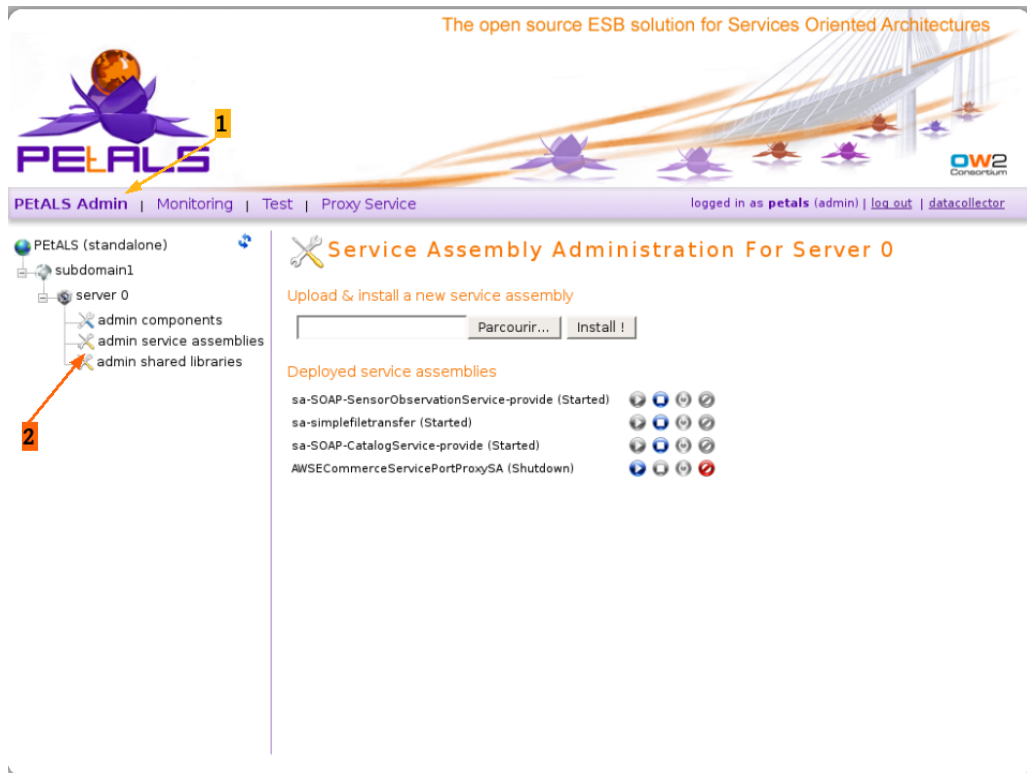
☒ PETALS.subdomain1.server0

Now you must select the server on which the web console will deploy the proxy.

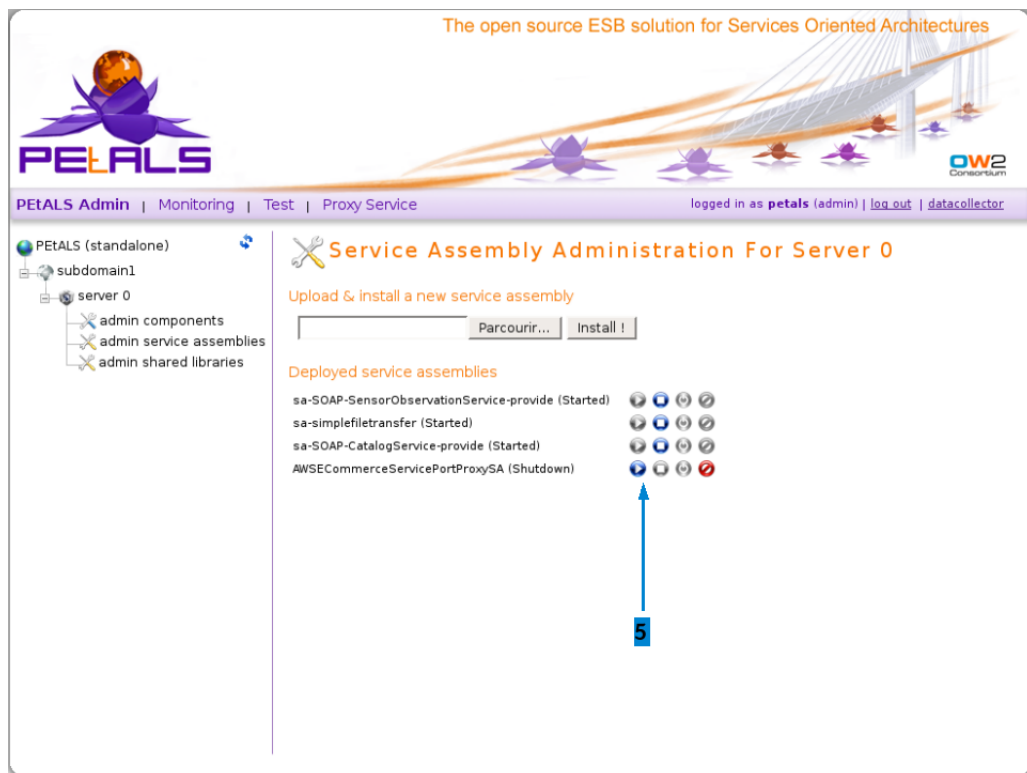
After, click on « Submit » to recover the zip file of SA. Then, the proxy is correctly deployed. (you can see normally a green message "Proxy added, and SA properly deployed")

## 8.6. Start the SA deployed

Now if you need to run the new SA generated, follow these



- 1 : First, click on the « PETALS Admin » menu to display administration part.
- 2 : Second, select « admin service assemblies », for display the SA installation frame.

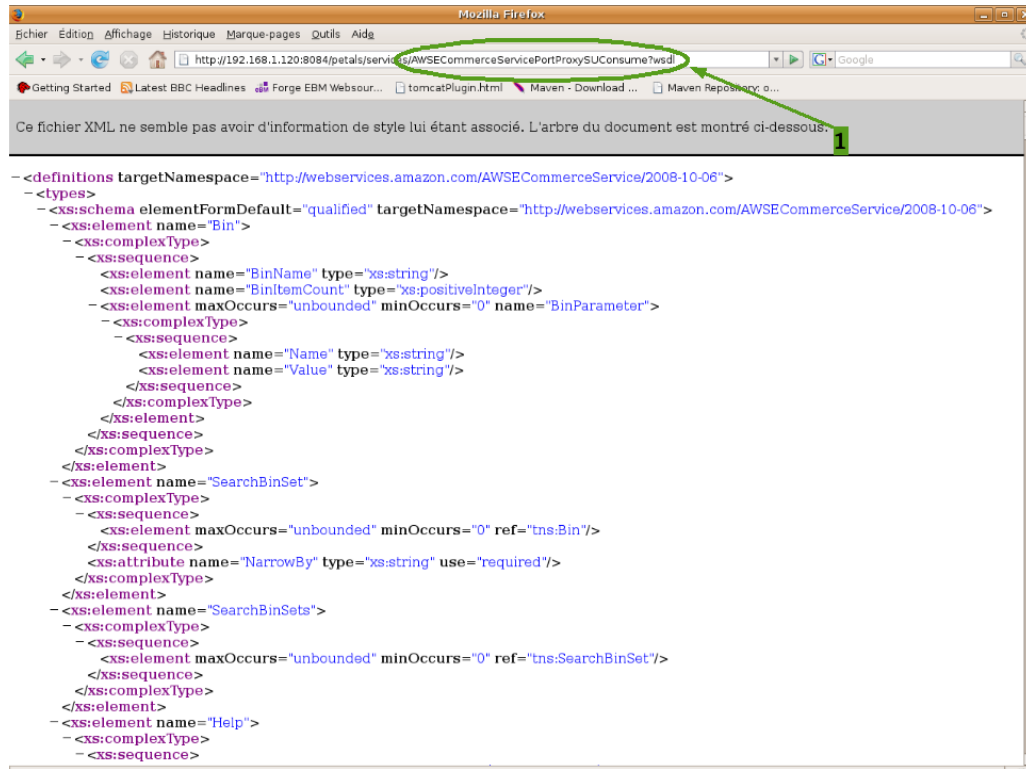


- 5 : After in the SA installation frame, click on « Start » to run the service.



## 8.7. Go to your service

Now you can access to the wsdl of your new proxyfied service to this address: `http://{host}:{port}/petals/services/{SUConsumeAddress}?wsdl`



- 1 : The « /SUConsumeAddress » is the name you provided in the form, when you create the SA zip file.