S-RAMP Guide

by Dr. Kurt T JBoss Stam (Red Hat) and Mr. Eric JBoss Wittmann (Red Hat)

Dedication

TBD

Preface

TBD

Chapter 1. Introduction to S-RAMP

1.1. The S-RAMP Specification

S-RAMP stands for SOA Repository Artifact Model and Protocol. *S-RAMP* [https://www.oasis-open.org/committees/s-ramp/charter.php] is a new specification worked on by the OASIS Technical Committee.

> The SOA Repository Artifact Model and Protocol (S-RAMP) TC defines a common data model for SOA repositories as well as an interaction protocol to facilitate the use of common tooling and sharing of data. The TC will define an ATOM binding which documents the syntax for interaction with a compliant repository for create, read, update, delete and query operations.
>
> — OASIS Charter *https://www.oasis-open.org/committees/s-ramp/charter.php*

The first version of the specification (1.0) should be finalized in the first half of 2013. Two of the developers on the project participated in the Technical Committee.

1.2. High Level Architecture

Chapter 2. Getting Started

2.1. Prerequisites

The S-RAMP application is written in Java. To get started make sure your system has the following:

- Java JDK 1.6 or newer
- Apache Ant 1.7 or newer to use the installer
- Maven 3.0.3 or newer to build and run the examples

2.2. Download, Installation and Configuration

The `s-ramp-<version>.zip` (or tar.gz) archive can be downloaded from the *http://www.jboss.org/overlord* website. Grab the latest, extract the archive and run:

```
ant install
```

The first thing this does is to *jBPM* [http://www.jboss.org/jbpm], *RESTEasy* [http://www.jboss.org/resteasy] and *ModeShape* [http://www.jboss.org/modeshape] from sourceforge.

Note

This download is slow and can take up to 15 minutes.

The downloads will be stored in the jbpm5 directory, so future installs will be faster. In this same directory it will create a `jbpm-installer` directory with jBPM and a sub-directory containing the application server `jboss-as-7.1.1.Final`. The application server has ModeShape and RESTEasy ed as modules. Next run:

```
ant configure
```

This will add some S-RAMP specific services to jBPM. It deploys the `s-ramp-server.war`, the `s-ramp-ui.war` and the `governance.war`. Finally, it points jBPM to obtain its workflow data from the S-RAMP repository, rather than BRMS/Drools-Guvnor. At this point we want to populate the S-RAMP repository with some initial data. For this the repository needs to be up and running. Go ahead and start the JBoss application server by running:

```
ant start
```

If your system support the `tail` command, you can run:

```
ant tail
```

This will monitor the startup process. It should be ready as soon as you see the following in the application server console output:

```
[exec] 12:00:13,905 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "s-ramp-ui.war"
[exec] 12:00:13,910 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "s-ramp-server.war"
[exec] 12:00:13,915 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "s-ramp-governance.war"
[exec] 12:00:13,918 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "jbpm-human-task-war.war"
[exec] 12:00:13,922 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "jbpm-gwt-console.war"
[exec] 12:00:13,925 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "jbpm-gwt-console-server.war"
[exec] 12:00:13,928 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "jbpm-form-builder.war"
[exec] 12:00:13,931 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "drools-guvnor.war"
[exec] 12:00:13,933 INFO  [org.jboss.as.server] (DeploymentScanner-threads -
 2) JBAS018559: Deployed "designer.war"
```

Tip

S-RAMP can be configured to use a remote jBPM server, so to cut down on bootup time you can deploy the S-RAMP applications on a dedicated application server.

Now you can upload the by running:

```
ant upload
```

This completes the installation process.

2.3. Check your Installation

To make sure your installation works you can fire up the *s-ramp-ui* [http://localhost:8080/s-ramp-ui]. You should see the GUI as shown in *Figure 2.1, “Welcome screen of the s-ramp-ui.”*:

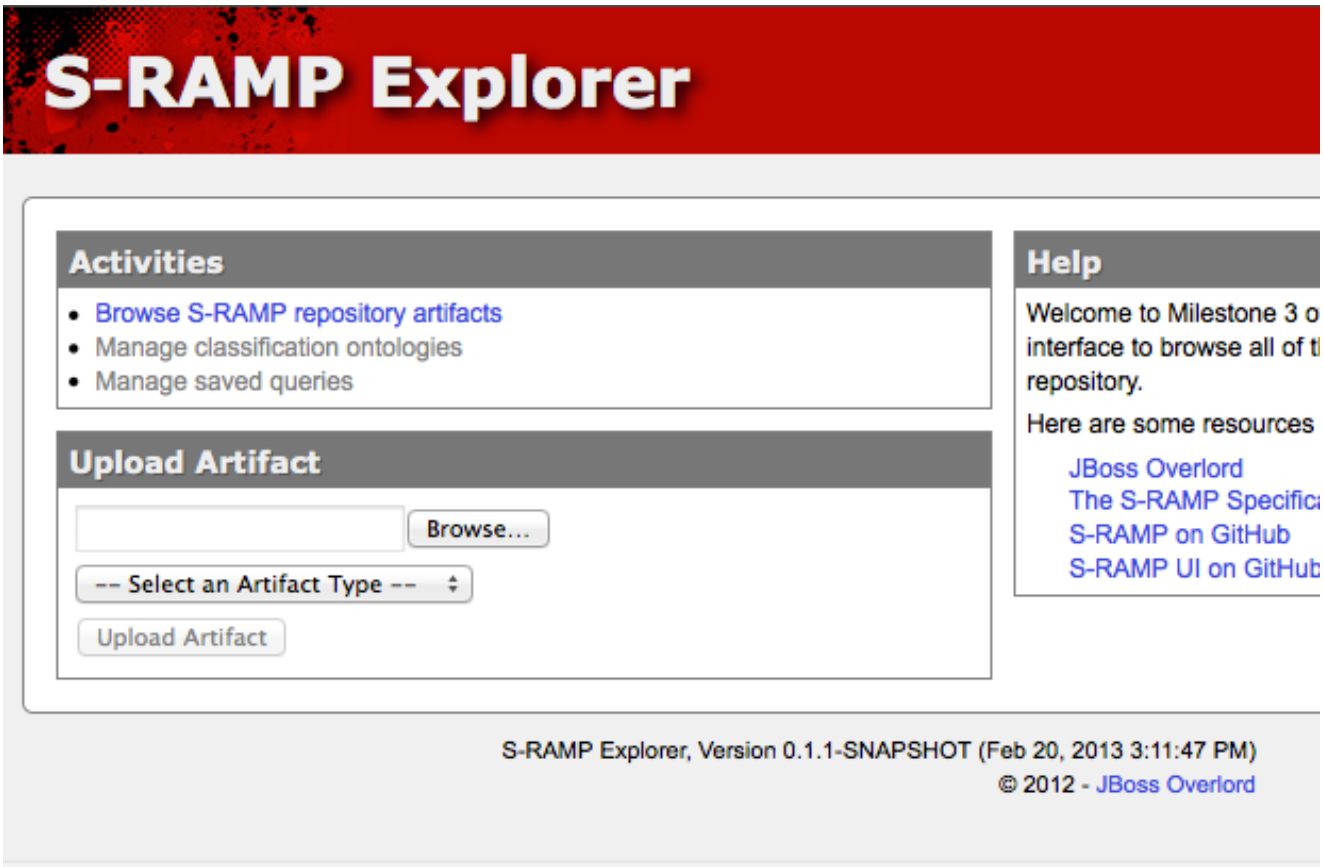

Figure 2.1. Welcome screen of the s-ramp-ui.

You can click on `Browse Artifacts` and see a list of files related to the S-RAMP default workflows. Alternatively you can fire up the `s-ramp shell` in the `bin` directory of the distribution:

```
./s-ramp.sh
************************************************************************
          _____       ______  ___ ___  ________
         /  ___|      | ___ \/ _ \|  \/  | ___ \
         \ `--. ______| |_/ / /_\ \ .  . | |_/ /
          `--. \______|    /|  _  | |\/| |  __/
         /\__/ /      | |\ \| | | | |  | | |
         \____/       \_| \_\_| |_|_|  |_|_|

  JBoss S-RAMP Kurt Stam and Eric Wittmann, Licensed under the
  Apache License, V2.0, Copyright 2012
************************************************************************
s-ramp>
```

To connect the shell to the server type `connect` and hit the tab key. It should auto-complete to say `s-ramp:connect http://localhost:8080/s-ramp-server` and when hitting the return key the cursor should go from red to green. To browse the artifacts in the repository run the following query:

```
s-ramp> s-ramp:query /s-ramp
Querying the S-RAMP repository:
        /s-ramp
Atom Feed (9 entries)
  Idx                    Type Name
  ---                    ---- ----
    1           ImageDocument user-properties.png
    2                Document overlord.demo.CheckDeployment-taskform.flt
    3         BrmsPkgDocument SRAMPPackage.pkg
    4           ImageDocument overlord.demo.SimpleReleaseProcess-image.png
    5           ImageDocument run-build-install.png
    6                Document overlord.demo.SimpleReleaseProcess-
taskform.flt
    7           ImageDocument audio-input-microphone-3.png
    8            BpmnDocument overlord.demo.SimpleReleaseProcess.bpmn
    9            TextDocument HttpClientWorkDefinitions.wid
```

In later chapters will go into more detail, but if this all worked you can be sure that your installation is in good working order.

Chapter 3. S-RAMP Data Structures

Chapter 4. S-RAMP REST API

Chapter 5. S-RAMP Samples

Chapter 6. SOA Governance

6.1. Introduction

To support SOA Governance management a governance application monitors a S-RAMP compliant repository for certain events. These events can then be used to kickoff governance workflows.

6.2. Concepts

TBD

6.3. Workflow

Organizations define policies and processes to managing artifacts. These policies and processes will be different for each organization. We therefore enlisted the help of a BPMN workflow engine. This should give users the needed flexibility to implement their specific workflow. We ship a number of best practices workflow that users can copy and or modify to their needs.

6.3.1. Repository Events using S-RAMP Queries

Events in the repository can be used to kickoff workflows. The governance application monitors an S-RAMP compliant repository using S-RAMP queries.

Appendix A. Example Appendix

One or more optional appendixes go here at section level 1.

A.1. Appendix Sub-section

Sub-section body.

Example Bibliography

The bibliography list is a style of AsciiDoc bulleted list.

Books

[walsh-muellner] Norman Walsh & Leonard Muellner. *DocBook - The Definitive Guide*. O'Reilly & Associates. 1999. ISBN 1-56592-580-7.

Articles

[abc2003] Gall Anonim. *An article*, Whatever. 2003.

Example Glossary

Glossaries are optional. Glossaries entries are an example of a style of AsciiDoc labeled lists.

A glossary term
: The corresponding (indented) definition.

A second glossary term
: The corresponding (indented) definition.

Example Colophon

Text at the end of a book describing facts about its production.

Index