

Installation Guide

DMDII-CHAMP

Exported on 01/11/2018

1 Table of Contents

1	Table of Contents.....	2
2	Runtime Environment	3
3	Database.....	4
4	Application Container.....	5
5	Build Environment	6
6	OntoView	7
6.1	Build OntoView from source.....	7
6.2	Install OntoView	8
6.3	Configure database.....	8
6.4	Start runtime environment	8
6.5	Verify Operation	8
7	Process Workflow.....	9
7.1	Build Process Workflow from source	9
7.2	Install Process Workflow	10
7.3	Configure database.....	10
7.4	Start runtime environment	10
7.5	Verify Operation	10

2 Runtime Environment

All instructions below require the [Java 8 JDK](#)¹ appropriate for the installation machine. The installation directory will further be referred to as **JAVA_HOME**. Add this directory to the PATH environment variable, and set the JAVA_HOME environment variable to this directory.

¹ <http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

3 Database

Both OntoView and Process Workflow require a triple store capable of consuming [SPARQL 1.1](#)² queries and update requests via HTTP, and returning results in [SPARQL 1.1 Query Results JSON](#)³ format. Apache Jena Fuseki is one such triple store. To install, download Fuseki 3.4.0 ([ZIP](#)⁴, [tarball](#)⁵) and decompress the archive in a location appropriate for the installation machine. The installation directory will further be referred to as **FUSEKI_HOME**. Set the FUSEKI_HOME environment variable to this directory.

To run the Fuseki server, execute the following:

```
FUSEKI_HOME/fuseki-server.bat
```

if on a Windows machine, or

```
FUSEKI_HOME/fuseki-server
```

otherwise.

² <https://www.w3.org/TR/sparql11-query/>

³ <https://www.w3.org/TR/sparql11-results-json/>

⁴ <http://archive.apache.org/dist/jena/binaries/apache-jena-3.4.0.zip>

⁵ <http://archive.apache.org/dist/jena/binaries/apache-jena-3.4.0.tar.gz>

4 Application Container

Both OntoView and Process Workflow are packaged as WAR files, which are run within an enterprise application container. JBoss Wildfly is one such open-source container. To install, download Wildfly 11.0.0.Final ([ZIP](#)⁶, [tarball](#)⁷) and decompress the archive in a location appropriate for the installation machine. The installation directory will further be referred to as **JBOSS_HOME**. Set the JBOSS_HOME environment variable to this directory.

The following instructions assume that the installation machine exposes the proper ports for Wildfly: the default port is 8080, and can be configured to something else. The root endpoint will further be referred to as **WEB_APP_ROOT**.

⁶ <http://download.jboss.org/wildfly/11.0.0.Final/wildfly-11.0.0.Final.zip>

⁷ <http://download.jboss.org/wildfly/11.0.0.Final/wildfly-11.0.0.Final.tar.gz>

5 Build Environment

Both OntoView and Process Workflow require the following build tools:

1. [Apache Maven](https://maven.apache.org/)⁸ (version 3 or above)
2. [Node.js/NPM](https://nodejs.org/en/)⁹

⁸ <https://maven.apache.org/>

⁹ <https://nodejs.org/en/>

6 OntoView

6.1 Build OntoView from source

The source directory will further be referred to as **ONTOVIEW_SOURCE**.

First, build the web application:

```
cd ONTOVIEW_SOURCE/src/main/javascript
npm install
bower install
ember build
```

Second, build the WAR:

```
cd ONTOVIEW_SOURCE
mvn install -P{fs-profile},{query-profile}
```

where {fs-profile} and {query-profile} are the selected profiles from below:

1. One file system profile: fs-mac, fs-unix, fs-windows
2. One query profile:
 - a. query-local: uses the Fuseki default query host and port with the OntoView default ontology graph IRI
 - b. query-remote: uses alternate query endpoints and graph IRI, configurable in the pom.xml file
 - i. This was removed in favor of requiring the properties normally set by a query profile instead. Otherwise, there would be hard-coded values according to the CUBRC instance, or they would be empty and have to be required anyways.

Choose the appropriate profiles for the installation machine. This will result in a WAR output located in the **ONTOVIEW_SOURCE/target** directory.

The project pom.xml file specifies the following properties:

- ontoview.query.location: the SPARQL 1.1 query endpoint. Set by query profiles.
- ontoview.query.ontology.graph: the graph which holds all of the ontologies. Set by query profiles.
- ontoview.user: the username which is running the application. Set by user profiles.
- user.properties.location: the file location on disk where user-configured properties will be stored. Set by file system profiles.

These can be overridden on the command line:

```
cd ONTOVIEW_SOURCE
mvn -D{property.name}={property.value} clean install
```

6.2 Install OntoView

Copy the WAR file from **ONTOVIEW_SOURCE**/target into **JBOSS_HOME**/standalone/deployments.

6.3 Configure database

Start the Fuseki server if it is not already running, and navigate to the web interface, then click on "manage datasets". Add a new persistent dataset named "ontoview", then click "upload data" next to the new dataset. Upload the CCO files into a graph with the ontology graph IRI configured above.

6.4 Start runtime environment

If the runtime environment is not currently running, execute the following:

```
JBOSS_HOME/bin/standalone.bat -b 0.0.0.0
```

if on a Windows machine, or

```
JBOSS_HOME/bin/standalone.sh -b 0.0.0.0
```

otherwise.

6.5 Verify Operation

In a web browser, navigate to **WEB_APP_ROOT**/ontoview to view the landing page. To change any settings, navigate to **WEB_APP_ROOT**/ontoview/settings.

7 Process Workflow

7.1 Build Process Workflow from source

The source directory will further be referred to as **PW_SOURCE**.

First, build the web application:

```
cd PW_SOURCE/src/main/javascript
npm install
bower install
ember build
```

Second, build the WAR:

```
cd ONTOVIEW_SOURCE
mvn clean install -P{fs-profile},{query-profile},{worker-profile},{user-profile}
```

where {fs-profile}, {query-profile}, and {worker-profile} are selected build profiles from below:

1. One file system profile: fs-mac, fs-unix, fs-windows
2. One query profile:
 - a. query-local: uses the Fuseki default query host and port with the Process Workflow default ontology graph IRI
 - b. query-remote: uses alternate query endpoints and graph IRI, configurable in the pom.xml file
 - i. This was removed in favor of requiring the properties normally set by a query profile instead. Otherwise, there would be hard-coded values according to the CUBRC instance, or they would be empty and have to be required anyways.
3. One data worker profile:
 - a. worker-rdf: uses the configured triple store to persist records in RDF
 - b. worker-fs: uses the local file system to persist JSON-API records
4. One user profile:
 - a. user-default: default user is "jboss".
 - b. user-current: uses the user name of whichever user is building the application

Choose the appropriate profiles for the installation machine. This will result in a WAR output located in the **PW_SOURCE/target** directory.

The project pom.xml file specifies the following properties:

- process-workflow.user: the username which is running the application. Set by user profiles.
- process-workflow.query.location: the SPARQL 1.1 query endpoint. Set by query profiles.
- process-workflow.query.ontology.graph: the graph which holds all of the ontologies. Set by query profiles.
- process-workflow.update.location: the SPARQL 1.1 update request endpoint. Set by query profiles.

- `process-workflow.data.controller.worker`: the type of data worker: either "filePersistenceWorker" or "dataControllerWorker". Set by data worker profiles.
- `process-workflow.data.location`: the location for data saved through the file system persistence mechanism, if enabled. Set by file system profiles.
- `process-workflow.default.query.data.graph`: the graph which holds the current data records.
- `process-workflow.default.query.deleted.record.graph`: the graph which holds the deleted data records.
- `process-workflow.default.query.delta.graph`: the graph which holds timestamped state changes from a previous version of a record to the next version.

These can be overridden on the command line:

```
cd ONTOVIEW_SOURCE
mvn -D{property.name}={property.value} clean install
```

7.2 Install Process Workflow

Copy the WAR file from **PW_SOURCE**/target into **JBOSS_HOME**/standalone/deployments.

7.3 Configure database

Start the Fuseki server if it is not already running, and navigate to the web interface, then click on "manage datasets". Add a new persistent dataset named "process-workflow", then click "upload data" next to the new dataset. Upload the CCO files into a graph with the ontology graph IRI configured above.

7.4 Start runtime environment

If the runtime environment is not currently running, execute the following:

```
JBOSS_HOME/bin/standalone.bat -b 0.0.0.0
```

if on a Windows machine, or

```
JBOSS_HOME/bin/standalone.sh -b 0.0.0.0
```

otherwise.

7.5 Verify Operation

In a web browser, navigate to **WEB_APP_ROOT**/process-workflow to view the landing page. To change any settings, navigate to **WEB_APP_ROOT**/process-workflow/settings.