

Last edited by  [Ondřej Vašíček](#) 16 hours ago

1. Configuration

Configuration changes only take effect after restarting the adapter. Output filter class changes currently only take effect after rebuilding the adapter.

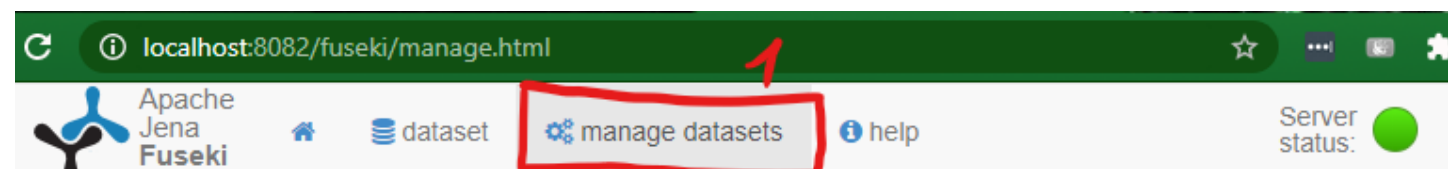
1) Triplestore

Configure host and port in conf/TriplestoreConf.ini

```
conf_example > TriplestoreConf.ini
1  ## THIS IS A EXAMPLE CONFIGURATION FILE
2  ##
3
4  --module=http
5      # port to run the triplestore on
6      jetty.http.port=8082
7      # host to run the triplestore on
8      jetty.http.host=localhost
9
10
11 --module=home-base-warning
12 --module=ext
13 --module=server
14 --module=jsp
15 --module=resources
16 --module=deploy
17 --module=jstl
18 --module=websocket
19
```

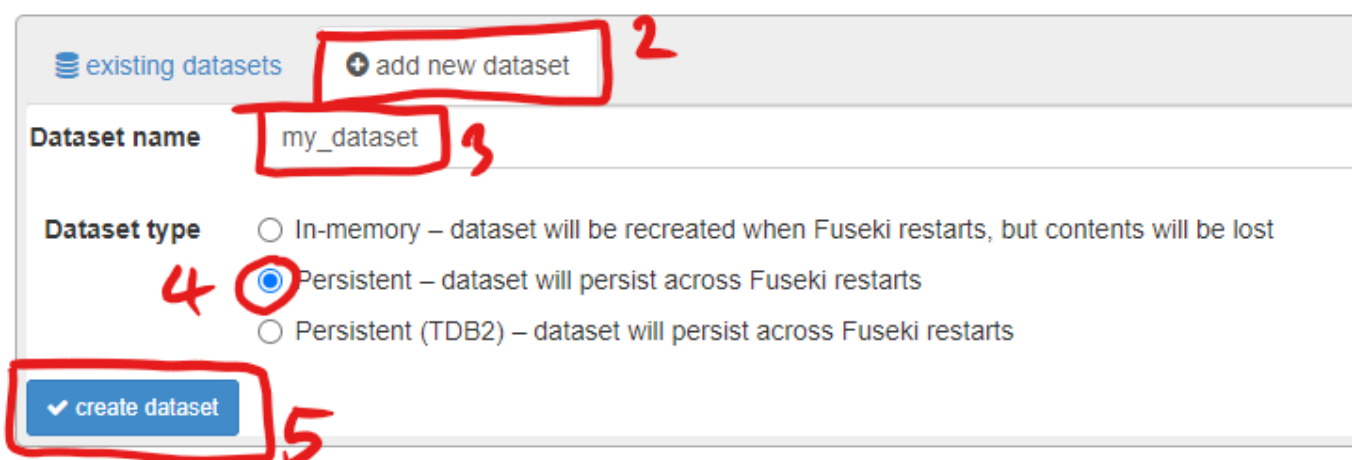
1.1) Creating a persistent dataset

First create dataset using the Fuseki Web UI. Set the dataset name at point 3.



Manage datasets

Perform management actions on existing datasets, including backup, or add a new dataset.



Look up the endpoints of your new dataset. Use these endpoints to configure the Analysis and the Compilation adapters.

The screenshot shows the Apache Jena Fuseki web interface at `localhost:8082/fuseki/dataset.html`. The interface includes a navigation bar with the Apache Jena Fuseki logo, a 'dataset' button (highlighted with a red box and labeled '1'), a 'manage datasets' link, and a 'help' link. The 'Server status' is shown as green.

Below the navigation bar, the 'Dataset:' dropdown menu is set to '/my_dataset' (highlighted with a red box and labeled '2').

The main content area is titled 'Available services' and lists the following endpoints:

- File Upload: `/fuseki/my_dataset/upload`
- Graph Store Protocol: `/fuseki/my_dataset/data`
- Graph Store Protocol (Read): `/fuseki/my_dataset/get`
- HTTP Quads: `/fuseki/my_dataset/`
- SPARQL Query: `/fuseki/my_dataset/quer` (highlighted with a red box and labeled '3')
- SPARQL Query: `/fuseki/my_dataset/sparql`
- SPARQL Update: `/fuseki/my_dataset/update` (highlighted with a red box)

Below the services list is a 'Statistics' section with a table showing performance metrics for the dataset '/my_dataset'.

Name	Overall	Overall good	Overall bad	Graph Store Protocol	HTTP Quads	File Upload	SPARQL Query	SPARQL Query	SPARQL Update
/my_dataset	0	0	0	0	0	0	0	0	0

Below the statistics is a 'Dataset size' section with a note: 'Note this may be slow and impose a significant load on large datasets:'. A button labeled 'count triples in all graphs' is available.

The 'Ongoing operations' section is currently empty, with a note: 'TBD. Will list any long-lasting operations that are ongoing or recently completed, e.g. backups.'

2) Compilation

In `conf/VeriFitCompilation.properties`

1. configure host and port
2. set the triplestore host and port according to your triplestore configuration
3. set the triplestore endpoint URIs based on your triplestore info (from 1.1)
4. name your sparql graph
5. set to true if you are using a persistent dataset
6. authentication settings
7. automatic deletion of old resources

```

conf_example > VeriFitCompilation.properties
1  ## THIS IS A DEFAULT/EXAMPLE CONFIGURATION FILE
2  ##
3  #####
4  ## URL CONFIGURATION
5  #####
6  adapter_host=http://localhost 1
7  adapter_port=8081
8
9  #####
10 ## TRIPLESTORE CONFIGURATION
11 ##### 2
12 sparql_query=http://localhost:8082/fuseki/verifit_compilation/query 3
13 sparql_update=http://localhost:8082/fuseki/verifit_compilation/update 3
14 sparql_graph=http://verifit_compilation 4
15
16 #####
17 ## PERSISTENCY CONFIGURATION
18 #####
19 persist_sut_dirs=false 5
20
21 #####
22 ## AUTHENTICATION CONFIGURATION
23 #####
24 enable_authentication=false
25 username=user
26 password=user 6
27
28 #####
29 ## AUTOMATIC CLEAN UP CONFIGURATION
30 #####
31 keep_last_n_enabled=false
32 keep_last_n=100 7

```

3) Analysis

In conf/VeriFitAnalysis.properties

1. configure host and port
2. set the triplestore host and port according to your triplestore configuration
3. set the triplestore endpoint URIs based on your triplestore info (from 1.1)
4. name your sparql graph
5. authentication settings
6. automatic deletion of old resources

```

conf_example > VeriFitAnalysis.properties
1  ## THIS IS A EXAMPLE CONFIGURATION FILE
2  ##
3
4  #####
5  ## URL CONFIGURATION
6  #####
7  adapter_host=http://localhost 1
8  adapter_port=8080
9
10 #####
11 ## TRIPLESTORE CONFIGURATION
12 ##### 2
13 sparql_query=http://localhost:8082/fuseki/verifit_analysis/query 3
14 sparql_update=http://localhost:8082/fuseki/verifit_analysis/update
15 sparql_graph=http://verifit_analysis 4
16
17 #####
18 ## AUTHENTICATION CONFIGURATION
19 #####
20 enable_authentication=false
21 username=user 5
22 password=user
23
24 #####
25 ## AUTOMATIC CLEAN UP CONFIGURATION
26 #####
27 keep_last_n_enabled=false
28 keep_last_n=100 6

```

3.1) Analysis tool definition

In conf/analysis_advanced/AnalysisTools/YourTool.properties

1. full path to the executable of your tool
2. default arguments to be always used with your tool (optional)
3. set to true to enable queuing for your tool (fifo one at a time)

```

conf_example > analysis_advanced > AnalysisTools > ExampleTool.properties
1  ## Path to the tool executable
2  ## IMPORTANT: Use double backslash on windows! (\\ instead of just \)
3  toolLaunchCommand=\\full/path/to/executable.sh 1
4
5  ## Arguments to always use on the command line when launching the tool
6  ## (e.g. to make the tools output readable by the
7  ## adapter). These will always be placed as the first command line
8  ## parameter.
9  #toolSpecificArgs=--example 2
10
11 ## If set to true, then only one AutomationRequest executing this
12 ## AutomationPlan will be running at a time and the
13 ## remaining ones will be placed in a queue.
14
15 oneInstanceOnly=False 3

```

In conf/analysis_advanced/AnalysisTools/YourTool.rdf

1. uniquely name your automation plan (will be the last part of its URI)
2. define parameters for your automation plan
3. parameters need a unique name (unique within the automation plan), watch out for collisions with common adapter parameters
4. parameters whose values should be inserted to the command line need a cmdlinePosition
5. occurrence determines whether a parameter is optional
6. value type gives information to clients on what value the parameter expects
7. launchSUT is a special input parameter along with SUTbuildCommand
8. default values can be specified for any parameter

```
conf_example > analysis_advanced > AnalysisTools > ExampleTool.rdf
15 <oslc_auto:AutomationPlan>
16   <dcterms:identifier>example</dcterms:identifier> <!-- What the last part of this AutoPlan's URI will co
17   <oslc_auto:usesExecutionEnvironment rdf:resource="https://url.to.your.tool.com"/> <!-- Non-functional.
18   <dcterms:title rdf:parseType="Literal">Example Tool</dcterms:title> <!-- Non-functional. Title of the
19   <dcterms:description rdf:parseType="Literal">Used as an example.</dcterms:description> <!-- Non-functi
20   <dcterms:creator rdf:resource="https://url.to.the.creator.com"/> <!-- Non-functional. Creator of the Au
21   <oslc_auto:parameterDefinition>
22     <oslc_auto:ParameterDefinition>
23       <oslc:name>arguments</oslc:name>
24       <fit:commandlinePosition rdf:datatype="http://www.w3.org/2001/XMLSchema#int">1</fit:commandline
25       <oslc:occurs rdf:resource="http://open-services.net/ns/core#Exactly-one"/>
26       <oslc:valueType rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
27       <dcterms:description rdf:parseType="Literal">Specify which arguments should be passed to the co
28     </oslc_auto:ParameterDefinition>
29   </oslc_auto:parameterDefinition>
30   <oslc_auto:parameterDefinition>
31     <oslc_auto:ParameterDefinition>
32       <oslc:name>launchSUT</oslc:name>
33       <fit:commandlinePosition rdf:datatype="http://www.w3.org/2001/XMLSchema#int">2</fit:commandline
34       <oslc:defaultValue>True</oslc:defaultValue>
35       <oslc:occurs rdf:resource="http://open-services.net/ns/core#Zero-or-One"/>
36       <dcterms:description rdf:parseType="Literal">This parameter definitions tells the Automation Pl
37     </oslc_auto:ParameterDefinition>
38   </oslc_auto:parameterDefinition>
39   <oslc_auto:parameterDefinition>
40     <oslc_auto:ParameterDefinition>
41       <oslc:name>SUTbuildCommand</oslc:name>
42       <fit:commandlinePosition rdf:datatype="http://www.w3.org/2001/XMLSchema#int">3</fit:commandline
43       <oslc:defaultValue>True</oslc:defaultValue>
44       <oslc:occurs rdf:resource="http://open-services.net/ns/core#Zero-or-One"/>
45       <dcterms:description rdf:parseType="Literal">This parameter definitions tells the Automation Pl
46     </oslc_auto:ParameterDefinition>
47   </oslc_auto:parameterDefinition>
48 </oslc_auto:AutomationPlan>
```

3.2) Output filter definition

In conf/analysis_advanced/PluginFilters/YourFilter.properties

- 1. set the name of your filter class
- 2. specify the name of an Automation Plan which your filter is meant for

```
conf_example > analysis_advanced > PluginFilters > ExamplePluginFilter.properties
1 # do not change this value
2 implements=cz.vutbr.fit.group.verifit.oslc.analysis.outputFilters.IFilter
3
4 # class with implementation of the filter -- change the last part only to match your .java file
5 class=pluginFilters.customPluginFilters.ExamplePluginFilter
6
7 # says which tool is this filter meant for -- needs to match an AutomationPlan identifier
8 tool=example
```

In conf/analysis_advanced/PluginFilters/YourFilter.class

- 1. uniquely name your class implementing the required interfaces
- 2. uniquely name your filter
- 3. other filters can be used within your filter
- 4. optionally do some processing with the inputs of your filter
- 5. optionally create new contributions with unique IDs


```
conf_example > analysis_advanced > PluginFilters > ExamplePluginFilter.java
35 public class ExamplePluginFilter implements IFilter, IExtension {
36     final String name = "example";
37
38     public void filter(List<Map<String, String>> inoutContributions) {
39         // run Contributions through one of the builtin parsers first to load stdout and
40         new AddStdoutAndStderrValues().filter(inoutContributions);
41
42         // look for data race detection reports in stdout
43         Boolean dataRaceFound = false;
44         for (Map<String, String> contrib : inoutContributions)
45         {
46             String title = contrib.get("title");
47             if (title.equals("stdout"))
48             {
49                 String contentsOfTheStdout = contrib.get("value");
50                 if (contentsOfTheStdout.contains("Data race detected at")) // check if
51                 {
52                     dataRaceFound = true;
53                 }
54             }
55         }
56
57         // create a contribution representing the result (based on the stdout contents)
58         Map<String, String> contrib = new HashMap<String, String>();
59         contrib.put("id", "example_id"); // use any identifier you want (can be used
60         contrib.put("title", "DataRaceDetected"); // use any name you want
61         contrib.put("description", "Holds the result of data race analysis.");
62         contrib.put("value", dataRaceFound.toString());
63         contrib.put("valueType", "http://www.w3.org/2001/XMLSchema#boolean");
64         inoutContributions.add(contrib);
65     }
66
67     @Override
68     public String getName() {
69         return name;
70     }
71 }
```