

Simple Query Processing

In this tutorial you will learn elementary steps in using Odysseus.

Step 1: Install/Start Nexmark

As sources we will use the Nexmark examples source. See [Getting Started with Nexmark](#) for the installation of the nexmark server.

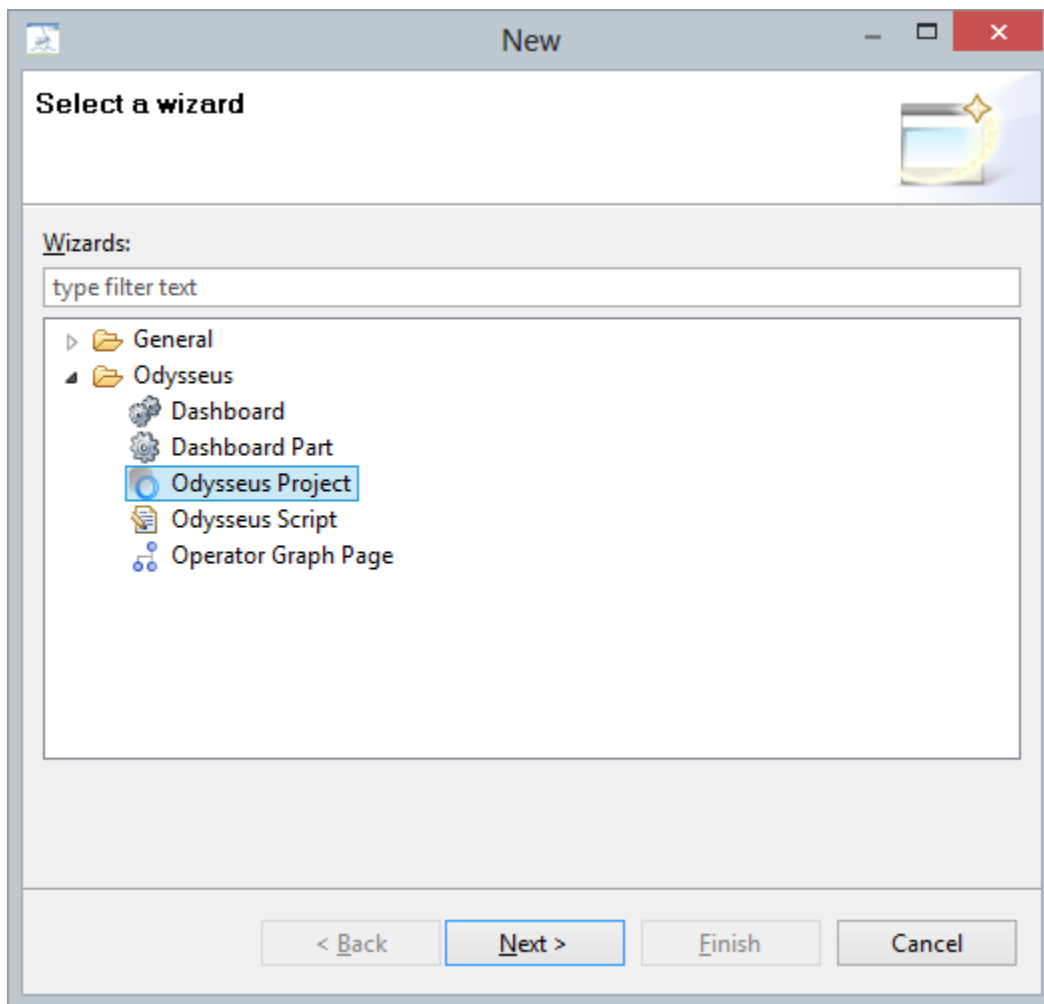
Step 2: Install/Start Odysseus

For this example you should use the standard Odysseus Studio (**Odysseus Server + Odysseus Studio Client**) (See [How to install Odysseus](#)).

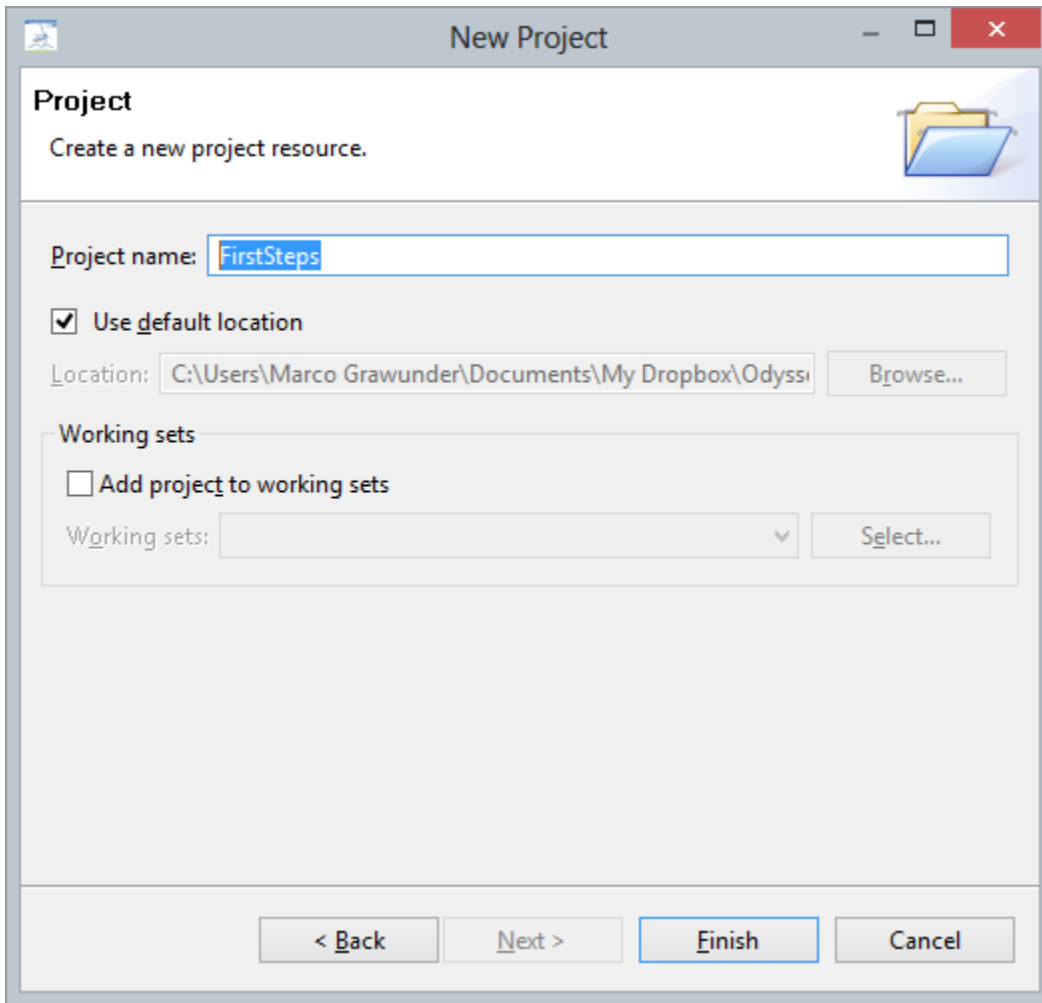
Step 3: Create a new project

Create a new project: FirstSteps

Select File/New... and choose Odysseus/Odysseus Project



Name it e.g. FirstSteps:



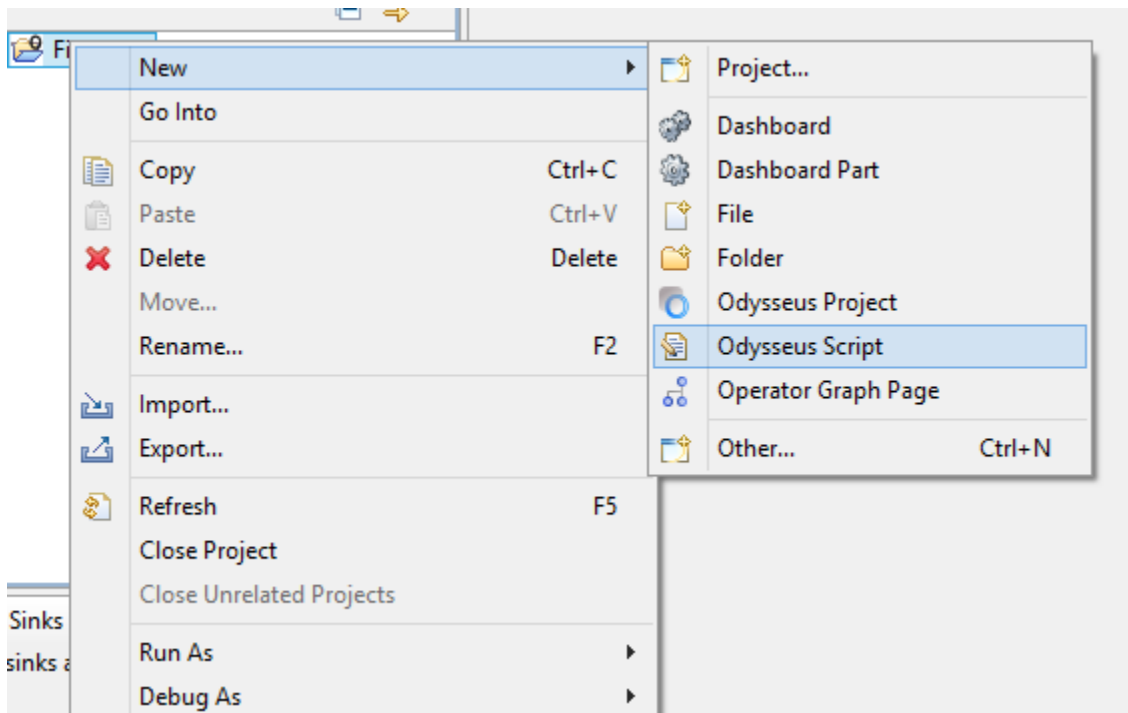
The new project is shown in the project explorer (hint if you do not see the project explorer, use Window/Show View/Other ... and choose General/Project Explorer).

Step 4: Describe sources

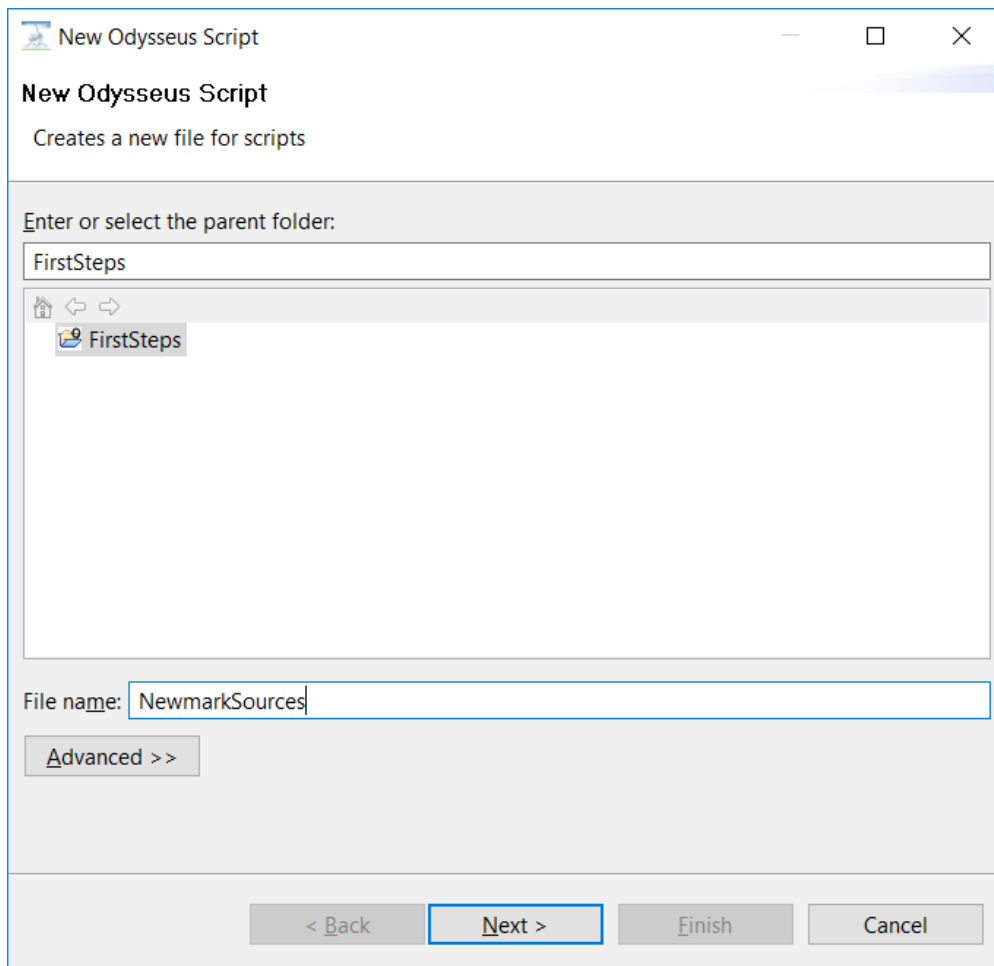
The queries will access sources that deliver events. In our example we will connect to the already started nexmark server. In this examples we will use [PQL](#) to create the sources. For detailed information about source definitions see [Access framework](#)

There are different ways to created sources/queries in Odysseus. We will use the [Odysseus Script](#) way.

Right click on the new created project and choose New/Odysseus Script



Give the script a name, e.g. NexmarkSources, and choose Next



There are different templates that can be used: Choose PQL Basic, and click on Finish.

New Odysseus Script

Template selection

Choose a template for your odysseus script to be preentered.

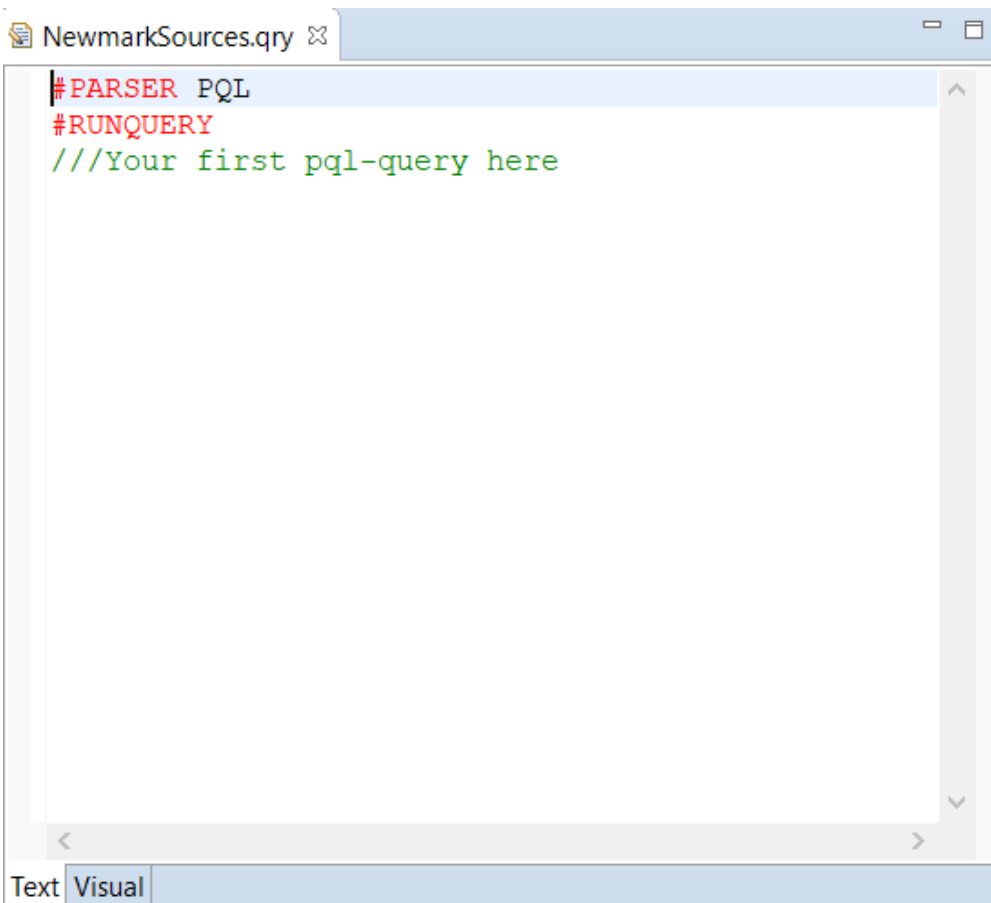
Template
PQL Basic

Description
Empty query with prepared PQL-Environment

Script
#PARSER PQL
#RUNQUERY
///*Your first pql-query here*

< Back Next > **Finish** Cancel

A new editor window will open as following



Now we have to tell Odysseus, how to access the different sources (person, bid, auction and category).

Start with person by typing the following in to the new editor (below #RUNQUERY):

```
nexmark:person := ACCESS({source='nexmark:person',
  wrapper='GenericPush',
  transport='TCPClient',
  protocol='SizeByteBuffer',
  dataHandler='Tuple',
  options=[
    ['host', 'localhost'],
    ['port', '65440'],
    ['ByteOrder', 'LittleEndian']
  ],
  schema=[
    ['timestamp', 'STARTTIMESTAMP'],
    ['id', 'INTEGER'],
    ['name', 'STRING'],
    ['email', 'STRING'],
    ['creditcard', 'STRING'],
    ['city', 'STRING'],
    ['state', 'STRING']
  ]
})
```


This statement describes a connection to the nexmark person server, delivered with Odysseus.

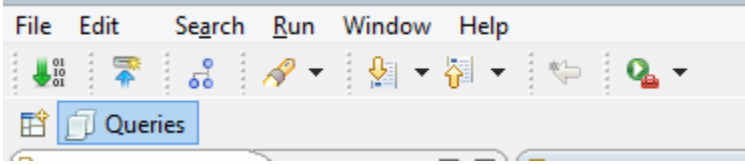
We create a new source with the name nexmark:person (and the type nexmark:person).

There are two types of parameters here. The first part is the physical connection to the server (We will not describe all parameters here (see [Access framework](#) for further information).If you have started the nexmark server on another machine change 'host' and if you started the person server on another port change 'port'.) and the second part describes the information this source delivers.

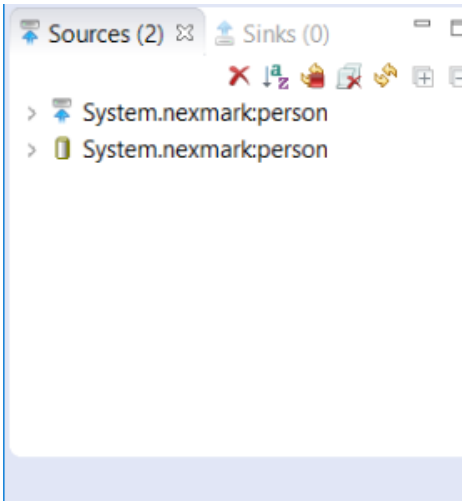
The most important part is the schema description. Here (somehow similar to a create table statement in SQL) the names and the types of the attributes are defined.

There are different built in datatypes available in Odysseus (see [Access framework](#) for further information). If the sources provides application time (as the nexmark server does), the datatype STARTTIMESTAMP can be used to state that a LONG value should be used as timestamp for this event.

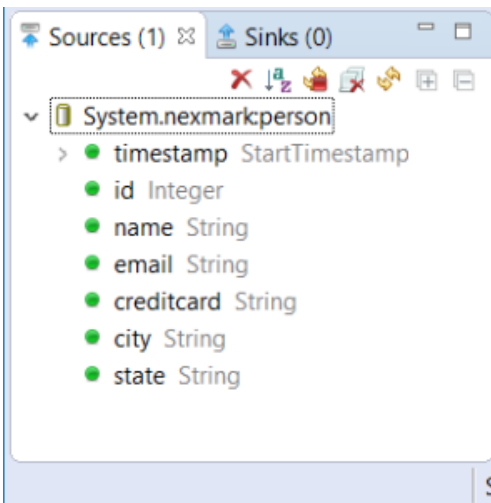
Now save the editor input (e.g. with CTRL-S) and run the query by using Execute Script (e.g. by right clicking on the file in the project explorer, on the editor it self or by using the button  in the toolbar)





If all is correct, there will be a new source and an access operator in the source-view.



You can use the arrow to show more information about the sources.



Now remove the sources again by clicking on the  symbol. It will remove all sources.  will remove the current marked source. A source can be removed by right clicking on that source and choosing: Drop Source, too.

The other sources are created the same way. To go further, add the following to the NexmarkSource editor (after the definition of the person source) and execute the script.

```

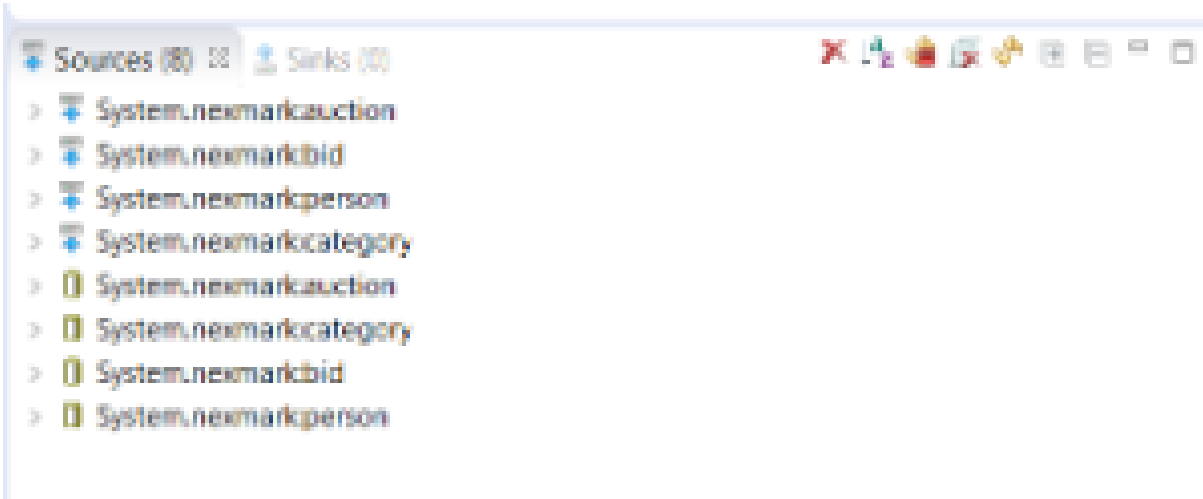
#RUNQUERY
nexmark:bid := ACCESS({source='nexmark:bid',
  wrapper='GenericPush',
  transport='TCPClient',
  protocol='SizeByteBuffer',
  dataHandler='Tuple',
  options=[
    ['host', 'localhost'],
    ['port', '65442'],
    ['ByteOrder', 'LittleEndian']
  ],
  schema=[
    ['nexmark:bid','timestamp', 'STARTTIMESTAMP'],
    ['nexmark:bid','auction', 'INTEGER'],
    ['nexmark:bid','bidder', 'INTEGER'],
    ['nexmark:bid','datetime', 'LONG'],
    ['nexmark:bid','price', 'DOUBLE']
  ]
})

#RUNQUERY
nexmark:auction := ACCESS({source='nexmark:auction',
  wrapper='GenericPush',
  transport='TCPClient',
  protocol='SizeByteBuffer',
  dataHandler='Tuple',
  options=[
    ['host', 'localhost'],
    ['port', '65441'],
    ['ByteOrder', 'LittleEndian']
  ],
  schema=[
    ['timestamp', 'STARTTIMESTAMP'],
    ['id', 'INTEGER'],
    ['itemname', 'STRING'],
    ['description', 'STRING'],
    ['initialbid', 'INTEGER'],
    ['reserve', 'INTEGER'],
    ['expires', 'LONG'],
    ['seller', 'INTEGER'],
    ['category', 'INTEGER']
  ]
})

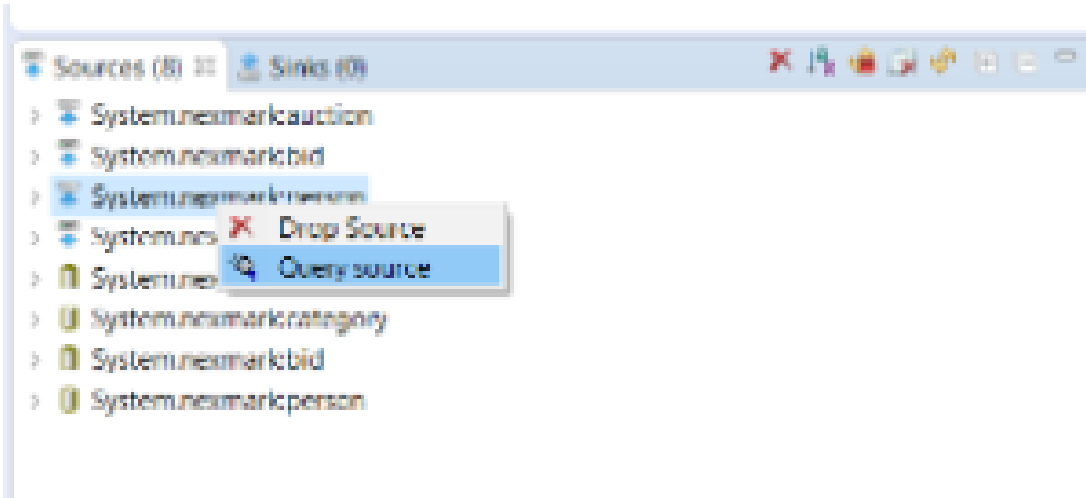
#RUNQUERY
nexmark:category := ACCESS({source='nexmark:category',
  wrapper='GenericPush',
  transport='TCPClient',
  protocol='SizeByteBuffer',
  dataHandler='Tuple',
  options=[
    ['host', 'localhost'],
    ['port', '65443'],
    ['ByteOrder', 'LittleEndian']
  ],
  schema=[
    ['id', 'INTEGER'],
    ['name', 'STRING'],
    ['description', 'STRING'],
    ['parentid', 'INTEGER']
  ]
})

```

After that you see a list of four sources and access operators in the source view:




To test, if everything is connected correctly, right click on System.nexmark:person and choose "Query Source"



A new window with person events should open. Your window should look somehow like this:

System....	System....	System....	System....	System....	System....	System....	Metadata
40000	7	Koos Do...	Docker...	4885 71...	Lima	North D...	40000 oo
35000	6	Jongho ...	Ginneke...	9018 90...	Tulsa	Huijsma...	35000 oo
30000	5	Wally R...	Roskies...	9374 83...	New	Oregon	30000 oo
25000	4	Luisa Be...	Belanch...	5538 80...	Miami	Utah	25000 oo
20000	3	Wing K...	Kusalik...	2536 14...	Idaho	Illinois	20000 oo
15000	2	Pius Sh...	Shmuel...	9106 91...	Gothern...	Vermont	15000 oo
10000	1	Rattasit ...	Schroer...	8116 36...	Abidjan	Schrodl	10000 oo
5000	0	Luitpold...	Martucc...	6573 12...	Pensacola	Georgia	5000 oo

Now close the window by clicking on .

If you choose show data again, the source will be beginning from start (but only if no other connection to the nexmark-server is open, e.g. via bids).

Step 5: Formulating Queries


Now we are ready to formulate queries over the nexmark sources. Queries in Odysseus are data pipelines. Every operator is responsible for a single subtask in the processing. It receives data from the operator nearer to the source and sends its output to all connected output operators (subscriber). In the following you will see simple examples of processing pipelines.

For this, create a new Odysseus Script file named query1 the same way as above.

We will use [Procedural Query Language \(PQL\)](#) in this example.

The simplest query is just to access all elements of the source. For this, write:

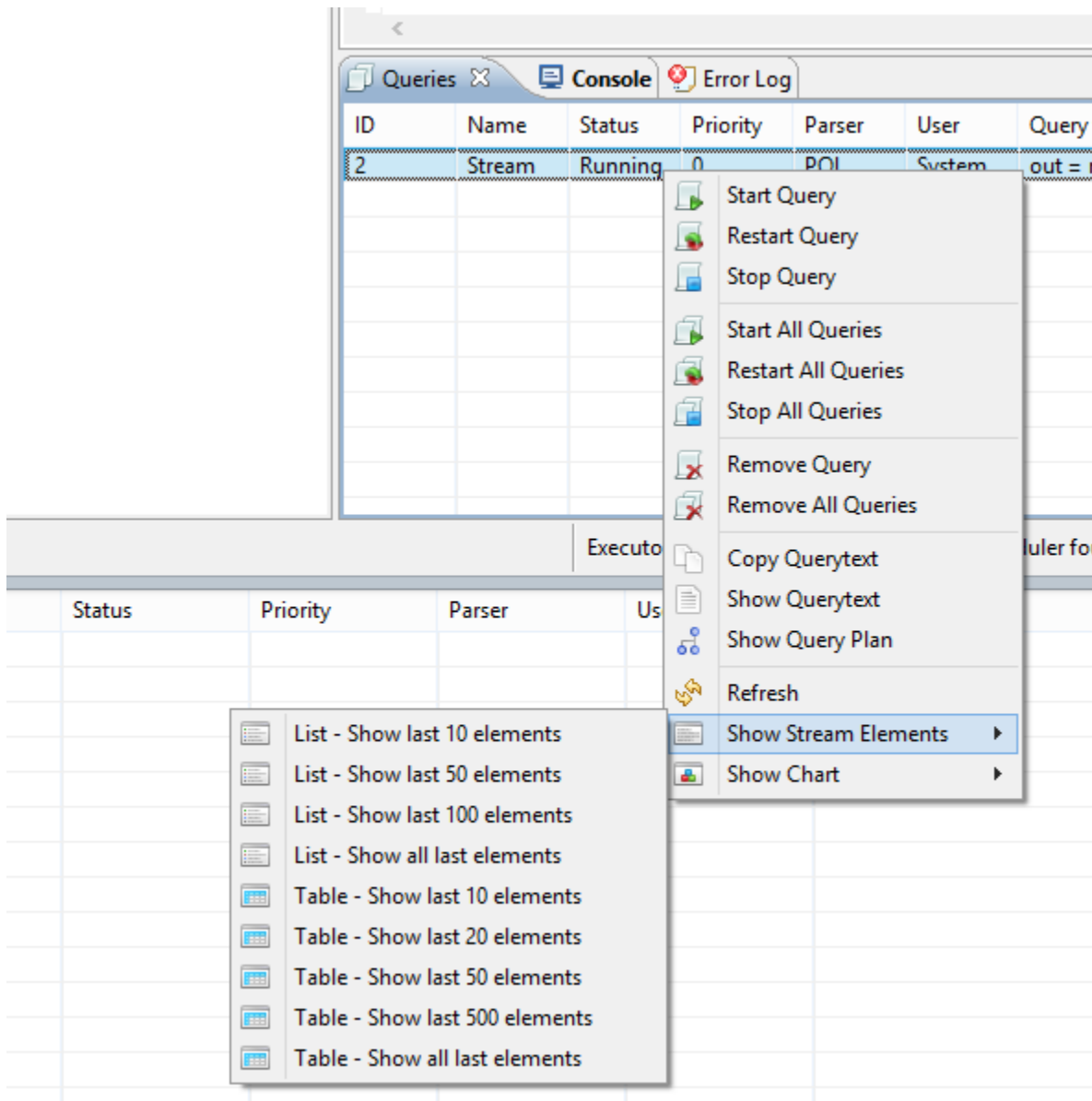
```
out = nexmark:person
```

save and execute the script with .

A new query should be seen in the Queries-view:

ID	Name	Status	Priority	Parser	User	Query text
2	Stream	Running	0	PQL	System	out = nexmark:person



Typically, the queries will run in the background. To see, what the query delivers, you can right right on the query in the Queries-View, select Show Stream Elements, and choose e.g. Table - Show last 50 elements:



Depending on the time, you click on the query, the current elements are shown. Hint: You have no chance to see older elements, because they are already processed.

The output should look somehow like:


System.nexmark:p...	System.nexmark:p...	System.nexmark:p...	System.nexmark:p...	System.nexmark:p...	System.nexmark:p...	System.nexmark:p...	Metadata
325000	64	Bashir Schahn	Schahn@panason...	6660 8478 8636 3503	Macon	New Hampshire	325000 oo
320000	63	Debendra McCon...	McConalogue@ct...	9901 6898 1783 2694	Reno	Chalfan	320000 oo
315000	62	Hideharu Chaudh...	Chaudhuri@fsu.edu	5033 9018 9666 3794	Richmond	North Dakota	315000 oo
310000	61	Tamal Kehrer	Kehrer@uni-mue...	6625 5878 8577 1763	Pensacola	Arizona	310000 oo
305000	60	Irek Mutschke	Mutschke@lante....	1519 5808 9396 6602	Greensboro	West Virginia	305000 oo
300000	59	Ljubomir Attimon...	Attimonelli@gte.c...	2442 1540 4225 1171	Aruba	Turnbull	300000 oo
295000	58	Norton Pesch	Pesch@lucent.com	3893 7845 6172 2612	Green	Nebraska	295000 oo
290000	57	Marjorie Tanishita	Tanishita@bellatla...	5588 8967 4334 5587	Porto	Jouvelot	290000 oo
285000	56	Tian Carpineto	Carpineto@cmu.e...	1333 6949 3380 9826	Knoxville	Virginia	285000 oo
280000	55	Wentong Nitta	Nitta@edu.cn	8685 9279 8308 1279	Greenville	Utah	280000 oo
275000	54	Wilbur Schiefer	Schiefer@upenn.e...	1207 3883 2810 8204	Asheville	Missouri	275000 oo
270000	53	Xiaogang Camari...	Camarinopoulos...	7060 1598 1978 8445	Manchester	Wisconsin	270000 oo
265000	52	Ataru Milicia	Milicia@concordi...	3514 7150 8379 3418	Mobile	Wind	265000 oo
260000	51	Shuhua Hobbs	Hobbs@broadque...	6209 8362 4113 7532	Kahului	Idaho	260000 oo
255000	50	Manjai Yoccoz	Yoccoz@uregina.ca	2040 4960 3157 4623	Warsaw	Bhashyam	255000 oo
250000	49	Fanya Matzen	Matzen@purdue.e...	5195 3198 5516 1545	Amarillo	Florida	250000 oo
245000	48	Anwar Rusmann	Rusmann@wisc.e...	7354 3926 7095 4652	Albany	Noga	245000 oo
240000	47	Yehuda Roeseler	Roeseler@ogi.edu	7814 8550 5594 3156	Brasilia	Montana	240000 oo

A query can be removed by selecting the query in the Queries view and choosing  in the toolbar above the query. With  all queries are removed.

You can also start and stop queries with the elements in the toolbar by choosing the seconds and the fourth button. Restarting (third button) is the same, as stopping and starting a query.

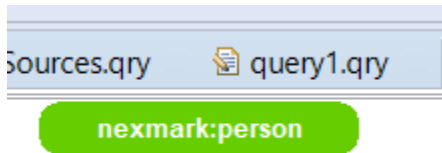


There are also buttons to do same same for all queries (the fifth, sixth and seventh button).

To see the current execution plan for a query double click on the query or choose from the context menu  Show Query Plan.

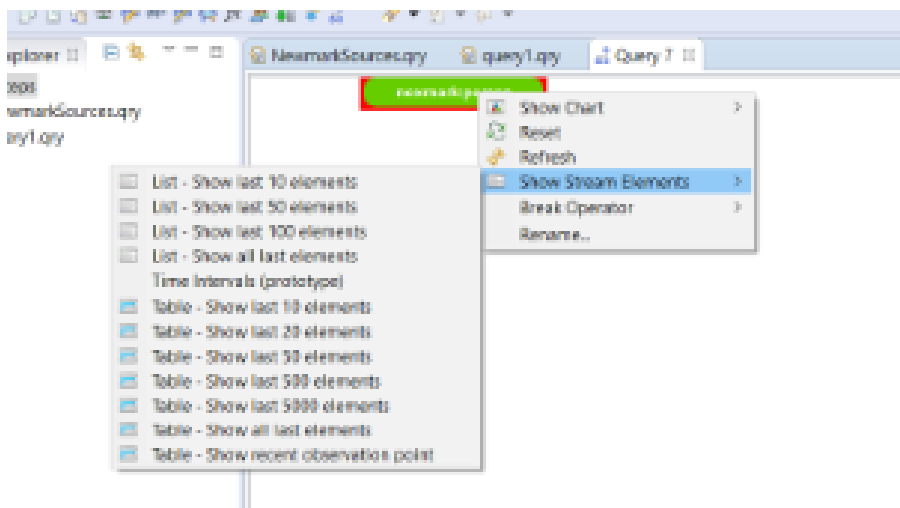
Hint: The button in the toolbar, shows the query plan for all current running queries (of this user).

The current query should look like in the following:

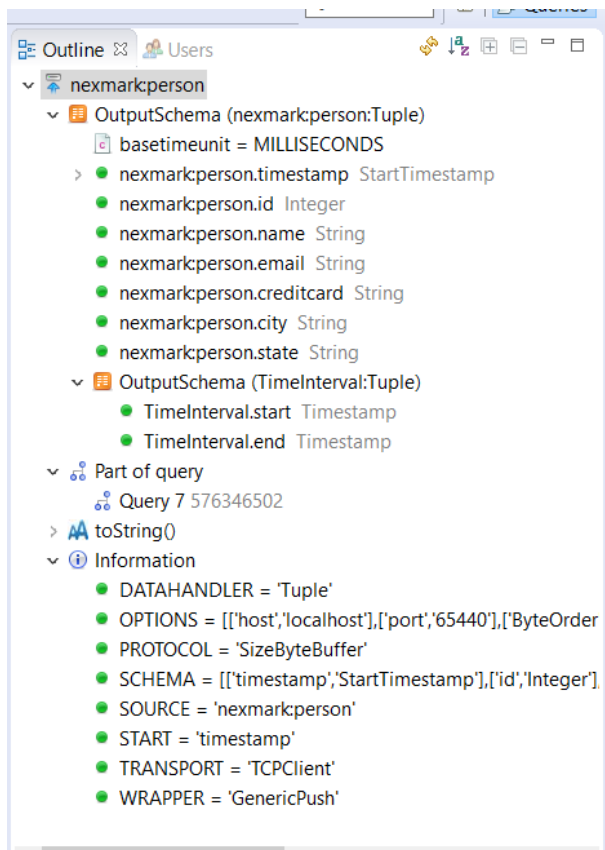


You will only see, a single operator, representing the access to the nexmark:person source.

You can right click on (each) operator in the view to see the data that this operator is currently processing (similar to the query source).



In the Outline-View you can see further information about each operator:



In the Operator Detail Info View further operator specific information can be found

The image shows a software window titled "Operator Detail Info" with a close button. It contains a tabbed interface with four tabs: "General", "OutputSchema", "ParameterInfo", and "Key-values". The "General" tab is active and displays the following configuration details:

Done	false
Type	Source
Class	ReceiverPO
Hash	1577475106
Name	nexmark:person
Owners	7
Open	true

At the bottom of the window, there is a "User" field.

This was a simple step through. More complex examples can be found in the [Tutorials](#) section.