



Connecting ITPilot to VDP using a JDBC Extractor

Revision 20200527

NOTE

This document is confidential and proprietary of **Denodo Technologies**.
No part of this document may be reproduced in any form by any means without prior written authorization of **Denodo Technologies**.

Copyright © 2023
Denodo Technologies Proprietary and Confidential

Note

ITPilot is deprecated and some or all components of ITPilot may be removed in the next major version.

See more details about this in the section [Features Deprecated in ITPilot 8.0](#).

Content

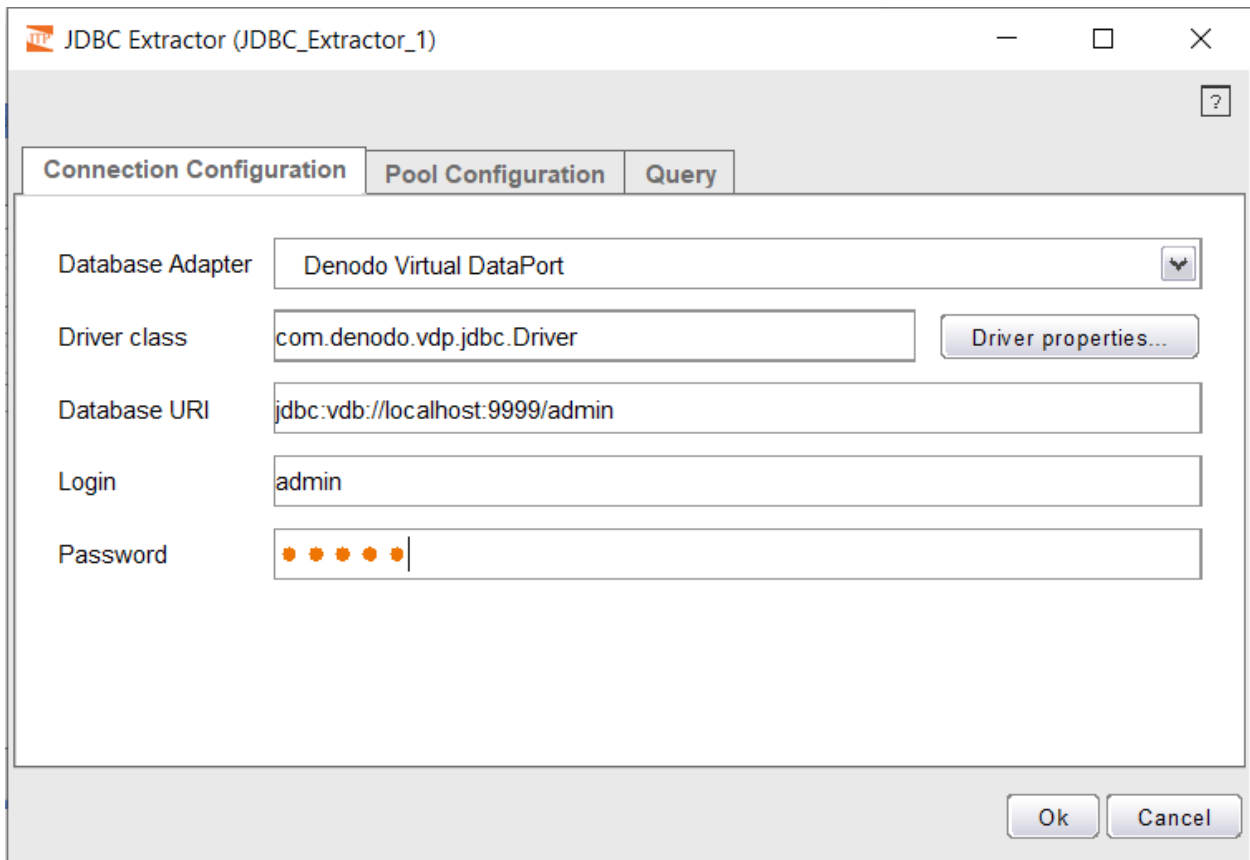
Sometimes an ITPilot wrapper needs to connect to external sources to obtain data useful for its execution. ITPilot provides a component that allows to send queries to any data source available through the JDBC protocol and returns a list of records containing the resultset from the query. This component is the JDBC Extractor.

This component accepts zero or more records and zero or more values as input arguments. These elements are used to assign variables to the component configuration parameters.

The output of the JDBC Extractor will be a record list, where the record structure is defined by the query performed on the data source.

If we need to get data from a VDP server, we can configure the JDBC Extractor component as follows:

1. Double click in the JDBC Extractor component and go to the “Connection Configuration” tab to set the connection details:
 - Database Adapter: Denodo Virtual Data Port
 - Driver Class: `com.denodo.vdp.jdbc.Driver`
 - Database URI: `jdbc:vdb://localhost:9999/<database_name>`
 - Login: User name.
 - Password: The user’s password.



JDBC Extractor (JDBC_Extractor_1)

Connection Configuration Pool Configuration Query

Database Adapter Denodo Virtual DataPort

Driver class com.denodo.vdp.jdbc.Driver Driver properties...

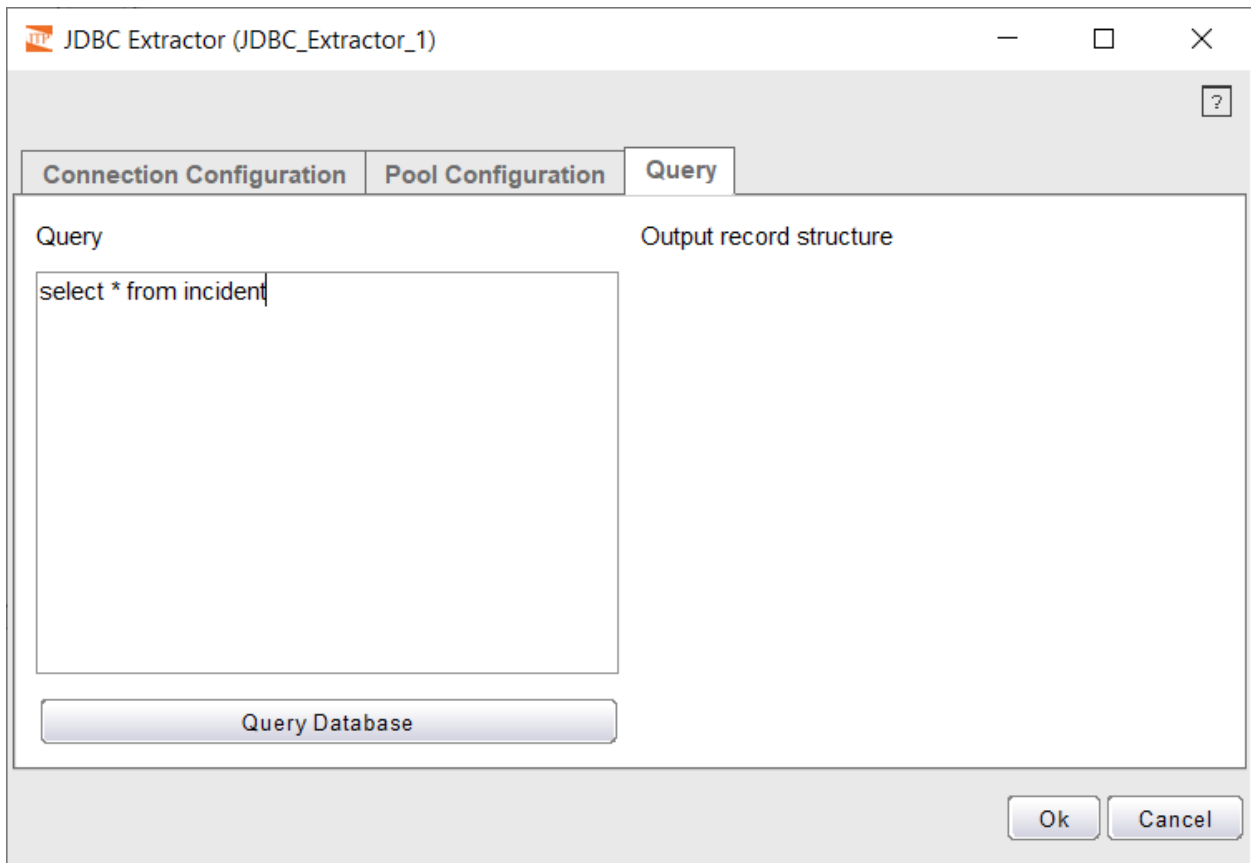
Database URI jdbc:vdb://localhost:9999/admin

Login admin

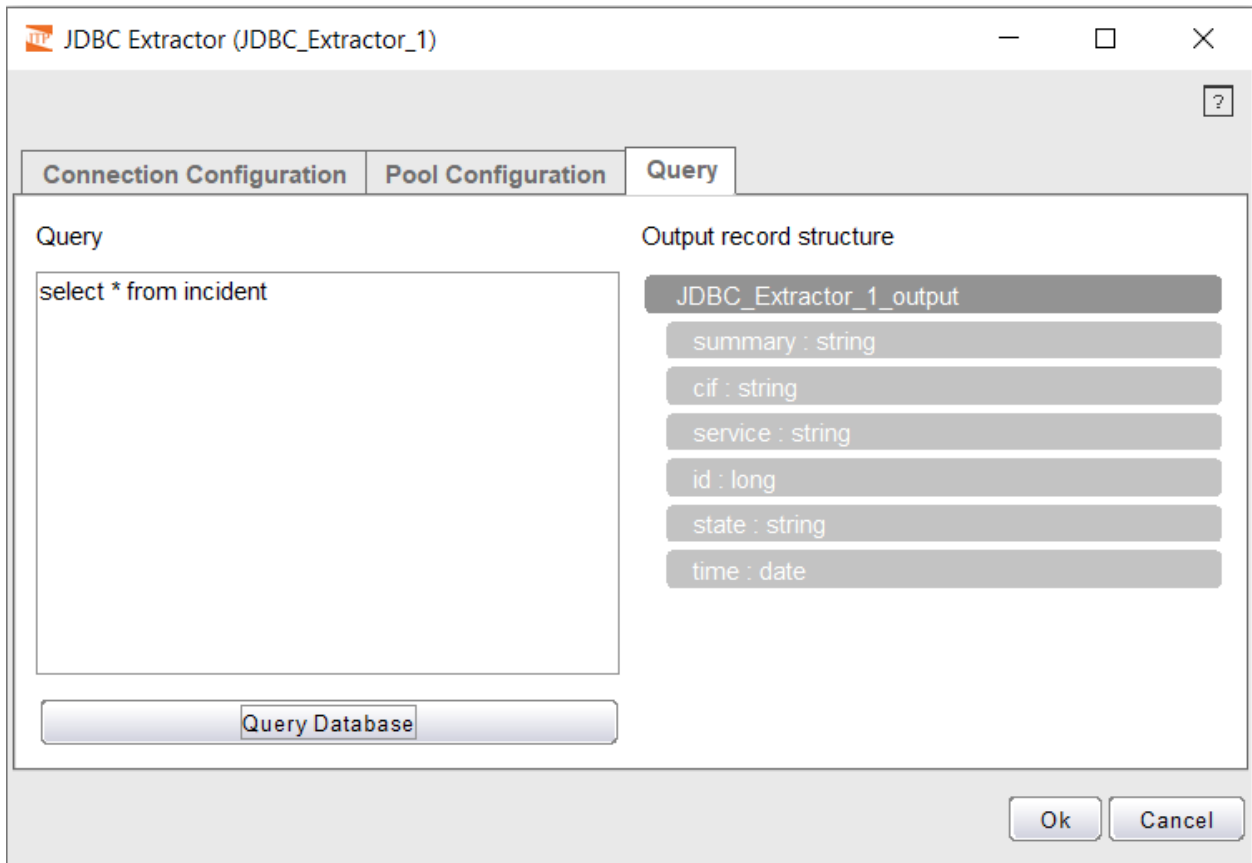
Password ●●●●●●

Ok Cancel

2. Go to the "Query" tab to specify the query to be executed in VDP.

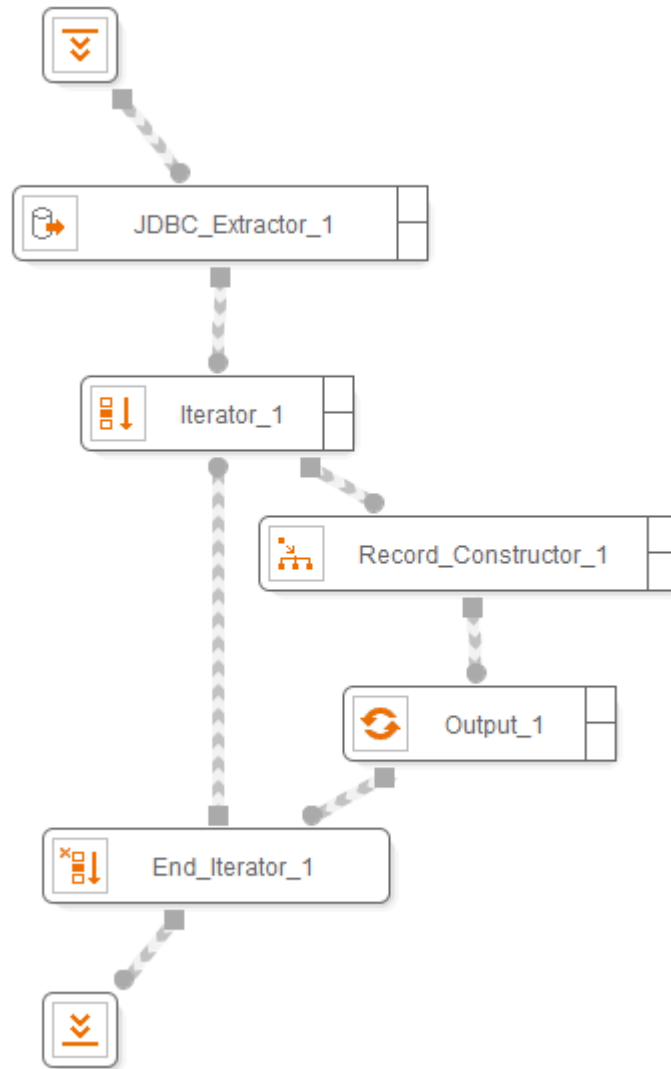


3. Click on the "Query database" button. In this step the JDBC Extractor component needs to execute a query to be able to obtain the structure that will be returned to the query and create the Output record structure.



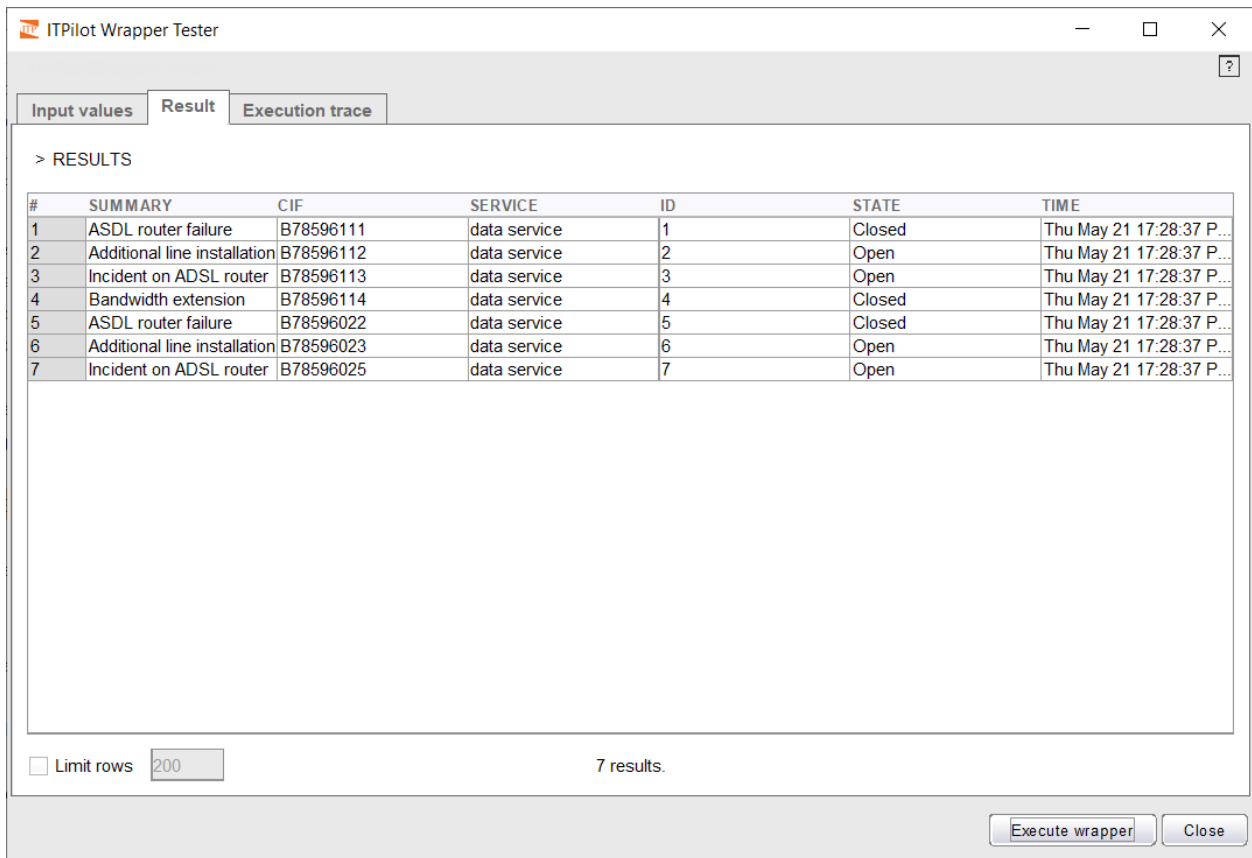
4. Click "Ok" to finish the component configuration.

As the output of the component is a record list, the results of the query will have to be iterated in a flow similar to the one below:



This is a very simple example of a process that executes a query and includes the results in the output of the wrapper.

This would be the result of the wrapper execution:



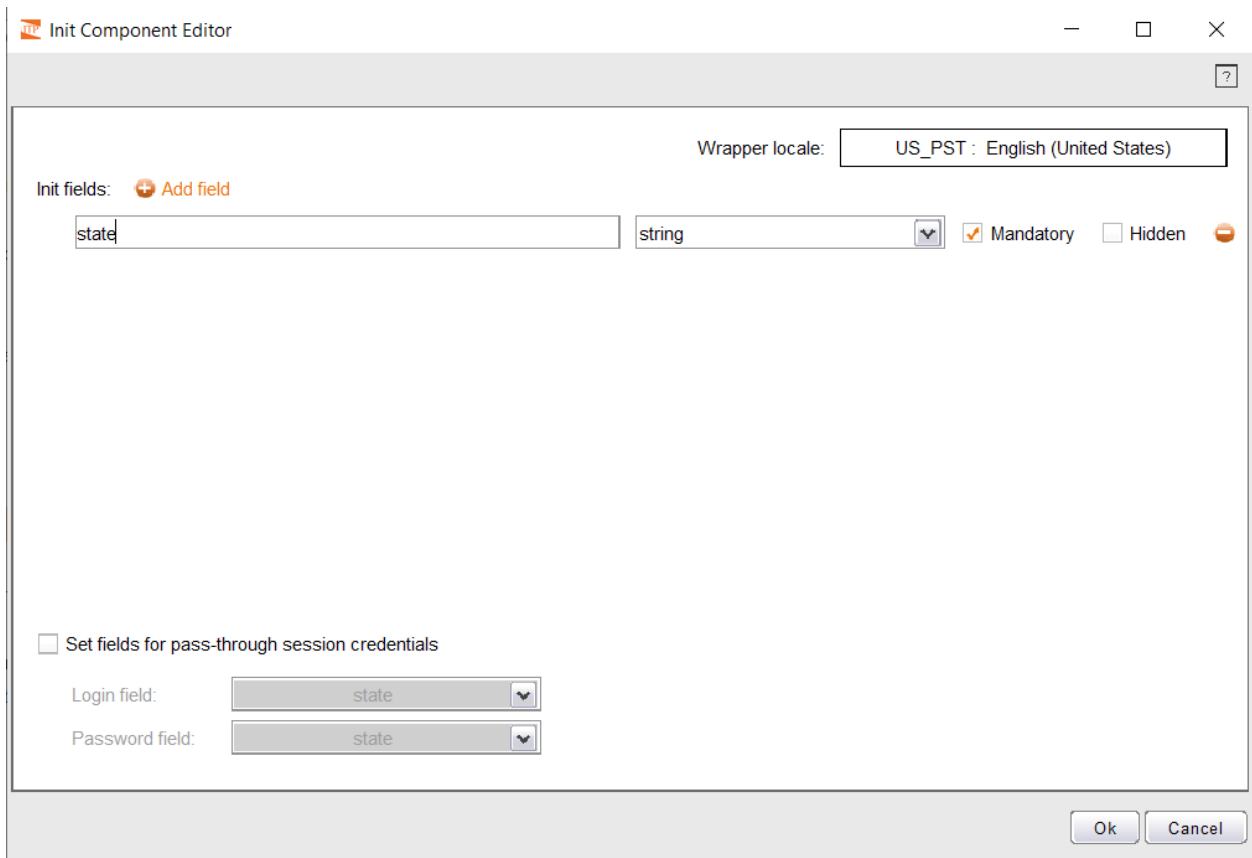
The screenshot shows a window titled "ITPilot Wrapper Tester" with three tabs: "Input values", "Result", and "Execution trace". The "Result" tab is active, displaying a table of results under the heading "> RESULTS".

| # | SUMMARY | CIF | SERVICE | ID | STATE | TIME |
|---|------------------------------|-----------|--------------|----|--------|--------------------------|
| 1 | ASDL router failure | B78596111 | data service | 1 | Closed | Thu May 21 17:28:37 P... |
| 2 | Additional line installation | B78596112 | data service | 2 | Open | Thu May 21 17:28:37 P... |
| 3 | Incident on ADSL router | B78596113 | data service | 3 | Open | Thu May 21 17:28:37 P... |
| 4 | Bandwidth extension | B78596114 | data service | 4 | Closed | Thu May 21 17:28:37 P... |
| 5 | ASDL router failure | B78596022 | data service | 5 | Closed | Thu May 21 17:28:37 P... |
| 6 | Additional line installation | B78596023 | data service | 6 | Open | Thu May 21 17:28:37 P... |
| 7 | Incident on ADSL router | B78596025 | data service | 7 | Open | Thu May 21 17:28:37 P... |

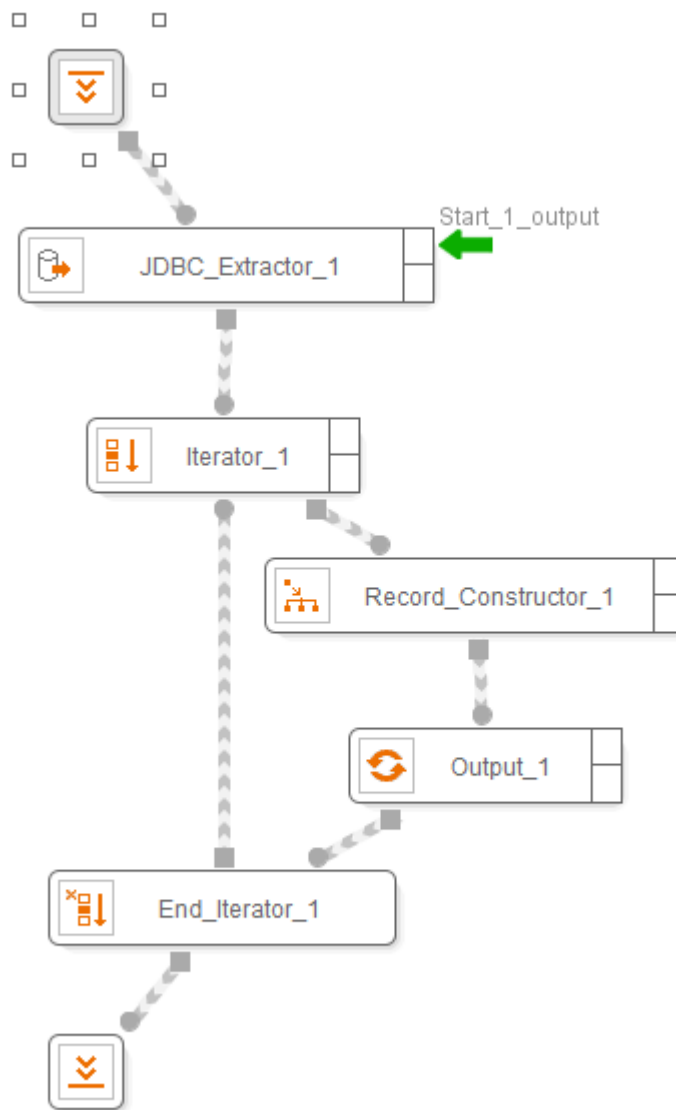
Below the table, there is a "Limit rows" checkbox set to "200" and the text "7 results.". At the bottom right, there are two buttons: "Execute wrapper" and "Close".

In this example, the same query will always be executed. It is also possible to parameterize the query using input values coming from the wrapper.

We are going to modify the example by adding an additional input parameter that we will use as an input for the JDBC Extractor component. Note that in this example we will use this input parameter as the input of the component, however any value created or extracted inside the wrapper can also be used as an input for the JDBC Extractor.

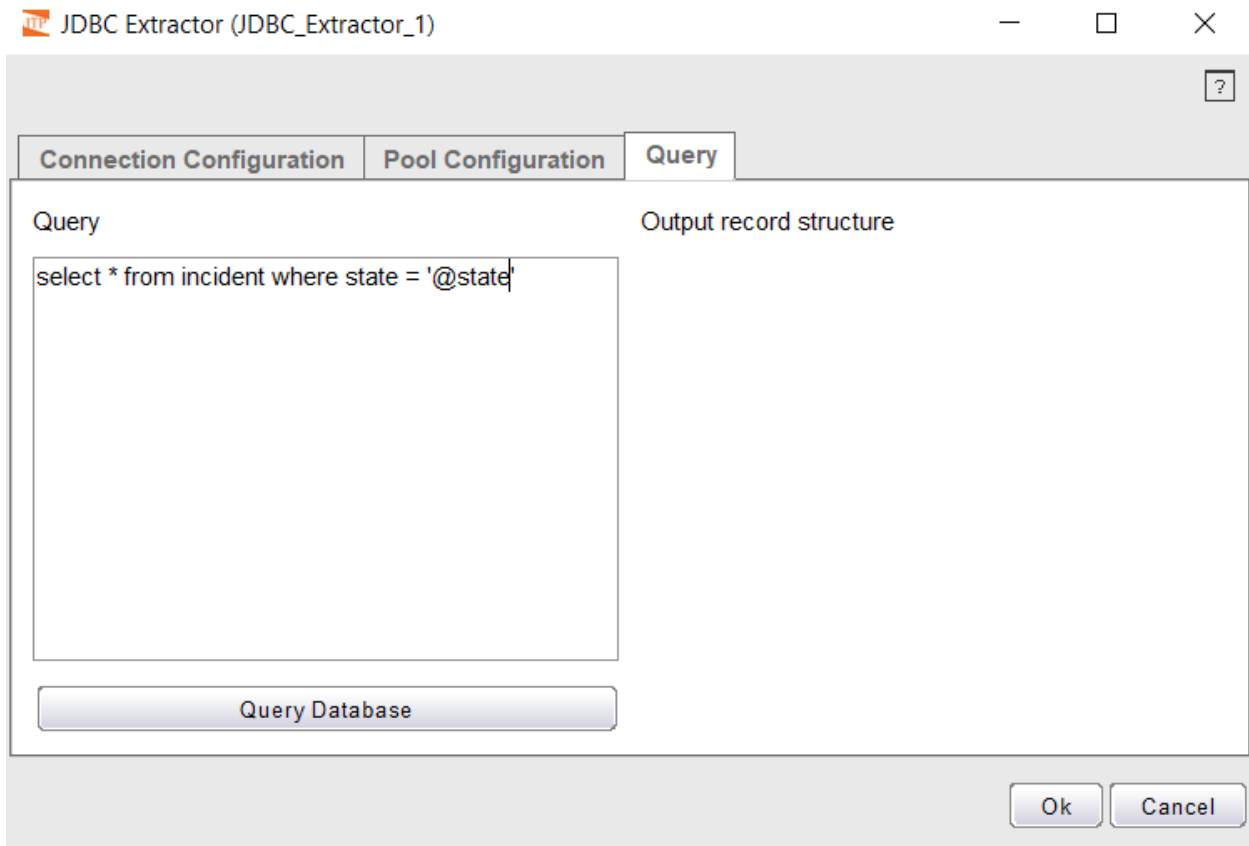


Once the parameter is defined, the JDBC Extractor can be modified by adding the input parameter as an input of the component and as a parameter of the query. In this example, the output of the Start component ("Start_1_Output" by default) will be the input value for the JDBC Extractor component.




By editing the query in the “Query” tab of the JDBC Extractor component, it is possible to specify the parameter using the syntax '@<parameter_name>' with the '@' character preceding the name of the parameter. For example:

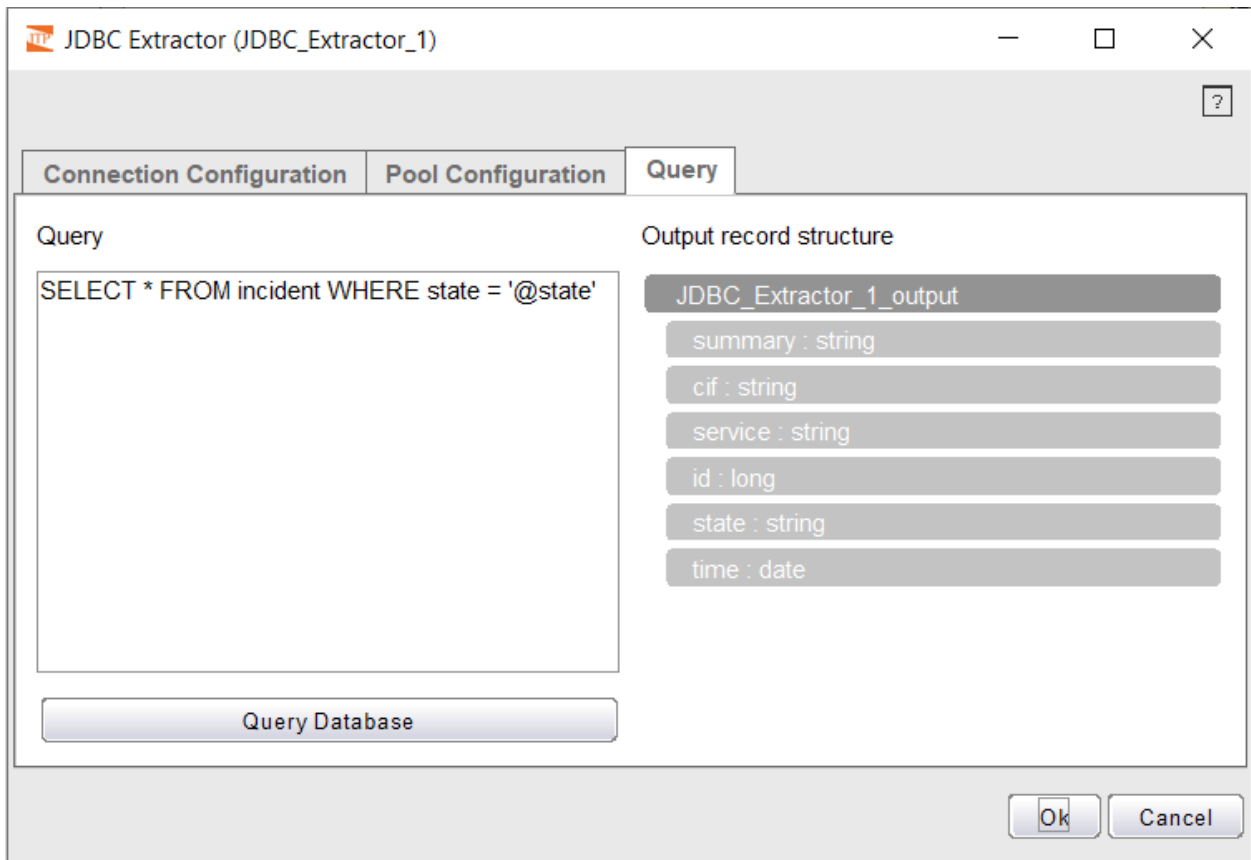
```
SELECT * FROM incident WHERE state = '@state'
```



When clicking the “Query Database” button, we will need to specify a state to allow the component to execute the query and get its output schema.

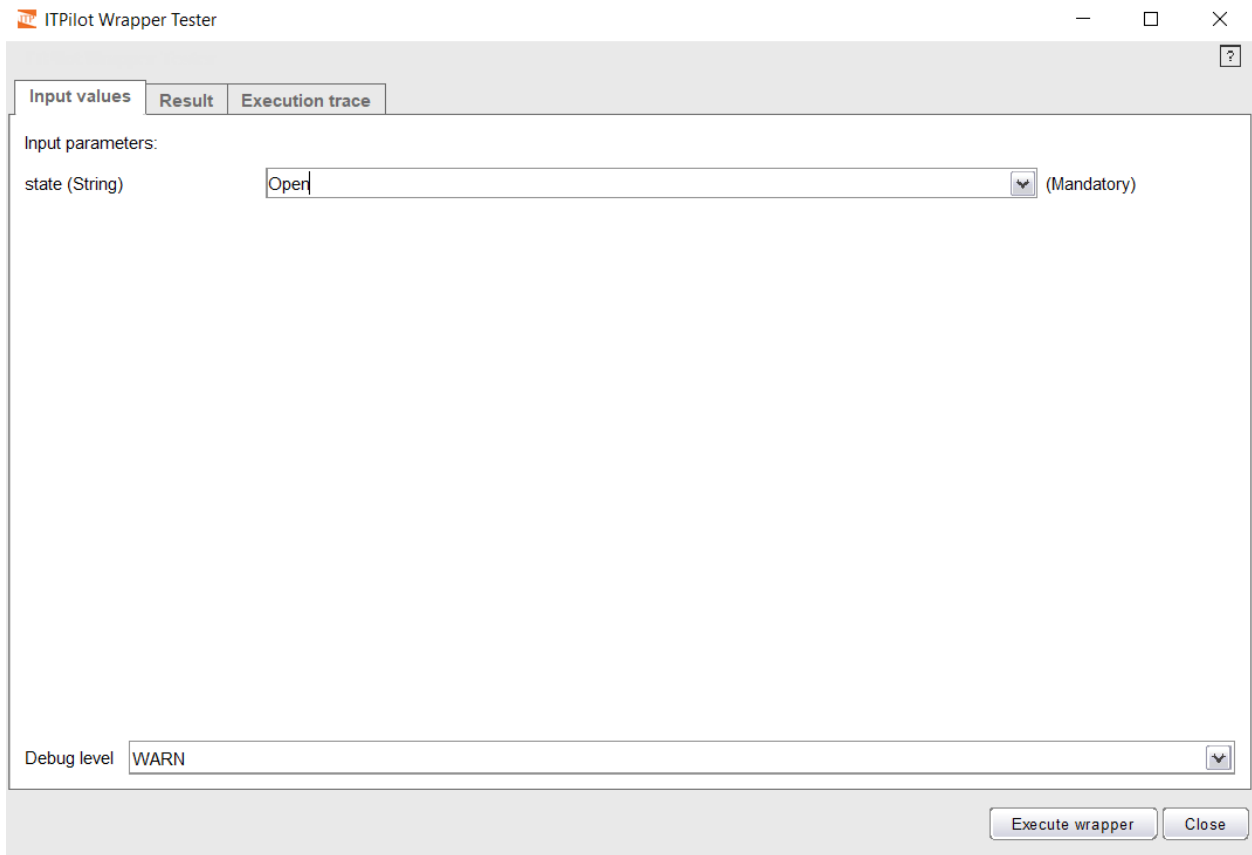
 Edit query values ✕

| NAME | VALUE | |
|-------|----------------------|---------------------------------|
| state | <input type="text"/> | <input type="checkbox"/> Hidden |



Click “Ok”. The wrapper will be configured to send the input parameter as a query parameter to the data source.

Now, to run the wrapper in the example, it will be necessary to provide a value for the input parameter:



The image shows a screenshot of a web-based application window titled "ITPilot Wrapper Tester". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Below the title bar, there are three tabs: "Input values", "Result", and "Execution trace", with "Input values" being the active tab. The main content area is titled "Input parameters:" and contains two input fields. The first field is labeled "state (String)" and has a dropdown menu with the value "Open" selected. To the right of this field is the text "(Mandatory)". The second field is labeled "Debug level" and has a dropdown menu with the value "WARN" selected. At the bottom right of the window, there are two buttons: "Execute wrapper" and "Close".

The output of the wrapper will contain the resultset of the query as before but with this version the query executed will be different depending on the value entered as input:

ITPilot Wrapper Tester

Input values | **Result** | Execution trace

> RESULTS

| # | SUMMARY | CIF | SERVICE | ID | STATE | TIME |
|---|------------------------------|-----------|--------------|----|-------|--------------------------|
| 1 | Additional line installation | B78596112 | data service | 2 | Open | Thu May 21 17:28:37 P... |
| 2 | Incident on ADSL router | B78596113 | data service | 3 | Open | Thu May 21 17:28:37 P... |
| 3 | Additional line installation | B78596023 | data service | 6 | Open | Thu May 21 17:28:37 P... |
| 4 | Incident on ADSL router | B78596025 | data service | 7 | Open | Thu May 21 17:28:37 P... |

Limit rows 4 results.

Execute wrapper Close

References

ITPilot Generation Environment Guide: section "[JDBC EXTRACTOR](#)"