ataccama

ONE Desktop Workshop

Debugging

Prepared for: v15.4.x

Prepared by: Ataccama

Dated: October 2024

Contents of the Document

1. Introduction 3

2. Tasks 4

2.1. Debugging expressions 4

2.2. Debugging steps 7

2.3. Debugging whole plans 8

3. Conclusion 9

1. Introduction

This workshop will show you how some of the Transformation steps that were used in the previous workshops can be tested and debugged before applying them to your plans.

In the previous activities, you were using the following steps:


Column Assigner


Regex Matching


Transliterate

We will also look at various debug options, namely:

A. Debugging a **single expression** field
B. Debugging a **whole step**
C. Debugging a **complete plan** together with all its logic

*The following activities require you to have completed the previous exercise (**ONE DESKTOP Workshop – Data Transformations**) so you have the previously created plans available.*

2. Tasks

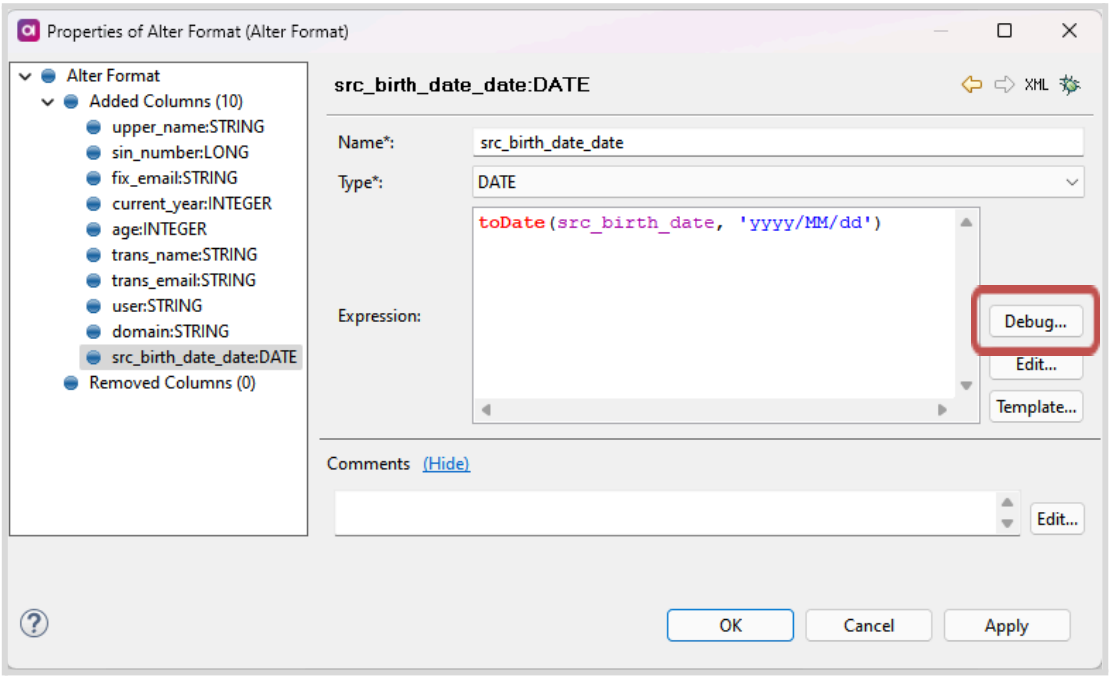
In this workshop, we will open the existing plan with the transformation steps you created before and look closely into ways how they can be tested.

2.1. Debugging expressions

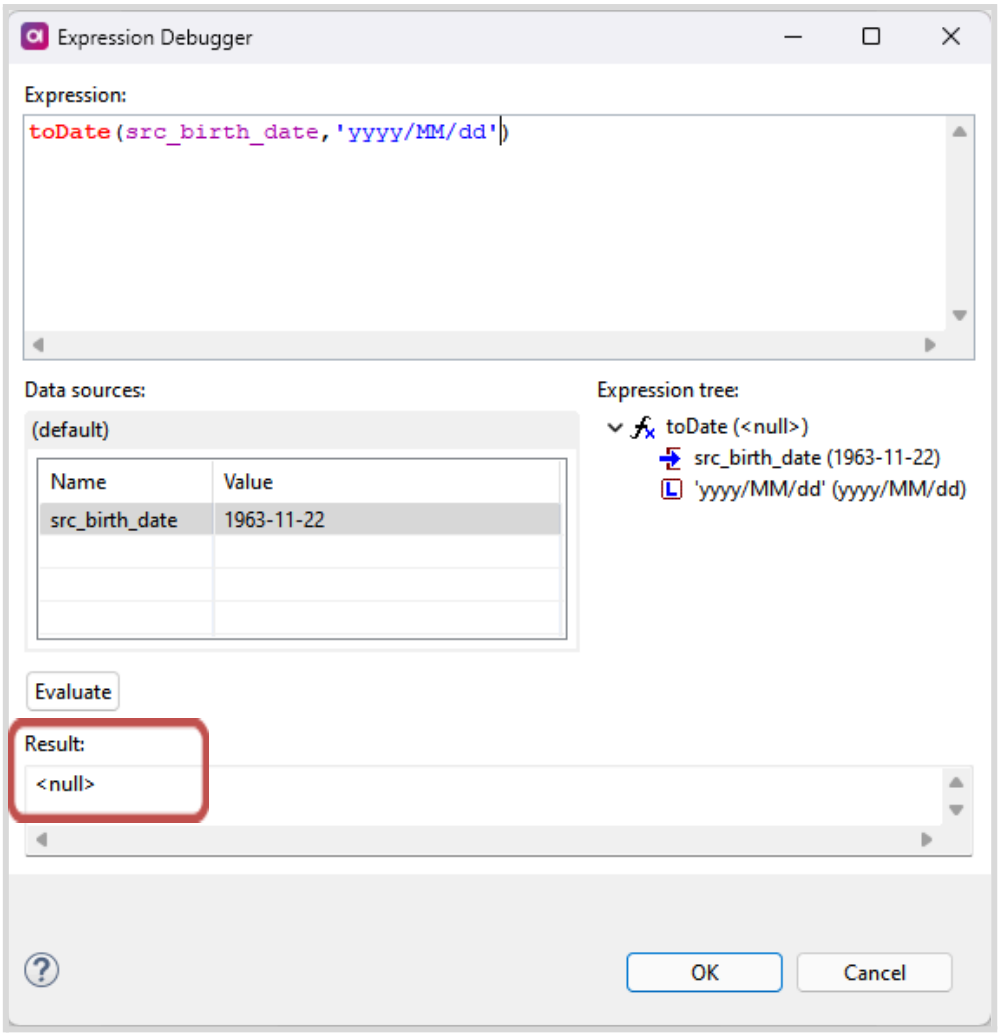
Sometimes it's needed to see the result of the applied function before it's executed within a plan. More complex logical operations will require testing and to ensure they process the data the way it should.

Let's see an example of how to debug an applied expression:

- Open the plan file from your previous workshop – it's likely called **03_party.plan**
- In the **Alter Format** step, add a new column **src_birth_date_date** in ***DATE*** format.
- Expand its row and add the following expression like this:



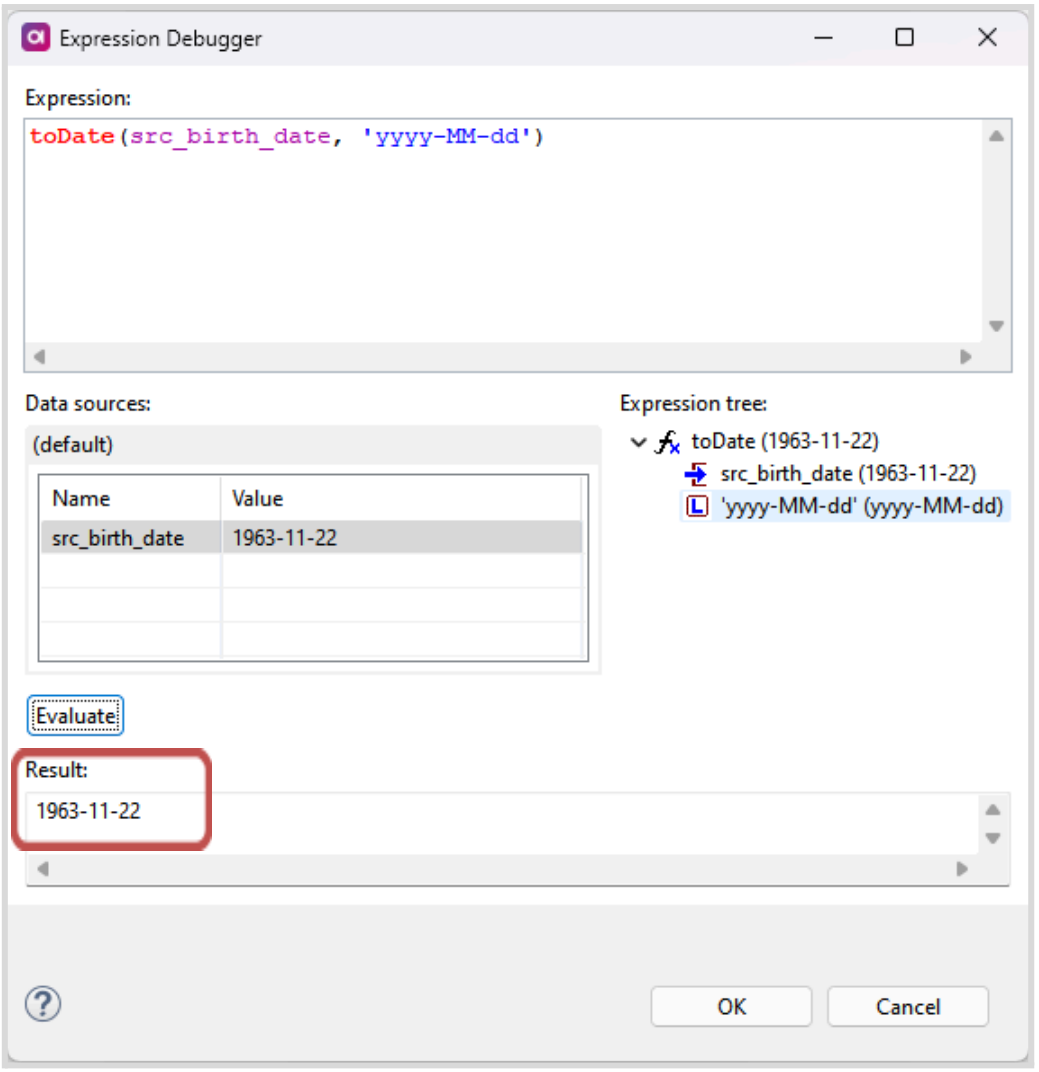
- Click the **'Debug...'** button on the right.
- Enter some sample values that appear in the source data and check how the expression evaluates it:



In the **Result:** window at the bottom you can see that the result is **<null>**. It turns out we have put an incorrect format mask in our expression!

- Fix the date format specified in the ***Expression***: field to a correct value and evaluate again.

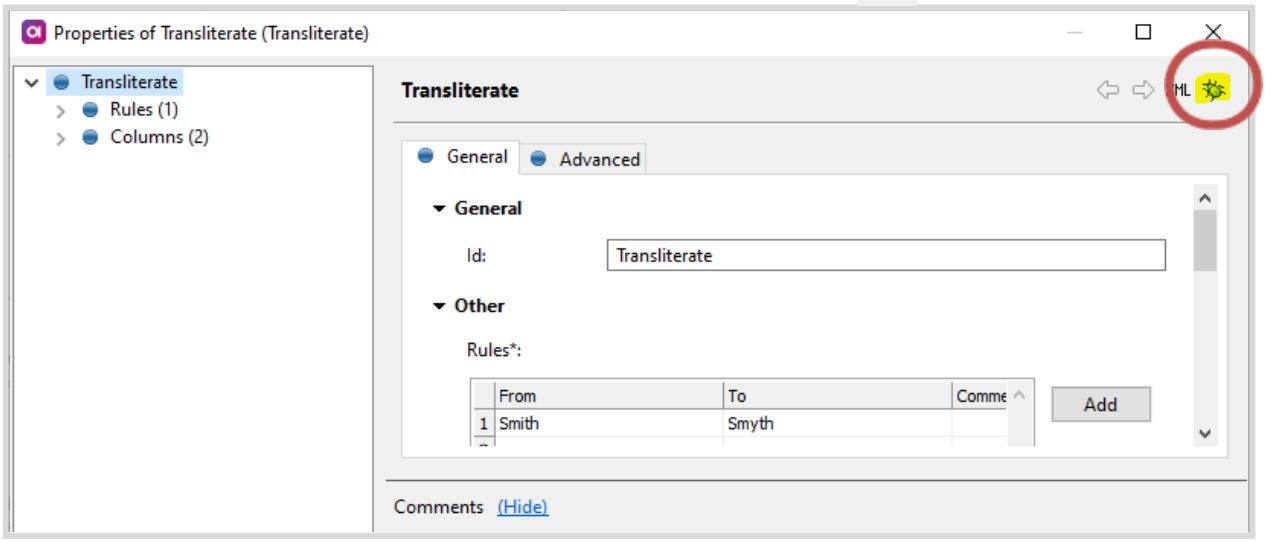
Much better! Your expression should now work as expected. Check the following window to verify your expression if in doubt:


Expression Debugger
Expression:
toDate(src_birth_date, 'yyyy-MM-dd')
Data sources:
(default)
Name
Value
src_birth_date
1963-11-22
Expression tree:
toDate (1963-11-22)
src_birth_date (1963-11-22)
'yyyy-MM-dd' (yyyy-MM-dd)
Evaluate
Result:
1963-11-22
OK
Cancel

2.2. Debugging steps

Sometimes debugging a single expression or a single function assignment is not sufficient. Let's test how to debug the function of the complete logic set appearing within a single step:

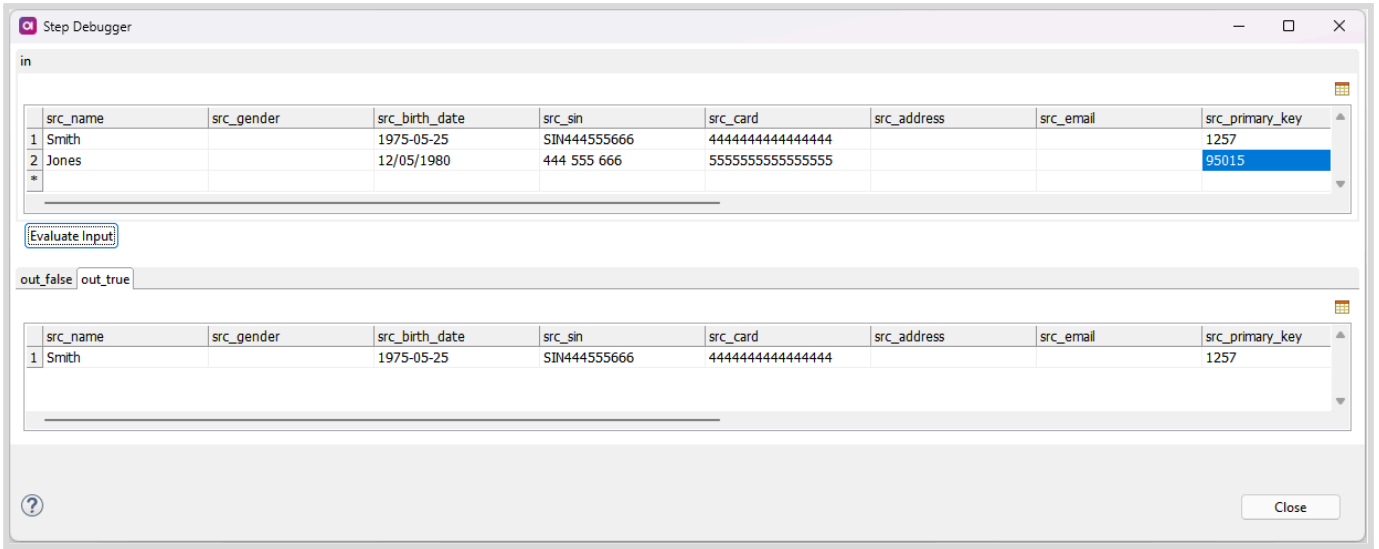
› Go to the **Transliterate** step and click the debug icon on the top right:



HINT

*Alternatively, you can right-click on a step and click the **Debug** option.*

› Enter values in the selected row(s) and see how the step evaluates the whole record:



2.3. Debugging whole plans

For cases where a complete solution within the whole plan should be tested, but you can't deliver the plan's output to the endpoint (e.g., a DB table with production data, generated file, etc.), you can also use a **Multiplicator** step and write a copy of the results into a separate text file to debug as you go along. This way you can test your definition and then swap the destination endpoint from the temporary output to the original one.

3. Conclusion

We have come to the end of this workshop. We have covered a few ways to test the output of expressions before deploying plans with modifications into production.

In the next workshop, we will start to evaluate the quality of our data and apply scores and explanations to give different weights to various types of DQ issues.