MANTA Flow Visualization User Documentation

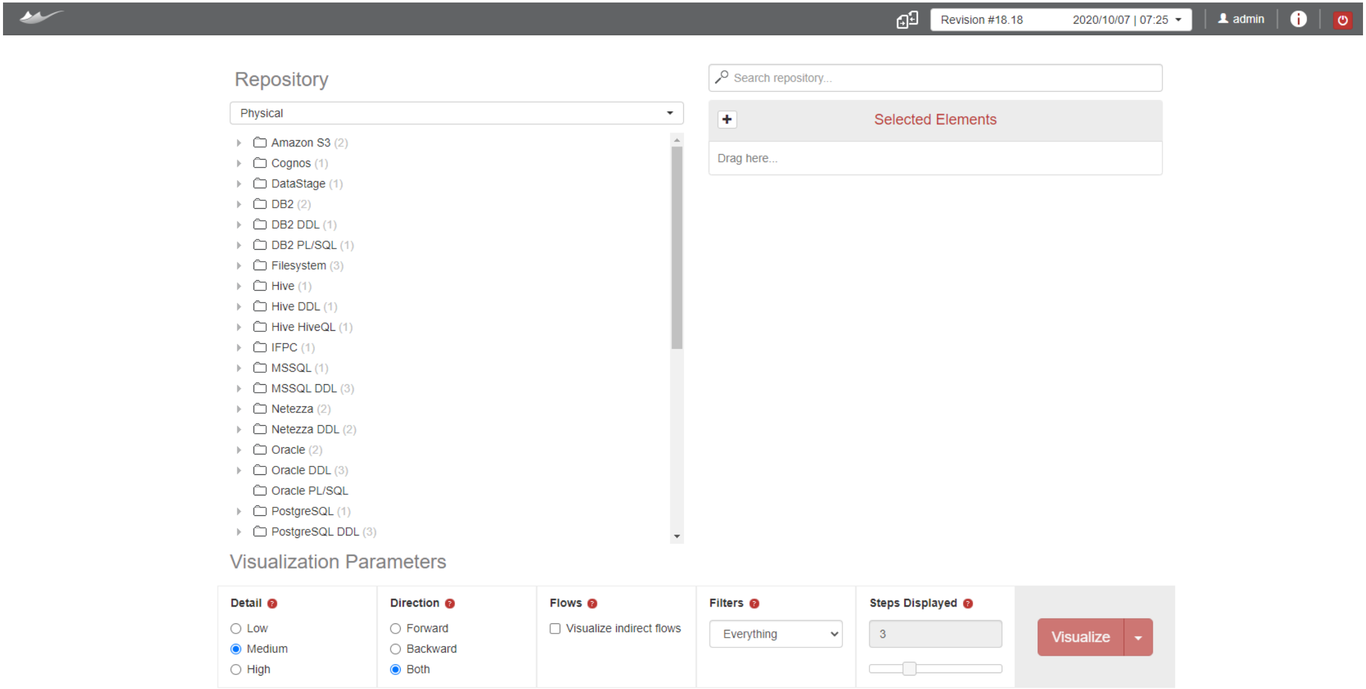
Object Catalog

To connect to the MANTA Flow Viewer application, enter this URL into a supported web browser

```
http://<server_name>:<port>/manta-dataflow-server/viewer
```

where the `<server_name>` and `<port>` are provided by your application administrator.

The main page of the application should open, as shown in the following image.



Revision #18.18 2020/10/07 | 07:25

admin

Repository

Physical

- Amazon S3 (2)
- Cognos (1)
- DataStage (1)
- DB2 (2)
- DB2 DDL (1)
- DB2 PL/SQL (1)
- Filesystem (3)
- Hive (1)
- Hive DDL (1)
- Hive HiveQL (1)
- IFPC (1)
- MSSQL (1)
- MSSQL DDL (3)
- Netezza (2)
- Netezza DDL (2)
- Oracle (2)
- Oracle DDL (3)
- Oracle PL/SQL
- PostgreSQL (1)
- PostgreSQL DDL (3)

Search repository...

Selected Elements

Drag here...

Visualization Parameters

Detail: ○ Low ● Medium ○ High

Direction: ○ Forward ○ Backward ● Both

Flows: □ Visualize indirect flows

Filters: Everything

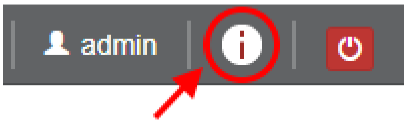
Steps Displayed: 3

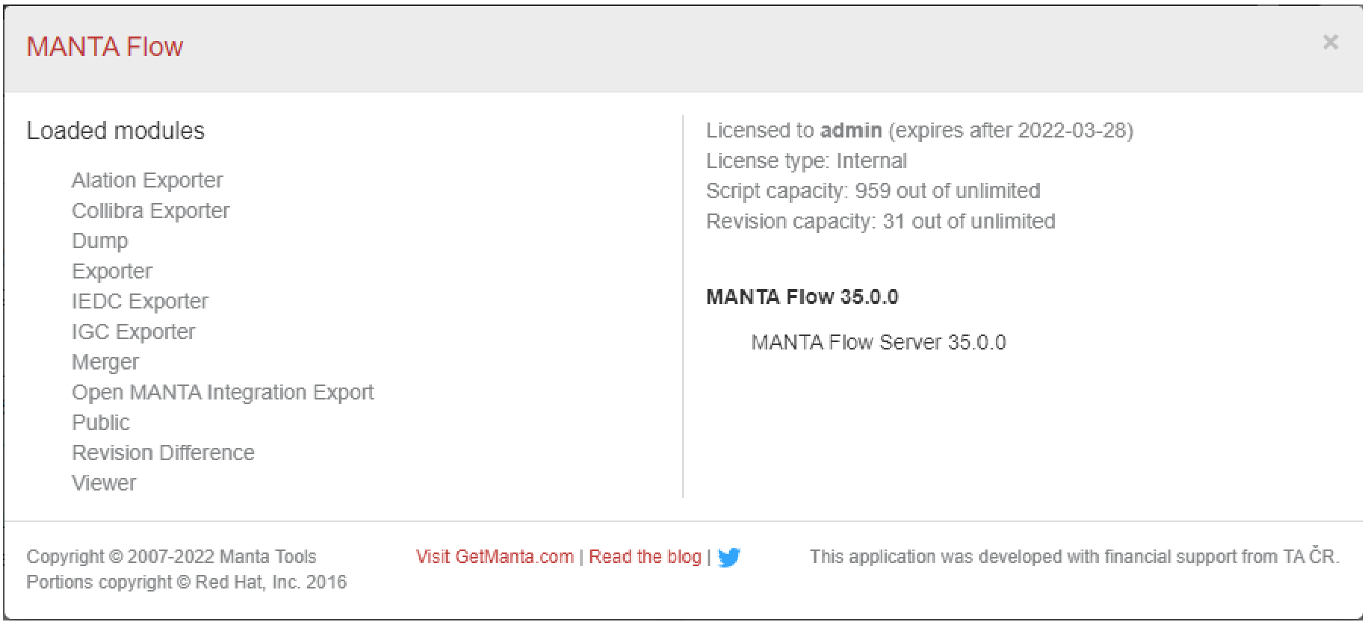
Visualize

There is an object catalog on the left side; a revision bar, a search bar, and a list of selected elements at the top; and a panel with visualization parameters at the bottom. In this section, all these elements will be introduced in this order.

Information about the Application and License

In the top right corner, you will find the *About* button, which will open a pop-up window with all the information about the application and its license.





MANTA Flow

Loaded modules

- Alation Exporter
- Collibra Exporter
- Dump
- Exporter
- IEDC Exporter
- IGC Exporter
- Merger
- Open MANTA Integration Export
- Public
- Revision Difference
- Viewer

Licensed to **admin** (expires after 2022-03-28)
License type: Internal
Script capacity: 959 out of unlimited
Revision capacity: 31 out of unlimited

MANTA Flow 35.0.0

MANTA Flow Server 35.0.0

Copyright © 2007-2022 Manta Tools
Portions copyright © Red Hat, Inc. 2016

Visit GetManta.com | Read the blog |

This application was developed with financial support from TA ČR.

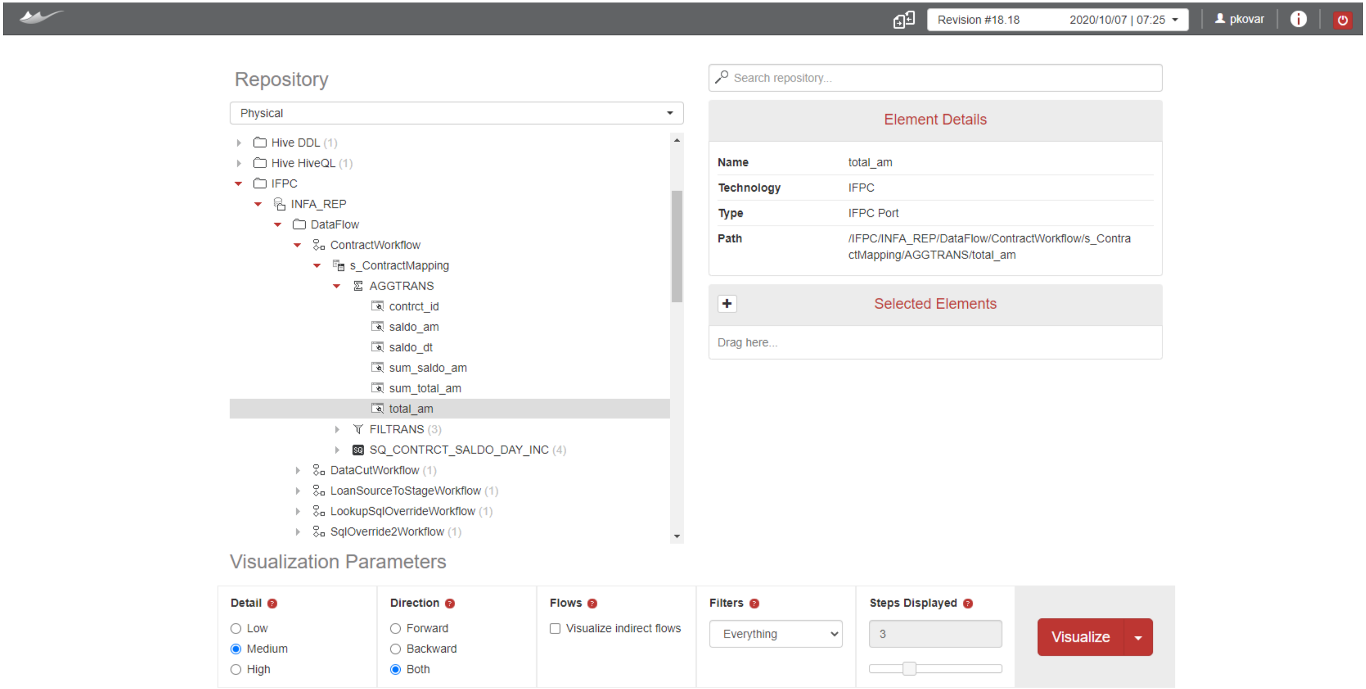
There is a list of all modules available for your version of the application on the left side. All the information about your license (including the licensee, expiration date, license type, and information about the capacities of the scripts and revisions) is on the right side. Information about the versions of the install package and MANTA Flow Server is also on the right, and you can find links to our website and social networks at the bottom of the window.

Browsing and Searching the Object Catalog

The object catalog contains all the elements that MANTA Flow has found and analyzed in your environment. In general, there are two groups of objects. The first group consists of real objects such as directories, files, databases, tables, columns, etc. The second group consists of imaginary objects that transform the real ones such as workflows, scripts, transformations, commands, etc. The root elements of the catalog join all the objects from one resource together. For example, there is a resource for all files containing data (elements from the first group) called *Filesystem*, and there is a resource containing all workflows and sessions from Informatica PowerCenter (elements from the second group) called *IFPC*.

Each element belongs to a particular layer (e.g., tables and files belong to the physical layer, assets belong to the business layer). In the case of two or more layers, the layer tabs are displayed above the element hierarchy. It is possible to select a layer by clicking on the appropriate tab. Only the elements belonging to the selected layer are visible.

It is possible to browse the hierarchy of objects by clicking on the arrow to the left of each element. For example, when you click on the arrow next to the IFPC element, all the repositories found are shown. On subsequent levels there are folders, workflows, sessions, transformations and, finally, ports. Each element has a type that is represented by an icon. You can click on an element to get further details such as its technology, type, and full path as well as the other attributes found in the *Element Details* box. (See the following illustration.)



Revision #18.18 2020/10/07 | 07:25 pkovar

Repository

Physical

- Hive DDL (1)
- Hive HiveQL (1)
- IFPC
  - INFA_REP
    - DataFlow
      - ContractWorkflow
        - s_ContractMapping
          - AGGTRANS
            - contrct_id
            - saldo_am
            - saldo_dt
            - sum_saldo_am
            - sum_total_am
            - total_am
          - FILTRANS (3)
          - SQ_CONTRCT_SALDO_DAY_INC (4)
      - DataCutWorkflow (1)
      - LoanSourceToStageWorkflow (1)
      - LookupSqlOverrideWorkflow (1)
      - SqlOverride2Workflow (1)

Search repository...

Element Details

Name	total_am
Technology	IFPC
Type	IFPC Port
Path	/IFPC/INFA_REP/DataFlow/ContractWorkflow/s_ContractMapping/AGGTRANS/total_am

Selected Elements

Drag here...

Visualization Parameters

Detail
○ Low
● Medium
○ High

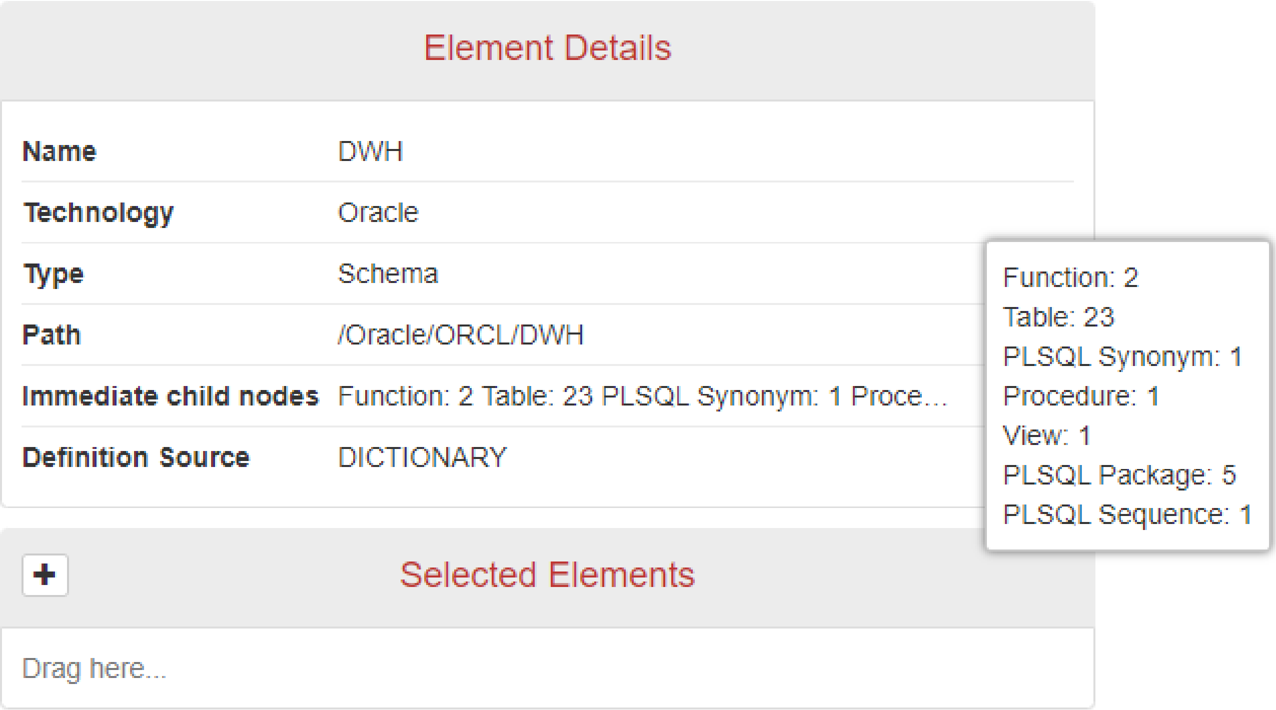
Direction
○ Forward
○ Backward
● Both

Flows
□ Visualize indirect flows

Filters
Everything

Steps Displayed
3

Each element has a number of immediate children in parentheses to the right of it. It is visible while an element is collapsed and is hidden after expanding the element. You can find detailed information about immediate children in the *Element Details* box, where they are also grouped by type.



Element Details

Name	DWH
Technology	Oracle
Type	Schema
Path	/Oracle/ORCL/DWH
Immediate child nodes	Function: 2 Table: 23 PLSQL Synonym: 1 Proce…
Definition Source	DICTIONARY

Function: 2
Table: 23
PLSQL Synonym: 1
Procedure: 1
View: 1
PLSQL Package: 5
PLSQL Sequence: 1

Selected Elements

Drag here...

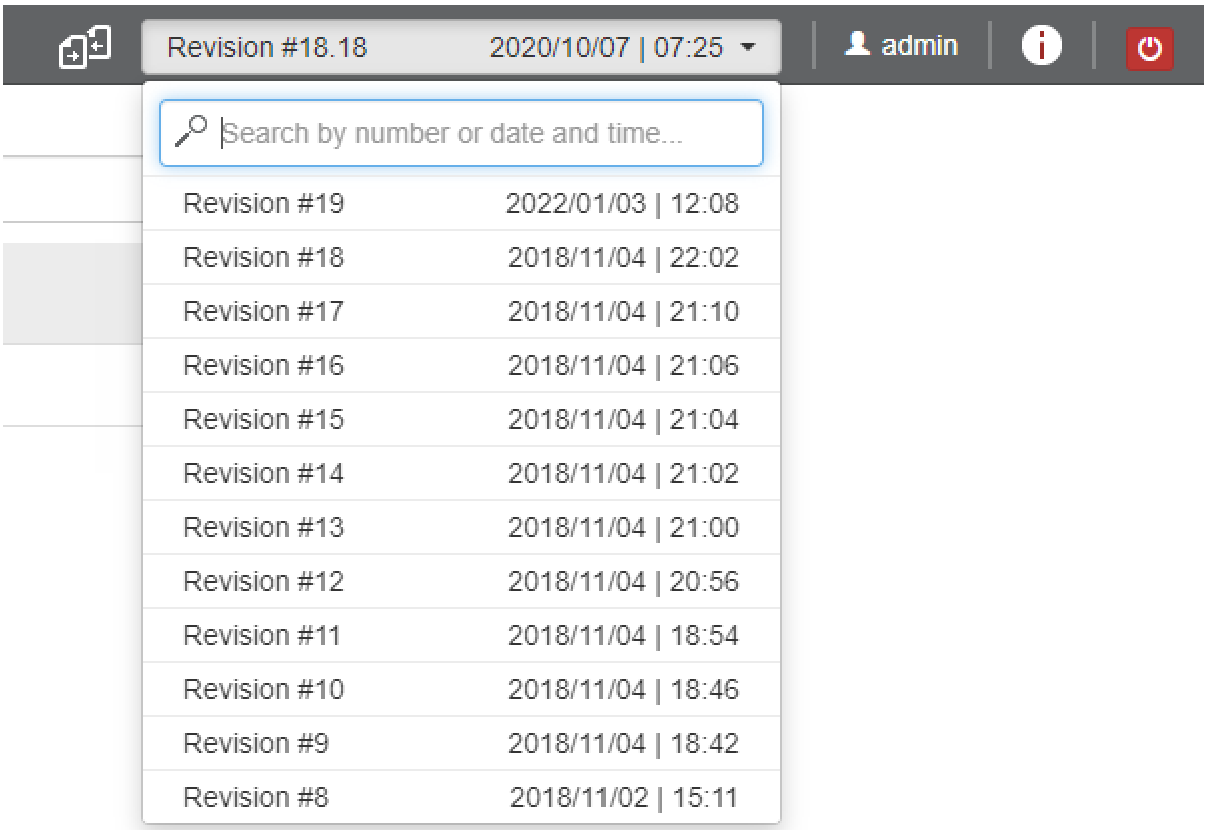
If you do not know exactly where to find the requested element, it is possible to find it using its name (or part of it). For example, if you want to find a particular column that is in a table with a name containing the word “customer”, just type this word into the search bar. If there are any elements in the repository with names containing this word, they will be shown in the list. In the case of two or more layers, only the elements belonging to the selected layer are shown. At the top of the list are the elements whose names exactly match the given word. Each element is further identified by its type (on the right) and its full path (on the left) so you can choose the correct table. If you click on an element, it will be shown in the object catalog and its details will also appear. Then you can expand the table in the catalog and select the requested column.

Selecting Element Versions

By default, you can see the latest version of the objects that were committed on the date displayed in the revision bar. Since the repository is versioned, you can also see previous versions of objects by selecting another revision from the the revision bar.

Searching for Revisions

By default, only the last revision and major revisions are displayed in the list of revisions.



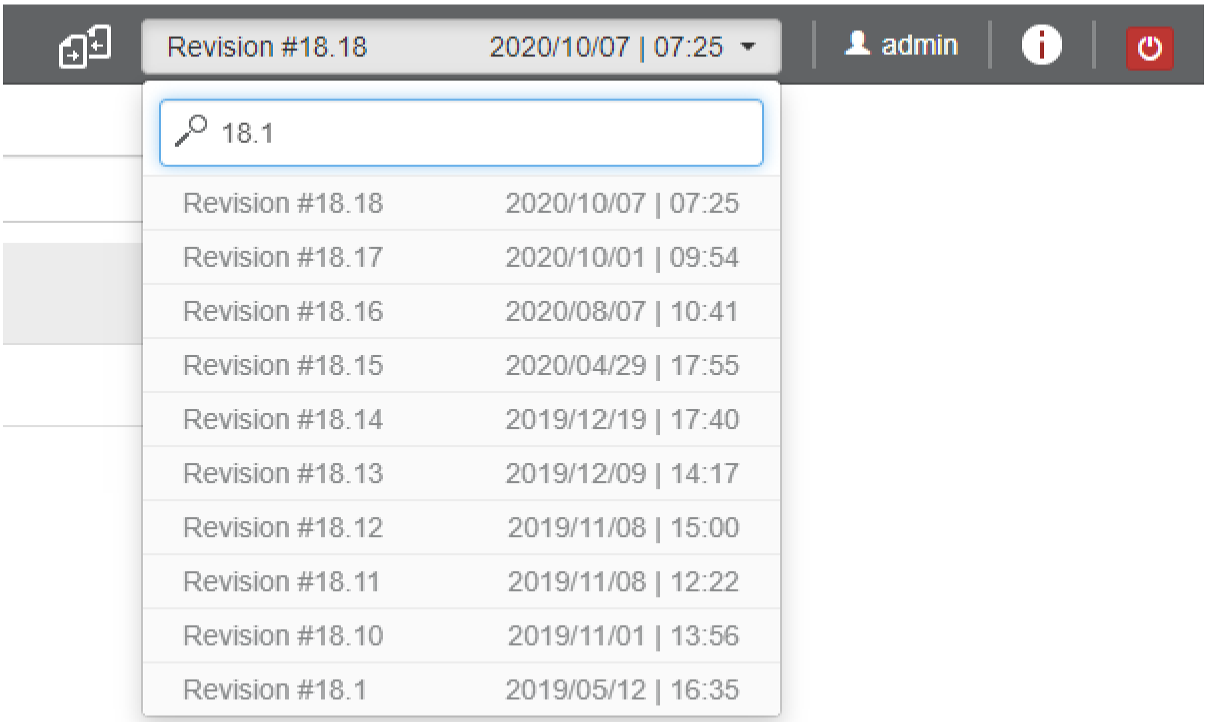
Revision #18.18 2020/10/07 | 07:25
admin
Search by number or date and time...

Revision #19	2022/01/03 | 12:08
Revision #18	2018/11/04 | 22:02
Revision #17	2018/11/04 | 21:10
Revision #16	2018/11/04 | 21:06
Revision #15	2018/11/04 | 21:04
Revision #14	2018/11/04 | 21:02
Revision #13	2018/11/04 | 21:00
Revision #12	2018/11/04 | 20:56
Revision #11	2018/11/04 | 18:54
Revision #10	2018/11/04 | 18:46
Revision #9	2018/11/04 | 18:42
Revision #8	2018/11/02 | 15:11

This is because there could be plenty of minor revisions and it would not be user-friendly to display all of them.

The only way to select a minor revision is to search for it using the text box above the revision list. If the search text box is not empty, the matching revisions will be displayed in the list. The revisions can be searched by number or date.

Search by revision number:



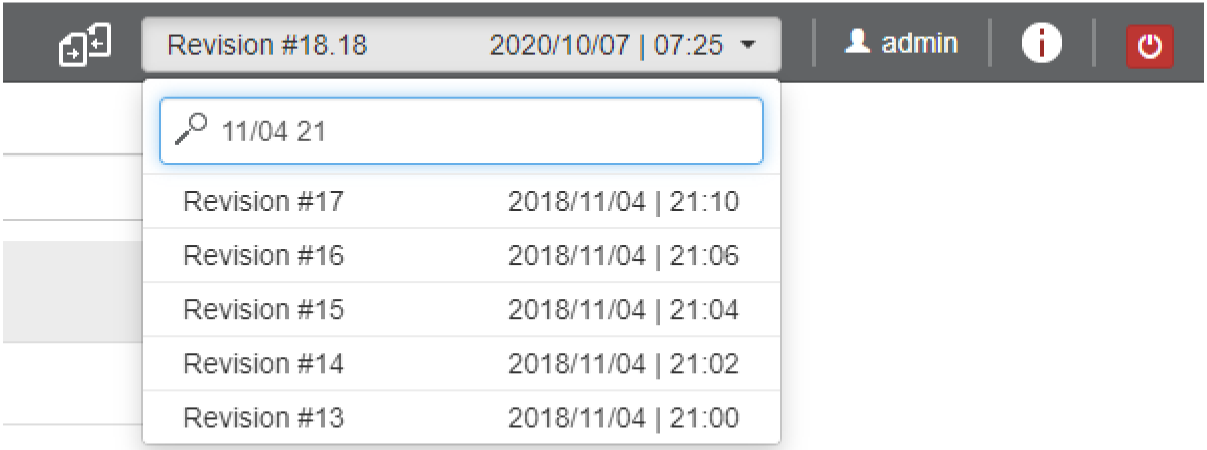
Revision #18.18 2020/10/07 | 07:25

admin

18.1

Revision #18.18	2020/10/07 | 07:25
Revision #18.17	2020/10/01 | 09:54
Revision #18.16	2020/08/07 | 10:41
Revision #18.15	2020/04/29 | 17:55
Revision #18.14	2019/12/19 | 17:40
Revision #18.13	2019/12/09 | 14:17
Revision #18.12	2019/11/08 | 15:00
Revision #18.11	2019/11/08 | 12:22
Revision #18.10	2019/11/01 | 13:56
Revision #18.1	2019/05/12 | 16:35

Search by revision date:



Revision #18.18 2020/10/07 | 07:25

admin

11/04 21

Revision #17	2018/11/04 | 21:10
Revision #16	2018/11/04 | 21:06
Revision #15	2018/11/04 | 21:04
Revision #14	2018/11/04 | 21:02
Revision #13	2018/11/04 | 21:00

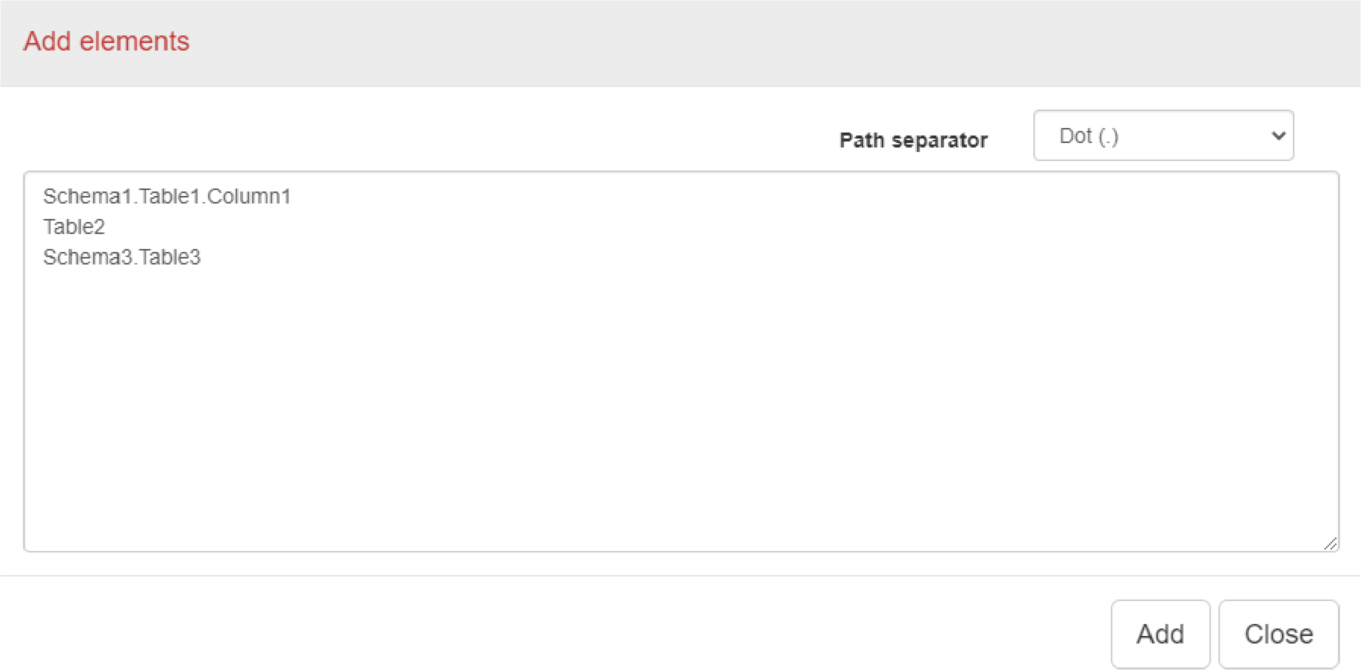
Selecting Elements for Data Flows

If you are interested in a dataflow analysis for one or more of the elements in the object catalog, you should first move them to the *Selected Elements* list. Once you have found them in the object catalog, there are two ways to get them there.

- You can select one or more elements by holding the `SHIFT` (or `CTRL`) key and then dragging them from the object catalog to the list of selected elements.
- You can double-click on one element.

You can also select one or more elements directly using the *Add Elements* dialog.

- Click on the *Direct Add* button in the title bar of the *Selected Elements* list. The following dialog box will appear.



Add elements

Path separator: Dot (.)

Schema1.Table1.Column1
Table2
Schema3.Table3

- In the *Path Separator* combo box, select a character to be a separator for element path entries.
- Insert the elements you want to select into the text area. For each element you can insert its name, a full ancestor from the top level, or a relative path from an ancestor of any level. The path entry separator is the character selected in the previous step. Each element must start on a new line. For an example, see the illustration above.
- Click on the *Add* button to confirm the selection.
  - Note: In the case of two or more layers, only elements belonging to the selected layer will be added to the selection. If any elements are filtered out because they are outside the selected layer, a warning message will be displayed.

Once you have some elements in the list, you can remove one by clicking on the trash bin image to the right of the item or remove all of them by clicking on the trash bin icon above.

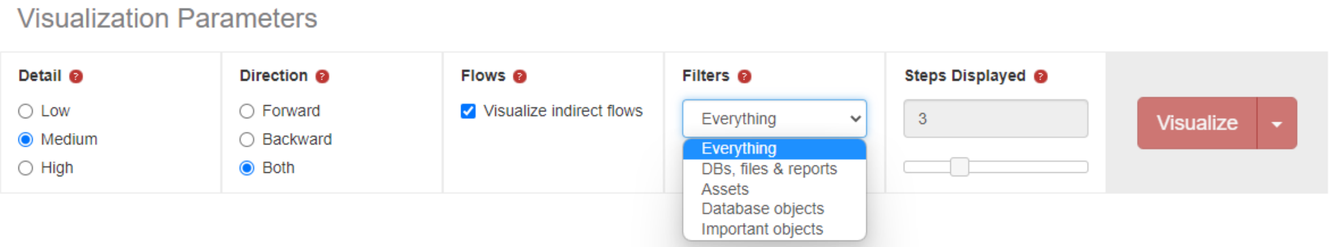
If you select an element which contains other elements, a dataflow analysis will be performed on all its descendants. For example, if you select a table, the data flow for all its columns will be analyzed.

If there are no elements in the *Selected Elements* list and you select one element in the repository, you can perform a dataflow analysis on that element without adding it to the *Selected Elements* list.

When you select a layer, all elements outside the selected layer will be removed from the selection.

Tuning Visualization Parameters

Once you have selected one or more elements for a dataflow analysis, it is possible to visualize the results by clicking on the red *Visualize* button in the lower right corner. Before doing that, you can tune several visualization parameters to allow you to get the clearest visual form for your case. Let's go over all these options. (See the following figure.)



Visualization Parameters

Detail: ○ Low ● Medium ○ High

Direction: ○ Forward ○ Backward ● Both

Flows: ☑ Visualize indirect flows

Filters: Everything

Steps Displayed: 3

Detail

The detail parameter helps you show the affected elements at the most suitable level of detail.

For example, imagine you want to see all the objects that are affected by data from a selected column. At the beginning of your analysis, you might not be interested in which particular columns or tables are affected, but you might want to see just the databases containing the affected columns and tables. In the next step, you might need to see not only the databases but also the affected tables. However, you probably still do not want to see all the affected columns. Finally, it could be necessary to show all the affected columns.

For each of these cases, a level of detail is set.

- Low detail—only the top elements are shown including databases, directories, and repositories; this level is suitable when only the big picture is necessary
- Medium detail—the middle elements such as tables, files, transformations, and statements are shown together with the top elements; this level is suitable for most analyses
- High detail—in addition to the top and middle elements, the bottom elements such as tables, files, and transformation columns and ports are shown; this level is suitable when the most precise analysis is necessary

This parameter only influences the initial arrangement of the visualization. Then you can set the level of detail for each object individually.

Direction

The direction parameter dictates whether you will see objects that are influenced by the selected elements and whether you will see objects that influence the selected elements.

For example, if it is necessary to enlarge the size of the data stored in a column, you might want to know which other columns in the environment must be enlarged to avoid losing data. On the other hand, you might want to know which source systems a business report is computed from. Finally, both pieces of information might be necessary at once.

For each of these cases, a direction option is set.

- Forward—only objects influenced by the selected elements are shown
- Backward—only objects that influence the selected elements are shown
- Both—both groups of objects are shown

Note that it is not possible to change this parameter after the visualization is shown.

Flows

The *Visualize Indirect Flows* parameter dictates whether you will also see objects that indirectly influence or are influenced by the selected elements.

Generally, there are two types of data flows.

- Direct data flow—two objects are connected with a direct data flow if and only if data from a source object is transformed into data in a target object
  - In the expression z = x + y;, z is the target object in the direct data flow from x and also the target object in the direct data flow from y
  - In the statement INSERT INTO T (C, D) SELECT A, B FROM S;, T.C is the target object in the direct data flow from S.A, and T.D is the target object in the direct data flow from S.B
- Indirect data flow—two objects are connected with an indirect data flow if and only if data from a source object influences which data will be stored in the target
  - In the expression if (p > 0) then z = x; else z = y; endif;, z is the target object in the indirect data flow from p
  - In the statement INSERT INTO T (C, D) SELECT A, B FROM S WHERE P > 0;, T.C and T.D are the target objects in the indirect data flows from S.P

There are two status options for the *Visualize Indirect Flows* parameter.

- Unchecked—you will only see the objects that are transitively connected to the selected elements by direct data flows
- Checked—you will see the objects that are transitively connected to the selected elements by direct data flows and also the objects that are connected to all these objects by indirect data flows (not transitively)

Note that it is not possible to change this parameter after the visualization is shown.

Filters

The filter parameter dictates which technologies the objects shown in the visualization will be from.

In large environments, data is stored in several types of databases or in several file systems and is transformed by several types of tools such as ETL or SQL. For many dataflow analyses, it is not necessary to see how the data is transformed but only where it is stored. Sometimes it is sufficient if the file system storage is also omitted and only the database storage is shown. On the other hand, you may need to see every detail, so you do not want to omit anything.

For each of these cases there is a filter option.

- Everything—you will see objects from all analyzed technologies
- DBs, files, and reports—you will only see database objects, files, and reports, not transformations
- Database objects—you will only see database objects, not files or transformations
- Important objects—you will only see database objects without inner tables, functions, or procedures

Each filter option is valid for a certain set of layers. Only filters valid for the selected layer are displayed. When the layer selection is changed, the filter option list will be updated.

This parameter only influences the initial arrangement of the visualization. Then more detailed filtering can be done.

Note that these filters can be configured by your application administrator, so they could differ from this documentation.

Steps Displayed

The steps displayed parameter selects the maximum distance of the initially-shown objects from the selected elements.

In large environments, the number of objects that influence or are influenced by the selected elements can be really huge. In these cases, it is not desirable to show all of them at once because it would be a really huge dataflow graph. In many cases, it is only necessary to analyze the nearest dataflow neighbors in detail or one or two dataflow paths from the source to the target systems. Therefore, the steps displayed parameter has to be set to prevent huge numbers of objects from being shown at once.

Do not worry that you won't be able to see anything when looking at the big picture (low detail is selected or only database objects should be shown) because the distance is counted between the objects that you will see. For example, let’s imagine that you have a table in a database that is exported to a file, and then this file is transformed by many transformations in an ETL tool into another file, which is finally loaded onto another table in another database. If you select the source table, set the filter to see only database objects, and set the steps displayed parameter to one, you will see the source and target tables connected.

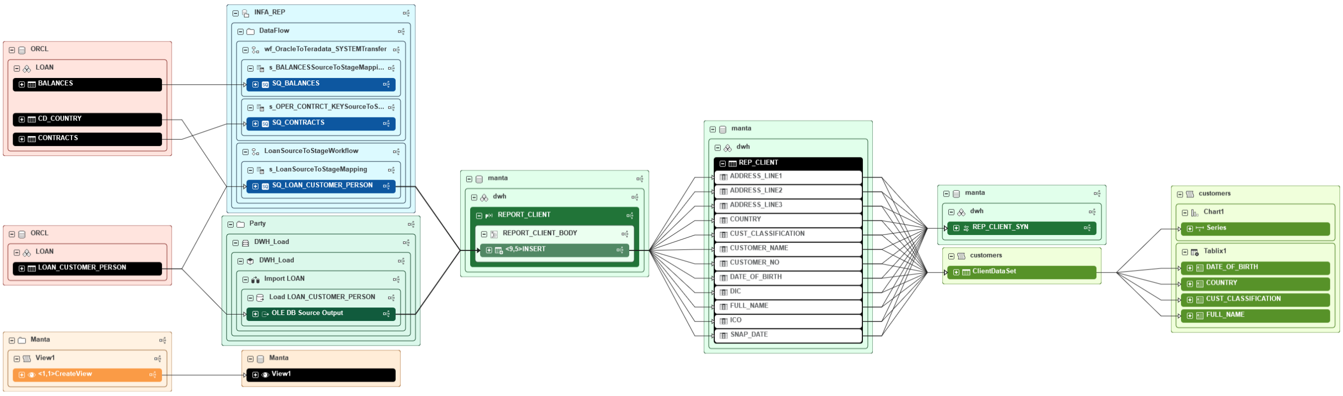
Moreover, remember that this is definitely only the initial number of steps because you can then discover further data flows in the visualized dataflow graph.

Dataflow Analysis

Once you have selected the elements, tuned the visualization parameters, and clicked on the *Visualize* button, or just clicked on a link in your documentation, the initial dataflow graph is shown. In this section, our visual representation of a data flow will be introduced together with all the actions that can be performed on it such as browsing, filtering, and exporting.

Introducing Data Flows

If you select your favorite table or file that is filled from a source system and then transformed into a target system and set a high level of detail in both directions, a visualization of indirect flows, no filter, and a steps displayed parameter of three, you can get a dataflow graph similar to the one below.



You can see several elements, some of which are connected by arrows, and make the following observations.

- Some elements contain other elements, so they create hierarchies similar to those in the object catalog.
  - For example, you can see a database that contains a table containing several columns.
  - In some cases, not all elements in a hierarchy are shown to preserve clarity such as none but the last directory in a path to a file[1].
  - Levels of hierarchy are also distinguished by color—leaves are white, their parents are saturated colors, and other elements are pastel colors[1].
- Some element hierarchies have the same color; others have different colors.
  - Colors distinguish between the various technologies presented in the environment.
  - For example, files are light blue while Teradata tables and views are ochre[1].
  - In some cases, one element hierarchy can have more colors, since objects from one technology logically contain objects from other technologies; for example, a TPT script contains a Teradata SQL script or an Oracle procedure contains its DDL body in its database schema.
- All elements have a label and an icon.
  - The label corresponds to the element’s simple name; imaginary element labels often contain their location in their parent script or their order in their statement.
  - The icon corresponds to the element type such as a database, table, column, file, script, workflow, etc.
- One or more elements are black[1].

  - Black elements are the elements selected from the object catalog screen; all the elements shown either influence or are influenced by these elements.
- Some elements have a plus or minus sign.
  - An element that contains other elements can be collapsed or expanded so you can see more or fewer details for each element.
- Some elements are connected by arrows—solid black ones or dashed blue ones[1].
  - A solid black arrow represents a direct data flow from a source element to a target element.
  - A dashed blue arrow represents an indirect data flow from a source element to a target element.
  - Arrows connect only the most nested elements.
- In some cases, wider variants of both solid black and dashed blue arrows connect elements.
  - These arrows signify that there are one or more paths between the source and target elements of (in)direct data flows, but the elements on these paths are not visible at the moment. It can happen when an element's technology is filtered (filtered paths) or the element is not within the maximum distance from the selected elements (aggregated paths).
  - It is possible to see what elements are between the source and target elements by unfiltering them.
- Some elements have a flow icon on the right.
  - This icon indicates that the data flow from the selected elements does not end in this element but continues to other elements that are not visible yet.
  - This happens when the steps displayed parameter is set lower than the maximum distance between the selected elements and some affected elements.
  - This icon is also present in elements that have both incoming and outgoing arrows but there are other arrows that have not been discovered yet.
  - You can discover all the data flows leading from or to this element by clicking on this icon.

Browsing Data Flows

Once you know what all the objects on a dataflow graph represent, you can try interacting with them.

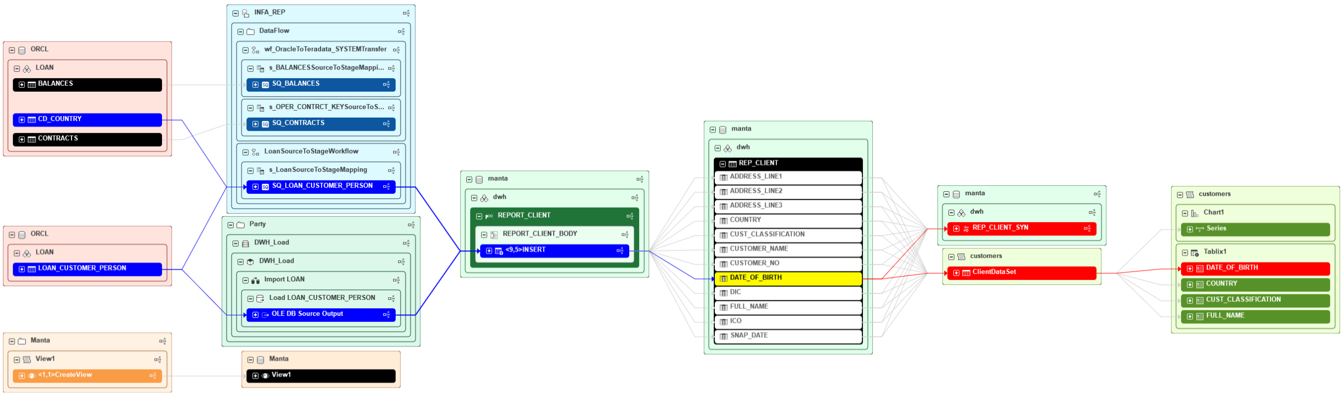
Moving and Zooming

When a dataflow graph is shown, it is possible that you will see only a few full elements and flows and that the others will be only partially visible. Now, you can do two things:

- Move a view to the other elements you want to see—hold the left mouse button somewhere outside the elements and arrows on the white background and move the mouse from one side to the other. When you are happy with your position or your mouse goes off the screen, just release the button.
- Zoom out to see more elements at once—you can either click on the magnifier icon with a minus sign in the lower left corner or you can roll the mouse wheel down. Similarly, you can either click on the magnifier icon with a plus sign in the lower left corner or you can roll the mouse wheel up to zoom in.

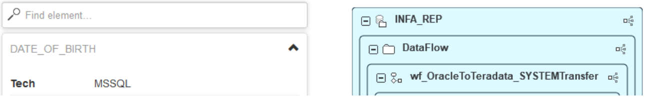
Selecting Elements and Flows

When you need to get more information about an element from a dataflow graph or when you need to see how an element is connected to others by direct data flows, click on it and you will see a change like in the image below.

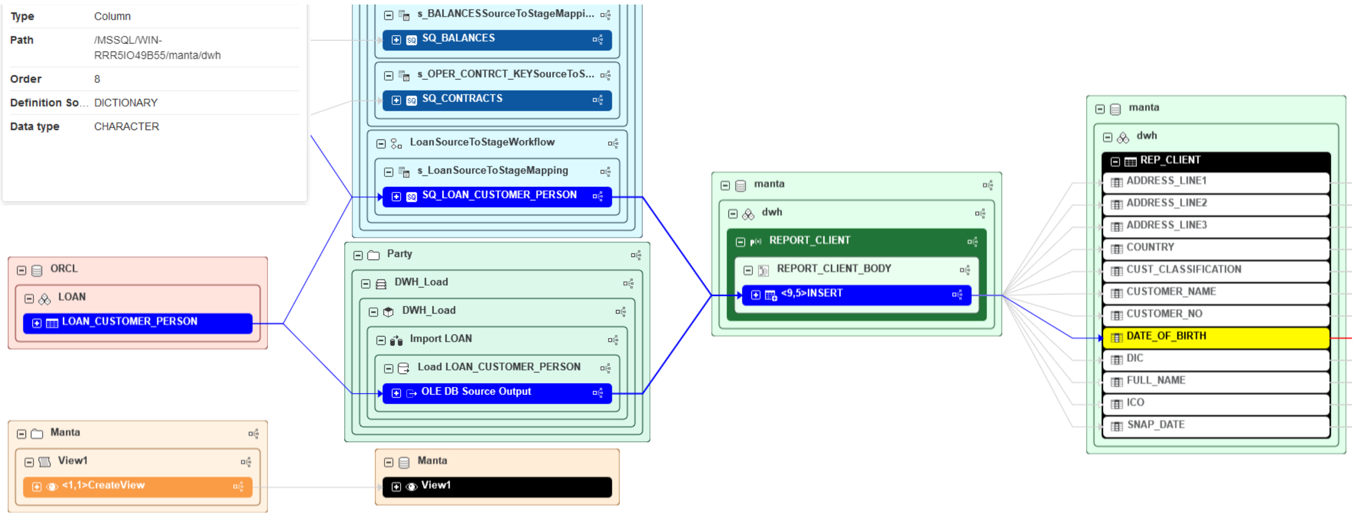


First, you can see that the selected element turns yellow. Then, all the elements connected to this element by direct data flows (even transitively) change color together with all the affected arrows. The other arrows are now lighter so the colored ones stand out. This feature is really helpful when you select a whole table for dataflow analysis and then want to see column data flows separately. Blue represents data flow to the selected element, red represents data flow from the selected element, and violet represents data flow both to and from the selected element.

Moreover, the name of the element appears in the upper left corner under the search bar. You can click on the down arrow there to see other element attributes, as you can see in the image below. You can hide these attributes by clicking on the same icon, or you can select another element to see its attributes.

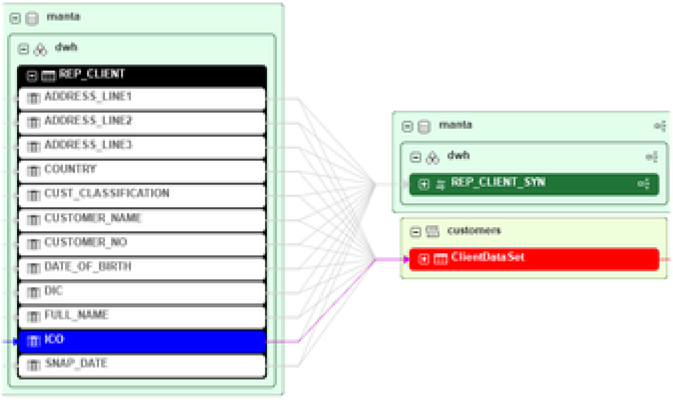


Tech MSSQL



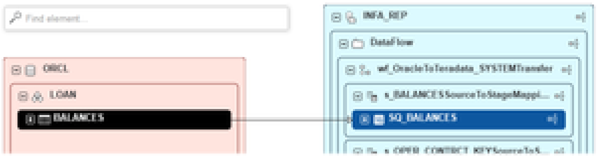
Type	Column
Path	/MSSQL/WIN-RRR5IO49B55/manta/dwh
Order	8
Definition So...	DICTIONARY
Data type	CHARACTER

When you move the mouse to an arrow, it turns red to make it easily distinguishable from the others. When you click on it, all the elements and arrows on the dataflow paths going through this arrow turn red like when you select an element. You can see the selected arrow in the image below.



To deselect elements and/or arrows, just click on the white background.

Finding Elements

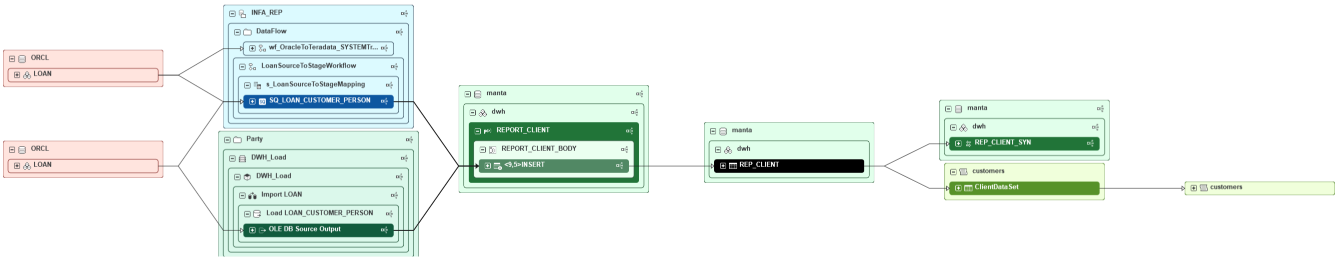


Find element...
ORCL
LOAN
BALANCES
INFA_REP
DataFlow
wf_OracleToTeradata_SYSTEMTransfer
s_BALANCESSourceToStageMappi...
SQ_BALANCES
s_OPER_CONTRCT_KEYSourceToS...

Sometimes, especially on larger dataflow graphs, you might want to find a visible element. It is easy. Just type its name into the search bar in the upper left corner. A list of possible elements will appear from which you can choose the desired element by name, full path, and type. Click on the desired element so that it is centered on your screen and the element is highlighted. If it is not what you were searching for, you can click on another element. When you are satisfied, just click on the white background to cancel the highlighting and hide the list.

Expanding and Collapsing Elements

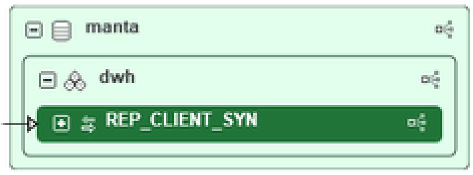
In this example, you can see all elements at their highest level of detail. However, this might not be necessary for some less interesting elements. In such a case, you can collapse an element containing other elements by clicking on its minus sign. The elements it contains will no longer be visible, as you can see in the following illustration.



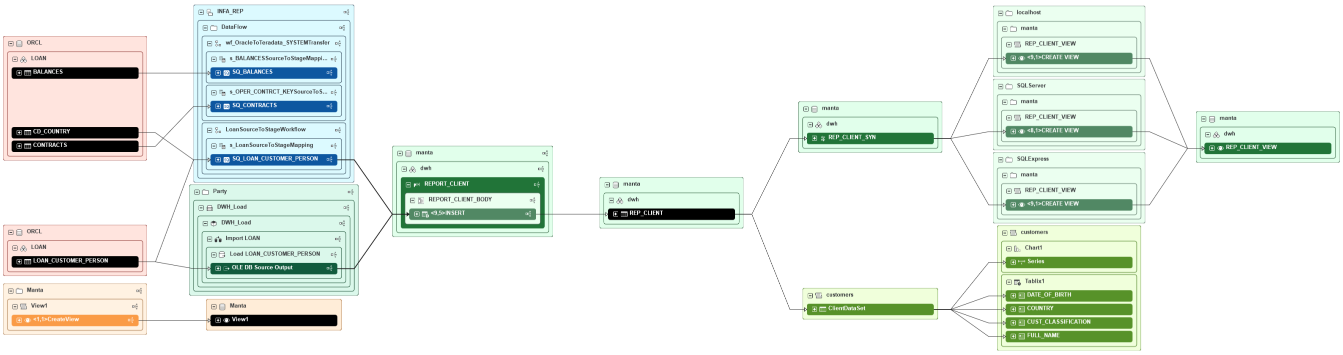
On the other hand, if you start your initial data flow at a medium level of detail, all the elements will look like those collapsed in the previous figure. In this case, you can expand the elements to see their children by clicking their plus sign.

Discover Further Data Flow

When the steps displayed parameter is set lower than the maximum distance between the selected elements and some influenced elements, some elements will have flow icons on the right as you can see in the image below.



That means that there are other elements that are transitively connected by data flow to the elements selected through the element with the icon. To see more elements on the path from the selected elements to the other connected elements through the element with the icon, just click on the icon. The icon will disappear and the new elements will appear like in the following illustration.



It will look the same as if you had set the steps displayed parameter to four for only this particular path, thus the other paths with other elements and flows that are not important to your analysis are not shown. By clicking on these icons, you can follow only those paths that are important.

When no icon is present in an element, you can be sure that it is only connected to elements that are already visible. (We certainly only consider those connections that start at the elements which were originally selected.)

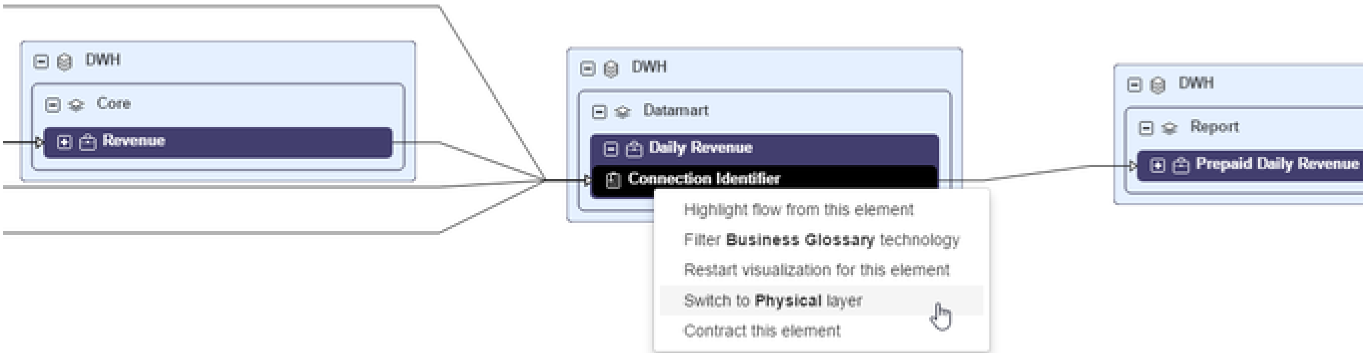
Restarting Data Flows

If you find an element that you want to do a dataflow analysis on, just right-click it and select *Restart Visualization* from the element options in the context menu. You will get an initial dataflow graph for this element with the same parameters as for the previous one.

Switching Data Flows to Another Layer

In the case of two or more layers, just one layer is active at a time and only the elements and flows belonging to the active layer are displayed. Typically, the active layer is selected on the object catalog page.

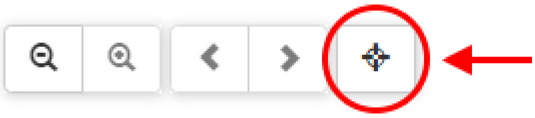
If an element maps or is mapped by another element in a different layer, it is possible to switch to that layer. Right-click the element and select *S witch to ... Layer* from the context menu, as shown in the image below.



After that, the visualization will restart from the mapping or mapped element in the selected layer.

Center the Camera on the Start Node

It is possible to center the camera on the start node by simply clicking on the button in the lower-left corner of the visualization.



Filtering Data Flows

MANTA Flow has two types of filters.

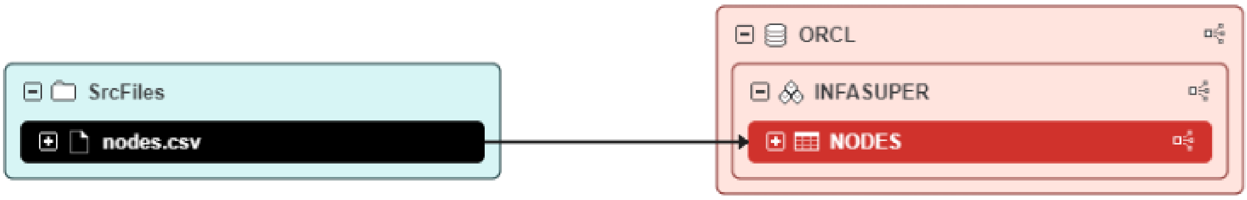
- Resource filters—auto-generated for every currently-existing resource
- Custom filters—configured by administrators; the following filter unit types are supported:
  - By resource—hides elements of the listed resources
  - By name—hides elements with names matching the defined regular expression
  - By type—hides elements of the listed types
  - By attribute—hides elements with the given attribute having any of the listed values

Both of them can be bound to the particular layer that the filter is valid for. If not, the filter is valid for all layers.

- Resource filters—the layer binding is determined by the layer the resource belongs to
- Custom filters—the layer binding is configured by the administrators

Both of them can be grouped into filter groups, and each group may contain filters of both types. The groups are configured by the administrators. The layer validity of the filter group is determined by the layer validity of the contained filters. The filter group is valid for a layer if and only if at least one filter in that group is valid for that layer.

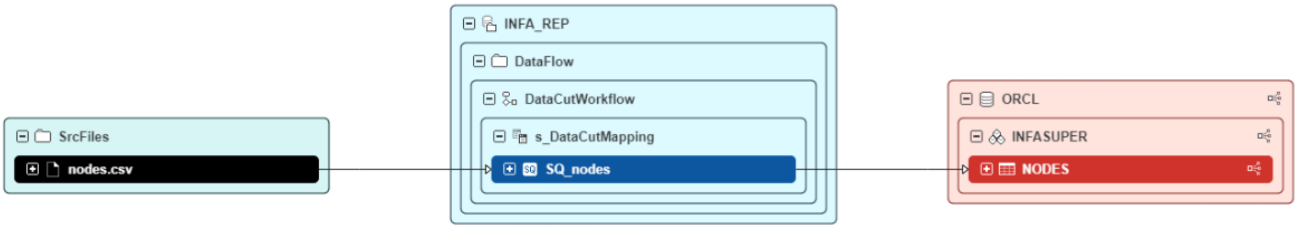
You can select a filter group as a visualization parameter on the catalog screen by choosing the appropriate item from the *Filters* options. Only groups valid for the selected layer are displayed. If you select the `/Oracle/orcl/infasuper` schema and choose medium detail, both directions, no visualization of indirect flows, the *DBs, Files & Reports* filter, and a steps displayed parameter of one, you can get a dataflow graph similar to the one below.



Now you can see the direct data flow between the source elements and the selected elements. There are no elements from transformation technologies. The transformation elements have been replaced by wider arrows.

Note that the distance is counted without filtered elements. That is why the displayed source elements fulfill the maximal depth one condition, although some other elements exist between the source element and the one that has been selected.

There are several ways to modify the filtering on the visualization screen. You can display the filtered elements behind a particular arrow by right-clicking on that arrow and selecting the *Unfilter* option. The arrow will be replaced by the hidden elements.



Other filtering options are available in the *Options* box in the upper right corner. Click on it to unpack it.

- Under the *Resources* tab, you can check the resources that you want to display and uncheck the ones that you want to be filtered out. Then confirm your selection using the *Apply* button.
-

Under the *Filters* tab, you can check the custom filters that you want to apply and uncheck the ones that you want to turn off. Then confirm your selection using the *Apply* button.

In both cases, only filters valid for the active layer are displayed.

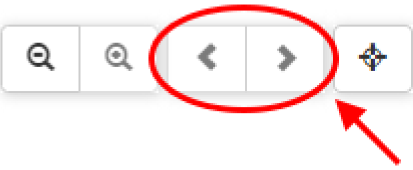
To filter by the technology (resource) of a particular element, right-click on that element and select *Filter ... Technology*.

To filter only one particular element (and all its descendants) manually, right-click on that element and select *Contract This Element*.

To cancel manual filtering and unfiltering (see the *Unfilter* command mentioned earlier), unpack the *Options* box in the upper right, go to the *Filte rs* tab, and push the *Clean Manual Visibility* button.

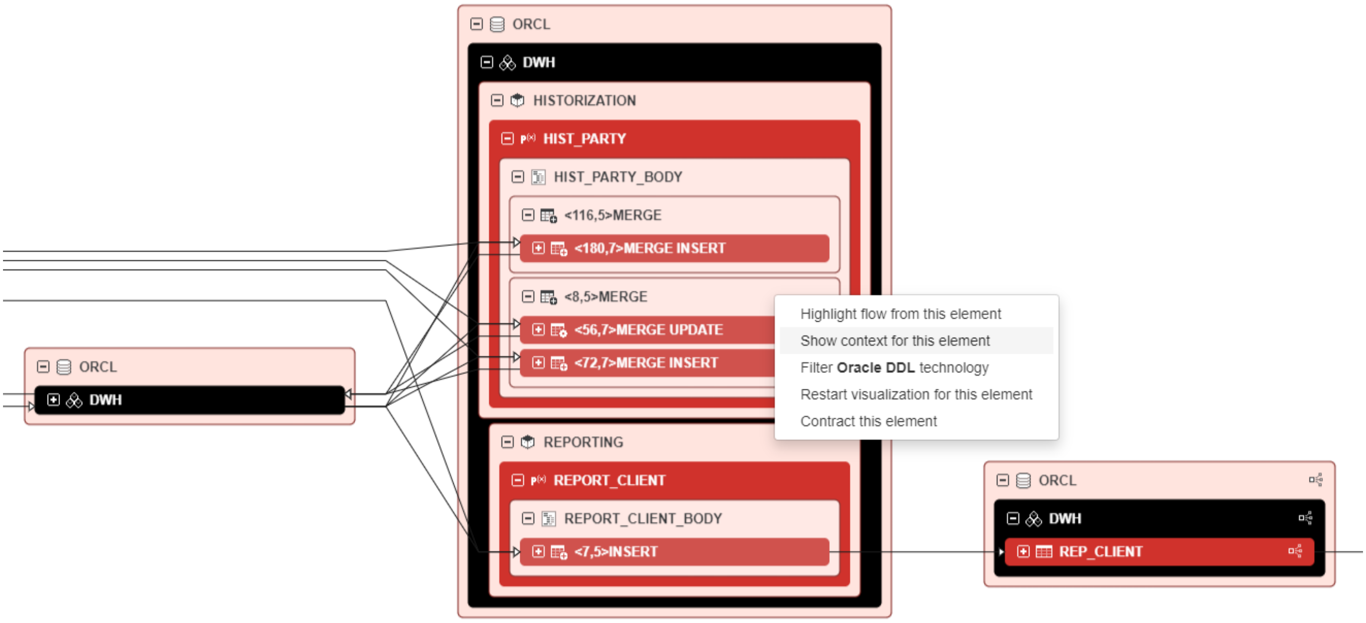
Undo Your Last Action

Each action that somehow changes the visualized data flow (such as expand/discover/filter) is recorded and can be reversed. In the default settings, the application stores 10 recorded states that can be browsed in both directions (undo/redo). This can be done using the buttons with the left and right arrows located in the lower-left corner (see the following illustration) or by using the well-known keyboard shortcuts `Ctrl+Z` for *Undo* and `Ctrl+Y` for *Redo*. Beware that when you go back a few actions using *Undo* and then change the data flow using *Expand*, all *Redo* states will be forgotten, as is standard undo/redo behavior in most applications.

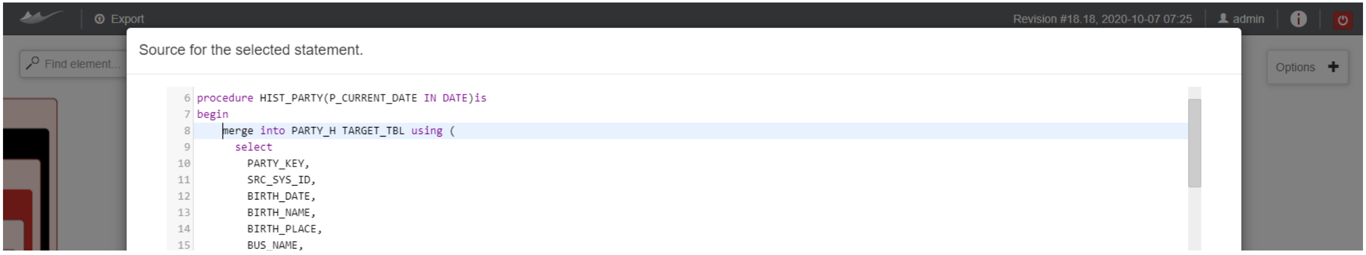


Viewing the Source Code

While browsing data flows, it can be interesting to see the original scripts that are represented by the flow. You can do this by choosing a node that represents a procedure or command inside a transformation element and selecting *Show Context for This Element*, as shown in the following illustration. You can also double-click on the transformation element.

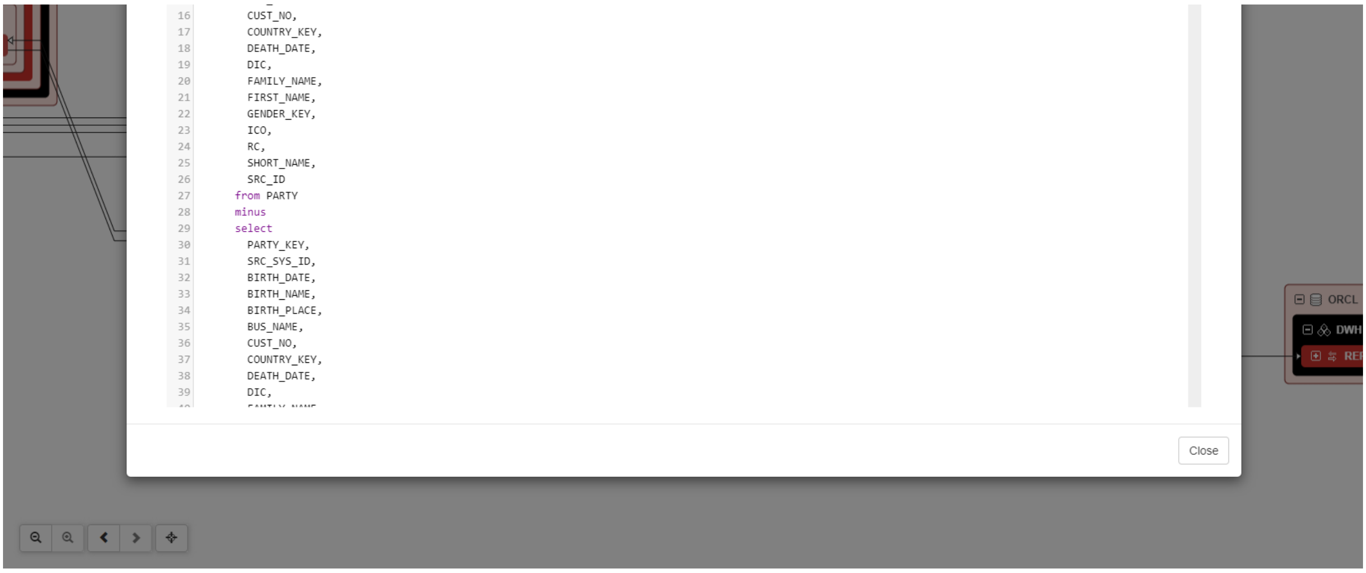


This will open a pop-up window with the entire formatted script, and the statement represented by the clicked element will be highlighted as shown in the following illustration.



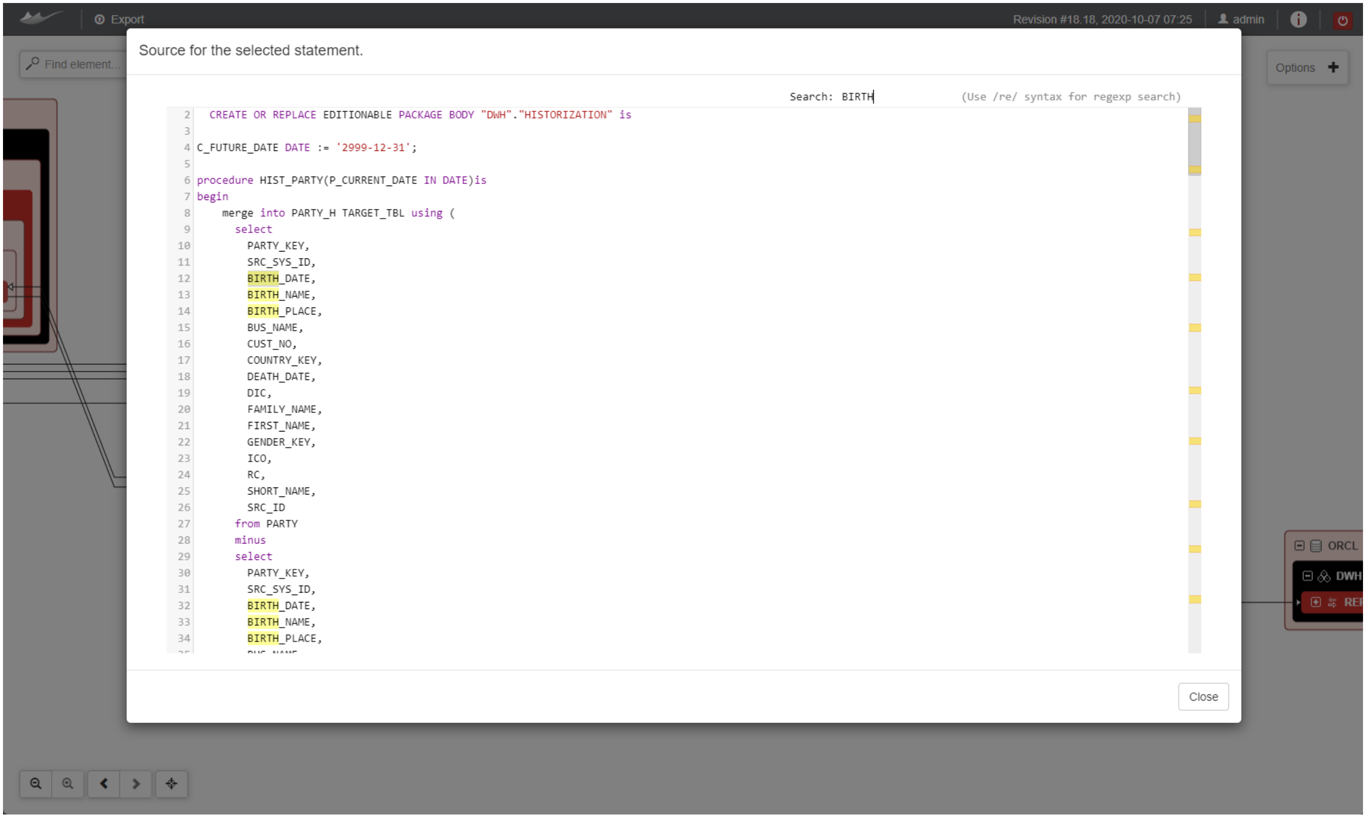
Source for the selected statement.

```
procedure HIST_PARTY(P_CURRENT_DATE IN DATE)is
begin
    merge into PARTY_H TARGET_TBL using (
      select
        PARTY_KEY,
        SRC_SYS_ID,
        BIRTH_DATE,
        BIRTH_NAME,
        BIRTH_PLACE,
        BUS_NAME,
```



```
        CUST_NO,
        COUNTRY_KEY,
        DEATH_DATE,
        DIC,
        FAMILY_NAME,
        FIRST_NAME,
        GENDER_KEY,
        ICO,
        RC,
        SHORT_NAME,
        SRC_ID
      from PARTY
      minus
      select
        PARTY_KEY,
        SRC_SYS_ID,
        BIRTH_DATE,
        BIRTH_NAME,
        BIRTH_PLACE,
        BUS_NAME,
        CUST_NO,
        COUNTRY_KEY,
        DEATH_DATE,
        DIC,
```

To do a full-text search, use the shortcut `Ctrl+F`. Type the string you are searching for in the *Search* field and press *Enter*. The first occurrence of the string after the cursor will be focused and all occurrences will be highlighted. Also, the appropriate parts of the vertical scrollbar will be highlighted, as shown in the following illustration.



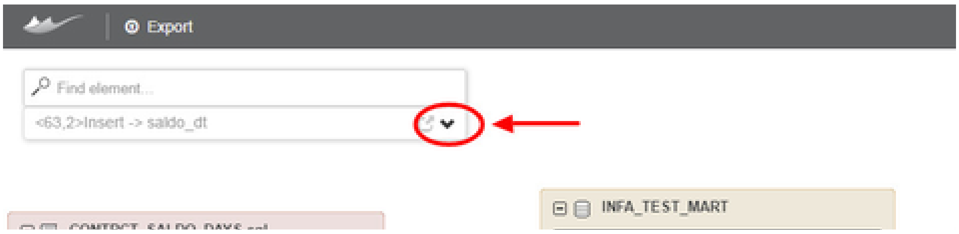
Source for the selected statement.

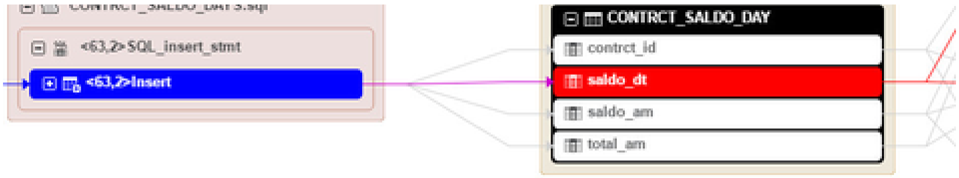
Search: BIRTH (Use /re/ syntax for regexp search)

```
  CREATE OR REPLACE EDITIONABLE PACKAGE BODY "DWH"."HISTORIZATION" is

C_FUTURE_DATE DATE := '2999-12-31';

procedure HIST_PARTY(P_CURRENT_DATE IN DATE)is
begin
    merge into PARTY_H TARGET_TBL using (
      select
        PARTY_KEY,
        SRC_SYS_ID,
        BIRTH_DATE,
        BIRTH_NAME,
        BIRTH_PLACE,
        BUS_NAME,
        CUST_NO,
        COUNTRY_KEY,
        DEATH_DATE,
        DIC,
        FAMILY_NAME,
        FIRST_NAME,
        GENDER_KEY,
        ICO,
        RC,
        SHORT_NAME,
        SRC_ID
      from PARTY
      minus
      select
        PARTY_KEY,
        SRC_SYS_ID,
        BIRTH_DATE,
        BIRTH_NAME,
        BIRTH_PLACE,
```

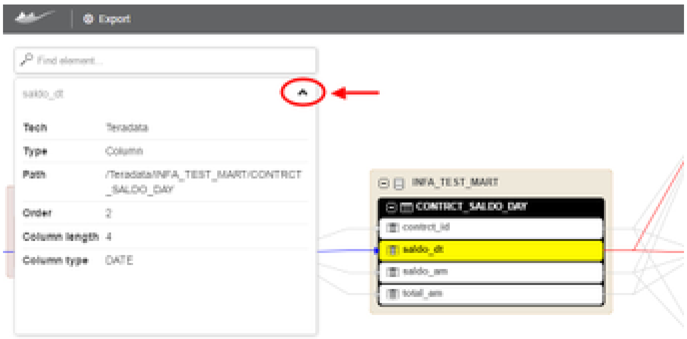
Details of Nodes and Edges

You can view the details of nodes and edges by simply selecting them. When a node or an edge is selected, it is possible to expand its detail by clicking on the expand arrow located next to the element name in the upper left corner of the screen just under the search box, as shown in the illustration below.





The expanded dialog box contains detailed element information including its attributes. Strings that are too long have a link next to them that opens an additional dialog box showing the whole value. It is also possible to collapse the dialog box, returning it to its previous state. An example is shown in the following illustration.



saldo_dt

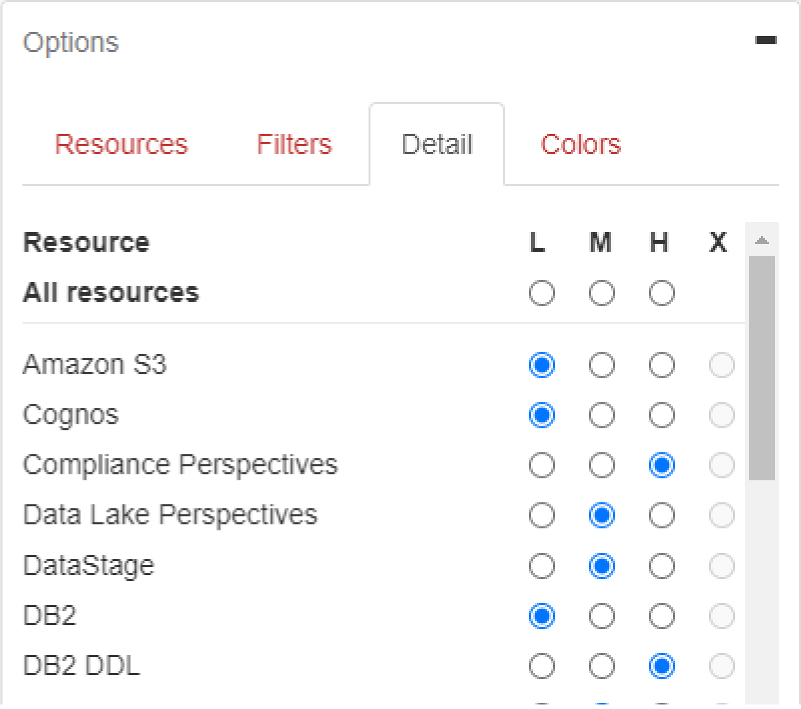
Tech	Teradata
Type	Column
Path	/Teradata/INFA_TEST_MART/CONTRCT_SALDO_DAY
Order	2
Column length	4
Column type	DATE

You can find two special attributes in the edge detail.

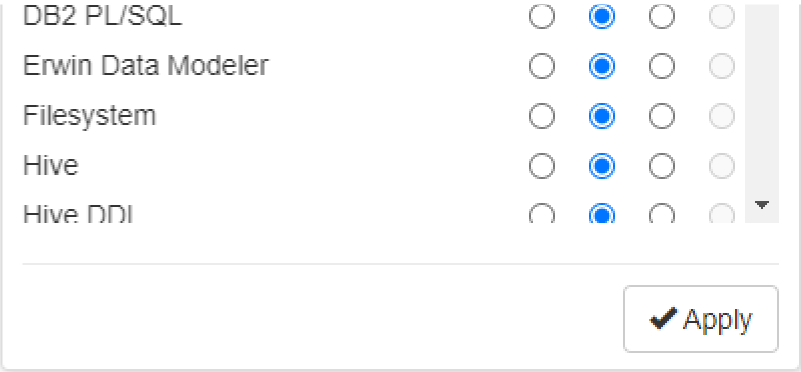
- **Selected edges**—the selected edge is between non-leaf elements such as tables; the value represents the number of represented edges between the leaves
- **Filtered nodes**—the selected edge is aggregated through several filtered nodes, for example, when you filter script resources and select an edge between two tables; the value represents the number of these filtered nodes

Bulk Settings for Level of Detail

It is possible to change the level of detail for all elements of a specific resource. This is done under the *Detail* tab in the *Options* dialog box as shown below.



Resource	L	M	H	X
All resources	○	○	○	
Amazon S3	●	○	○	○
Cognos	●	○	○	○
Compliance Perspectives	○	○	●	○
Data Lake Perspectives	○	●	○	○
DataStage	○	●	○	○
DB2	●	○	○	○
DB2 DDL	○	○	●	○



DB2 PL/SQL	○	●	○	○
Erwin Data Modeler	○	●	○	○
Filesystem	○	●	○	○
Hive	○	●	○	○
Hive DDL	○	●	○	○

The GUI settings are listed in a table. The resource list is on the left side and the level of detail is on the right side. In the case of two or more layers, only resources belonging to the active layer are visible.

There are four options for each resource.

- L—Low detail—shows only databases, folders, directories, etc.
- M—Medium detail—also shows tables, procedures, transformations, scripts, etc.
- H—High detail—shows all elements including columns, ports, and attributes
- X—Custom detail—preserves the element settings which have been manually selected by the user

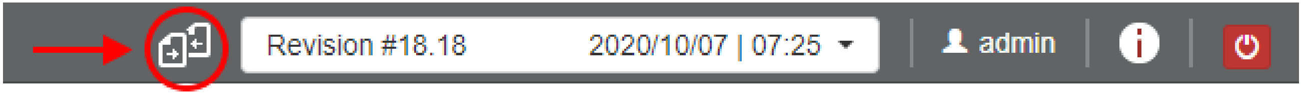
Confirm the changes by clicking on *Apply*.

Graphically Comparing Two Revisions

Start the Revision Comparison

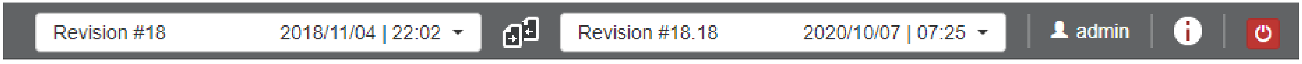
You can use MANTA Visualization to view the differences between two selected revisions by doing the following.

1. Select the main revision.
2. Click on the *Compare Revisions* button.



Revision #18.18 2020/10/07 | 07:25

3. Select the older revision that it should be compared to.



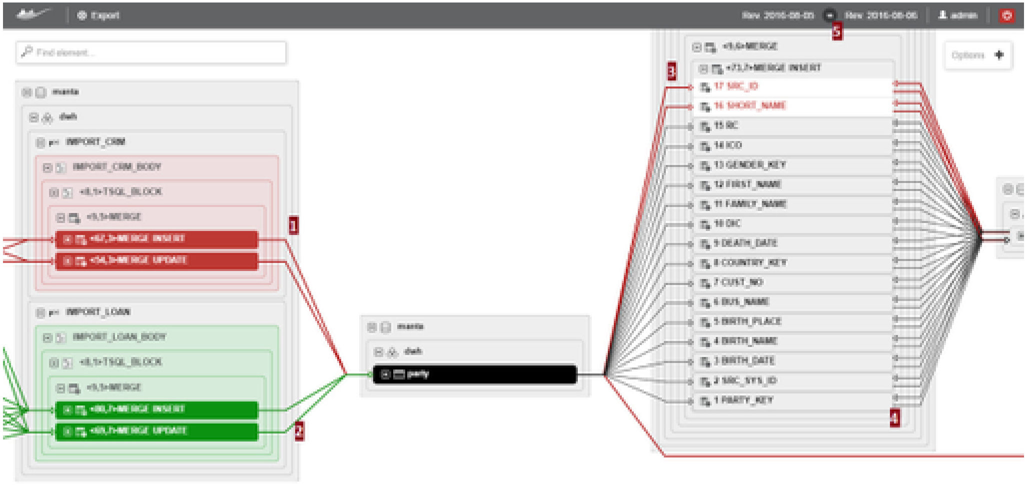
Revision #18 2018/11/04 | 22:02

Revision #18.18 2020/10/07 | 07:25

4. Select the starting element and visualization parameters as you would for a normal visualization.
5. Display the visualization by clicking on the *Visualize* button.

Comparison Layout

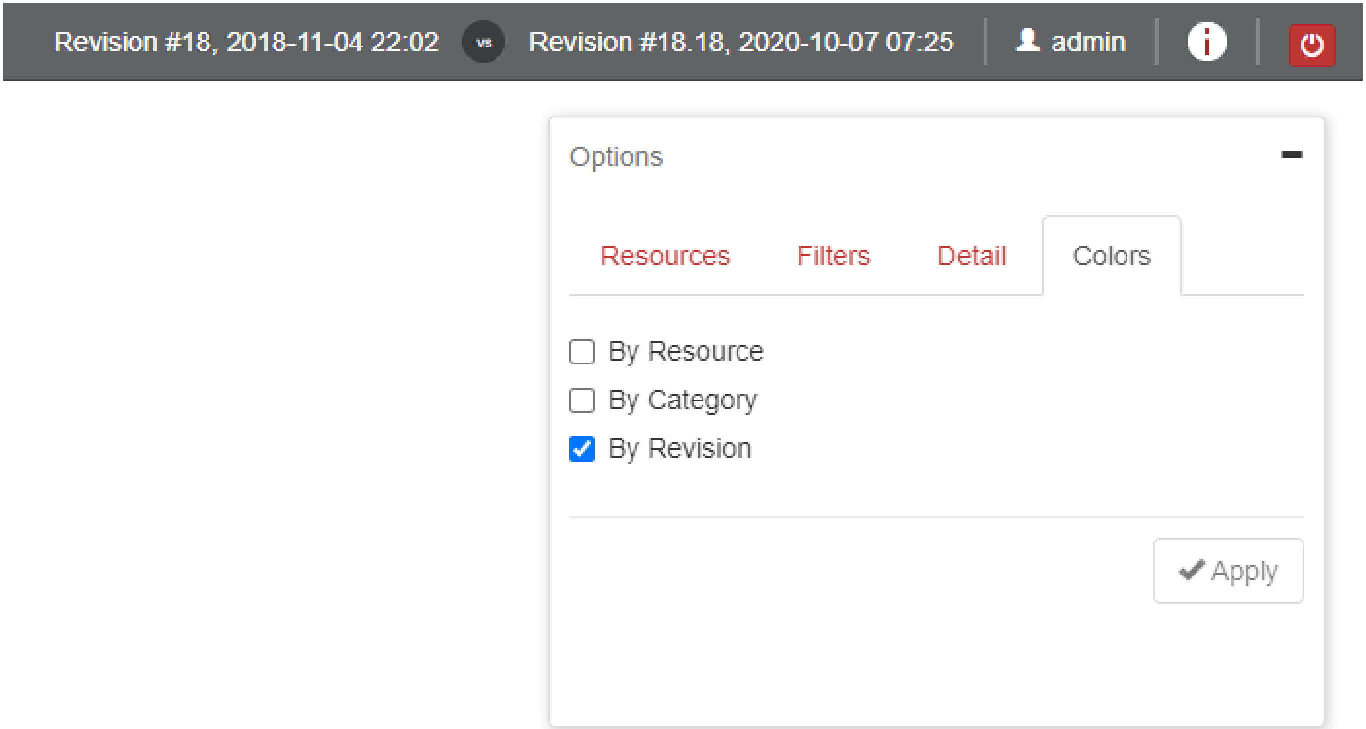
The layout for revision comparison is based on the normal visualization but with several special effects. The highlighting is similar to the standard highlighting style for showing differences. This style is used by diff tools for version control system applications like GIT or SVN. That means that new objects will be green and the old ones will be red.



Detailed description:

1. The merge statement in the procedure `IMPORT_CRM` was only in the older revision.
2. The merge statement in the procedure `IMPORT_LOAN` was only in the newer revision.
3. The merge insert was updated in the new revision by removing two columns—`SRC_ID` and `SHORT_NAME`.
4. The rest of the columns are the same in both revisions.
5. The header shows the dates of the revisions being compared. You can see the detail by hovering over them with the cursor.

It is possible to change the comparison color layout to the normal visualization in the color layer settings.



Revision #18, 2018-11-04 22:02 vs Revision #18.18, 2020-10-07 07:25

Options

☐ By Resource
☐ By Category
☑ By Revision

Exporting Data Flows

When you find the information you are looking for, there are three different ways you can export it using the *Export* menu in the upper left corner.

- Export all visible elements as a PNG image at 1:1 zoom. This export is ideal as an attachment to a change requirement specification document.
- Export the initial dataflow graph as a permalink, but note that only the initial dataflow graph is linked, not any changes that you have made by expanding/collapsing, following, or filtering. This export is ideal as environment documentation because it is always up-to-date.
- Export all elements that influence or are influenced by the selected elements to a ZIP archive containing two CSV files (`relations.csv` and `vertices.csv`), but note that all the elements affected, not only the visible ones, are exported. This export is ideal for further processing. Note that it is also possible to perform this export directly from the repository screen without dataflow visualization by clicking on the arrow on the *Visualize* button and selecting the *Export* option.

Structure of Exported CSV Files

The first file, `relations.csv`, describes all data flows (one row for each data flow). Example:

Let's imagine that we have a direct data transfer from column `t1c2` of table1 to column `t2c1` of table2. Both tables belong to a database called "db" under the database system Teradata.

Field	Example value	Description
TYPE	DIRECT	Either DIRECT or FILTER, depending on the type of data flow
SourcePath	Teradata.db.table1.t1c2	The full address of the source
TargetPath	Teradata.db.table2.t2c1	The full address of the target
SourceColumnName	t1c2	The name of the source column

SourceColumnType	Column	
TargetColumnName	t2c1	The name of the target column
TargetColumnType	Column	
SourceObjectName	table1	The name of the source object (table)
SourceObjectType	Table	The object type is usually TABLE or VIEW
TargetObjectName	table2	The name of the target object (table)
TargetObjectType	Table	The object type is usually TABLE or VIEW
SourceGroupName	db	The name of the source group (database)
SourceGroupType	Database	The group type is usually *Database*
TargetGroupName	db	The name of the target group (database)
TargetGroupType	Database	The group type is usually *Database*
SourceResourceName	Teradata	Resource name of the source (Teradata, Oracle, etc.)
SourceResourceType	Teradata	
TargetResourceName	Teradata	Resource name of the target (Teradata, Oracle, etc.)
TargetResourceType	Teradata	
RevisionState	STABLE	State of the object between two compared revisions (NEW, DELETED, STABLE, INNER)

The second file, `vertices.csv`, contains information about all the elements in the reference view (i.e., databases, tables, columns, etc.). If the element has any attributes, the file contains one row for each of its attributes (and they differ in their `AttributeName` and `AttributeValue`). If the element doesn’t have any attributes, the file contains one row about this element (in this case the `AttributeName` and `AttributeValue` remain empty).

Let’s imagine that the reference view contains the column `Teradata.db.table1.t1c1`. Then the file contains records about this column, about the table `Teradata.db.table1`, about the database `Teradata.db`, and about the resource `Teradata`.

Field	Example—Column	Example—Table	Description
FullNodeName	Teradata.db.table1.t1c1	Teradata.db.table1	The full address of the node
ColumnName	t1c1	-	The name of the column
ColumnType	Column	-	
ObjectName	table1	table1	The name of the object
ObjectType	Table	Table	The object type is usually TABLE or VIEW
GroupName	db	db	The name of the group (database)
GroupType	Database	Database	The group type is usually *Database*
ResourceName	Teradata	Teradata	Teradata, Oracle, etc.
ResourceType	Teradata	Teradata	Teradata, Oracle, etc.
RevisionState	STABLE	STABLE	State of the object when comparing two revisions (NEW, DELETED, STABLE, INNER)
AttributeName	datatype	-	The name of one of the attributes
AttributeValue	integer	-	The value of this attribute
NodeName	t1c1	table1	The name of the node (as of MANTA 34)
NodeType	Column	Table	The type of the node (as of MANTA 34)

The order of elements in the exported files is not exactly defined. For example, it is not guaranteed that the *CreateView* element will precede the *Delete* element (delete a record from the view) and vice versa.

The columns *NodeName* and *NodeType* have following reason: Not all of the objects in the file *vertices.csv* in previous versions (up to MANTA 34 exclusive) had an object type. This made it difficult for a user to sort through the list of exported nodes from a MANTA visualization. Using this two new columns it is easier to sort exported nodes by type. Therefore these two columns are populated all the time in each row.

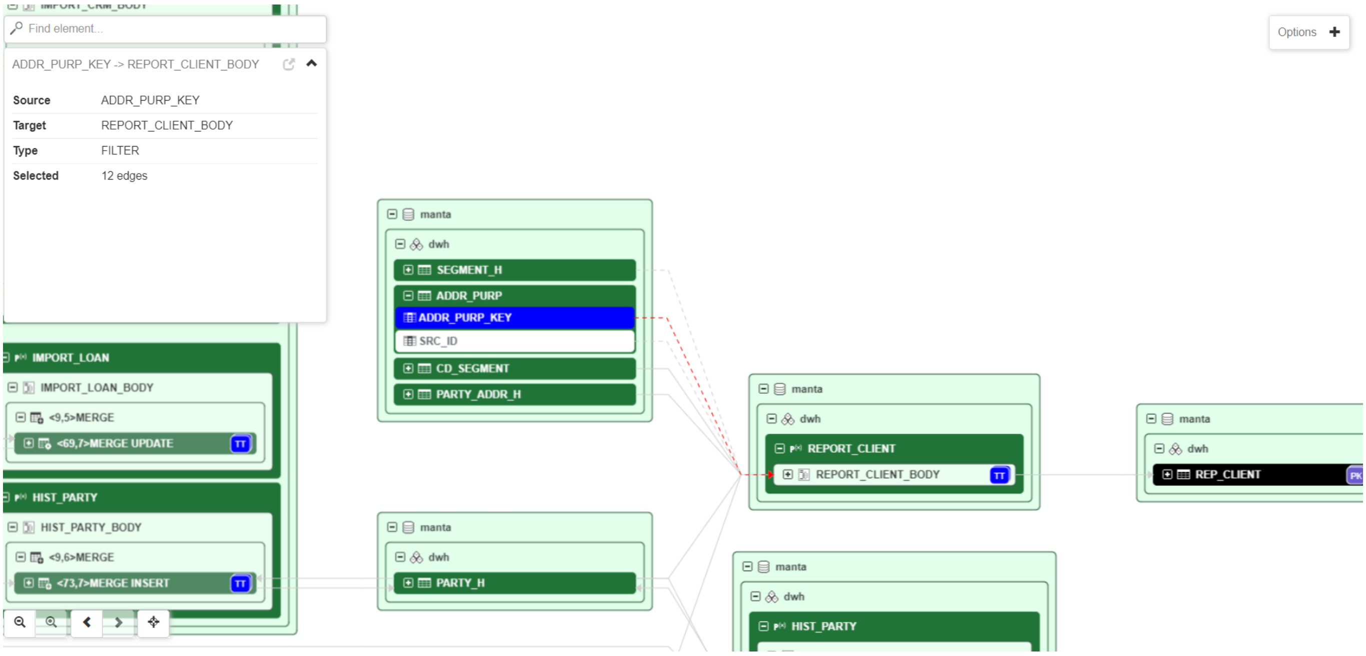
[1] This is a default configuration that can be changed by your application administrator.

Indirect Flow Edge Categorization

Indirect flow edges contain the attribute `EDGE_FILTER_TYPE` (as of R36) describing the type of operations where the condition comes from.

Possible `EDGE_FILTER_TYPE` Values Grouped into Three Logical Categories

- Represents a situation where the source affects the value of the target expression
  - `Expression`—Expression-level indirect edges such as built-in functions (`DECODE`, `SUBSTRING` indices, etc.), `CASE` expressions, etc.
  - `AnalyticWindow`—Sources participating in the definition of the analytic window for the ordered analytical functions
  - `Index`—Array index; key or field for map, JSON, XML
- Affects the whole dataset (typically affects what rows will be present or in what order)
  - `Where`—Limits the rows included in the target based on a condition
  - `JoinCondition`—Defines how a join will be performed
  - `GroupBy`—Row grouping for aggregate functions
  - `Having`—Limits aggregated rows included in the target
  - `OrderBy`—Affects row order
  - `Limit`—Limits rows included in the target based on output row count / percentage
  - `Pivot`—Pivoted columns
  - `Transpose`—Transposition of a dataset
  - `Distinct`—Removal of key columns for duplicates
  - `Temporal`—A temporal condition
- Affects whether the given operation will be executed at all
  - `If`—A condition (`if`, `case` / `switch`) affecting whether the operation will be executed
  - `Loop`—A condition affecting loop execution (`while`, `for`, `exit`, `continue`)
  - `TriggerCondition` —Trigger condition
  - `ExceptionHandling`—Connects inputs to exception handling operations



Find element...

ADDR_PURP_KEY -> REPORT_CLIENT_BODY

Source	ADDR_PURP_KEY
Target	REPORT_CLIENT_BODY
Type	FILTER
Selected	12 edges

Transformation Logic

Purpose

This functionality generates transformation descriptions, showing exactly how the value for a target (e.g., column, routine return value, etc.) is computed. The descriptions are stored on transformation column nodes immediately upstream of the target. The description does not include conditions under which the target is affected by the transformation. (That kind of information is provided in the form of filter edges in the lineage graph.)

This results in three types of transformation descriptions.

- Text version for visualization
  - Used in visualization
  - Built without the context of other scripts, which means that the user-defined routine definition is not incorporated into the transformation description
- Tag with a transformation classification

  - Visible in the visualization
  - Generated when generating the text version
  - Possible values:
    - NON_TRANSFORMING—value of one column is directly inserted into another column
    - TRANSFORMING—if the data flow contains any kind of logic
- Object version for export
  - Used for exports to third-party applications
  - Merges the user-defined routine definition from the other scripts into the final transformation description

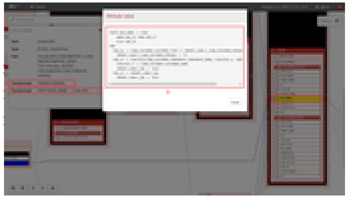
Supported Source Systems

- Oracle
- Teradata
- Microsoft SQL Server
- PostgreSQL
- Embedded SQL queries
  - DataStage
  - IPC
  - Tableau
  - SSIS (limited to queries in database source components and SQL Task)

Where the Transformation Description and Classification Is Shown

Text Version for Visualization

- *MANTA Server Viewer Visualize Node Properties Transformation Logic*
- **It is generated on nodes leading to a database persistent entity (like a column) and output points of routines.**
  - The transformation description is not directly on the target node so that one can distinguish between multiple writes to the same target.
- As the value is typically longer, it is necessary to open the *Attribute Value* dialog box to see the whole thing.
- An example of the resulting transformation description:



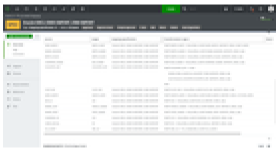
A—selected node
B—tag with the transformation classification
C—location of the text version of the transformation description
D—value of the text version of the transformation description

Tag with the Transformation Classification

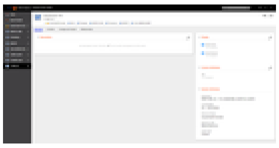
- *MANTA Server Viewer Visualize Node Properties Transformation Type*
- It is generated on the same nodes as the text version.

Object Version for Export

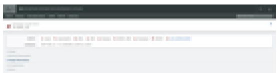
- In Collibra DGC, the transformation logic is available as the *Transformation Logic* attribute of field mapping complex relations. It can be shown on the *Mapping Specification* asset overview page, on the overview pages of the source and target attributes (e.g., column), and in data lineage diagrams.



- In Informatica EDC, the transformation logic is available as the *Expression* System attribute of attribute-level transformation assets (e.g., column). It is shown on the overview page of the particular transformation assets.



- In IBM IGC, the transformation logic is available as the *EXPRESSION* attribute of attribute-level transformation assets (e.g., column). It is shown on the overview page of the particular transformation assets.



How to Read Expressions

Each line of transformation description consists of:

```
TARGET := EXPRESSION_ELEMENT_1 EXPRESSION_ELEMENT_2 .. EXPRESSION_ELEMENT_n
```

`TARGET` is typically the column for which the transformation description is generated or the definition of the `EXPRESSION_ELEMENT` from the line for which it is a sub-line.

`EXPRESSION_ELEMENT` is typically the qualified name of a source column, operator, keyword, or variable defined in its sub-line.

This is followed by lines explaining the elements which can be drilled down (like variables). These sub-lines are indented and are valid only for their direct parent line.

Details:

- Access to an item from an array is represented as "`MY_ARR[]`".

Exceptions:

- In the definition of the OUTPUT parameter of the called routine, the TARGET (left side) is missing.
- CASE Statement is multi-lined for better readability.

Example 1: From an Oracle Import Script

```
PARTY.BUS_NAME := CASE
    WHEN VAR_16 THEN VAR_17
    ELSE VAR_19
END
  VAR_16 := LOAN_CUSTOMER.CUSTOMER_TYPE <> IMPORT_LOAN.
C_LOAN_CUSTOMER_PERSON
    IMPORT_LOAN.C_LOAN_CUSTOMER_PERSON := 'P'
  VAR_17 := COALESCE(LOAN_CUSTOMER_CORPORATE.CORPORATE_NAME, COALESCE.2,
IMPORT_LOAN.C_NA)
    COALESCE.2 := LOAN_CUSTOMER.CUSTOMER_NAME
    IMPORT_LOAN.C_NA := 'N/A'
  VAR_19 := IMPORT_LOAN.C_NA
    IMPORT_LOAN.C_NA := 'N/A'
```

Here is some information about each line of the text version of the transformation description in the preceding example.

1. The first line tells us that this transformation description is for column `BUS_NAME` from table `PARTY` and its value is created from a CASE statement.
2. The conditional value for the CASE statement, where the temporal variables (`VAR_16` and `VAR_17`) generated only for this transformation description are used. They are used when they represent a sub-expression that is too long to inline or would have a negative effect on transformation description readability. They have the format `VAR_xx` (which can be parametrized in an XML file).
3. Part of the CASE statement like on line 2.
4. End line of the CASE statement.

5. Sub-line of line 1 (visible by indentation) that defines `VAR_16`.
6. Sub-line of line 5 that defines the literal value of the constant `C_LOAN_CUSTOMER_PERSON`.
7. Sub-line of line 1 that defines the value of `VAR_17` as a result of calling the built-in function COALESCE. The first parameter is directly in this line, but the second and third parameters are defined in the following sub-lines.
8. Sub-line of line 7 that defines the value of the second parameter of the function COALESCE as the column `CUSTOMER_NAME`.
9. Sub-line of line 7 that defines the value of the third parameter of the function COALESCE as literal `'N/A'`.
10. Sub-line of line 1 that defines the value of `VAR_19`. Here the value `IMPORT_LOAN.C_NA` is not inlined since it is detected as a variable that can have a meaningful name, which can help with the readability of the resulting transformation description.
11. Sub-line of line 10 that defines the value of `IMPORT_LOAN.C_NA` as literal `'N/A'`.

Example 2: From an Oracle Function Definition Script

```
TEST_TL_FNC1.RETVAL := TEST_TL_FNC1.IN1 + 2
  DWH.TEST_TL_FNC1(TEST_TL_FNC1.IN1)
```

Here is some information about each line of the text version of the transformation description in the preceding example.

1. The first line tells us that this transformation description is for a return value from the user-defined function `TEST_TL_FNC1` and its value is created as the value of the input parameter `IN1` plus 2.
2. This line defines the input parameters inside user-defined routines since it is variable and it could be unclear from where it originates. (The same line format is used for sub-lines that define output parameters from routines. But it should be clear which type it is from context, as this line only occurs in the transformation description of the output point of the routine so the routine name matches the qualified name of the final target.)

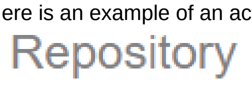
Limitations

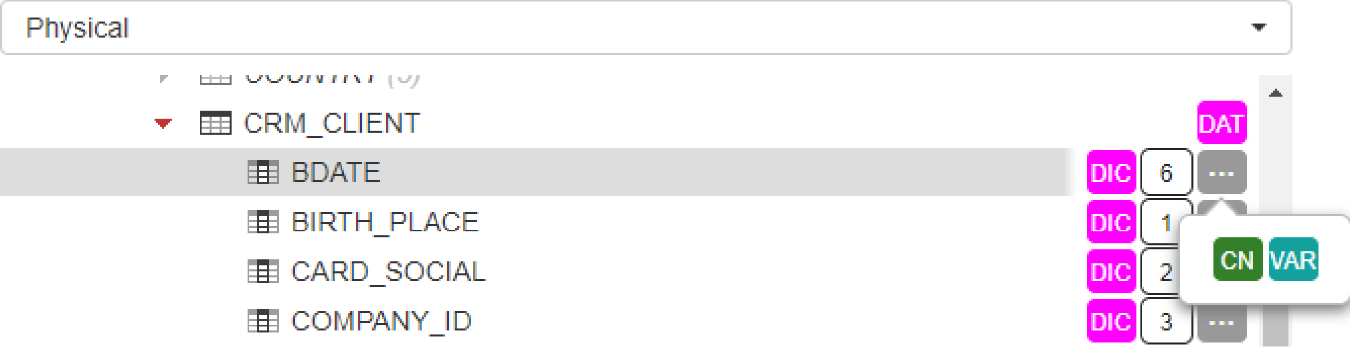
- The transformation logic module is still in its experimental version. This means:
  - Some less common constructs are not supported yet.
  - The level of support varies across technologies.
  - The format of generated transformation description may change in future releases.
  - It is not guaranteed that new versions of MANTA will be able to read the descriptions (object version) created by an older version, that is, these descriptions may be lost in old repository revisions after the update.
- Most conditions (like WHERE clauses and IF blocks) are intentionally omitted from descriptions because they would make the description too long which would work against the intended usage of this information. If they are needed, they can be displayed in MANTA using the option to *Visualize Indirect Flows*.
- The command execution order is not considered. This means that in the case of a `SET1; USE; SET2;` sequence of commands, the description for `USE` will show values from both `SET1` and `SET2`.

Active Tags in MANTA Flow Visualization

Active tags are a specialized kind of attribute visible next to the object name in the repository tree and dataflow visualization. Available for nodes, this is an attribute that also carries its own metadata characteristics. The set of active tags available in your MANTA instance is configured by the MANTA Administrator. The active tag feature is available in MANTA as of version 3.33.

Here is an example of an active tag visualization. The active tags are the boxes on the right, next to the object name.


Repository



Note: If you wish to add a new active tag or modify an existing one, please contact your MANTA administrator.

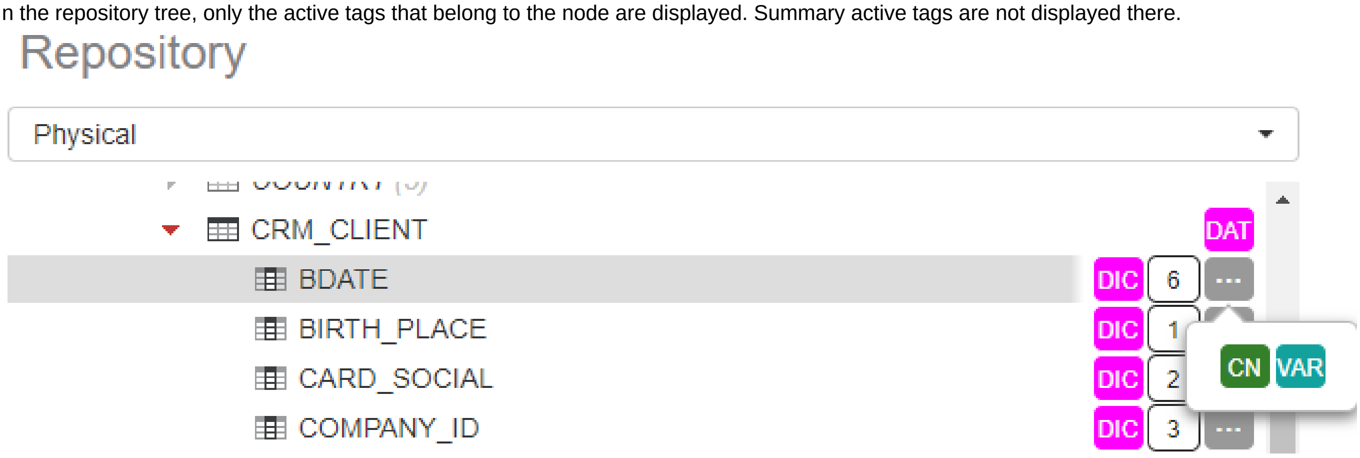
Displaying Active Tags in MANTA

Active tags are displayed in the repository tree and in the data lineage. In both cases, the active tags that belong to the node are displayed next to the node. In the data lineage, so-called summary tags are also displayed. A summary active tag is an indicator that some of the node's descendants have that active tag. Summary active tags are only displayed in the data lineage view.

Displaying Active Tags in the Repository Tree

In the repository tree, up to 3 active tag icons are displayed for each node. If the node has more active tags, the hidden tags will appear in the tooltip when hovering over the grey ellipsis icon.

In the repository tree, only the active tags that belong to the node are displayed. Summary active tags are not displayed there.

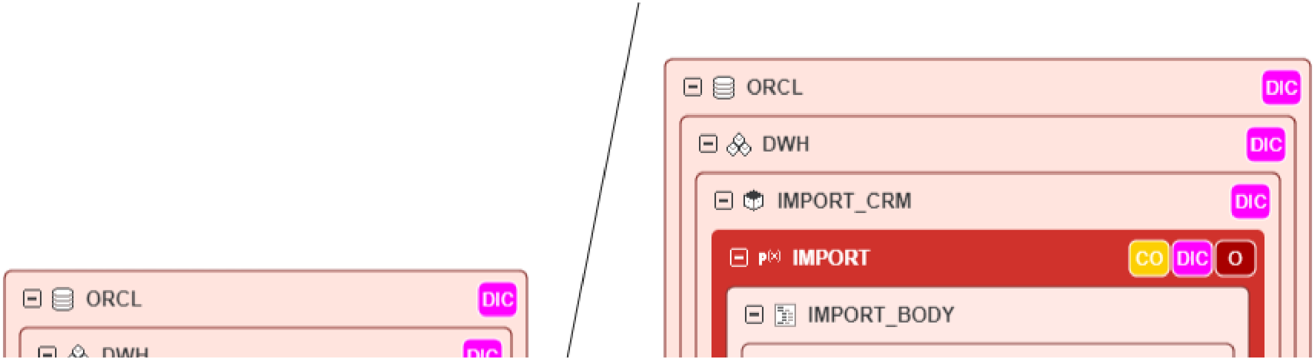


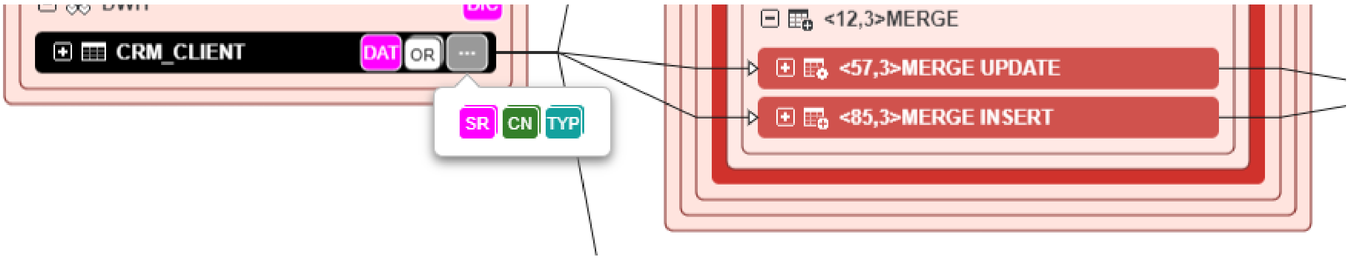
Displaying Active Tags in the Data Lineage

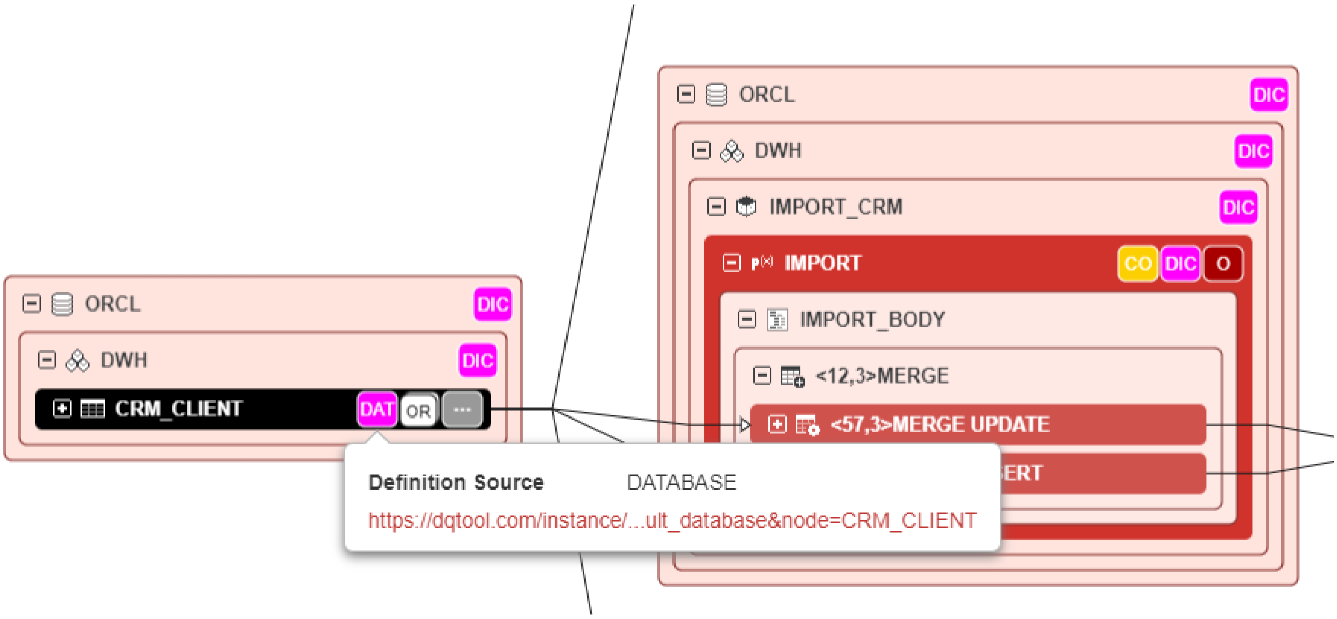
Up to 3 active tag icons are displayed for each node in the lineage visualization. If the node has more than 3 active tags, the hidden tags will appear in the tooltip when hovering over the grey ellipsis icon.

When the node is collapsed (left part of the image below), summary active tags are displayed if any of the node descendants have an active tag. The summary active tag icon is displayed as two smaller icons on top of each other (e.g., the icons "OR", "SR", "CN", and "TYP" in the image below).

For the summary tag, the tooltip displays information on how many descendants the active tag has. For ordinary active tags, the tooltip displays attributes and links based on the configuration.







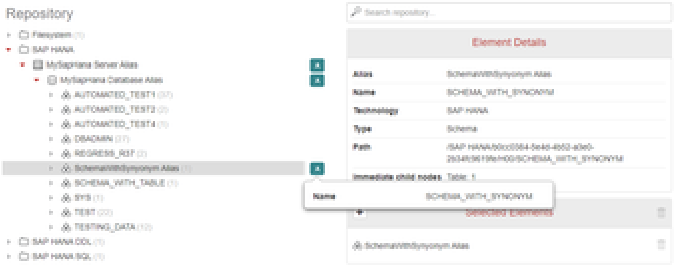
Definition Source DATABASE
https://dqtool.com/instance/...ult_database&node=CRM_CLIENT

Display User-Supplied Aliases in MANTA Flow Visualization

As of MANTA R38, it’s possible to display aliases (defined and imported with the Open MANTA extensions import) instead of physical node names. This lets you see more user-friendly, less technical names in the MANTA Flow repository tree and visualization. You can’t create an alias for the resource name (the top level of the repository tree), but you can do so for all levels under resources. The most common use of aliases is to supplement server names that could be confusing.

If you provide an alias for a node, it’ll be displayed with the active tag “A”. Hover over the tag to see the original name of the node.

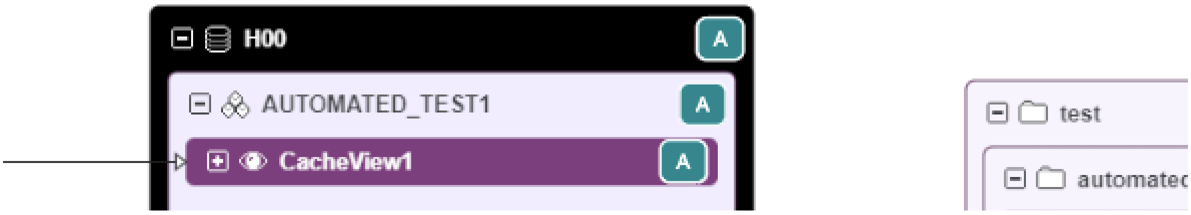
Repository Tree

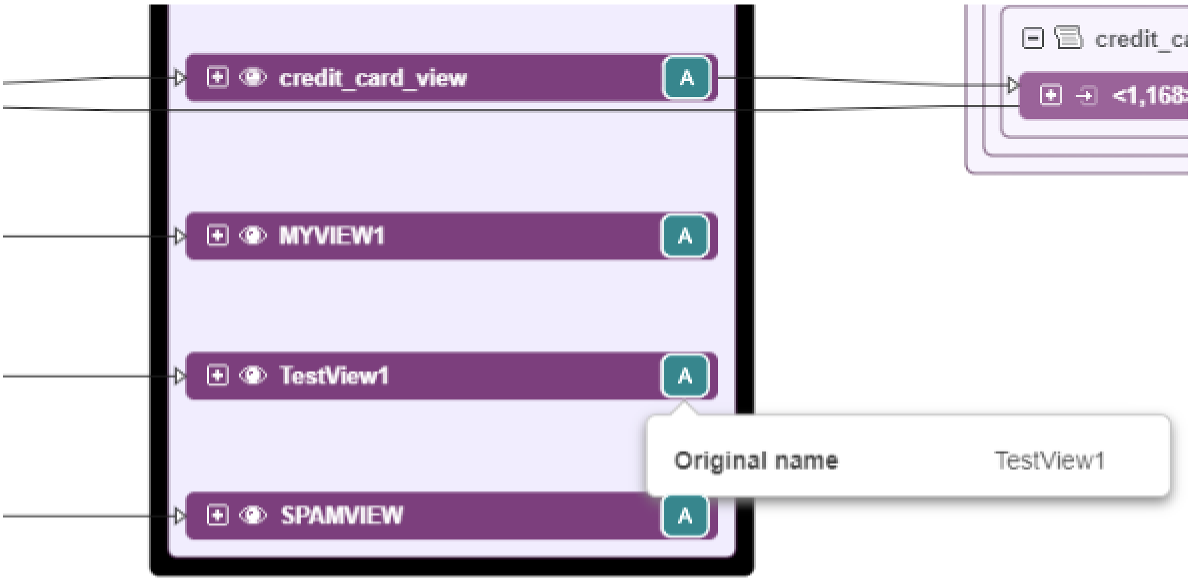


Alias	SchemaWithSynonym Alias
Name	SCHEMA_WITH_SYNONYM
Technology	SAP HANA
Type	Schema
Path	/SAP HANA/50cc0384-5e4d-4b52-a3e0-2634fc9619fe/H00/SCHEMA_WITH_SYNONYM
Immediate child nodes	Table: 1

Name	SCHEMA_WITH_SYNONYM

Lineage Visualization





Known Issues

In R38, you can’t search for items in the object catalog using their aliases—you’ll still need to use the physical name.

How to Supply the Aliases

1. Create an Open MANTA extensions import connection.
2. Create the respective Open MANTA extensions import CSV files: `layer.csv`, `resource.csv`, `node.csv`, and `node_attribute.csv`.
3. Create the node(s) you want to add an alias to in the files above.
4. Add the entry `MANTA_ALIAS` into `node_atttributes.csv`. Then enter the alias.
5. Import the data as part of your regular lineage scan.

How the Aliases Are Displayed

Once you’ve imported the aliases, you’ll see them instead of the node name in the repository tree and the lineage visualization. Hover over the alias active tag “A” to see the original node name. See the images above for an example.

If you open the node detail, both the alias and name of the node are displayed.