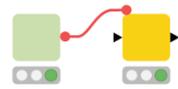


CONTROL

Flow Variables allow for the parameterization of a workflow. A Flow Variable is a parameter that can assume different values at different execution points in the workflow & overwrite configuration settings in upcoming nodes.



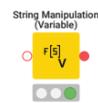
Hidden Flow Variable Ports

Each node has two hidden Flow Variable ports to accept incoming Flow Variables & to propagate them to the upcoming nodes. To make these ports visible, right-click the node & select **Show Flow Variable Ports**. Only ports of the same type can be connected.

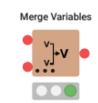


Creating a Flow Variable

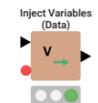
1. Use a Configuration or a Widget node to create a Flow Variable at any point in your workflow.
1. Use any of the nodes converting data into Flow Variables.
3. Via the node configuration window in the **Flow Variables** tab, fill in a blank box with the name of the Flow Variable.



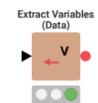
String Manipulation (Variable): This node is the Flow Variables version of the String Manipulation node. Similarly, other nodes have their own version for Flow Variables like the Rule Engine Variable node & the Math Formula (Variable) node.



Merge Variables: Combines Flow Variables from two or more separate branches. To add a branch click the three dots in the bottom left corner. If Flow Variables with the same name are collected, the Flow Variable in the top most connection is retained.

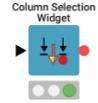


Inject Variables (Data): Adds (injects) the Flow Variables at its Flow Variable input port into the data table at the top input port. The input data table is forwarded (unaltered) in the node output port.



Extract Variables (Data): Extracts the Flow Variables coming in through the input data port & produces them as standalone Flow Variables at the output port.

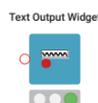
Widget and Configuration nodes create one or more new flow variables & make them available at the output port. Widget nodes create a UI item for the composite view or the KNIME WebPortal (textbox, radio button, etc.) to create and control the flow variable. Configuration nodes create a UI item in the configuration dialog of a component & are not visible in the composite view or the KNIME WebPortal.



Column Selection Widget: Creates a list of selectable columns from the input data table in the form of a menu or radio buttons. The node produces the name of the selected column in a flow variable at its output port.



Interactive Range Slider Filter Widget: Creates a slider to filter data to only include rows with values in the selected column within the specified range. The slider can interact with views from other JavaScript based nodes in the same composite view.



Text Output Widget: Creates a paragraph of either free, preformatted, or HTML text.



Credentials Widget: Creates fields to enter credentials (username & password) in the form of text boxes. The text in the password box is masked. The node produces these credentials in a flow variable at its output port.



Single Selection Configuration: Creates a list of options of type String in the form of a menu or radio buttons. These options can be defined in the configuration dialog together with the selected default value. The node produces the value of the selected option in a flow variable at its output port.



Boolean Configuration: Creates a boolean selection for an enabled/disabled flag (1/0) in the form of a checkbox. The node produces the value of the selected option in a flow variable at its output port (1 if the checkbox is enabled, 0 if disabled).

A Metanode or Component is a node that contains other nodes.



Metanodes just collect nodes inside and are an efficient way to clean up your workflow.

Creating a Metanode or Component

Select all relevant nodes, right-click and select **Create Metanode...** for a Metanode or **Create Component...** for a Component. Right-clicking a Metanode or Component opens the context menu with a number of options such as open, expand, setup, or reconfigure, and save as template.



Components encapsulate & abstract functionality, can have their own dialog and can have their own sophisticated, interactive views. They can be reused in your own workflows but also shared with others: via KNIME Server or the KNIME Hub. They can also represent web pages in a Data App deployed to others via KNIME Server. Flow Variables cannot enter or exit a Component, unless explicitly configured in the component's input and output nodes.

A loop is a sequence of operations that is repeated until a condition is met. It has a start, an end, & a loop body of operations. A loop is implemented via a Loop Start node, a Loop End node, & a number of nodes in between for the body of operations. Different Loop Start nodes provide alternative ways to iterate on the input data. Different Loop End nodes provide alternative ways to collect results. The end condition can be defined in either the Loop Start node or the Loop End node, depending on the kind of loop. Some nodes to start & end a loop work on data & others on Flow Variables. These nodes can be paired up freely - loops can start with data & end up with Flow Variables & vice versa.

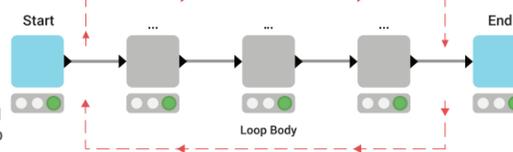


Table Row to Variable Loop Start: Starts a loop iterating over input data rows one at a time i.e. each iteration is dedicated to just one row. At each iteration, the row values are converted into Flow Variables & named after the column headers.



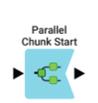
Group Loop Start: Starts a loop iterating over groups of input data rows i.e. each iteration works on a different group of the input data. Groups are extracted from values in selected columns, as in the configuration window of the GroupBy node.



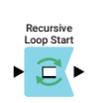
Column List Loop Start: Starts a loop iterating over a selected list of columns. At each iteration, the current column & the remaining columns are passed into the loop body.



Generic Loop Start: Starts a loop. It should be paired with a loop end node defining the end condition.



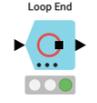
Parallel Chunk Start: Starts a parallel loop iterating over input data rows in chunks of equal size which get processed in parallel in the loop body. The number of parallel iterations, (chunks), is defined either automatically based on the size of the input data, or manually.



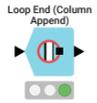
Recursive Loop Start: Starts a recursive loop iterating on the updated input table. The input data table for each iteration is the output data table from the previous iteration. The first iteration works on the data table provided at the input port of the node. This is the only loop where the updated data table feeds the next iteration.



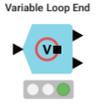
Feature Selection Loop Start (1:1): Starts a feature selection loop iterating on a set of input columns (features) to extract the subset that optimizes a given Flow Variable. Used for feature selection against a given model & against a given performance metric. Together with the Feature Selection Loop End node, it's used in the Backward/Forward Feature Selection metanode.



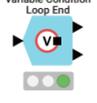
Loop End: Ends a loop by concatenating the resulting rows from each iteration.



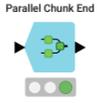
Loop End (Column Append): Ends a loop by joining together the resulting columns from each iteration on the column containing the row IDs.



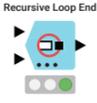
Variable Loop End: Loop End nodes do not only work on data. At the end of a loop you might want to pass the results as a Flow Variable - like in the Variable Loop End node.



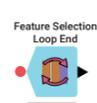
Variable Condition Loop End: Ends a loop when a condition is met, i.e. a specific value in a Flow Variable.



Parallel Chunk End: Ends a parallel loop by concatenating the resulting data rows from each chunk.

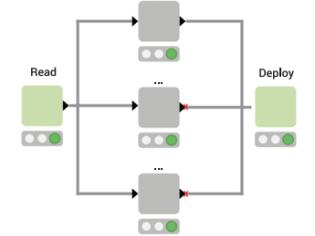


Recursive Loop End: Ends a recursive loop. The top input port collects the resulting data rows. The lower input port collects the updated data table to be passed back to the Recursive Loop Start node. Also defines the loop end condition, such as max. number of iterations, min. number of rows, & a specific value for a Flow Variable.



Feature Selection Loop End: Ends a feature selection loop. It sets the variable with the metric to be maximized/minimized for best feature selection. The top output port of the node produces the evaluation metrics for all column subsets. The bottom output port produces a summary of the different feature sets with the associated scores.

A switch construct allows you to conditionally execute different sequences of operations via nodes located on different workflow branches. All start with a Switch Start node and optionally end with a Switch End node. In between, a number of parallel branches implement various operation sequences.



A switch construct works on data, flow variables, database queries, or any other port type defined via the dynamic ports of the start and end nodes. You can pair up different port types.



CASE Switch Start: Selectively activates only one of its three output ports, enabling three alternative paths for the input. The active output port can be configured manually or automatically via the value of a Flow Variable. The IF Switch node performs the same task but with only two alternative output ports (both can be active at the same time).



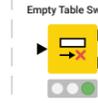
CASE Switch End: Collects the results from the active one among the branches connected to its input ports. The End IF node works similarly & is paired with the IF Switch node.



Try (Data Ports): Starts a try-catch construct to enable an alternative path for the data flow in case of failure in the main branch. One branch is defined as the main branch while the other is set as the secondary branch. If execution fails in the main branch, the secondary branch is activated. It must be closed by a Catch node.



Catch Errors (Data Ports): Closes a try-catch construct started with a Try node & collects the results from the active branch.

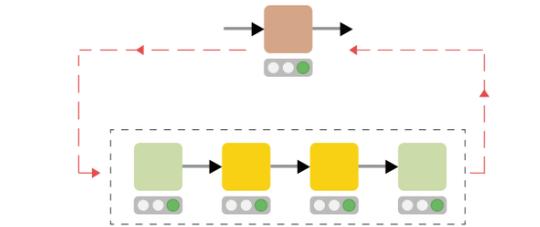


Empty Table Switch: Provides an alternative path for the data flow in case the main branch has no data rows. It activates the top output port & deactivates the bottom output port if the input table has at least one data row. It deactivates the top output port & activates the bottom output port if the input table is empty.



Active Branch Inverter: Changes the activity status of the branch. If the input port is active, the output port becomes inactive & vice versa. It's often used to force a branch to produce an output even if it's inactive & vice versa (to deactivate a branch even if it's active).

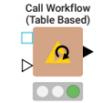
ORCHESTRATION



Call Workflow Service: Triggers the execution of a workflow stored in the LOCAL workspace or on KNIME Server. Data exchange with the triggered workflow can happen via any port type via the Workflow Service Input/Output nodes.



Workflow Service Input: Receives data via any port type from the caller workflow via the Call Workflow Service node. Enables an efficient data exchange between workflows only, excluding third-party software. The corresponding "Workflow Service Output" node returns the results to the caller workflow.



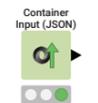
Call Workflow (Table Based): Triggers the execution of a workflow stored in the LOCAL workspace or on KNIME Server. Data exchange with the triggered workflow can happen via data tables, Flow Variables, or credentials via the respective Container Input/Output nodes.



Container Input (Table): Receives a data table from the caller workflow. If no input is provided, the template default data table is used. Similar nodes are available to exchange Flow Variables & credentials. The corresponding "Container Output (Table)" node returns the results as a data table.



Call Remote Workflow (Row Based): Triggers the execution of a remote external workflow via REST. Remote means stored on a KNIME Server accessed using the server URL & credentials. Data exchange with the triggered workflow happens via JSON format.



Container Input (JSON): Receives a JSON data structure from the caller workflow. If no input is provided, the template default JSON structure is used. The corresponding "Container Output (JSON)" node returns the results as a JSON structure.



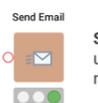
GET Request: Calls a REST service in GET mode. The node can send one single service request set in the configuration window, or multiple service requests stored in a column of the input table. Responses are saved in the output data table. Options to set authentication, request header, & response header are available.



KNIME Server Connector: Connects to a KNIME Server using the server URL & credentials. After the connection has been created, new directories on the server can be created & remote files can be accessed, created, & deleted.



Timer Info: Reports the number of executions & execution times for each node in a workflow. Both single node & total workflow execution time are reported. Execution times for nodes inside metanodes can also be reported.



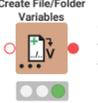
Send Email: Sends HTML or text formatted emails using an external SMTP to a recipient - including the message & possible attachments.



Save Workflow: Saves the (also partially - up to here) executed workflow.



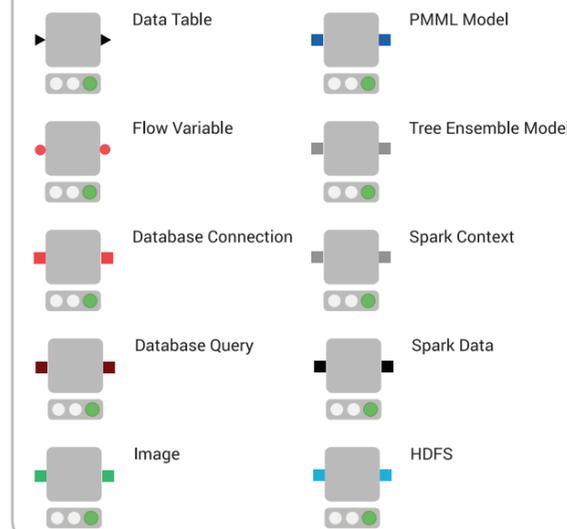
Create Folder: Creates a new folder and outputs the folder location as a flow variable of type Path.



Create File/Folder Variables: Creates a list of Path type Flow Variables pointing to files/folders relative to a selected base location.

Node ports

Different types of data pass through different node ports. Only ports of the same type can be connected. Here are some examples of ports for frequently used data types.



Resources

E-Books: KNIME Advanced Luck covers these advanced features & more. Practicing Data Science is a collection of data science case studies from past projects. Both available at knime.com/knimepress

KNIME Blog: Engaging topics, challenges, industry news, & knowledge nuggets at knime.com/blog

E-Learning Course: Take our free online self-paced course to learn about the different steps in a data science project (with exercises & solutions to test your knowledge) at www.knime.com/knime-self-paced-courses

KNIME Hub: Browse and share workflows, nodes, and components. Add ratings, or comments to other workflows at hub.knime.com

KNIME Forum: Join our global community & engage in conversations at forum.knime.com

KNIME Server: For team-based collaboration, automation, management, & deployment check out KNIME Server at www.knime.com/knime-server

Extend your KNIME knowledge with our collection of books from KNIME Press. For beginner and advanced users, through to those interested in specialty topics such as topic detection, data blending, and classic solutions to common use cases using KNIME Analytics Platform - there's something for everyone. Available for download at www.knime.com/knimepress.

