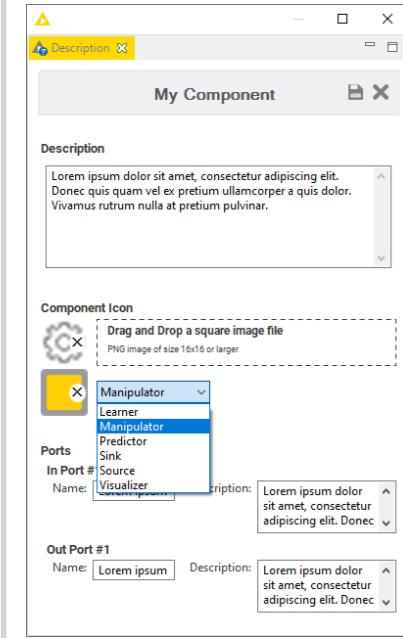


Cheat Sheet: Components with KNIME Analytics Platform

COMPONENTS DESCRIPTION



Description: Documents the purpose and usage of the component and defines the appearance of the component. To access and edit the component description, open the Description panel from inside the component and select the empty canvas.

Description: Provides the text describing the use case, requirements, licensing and copyrights, disclaimers, and any other extra information you would like to add.

Icon: Drag and drop a PNG 16x16 px from your local file system. The icon will appear on the component, giving it a unique look.

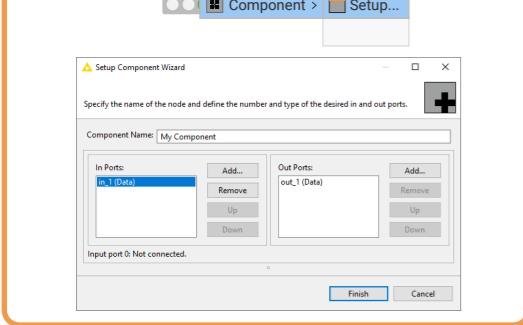
Color: Usually the color of nodes and components is associated with a precise category. Pick the category that best fits your use case: The selected color appears behind the logo.

In/Out Ports: Describe the ports requirements here. For example: What kind of input column types are supported? Are there new rows/columns in the output? Does the input/output connect only to a precise other node or component?

Configure Options: The component description also lists all the settings available inside the component dialog. To add text describing the settings usage open each configuration node dialog and enter it there node by node.

COMPONENT SETUP

You can still change component names and ports after component creation. The panel offers a text field to edit the title, buttons to change the order, remove or add. When adding a new port, a drop down menu appears to define the port type.

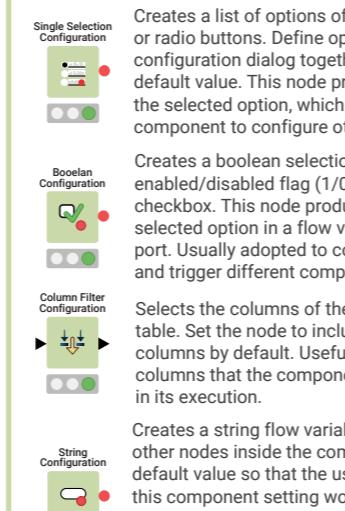


VERIFIED COMPONENTS

Verified Components behave like actual nodes and are regularly released on the KNIME Hub. Browse them all at knime.com/verified-component. Each new component is assigned to one of the 9 categories, represented by a color and symbol.

AutoML	Model Monitor View	SARIMA Learner	Time Series Example
Animated Bar Chart	Measure Fractional Years	Web Text Scraper	Finance Analytics Example
Data Manipulation Example	Global Feature Importance	Molecular Properties Filter	Model Interpretability Example

CONFIGURATION DESCRIPTION



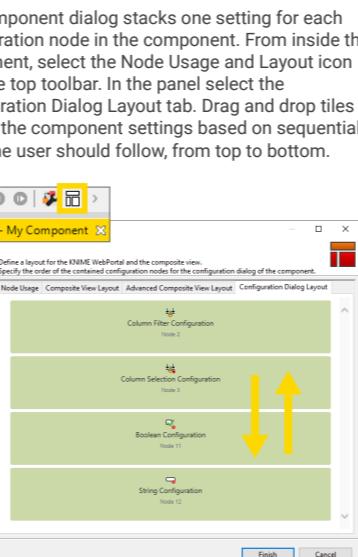
Single Selection Configuration: Creates a list of options of type String for a menu or radio buttons. Define options in the configuration dialog together with the selected default value. This node produces the value of the selected option, which can be used inside the component to configure other node settings.

Boolean Configuration: Creates a boolean selection for an enabled/disabled flag (1/0) in the form of a checkbox. This node produces the value of the selected option in a flow variable at its output port. Usually adopted to configure switch nodes and trigger different component modes.

Column Filter Configuration: Selects the columns of the component input table. Set the node to include or exclude all columns by default. Useful to remove columns that the component should not use in its execution.

String Configuration: Creates a string flow variable for configuring other nodes inside the component. Pick a default value so that the user understands how this component setting works. The string flow variable is useful to determine how to rename the component output columns or for text displayed in the component's composite view.

DIALOGUE LAYOUT PANEL



The component dialog stacks one setting for each configuration node in the component. From inside the component, select the Node Usage and Layout icon from the top toolbar. In the panel select the Configuration Dialog Layout tab. Drag and drop tiles to change the component settings based on sequential steps the user should follow, from top to bottom.

WIDGET NODES

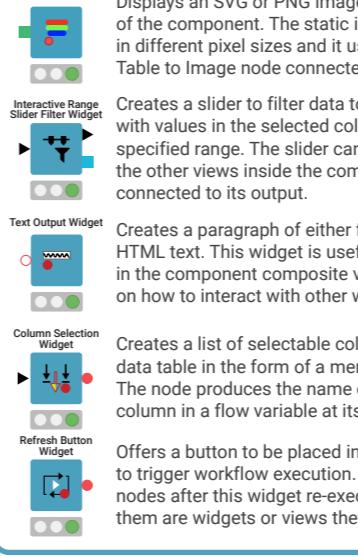


Image Output Widget: Displays an SVG or PNG image in the composite view of the component. The static image can be displayed in different pixel sizes and it usually comes from a Table to Image node connected to this widget.

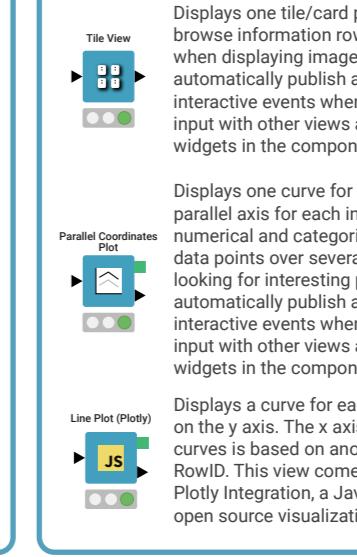
Interactive Range Slider Filter Widget: Creates a slider to filter data to only include rows with values in the selected column within the specified range. The slider can interact with most of the other views inside the component when connected to its output.

Text Output Widget: Creates a paragraph of either free, preformatted, or HTML text. This widget is useful to show styled text in the component composite view, guiding the user on how to interact with other widget and views.

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.

Refresh Button Widget: Offers a button to be placed in the composite view to trigger workflow execution. When clicked, the nodes after this widget re-execute and if any of them are widgets or views they are also updated.

VIEW NODES

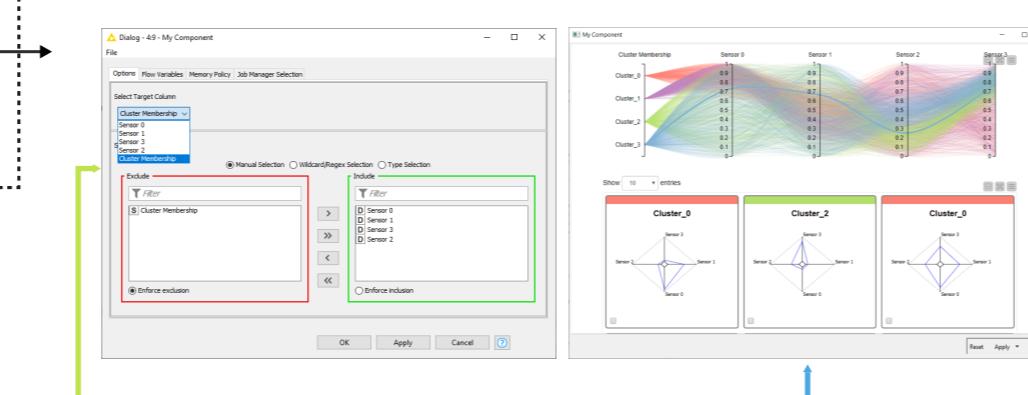


Tile View: Displays one tile/card per row. Useful to browse information row after row esp. when displaying image or text data. It can automatically publish and subscribe to interactive events when it shares the same input with other views and/or interactive widgets in the component.

Parallel Coordinates Plot: Displays one curve for every row and one parallel axis for each included column, both numerical and categorical. Useful to explore data points over several dimensions, looking for interesting patterns. It can automatically publish and subscribe to interactive events when it shares the same input with other views and/or interactive widgets in the component.

Line Plot (Plotly): Displays a curve for each selected column on the y axis. The x axis is between the curves is based on another column or the RowID. This view comes from the KNIME Plotly Integration, a JavaScript based open source visualization library.

COMPOSITE VIEW LAYOUT

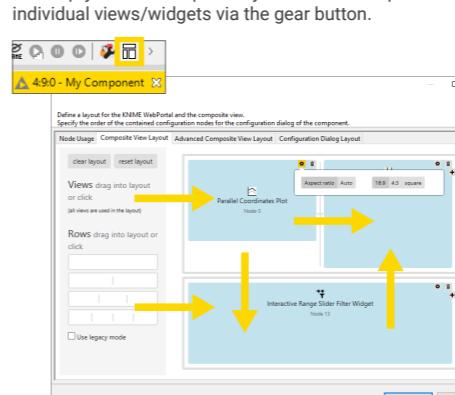


Create: To create a component multiple-select all the nodes you want to encapsulate in the component. Now right-click the selection and select "Create Component".

Configure: If you included a Configuration node in the component, open the configuration dialog. Change the component settings here before executing, just like any KNIME node.

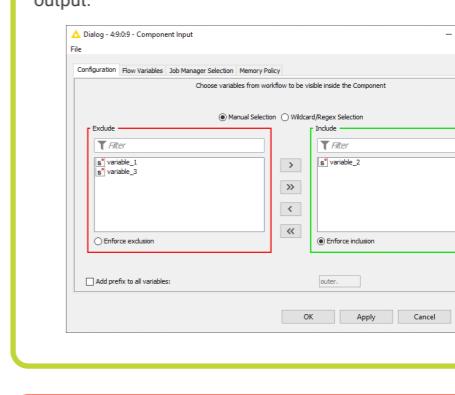
Open View: After executing a component that contains a Widget or View node, you can open its interactive view. In the opened composite view, interactivity across the single views and widgets nodes in the component is activated by default: visualizations sharing the same input table are usually connected via selection and filter events.

FLOW VARIABLE FILTER



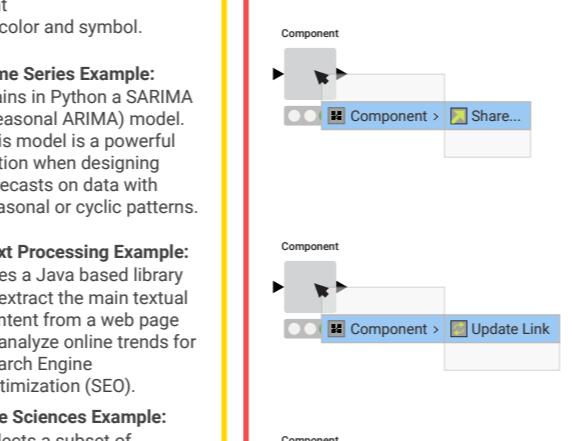
By default KNIME stacks views and widgets from top to bottom in the composite view. Access the Composite View Layout tab via the Node Usage Layout panel from inside the component. Define the layout of the composite view with tables: reset/clear the current layout (top left buttons); drag rows from the left; add columns using "+"; drag the views/widgets on the left into the empty structure. Optionally set the ratio or pixel sizes of individual views/widgets via the gear button.

ERROR HANDLING



Fails on purpose when a certain condition is met. A custom error message appears on the node and on the outside of the component. Use it to detect whether the input or configurations of the component satisfy minimum requirements and provide the user with an intuitive message on what should be fixed after making the component fail on purpose.

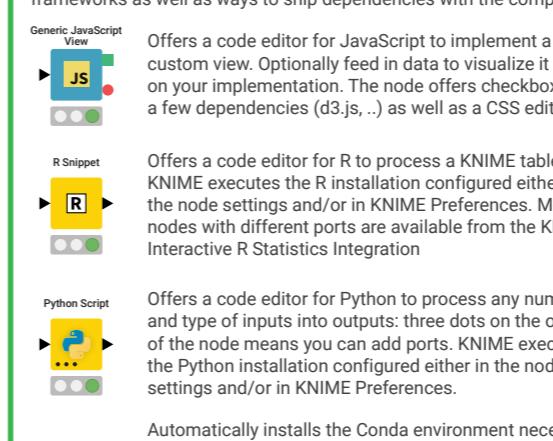
SHARED COMPONENTS



Components can be reused as your personal customized KNIME nodes or they can also be shared with others via KNIME Hub and KNIME Server.

Component	Component > Share...
Component	Component > Update Link
Component	Component > Disconnect Link

SCRIPTED COMPONENTS



Component builders can optionally implement component functionalities through coding. KNIME supports a number of coding frameworks as well as ways to ship dependencies with the component.

Generic JavaScript View	R Snippet	Python Script	Conda Environment Propagation

Generic JavaScript View: Offers a code editor for JavaScript to implement a custom view. Optionally feed in data to visualize it based on your implementation. The node offers checkboxes for a few dependencies (d3.js,..) as well as a CSS editor.

R Snippet: Offers a code editor for R to process a KNIME table. KNIME executes the R installation configured either in the node settings and/or in KNIME Preferences. More nodes with different ports are available from the KNIME Interactive R Statistics Integration

Python Script: Offers a code editor for Python to process any number and type of inputs into outputs: three dots on the outside of the node means you can add ports. KNIME executes the Python installation configured either in the node settings and/or in KNIME Preferences.

Conda Environment Propagation: Automatically installs the Conda environment necessary for your component to execute the downstream R/Python nodes. The environment usually includes the R/Python installation plus precise versions of the libraries. Useful to share scripted components with the necessary dependencies. Requires installation of Anaconda or Miniconda and minimal configuration in KNIME Preferences.

Resources

E-Books: KNIME Advanced Luck covers 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 Courses: Take our free online self-paced courses 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

KNIME Press

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.

