Open for Innovation
KNIME

KNIME File Handling Guide

KNIME AG, Zurich, Switzerland
Version 5.2 (last updated on 2024-04-08)

Table of Contents

Introduction . . . 1
Basic concepts about file systems . . . 2
  Working directory . . . 2
  Hidden files . . . 2
  Path syntax . . . 2
KNIME Analytics Platform and file systems . . . 3
  Standard file systems . . . 3
  Connected file systems . . . 10
Read and write from or to a connected file system . . . 21
  Reader nodes . . . 21
  Writer nodes . . . 26
Path data cell and flow variable . . . 28
  Creating path data cells . . . 29
  Manipulating path data cells . . . 29
  Creating path flow variables . . . 30
  String and path type conversion . . . 30
File Folder Utility nodes . . . 32
  Table based input Utility nodes . . . 33
Compatibility and migration . . . 34
  How to work with workflows that contain both old and new file handling nodes . . . 34
  How to migrate your workflows from old to new file handling nodes . . . 35

Introduction

With the move towards cloud and hybrid environments we had to rethink and rewrite the existing file handling infrastructure in KNIME Analytics Platform to provide our users with a better user experience.

With KNIME Analytics Platform release 4.3 we introduced a new file handling framework thanks to which you can migrate workflows between file systems or manage various file systems within the same workflow in a more convenient way.

In this guide the following topics are covered:

- Basic concepts concerning file systems
- How to access different file systems from within KNIME Analytics Platform
- How to read and write from and to different file systems and conveniently transform and adjust your data tables when importing them into your workflow
- The new Path type and how to use it within the nodes that are build on the basis of the file handling framework
- Finally, in the Compatibility and migration section you will find more detailed information about:
  - How to distinguish between the old and new file handling nodes
  - How to work with workflows that contain both old and new file handling nodes
  - How to migrate your workflows from old to new file handling nodes.

© 2023 KNIME AG. All rights reserved.

Basic concepts about file systems

In general, a file system is a process that manages how and where data is stored, accessed and managed.

In KNIME Analytics Platform a file system can be seen as a forest of trees where a folder constitutes an inner tree node, while a file or an empty folder are the leaves.

Working directory

A working directory is a folder used by KNIME nodes to disambiguate relative paths. Every file system will have a working directory whether it is explicitly configured or implicitly.

Hidden files

Within KNIME Analytics Platform hidden files and folders are not displayed when browsing a file system. However they can be referenced knowing the path. Hidden files currently only exist for the local file systems in KNIME Analytics Platform:

- On Linux and MacOS their filename starts with a dot `"."`
- On Windows instead they are treated as regular files and folders

Path syntax

A path is a string that identifies a file or folder position within a file system. The path syntax depends on the file system, e.g. a Windows local file system might look like `C:\Users\username\file.txt`, while on Linux and most other file systems in KNIME Analytics Platform might look like `/folder1/folder2/file.txt`.

Paths can be distinguished in:

- Absolute: An absolute path uniquely identifies a file or folder. It starts always with a file system root.
- Relative: A relative path does not identify one particular file or folder. It is used to identify a file or folder relative to an absolute path.

© 2023 KNIME AG. All rights reserved.

KNIME Analytics Platform and file systems

Different file systems are available to use with KNIME Analytics Platform. Reader and writer nodes are able to work with all supported file systems.

File systems within KNIME Analytics Platform can be divided into two main categories:

- Standard file systems
- Connected file systems

Standard file systems

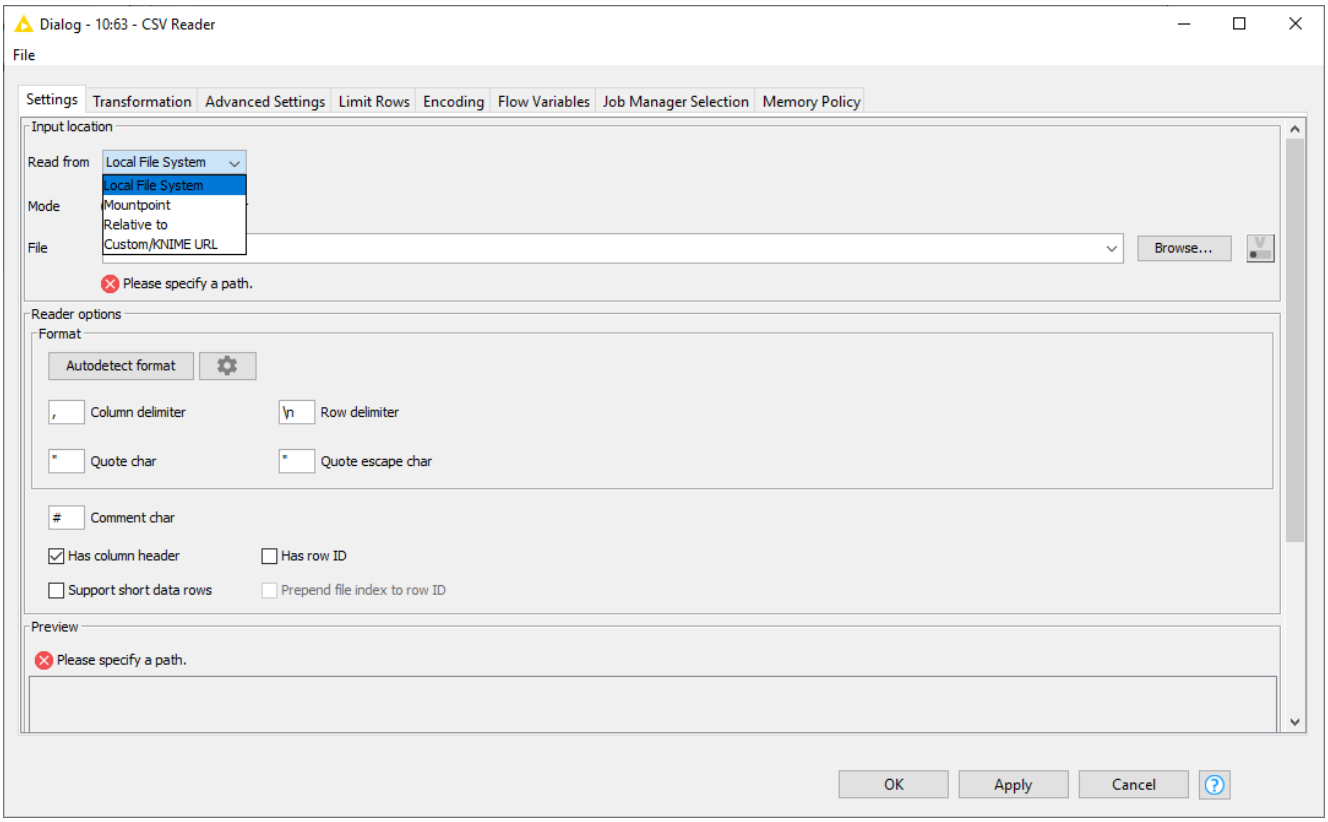
Standard file systems are available at any time meaning they do not need a connector node to connect to it.

Their working directory is pre-configured and does not need to be explicitly specified.

To use a reader node to read a file from a standard file system drag and drop the reader node for the file type you want to read, e.g. CSV Reader for a `.csv` file, into the workflow editor from the node repository.

Right click the node and choose *Configure…* from the context menu. In the *Input location* pane under the tab *Settings* you can choose the file system you want to read from in a drop-down menu.

© 2023 KNIME AG. All rights reserved.



The following standard file systems are available in KNIME Analytics Platform:

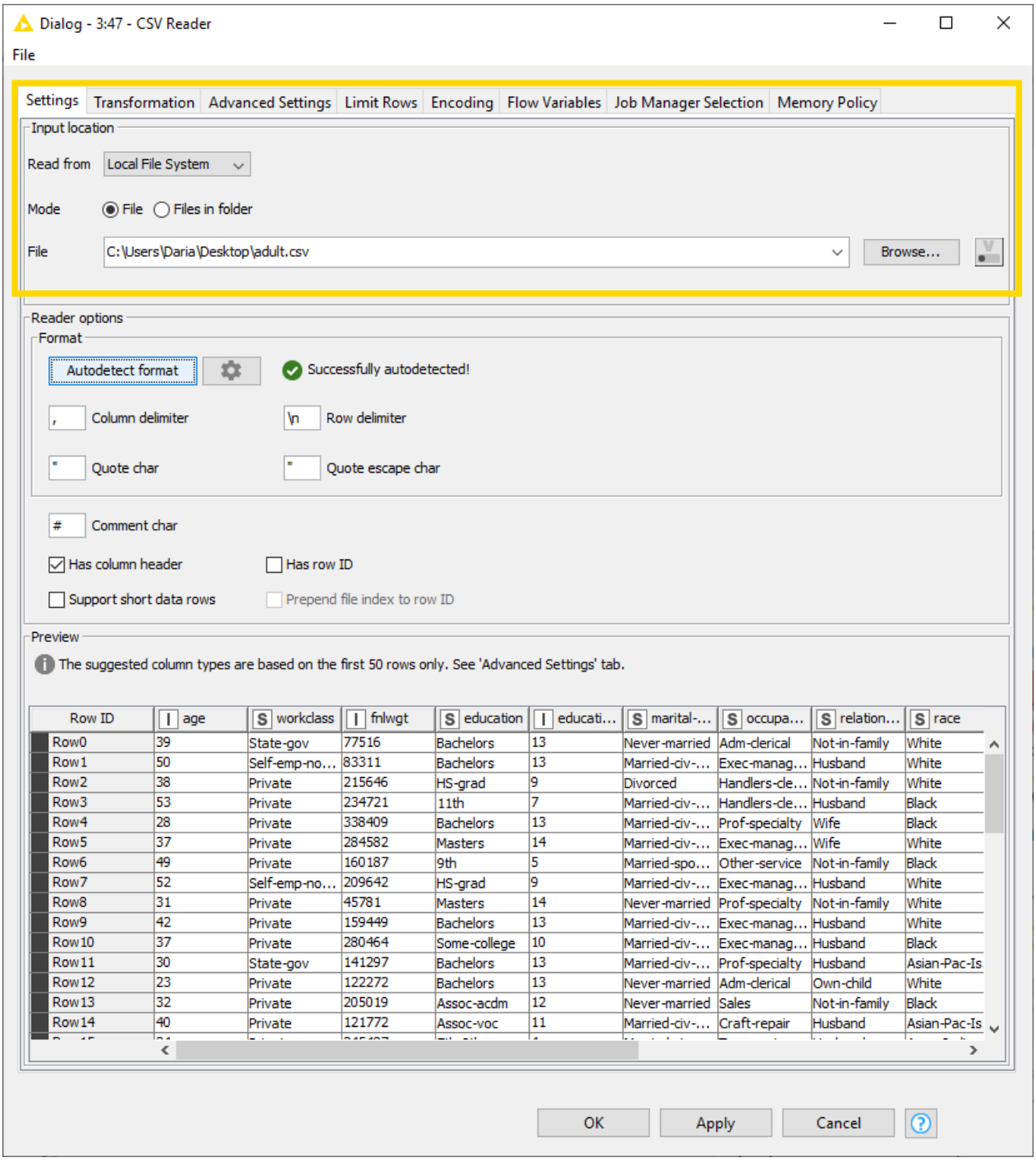
- Local file system
- Mountpoint
- Relative to
  - Current workflow
  - Current mountpoint
  - Current workflow data area
- Custom/KNIME URL

Local file system

When reading from *Local File System* the path syntax to use will be dependent on the system on which the workflow is executing, i.e. if Windows or UNIX operative system.

The working directory will be implicit and will correspond to the system root directory.

© 2023 KNIME AG. All rights reserved.



You can also have access to network shares supported by your operating system, via the local file system within KNIME Analytics Platform.

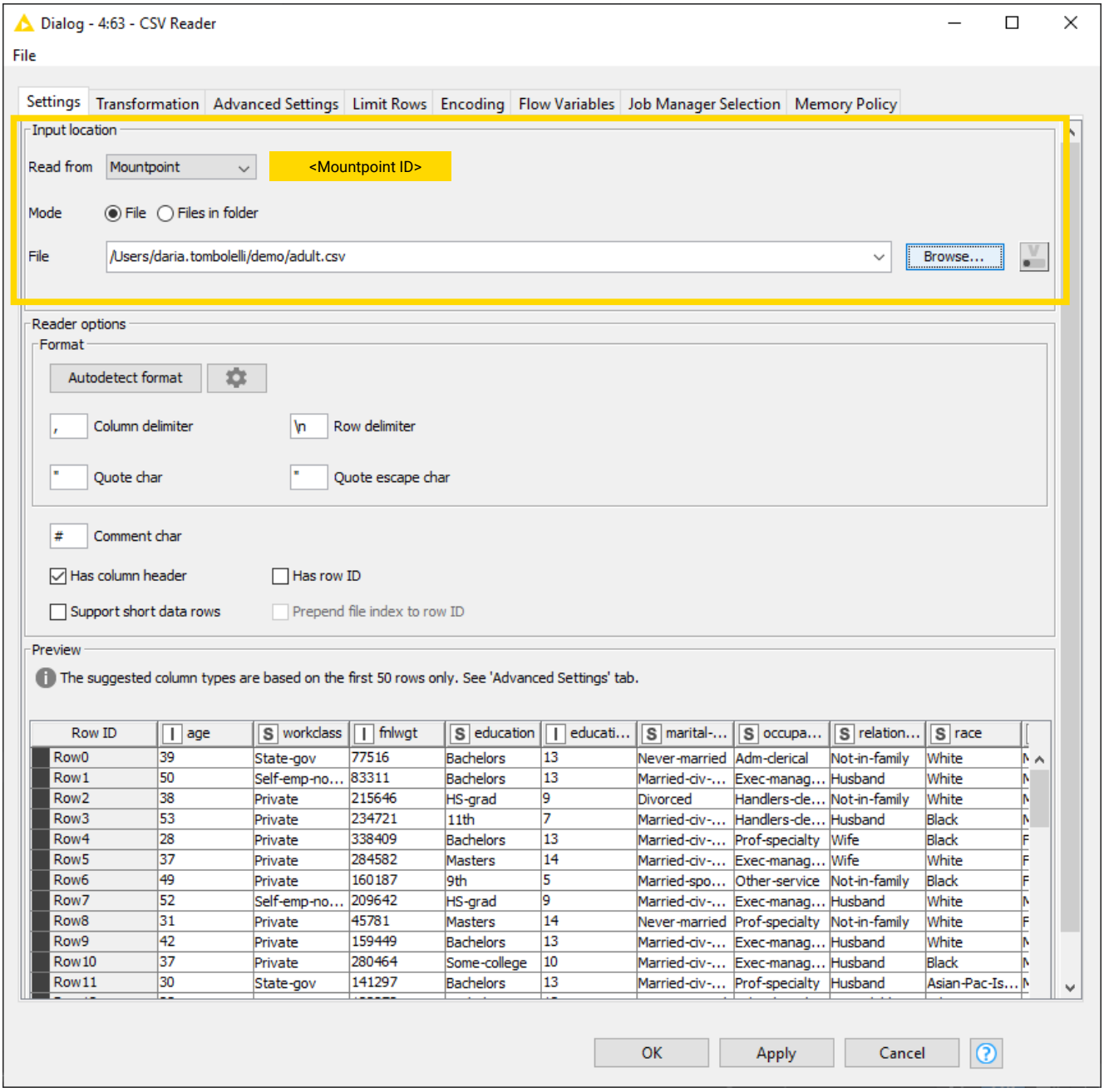
Please be aware that when executing a workflow on KNIME Server version 4.11 or higher the local file system access is disabled for security reasons. The KNIME Server administrators can activate it, but this is not recommended. For more information about this please refer to the KNIME Server Administration Guide.

© 2023 KNIME AG. All rights reserved.

Mountpoint

With the *Mountpoint* option you will have access to KNIME mount points, such as LOCAL, your KNIME Server mount points, if any, and the KNIME Hub. You have to be logged in to the specific mountpoint to have access to it.

The path syntax will be UNIX-like, i.e. `/folder1/folder2/file.txt` and relative to the implicit working directory, which corresponds to the root of the mountpoint.



Please note that workflows inside the mountpoints are treated as files, so it is not possible to read or write files inside a workflow.

© 2023 KNIME AG. All rights reserved.

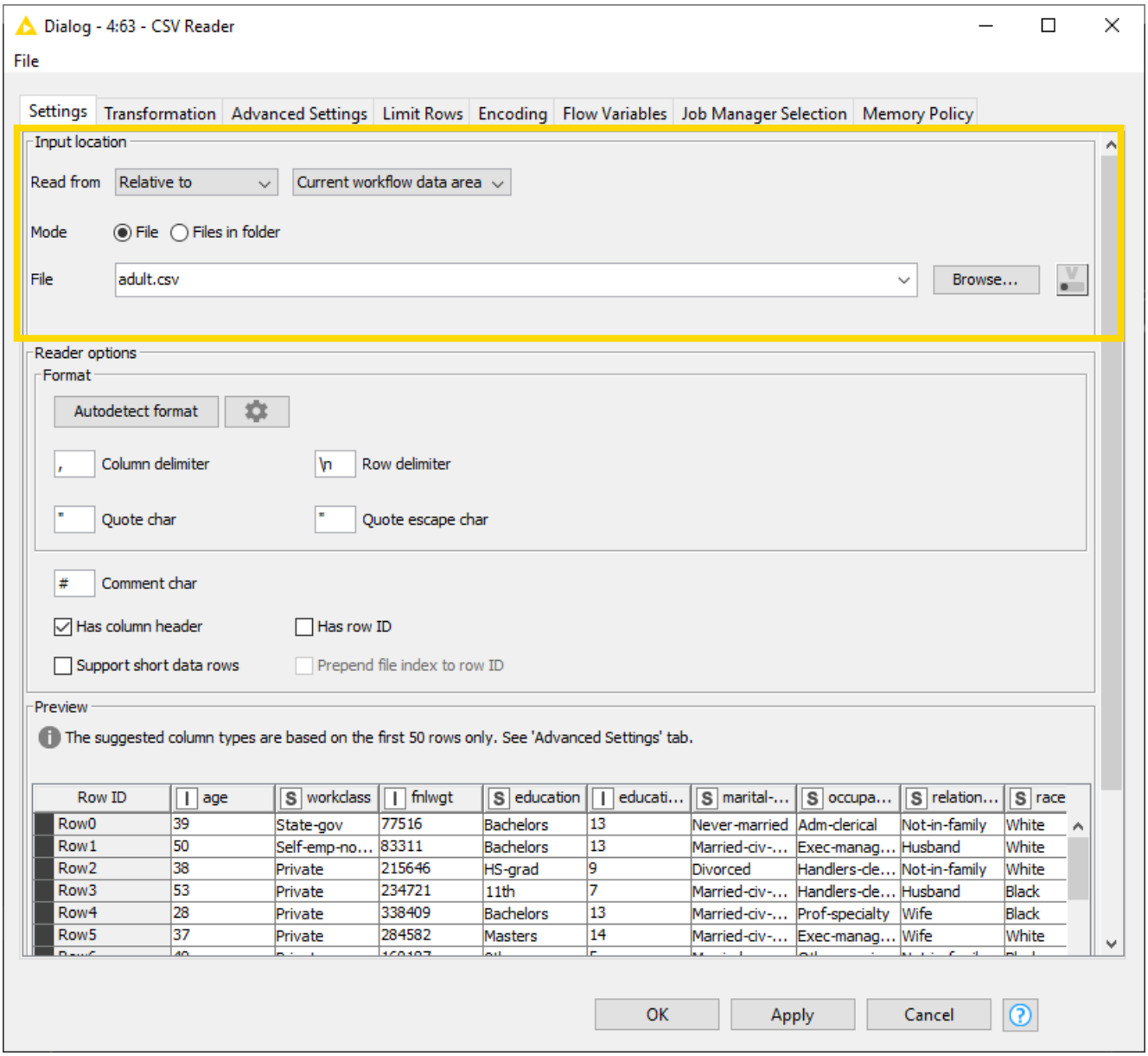
Relative to

With the *Relative to* option you will have access to three different file systems:

- *Current mountpoint* and *Current workflow*: The file system corresponds to the mountpoint where the currently executing workflow is located. The working directory is implicit and it is:
  - *Current mountpoint*: The working directory corresponds to the root of the mountpoint
  - *Current workflow*: The working directory corresponds to the path of the workflow in the mountpoint, e.g. `/workflow_group/my_workflow`.
- *Current workflow data area*: This file system is dedicated to and accessible by the currently executing workflow. Data are physically stored inside the workflow and are copied, moved or deleted together with the workflow.

All the paths used with the option *Relative to* are of the type `folder/file` and they must be relative paths.

© 2023 KNIME AG. All rights reserved.



In the example above you will read a `.csv` file from a folder `data` which is in:

```
<knime-workspace>/workflow_group/my_workflow/data/
```

Please note that workflows are treated as files, so it is not possible to read or write files inside a workflow.

When the workflow is executing on KNIME Server the options *Relative to → Current mountpoint* or *Current workflow* will access the workflow repository on the server. The option *Relative to → Current workflow data area*, instead, will access the data area of the **job copy** of the workflow. Please be aware that files written to the data area will be lost if the job is deleted.

© 2023 KNIME AG. All rights reserved.

Custom/KNIME URL

This option works with a pseudo-file system that allows to access single files via URL. It supports the following URLs:

- `knime://`
- `http(s)://` if authentication is not needed
- `ssh://` if authentication is not needed
- `ftp://` if authentication is not needed

For this option, you can also set up manually a timeout parameter (in milliseconds) for reading and writing.

The URL syntax should be as follows:

- `scheme:[//authority]path[?query][#fragment]`
- The URL must be encoded, e.g. spaces and some special characters that are reserved, as `?`. To encode the URL you can use any available online URL encoder tool.

Using this option you can read and write single files, but you would not be able to move and copy files or folders. However, listing files in a folder, i.e. browsing, is not supported.

Standard file systems connector nodes

With KNIME Analytics Platform version 4.5 new connectors nodes for standard file systems have been introduced. These nodes allow you to prototype workflows using a specific file system, by providing access to the standard file systems. The resulting output port allows downstream nodes to access files, e.g. to read or write, or to perform other file system operations, e.g. browse/list files, copy, move.

Please note that in many cases it is not necessary to use these connector nodes to access the standard file system. Nodes that require file system access (e.g. the File Reader node) typically provide standard file system access. The purpose of these connector nodes is that the working directory can be chosen, which makes file access with relative paths more convenient.

Local File System Connector node

This node provides access to the file system of the local machine.

In the Local File System node configuration dialog you can choose to use a custom working directory. If this option is not set, the default working directory is the home directory of the

© 2023 KNIME AG. All rights reserved.

current operating system.

Mountpoint Connector node

This node provides a file system connection with access to a mountpoint, for example "LOCAL", or "My-KNIME-Hub". It can also provide a file system connection with access to the mountpoint which contains the current workflow, similar to the option *Relative to > Current mountpoint*.

In the Mountpoint Connector node configuration dialog you can specify the mountpoint to access.

- *Current mountpoint*: If chosen, the file system connection will give access to the mountpoint which contains the current workflow. If you are not using this connector node, this option is equivalent to choosing *Read from > Relative to > Current mountpoint*. Selecting the option *Set working directory to the current workflow*, will additionally set the working directory of the file system connection to the location of the current workflow. This is then equivalent to choosing *Read from > Relative to > Current workflow*.
- *Other mountpoint*: If chosen, the file system connection will give access to the selected mountpoint. Unconnected mountpoints are greyed out and can still be selected, but you need to go to the KNIME Explorer and connect to the mountpoint before executing this node. A mountpoint is displayed in red if it was previously selected but is no longer available. You will not be able to save the dialog in this case.

Finally, you can specify a *Working directory* using a Path syntax.

Workflow Data Area Connector

This node provides a file system connection with access to the data area of the current workflow.

If you are not using this connector node, this option is equivalent to choosing *Read from > Relative to > Current workflow data area*.

In the Workflow Data Area Connector node configuration dialog you can specify a *Working directory* using a Path syntax.

Connected file systems

Connected file systems instead require a connector node to connect to the specific file system. In the connector nodes configuration dialog it is possible to configure the most convenient working directory.

© 2023 KNIME AG. All rights reserved.

The file system Connector nodes that are available in KNIME Analytics Platform can be divided into two main categories:

Connector nodes that need an Authenticator node:

- Amazon S3 Connector node with Amazon Authenticator node
- Google Cloud Storage Connector node with Google Authenticator node
- Google Drive Connector node with Google Authenticator node
- SharePoint Online Connector node with Microsoft Authenticator node
- Azure Blob Storage Connector node with Microsoft Authenticator node
- Azure Data Lake Storage Gen2 node with Microsoft Authenticator node

Connector nodes that do not need an Authenticator node:

- Databricks File System Connector node
- HDFS Connector node
- HDFS Connector (KNOX) node
- Create Local Bigdata Environment node
- SSH Connector node
- HTTP(S) Connector node
- FTP Connector node
- Space Connector node
- KNIME Server Connector node

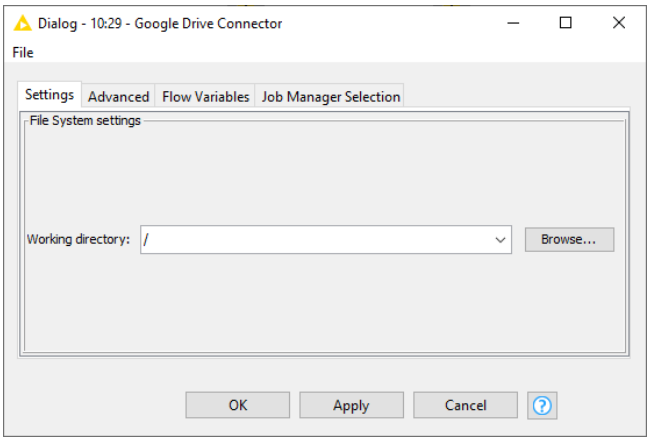
File systems with external Authenticator

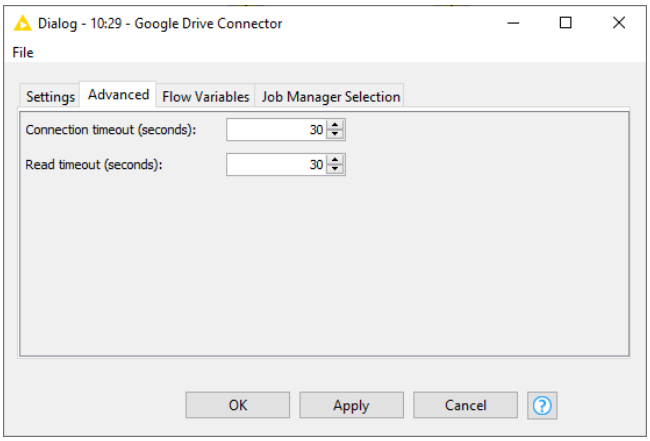
The path syntax varies according to the connected file system, but in most cases it will be UNIX-like. Information on this are indicated in the respective Connector node descriptions.

Typically in the configuration dialog of the Connector node you will be able to:

- Set up the working directory: In the *Settings* tab type the path of the working directory or browse through the file system to set up one.
- Set up the timeouts: In the *Advanced* tab set up the connection timeout (in seconds) and the Read timeout (in seconds).

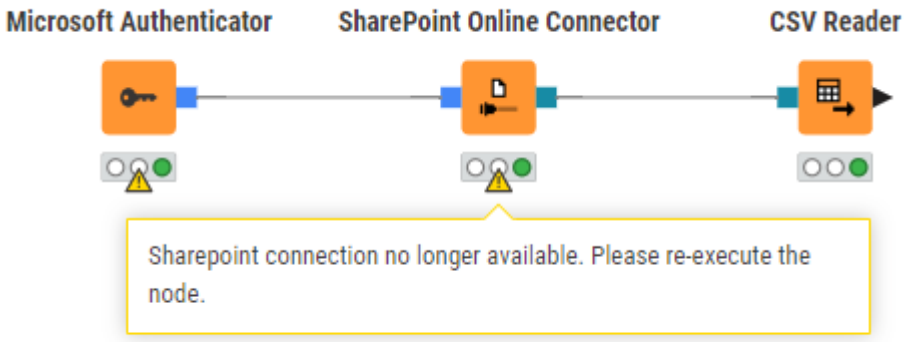
© 2023 KNIME AG. All rights reserved.





Most connectors will require a network connection to the respective remote service. The connection is then opened when the Connector node executes and closed when the Connector node is reset or the workflow is closed.

It is important to note that the connections are not automatically re-established when loading an already executed workflow. To connect to the remote service you will then need to execute again the Connector node.



Amazon file system

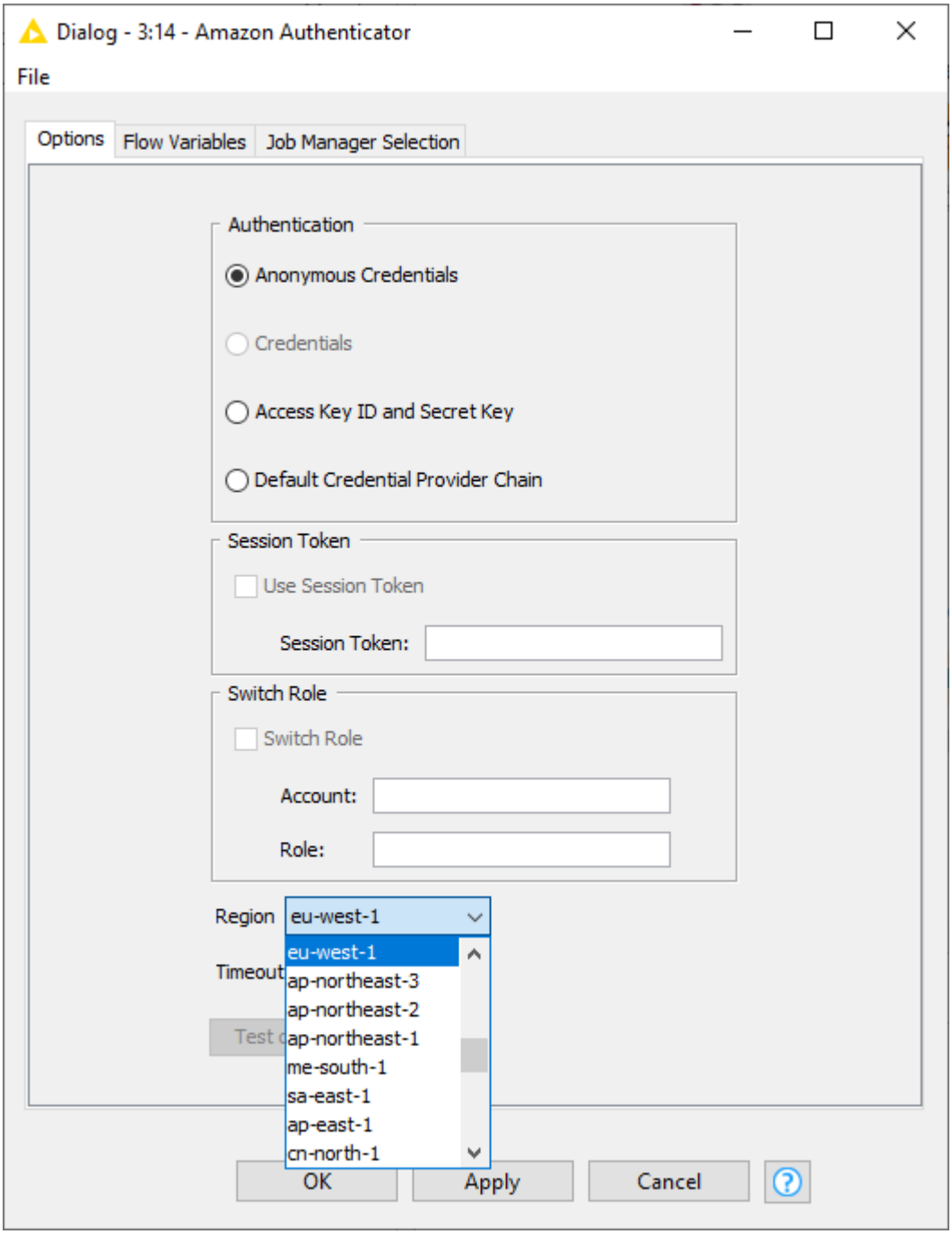
To connect to Amazon S3 file system you will need to use:

- Amazon Authenticator node
- Amazon S3 Connector node

© 2023 KNIME AG. All rights reserved.

Amazon S3 file system normalizes paths. Amazon S3 allows paths such as `/mybucket/.././file`, where `".."` and `"."` must not be removed during path normalization because they are part of the name of the Amazon S3 object. When such a case is present you will need to uncheck *Normalize paths* option from the Amazon S3 Connector node configuration dialog.

Please be aware that each bucket in Amazon S3 belongs to an AWS region, e.g. eu-west-1. To access the bucket the client needs to be connected to the same region. You can select the region to connect to in the Amazon Authenticator node configuration dialog.



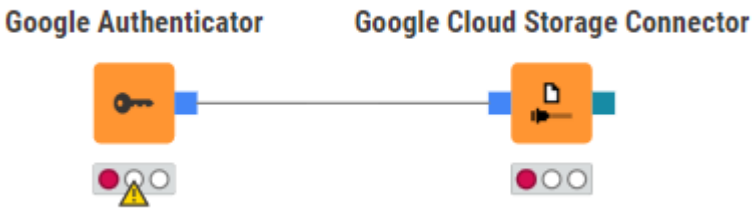
© 2023 KNIME AG. All rights reserved.

Google file systems

We support two file systems related to Google. However, even though they both belong to Google services the corresponding Connector nodes use a different authentication type, and hence Authenticator node.

To connect to Google Cloud Storage you will need to use:

- Google Cloud Storage Connector node
- Google Authenticator node



Also the Google Cloud Storage Connector node, as the Amazon S3 Connector node, normalizes the paths.

To use the Google Authenticator node you will need to create a project at console.developers.google.com.

The specific Google API you want to use has to be enabled under `APIs`.

After you create your Service Account you will receive a p12 key file to which you will need to point to in the Google Authenticator node configuration dialog.

To connect to Google Drive, instead, you will need to use:

- Google Drive Connector node
- Google Authenticator node

The root folder of the Google Drive file system, contains your `Shared drivers`, in case any is

© 2023 KNIME AG. All rights reserved.

available, and the folder `My Drive`. The path of your `Shared drivers` will then be `/shared_driver1/`, while the path of your folder `My Drive` will be `/My Drive/`.

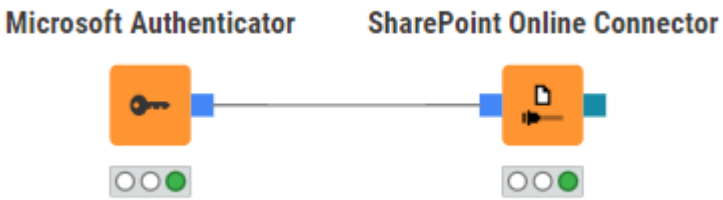
Microsoft file systems

We support three file systems related to Microsoft.

To connect to SharePoint Online, Azure Blob Storage, or to Azure Data Lake Storage Gen2, you will need to use:

- SharePoint Online Connector node, or Azure Blob Storage Connector node, or Azure ADLS Gen2 Connector node
- Microsoft Authenticator node

The SharePoint Online Connector node connects to a SharePoint Online site. Here document libraries are represented as top-level folders.



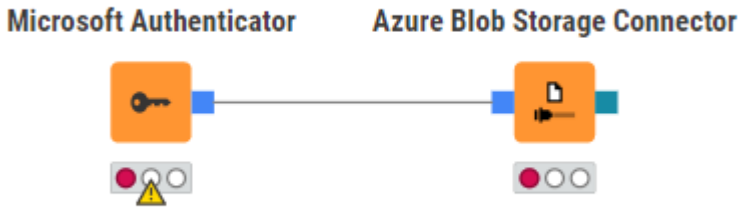
In the node configuration dialog you can choose to connect to the following sites:

- *Root Site*: Root site of the your organization
- *Web URL*: https URL of the SharePoint site (same as in the browser)
- *Group site*: Group site of a Office365 user group
- *Subsite*: Connects to subsite or sub-sub-site of the above

The Azure Blob Storage Connector node connects to an Azure Blob Storage file system.

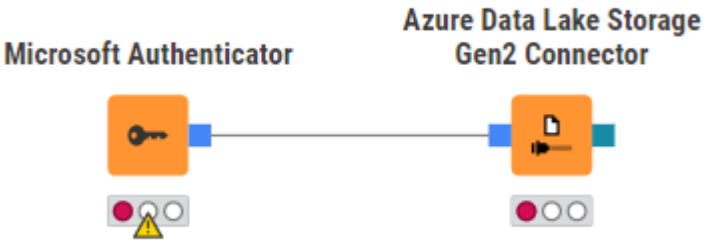
The path syntax will be UNIX-like, i.e. `/mycontainer/myfolder/myfile` and relative to the root of the storage. Also Azure Blob Storage Connector node performs paths normalization.

© 2023 KNIME AG. All rights reserved.



The Azure ADLS Gen2 Connector node connects to an Azure Blob Storage file system.

The path syntax will be UNIX-like, i.e. `/mycontainer/myfolder/myfile` and relative to the root of the storage.



The Microsoft Authenticator node offers OAuth authentication for Azure and Office365 clouds.

It supports the following authentication modes:

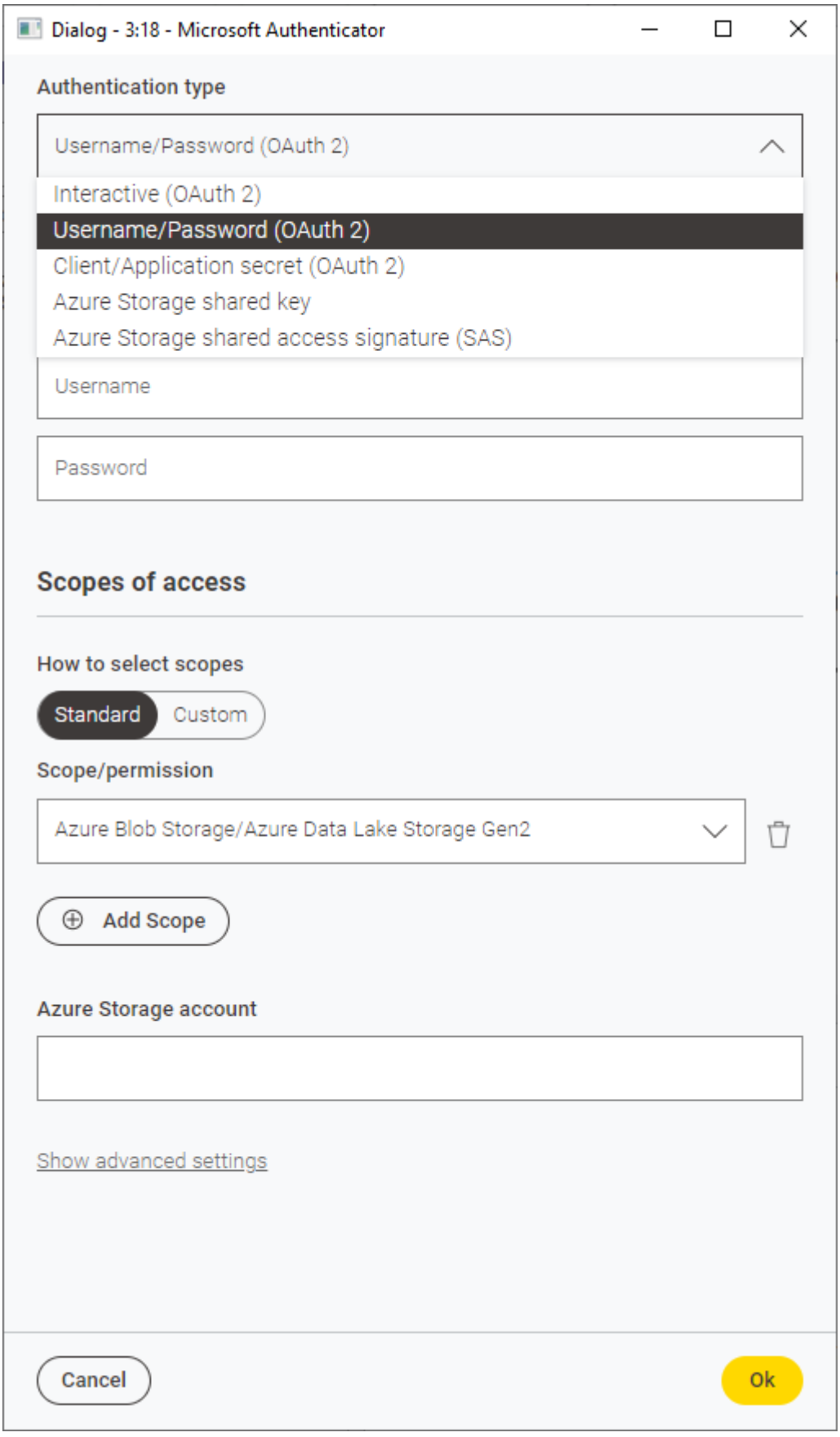
- *Interactive authentication*: Performs an interactive, web browser based login by clicking on *Login* in the node dialog. In the browser window that pops up, you may be asked to consent to the requested level of access. The login results in a token being stored in a configurable location. The token will be valid for a certain amount of time that is defined by your Microsoft Entra ID settings.
- *Username/password authentication*: Performs a non-interactive login to obtain a fresh token every time the node executes. Since this login is non-interactive and you get a fresh token every time, this mode is well-suited for workflows on KNIME Server. However, it also has some limitations. First, you cannot consent to the requested level of access, hence consent must begiven beforehand, e.g. during a previous interactive login, or by an Microsoft Entra ID directory admin. Second, accounts that require multi-factor authentication (MFA) will not work.
- *Shared key authentication (Azure Storage only)*: Specific to Azure Blob Storage and Azure Data Lake Storage Gen2. Performs authentication using an Azure storage

© 2023 KNIME AG. All rights reserved.

account and its secret key.

- *Shared access signature (SAS) authentication (Azure Storage only)*: Specific to Azure Blob Storage and Azure Data Lake Storage Gen2. Performs authentication using shared access signature (SAS). For more details on shared access signatures see the Azure storage documentation.

© 2023 KNIME AG. All rights reserved.



© 2023 KNIME AG. All rights reserved.

File systems without external Authenticator

All the Connector nodes that do not need an external Authenticator node will connect upon execution to a specific file system. This allows the downstream nodes to access the files of the remote server or file system.

Space Connector node

The Space Connector node connects to a KNIME Hub instance. It provides a file system connection with access to a KNIME Hub space.

You can either use it without a preceeding KNIME Hub Authenticator node or you can use the Authenticator node.

If you are not using a KNIME Hub Authenticator node:

- If your workflow is located in a local space you need to first connect to the KNIME Hub from the Analytics Platform.
- If your workflow is located in a space on a KNIME Hub instance it is also possible to always connect to the space that contains the current workflow.

Alternatively, you can use a KNIME Hub Authenticator node, by adding the node and connecting it to a KNIME Hub instance. Then you can connect the Authenticator output port to the Space Connector node. To do so you can either add a connection port by clicking the + icon that appears when hovering over the Space Connector node or by dragging and dropping the output port of the Authenticator node into the Space Connector node. The port will be added and the two nodes connected. To authenticate against a KNIME Hub instance by using the KNIME Hub Authenticator node you will need to provide the URL and an application id/password. To create an application password please follow the instructions on the KNIME Hub User Guide.

In any case, the resulting output port allows downstream nodes to access files, e.g. to read or write, or to perform other file system operations (browse/list files, copy, move, …) in the selected Space. Files outside the configured space cannot be accessed.

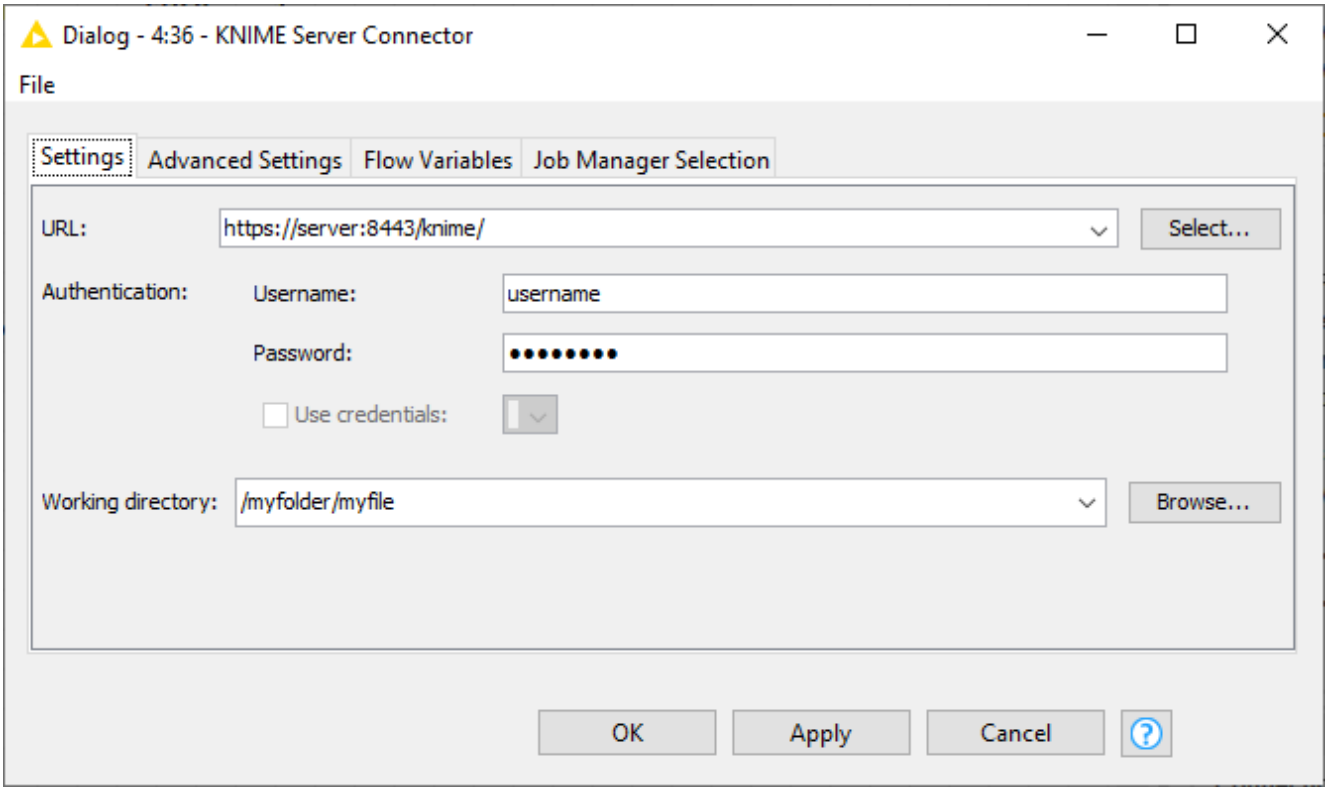
KNIME Server Connector node

With the KNIME Server Connector node you are able to connect to a KNIME Server instance.

When opening the KNIME Server Connector node configuration dialog you will be able to either type in the URL of the KNIME Server you want to connect to, or to *Select…* it from those available among your mountpoints in KNIME Explorer. You do not need to have the KNIME

© 2023 KNIME AG. All rights reserved.

Server mountpoint set up or already connected to use this node.



You can authenticate either by typing in your username and password or by using the selected credentials provided by flow variable, if any is available. Please be aware that when authenticating typing in your username and password the password will be persistently stored in an encrypted form in the settings of the node and will be then saved with the workflow.

SMB Connector node

With SMB Connector node you can connect to a remote SMB server (e.g. Samba, or Windows Server). The resulting output port allows downstream nodes to access the files in the connected file system.

This node generally supports versions 2 and 3 of the SMB protocol. It also supports connecting to a Windows DFS namespace.

When opening the SMB Connector node configuration dialog you can choose to connect to a *File server* host or a Windows *Domain*. Choosing *File server* specifies that a direct connection shall be made to access a file share on a specific file server. A file server is any machine that runs an SMB service, such as those provided by Windows Server and Samba.

Choosing *Domain* specifies that a connection shall be made to access a file share in a Windows Active Directory domain.

© 2023 KNIME AG. All rights reserved.

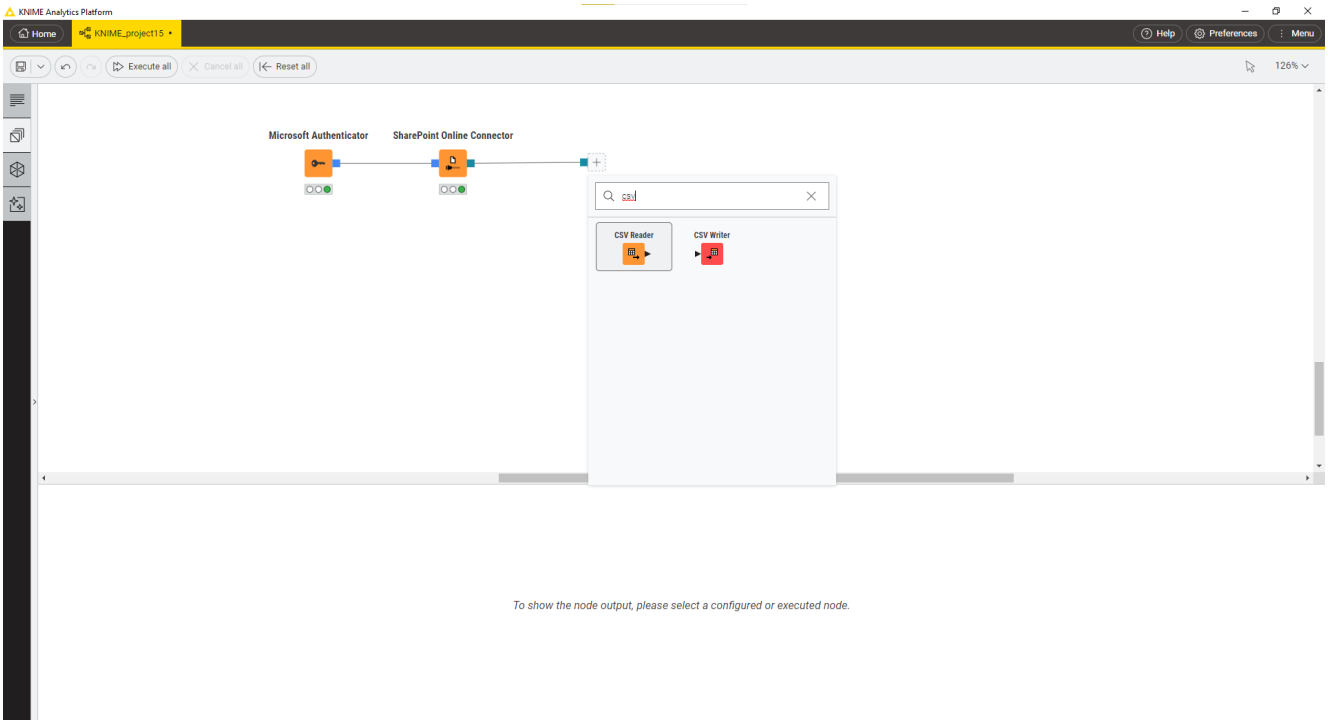
Read and write from or to a connected file system

When you successfully connect to a connected file system you will be able to connect the output port of the Connector node to any node that is developed under the File Handling framework.

To do this you will need to activate the corresponding dynamic port of your node.

You can also enable dynamic ports to connect to a connected file system in Utility nodes.

To add a port to one of the nodes where this option is available, drag and drop the output port of the Connector node onto the workflow canvas and select the node you want to add from the panel that opens. This will automatically add the node with the port and connect it.



Reader nodes

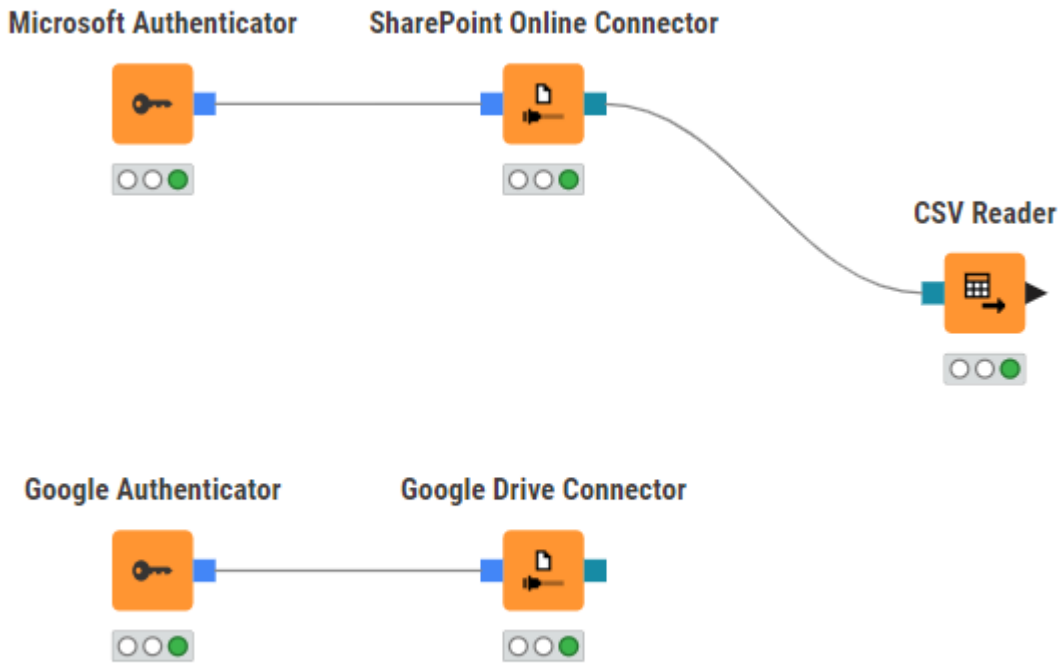
A number of Reader nodes in KNIME Analytics Platform are updated to work within the File Handling framework.

See the How to distinguish between the old and new file handling nodes section to learn how to identify the reader nodes that are compatible with the new File Handling framework.

© 2023 KNIME AG. All rights reserved.

You can use these Reader nodes with both Standard file systems and Connected file systems. Moreover, using the File System Connection port you can easily switch the connection between different connected file systems.

In the following example a CSV Reader node with a File System Connection port is connected to a SharePoint Online Connector node and it is able to read a `.csv` file from a SharePoint connected file system. Exchange the connection with any other Connector nodes, i.e. Google Drive, to read a `.csv` file from the other file system.

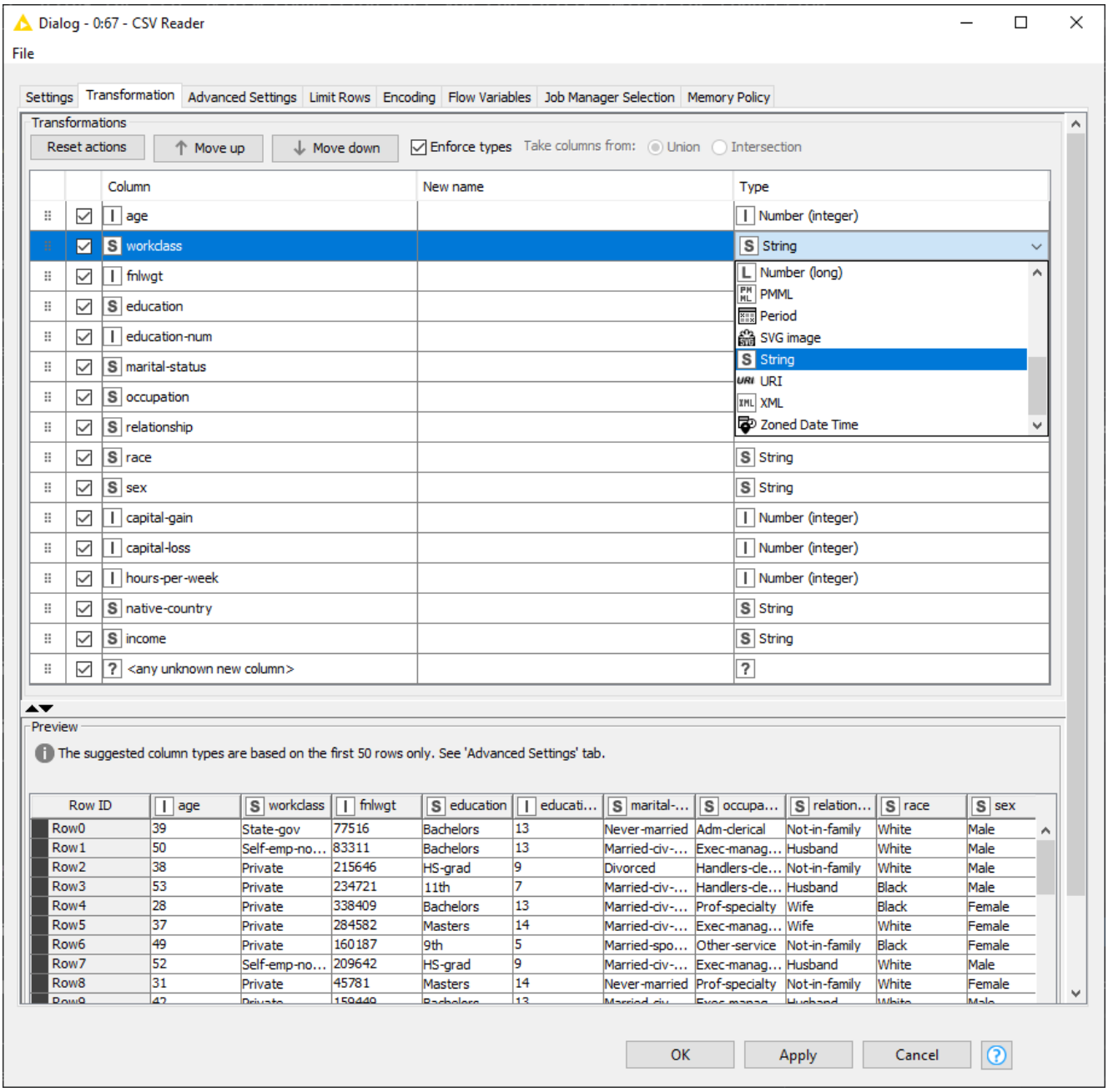


Transformation tab

The new table based reader nodes e.g. Excel Reader or CSV Reader allow you to transform the data while reading it via the *Transformation* tab. To do so the node analyzes the structure of the file(s) to read in the node dialog and stores the file structure and the transformation that should be applied. When the node is executed this information is used to read the data and apply the transformations before creating the KNIME data table. This allows the nodes to return a table specification during configuration of the node and not only once the node is executed, assuming that the file structure does not change. The benefits of this is that you can configure downstream nodes without executing the reader node first and improves execution speed of the node.

© 2023 KNIME AG. All rights reserved.

To speedup the file analysis in the node dialog by default the reader node only reads a limited amount of rows which might lead to mismatching type exceptions during execution. To increase the limit or disable it completely open the *Advanced Settings* tab and go to the *Table specification* section.



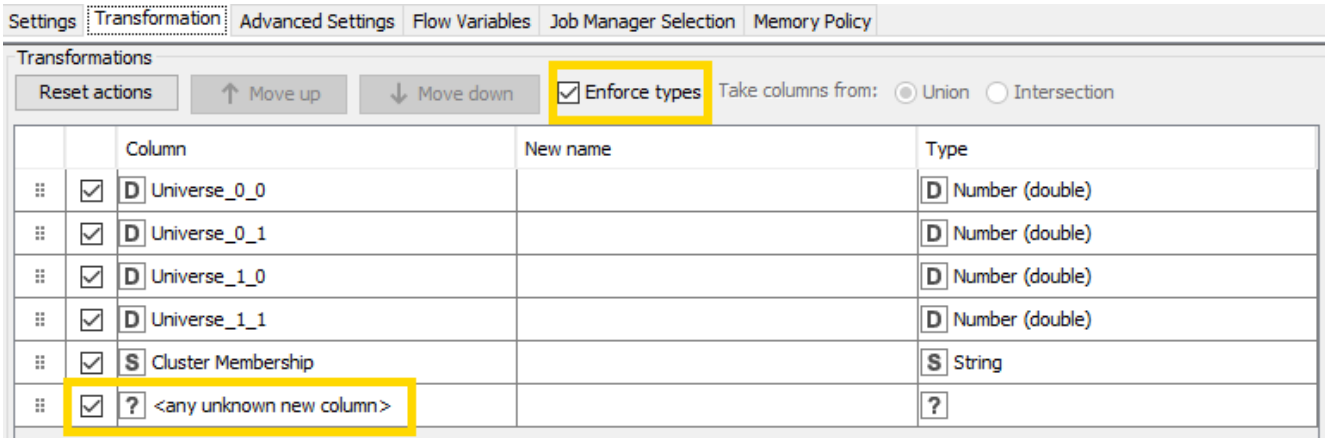
To change the transformation in the updated reader nodes configuration dialog, go to the *Transformation* tab after selecting the desired file. This tab displays every column as a row in a table that allows modifying the structure of the output table. It supports reordering, filtering and renaming columns. It is also possible to change the type of the columns. Reordering is done via drag and drop. Just drag a column to the position it should have in the output table. Note that the positions of columns are reset in the dialog if a new file or folder is selected.

If you are reading multiple files from a folder, i.e. with the option *Files in folder* in the *Settings*

© 2023 KNIME AG. All rights reserved.

tab, you can also choose to take the resulting columns from *Union* or *Intersection* of the files you are reading in.

The *Transformation* tab provides two options which are important when dealing with changing file structures if you control the reader via flow variables.



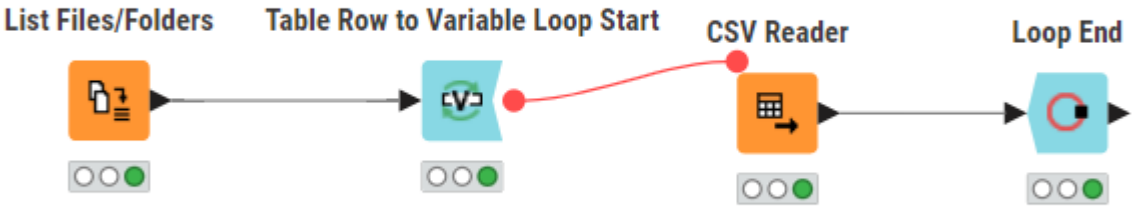
The *Enforce types* option controls how columns whose type changes are dealt with. If selected, the node attempts to map to the KNIME type you configured in the transformation tab and fails if that is not possible. If unselected, the KNIME type corresponding to the new type is used whether it is compatible to the configured type or not. This might cause downstream nodes to fail if the type of a configured column has changed.

The `<any unknown new column>` option is a place holder for all previously unknown columns and specifies whether and where these columns should be added during execution.

Controlling readers via flow variables

As described in the *Transformation* tab section, the reader assumes that the structure of the file will not change after closing the node dialog. However when a reader is controlled via a flow variable the resulting table specification can change. This happens if settings that affect which data should be read are changed via a flow variable. One example is the reading of several CSV files with different columns in a loop where the path to the file is controlled by a flow variable.

© 2023 KNIME AG. All rights reserved.



During configuration the node checks if the settings have changed. If that is the case the node will not return any table specification during configuration and will analyze the new file structure with each execution. In doing so it will apply the transformations to the new discovered file structure using the `<any unknown new column>` and *Enforce types* option.

> Currently the new reader nodes only detect changes of the file path. Other settings via flow variables that might influence the structure such as a different sheet name in the *Excel Reader* or a different column separator in the *CSV Reader* are not monitored with the current version. We are aware of this limitation and plan to change this with KNIME Analytics Platform 4.4. Until then you need to select the *Use new schema* option which is described in the next section if you encounter this problem.

Reading the same data files with changing structure

As described in the *Transformation* tab section, the reader assumes that the structure of the file will not change after closing the node dialog. However the structure of the file can change (e.g. by adding additional columns) if it gets overwritten. In this case you need to select the *Use new schema* option from the *When schema in file has changed* setting on the *Advanced* tab. Enabling this option forces the reader to compute the table specification during each execution thus adapting to the changes of the file structure.

> The *Use new schema* option will disable the *Transformation* tab and the node will not return any table specification during configuration.

> The option *Ignore (deprecated)* is enabled only for the nodes created with KNIME Analytics Platform versions < 5.2.2, if the *Support changing schemas* option was not selected. In this case the node did ignore a changed schema which might have resulted in wrong data.

© 2023 KNIME AG. All rights reserved.

Append a path column

In the *Advanced Settings* tab you can also check the option *Append path column* under *Path column* pane. If this option is checked, the node will append a path column with the provided name to the output table. This column will contain for each row which file it was read from. The node will fail if adding the column with the provided name causes a name collision with any of the columns in the read table. This will allow you to then distinguish the file from which a specific row is read from in case you are reading multiple files and are concatenating them into a single data table.

Writer nodes

Also Writer nodes can be used in KNIME Analytics Platform to work within the File Handling Framework.

You can use these Writer nodes with both Standard file systems and Connected file systems. Moreover, using the File System Connection port you can easily switch the connection between different connected file systems.

An output File System Connection port can be added to Writer nodes and this will allow them to be easily connected to different file systems and will be able to write persistently files to them.

In the following example a CSV Writer node with a File System Connection port is connected to a Google Drive Connector node and it is able to write a `.csv` file to a Google Drive connected file system. A CSV Reader node reads a `.csv` file from a SharePoint Online File System, data is transformed, and the resulting data is written to the Google Drive file system.

© 2023 KNIME AG. All rights reserved.

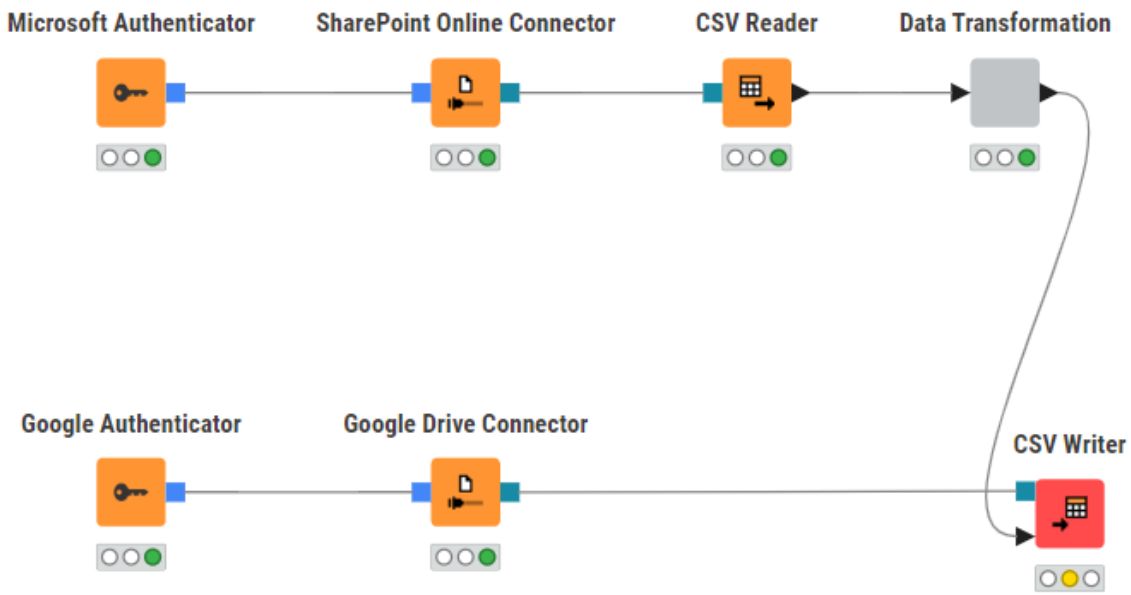


Image Writer (Table) node

The Image Writer (Table) node is also available with a table based input.

You can give to this node an input data table where a column contains, in each data cell, images. Image Writer (Table) is able to process the images contained in a column of the input data table and write them as files in a specified output location.

© 2023 KNIME AG. All rights reserved.

Path data cell and flow variable

Files and folders can be uniquely identified via their path within a file system. Within KNIME Analytics Platform such a path is represented via a path type. A path type consists of three parts:

1. **Type:** Specifies the file system type e.g. local, relative, mountpoint, custome_url or connected.
2. **Specifier:** Optional string that contains additional file system specific information such as the location the relative to file system is working with such as workflow or mountpoint.
3. **Path:** Specifies the location within the file system with the file system specific notation e.g. `C:\file.csv` on Windows operating systems or `/user/home/file.csv` on Linux operating systems.

Path examples are:

- Local
  - `(LOCAL, , C:\Users\username\Desktop)`
  - `(LOCAL, , \\fileserver\file1.csv)`
  - `(LOCAL, , /home/user)`
- RELATIVE
  - `(RELATIVE, knime.workflow, file1.csv)`
  - `(RELATIVE, knime.mountpoint, file1.csv)`
- MOUNTPOINT
  - `(MOUNTPOINT, MOUNTPOINT_NAME, /path/to/file1.csv)`
- CUSTOM_URL
  - `(CUSTOM_URL, , https://server:443/my%20example?query=value#frag)`
  - `(CUSTOM_URL, , knime://knime.workflow/file%201.csv)`
- CONNECTED
  - `(CONNECTED, amazon-s3:eu-west-1, /mybucket/file1.csv)`
  - `(CONNECTED, microsoft-sharepoint, /myfolder/file1.csv)`
  - `(CONNECTED, ftp:server:port, /home/user/file1.csv)`
  - `(CONNECTED, ssh:server:port, /home/user/mybucket/file1.csv)`

© 2023 KNIME AG. All rights reserved.

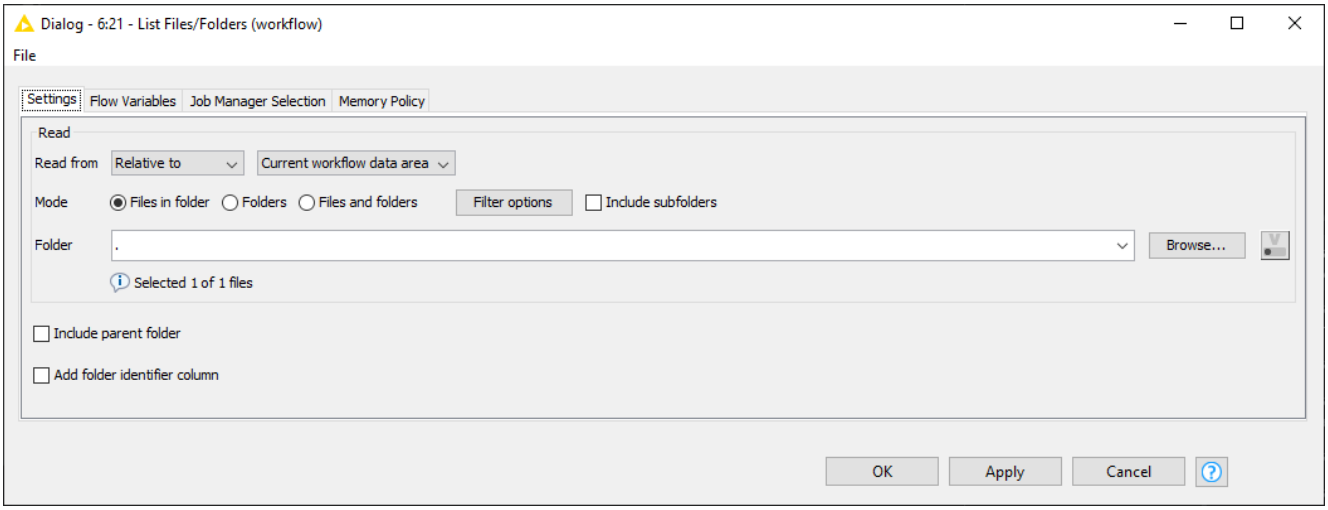
    - `(CONNECTED, http:server:port, /file.asp?key=value)`

A path type can be packaged into either a Path Data Cell or a Path Flow Variable. By default the Path Data Cell within a KNIME data table only displays the path part. If you want to display the full path you can change the cell renderer via the context menu of the table header to the *Extended path* renderer.

Creating path data cells

In order to work with files and folders in KNIME Analytics Platform you can either select them manually via the node configuration dialog or you might want to list the paths of specific files and/or folder. To do this you can use the *List Files/Folders* node. Simply open the dialog and point it to the folder whose content you want to list. The node provides the following options:

- *Files in folder*: Will return a list of all files within the selected folder that match the *Filter options*.
- *Folders*: Will return all folders that have the selected folder as parent. To include all sub folders you have to select the *Include subfolders* option.
- *Files and folders*: Is a combination of the previous two options and will return all files and folders within the selected folder.



Manipulating path data cells

Since KNIME Analytics Platform version 4.4 you can also use the Column Expressions node which supports now the Path type data cells. This node provides the possibility to append an arbitrary number of columns or modify existing columns using expressions.

© 2023 KNIME AG. All rights reserved.

Creating path flow variables

There are two ways to create a path flow variable. The first way is to export it via the dialog of the node where you specify the path. This could be a *CSV Writer* node for example where you want to export the path to the written file in order to consume it in a subsequent node. The second way is to convert a path data cell into a flow variable by using one of the available variable nodes such as the *Table Row to Variable* or *Table Row to Variable Loop Start* node.

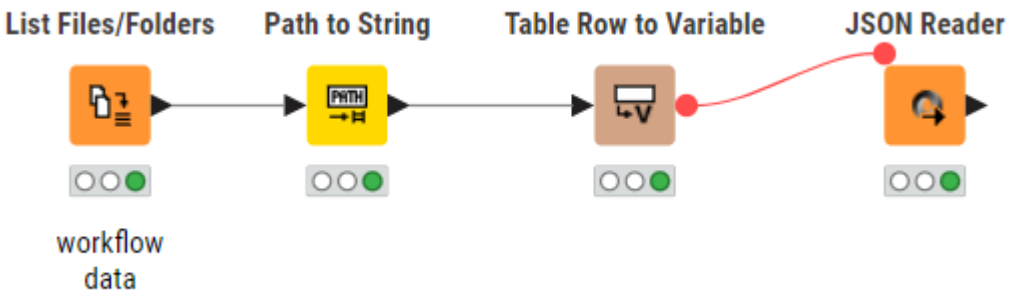
String and path type conversion

Until now not all nodes that work with files have been converted to the new file handling framework and thus do not support the path type. These nodes require either a String or URI data cell or a string flow variable.

From path to string

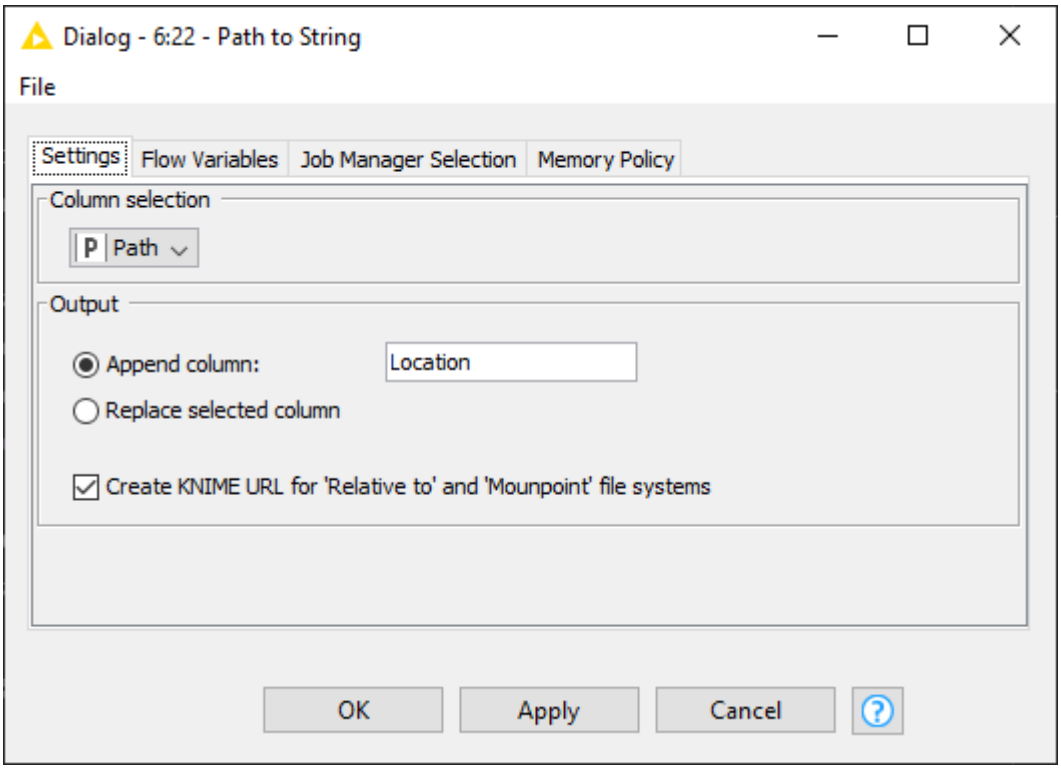
The *Path to String* node converts a path data cell to a string data cell. By default the node will create a String representation of the path that can be used in a subsequent node that still requires the old string or URI type, e.g. *JSON Reader*.

You can download this example workflow from KNIME Hub.



If you only want to extract the plain path you can disable the *_Create KNIME URL for 'Relative to' and 'Mountpoint' file system* option in the Path to String node configuration dialog.

© 2023 KNIME AG. All rights reserved.



Similar to the *Path to String* node, the *Path to String (Variable)* node converts the selected path flow variables to a string variable.

If you want to use a node that requires a URI cell you can use the *String to URI* node after the *Path to String* node.

From string to path

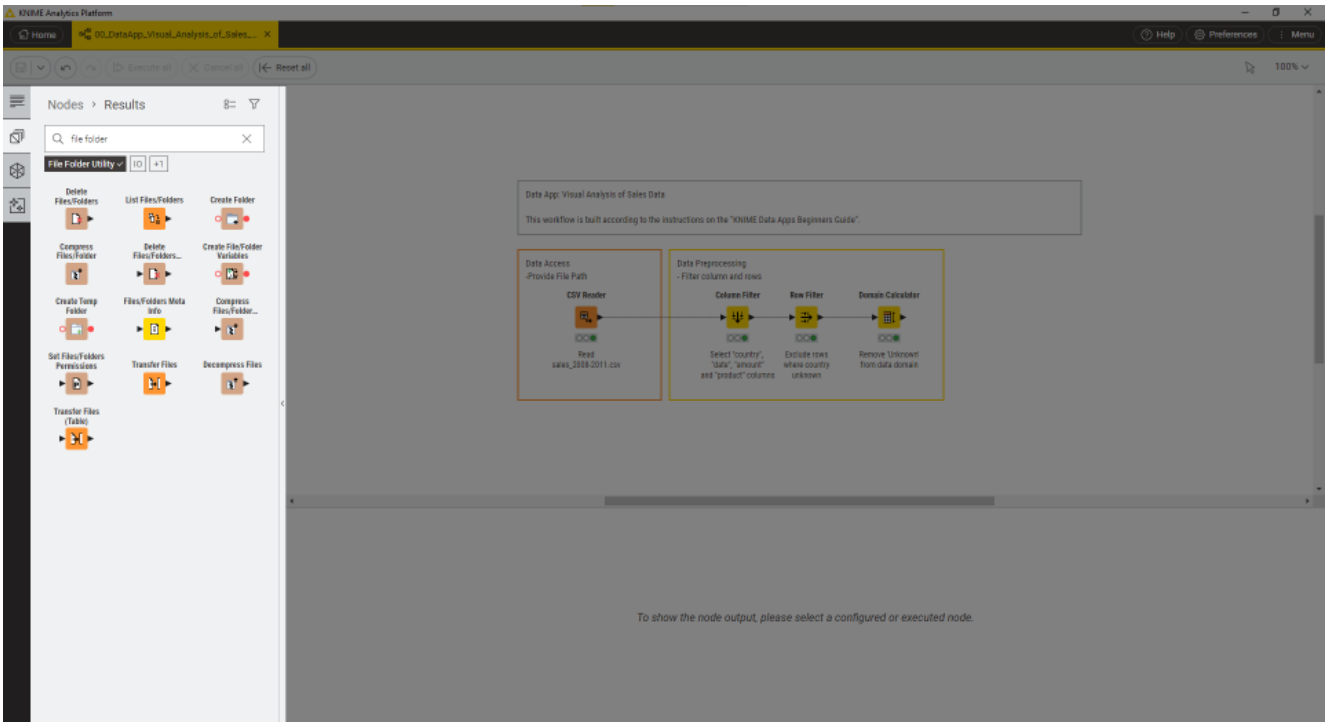
In order to convert a string path to the path type you can use the *String to Path* node. The node has a dynamic File System port that you need to connect to the corresponding file system if you want to create a path for a connected file system such as Amazon S3.

Similar to the *String to Path* node the *String to Path (Variable)* node converts a string flow variable into a path flow variable.

© 2023 KNIME AG. All rights reserved.

File Folder Utility nodes

With the introduction of the new file handling framework with KNIME Analytics Platform release 4.3, we also updated the functionality of the utility nodes.



You can add a dynamic port to connect to a connected file system directly with the File Folder Utility nodes. In this way you can easily work with files and folders in any file system that is available.

In the example below the Transfer Files node is used to connect to two file systems a source file system, Google Drive in this example, and a destination file system, SharePoint Online in this example, to easily transfer files from Google Drive to SharePoint Online.

© 2023 KNIME AG. All rights reserved.

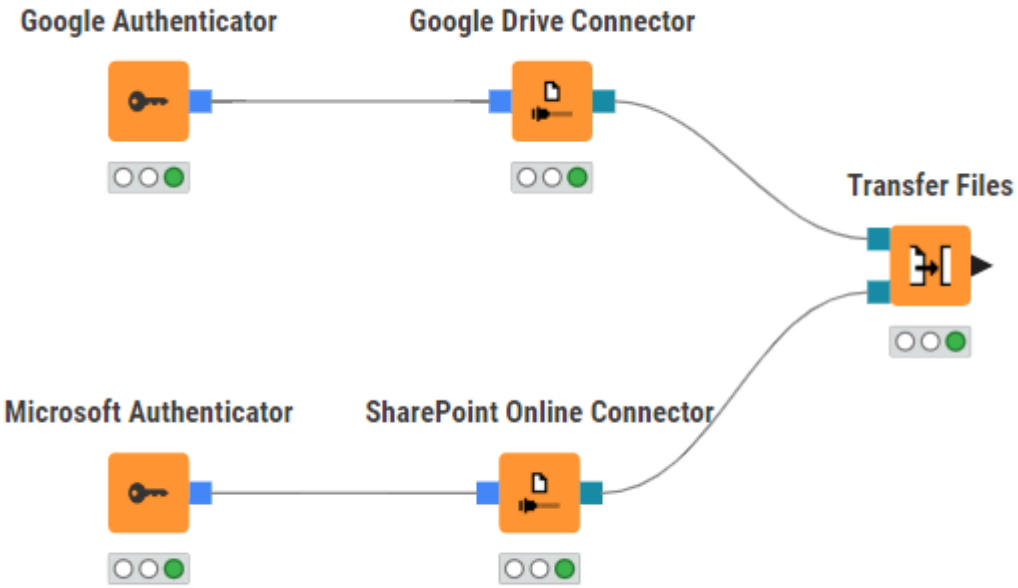


Table based input Utility nodes

Some of the File Folder Utility nodes are also available with a table based input. Those are the ones that have *(Table)* in their name, e.g. Compress Files/Folder (Table), Delete Files/Folders (Table), and Transfer Files (Table) nodes. You can give to these nodes an input data table where a column contains, in each data cell, Paths to files and these are processed by the File Folder Utility (Table) nodes.

For example, with Compress Files/Folder (Table) and Delete Files/Folder (Table) nodes you can respectively compress or delete the files whose Paths are listed in a column of the input data table. With Transfer Files (Table) you can transfer all the files whose Paths are listed in a column of the input data table.

Since the new utility nodes (e.g. *Compress Files/Folder*, *Delete Files/Folders* and *Transfer Files*) do not provide the full functionality of the old nodes yet, as of now the nodes, including the required connector nodes for the different file systems, are still available but marked as *(legacy)*. You can find them in the node repository under the *File Handling (legacy)* category. They will be deprecated once the new nodes have the full functionality. If you are missing some functionality in the new nodes please let us know via the KNIME Forum.

© 2023 KNIME AG. All rights reserved.

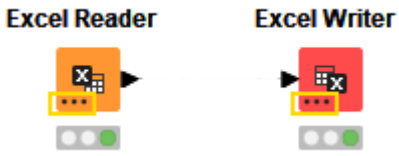
Compatibility and migration

With the 4.3 release of the KNIME Analytics Platform we introduced the new file handling framework. The framework requires a rewrite of the existing file reader and writer nodes as well as the utility nodes. However not all nodes provided by KNIME or the Community have been migrated yet which is why we provide this section to help you to work with old and new file handling nodes side by side.

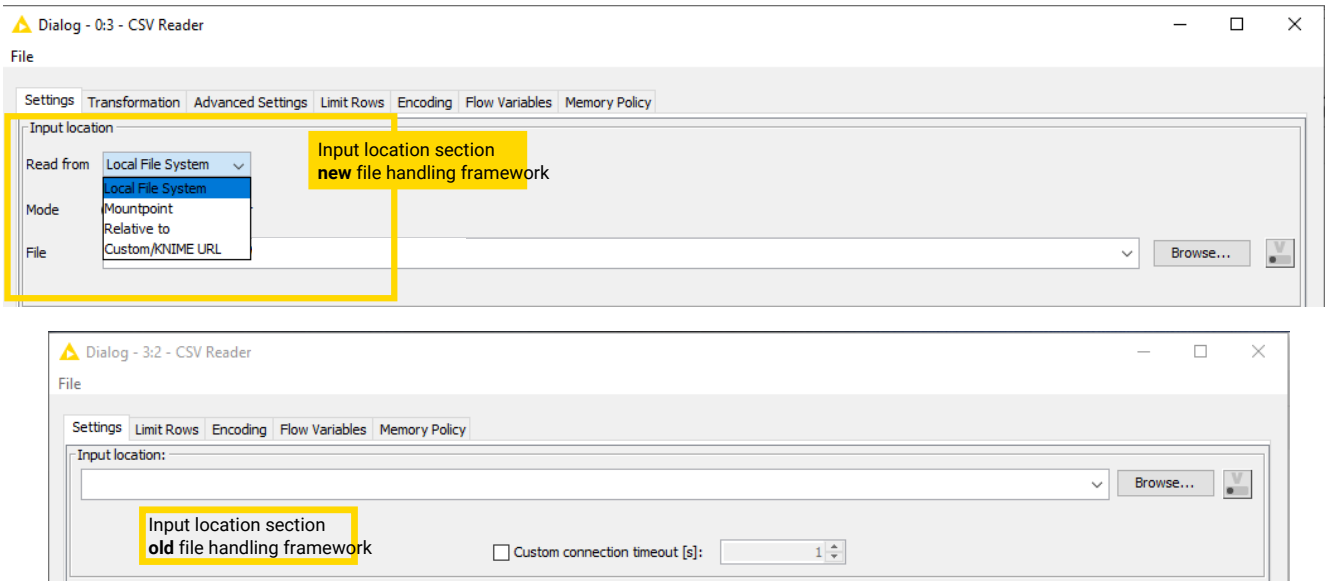
How to work with workflows that contain both old and new file handling nodes

How to distinguish between the old and new file handling nodes

You can identify the new file handling nodes by the three dots for the dynamic port on the node icon that allow you to connect to a connected file system.



Another way to identify a new file handling node is by the *Input location* section in the node configuration dialog. Here, for nodes that have been migrated to the new file handling framework you will typically find a drop-down menu that allows you to specify the file system that should be used. The following image shows the *Input location* section of the configuration dialog of a typical reader node in the new and old file handling framework.

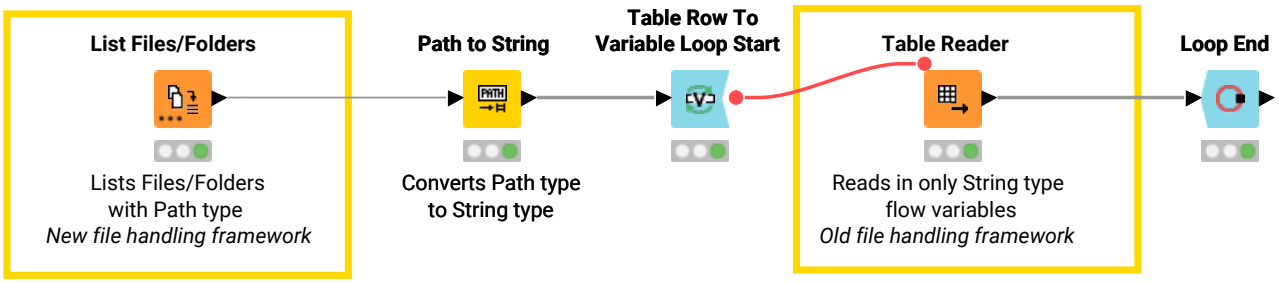


© 2023 KNIME AG. All rights reserved.

Working with old and new flow variables and data types

Using flow variables you can specify the location of a file to read or write automatically. Because not all nodes provided by KNIME or the Community have been migrated yet you might face the problem that the newer file nodes support the new Path type whereas older nodes still support a string flow variable which can be either a file path or a URI.

The List Files/Folders node returns a table with a list of files/folders in the new Path type. To use the Path type in a node that has not been migrated yet (e.g. in the Table Reader), you need to first convert the new Path type using the Path to String node into a string.



Some older nodes require a URI as input. To convert the Path type to an URI you first need to convert it to a string using the Path to String node and then convert the string to an URI using the String to URI node.

If you already have a Path type flow variable you can use the Path to String (Variable) node to convert the Path type flow variable into a string flow variable.

If you want to convert an existing file path or string to the new Path type you can either us the String to Path node to convert data columns or the String to Path (Variable) node to convert string variable.

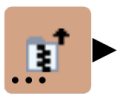
How to migrate your workflows from old to new file handling nodes

This section should help you to migrate your workflows from the old file handling framework to the new file handling framework. The new framework provides lots of advantages over the old framework which come at the cost of some changes in the usage which we will address here.

In order to standardize the node names and better capture the functionality of the actual node we have renamed some of the nodes. In addition we have also removed some nodes whose functionality has been integrated into another node such as the *Excel Writer* which replaces the original *Excel Writer (XLS)* as well as the *Excel Sheet Appender* node.

Table 1. Old (<4.2) and new (4.3+) file handling framework nodes conversion table

© 2023 KNIME AG. All rights reserved.

Node Icon	4.2 Old		4.3 New
	Zip Files	→	• Compress Files/Folder • Compress Files/Folder (Table)
	Unzip Files	→	Decompress Files
	Create Directory	→	Create Folder
	Delete Files	→	• Delete Files/Folders • Delete Files/Folders (Table)
	• Copy/Move Files • Download • Upload	→	• Transfer Files • Transfer Files (Table)
	• List Remote Files • List Files	→	List Files/Folders
	Create Temp Dir	→	Create Temp Folder

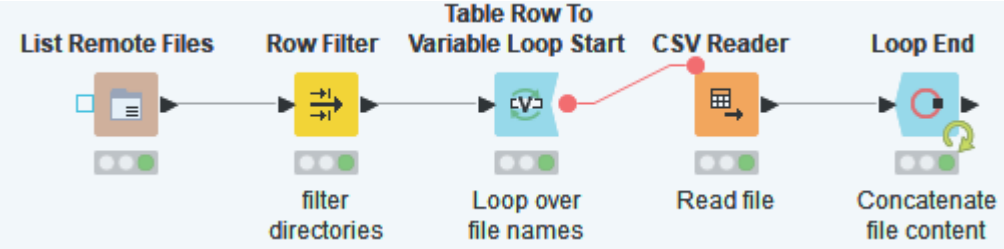
© 2023 KNIME AG. All rights reserved.

Node Icon	4.2 Old		4.3 New
	Connection (e.g. SSH Connection)	→	Connector (e.g. SSH Connector)
	Excel Reader (XLS)	→	Excel Reader
	Read Excel Sheet Names (XLS)	→	Read Excel Sheet Names
	• Excel Writer (XLS) • Excel Sheet Appender	→	Excel Writer
	File Reader	→	• File Reader • File Reader (Complex Format)

Reading multiple files into a single KNIME data table

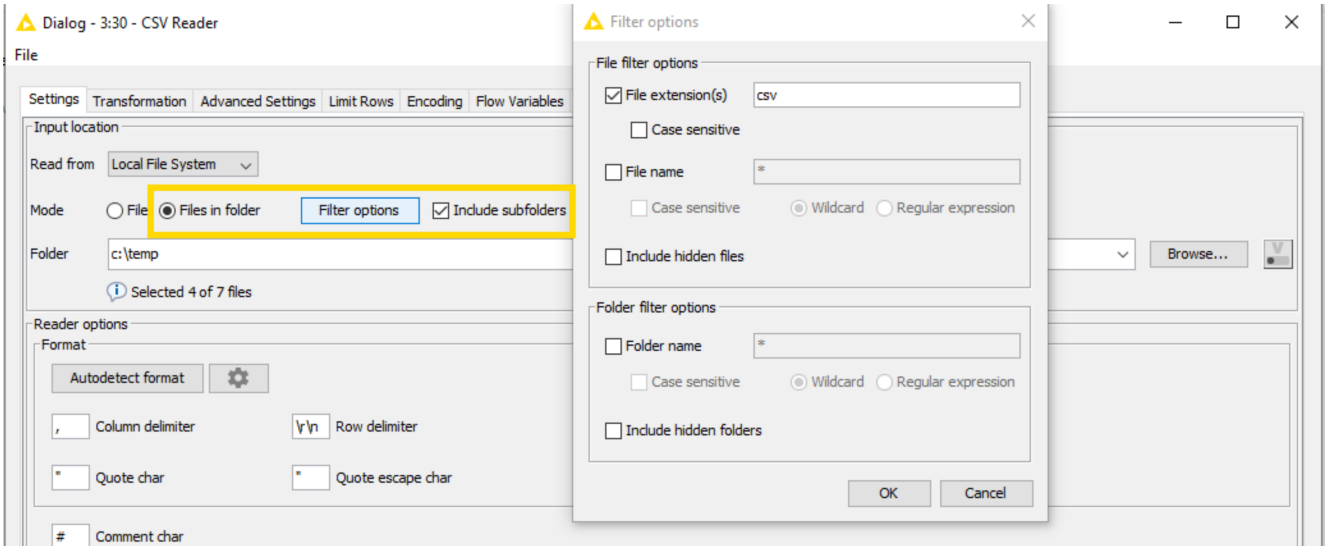
Using the old file handling framework you had to use loops in order to read multiple files into a single KNIME data table. So your workflows looked like the following image.

This is a workflow where the old file handling framework is used.



© 2023 KNIME AG. All rights reserved.

With the new file handling framework you no longer need to use loops since the reading of multiple files into a single KNIME data table is now built into the reader nodes them self. All you have to do is to change the reading mode to *Files in Folder*. You can than open the Filter options dialog by clicking on the *Filter options* button. Here you can filter the files that should be considered e.g. based on their file extension. In addition you can specify if files in subfolders should be included or not.



If you continue to use a loop to read multiple files and encounter problems during the execution have a look at the Controlling readers via flow variables section.

Reading from or writing to remote file systems

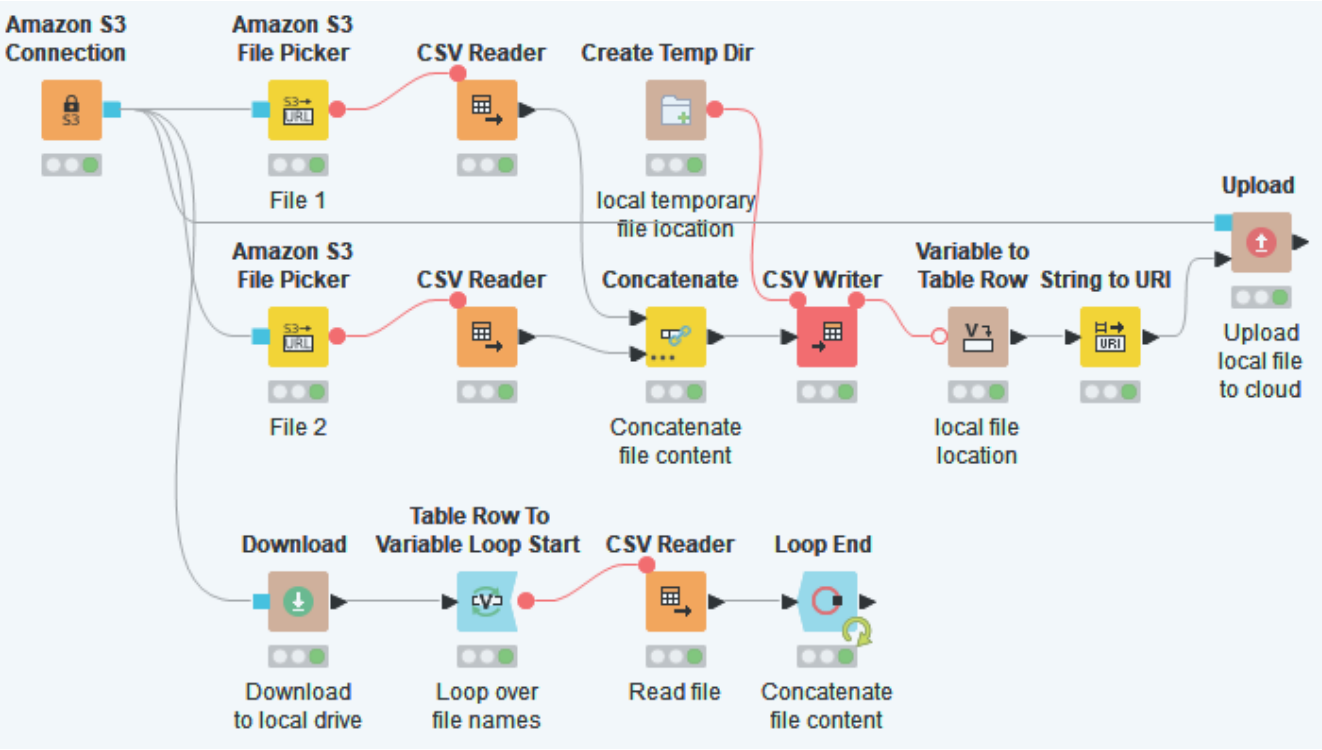
With the old file handling framework, when working with remote file systems such as Amazon S3 or HDFS you had two options when reading files:

1. Download the file to a local folder and then point the reader node to the local copy
2. If available, use one of the File Picker nodes (e.g. Amazon S3 File Picker, Azure Blob Store File Picker or Google Cloud Storage File Picker) to create a signed URL to pass into the reader node.

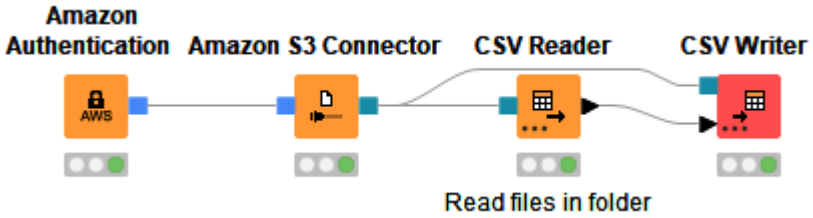
In order to write a file to a remote file system you had to first write the file to your local hard drive and then upload it to the remote file system.

© 2023 KNIME AG. All rights reserved.

This is a workflow where the old file handling framework is used



With the new file handling framework you no longer need to use any additional nodes to work with files in remote file systems but simply connect the reader and writer nodes to the corresponding connector node via the dynamic ports.



Working with KNIME URL

When working with the nodes of the new file handling framework you no longer need to create KNIME URLs but can use the built-in convenience file systems Relative to and Mountpoint. The following table lists the relation between the KNIME URL and the two new file systems.

KNIME URL	Convenient file system
knime://knime.node/	No direct replacement due to security reasons but check out Relative to → Current workflow data area

© 2023 KNIME AG. All rights reserved.

KNIME URL	Convenient file system
knime://knime.workflow/	Relative to → Current workflow*
knime://knime.mountpoint/	Relative to → Current mountpoint
knime://<mountpoint_name>/	Mountpoint → <mountpoint_name>

Due to security reasons KNIME workflows are no longer folders but considered as files. So you can no longer access data within a workflow directory except for the newly created workflow data area.

If you need the KNIME URL for a *Relative to* or *Mountpoint* path you can use the Path to String and Path to String (Variable) nodes with the *Create KNIME URL for 'Relative to' and 'Mountpoint' file systems* option enabled, as it is by default.

We encourage you to use the new convenience file systems Relative to and Mountpoint in favor of the KNIME URLs. For more details on how to convert from the new file systems to the old KNIME URLs see the Working with old and new flow variables section.

Excel Appender

The functionality of the *Excel Appender* node has been integrated into the new Excel Writer node. The node allows you to create new Excel files with one or many spreadsheets but also to append any number of spreadsheets to an existing file. The number of sheets to write can be changed via the dynamic ports option of the node.

File Reader and File Reader (Complex Format) nodes

With KNIME Analytics Platform release 4.4.0 we:

- Improved the **File Reader node**
- Introduced a new **File Reader (Complex Format) node**

The **File Reader node** now supports the new File Handling framework and can use path flow variables and connect to file systems. The File Reader node is able to read the most common text files.

The newly introduced **File Reader (Complex Format) node** also supports the new File Handling framework, and it is able to read complex format files. We recommend to use the

© 2023 KNIME AG. All rights reserved.

File Reader (Complex Format) node only in case the File Reader node is not able to read your file.

With the File Reader node you can also select the *Use new schema* option in the *Advanced Settings* tab of its configuration dialog. This allows the node to support eventual input file structure changes between different invocations. This is instead not possible with the File Reader (Complex Format) node that does not support files changing schemas. Thus, if the File Reader (Complex Format) node is used in a loop, you should make sure that all files have the same format (e. g. separators, column headers, column types). The node saves the configuration only during the first execution. Alternatively, the File Reader node can be used.

© 2023 KNIME AG. All rights reserved.

KNIME AG
Talacker 50
8001 Zurich, Switzerland
www.knime.com
info@knime.com

The KNIME® trademark and logo and OPEN FOR INNOVATION® trademark are used by KNIME AG under license from KNIME GmbH, and are registered in the United States. KNIME® is also registered in Germany.