

KNIME WebPortal User Guide

KNIME AG, Zurich, Switzerland
Version 4.7 (last updated on 2018-12-13)



Table of Contents

| | |
|--|----|
| Introduction | 1 |
| Publishing workflows and report workflows | 2 |
| WebPortal browser | 3 |
| QuickForm nodes | 4 |
| Example | 4 |
| Available Quickform nodes | 5 |
| Configuration of Quickform nodes | 9 |
| Creating workflows for KNIME WebPortal | 11 |
| Wrapped Metanodes | 11 |
| Stepping through pages - Wizard execution | 11 |
| JavaScript visualizations | 12 |
| Layouting | 12 |
| Best practices for creating workflows with Quickform nodes | 17 |
| WebPortal and reports | 19 |
| The WebPortal page | 20 |
| Main page | 20 |
| Details pane | 21 |
| Settings page | 22 |
| Using legacy Quickform nodes | 23 |
| Accessibility support | 24 |
| Direct linking to workflows in the WebPortal | 25 |
| Available workflow parameters | 25 |
| Setting Quickform parameters | 26 |

Introduction

KNIME WebPortal is an extension to KNIME Server. The WebPortal provides a web interface that lists all accessible workflows, allows to execute them and to investigate the results. The workflows' input and output can be parameterized using "QuickForms". In addition, report templates created with KNIME Report Designer that are uploaded together with their associated workflow to the shared repository on KNIME Server can be filled dynamically with data and downloaded in various formats.

Publishing workflows and report workflows

Workflows and the associated report templates are published on KNIME WebPortal by simply uploading the workflow from the local workspace to the server. This can be done either via the context menu in KNIME Explorer (copy/paste), or by dragging and dropping a workflow from a local workflow repository to a ServerSpace repository. Access permissions can be assigned to restrict the access to certain user groups.

Please refer to the [KNIME Explorer User Guide](#) for more information.

WebPortal browser

You can use a standard web browser to connect to KNIME WebPortal. The address (URL) to get to the login page is:

```
http://<server-address>/knime
```

The last part of the address may have been changed by your server administrator.

After providing the user's login name and password, a listing of all available workflows is displayed. Only workflows the user can read and execute are shown.

QuickForm nodes

The nodes in the node repository category "Workflow Control → QuickForms" can be used to create flow variables and to assign them values. These values can then be used in the workflow to control the execution of nodes. There are different quick form nodes for different types of variables (e.g. "Integer Input", "String Input", "Double Input", etc.).

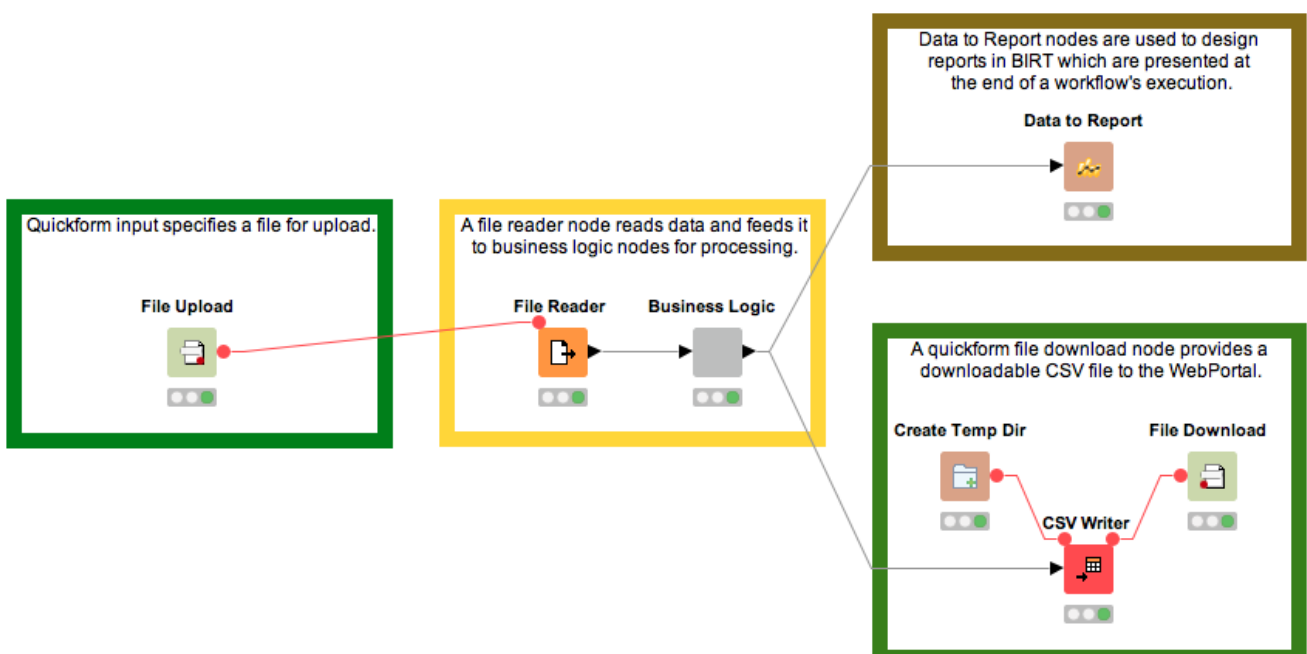
QuickForm nodes are especially useful when used in workflows designed for KNIME Server deployment via the WebPortal. When the WebPortal encounters a QuickForm node in a workflow, it prompts the user to enter new values for its variables before proceeding with the workflow execution. The new values are injected into the workflow, and used to parameterize its execution.

Example

Imagine we want to show a workflow in the WebPortal that allows the user to convert an XLS-file into a CSV-file (comma separated values text file). The workflow is very simple: just connect an Excel Reader (XLS) with a CSV Writer.

For the WebPortal, we add nodes to enable the user to upload the XLS-file and to download the result, and to write to a temporary location on the server. After the workflow is created it is uploaded to KNIME Server.

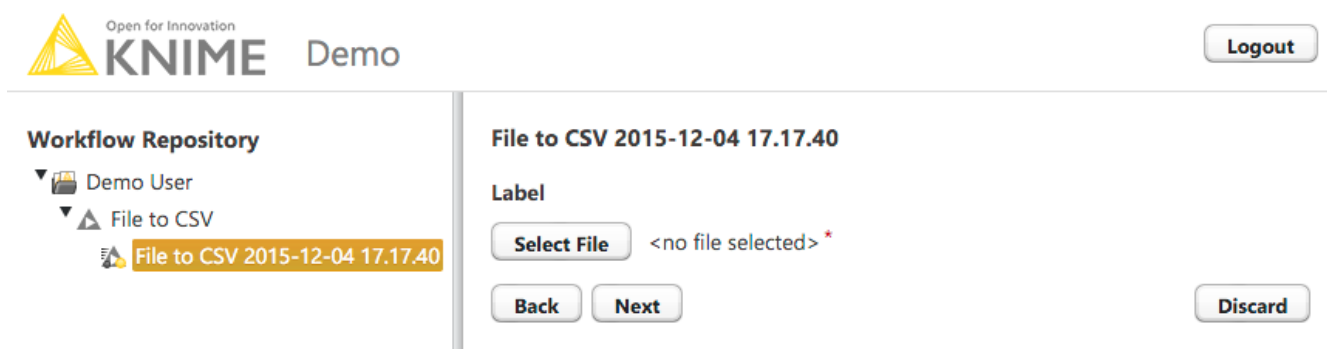
Example workflow



This example demonstrates a common sequence of nodes used to upload and download data using KNIME WebPortal. In addition to the "regular" nodes used for reading and analyzing the data (center, yellow) there are additional QuickForm nodes (left and lower right, green) that provide the workflow with file import and export capabilities via a browser. Please note, that a functioning workflow needs to nest the QuickForm nodes into a Wrapped Node. This has been omitted in the picture, to better show the concept.

Finally, a Data to Report node (upper right, brown) is used to send data to a predefined BIRT Report.

WebPortal page for the example workflow









The screenshot shows the KNIME WebPortal interface. At the top left is the KNIME logo with the tagline "Open for Innovation" and the word "Demo". At the top right is a "Logout" button. The main area is divided into two panes. The left pane, titled "Workflow Repository", shows a tree view with "Demo User" expanded to show "File to CSV", which is further expanded to show "File to CSV 2015-12-04 17.17.40" selected. The right pane shows the details for the selected workflow, including the title "File to CSV 2015-12-04 17.17.40", a "Label" field, a "Select File" button with the text "<no file selected> *", and "Back" and "Next" buttons. A "Discard" button is located at the bottom right of the right pane.



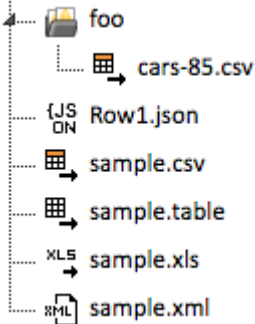


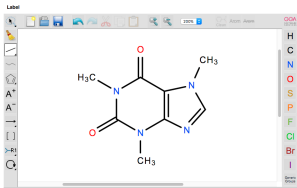

After logging in to KNIME WebPortal a user can select from available workflows. On the right pane of the page, the components corresponding to the QuickForm input nodes in the workflow are shown. After selecting a file, the workflow can be executed on the server by clicking Run. After successful execution, the QuickForm output nodes are displayed providing the user with a download link of the converted result file:









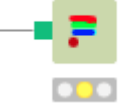
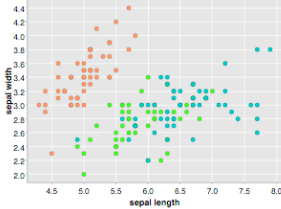
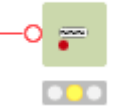
The screenshot shows the KNIME WebPortal interface. At the top left is the KNIME logo with the tagline "Open for Innovation" and the word "Demo". At the top right is a "Logout" button. The main area is divided into two panes. The left pane, titled "Workflow Repository", shows a tree view with "Demo User" expanded to show "File to CSV", which is further expanded to show "File to CSV 2015-12-04 17.17.40" selected. The right pane shows the details for the selected workflow, including the title "File to CSV 2015-12-04 17.17.40", a "Label" field, a "File" link, and "Back" and "Next" buttons. A "Discard" button is located at the bottom right of the right pane.

Available Quickform nodes

| Node Name | Displayed component in KNIME WebPortal | User Input | Output |
|---|--|---|------------------------------|
| Input Nodes | | | |
| Boolean Input  | Label <input checked="" type="checkbox"/> | Boolean values | Checked = 1 Unchecked = 0 |
| String Input  | Label <input type="text"/> | Any user input is accepted | String |
| Integer Input  | Label <input type="text" value="2"/> | Integer values | Integer |
| Double Input  | Label <input type="text" value="0.3"/> | Floating point numbers | Double |
| Date Input  | Label Date: <input type="text" value="2015-12-04"/> Time: <input type="text" value="17"/> : <input type="text" value="33"/> | A date (as string) (or a selected date from the calendar form). | String |
| Credentials Input  | Label User: <input type="text"/> Enter Description: <input type="text"/> Password: <input type="text"/> | User credentials (user name and password) for later use in authenticated nodes. | Credentials Flow Variable |

| Node Name | Displayed component in KNIME WebPortal | User Input | Output |
|---|---|--|--|
| File Upload  | Label <input type="button" value="Select File"/> <no file selected> * | Upload a file to the server using a temporary folder. | Path to the uploaded file |
| File Chooser  |  | Select one or multiple remote files, workflows or folders. | Table with paths to selected items (as knime:// protocol). First path is also output as flow variable. |
| List Box Input  | Label <input type="text"/> | Separate string inputs | Data Table |
| Molecule String Input  |  | Molecule String | String plus SVG image of the molecule (if supported by configured sketcher) |
| Selection Nodes | | | |
| Single Selection  | Label <input checked="" type="radio"/> a <input type="radio"/> b <input type="radio"/> c | Choice of the available values. The available selection depends on the node's configuration. | String |

| Node Name | Displayed component in KNIME WebPortal | User Input | Output |
|---|--|---|--|
| Multiple Selections  | Label <input checked="" type="checkbox"/> a <input type="checkbox"/> b <input checked="" type="checkbox"/> c <input type="checkbox"/> d | Multiple selections | Column of selections |
| Column Selection  | Label <input type="text" value="class"/> | Column name | String |
| Value Selection  | Label <input type="text" value="class"/> <input type="text" value="Iris-setosa"/> | Value of a column | String |
| Filter Nodes | | | |
| Column Filter  | Label <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid red; padding: 5px;"> Excludes sepal length sepal width petal length petal width </div> <div style="border: 1px solid green; padding: 5px;"> Includes class </div> </div> | Select columns from a KNIME Table | Empty data table with selected columns. Often used with Reference Column Filter. |
| Value Filter  | Label <input type="text" value="class"/> <div style="display: flex; justify-content: space-between;"> <div style="border: 1px solid red; padding: 5px;"> Excludes Iris-setosa Iris-virginica </div> <div style="border: 1px solid green; padding: 5px;"> Includes Iris-versicolor </div> </div> | Values of a column | Data Table |
| Output Nodes | | | |
| File Download  | Label Link | The flow variable specified must contain an absolute file path. | String |

| Node Name | Displayed component in KNIME WebPortal | User Input | Output |
|--|--|--------------------|-------------------------------|
| <p>Image Output</p>  | <p>Label</p>  | <p>KNIME Image</p> | <p>SVG or PNG image</p> |
| <p>Text Output</p>  | <p>Label</p> <p>Sample output text</p> | <p>Any text</p> | <p>String or HTML content</p> |

Configuration of Quickform nodes

All QuickForm input nodes have very similar configuration dialogs. Below is as representative example of the dialog of a String Input node, the right hand picture shows the corresponding display in the WebPortal web page:

Dialog - 0:29:0:19 - String Input

Control Flow Variables Job Manager Selection Memory Policy

Label: Result Column: 1

Description: Please enter the name used for the result column 2

Hide in Wizard: 3

Hide in Dialog:

Variable Name: resultColumn 4

Parameter Name: result-column

Regular Expression: [dropdown]

Validation Error Message: [text field]

Common Regular Expressions: [dropdown] Assign

Default Value: Result 5

OK Apply Cancel ?

Result Column:

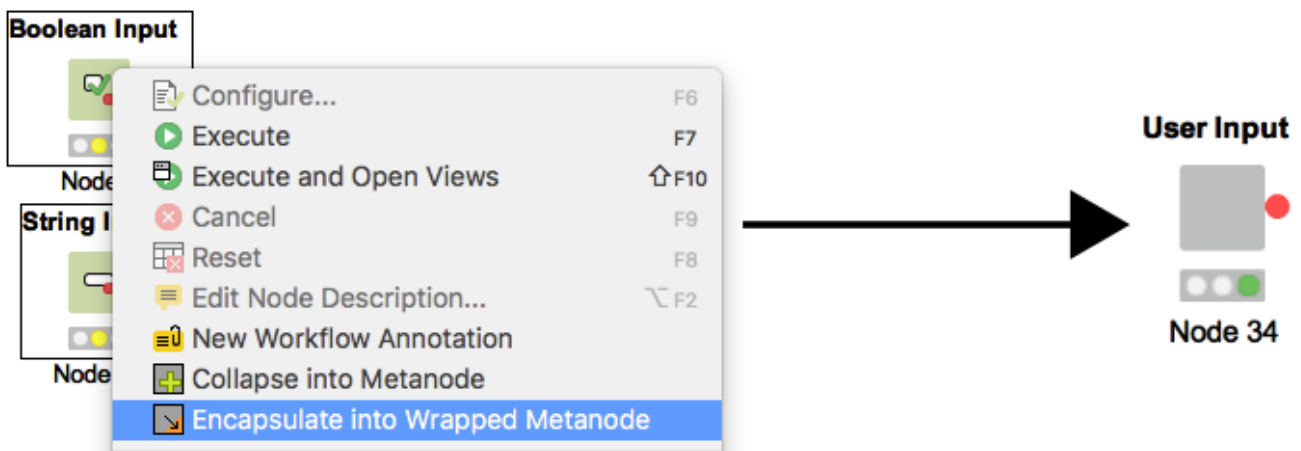
Result Please enter the name used for the result column

1. The text displayed above this input field.
2. The tooltip text seen on mouse over.
3. Check, if this Quickform is not supposed to be shown in KNIME WebPortal, you can use this option e.g. for injecting static input into the workflow
4. The name of the variable that carries the value of this Quick Form node. It should be a unique and will automatically be assigned a type.
5. The default value assigned to the variable named above. Used both for workflow testing and as the default value in KNIME WebPortal.

Creating workflows for KNIME WebPortal

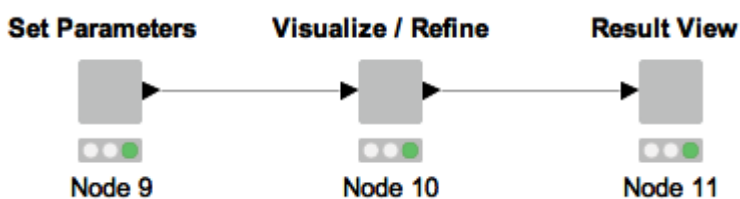
Wrapped Metanodes

A workflow execution on the WebPortal is comprised of one or more pages or wizard steps, where a user gets guided through the process. Each page or step is modelled in the workflow by the use of Wrapped Metanodes. These nodes are an extension to Meta Nodes and can easily be created by selecting the nodes to be included in a page and selecting "Encapsulate into Wrapped Metanode" from the context menu.

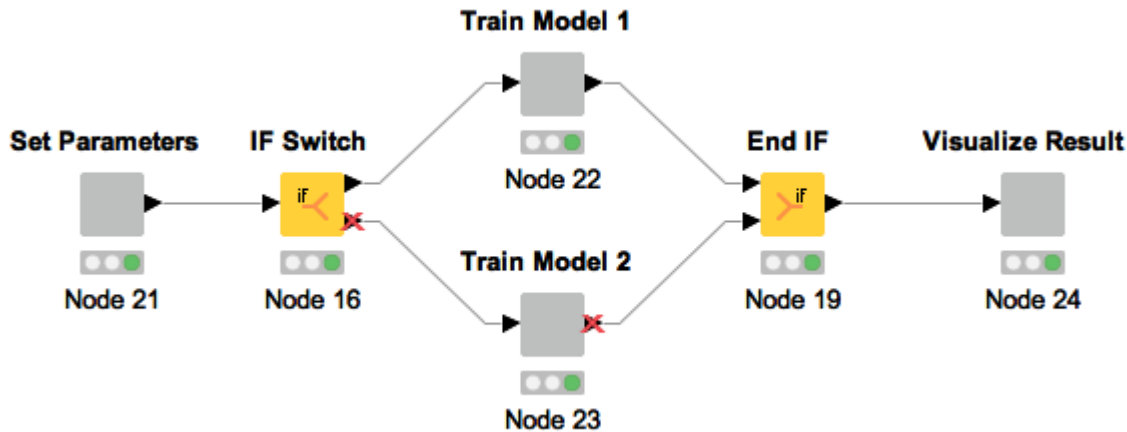


Stepping through pages - Wizard execution

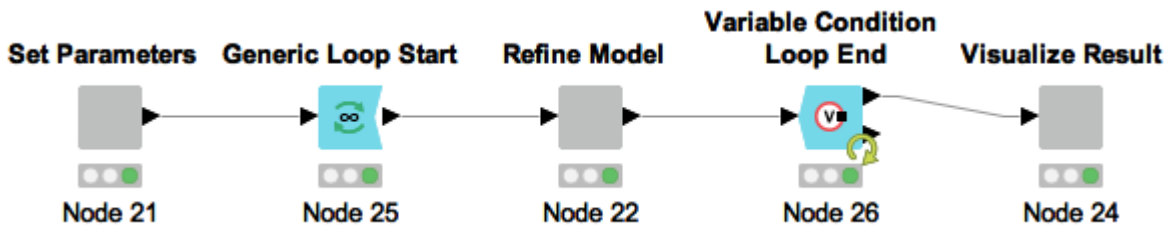
As each Wrapped Metanode represents one WebPortal page, a series of 3 pages can be modelled as a sequence of three Wrapped Metanode:



To dynamically show and hide pages during execution it is possible to disable a Wrapped Metanode, e.g. by means of an IF or CASE block.



It is also possible to use Wrapped Metanodes inside loops for example to iterate over an item set or recursively refine a model.



JavaScript visualizations

Each component that can be displayed in the WebPortal is a JavaScript-based node visualization.

This can be a Quickform node, but also an interactive chart or other component. KNIME provides a basic set of additional JavaScript-based interactive nodes in the KNIME Labs category JavaScript views. If not present these nodes can be installed, by selecting *Install KNIME extension*, browsing to *KNIME Labs Extensions* and selecting *KNIME JavaScript views* from the installation dialog.

Layouting

On any Wrapped Metanode that contains at least one Quickform or JavaScript-enabled view, a layout can be defined. The layout editor can be accessed on the top toolbar, when active:



The layout is based on the JavaScript framework Bootstrap and uses its 12 column-based, responsive grid system. To get more information and documentation about Bootstrap please

visit the [Bootstrap website](#).

The layout structure is saved in a JSON format and at the moment is editable directly in a text editor upon opening the layout dialog. In the future the text editor will be replaced by a visual component to allow for easier setup.

If no layout information exists, when the editor is opened an initial layout is created. This layout incorporates all applicable nodes at the time of creation and is a good starting point for understanding the structure and further customization and fine tuning.

The summary below describes the JSON structure in a non-normative descriptive way.

Row

A row is the outer most element that can be defined and is the first element inside the layout container. The JSON structure's outer layer is an array of rows. A row contains a number of layout-columns.

To further customize a row, optional fields can be used. `additionalClasses` can be used to provide an array of class names to append to the created HTML row element, `additionalStyles` is an option to directly insert CSS style commands on the element. For example, to create a visual separator between one row and the next, a bottom border can be used like this:

```
"additionalStyles" : [ "border-bottom: thin solid grey;" ]
```

Column

A column is a layout element inside a row, which determines the width of its components. To define a width, a number between 1 and 12 (corresponding to the Bootstrap column widths) has to be used. 12 means taking up 100% of the width, whereas 6 would be 50% of the width. This way it is possible to define a layout with components side by side, by providing their relative proportion. E.g. if three components are to be laid out horizontally with equally distributing the space, use a row with 3 columns, which each have a set width of 4. If the sum of widths for a particular column is larger than 12, the column is wrapped into a new line.

Additionally, it is possible to define the widths of the columns so they adapt to the screen size. With this option, responsive layouts can be achieved.

To define the width of a column, use one or more of the following fields: `widthXS`, `widthSM`, `widthMD`, `widthLG` which corresponds to the [Bootstrap size definitions](#) of XS, SM, MD and LG.

The content of a column can be an array of one of three things. Either another set of rows, which provides the possibility to create nested layout structures, or regular HTML content to insert plain HTML elements into the layout, or a node reference to embed the contents of a JavaScript-enabled KNIME node. As for rows, it is also possible to use the optional fields `additionalClasses` and `additionalStyles` to further customize the column.

HTML content

It is possible to include plain HTML into the layout, by placing a content element of type `html` inside a column. One use case would be to provide a title or descriptive text about the page. To insert the content a single field `value` is used. Please note that since Bootstrap is present during runtime, also the regular Bootstrap elements can be used. This way glyphs, labels, badges, jumbotrons, ... are available to define the layout of the page. Example:

```
[...]
"content":[{
  "type":"html",
  "value":"<h2 >Title defined in layout</h2><p><span class='glyphicon glyphicon-th' \
        aria-hidden='true' style='margin-right: 5px;'></span>Glyphicon</p>"
}]
[...]
```

View content

To embed the contents of a KNIME node inside the layout, a content element with type `view` has to be used. The element has quite a few ways to customize the sizing and behavior of the content, which are explained in the table further down.

Referencing the node is done by the field `nodeID`, which takes the ID-suffix of the node as a string argument. If nodes exist inside the Wrapped Metanode, which are not referenced by the layout, a warning message appears underneath the editor. Also referencing nodes twice or referencing non-existing nodes will issue a warning.

To understand the sizing concepts better it is important to know, that each node's content is wrapped in it's own `iframe` element to encapsulate the implementation and avoid reference and cross-scripting issues. As such, `iframe` elements do not adapt to their content's size automatically and need to be resized to achieve the desired behavior. In principal there is three options implemented:

Size-based methods

This method uses an [iframe-resizer library](#) to resize the `iframe` according to the sizing of its contents. This means that the content has to have a concrete size explicitly or implicitly set. For example all Quickforms are best laid out using this approach. To determine the size is possible using different approaches. Depending on the desired use case, browser environment and view implementation one or another might be better suited. The [iframe-resizer GitHub page](#) explains the available calculation methods. Size-based resize methods all start with the prefix `view...` in the JSON structure.

Aspect-ratio based methods

If a node's view is set to adapt to its parent's size, rather than implicitly providing a size, the size-based methods will not work or cause strange behavior. This is for example the default setting on the JavaScript Scatter Plot. To allow these views to take up appropriate available space in the layout an aspect ratio setting can be used. The width is taken as 100% of the available width at that position in the layout and the height is calculated according to the given ratio. At the moment two options exist: 4 by 3 and 16 by 9. To achieve this resizing behavior, [Bootstrap's responsive embed](#) concept is used. Aspect-ratio based resize methods start with the prefix `aspectRatio` in the JSON structure.

Manual method

It is also possible to completely manually trigger resize events at appropriate times. This requires the node's implementation to make the appropriate resize calls itself.

List of available fields

| Field name | Explanation / Possible Values |
|------------------------|--|
| <code>nodeID</code> | ID-suffix of referenced node |
| <code>minWidth</code> | Constrain the size of the <code>iframe</code> by setting a minimum width in pixels. |
| <code>minHeight</code> | Constrain the size of the <code>iframe</code> by setting a minimum height in pixels. |
| <code>maxWidth</code> | Constrain the size of the <code>iframe</code> by setting a maximum width in pixels. |

| Field name | Explanation / Possible Values |
|-------------------|--|
| maxHeight | Constrain the size of the <code>iframe</code> by setting a maximum height in pixels. |
| resizeMethod | The resize method used to correctly determine the size of the <code>iframe</code> at runtime. Can be any of the following values: <code>viewBodyOffset</code> , <code>viewBodyScroll</code> , <code>viewDocumentElementOffset</code> , <code>viewDocumentElementScroll</code> , <code>viewMax</code> , <code>viewMin</code> , <code>viewGrow</code> , <code>viewLowestElement</code> , <code>viewTaggedElement</code> , <code>viewLowestElementIEMax</code> , <code>aspectRatio4by3</code> , <code>aspectRatio16by9</code> , <code>manual</code> |
| autoResize | Boolean only working with size based resize methods. Use this to enable or disable automatic resizing upon window size or DOM changes. Note that the initial resize is always done. |
| resizeInterval | Number only working with size based resize methods. Sets the interval to check if resizing needs to occur. The default is 32 (ms). |
| scrolling | Boolean only working with size based resize methods. Enables or disables scroll bars inside <code>iframe</code> . The default is <code>false</code> . |
| sizeHeight | Boolean only working with size based resize methods. Enables or disables size adaption according to content height. The default is <code>true</code> . |
| sizeWidth | Boolean only working with size based resize methods. Enables or disables size adaption according to content width. The default is <code>false</code> . |
| resizeTolerance | Number only working with size based resize methods. Sets the number of pixels that the content size needs to change, before a resize of the <code>iframe</code> is triggered. The default is 0. |
| additionalClasses | Array of additional classes added to the HTML container element. |
| additionalStyles | Array of additional CSS style declaration added to the HTML container element. |

Debugging resize behavior

If the WebPortal is used in debug mode, log messages are printed to the browser's console for all size-based resize method components. Also all JavaScript libraries will be included in

a non-minified version to allow for JavaScript debugging in the browser.

Best practices for creating workflows with Quickform nodes

Quickform nodes in a workflow define points of interaction in a workflow when it is executed via the WebPortal. This allows skilled analysts to expose powerful tools to users who may not have the training or time to run a full featured workflow from KNIME Analytics Platform. This is usually done by injecting flow variables from the Quickform node into another node's settings to drive the configuration of that node on a per setting basis. Variables created by a Quickform node are available at the node's variable output port (circular and red). To use the variable, connect the Quickform variable port to the variable port of the target node whose settings you wish to override. If a target node doesn't have a flow variable port you can make it visible with *Show Flow Variable Ports* from its context menu. Alternatively, connect the output port to the upper left corner of the node (a convenient shortcut).

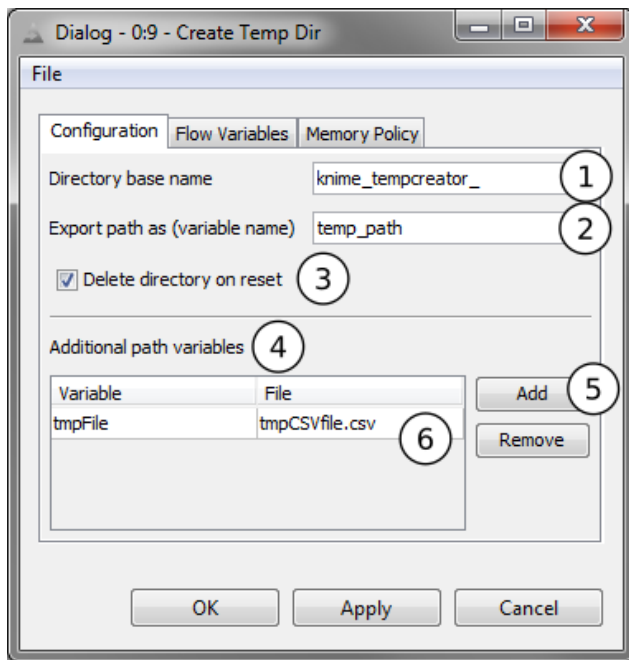
The overall process for developing a workflow looks like this:

1. Build a workflow and execute it with relevant test data.
2. Parameterize the workflow by adding Quickform nodes as needed. Use the variables generated by these nodes to override the settings of your target nodes.
3. Make the workflow portable by removing any hard-coded paths.
4. Reset the workflow, save it, and upload it to the server and set appropriate permissions.

Create Temp Dir node

The *Create Temp Dir* node creates a temporary folder when it is executed (in the Java Temp folder, set in the KNIME properties). It can also be configured to create a new temporary file in this folder and store the full absolute path to that file in a flow variable. (In fact the file is not created but the generated file path in the new temp folder certainly doesn't exist.)

This is the configuration dialog of the *Create Temp Dir* node:



Create Temp Dir node settings:

1. Prefix for the new folder. Unique names will be created automatically.
2. Flow variable to be created containing the absolute path of the new folder.
3. If enabled, the newly created folder will be deleted when the node is reset (along with all contained files).
4. Create file paths:
5. Add a new variable and file reference.
6. For each row in this table, a new workflow variable is created. The variable name is entered on the left and the file to be created on the right. The final variable will refer to the full path including folder and filename.

With this node you can create temporary files (in unique temporary folders) and use them in writer nodes. If the variable that carries the name of the temporary file is then selected in a File Download node, the file written is made available for download in the WebPortal.

Java Edit Variable node

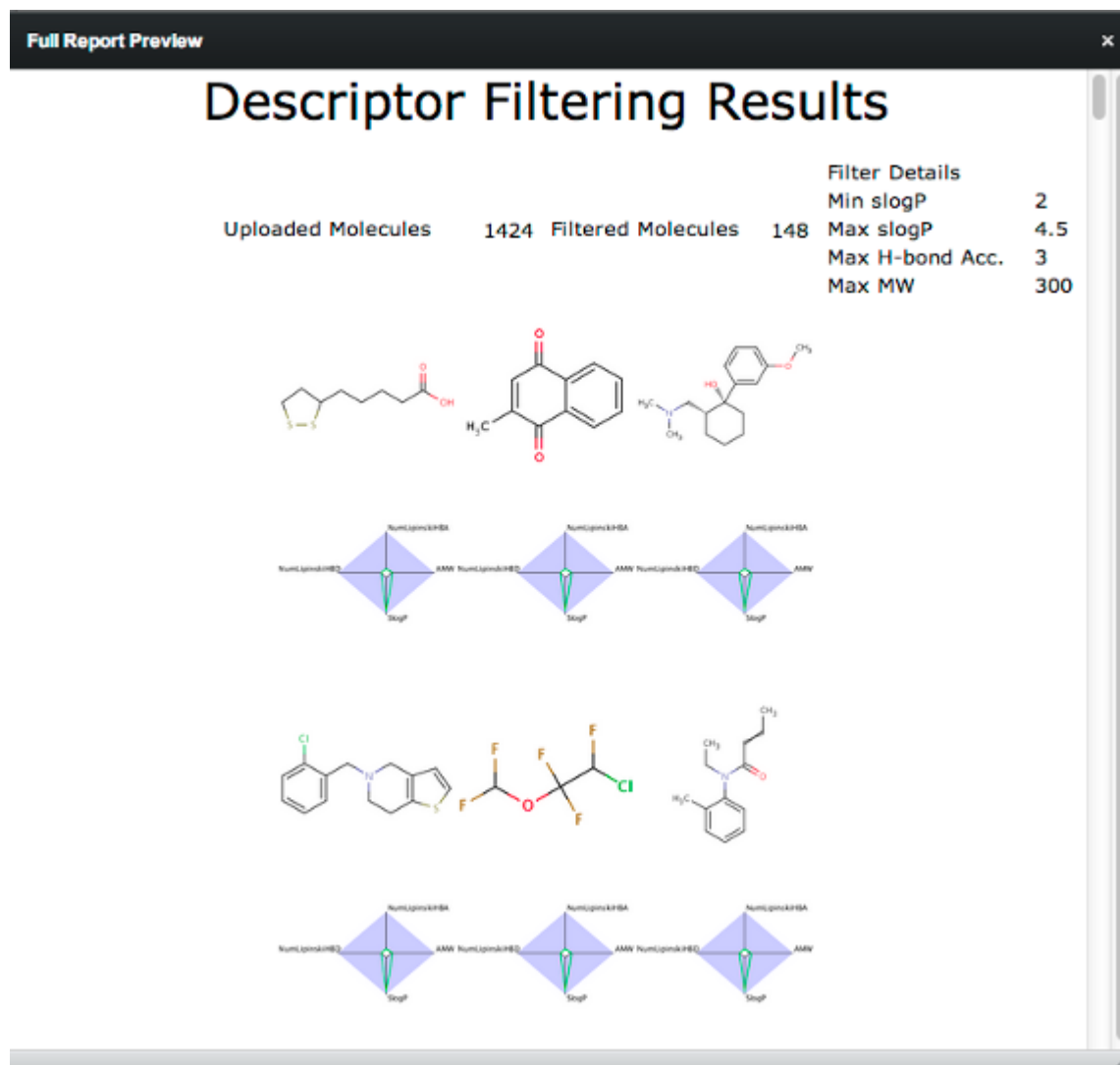
Another node that could be useful when flow variables are used is the *Java Edit Variable* node. It allows you to modify the value of an existing variable or to create a new variable and/or derive its value from another existing variable.

Java IF (Table) node

With the *Java IF (Table)* node you can selectively execute branches in the workflow, based on the value of a flow variable. Together with the String Radio Buttons node the KNIME WebPortal user could select different workflow branches for execution.

WebPortal and reports

If a workflow with a report template is uploaded to KNIME Server and accessed through the WebPortal, the report is generated and displayed after successful execution of the workflow:



A preview of the report is shown in the result page and the HTML preview can be opened in a new window. Download links are offered for PDF, XLS, PPT and DOC format.

The WebPortal page

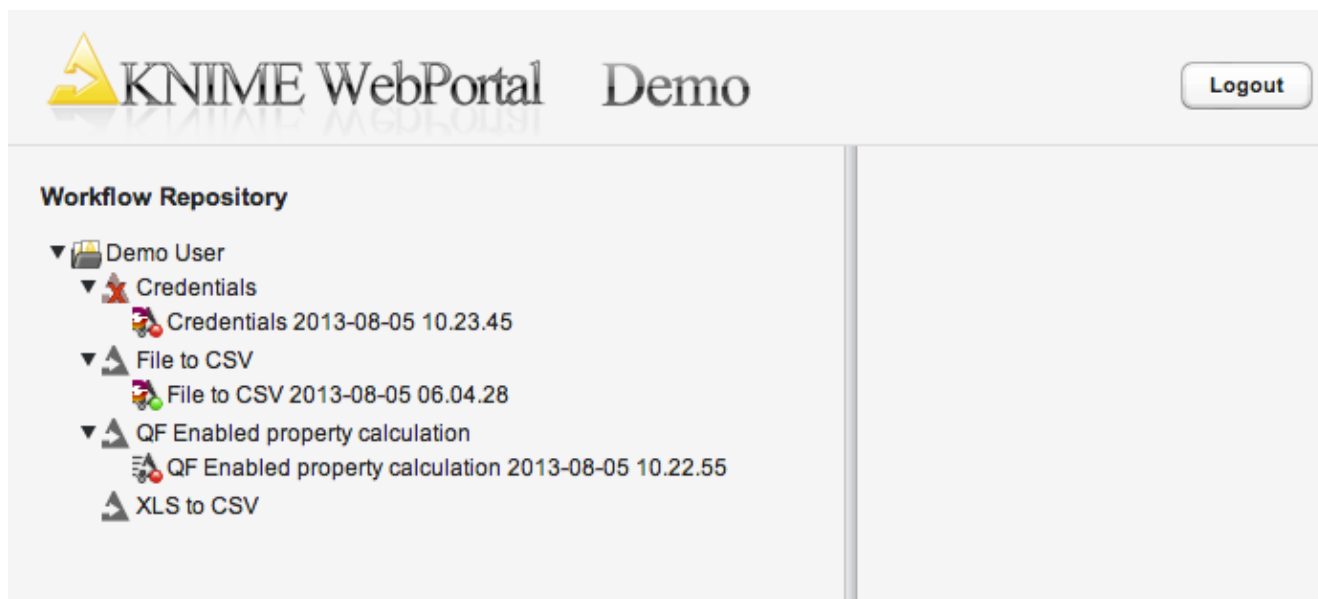
Workflows that are uploaded to KNIME Server are available through the WebPortal for execution. The WebPortal (default) entry page is:

```
http://<server-address>/knime
```

The last part of the address may have been changed by your server administrator.

Main page

After logging in, the WebPortal main page is displayed showing the available workflows displayed in a structured list in the left frame of the page. Only workflows that are both readable and executable by the logged in user will be visible.



In the above example the user has only access to the flows in the workflow group "Demo User" and four workflows are available for execution in the WebPortal.

The items below a workflow (with a green or red indicator dot) are previously executed flows which are kept in memory ("workflow jobs"). They can be discarded individually with a right-click on the job, or with a right-click on the flow.

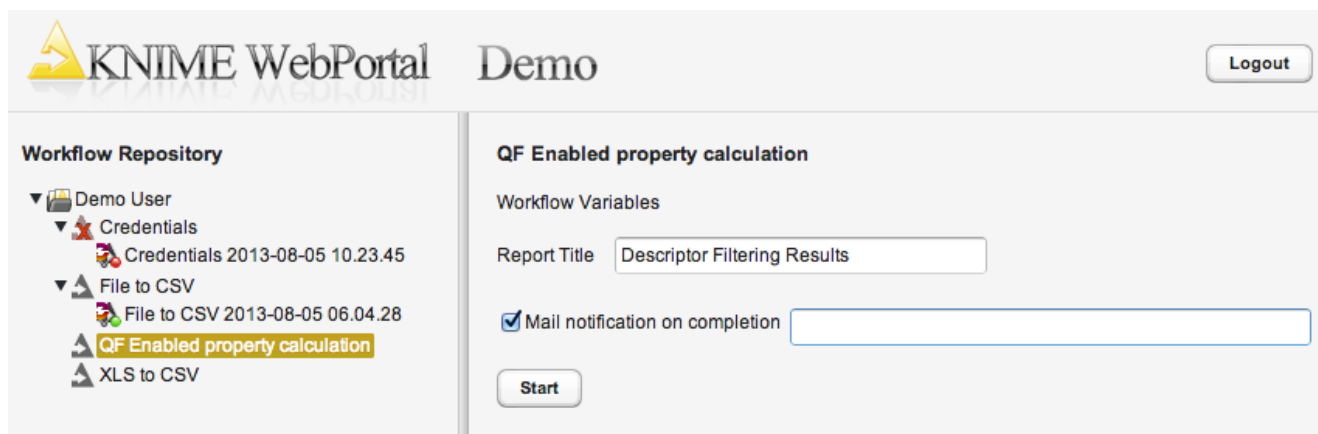
Jobs with a green indicator represent successfully executed flows. If they are selected, the result from the previous run is available.

Jobs with a red indicator represent flows previously executed but that have failed. If selected, the error message is displayed again.

If a job itself is red, this indicates that the workflow has been modified since the job was executed while if the Workflow has a red x across it, this means that the workflow which owns that job has been deleted.

Details pane

If a workflow is selected in the left section, its details page is shown in the section on the right. An example is illustrated below:



If Meta Information has been entered before the flow was uploaded, it is displayed beneath the workflow name. If the option "Mail notification on completion" option is selected, an email address must be entered. The server sends a notification email to this address as soon as the workflow execution finishes. After all information is entered, the flow execution can be initiated by clicking the "Start" button. A new job is then created.

If the job finishes the result can be inspected by selecting it in the left pane of the page. The example job shown below executed successfully (indicated by the green icon and the green line of text on the right).

Beneath these status lines the result files are available for download. The workflow contains Quick Form nodes that provide the path to two files. If the user clicks on one of the links the download of the corresponding file starts.

This example flow is generating a report. The preview is shown in the details pane. At the bottom of the pane the report can be previewed in a new window and is also available for download, either in PDF or XLS format, as PowerPoint presentation or as Word document.

If the user launches long-running jobs on the server through the WebPortal, the user can logout and come back any time later, re-connect and get the status of his jobs. The jobs can be canceled, or finished jobs can be inspected and the results can be downloaded.

Workflow Repository

- ▼ Demo User
 - ▼ Credentials
 - ▼ Credentials 2013-08-05 10.23.45
 - ▼ File to CSV
 - ▼ File to CSV 2013-08-05 06.04.28
 - ▼ QF Enabled property calculation
 - ▼ QF Enabled property calculation 2013-08-05 10.41.31
 - ▼ XLS to CSV

QF Enabled property calculation 2013-08-05 10.41.31

Workflow executed successfully

This workflow was designed to demonstrate the basic functionality of a web enabled KNIME workflow. Upload an SDF file

[Back](#) [Discard](#)

Report Preview for QF Enabled property calculation 2013-08-05 10.41.31

Descriptor Filtering Results

| | | Filter Details | |
|--------------------|------|--------------------|-----|
| Uploaded Molecules | 1424 | Filtered Molecules | 148 |
| | | Min slogP | 2 |
| | | Max slogP | 4.5 |
| | | Max H-bond Acc. | 3 |
| | | Max MW | 300 |

CCCCCCCC(=O)O
Cc1ccc2c(c1)C(=O)c3ccccc23=O
CC(C)C(O)C1=CC=C(C=C1)OC

[Open HTML preview in a new window](#)
 Download Report as
[PDF](#) [XLS](#) [PPT](#) [DOC](#) [PS](#) [ODP](#) [ODT](#) [ODS](#)

The job (and all its results) can be discarded by clicking the "Discard" button. Its icon will disappear from the left Workflow Repository view and server resources will be freed. Jobs can also be discarded by right-clicking on their icon in the Workflow Repository (on the left). All jobs of a flow can be removed by right-clicking on the flow. Please note that jobs stay in the server's main memory until they are discarded. If too many users keep too many jobs alive, the server will run out of memory.

Settings page

The settings page allows one to change the password of the currently logged-in user. It can be accessed via the "Settings"-link in the upper-right corner of the WebPortal's main page.

Using legacy Quickform nodes

If a workflow contains legacy Quickform nodes and no regular Quickform or Wrapped nodes it will still be executed in the same manner, as in previous versions of KNIME Server. Please see the WebPortal User Guide for KNIME Server 4.1 for documentation on the nodes and execution behavior. Note that legacy Quickforms will have to be separately installed on the server executor and any client designing or executing legacy workflows. The plugin is found under *KNIME & Extensions* → *KNIME Quick Forms (legacy)* in the "Install KNIME Extensions" dialog.

Accessibility support

KNIME WebPortal provides full support of screen reader software as well as complete keyboard-only access.

Any screen reader that follows [ARIA](#) guidelines can be used for the work in the WebPortal. The recommended screen reader software includes JAWS (v.16), NVDA (v.2016.3) and ChromeVox (v.53.0.2784.1).

The WebPortal can be completely controlled by using keyboard only, implementing the ARIA best practices on keyboard navigation:

- *Tab* (*Shift + Tab*) key moves the focus to the next (previous) element.
- *Enter* key performs the default action on focused element.
- To navigate in the Workflow Repository tree the following shortcuts are available:
 - *Up arrow* and *Down arrow* keys move between the tree nodes.
 - *Left arrow* key on an expanded node closes the node.
 - *Left arrow* key on a closed or end node moves focus to the node's parent.
 - *Right arrow* key expands a closed node, moves to the first child of an open node, or does nothing on an end node.
 - *Enter* key expand/collapse a workflow group or activate a workflow.
 - *Home* key moves to the top node in the tree view.
 - *End* key moves to the last visible node in the tree view.

To operate with UI components of the WebPortal standard shortcuts can be used. For more details see the [ARIA specification page](#).

Direct linking to workflows in the WebPortal

It is possible to link directly to specific workflows in KNIME WebPortal. URLs are generally set up like:

```
http://<server-address>/knime/#/<ItemPath>&<WorkflowParameters>
```

- <ItemPath> is the path to a workflow, workflow group, or workflow job. A workflow job is referenced with its ID like: `WorkflowGroup/Workflow?exec=job_id`.
- <WorkflowParameters> can appear in any order, but have to be after the <ItemPath>. Parameters are always appended with a leading `&`.

Available workflow parameters

`&single` — Single mode: hides the workflow tree, so that only the selected item is visible.

`&run` — Auto run: If appended to workflow path, a new job is automatically executed.

`&root=<RootPath>` — Set root of workflow tree: If set to a valid path, only items underneath the given <RootPath> are available.

`&pm:<name>=<value>` — Set quickform parameters: Sets the named quickform parameter to the specified value. Please see more specific explanation about this feature on the following page.

`&emails=sample@mail.com` — Enable email notification: enables email notification and sets the specified comma separated list of email addresses.

`&formats=<formats>` — Set report formats: sets the report formats included as attachments in the notification email specified by a comma separated list. Available formats are: pdf (enabled by default), html, doc, docx, xls, xlsx, ppt, pptx, ps, odt, ods and odp. This list might have been changed by your server administrator to reduce the number of options available.

`&user=<username>` — Set the user name for login to the WebPortal. Use in conjunction with `&pw` to automatically log in.

`&pw=<password>` — Set the password for login to the WebPortal. Use in conjunction with `&user` to automatically log in. Please note that the password is transmitted in plain text.

A complete URL might look like:

```
http://localhost:8080/knime/#/demo/file%20to%20csv&pm:title=foo&emails=sample@mail.com&run&single
```

Setting Quickform parameters

With the `&pm`-parameter it is possible to override the default value of most Quickform nodes. They are addressable by using the same name as the one set in the "Parameter name" field in the configuration dialog of the corresponding node.

It is best practice to keep those names unique, however it is possible to address nodes, when there are multiple of the same parameter names in the workflow by appending the node id (e.g. `param-name-32:34`).

There are a few nodes, which require a special format, which are listed in the following.

| Quickform Node | Format Remarks |
|-------------------|--|
| Date Input | Use <code>yyyy-MM-dd'T'HH:mm:ss</code> as date format, alternatively other ISO-conform formatting strings can be used to just address date, time or zoned date & time components. |
| File Chooser | Use absolute <code>knime://</code> protocol path |
| Boolean Input | Use <code>true</code> or <code>false</code> |
| Double Input | Use American number notation |
| Credentials Input | Use <code>username:password</code> as value. Please note that the credentials and user name cannot contain the "=" and ":" symbols and that the password is transmitted in plain text. |

Quickform nodes that are not configurable over URL parameters are:

- Multiple Selections
- Value Selection
- Column Filter
- Value Filter

KNIME AG
Technoparkstrasse 1
8005 Zurich, Switzerland
www.knime.com
info@knime.com