movizon CONTROL **Resource Table Settings**

Managed data in movizon CONTROL (mC) are referred to as resources. While there were up to 10 fixed resource types in older versions, version 4 allows you to freely customize resource types. Therefore the browser UI of mC needs to support the same flexibility regarding resource visualization. There are no hardcoded settings to tell the UI to display a resource type called "vehicles" with a relevant context that shows for example current speed, mileage and energy values. This context information can be configured via resource table settings, which currently consist of four resources in type `settings`: `tableColumns`, `tableDetails`, `tableActions` and `tableButtons`.

Table Columns [settings/tableColumns]

The setting `tableColumns` defines the width, order and content of the columns of a resource table. Also the column sort will be determined by the `type` property of this setting. If there is no entry in `tableColumns` for a resource type, a standard layout will be used. It has lock, ID, modification and responsibility columns.

Type Descriptions

type	**description**	**fields**	**JSON example**

string	column definition to display the value of the keyed resource field as string	header
		width
		key
		color
		colors
		link
			{
			"type": "string",
			"header": "Assignment",
			"width": 70,
			"key": "targetId",
			"link": "vehicles"
			}

number	column definition to display the value of the keyed resource field as number	header
		width
		key
		unit
		divisor
			{
			"type": "number",
			"header": "Mileage",
			"width": 60,
			"key": "mileage",
			"unit": "km",
			"divisor": 1000
			}

percent	column definition to display the value of the keyed resource field as percentage with background color transition from red to yellow to green	header
		width
		key
		maxValue
			{
			"type": "percent",
			"header": "Update",
			"width": 60,
			"key": "update.progress",
			"maxValue": "1"
			}

date	column definition to display the value of the keyed resource field as date-time string; value must be UNIX timestamp in milliseconds	header
		width
		key
			{
			"type": "date",
			"header": "Modified",
			"width": 120,
			"key": "timeUpdate"
			}

interval	column definition to display the value of the keyed resource field as interval string along with appropriate unit; value must be in milliseconds	header width key	{ "type": "interval", "header": "Delta TX", "width": 60, "key": "deltaTx"

			}

priority	column definition to display the value of the keyed resource field with background color transition from transparent (lowest priority, highest number) to red (highest priority, lowest number)	header
		width
		key
			{
			"type": "priority",
			"header": "Priority",
			"width": 60,
			"key": "priority"
			}

arrayLength	column definition to display the number of items of the keyed resource field that must be a valid array	header
		width
		key
		unit
			{
			"type": "arrayLength",
			"header": "Fulcrums",
			"width": 50,
			"key": "fulcrums"
			}

arraySum	column definition to display the sum of all numbers of the keyed resource field that must be a valid array	header
		width
		key
		unit
		divisor
			{
			"type": "arraySum",
			"header": "Length",
			"width": 80,
			"key": "length",
			"unit": "m",
			"divisor": 1000
			}

mapSum	column definition to display the sum of all number of the keyed resource field that must be a map with key-value-pairs whose values are numbers	header
		width
		key
		unit
			{
			"type": "mapSum",
			"header": "Influence",
			"width": 60,
			"key": "responsibilityCount"
			}

alert	column definition to display an alert LED for each resource; alert state will be calculated automatically according to model.dashboard definition	icon
		width
			{
			"type": "alert",
			"icon": "alert",
			"width": 21
			}

led	column definition to display the value of the keyed resource field as LED; value must be boolean (in combination with color field) or match to any of the keys defined in colors object	icon
		width
		key
		negated
		color
		colors
			{
			"type": "led",
			"icon": "heart",
			"width": 21,
			"key": "status",
			"colors": {
			"ASSIGNED": "green",
			"SUSPENDED": "gold",
			"DELAYED": "darkorange",
			"NO_ROUTE": "red",
			"NO_TARGET": "red"
			}
			}

Field Descriptions

field	description
header	text to be displayed as column header
width	with in pixels of this column; will be extended automatically if too small for header but overlong row entries will be cutted
key	multilayered definition possible, e.g. x.y
color	color of LED for type led or background color of row entry for type string; any

	css value will work; as background color it's best to use one of the following predefined values:
	var(--yellow-tint)
	var(--orange-tint)
	var(--red-tint)
	var(--pink-tint)
	var(--purple-tint)
	var(--blue-tint)
	var(--teal-tint)
	var(--green-tint)

colors	object with key-value pairs for color definition for type led and string; key has to match value of the keyed resource, value must be a css color value (see above)
link	if value of link is valid resource type, string will act as link to resource
unit	value will display as unit in brackets next to row entry
divisor	divisor to divide the value of the keyed resource field before display; value will be rounded to two digits after decimal point as well
maxValue	maximum value of the keyed resource field; e.g. for type percent: 1 (0..1) or 100 (1..100)
icon	icon to be displayed as column header
negated	if true, LED will glow if the value of the keyed resource field is false

Table Details [settings/tableDetails]

The setting `tableDetails` defines the content of the unfolded area that shows after selecting a single resource in the resource table. There is no limit for the number of detail boxes in this area. The boxes will be arranged automatically to use space efficiently. If there is no entry in `tableDetails` for a resource type, a JSON representation of the selected resource will be displayed.

Type Descriptions

type	**description**	**fields**	**JSON example**

string	definition to display a the value of a single field of the selected resource	label
		key
			{
			"type": "string",
			"label": "message",
			"key": "message"
			}

json	definition to display the JSON representation of the entire resource	label	{
			"type": "json",
			"label": "entries"
			}

map	definition to display a simple map table of multiple field values of the selected resource	label
		entries
			{
			"type": "map",
			"label": "UDP",
			"entries": {
			"IP": "rawInput.ip",
			"Port": "rawInput.port"
			}
			}

errors	definition to display an error history of the selected resource as sorted list;	label key	{ "type": "errors",

	keyed error array must contain error event objects that contains the following fields: timestamp, key, value		"label": "errors", "key": "errorHistory" }

dashboard	definition to display a dashboard with LED-label-pairs of the selected resource; dashboard is defined in linked model resource	label	{
			"type": "dashboard",
			"label": "status"
			}

Field Descriptions

field	description
label	header label of detail box, always displayed in capital letters
key	name of resource field whose value should be displayed; multilayered definition possible, e.g. rawInput.ip
entries	key-value-pairs for map display of resource fields; key will act as label definition, value act as name of resource key

Table Actions [settings/tableActions]

Each resource table has a three-dots-button to open a dropdown that lists extended actions. These actions are linked to the current selection of resources. Some actions require a minimum number of selected resources, some actions will alter its label depending on the state of fields. If there is no entry in `tableActions` for a resource type, a default selection will be used. It has lock, delete, upload and download actions.

Type Descriptions

type	description	fields	JSON example

button	action to set the value of the keyed field of all selected resources to true; mostly used to request something from a resource manager script that will set same value to false afterwards	icon
		label
		key
			{
			"type": "button",
			"icon": "clear",
			"label": "Clear",
			"key": "clearRequest",
			}

switch	action to switch the value of the keyed field of all selected resources; if values are in mixed state, first all are set to true, then false	icon
		label
		opposite
		key
			{
			"type": "switch",
			"icon": "bulb",
			"label": "Hide",
			"opposite": "Unhide",
			"key": "hidden"
			}

delete	action to delete all selected resources; will show warning message before deletion	icon
		label
		warning
			{
			"type": "delete",
			"icon": "trash",
			"label": "Delete",
			"warning": "You are about to delete {0} {1}. This action affects the entire system and cannot be undone."
			}

			}

upload	action to add resources via JSON file uploader	icon
		label
			{
			  "type": "upload",
			  "icon": "upload",
			  "label": "Upload"
			}

download	action to download all selected resources as JSON file that contains single resource object or array of resource objects	icon
		label
			{
			  "type": "download",
			  "icon": "download",
			  "label": "Download"
			}

evaluate	action to execute script evaluation; arguments will be passed as array in the following order: [optional] custom arguments, [optional] form input object, IDs of selected resources; if filename and extension are both set, result (last line of code) will be downloaded after script evaluation	icon
		label
		script
		mode
		args
		form
		filename
		extension
			{
			  "type": "evaluate",
			  "icon": "move",
			  "label": "Move",
			  "script": "layoutTools",
			  "mode": "combined",
			  "args": ["waypoints","move"],
			  "form": {
			    "heading": "Move",
			    "width": 100,
			    "elements": [
			      {
			        "heading": "Move X [mm]",
			        "tagName": "input",
			        "key": "moveX",
			        "defaultValue": 0
			      },
			      {
			        "heading": "Move Y [mm]",
			        "tagName": "input",
			        "key": "moveY",
			        "defaultValue": 0
			      }
			    ]
			  }
			}

[DEPRECATED] resultImport	action to upload a file whose content will be passed as argument along with selected resource IDs to the specified script; action is deprecated since v4.0.9, use type evaluate instead	icon
		label
		script
		mode
		contentType
			{
			  "type": "resultImport",
			  "icon": "chip",
			  "label": "Update",
			  "script": "targetUpdater",
			  "mode": "combined",
			  "contentType": "bytes"
			}

[DEPRECATED] resultExport	action to download the result (last line of code) of a script evaluation with IDs of selected resources as argument; action is deprecated since v4.0.9, use type evaluate instead	icon
		label
		script
		mode
		filename
		extension
			{
			  "type": "resultExport",
			  "icon": "download",
			  "label": "Export",
			  "script": "licenseExport",
			  "mode": "individual",
			  "filename": "license",
			  "extension": "txt"
			}

mapAim	action to find a resource on the map; map view will be zoomed and centered to focus on resource; resource needs to have model and position in order to be displayed on map	icon
		label
			{
			  "type": "mapAim",
			  "icon": "find",
			  "label": "Find"
			}

io	action to open live view of current I/O state of selected resource; resource needs to have rawInput.data and rawOutput.data fields as byte array	icon
		label
			{
			"type": "io",
			"icon": "io",
			"label": "I/O"
			}

remote	action to open remote control window for selected resource; will set remote field that contains control input for script evaluation	icon
		label
			{
			"type": "remote",
			"icon": "remote",
			"label": "Remote"
			}

Field Descriptions

field	**description**
icon	icon of action entry displayed in dropdown
label	label of action entry displayed in dropdown
opposite	opposite label of action entry, e.g. hide → unhide; will show if value of keyed field of all selected resources is boolean true
key	name of resource field that should be linked with the action
warning	warning message that shows inside yes-no-popup before executing action; may contain placeholders: {0} for number of selected resources, {1} for resource type
script	ID of script to use for action; entry will not show if script is not available; user must have UPDATE access for scripts but script may be locked; last line of script code will be displayed in result popup after running script
mode	script evaluation mode; possible values: individual → runs one task of script for each selected resource parallel combined → runs one task of script and passes all selected resource IDs
args	array of custom arguments to be passed to script, e.g. ["myArg1", "myArg2"]; additional arguments depending on type and other fields will be appended
form	definition of one-column-form for data input before script evaluation as JSON object; input values will be passed to script in args array as input object with user defined keys; possible fields: heading → title of form width → width of input column of form in pixels entries → array of input elements; possible fields: heading → title of input element, displayed above key → key of value in input object defaultValue → value of input element when form is opened placeholder → text shown inside input element if no input contentType → for dropBox; type of uploaded file; possible values: bytes → file content as raw byte array string → file content as string objects → file content as JSON object or array [custom field] → additional fields will be set as element attributes tagName → HTML tag name of input element; possible values: input → standard HTML input element list-select → dropdown list select; takes *options* array date-select → dropdown calendar date select time-select → time select with [HH:mm:ss] drum rolls color-select → RGBA color select with sliders drop-box → drop file to upload content
filename	name of file to be downloaded after export action, e.g. **report**.txt

extension	extension of file to be downloaded after export action, e.g. report.**txt**

Table Buttons [settings/tableButtons]

By default there is a button in the header bar above the map to open the resource table for each resource type that has at least one resource. The setting `tableButtons` let you customize the display properties of these buttons. `tableButtons` contains a `display` object. This object takes a key-value-pair for each resource type:

JSON example of display object of setting tableButtons

```
{
    "vehicles": "filled",
    "stations": "filled",
    "jobs": "filled",
    "scripts": "filled",
    "waypoints": "filled",
    "links": "filled",
    "zones": "filled",
    "models": "filled",
    "envelopes": "filled",
    "users": "filled",
    "internals": "never"
}
```

The entry order in this object determines the order of the buttons inside the header bar. Buttons for unlisted resource types will be appended at the end but before the icon buttons. The value of each entry determines if a button should be displayed (`filled`) or not (`never`). This does not relate to user access settings, so regardless of this setting, a user will not see a button if he has no READ access for the corresponding resource type. In the other case a user still has READ access - so he can see resources on the map - even if the table button is set to `never`.

Icons

check	cross	dropdown	more	boxes
edit	trash	upload	download	open
find	zoomin	zoomout	clockwise	counterclockwise
merge	split	move	mirror	select
front	back	resetview	undo	FN fn
circle	rectangle	polygon	bezier	clear
waypoints	zone	user	gear	crosshair
lock	key	eye	bell	popup
clock	heart	bulb	mail	pin
alert	hand	closure	stop	plus
wifi	chip	io	LOG log	DXF dxf
remote	rocket	skull	traffic	loop