Assist Visual Editor Gadgets

Aptean Ltd
Copyright © 2011-2025.

Contents

1 Assist Visual Editor Gadgets

1.1 Resources

- https://www.mediawiki.org/wiki/VisualEditor/Gadgets
- https://www.mediawiki.org/wiki/Extension:Gadgets#Usage
- https://www.mediawiki.org/wiki/VisualEditor/Gadgets/Add_a_tool

List of common gadgets

- https://meta.wikimedia.org/wiki/Gadgets
- https://meta.wikimedia.org/wiki/Wikimedia_Blog/Drafts/VisualEditor_gadgets
- https://www.mediawiki.org/wiki/VisualEditor/Gadgets

List of common snippets

- https://www.mediawiki.org/wiki/Snippets

Example of replace:

- https://en.wikipedia.org/wiki/User:%D7%A2%D7%A8%D7%9F/veReplace.js
- https://en.wikipedia.org/wiki/User:%D7%A2%D7%A8%D7%9F/veReplace

First, the gadgets extension must be installed - see Extension:Gadgets.

In general, you have to

- create the Gadget loader js with a unique name e.g. MediaWiki:Gadget-veNAMELoader.js

```
mw.libs.ve.addPlugin('ext.gadget.veNAME');
```

- Add a description to the loader e.g. MediaWiki:Gadget-veNAMELoader

```
Adds SOME FUNCTIONALITY to VisualEditor
```

- Create the gadget js e.g. MediaWiki:Gadget-veNAME.js
- Actually load the gadgets into MediaWiki:Gadgets-definition

```
* veNAMELoader[ResourceLoader|default|dependencies=ext.visualEditor.desktopArticleTarget.init]|veNAMELoader
* veNAME[ResourceLoader|default|rights=hidden|hidden|dependencies=ext.visualEditor.core]|veNAME.js
```

1.2 Examples

1.2.1 Center

- MediaWiki:Gadget-veCenterLoader.js

```
mw.libs.ve.addPlugin('ext.gadget.veCenter');
```

- MediaWiki:Gadget-veCenterLoader

```
Add Center to the Format list in VisualEditor
```

- MediaWiki:Gadget-veCenter.js

```
mw.loader.using( [ 'ext.visualEditor.core', 'ext.visualEditor.mwtransclusion' ] ).then(function () {

// --------- (start of ve.ui.CenterAction definition) ----------------------------------------------
// This is based on [lib/ve/src/ui/actions/ve.ui.BlockquoteAction.js] from Extension:VisualEditor.

        ve.ui.CenterAction = function VeUiCenterAction() {
                ve.ui.CenterAction.super.apply( this, arguments );
        };
```


Ready for What's Next, Now™

```
	OO.inheritClass( ve.ui.CenterAction, ve.ui.Action );

	ve.ui.CenterAction.static.name = 'center';
	ve.ui.CenterAction.static.methods = [ 'wrap', 'unwrap', 'toggle' ];

	ve.ui.CenterAction.prototype.isWrapped = function () {
		var fragment = this.surface.getModel().getFragment();
		return fragment.hasMatchingAncestor( 'center' );
	};
	ve.ui.CenterAction.prototype.toggle = function () {
		return this[ this.isWrapped() ? 'unwrap' : 'wrap' ]();
	};
	ve.ui.CenterAction.prototype.wrap = function () {
		var
			surfaceModel = this.surface.getModel(),
			selection = surfaceModel.getSelection(),
			fragment = surfaceModel.getFragment( null, true ),
			leaves, leavesRange;

		if ( !( selection instanceof ve.dm.LinearSelection ) ) {
			return false;
		}

		leaves = fragment.getSelectedLeafNodes();
		leavesRange = new ve.Range(
			leaves[ 0 ].getRange().start,
			leaves[ leaves.length - 1 ].getRange().end
		);
		fragment = surfaceModel.getLinearFragment( leavesRange, true );

		fragment = fragment.expandLinearSelection( 'siblings' );

		while (
			fragment.getCoveredNodes().some( function ( nodeInfo ) {
				return !nodeInfo.node.isAllowedParentNodeType( 'center' ) || nodeInfo.node.
			} )
		) {
			fragment = fragment.expandLinearSelection( 'parent' );
		}

		// Wrap everything in a <center> tag
		fragment.wrapAllNodes( { type: 'center' } );

		return true;
	};
	ve.ui.CenterAction.prototype.unwrap = function () {
		var
			surfaceModel = this.surface.getModel(),
			selection = surfaceModel.getSelection(),
			fragment = surfaceModel.getFragment( null, true ),
			leaves, leavesRange;

		if ( !( selection instanceof ve.dm.LinearSelection ) ) {
			return false;
		}

		if ( !this.isWrapped() ) {
			return false;
		}

		leaves = fragment.getSelectedLeafNodes();
		leavesRange = new ve.Range(
			leaves[ 0 ].getRange().start,
			leaves[ leaves.length - 1 ].getRange().end
		);
		fragment = surfaceModel.getLinearFragment( leavesRange, true );

		fragment
			// Expand to cover entire <center> tag
			.expandLinearSelection( 'closest', ve.dm.CenterNode )
			// Unwrap it
			.unwrapNodes( 0, 1 );

		return true;
	};
	ve.ui.actionFactory.register( ve.ui.CenterAction );

// --------- (end of ve.ui.CenterAction definition) ----------------------------------------------
```


Ready for What's Next, Now™

```
	ve.ui.CenterFormatTool = function VeUiCenterFormatTool() {
		ve.ui.CenterFormatTool.super.apply( this, arguments );
	};
	OO.inheritClass( ve.ui.CenterFormatTool, ve.ui.FormatTool );

	ve.ui.CenterFormatTool.static.name = 'center';
	ve.ui.CenterFormatTool.static.group = 'format';
	ve.ui.CenterFormatTool.static.title = 'Center';
	ve.ui.CenterFormatTool.static.format = { type: 'center' };
	ve.ui.CenterFormatTool.static.commandName = 'center';
	ve.ui.toolFactory.register( ve.ui.CenterFormatTool );

	ve.ui.commandRegistry.register(
		new ve.ui.Command(
			'center', 'center', 'toggle',
			{ supportedSelections: [ 'linear' ] }
		)
	);
	ve.ui.triggerRegistry.register(
		'center', {
			mac: new ve.ui.Trigger('cmd+j'),
			pc: new ve.ui.Trigger('ctrl+j')
		}
	);
} );
```

- MediaWiki:Gadgets-definition

```
*veCenterLoader[ResourceLoader|dependencies=ext.visualEditor.desktopArticleTarget.init]|veCenterLoader.js
*veCenter[ResourceLoader|rights=hidden|hidden|dependencies=ext.visualEditor.core]|veCenter.js
```

1.2.2 Replace - not required - already enabled in latest VE

- MediaWiki:Gadget-veReplaceLoader.js

```
mw.libs.ve.addPlugin('ext.gadget.veReplace');
```

- MediaWiki:Gadget-veReplaceLoader

```
Adds replace button to VisualEditor
```

- MediaWiki:Gadget-veReplace.js

```
/* Translate the following to your language: */
mw.loader.using('ext.visualEditor.core').then(function () {

if (!mw.messages.exists( 've-SearchAndReplaceDialog-title' )) {
	mw.messages.set({
		've-SearchAndReplaceDialog-title': 'Search and replace',
		've-SearchAndReplaceDialog-from-label': 'From:',
		've-SearchAndReplaceDialog-to-label': 'To:',
		've-SearchAndReplaceDialog-from-placeholder': 'From text',
		've-SearchAndReplaceDialog-to-placeholder': 'To text',
		've-SearchAndReplaceDialog-replaceAll': 'Replace all',
		've-SearchAndReplaceDialog-replace': 'Replace',
		've-SearchAndReplaceDialog-matchcase': 'Match case',
		've-SearchAndReplaceDialog-replace-complete': 'Found and replaced $1 occurrences',
		've-ReplaceTool-ToolbarButton': 'Replace'
	});
}
/* end of translations */

/*!
 * VisualEditor replace gadget
 *
 * @copyright [[User:???|Eranroz]] and [[User:Ravid ziv|Ravid ziv]]
 * @license The MIT License (MIT)
 */
function extractText(){
	var nodes = [];
	var model = ve.init.target.getSurface().getModel();
	function getTextNodes( obj ) {
```


Ready for What's Next, Now™

```
                var i;

                for ( i = 0; i < obj.children.length; i++ ) {
                        if ( obj.children[i].type == 'text'){
                                nodes.push(obj.children[i]);
                        }

                        if ( obj.children[i].children ) {
                                getTextNodes( obj.children[i] );
                        }
                }
        }
        getTextNodes(ve.init.target.getSurface().getModel().getDocument().getDocumentNode());
        return nodes;
}

function searchAndReplace( fromText, toText, replaceAll, matchCase ) {
        var textNodes = extractText();
        var model = ve.init.target.getSurface().getModel();
        var firstIndex = 0;
        var numReplacements = 0;
        for (var nodeI = 0; nodeI < textNodes.length; nodeI++) {
                var node = textNodes[nodeI];
                var nodeRange = node.getRange();
                var nodeText = model.getLinearFragment(nodeRange).getText();

                var fromIndex = matchCase? nodeText.toUpperCase().indexOf( fromText.toUpperCase(), firstInd
                if ( fromIndex == -1 ) {
                        firstIndex = 0;
                        continue;
                }
                var start = nodeRange.from+fromIndex;
                var end = start+fromText.length;
                if (!replaceAll && model.selection.start > start) {
                        continue;//skip replacements before selection
                }
                var removeRange = new ve.Range( start, end );
                var transaction = ve.dm.Transaction.newFromReplacement(
                        ve.init.target.getSurface().getView().getDocument().model,
                        removeRange,
                        toText
                );
                var newSelection = new ve.Range(0,0);
                if (!replaceAll) {
                        newSelection = new ve.Range( start, start+toText.length );
                }
                ve.init.target.getSurface().getView().changeModel(transaction, newSelection);
                numReplacements++;
                if (!replaceAll) {
                        break;
                }
                firstIndex = fromIndex + toText.length;
                nodeI = nodeI -1;
        }
        if (numReplacements==0 || replaceAll) {
                mw.notify( mw.msg( 've-SearchAndReplaceDialog-replace-complete', numReplacements ) );
        }
}

ve.ui.SearchAndReplaceDialog = function( manager, config ) {
        // Parent constructor
        ve.ui.SearchAndReplaceDialog.super.call( this, manager, config );

};
/* Inheritance */

OO.inheritClass( ve.ui.SearchAndReplaceDialog, ve.ui.FragmentDialog );

ve.ui.SearchAndReplaceDialog.prototype.getActionProcess  = function ( action ) {
        var fromVal = this.fromInput.getValue(),
                toVal = this.toInput.getValue(),
                matchCase = this.matchCaseCheckbox.getValue();

        if ( action === 'replace' ) {
                return new OO.ui.Process( function () {
                        searchAndReplace( fromVal, toVal, false, matchCase );
                }, this );
        } else if ( action === 'replace-all' ) {
```


Ready for **What's Next, Now**™

```
                return new OO.ui.Process( function () {
                        searchAndReplace( fromVal, toVal, true, matchCase );
                        this.close( );
                }, this );
        }
        return ve.ui.MWMediaDialog.super.prototype.getActionProcess.call( this, action );
};

ve.ui.SearchAndReplaceDialog.prototype.getBodyHeight = function () {
        return 200;
};

/* Static Properties */
ve.ui.SearchAndReplaceDialog.static.name = 'search';
ve.ui.SearchAndReplaceDialog.static.title = mw.msg( 've-SearchAndReplaceDialog-title' );
ve.ui.SearchAndReplaceDialog.static.size = 'medium';

ve.ui.SearchAndReplaceDialog.static.actions = [
        {
                'action': 'replace',
                'label': mw.msg( 've-SearchAndReplaceDialog-replace' ),
                'flags': [ 'constructive' ],
                'modes': 'insert'
        },
        {
                'label': OO.ui.deferMsg( 'visualeditor-dialog-action-cancel' ),
                'flags': 'safe',
                'modes': [ 'edit', 'insert', 'select' ]
        },
        {
                'action': 'replace-all',
                'label': mw.msg( 've-SearchAndReplaceDialog-replaceAll' ),
                'flags': [ 'constructive' ],
                'modes': 'insert'
        }
];

ve.ui.SearchAndReplaceDialog.prototype.initialize = function () {
        ve.ui.SearchAndReplaceDialog.super.prototype.initialize.call( this );
        this.panel = new OO.ui.PanelLayout( { '$': this.$, 'scrollable': true, 'padded': true } );
        this.inputsFieldset = new OO.ui.FieldsetLayout( {
                '$': this.$
        } );
        // input from
        this.fromInput = new OO.ui.TextInputWidget(
                { '$': this.$, 'multiline': false, 'placeholder': mw.msg( 've-SearchAndReplaceDialog-from-p
        );
        //input to
        this.toInput = new OO.ui.TextInputWidget(
                { '$': this.$, 'multiline': false, 'placeholder': mw.msg( 've-SearchAndReplaceDialog-to-pla
        );
        this.fromField = new OO.ui.FieldLayout( this.fromInput, {
                '$': this.$,
                'label': mw.msg( 've-SearchAndReplaceDialog-from-label' )
        } );
        this.toField = new OO.ui.FieldLayout( this.toInput, {
                '$': this.$,
                'label': mw.msg( 've-SearchAndReplaceDialog-to-label' )
        } );


        this.matchCaseCheckbox = new OO.ui.CheckboxInputWidget( {
                '$': this.$
        } );
        var matchCaseField = new OO.ui.FieldLayout( this.matchCaseCheckbox, {
                '$': this.$,
                'align': 'inline',
                'label': mw.msg( 've-SearchAndReplaceDialog-matchcase' )
        } );

        this.inputsFieldset.$element.append(
                this.fromField.$element,
                this.toField.$element,
                matchCaseField.$element
        );
        this.panel.$element.append(     this.inputsFieldset.$element );
        this.$body.append( this.panel.$element );
```


Ready for **What's Next, Now**™

```
};

ve.ui.windowFactory.register( ve.ui.SearchAndReplaceDialog );

//---------- replace tool -----------------

function ReplaceTool( toolGroup, config ) {
        OO.ui.Tool.call( this, toolGroup, config );

}
OO.inheritClass( ReplaceTool, OO.ui.Tool );

ReplaceTool.static.name = 'ReplaceTool';
ReplaceTool.static.title = mw.msg('ve-ReplaceTool-ToolbarButton');

ReplaceTool.prototype.onSelect = function () {
        this.toolbar.getSurface().execute( 'window', 'open', 'search', null );
};

ReplaceTool.prototype.onUpdateState = function () {
        this.setActive( false );
};

ve.ui.toolFactory.register( ReplaceTool );
});
```

- MediaWiki:Gadgets-definition

```
*veReplaceLoader[ResourceLoader|dependencies=ext.visualEditor.viewPageTarget.init]|veReplaceLoader.js
*veReplace[ResourceLoader|rights=hidden|hidden|dependencies=ext.visualEditor.core]|veReplace.js
```

Note: may need to be desktopArticleTarget for the loader, as follows:

```
*veReplaceLoader[ResourceLoader|dependencies=ext.visualEditor.desktopArticleTarget.init]|veReplaceLoader.js
```

1.2.3 Autonumber Headings

The settings for autonumbering of sections has been removed from user preferences in later versions.

The following re-enables it as a gadget.

MediaWiki:Gadgets-definition:

```
* autonum[ResourceLoader]|autonum.css|autonum.js
```

MediaWiki:Gadget-autonum

```
Allow ability to auto-number headings when viewing pages (replacing setting removed from MediaWiki v1.39)
```

MediaWiki:Gadget-autonum.js

```
/**
 * Auto-number headings
 *
 * @source https://www.mediawiki.org/wiki/Snippets/Auto-number_headings
 * @author Krinkle
 * @version 2024-07-28
 */
var toc = document.querySelector('#toc');
if (toc) {
  document.body.classList.add('tpl-autonum-attr');
  // Support legacy Parser: <h2><span class=mw-headline id=?>
  // Support Parsoid: <section><div class=mw-heading><h2 id?>
  document.querySelectorAll('.mw-parser-output :is(h1,h2,h3,h4,h5,h6) .mw-headline[id], .mw-parser-output .
    var num = toc.querySelector('a[href="#' + CSS.escape(headline.id) + '"] .tocnumber');
    if (num) headline.setAttribute('data-autonum', num.textContent);
  });
} else {
  document.body.classList.add('tpl-autonum');
}
```


Ready for What's Next, Now™

MediaWiki:Gadget-autonum.css

```
/**
 * Auto-number headings
 *
 * @source https://www.mediawiki.org/wiki/Snippets/Auto-number_headings
 * @author Krinkle
 * @version 2024-07-28
 */

/**
 * CSS mode:
 * Insert numbers on pages without a TOC. This could in principle work for all pages,
 * but to ensure consistency between the TOC and heading numbers we let JS follow
 * the TOC if there is one.
 */
.tpl-autonum .mw-parser-output {
  counter-reset: autonum-h2 autonum-h3 autonum-h4 autonum-h5 autonum-h6;
}
.tpl-autonum .mw-parser-output h2 {
  counter-reset: autonum-h3 autonum-h4 autonum-h5 autonum-h6;
}
.tpl-autonum .mw-parser-output h3 {
  counter-reset: autonum-h4 autonum-h5 autonum-h6;
}
.tpl-autonum .mw-parser-output h4 {
  counter-reset: autonum-h5 autonum-h6;
}
.tpl-autonum .mw-parser-output h5 {
  counter-reset: autonum-h6;
}
.tpl-autonum .mw-parser-output h2 .mw-headline:before,
.tpl-autonum .mw-parser-output .mw-heading h2:before {
  counter-increment: autonum-h2;
  content: counter(autonum-h2) " ";
}
.tpl-autonum .mw-parser-output h3 .mw-headline:before,
.tpl-autonum .mw-parser-output .mw-heading h3:before {
  counter-increment: autonum-h3;
  content: counter(autonum-h2) "." counter(autonum-h3) " ";
}
.tpl-autonum .mw-parser-output h4 .mw-headline:before,
.tpl-autonum .mw-parser-output .mw-heading h4:before {
  counter-increment: autonum-h4;
  content: counter(autonum-h2) "." counter(autonum-h3) "." counter(autonum-h4) " ";
}
.tpl-autonum .mw-parser-output h5 .mw-headline:before,
.tpl-autonum .mw-parser-output .mw-heading h5:before {
  counter-increment: autonum-h5;
  content: counter(autonum-h2) "." counter(autonum-h3) "." counter(autonum-h4) "." counter(autonum-h5) " ";
}
.tpl-autonum .mw-parser-output h6 .mw-headline:before,
.tpl-autonum .mw-parser-output .mw-heading h6:before {
  counter-increment: autonum-h6;
  content: counter(autonum-h2) "." counter(autonum-h3) "." counter(autonum-h4) "." counter(autonum-h5) "."
}

/**
 * JS mode: When a TOC is present, autonum.js sets the data-autonum attribute
 * and we display that instead of an automatic counter.
 */
.tpl-autonum-attr .mw-parser-output .mw-headline[data-autonum]:before,
.tpl-autonum-attr .mw-parser-output h2[data-autonum]:before ,
.tpl-autonum-attr .mw-parser-output h3[data-autonum]:before ,
.tpl-autonum-attr .mw-parser-output h4[data-autonum]:before ,
.tpl-autonum-attr .mw-parser-output h5[data-autonum]:before ,
.tpl-autonum-attr .mw-parser-output h6[data-autonum]:before {
  content: attr(data-autonum) " ";
}
```


Ready for **What's Next, Now**™