

Instructions

# Sellector

# Client Templating

Version 01 28.09.2010 / creation draft

Version 02 14.10.2010 / amendments

Version 03 10.11.2010 / amendments

Version 04 28.12.2010 / new functionalities - major release

Version 05 11.01.2011 / inputAreaTemplating

Version 06 18.08.2011 / new functions - major release

Copyright Sellector GmbH

All rights reserved

# Contents

[General Description](#)

[Tuning the Presentation](#)

[Result pane](#)

[General adjustment of the result pane](#)

[Tuning the product presentation](#)

[Simple footer ind product presentation](#)

[Template for product presentation](#)

[Template for detailed product description \(when hovering\)](#)

[Selection pane](#)

[Presentation of the selection area](#)

[Flexible Positioning of the Selection Elements](#)

[Presentation of the selection elements](#)

[Input field for text search](#)

[Preset of selection elements](#)

[Special Features](#)

[open product page - target:](#)

[Automatic result paging - slide show](#)

[Animation when paging](#)

[Dynamically generated destination URLs - affilitatMode](#)

[Bookmark-able selection state and browser back button](#)

[General Notice](#)

[Typical Problems with Javascript Templating](#)

[Special Characters](#)

# General Description

Within the Selector product data and presentation are handled separately. While all product data is updated and maintained on the Selector-Server using the website [www.selector.com](http://www.selector.com), the presentation of selection elements and results is configured directly within the target web page.

This allows for using the same Selector data in different formats e.g. on different web sites, differently preset landing pages or in different steps of the purchase process.

The configuration of presentation is done via the Client Templating which is described in this document. Through Client Templating you are given the means to adapt the presentation of a Selector within very wide limits. Thus, the appearance of the results and the presentation of the selection elements can be customized to your needs. In addition special features such as slide-show function, browser-back-button or affiliate mode can be activated.

# Adjusting the Presentation

Together with the standard code snippet for activating the Selector (www.selector.com -> Login -> Create -> Integrate) the following template functions to adjust the presentation of the selection and result pane are specified within the java script variable `selectorPresentation` as follows:

```
<script type="text/javascript" src="http://www.selector.com/scripts/selectorControl.js"></script>
<script type="text/javascript">
  var selectorPresentation={
    // place variables to adapt presentation of
    // selection and result panes here
  };
  selectorLoader( <ENTER YOUR SELECTOR ID>, "InputArea", "ResultArea")
</script>
```

# Result pane

## General adjustment of the result pane

```
resultAreaTemplate:{
  backgroundColor: '#ffc050', // or 'transparent'
  backgroundImage: '', // URL
  backgroundRepeat: '', // no-repeat, repeat-y, repeat-x, repeat
  headerHeight: 20,
  headerContent: 'Your HTML and Selector tags see below',
  footerHeight: 15,
  footerContent: 'Your HTML and Selector tags see below',
  arrowImageLeft: 'http://www.something.com/arrowLeft.png',
  arrowImageRight: 'http://www.something.com/arrowRight.png',
  arrowImageWidth: 20, // in px
  arrowImageHeight: 20,
  arrowTransparency: 100, // in %
  arrowMarginTop: 300, // in pixel from top
  arrowMarginLR: 0 // in pixel from left/right
},
```

Selector tags for use within the headerContent and footerContent:

```
<productsMatching/>
<page/>
<pagesAvailable/>
<lastUpdatedDDMMYYYY/>
<lastUpdatedYYYYMMDD/>
<lastUpdateTime/>
```

## Designing the product presentation

### Simple footer for product presentation

```
resultTileFooter: {
  height: 12,
  content: 'Your HTML and Selector tags see below'
},
```

Selector tags for use within content:

```
<selectorPrice/>
<selectorPriceDecimal/>
```

Attention: When resultTileTemplate ( see below) is present, resultTileFooter will be disregarded.

## Template for product presentation

(overrides resultTileFooter above):

```
resultTileTemplate: {
  width: 150,                // width of the result tile
  height: 110,              // height of the result tile
  spacing: 30,              // distance between the tiles
  imgWidth: 75,             // width of the image
  imgHeight: 50,           // height of the image
  clickable 'tile',        // or 'image'
  background-color: '#ffc050', // or 'transparent'
  backgroundImage: '',     // URL
  backgroundRepeat: '',    // no-repeat, repeat-y, repeat-x, repeat
  content: 'your HTML and Selector tags see below'
  // content contains the HTML content for the result tile
},
```

Selector tags for use within content:

```
<selectorProductName/>
<selectorProductId/>
<selectorCustomProductText/>
<selectorSmallImage/>
<selectorDescription/>
<selectorDestination/>
<selectorPrice/>
<selectorPriceDecimal/>
```

## Template for detailed product description (when hovering)

(overrides resultTileFooter above):

```
detailsTileTemplate: {
  width: 500,
  height: 400,
  imgWidth: 400,
  imgHeight: 300,
  backgroundColor: '#ffffff', // or transparent
  backgroundImage: '',       // URL
  backgroundRepeat: '',     // no-repeat, repeat-y, repeat-x, repeat
  shadow: false,           // suppresses the details shadow
  content: 'Your HTML and Selector-Tags see below'
},
```

Selector tags for use within content:

```
<selectorProductName/>
<selectorProductId/>
<selectorCustomProductText/>
<selectorSmallImage/>
<selectorLargeImage/>
<selectorDescription/>
<selectorDestination/>
```

```
<selectorPrice/>
<selectorPriceDecimal/>
```

## Selection pane

### Presentation of the selection area

```
inputAreaTemplate: {
    backgroundColor: '#ffc050', // or transparent
    matchCountForAll: true     // activates the match count for all
}                             // selection elements (except sliders)
```

Notice:

The match count determines for the current selection (state of all selection elements), how many matching products feature each selection element's attribute (except for sliders).

Example: Brand

- Sony (20)
- Panasonic (5)
- Aiwa (2)

Besides the above mentioned means using the general `matchCountForAll` (activates match count for all selection elements) it is possible to set and configure the match count separately for each selection element in `choicePresentation` (see below). Doing so it is also possible to set how to handle (hide, disable, etc.) selection elements with no matching product (match count is '0').

### Flexible Positioning of the Selection Elements

Alternatively to the standard positioning of the selection pane selection elements can also be distributed over multiple, independent selection panes.

This allows for a very flexible positioning of the selection elements. Thus selection can be positioned horizontally, in separate locations of the webpage or even be displayed sequentially allowing for step-by-step configurators.

Initially for each intended selection pane the web site's HTML code has to be added with a HTML div element at the desired position. Each of these div elements must feature an individual id ('id' attribute within HTML tag) and, if the pane is not displayed on Selector startup, additionally must be equipped with a 'width' and a 'height' attribute set to absolute values. Then each of the Selector's selection elements are assigned to the above created selection pane boxes using the variable `element` within `box` within `inputAreaTemplate` as shown in the following example.

```

inputAreaTemplate: {
  box: [
    {
      divId: "selectionarea_1",
      element:[
        {
          name: 'Price'
        },
        {
          name: 'Manufacturer'
        },
        {
          name: 'Size'
        }
      ]
    },
    {
      divId: "selectionarea_2",
      element:[
        {
          name: 'Weight in kg'
        },
        {
          name: 'Warranty'
        }
      ]
    }
  ]
}

```

The selection elements can be identified using `name` by their selector table column header.

The order of the selection elements within the above shown `element` list determines the presentation order within the given selection pane.

The width of the selection elements (e.g. slider, drop down, click panel) is defined by the width of the corresponding selection pane HTML div element. If selection elements shall be positioned horizontally within a selection pane, the selection element width must be set using the optional parameter `singleElementWidth` (see below). Except for check groups and click panels there will be NO line feed after each element and the following element will be placed to the right of the previous element.

```

inputAreaTemplate: {
  box: [
    {
      divId: "selectionpane_3",
      singleElementWidth: 170,
      element:[

```

```

        {
            name: 'Price'
        },
        {
            name: 'Manufacturer'
        }
    ]
}
]
}

```

#### Note:

If a selection pane is defined using the variable `box` the standard selection pane set in the `selectorLoader` function call will be ignored.

### Presentation of the selection elements

Change over between `dropDown`, clickable graphics, and check group, as well as between single check box and clickable graphics:

```

inputAreaTemplate: {
    choicePresentation: [
        {
            // required Parameters
            name: 'Mobile phone type', // name of selection element
            type: 'clickPanel', // type of presentation
            // The type of presentation can be changed here
            // Selection groups can be displayed as follows:
            // 'clickPanel', 'multipleClickPanel',
            // 'checkGroup', 'multipleCheckGroup',
            // 'dropDown'
            // Singel check boxes can be displayed as follows:
            // 'clickElement', 'checkBox',
            // Numeric Sliders can be displayed as follows:
            // 'numRangeSlider', 'numMinSlider', 'numMaxSlider'
            // The input field for the text search can be activated with:
            // 'searchBox'
            // general settings for selection element
            screenName: '<span title="Please select hier">Manufacturer</span>',
                // Replaces the name of the selection element
                // (column header within Selector table) with HTML
            screenNameStyle: '',
                // Sets the Inline-Style of the screen name
                // '' removes all Inline-Style (for CSS usage)
            sort: 'list', // up, down, list
            sortList: [
                'Bar type',
                'PDA'
            ]
        }
    ]
}

```

```

]
noMatchHighlight: 'disable', // 'disable', 'hide', 'fade'
                                // disable not available for clickPanels
noMatchOpacity: 50,           // if noMatchHighlight: 'fade'
                                // not available for dropDown
matchCountContent: ' (<inputElementMatchCount/>)',
                                // Presentation of Match-Count
                                // Use default setting for matchCountContent: true
// Selector tags for use within matchCountContent:
//   <inputElementMatchCount/>           Number

// special settings for dropDown elements
defaultText: '',              // replaces '---' in dropDown
defaultTextColor: '#888',

// special settings for slider elements
sliderDesign: { // customize appearance of slider
  height: 20,           // overa height of slider
  handleRight: {
    width: 11,
    height: 20,
    image: "http:www.images.com/sliderHandles.png",
    imageOffsetX: -11,
    imageOffsetY: 0
  },
  handleLeft: {
    width: 11,
    height: 20,
    image: "http:www.images.com/sliderHandles.png",
    imageOffsetX: 0,
    imageOffsetY: 0
  },
  trackActive:{
    height: 10,
    image: "http:www.images.com/sliderTrack.png",
    imageOffsetX: -10,
    imageOffsetY: 0,
    color: "#a03422"
  },
  trackInactive:{
    height: 6,
    image: "http:www.images.com/sliderTrack.png",
    imageOffsetX: 0,
    imageOffsetY: 0,
    color: "#ccc"
  },
  colorBorder: "#ccc",

```

```

    colorHandle: "#a00",
    trackOffsetX: 10,
  }
  sliderValueUnit: " €", // unit symbol behind slider values
  sliderSingleUnit: true, // only one unit symbol

// special settings for numeric slider elements
  decimalDigits: 2, // umber of decimal digits
  step: 0.1, // minimum step width of slider
  lockStep: true, // snap in for slider handles
  samplePoint: [ // for non linear slider tracks
    {
      trackPosition: 9, // Position on slider track in %
      value: 1000 // Value at given position
    },
    {
      trackPosition: 90,
      value: 4000
    }
  ],

// special settings for clickPanel and clickElement
  width: 32, // width of click element
  height: 32, // height of click element
  content: 'Your HTML and Selector tag <selectorCellContent/>',
  contentInactive: 'HTML when element is not selected',
  mapping: [ // 'mapping' überschreibt 'content'
    // Maps the values of your Selector table to corresponding
    // graphics for presentation within the clickPanel
    {
      sourceValue: 'PDA',
      displayValue: '',
      displayInactive: ''
    },
    {
      sourceValue: 'Slider phone',
      displayValue: 'Your HTML when selected...'
    }
  ],

// Selector tags for use within content, contentInactive, mapping:
//   <inputElementMatchCount/>           Number
//   <matchCountContent/>               see above
floating: 'left', // left, right, none
spacing: 3, // margin between click elements
frameColor: '#333333', // frame color around selected element
frameWidth: 2, // thickness of frame
opacity: 50, // transparency for unselected elements
// if highlight is set to 'fade'
highlight: 'frame' //'frame', 'fade'

```

```
        // (highlight is overridden by 'mapping.displayInactive')
    }
}
]
```

### **Input field for text search**

The text search is activated by an additional entry in `choicePresentation` by setting `name` to a free definable term and using the following type: `'searchBox'`.

The input field for the search will then be displayed at the end of the input pane. To place the search field in any other position the above discussed `box` variable can be utilized by referencing the search field with the earlier defined name.

# Preset of selection elements

(On Selector start-up)

When including the name of a drop down or check box into selectorPreset will set this very element to be pre-selected. The name of the element comprises of the name of the checkGroup or the dropDown (the name of the corresponding column in the selector table) followed by a vertical bar | (pipe symbol) followed by the name of the element to be selected (e.g. 'Mobile phone type|slider phone').

With range sliders the parameters lowDefault, highDefault, min and max can be specified. Min and max define the beginning and end of the slider track while lowDefault and highDefault set the initial position of the control knobs on the track.

```
var selectorPreset = {
    presetValue: [
        {name: 'height', lowDefault:811, highDefault:4005}, //min and max
        {name: 'price', lowDefault:5, highDefault:20},
        {name: 'place|garden'}
    ]
};
```

# Special Features

The following special features are declared within the following JS variable:

```
var selectorFeature = {
    // variables for activation and control of special functions
};
```

## open product page - target:

```
openResultTarget: '_blank', // _blank, _parent, _self, _top
```

## Automatic result paging - slide show

```
slideShow:{
    wait: 6,           // wait time between two Selector result pages
    restartsec: 15    // restart the slide show after last entry in sec
    maxPages: 10      // Number of pages Anzahl for slide show
                    // e.g. only first X pages with top products
}
```

## Animation when paging

```
paging: "slide", // shift, fade, slide
```

## Dynamically generated destination URLs - affiliateMode

```
affiliatePattern: "<selectorDestination/>?affiliateID=<affiliateId/>",  
affiliateID: 296270
```

Selector tags for use within affiliatePattern:

```
<affiliateId/>
```

Notice:

To use the affiliate mode is a booking / activation required in the backend.

## Bookmark-able selection state and browser back button

The current selection state of the Selector can be added to browser URL field to enable bookmarking or link copy-pasting for the Selector web page including the current selection. Optionally the selection state can also be added to the product destination URL, for instance in order to display on the product target page a Selector with the current settings. Also the browser navigation buttons (back and forward) can be activated to navigate through the selection history.

```
selectStateUrl:{                                // adds selection state to URL field  
    addToDestination: true,                    // adds selection state to destination  
URL      enableBackButton: true                // activates browser navigation  
buttons  
}
```

# General Notice

## Typical Problems with Javascript Templating

Selector templating uses the web browser independent script language Javascript. The templates within this document are written in the Javascript Object Notation JSON. Some web browsers are very strikt when interpreting Javascript and JSON. One erroneously set comma can already cause that the web page is not or not completely displayed.

Here are two easy rules that help to avoid most problems:

1. **Never put a comma (',') before any closing bracket ('}' oder ']')**
2. **Indent elements within brackets.**  
**Indent your code whenever you set a bracket. This way the structure remains clear and mistakes are easy to spot.**

## **Special Characters**

When using special characters within your Selector table and the Selector templating, make sure the web page is encoded in the UTF-8 file format. Otherwise the web browser will receive the correct characters from the Selector server, but the characters set within the template will be interpreted incorrectly by the web browser.

While most content management and shop systems are safe since they manage their web page files in the UTF-8 format, e.g. local test pages or manually created webpages are usually stored in the windows-ansi format.

If you are using Windows OS you can use the windows editor named 'notepad to open the web page file and in then 'save as' menu change 'encoding' to 'UTF-8' and store the file.

11.01.2011 / BB