Search..

Getting started

Components

About

Population

Windows

About Create window Window caption

Buttons

lcon Title

Buttons events Shadow Customize

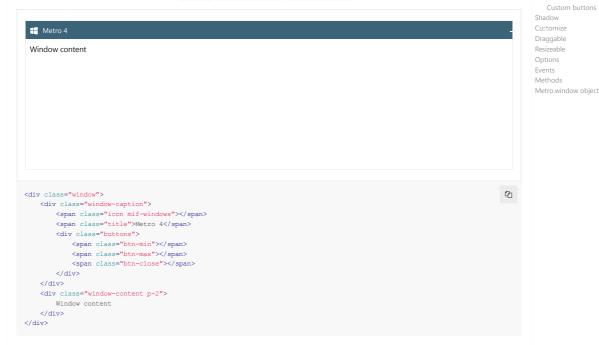
Draggable Resizeable Events

Windows

Metro 4 provides special classes to create window elements and manipulate them.

About

Metro 4 build windows with using next classes: .window, .window-caption, .window-content and result window has next structure:



Create window

To create window add attribute data-role="window" to your block element.

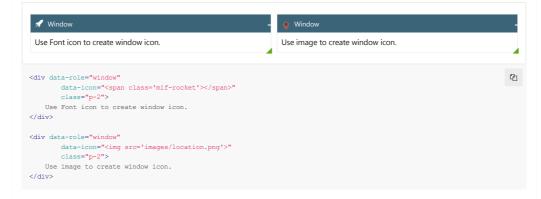


Caption

The caption may include the following elements: icon, title, system buttons

Icon

To add <code>icon</code> to window you must add attribute <code>data-icon="..."</code> to element



Title

To add title to window you must add attribute data-title="..." to element.

```
Window title
 Set the title with data-title attribute
<div data-role="window" data-title="Window title" class="p-2">
                                                                                                                                 4
    Set the title with data-title attribute.
</div>
```

Buttons

Aiax Animation Visibility Effects Subtree functions Attributes Html, text and value

Css and classes Position and size Manipulation

DataSet Events Utils

Containers Grid system Typography Tables

Buttons Images Lists

Checkbox File input Input Input materia Keypad Rating Radio Select

Slider Double Slider Spinner

Tag input Textarea

App bar Bottom navigation Bottom sheet Menu Ribbon Menu

Side bar Side navigation

Accordion Badge Carousel Cards Cube

Counter Charms

Chat

Image compare Image magnifier Gravatar List

Master NavView Panels Progress & Activity

Stepper Splitter Tabs Tabs material

Table Tiles TreeView

Dialogs Info box

Wizard

Hints Notify system Popovers

Windows

Calendar Calendar picker Date picker Time picker Countdown

Media

Video player
Audio player
Tools
Collapse
Color module
Draggable
Dropdown
Form validator
Hotkeys
Micro templates
Ripple
Storage
Session storage
Sorter

Utilities Animations Additional

Touch and swipe

You can add system buttons to your window such as: close, min and max. To create buttons you must add relevant attributes:

data-btn-close="true", data-btn-min="true", data-btn-max="true".

```
Vindow title

On/off window system buttons with attributes.

<div data-role="window" id="-window-buttons"
    data-title="Window title"
    data-btn-close="true"
    data-btn-min="true"
    data-btn-max="true"
    class="p-2">
    On/off window system buttons with attributes.
</div>

v btn-close

v btn-min

v btn-max
```

```
cinput checked type="checkbox"
   data-role="checkbox"
   data-caption="btn-close"
   onclick="%('#-window-buttons').attr('data-btn-close', %(this).is(':checked'))">

cinput checked type="checkbox"
   data-role="checkbox"
   data-role="btn-min"
   onclick="%('#-window-buttons').attr('data-btn-min', %(this).is(':checked'))">

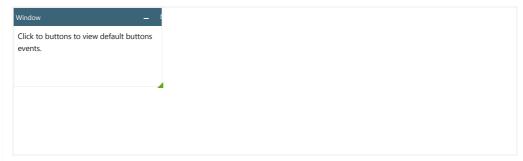
cinput checked type="checkbox"
   data-caption="btn-min"
   onclick="%('#-window-buttons').attr('data-btn-min', %(this).is(':checked'))">

cinput checked type="checkbox"
   data-role="checkbox"
   data-role="checkbox"
   data-role="checkbox"
   data-caption="btn-max"
   onclick="%('#-window-buttons').attr('data-btn-max', %(this).is(':checked'))">
```

Buttons events

When Metro 4 created window buttons, also created default event for this buttons. You can extend buttons events with custom events over special attributes: data-on-close-click="...", data-on-min-click="...", data-on-max-click="...", data-on-caption-dbl-click="...".

The default events are shown in the example below:



Custom buttons

You can add a custom buttons to window caption. To add custom buttons, use attribute data-custom-buttons

```
Click to buttons to view default buttons events.

<script>
var customButtons = {
```

```
html: "<span class='mif-rocket'></span>",
           cls: "sys-button",
           onclick: "alert('You press rocket button')"
           html: "<span class='mif-user'></span>",
           cls: "alert",
           onclick: "alert('You press user button')"
       },
           html: "<span class='mif-cog'></span>",
           cls: "warning",
           onclick: "alert('You press cog button')"
   1;
</script>
<div class="p-2"
    data-role="window"
    data-draggable="true"
    data-title="Window"
    data-btn-close="true"
    data-btn-min="true"
    data-btn-max="true"
    data-custom-buttons="customButtons"
     data-width="300" data-height="160">
   Click to buttons to view default buttons events.
```

```
</div>
Add shadow to window
You can add _{\tt shadow} to window with attribute _{\tt data-shadow="true"} .
  Window title
   Add shadow to window with attribute data-shadow
                                                                                                                                      đ
     data-title="Window title"
      data-shadow="true"
       class="p-2">
      Add shadow to window with attribute data-shadow
  </div>
  Enable shadow
Customize
You\ can\ change\ \ {\tt default\ colors}\ \ for\ \ {\tt caption}\ \ and\ \ {\tt content}\ . \ To\ set\ it\ use\ attributes\ \ {\tt data-cls-caption="..."}\ \ and\ \ {\tt data-cls-content="..."}\ .

✓ The rocket

    You can customize window with own classes
  <div class="p-2"
                                                                                                                                      đ
      data-role="window"
data-icon="<span class='mif-rocket'></spam>
      data-title="The rocket"
data-cls-caption="bg-red"
       data-cls-content="bg-dark fg-yellow">
     This is a simple DIV transformed to window with data-role.
 </div>
Draggable window
To add draggable feature to window add attribute data-draggable="true" to your element. About draggable you can read in this article.
 Drag this window on the
 caption.
                                                                                                                                      đ
       data-draggable="true"
data-width="200"
          data-height="160">
     Drag this window on the caption.
  </div>

    Enable draggable

Resizeable window
To make window as resizable add attribute data-resizable="true" to your window.
Window
 Resize the window on hold and move
 bottom right corner of the window.
                                                                                                                                     đ
  <div class="p-2"
     data-role="window"
      data-resizable="true"
      data-draggable="true"
      data-width="300"
      data-height="160">
      Resize the window on hold and move bottom right corner of the window.
```



Available options

All options you can set over data-* attribute:

Option	Data attribute	Default	Desc	Observe
width	data-width	auto	Set initial width to window	~
height	data-height	auto	Set initial height to window	~
btnClose	data-btn-close	true	Add close button	~
btnMin	data-btn-min	true	Add min button	~
btnMax	data-btn-max	true	Add max button	~
clsWindow	data-cls-window		Additional classes for window	~
clsCaption	data-cls-caption		Additional classes for caption	~
clsContent	data-cls-content		Additional classes for content	~
draggable	data-draggable	true	Add draggable to window	~
dragElement	data-drag-element	.window-caption	Set the drag element	8
dragArea	data-drag-area	parent	Set the area where element can be draggable	8
shadow	data-shadow	false	Add shadow to window	~
icon	data-icon		Window icon	~
title	data-title	Window	Window title	~
content	data-content		Window content	~
resizable	data-resizable	true	Add resizable to window	~
top	data-top	auto	Set window top position	~
left	data-left	auto	Set window left position	~
place	data-place	auto	Set window place (can be: center, top-left, top-center, top-right, right-center, bottom-right, bottom-center, bottom-left, left-center)	~
closeAction	data-close-action	Metro.actions.REMOVE	Action when window close. Metro.actions.REMOVE or Metro.actions.HIDE	8

Events

Event	Data-*	Desc
onDragStart	data-on-drag-start	Generated when the user starts dragging the window
onDragStop	data-on-drag-stop	Generated when the user stops dragging the window
onDragMove	data-on-drag-move	Generated when the user dragging the window
onCaptionDblClick	data-on-caption-dbl-click	Generated when the user double clicks on the window caption
onCloseClick	data-on-close-click	Generated when the user click close button
onMaxClick	data-on-max-click	Generated when the user click max button
onMinClick	data-on-min-click	Generated when the user click min button
onResizeStart	data-on-resize-start	Generated when the user starts resizing the window
onResizeStop	data-on-resize-stop	Generated when the user stops resizing the window
onResize	data-on-resize	Generated when the user resize the window
onWindowCreate	data-on-window-create	Generated when window was created
onDestroy	data-on-window-destroy	Generated when the window destroyed

How to use event you can see in this example. The example source you can find on github.

Methods

The window has a several methods to interact with it.

Method	Desc			
minimized()	Toggle minimized state			
maximized()	Toggle maximized state			
close()	Close window			
hide()	Add class .no-visible to window			
show()	Remove class .no-visible from window			

Metro.window object

Added in 4.2.1

You can use object Metro.window to manipulate of windows components. This object contains next methods:

- isWindow(el) checks, if element has role window
- show(el) show hidden window
- hide (el) hide window
- toggle (el) toggle show/hide window
- isOpen(el) checks, if a window is showed
- close (e1) close window
- create ({...}) create window
- min(el, bool) minimized or unminimized window
- max(el, bool) maximized or unmaximized window