Search...

Getting started
Components

M4Q
About
Population
Constructor
Ajax
Animation
Loops
Visibility
Effects
Subtree functions
Attributes
Html, text and value
Css and classes
Position and size
Manipulation
DataSet
Events
Utils

Base
Containers
Grid system
Typography
Tables
Forms
Buttons
Images
Figures
Lists

Form components
Checkbox
File input
Input
Input material
Keypad
Rating
Radio
Select
Slider
Double Slider
Spinner
Switch
Tag input
Textarea

Menus
App bar
Bottom navigation
Bottom sheet
Menu
Ribbon Menu
Side bar
Side navigation

Controls
Accordion
Badge
Carousel
Cards
Cube
Counter
Charms
Chat
Donut
Image compare
Image magnifier
Gravatar
List
ListView
Master
NavView
Panels
Progress & Activity
Streamer
Stepper
Splitter
Tabs
Tabs material
Table
Tiles
TreeView
Wizard

Information
Dialogs
Info box
Hints
Notify system
Popovers
Toasts
Windows

Date & time
Calendar
Calendar picker
Date picker
Time picker
Countdown
Formatting date

Media
Video player
Audio player

Tools
Collapse
Color module
Draggable
Dropdown
Form validator
Hotkeys
Micro templates
Ripple
Storage
Session storage
Sorter
Touch and swipe

Utilities
Animations
Additional

Dialog

Metro 4 provides the ability to easily create dialogs. Dialogs can be predefined or created in runtime.

Table of contents

About dialog

Dialogs inform users about a specific task and may contain critical information, require decisions, or involve multiple tasks. Metro 4 build `dialog` with using next classes: `.dialog`, `.dialog-title`, `.dialog-content` and `.dialog-actions` and result `dialog` has next structure:

Use Windows location service?

Bassus abactors ducunt ad triticum. A fraternal form of manifestation is the bliss.

Disagree Agree

```
<div class="dialog">
    <div class="dialog-title">Use Windows location service?</div>
    <div class="dialog-content">
        Bassus abactors ducunt ad triticum.
        A fraternal form of manifestation is the bliss.
    </div>
    <div class="dialog-actions">
        <button class="button">Disagree</button>
        <button class="button primary">Agree</button>
    </div>
</div>
```

Create dialog

Metro 4 provides two ways for create dialogs:

- Predefined method
- Runtime method

Predefined method

First-off you must create `HTML` element with special structure and define all actions required for dialog, include dismiss action. Then add any options over `data-*` attributes. And at the end add an attribute `data-role="dialog"` to your element for initialization `dialog`. After the initialization, the dialog will be hidden until it is called.

```
<div class="dialog" data-role="dialog">
    <div class="dialog-title">Use Windows location service?</div>
    <div class="dialog-content">
        Bassus abactors ducunt ad triticum.
        A fraternal form of manifestation is the bliss.
    </div>
    <div class="dialog-actions">
        <button class="button js-dialog-close">Disagree</button>
        <button class="button primary js-dialog-close">Agree</button>
    </div>
</div>
<button class="button primary"
    onclick="Metro.dialog.open('#demoDialog1')">Open dialog</button>
```

Open dialog

Create dialog at runtime

To create `dialog` at `runtime` you mus execute method `Metro.dialog.create` with required options.

```
Metro.dialog.create({
    title: "Use Windows location service?",
    content: "<div>Bassus abactors ducunt ad triticum...</div>",
    closeButton: true
});
```

Open dialog

Options

You can set any options to define dialog:

Options	Data-*	Default	Description
title	`data-title`		Set dialog title
closeButton	`data-close-button`	false	If true, close button will be added to dialog to top left corner
content	`data-content`		Set dialog content. Can be simple html, m4q object or function. Simple html text requires wrapping in a tag
actions	`data-actions`	{}	Set dialog actions.
actionsAlign	`data-actions-align`	right	Set dialog actions align.
defaultAction	`data-default-action`	true	If you are not defined actions to dialog will be added default action
overlay	`data-overlay`	true	Add overlay when dialog is opened
overlayColor	`data-overlay-color`	#000000	Overlay color. Can be hex color value or transparent
overlayAlpha	`data-overlay-alpha`	0.5	Overlay color alpha channel value.
overlayClickClose	`data-overlay-click-close`	false	Close dialog when user click on overlay
width	`data-width`	480	Dialog width
closeAction	`data-close-action`	true	If this options is true Metro 4 add event to elements with class `.js-dialog-close` for close dialog
clsDialog	`data-cls-dialog`		Add additional classes to dialog
clsTitle	`data-cls-title`		Add additional classes to dialog title
clsContent	`data-cls-content`		Add additional classes to dialog content
clsAction	`data-cls-actions`		Add additional classes to dialog actions block
clsDefaultAction	`data-cls-default-actions`		Add additional classes to dialog default action button
autoHide	`data-auto-hide`	0	If this options > 0, dialog closed after this timeout in milliseconds
removeOnClose	`data-remove-on-close`	false	If this options is `true` dialog will be removed after closes
show	`data-show`	false	If this options is `true` dialog will be showed immediate after initializing
onDialogCreate	`data-on-dialog-create`	Metro.noop	This `callback` executed after dialog is created
onOpen	`data-on-open`	Metro.noop	This `callback` executed after dialog is open
onClose	`data-on-close`	Metro.noop	This `callback` executed after dialog is close
onShow	`data-on-show`	Metro.noop	This `callback` executed after dialog is showing
onHide	`data-on-hide`	Metro.noop	This `callback` executed before dialog is hiding

Dialog actions

If you add custom actions to dialog, you can use next options for each action:

- **cls** - additional class for action element
- **caption** - caption for action element
- **onclick** - event function for action element

Open dialog

```
<button class="button info"
    onclick="openDemoDialogActions()">Open dialog</button>
<script>
    function openDemoDialogActions(){
        Metro.dialog.create({
            title: "Use Windows location service?",
            content: "<div>Bassus abactors ducunt ad triticum...</div>",
            actions: [
                {
                    caption: "Agree",
                    cls: "js-dialog-close alert",
                    onclick: function(){
                        alert("You clicked Agree action");
                    }
                },
                {
                    caption: "Disagree",
                    cls: "js-dialog-close",
                    onclick: function(){
                        alert("You clicked Disagree action");
                    }
                }
            ]
        });
    }
</script>
```

Utilities

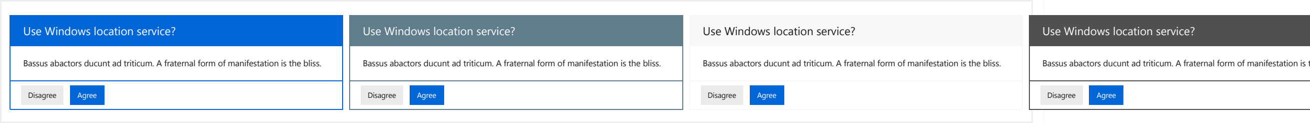
Metro 4 contains special object `Metro.dialog` to manipulate `dialogs`. This object contains next methods:

- **create(options)** - create dialog with options
- **open(el, content, title)** - open dialog
- **close(el)** - close dialog
- **toggle(el)** - toggle dialog state

- **remove(el)** - remove dialog from DOM
- **isDialog(el)** - check if element is dialog instance
- **isOpen(el)** - check if dialog is open

Accent dialogs

You can use accent classes tyo create accent dialogs.



Use Windows location service?

Bassus abactors ducunt ad triticum. A fraternal form of manifestation is the bliss.

Use Windows location service?

Bassus abactors ducunt ad triticum. A fraternal form of manifestation is the bliss.

Use Windows location service?

Bassus abactors ducunt ad triticum. A fraternal form of manifestation is the bliss.

Use Windows location service?

Bassus abactors ducunt ad triticum. A fraternal form of manifestation is t