

# Modul:Message box/Doku

Ausgabe: 17.04.2026

Letzte Änderung: 23.02.2022

Seite von

## Inhaltsverzeichnis

## Modul:Message box/Doku

Dies ist die Dokumentationsseite für [Modul:Message box](#)

[Vorlage:Lua](#)

This is a meta-module that implements the message box templates [Vorlage:TI](#), [Vorlage:TI](#), [Vorlage:TI](#), [Vorlage:TI](#), [Vorlage:TI](#), [Vorlage:TI](#), and [Vorlage:TI](#). It is intended to be used from Lua modules, and should not be used directly from wiki pages. If you want to use this module's functionality from a wiki page, please use the individual message box templates instead.

### Usage

To use this module from another Lua module, first you need to load it.

```
local messageBox = require('Module:Message box')
```

To create a message box, use the `main` function. It takes two parameters: the first is the box type (as a string), and the second is a table containing the message box parameters.

```
local box = messageBox.main( boxType, {  
    param1 = param1,  
    param2 = param2,  
    -- More parameters...  
})
```

There are seven available box types:

Box type	Template	Purpose
----------	----------	---------

<code>mbox</code>	<a href="#">Vorlage:TI</a>	For message boxes to be used in multiple namespaces
<code>ambox</code>	<a href="#">Vorlage:TI</a>	For article message boxes
<code>cmbox</code>	<a href="#">Vorlage:TI</a>	For category message boxes
<code>fmbox</code>	<a href="#">Vorlage:TI</a>	For interface message boxes
<code>imbox</code>	<a href="#">Vorlage:TI</a>	For file namespace message boxes
<code>tmbox</code>	<a href="#">Vorlage:TI</a>	For talk page message boxes

ombox [Vorlage:TI](#) For message boxes in other namespaces

See the template page of each box type for the available parameters.

## Usage from #invoke

As well as the `main` function, this module has separate functions for each box type. They are accessed using the code `{{#invoke:Message box|mbox|...}}`, `{{#invoke:Message box|ambox|...}}`, etc. These will work when called from other modules, but they access code used to process arguments passed from `#invoke`, and so calling them will be less efficient than calling `main`.

## Technical details

The module uses the same basic code for each of the templates listed above; the differences between each of them are configured using the data at [Module:Message box/configuration](#). Here are the various configuration options and what they mean:

- `types` – a table containing data used by the `type` parameter of the message box. The table keys are the values that can be passed to the `type` parameter, and the table values are tables containing the class and the image used by that type.
- `default` – the type to use if no value was passed to the `type` parameter, or if an invalid value was specified.
- `showInvalidTypeError` – whether to show an error if the value passed to the `type` parameter was invalid.
- `allowBlankParams` – usually blank values are stripped from parameters passed to the module. However, whitespace is preserved for the parameters included in the `allowBlankParams` table.
- `allowSmall` – whether a small version of the message box can be produced with `"small=yes"`.
- `smallParam` – a custom name for the small parameter. For example, if set to `"left"` you can produce a small message box using `"small=left"`.
- `smallClass` – the class to use for small message boxes.
- `substCheck` – whether to perform a subst check or not.
- `classes` – an array of classes to use with the message box.
- `imageEmptyCell` – whether to use an empty [Vorlage:Tag](#) cell if there is no image set. This is used to preserve spacing for message boxes with a width of less than 100% of the screen.
- `imageEmptyCellStyle` – whether empty image cells should be styled.
- `imageCheckBlank` – whether `"image=blank"` results in no image being displayed.
- `imageSmallSize` – usually, images used in small message boxes are set to 30x30px. This sets a custom size.
- `imageCellDiv` – whether to enclose the image in a div enforcing a maximum image size.
- `useCollapsibleTextFields` – whether to use text fields that can be collapsed, i.e. `"issue"`, `"fix"`, `"talk"`, etc. Currently only used in `ambox`.
- `imageRightNone` – whether `imageright=none` results in no image being displayed on the right-hand side of the message box.
- `sectionDefault` – the default name for the `"section"` parameter. Depends on `useCollapsibleTextFields`.
- `allowMainspaceCategories` – allow categorisation in the main namespace.
- `templateCategory` – the name of a category to be placed on the template page.
- `templateCategoryRequireName` – whether the `name` parameter is required to display the template category.
- `templateErrorCategory` – the name of the error category to be used on the template page.
- `templateErrorParamsToCheck` – an array of parameter names to check. If any are absent, the `templateErrorCategory` is applied to the template page.

