

# remove-restriction macro

[Where to use it?](#) | [Parameters](#) | [Example](#) | [All examples](#) | [See also](#)

## Overview

The `{remove-restriction}` macro is used to remove View or Edit content-level permissions.

## Where to use it?

Optional.

One or more `{remove-restriction}` macros can be put in a `{trigger}` macro.

If you want to add restrictions you can use the `{add-restriction}` macro.

You can remove/add or add/remove restrictions using these macros in any order.

If you want to add restrictions immediately after removing restrictions of the same type, you can use the `{set-restrictions}` macro instead†.



Restrictions don't give users permissions to content, but rather limit the users who have those permissions.

For example, if everyone has edit/view permissions for the space and you add edit restrictions to a page for user A and user B, only user A and user B can edit the page, and everyone else just view the page.

If you remove restrictions then you simply remove the limits you set for the permission, not the Confluence page permissions.

## Parameters

Parameter	Required	Default	Notes
<code>type</code>			What type of permission to remove? <ul style="list-style-type: none"><li><code>view</code> – assign the user/group view permission</li><li><code>edit</code> – assign the user/group edit permission</li></ul> <p>Note: The permissions are set using Confluence's content-level permissions system and can thus be altered via the padlock icon.</p>
<code>user</code>			One or more users to remove <code>type</code> permission from: <ul style="list-style-type: none"><li>Comma-separated list of usernames</li><li>Supports <a href="#">Value References</a></li></ul>
<code>group</code>			One or more user groups to remove <code>type</code> permission from: <ul style="list-style-type: none"><li>Comma-separated list of user groups</li><li>Supports <a href="#">Value References</a></li></ul>



If neither `user` or `group` parameters are defined, then all content-level permissions of `type` will be removed.

† In some cases, due to Confluence permission threads, rather than using `{remove-restriction}` macro then `{add-restriction}`, it may be useful to use `{set-restrictions}` macro.

The `{set-restrictions}` macro will clear any existing restriction of the type being set (`view` or `edit`) before setting that permission for the listed users or groups.

## Example

```
{workflow:Restrictions}
  {state:Review|approved=Published}
    {approval:Review}
  {state}
  {state:Published}
  {state}
  {trigger:pageapproved|approval=Review}
    {remove-restriction:type=view}
    {remove-restriction:type=edit}
  {trigger}
{workflow}
```

## All examples

- [Add, remove and set page restrictions](#)
- [Require Parameters on State Transitions](#) — Require workflow parameter values to be set before moving into a workflow state.

## See also

External Links:

- [Page Restrictions](#)

[Workflow Authoring Guide](#):

- [Conditions](#)
- [Roles and Permissions](#)
- [Value References](#)
- [{add-restriction} macro](#)
- [{set-restrictions} macro](#)