

# Labels

[Automatic workflow application](#) | [Enable or disable workflow features](#) | [Adding and removing labels](#) | [Trigger actions when labels change](#) | [Value references](#) | [Sticky labels](#) | [Content labelling compliance](#) | [Macros](#) | [Events](#) | [Value references](#) | [Examples](#) | [See also](#)

## Overview

As a Confluence user, you will be familiar with the concept of [labels](#) as a method of [folksonomy](#), but they can also be integrated with your workflows in order to:

- Automatically apply workflows
- Enable or disable workflow functions
- Add or remove labels in response to [Events](#)
- Perform [Actions](#) when labels are added or removed
- Make labels "sticky", which prevents their removal
- Ensure compliance with content labelling policy

## Automatic workflow application

To use this feature, the space must be running in [Space Mode](#).

Use the `label` and, optionally, `invertlabel` parameters of the `{workflow}` macro to specify which labels the workflow should or should not be applied to:

```
{workflow:name=Apply by label|label=test,testing}
  {state:Test works}
  {state}
{workflow}
```

In this example, the workflow, when applied in [Space Mode](#), will automatically be assigned to any content with the `label` "test" and/or "testing".

If you added the `invertlabel=true` parameter, it would be automatically applied to any content that does not have either of the labels "test" or "testing".

## Enable or disable workflow features

Any macro which supports [Conditions](#) can be disabled or deactivated by the `haslabel` condition.

For example, you can disable state selection or reviews if labels are missing:

```
{workflow:name=Disable review}
  {state:First}
  {state-selection:
states=Second|haslabel=first}
  {state}
  {state:Second|approved=Third}
  {approval:Review|haslabel=!second}
  {state}
  {state:Third}
  {state}
{workflow}
```

In this example:

- Transition from `First` to `Second` state requires the content be labelled "first"
- The review in the `Second` state will be disabled if the content is labelled "second"

Macros which support conditions:

- [approval macro](#)
- [state-selection macro](#)
- [trigger macro](#)

## Adding and removing labels

You can use [Events](#) to [Trigger](#) the [Actions](#) which can add or remove labels:

In this example:

- Entering the `First` state will add "first" and remove "second" labels
- Entering the `Second` state will add "second" and remove "first" labels

```

{workflow:name=Add or remove label}
  {state:First}
  {state}
  {state:Second}
  {state}
  {trigger:statechanged|state=First}
    {set-label:First}
    {remove-label:Second}
  {trigger}
  {trigger:statechanged|state=Second}
    {set-label:Second}
    {remove-label:First}
  {trigger}
{workflow}

```

Note: The label changes will take effect as soon as you transition between the states, but you'll need to refresh the page to update the on-screen list of labels for that page.

Action macros:

- [remove-label macro](#) — Remove a label from the content
- [set-label macro](#) — Add a label to the content
- [workflow-reporter macro](#)

## Trigger actions when labels change

When labels are added or removed from a page, the `labeladded` or `labelremoved` [Events](#) are sent. [Triggers](#) listening for those events can determine which label was added or removed via the `@label@`, `@addedlabel@` or `@removedlabel@` [Event references](#).

## Value references

You can access extensive information about content labels by accessing the [Label Supplier](#) via a [Supplier Reference](#) (requires [ServiceRocket Reporting for Confluence app](#)).

## Sticky labels

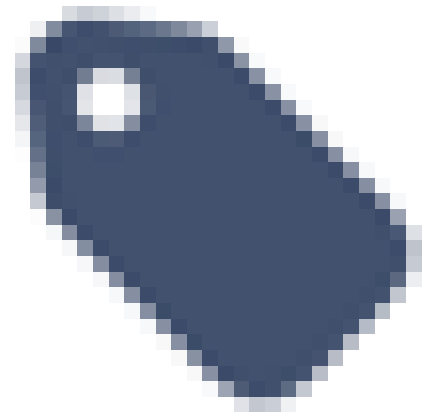
You can prevent labels from being removed (but [Admins](#) can still remove them) by using the `stickylabels` parameter on the `{workflow}` macro.

## Content labelling compliance

In situations where business or regulatory compliance mandates a content labelling policy, you can enforce that via the workflow.

For example, if all documents (pages) relating to a specific regulation must be labelled, you can either make the workflow add those labels, or require that content be labelled in order for the workflow to be automatically applied.

Our existing customers use the `{workflow-instructions}` macro to convey content labelling policy to content producers at the point of initial



content creation. As labels can be added whilst creating a page or blog post (by clicking the on at the top of the editor screen), this results in far fewer unlabelled documents being created.

ic

## Macros

- [approval macro](#) — Add content-focussed reviews to states
- [remove-label macro](#) — Remove a label from the content
- [set-label macro](#) — Add a label to the content
- [state-selection macro](#) — Specify direct state transitions

- [trigger macro](#) — Trigger actions when events are fired
- [workflow macro](#) — Defines a workflow template
- [workflow-reporter macro](#)

## Events

Label changes will generate [Events](#), which can be used to [Trigger Actions](#):

- `labeladded`
- `labelremoved`

## Value references

Information about labels can be obtained from [Value References](#):

- [Event references](#)
- [Supplier References](#)

## Examples

- [Space mode workflow application](#)

## See also

[Workflow Authoring Guide](#):

- [Templates](#)
- [Transitions](#)
- [Reviews](#)
- [Events](#)
- [Triggers](#)
- [Actions](#)

[Administration Guides](#):

- [Space Mode](#)
- [Document Management - Space Tools](#)
- [Workflows - Global](#)

[Atlassian Marketplace](#):

- [Label Manager for Confluence](#)
- [Label Tools for ScriptRunner Confluence](#)