

State expiry dates

[Specific date](#) | [Durations \(ISO 8601\)](#) | [Stale content reviews](#) | [Metadata use for the due date](#) | [Using a workflow parameter value for the due date](#) | [User editable dates](#) | [changedue date vs. due date](#) | [Remember assignees](#)

Overview

The `{state}` macro can have a `duedate` specified to set its expiry date.

This is useful in a number of circumstances – for example, published content might need a periodical review to make sure it doesn't become outdated.

Upon expiry, you can use the `expired` parameter to transition to another state, or use the `stateexpired` and `stateexpiryupdated` events to trigger some actions.

In this example, we'll look at how to set the expiry date, how to transition on expiry, and how to make the expiry date editable.

Specific date

In this example we'll set a specific date, which must be in the format: `YYYY-MM-DD HH:mm`

```
{workflow:name=Specific date}
  {state:Already expired|duedate=2020-01-20 12:00|expired=Old}
  {state}
  {state:Old}
  {state}
{workflow}
```

Note: The time zone is determined by the [Java Runtime Environment](#) in which your Confluence server or datacenter is running.

(Will that workflow ever transition to the **Old** state? **Yes**, as the workflow's background processor will find all pages that have expired but not yet triggered an expiry event.)

Durations (ISO 8601)

Another way to set the `duedate` is to specify a *duration*, rather than a *date*.

When a duration is set, the actual expiry date is automatically calculated *at the moment when the workflow enters the state*:

Actual expiry date = Date when state was entered + Duration

The duration is specified by a code starting with "**P**" (for Period) followed by codes for the number of **Y**ears, **M**onths, **W**eeks and **D**ays, or any combination of those.

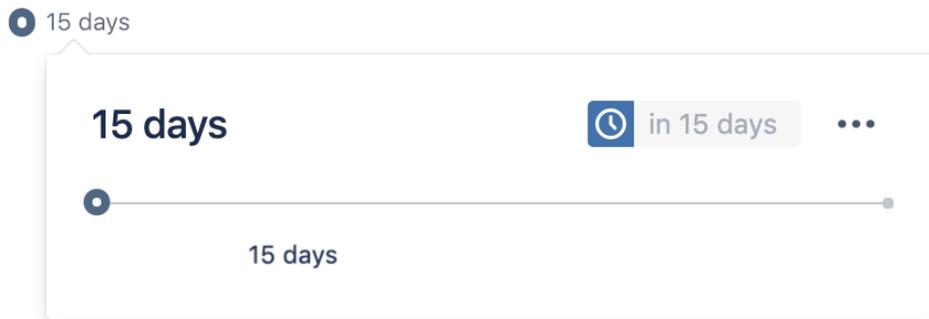
Units	5 * Unit
Years	5Y
Months	5M
Weeks	5W
Days	5D

For example, a **P**eriod of **2W**eeks and **1D**ay (a total of 15 days) would look like: **P2W1D**.

A **T**ime period can also be added using **H**ours, **M**inutes and **S**econds, for example **P1DT1H30M** (a total of 25 hours and 30 minutes).

```
{workflow:name=Duration}
  {state:15 days|duedate=P2W1D|expired=Old}
  {state}
  {state:Old}
  {state}
{workflow}
```

The result looks like this:



If the **15 days** state remains active for 15 days (**P2W1D**), it will expire causing an **expired** transition to the **Old** state.

Note:

- **P** is the period designator (for the period) that precedes the start of the duration representation.
- **T** is the time designator that precedes any time component of the duration representation

For example, **P1M** is a period of one month and **PT1M** is a period of one minute.

Stale content reviews

As we mentioned earlier, state expiry is useful for performing periodical checks on published or approved content to ensure that it's not become outdated.

```
{workflow:Stale content}
  {state:Editing|submit=Review}
  {state}
  {state:Review|approved=Published|rejected=Editing}
    {approval:Review|assignable=true}
  {state}
  {state:Published|duedate=P1Y|expired=Review|final=true|updated=Editing}
  {state}
  {trigger:stateexpired|state=Published}
    {set-message}
      This content is more than one year old and has been submitted for review.
    {set-message}
  {trigger}
{workflow}
```

This can also be done in [workflow builder](#) - the final state **Published** would be edited to add the **duedate** as a period of 1 year.

Edit a state

Edit a state from the Workflow *State Content*



Name * Colour

Published 

Description

State description

 DUE DATE

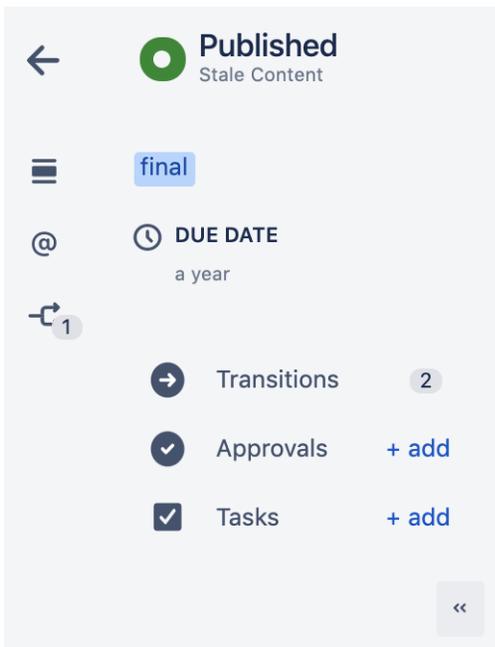
1 Day(s) ▼ Period ▼

Valid formats: durations (ISO_8601), dates (YYYY-MM-DD HH:mm) or value references

Users will be able to change and remove the date

[Apply](#) [Delete](#) [Cancel](#)

In workflow builder the **Published** state would be:



The screenshot shows the workflow builder interface for the 'Published' state. At the top, there is a back arrow, a green circle icon, and the text 'Published State Content'. Below this, there is a 'final' label. The 'DUE DATE' section is set to '1 a year'. There are three sections: 'Transitions' with a count of 2, 'Approvals' with a '+ add' button, and 'Tasks' with a '+ add' button. A double arrow icon is visible at the bottom right.

Content in the **Published** state will expire after one year (P1Y).

Published

Published

in a year

Activity report

Final: Published

This is a one year period from the date of the transition into the **Published** state.

The expiry of the content causes an **expired** transition to the **Review** state.

State Expiry Update

Created by admin, last modified less than a minute ago Review (view "**Published**")

i This content is more than one year old and has been submitted for review.

View-only users will still see the published version of the content (see [Publishing](#) for details).

State Expiry Update

Created by admin, last modified 10 minutes ago

i This content is more than one year old and has been submitted for review.

The message about the fact that it's more than one year old will also be displayed on-screen when edit permission users view the published content.

State Expiry Update

Created by admin, last modified on Dec 05, 2019 Published (view "**Review**")

i This content is more than one year old and has been submitted for review.

The `{set-message}` is triggered by the `stateexpired` event when it occurs in the **Published** state.

To remove the on-screen message you will need to add another trigger to the workflow. This can be on a **pageapproved** event when the stale content has been reviewed.

```
{trigger:pageapproved|approval=Review}
  {set-message}{set-message}
{trigger}
```

The triggers can be added to the workflow by copying and pasting the trigger markup using [workflow builder](#).



Edit Workflow

Edit details from Workflow *Stale Content*



▼ ADVANCED

Triggers

```
1 {trigger:stateexpired|state=Published}
2   {set-message}
3   This content is more than one year old and has been
4   submitted for review.
5   {set-message}
6 {trigger}
7 {trigger:pageapproved|approval=Review}
8   {set-message}{set-message}
9 {trigger}
```

Declare workflow triggers in wiki markup format to launch automated actions on certain events.

Find out more about how workflow triggers work [here](#).

Apply

Cancel

Metadata use for the due date

The `stateexpiryupdated` event could be used to always use the same due date in different states. So whenever the `duedate` is updated in one state, this will be reflected in the other states

```
{workflow:name=Shared Due Date}
  {state:First state|submit=Second state|duedate=PLM|changeduedate=true}
  {state}
  {state:Second state|final=true|updated=First state|duedate=@MyDueDate@}
  {state}
  {trigger:stateexpiryupdated|state=First state}
    {set-metadata:MyDueDate}@duedate@{set-metadata}
  {trigger}
{workflow}
```

In this example, a trigger is used to update the **MyDueDate** metadata value when the `stateexpiryupdated` event occurs in **First State**.

- the **MyDueDate** metadata value is set as the `duedate` value in **First State**.
- the **MyDueDate** value is then used as the `duedate` for **Second State**.

See: [Value references](#).

Using a workflow parameter value for the due date

In some cases, we might want a shorter or longer expiry date.

If the workflow is stored at global level and used in many spaces, we obviously need a way to allow space admins to define a `duedate` which is applicable to the content in their space.

We can do that using the `{workflowparameter}` macro, as shown below:

```
{workflow:Stale content - editable duedate}
  {workflowparameter:Stale Content Duration|type=duration|description=Duration for periodical review of
published content}
  PLY
  {workflowparameter}
  {state:Editing|submit=Review}
  {state}
  {state:Review|approved=Published|rejected=Editing}
  {approval:Review|assignable=true}
  {state}
  {state:Published|duedate=@Stale Content Duration@|expired=Review|final=true|updated=Editing}
  {state}
  {trigger:stateexpired|state=Published}
  {set-message}
  This content is more than one year old and has been submitted for review.
  {set-message}
  {trigger}
{workflow}
```

A workflow parameter can be easily added to the workflow using workflow builder.



Add parameter

Add an editable value reference to the Workflow *Stale Content*



Name *	Type *
<input type="text" value="Stale Content Duration"/>	<input type="text" value="duration"/>

Value
<input type="text" value="1"/> <input type="text" value="Year(s)"/> <input type="text" value="Period"/>

Enter a value or value reference.
Find out more about how parameters work [here](#).

Description
<input type="text" value="Duration for periodical review of published content"/>

ATTRIBUTES

Editable
Users will be able to change the parameter

This adds the `workflowparameter` markup to the workflow.

```
{workflowparameter:Stale Content Duration|type=duration|description=Duration for periodical review of
published content}
  P1Y
{workflowparameter}
```

The state can be edited in workflow builder to add the `duedate` and use the value of the workflow parameter for the `duedate`.



Edit a state

Edit a state from the Workflow *Stale Content with editable duedate*



Name *

Colour



Description

🕒 DUE DATE

Stale Content Duration Value Reference

Valid formats: durations (ISO_8601), dates (YYYY-MM-DD HH:mm) or value references

Users will be able to change and remove the date

Apply Delete Cancel

Editing the state in workflow builder adds the `@Stale Content Duration@` as the value for the **Published** state `duedate` in the workflow markup.

```
{state:Published|duedate=@Stale Content Duration@|expired=Review|final=true|updated=Editing}
{state}
```

We've made two changes to the previous example:

- A workflow parameter called `Stale Content Duration` has been added, with a default value to `P1Y` (1 year).
- The resulting value reference, `@Stale Content Duration@`, has been used for the `duedate` value in the **Published** state.

The [space admin](#) can now view and edit the default value in [Parameters - Space Tools](#), assuming this workflow is applied in [space mode](#):

Space Tools

Overview Permissions Content Tools Look and Feel Integrations Document Management Publishing Tasks

Document Management Configuration Parameters Notifications Import Clear History Audit Log

Workflow Parameters

You can set the values of the Workflows Parameters for this space.

Name	Value	
> Stale Content Duration	<input type="text" value="1"/>	<input type="text" value="Years"/> <input type="button" value="Save"/> <input type="button" value="Cancel"/>

This allows each space to have a different value, without needing to edit the workflow itself – they just set the parameter value instead.

Another example of this approach can be seen in the examples associated with our [Communardo Metadata for Confluence app](#) integration.

User editable dates

You can allow users to edit the `duedate` by using the `changeduedate` parameter.

By adding `changeduedate=true` to the **Published** state, users who can edit content will be able to change the due date via the [workflow popup](#) (see [State expiry date](#) for more information).

```
{state:Published|changeduedate=true|duedate=@Stale Content  
Duration@|expired=Review|final=true|updated=Editing}  
{state}
```

The `changeduedate=true` can be set by editing the state using workflow builder.

Edit a state

Edit a state from the Workflow *Stale Content*



Name * Colour

Published 

Description

State description

DUE DATE

Stale Content Duration Value Reference

Valid formats: durations (ISO_8601), dates (YYYY-MM-DD HH:mm) or value references

Users will be able to change and remove the date

Apply Delete Cancel

In the workflow popup a user can simply click the existing expiry date to edit it, showing a screen like this:

Published

Set Due Date

Set or remove a due date for the state

2020-08-20 09:30 ✕

< **August 2020** >

SUN	MON	TUE	WED	THU	FRI	SAT	
							5:30 AM
							6:00 AM
							6:30 AM
							7:00 AM
26	27	28	29	30	31	1	7:30 AM
2	3	4	5	6	7	8	8:00 AM
9	10	11	12	13	14	15	8:30 AM
16	17	18	19	20	21	22	9:00 AM
							9:30 AM
23	24	25	26	27	28	29	10:00 AM
30	31	1	2	3	4	5	10:30 AM
							11:00 AM

Save Cancel

If they change the due date, it will override the value defined by the `@State Content Duration@` workflow parameter but only on that individual page or blog post.

This is extremely useful in cases where authors might be aware of planned changes to a product, service or process because they can ensure a review occurs around the time when that will happen.

changeduedate vs. duedate

The `duedate` and `changeduedate` parameters can be used on their own or together. The following table describes what will happen with each possible combination:

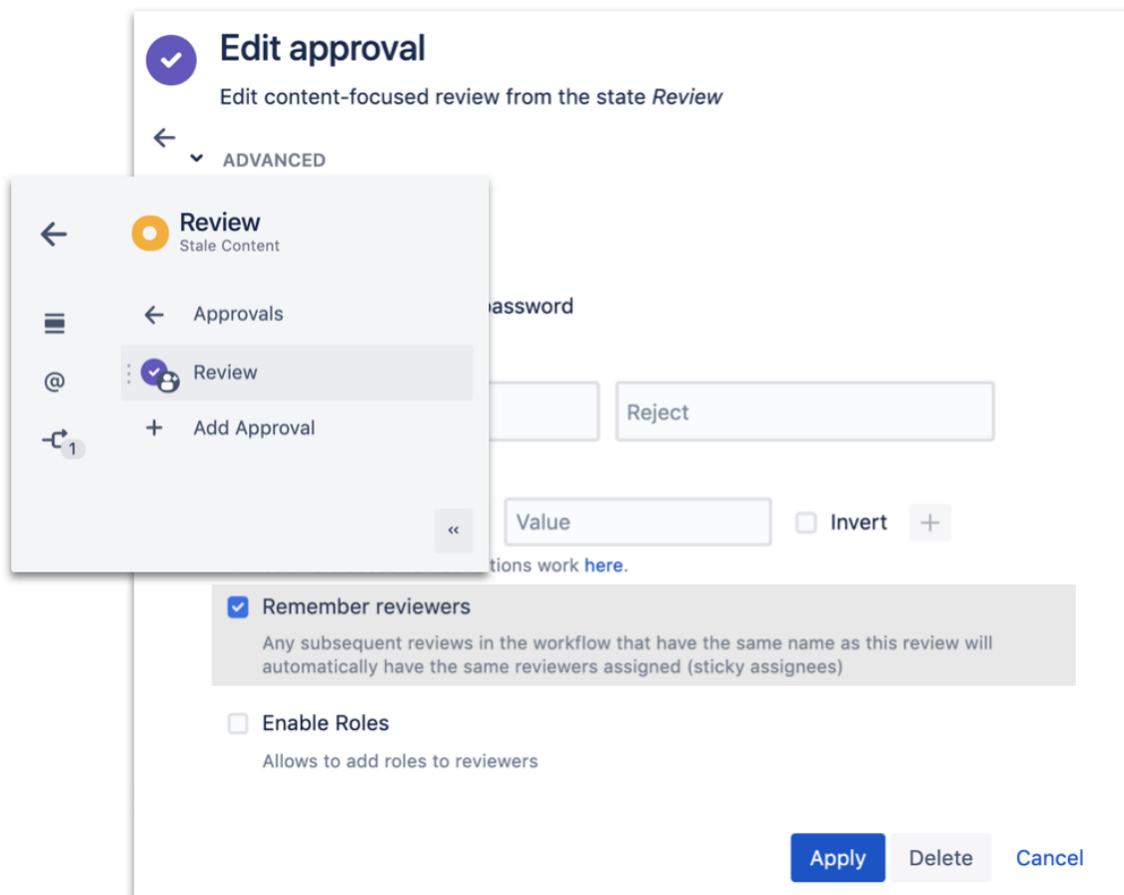
<code>duedate</code>	<code>changeduedate</code>	Description
not set	false	State will not have an expiry date nor can users add an expiry date.
set	false	State will have an expiry date based on the <code>duedate</code> parameter that the user cannot change.
not set	true	State will not initially have an expiry date, but the user can add an expiry date.
set	true	State will initially have an expiry date based on the <code>duedate</code> parameter, but the user can change the expiry date.

Remember assignees

Wouldn't it be nice if subsequent reviews called back the original reviewers? You can do that by adding the `rememberassignees` parameter to the `{approval}` macro:

```
{approval:Review|rememberassignees=true|assignable=true}
```

The `rememberassignees` parameter can be set to `true` by editing the approval in workflow builder to check the **Remember reviewers** option.



Now when the published content expires and it gets sent back for review, the original reviewers will be contacted automatically. 😊