

transitivelyLinkedIssues()

This function returns a list of issues that are **directly** or **transitively** linked to the **current issue**.

Transitively linked issues are issues which are indirectly linked to the current issue (e.g. which are linked to an issue which itself is linked to the current issue).

Syntax

transitivelyLinkedIssues(linkTypes) #Output: Issue list

Examples

Parser expression	Description
<div>transitivelyLinkedIssues ("relates to")</div>	<p>This example returns an issue list with all issues that are directly or transitively linked to the current issue using the link type "relates to".</p> <div><p>Consider the following example:</p><p>ISSUE-1 [relates to] ISSUE-2</p><p>ISSUE-2 [relates to] ISSUE-3</p><p>In this case ISSUE-3 is transitively linked to ISSUE-1.</p><p>The example would return [ISSUE-2, ISSUE-3].</p></div>
<div>transitivelyLinkedIssues(" ")</div>	<p>This example returns an issue list with all issues that are directly or transitively linked to the current issue using any link type.</p>

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
linkTypes	<div>TEXT</div>	<p>A comma separated list with link types.</p> <p>When passing an empty text ("") all link types will be used to search for directly or transitively linked issues.</p>

Output

This function returns an

ISSUE LIST

.

If there are no directly or transitively linked issues, the function returns

NULL

.

Variant where you additionally can define an **issue list**.

Syntax

```
transitivelyLinkedIssues(linkTypes, issueList) #Output: Issue list
```

Examples

Parser expression	Description
<pre>transitivelyLinkedIssues ("relates to", epic())</pre>	This example returns an issue list with all issues that are directly or transitively linked to the current issue's epic using the link type " relates to ".
<pre>transitivelyLinkedIssues(" ", subtasks())</pre>	This example returns an issue list with all issues that are directly or transitively linked to the current issue's sub-tasks using any link type.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
linkTypes	<input type="text" value="TEXT"/>	A comma separated list with link types. When passing an empty text ("") all link types will be used to search for directly or transitively linked issues.
issueList	<input type="text" value="ISSUE LIST"/>	Any given issue list. Usually this value is retrieved from a function (e.g. linkedIssues() or subtasks()).

Output

This function returns an

Variant where you define **issue keys** instead of an issue list.

Syntax

```
transitivelyLinkedIssues(linkTypes, issueKeys) #Output: Issue list
```

Examples

Parser expression	Description
-------------------	-------------

```
transitivelyLinkedIssues("", %  
{parent.key})
```

This example returns an **issue list** with all issues that are directly or transitively linked to the current issue's parent using any link type.

Additional information

Parameters used in this function

Parameter	Input (data type)	Description
linkTypes	<input type="text" value="TEXT"/>	A comma separated list with link types. When passing an empty text ("") all link types will be used to search for directly or transitively linked issues.
issueKeys	<input type="text" value="TEXT"/>	A comma separated list with issue keys.

Output

This function returns an



Use cases and examples

Use case

No content found.