

## Insertion/Extender Duplicates Filter

When you build a structure using both **insert** and **Extend** generators, there is a chance you will wind up with duplicate issues – the **Inserter/Extender Duplicate** filter allows you to quickly remove those duplicate issues from your structure.

Here's how this can happen:

1. You use an Insert generator to add every Story from a current project. Those stories are added to the the top level of your structure.
2. You then use the [Linked Issues Extender](#) to add issues linked to those stories - these issues will be placed below your existing stories in the structure.
3. If any of the original stories are linked to each other, they will appear more than once in your structure – at the top level (because they were added by the inserter) and as children of other issues (because they were added again by the extender).

The Insertor/Extender Duplicate Filter will remove such issues from the top level and only keep the children. Please see examples below for a more detailed explanation.

## Examples

## Basic Links

Imagine we have a project with issues Story 1, Story 2, Story 3, and Story 4, and some of the issues are blocking other issues:

- Story 1 is blocked by Story 2
- Story 2 is blocked by Story 3 and Story 4

In our structure, we want to see all issues from our project arranged based on the existing "Blocks" links.

After you add all four issues by a JQL Inserter and add a Links Extender, you will get the following hierarchy:

Project Dependencies

Summary

Project Dependencies

% Add issues linked by **Blocks**: parent **is blocked by** children

+ Insert issues: **Project = Demo**

▼ Story 1

▼ Story 2

Story 3

Story 4

▼ Story 2

Story 3

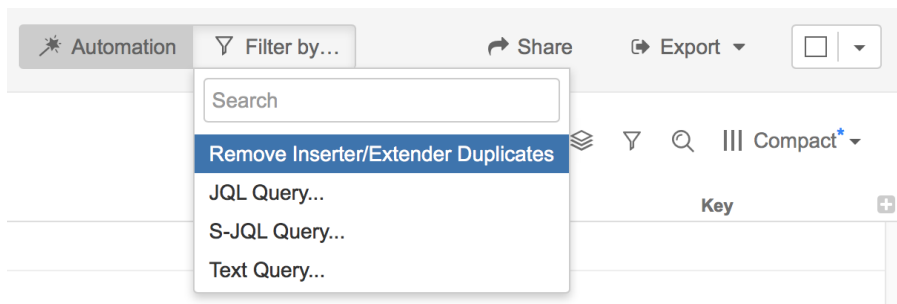
Story 4

Story 3






Story 4

You can see that some issues have been duplicated, because the Extender adds the children under parents, even if they are already in the structure.

Now let's add the Duplicates Filter:



As a result, we get a structure with the hierarchy and no duplicates:

Project Dependencies	
Summary	
	 Project Dependencies
 Remove Inserter/Extender Duplicates	
 Add issues linked by <b>Blocks</b> : parent is blocked by children	
 Insert issues: <b>Project = Demo</b>	
▼ 	Story 1
▼ 	Story 2
	Story 3
	Story 4

## Multiple Parents

We have the same situation as in the example above, but we have one story that blocks two other stories, so it should be shown under both of them:

- Story 1 is blocked by Story 2 and Story 3
- Story 2 and Story 3 are blocked by Story 4

Without Duplicate Filter, it looks like this:

## Project Dependencies ▾

### Summary



### Project Dependencies

🔗 Add issues linked by **Blocks**: parent is blocked by children

+ Insert issues: **Project = Demo**

▾ Story 1

▾ Story 2

Story 4

▾ Story 3

Story 4

▾ Story 2

Story 4

▾ Story 3

Story 4

Story 4

With the filter applied, any identical instances are removed:

## Project Dependencies ▾

### Summary



### Project Dependencies

🗑 Remove Inserter/Extender Duplicates

🔗 Add issues linked by **Blocks**: parent is blocked by children

+ Insert issues: **Project = Demo**

▾ Story 1

▾ Story 2

Story 4

▾ Story 3

Story 4

In this example, Story 4 still appears twice – this is because these two instances are not identical. One is blocking Story 2, while the other is blocking Story 3. Both need to be expressed in the hierarchy.

## Link Cycles


If there are link cycles between the issues (both issues link to each other), the Duplicates Filter will remove one of the branches and keep the other, to make sure all the issues added by the Insert and Extend generators are in the structure.


In this example, Story 1 blocks Story 2 and Story 2 blocks Story 1.


Without the filter, we get the following structure:

Project Dependencies

Summary

 Project Dependencies

 Add issues linked by **Blocks**: parent is **blocked by** children

 Insert issues: **Project = Demo**

Story 1

Story 2

Story 1

Story 2

Story 1

Story 2


Story 3


Story 4


With the filter, one of the branches with the cycle gets removed:


Project Dependencies

Summary

 Project Dependencies

 Remove Inserter/Extender Duplicates

 Add issues linked by **Blocks**: parent is **blocked by** children

 Insert issues: **Project = Demo**

Story 2

Story 1

Story 2

Story 3

Story 4