

## Insertion/Extender Duplicates Filter

This filter should be used when you want to add a number of issues into your structure using an Inserter and then arrange them into a hierarchy based on the links which exist between them.

Here is how you can do it:

1. The **Insert**er adds a number of issues on the top level.
2. The **Link** Extender adds the issues which are linked to them as children.

If some of the children have already been added to the top level by the Insertor, you will get duplicates - an issue at the top level also added as a child.

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

## Basic Example

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

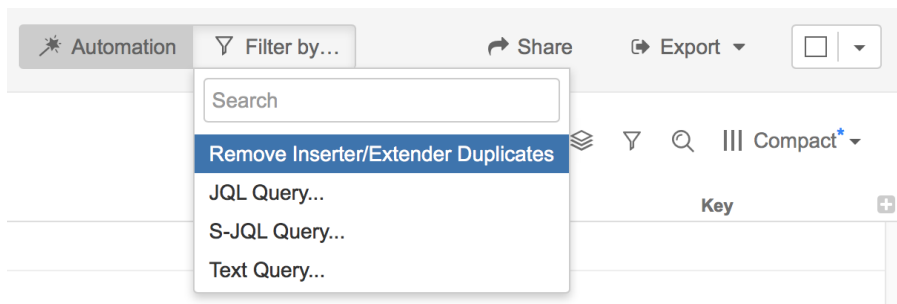
Story 4

Story 3

Story 4

You can see that the issues are now duplicating as the Extender just adds the children under parents, not moves them.

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 Example

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

So 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

And with the filter applied all the duplicates are removed. Please note that Story 4 is present in two place to reflect the hierarchy and in this case these two instances are not duplicating each other:

## 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

## Example With Link Cycles


If there are link cycles between the issues, the Filter will remove one of the branches and will keep the other to make sure all the issues added by the inserter and extender 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


And as we add 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