

Pending changes

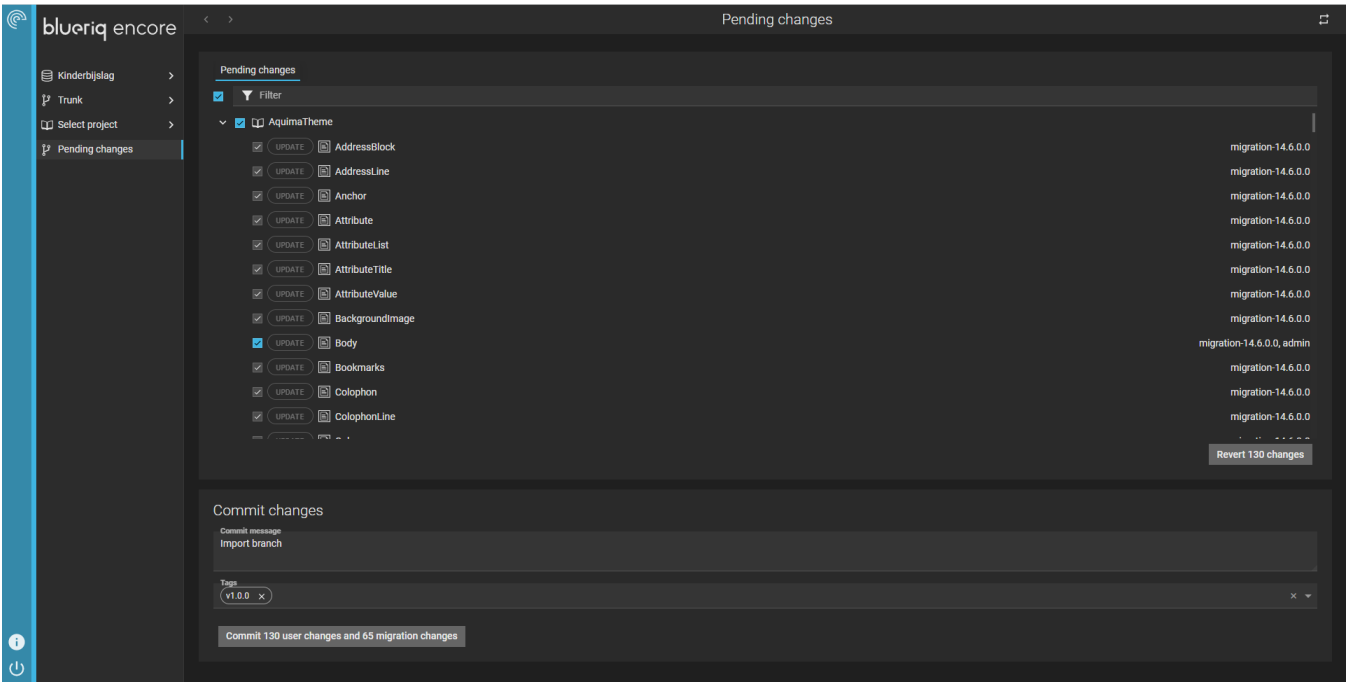
By committing pending changes you keep track of the history of your development. Committing enables you to find out when and why a certain change was made, who made the change, or which releases might be affected by a bug. Each time you commit your changes, a 'snapshot' is recorded in the version control system. These 'snapshots' are called revisions, and can be viewed from the [commit history view](#).

Common practice is to commit your changes as soon as you complete building and testing a new feature or bug fix. In order to get a good record of changes, each modeler commits his own changes.

In the pending changes view in Blueriq Encore, all changes made in the branch are displayed. So if you branch contains multiple projects, changes to all projects are listed for registration. These changes are shown in a tree structure with 3 levels: project, module, element.

The pending changes view can be opened from the branch menu in the navigation panel after selecting a branch.

An example of the pending changes view:



You can navigate to elements that are created or updated using CTRL + click. You can select all or a subset of changes in a certain branch to be committed. Migration changes are of an informative kind: they are selected by default and cannot be unselected or reverted.

Action	Description
Refresh	Use the refresh button in the title bar on the top right to refresh the list of pending changes.
Filter	Can be used to filter in the tree of changes. You can use the filter on the action (create, update, delete) element type, element name, and author.
Revert	Before committing all changes, you should check them first to see if they are still relevant. It might be possible that some of the changes are incorrect or obsolete, and you don't want to process them. You can do this by reverting the changes. This will bring the element back to the state of the previous revision. Note that reverting is different from unselecting the change; the latter action will only postpone the registration, but it will still be part of your branch. Be careful; reverting changes cannot be undone.
Commit message	<div>In the commit section there is a commit message, this message is required. This message is important because it provides insight in the reason changes were made and what changes were made to your project(s). The message will appear in the commit history view. <u>The registration message cannot be altered once the changes are committed.</u></div> <div> It's considered a good practice to keep the message short and start the commit message with a verb. I.e., "Create repository," "Add feature x," and "Fix component y." You may also use emojis in the commit message, such as 🚨 for a critical hotfix.</div>
Tags	Next to the registration message are the tags. Tags are used to put a label on a specific revision. For instance, the tag 'RC_1' could indicate that this revision is release candidate 1. You can put multiple tags on a revision. Tagging can be done during change registration, but you can change, add, or remove tags at any time from the commit history view .