

# Cohesic XTDB Docs

## Summary

1. Entity Revert .....	1
1.1. Time travelling .....	2

This document describes in more detail some of the functionality implemented in [com.cohesic/xtdb](https://github.com/cohesic/xtdb).

It assumes that you have knowledge about [XTDB's](#) main concepts (entity, document, transaction, etc.).

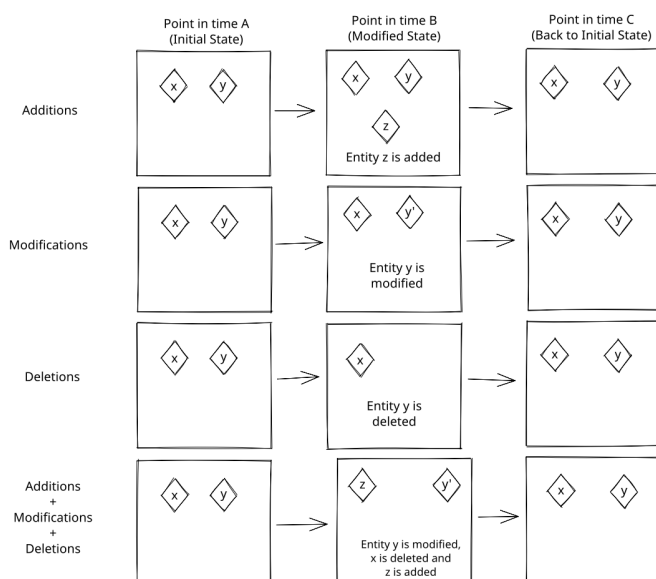
## 1. Entity Revert

Reverting an entity is the act of taking the snapshot of an entity at a point in time in the past, removing any modifications that might have been made since. There are three main operations we can perform on an entity:

1. creation
2. deletion
3. modification of entity attributes

All of them can be reverted consistently using [com.cohesic.xtdb/reverse-tx-ops-before-time](#), see below diagram.

*Operations on entities and how their state reverts back to a point in time*

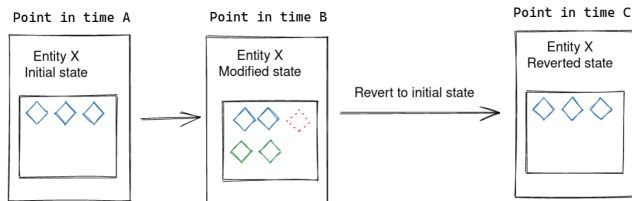


## 1.1. Time travelling

In order to revert an entity we need go back in time and identify changes that happened before a certain time. There is no way, at the point of this writing, for XTDB to do this via direct query.

Therefore, we leverage the XTDB's [History API](#), filtering out the history items until we reach the one we need. Once we find the history item and its associated document the library computes new transactions, as data, that revert the entity. It is up to the consumer of the library to submit these newly minted transactions to XTDB.

*Reverting by transacting to the entity initial state*



As we mentioned above, there are three ways to modify an entity. In the following we show how the XTDB history looks like for each of them:



history is in descending order, aka most recent at the top

- **Newly created entity:**
  - Its history includes only a single item, the latest one.

```
[{:xt/tx-time #inst "2022-09...",
 :xt/tx-id 2,
 :xt/valid-time #inst "2022-09...",
 :xt/content-hash #xtldb/id "c95f149636e0a10a78452298e2135791c0203527"
 :xt/doc {:xt/id 3 :entity/attribute "bar"}}]
```

- **Modified entity:**
  - The first item in its history includes the last changes made to the entity and previous items include documents at different points in time.

```
[{:xt/tx-time #inst "2022-09...",
 :xt/tx-id 2,
 :xt/valid-time #inst "2022-09...",
 :xt/content-hash #xtldb/id "b95f149636e0a10a78452298e2135791c0203528"
 :xt/doc {:xt/id 2 :entity/attribute "Mod2 foo"}}
{:xt/tx-time #inst "2022-08...",
 :xt/tx-id 2,
 :xt/valid-time #inst "2022-08...",
 :xt/content-hash #xtldb/id "a95f149636e0a10a78452298e2135791c0203529"
 :xt/doc {:xt/id 2 :entity/attribute "Mod1 foo"}}
{:xt/tx-time #inst "2022-07...",
 :xt/tx-id 1,
```

```
::xt/valid-time #inst "2022-07...",  
::xt/content-hash #xt/db/id "a95f149636e0a10a78452298e2135791c0203529"  
::xt/doc {:xt/id 2 :entity/attribute "foo"}}
```

- **Deleted entity:**

- Its history is similar to the above with the only difference being that the first element denotes a "delete transaction", which is still a valid, yet special, history item. This item is as a map with a **nil** document and a content hash filled with zeros.

```
[{:xt/tx-time #inst "2022-09...",  
 :xt/tx-id 2,  
 :xt/valid-time #inst "2022-09...",  
 :xt/content-hash #xt/db/id "0000000000000000000000000000000000000000"  
 :xt/doc nil}  
{:xt/tx-time #inst "2022-08...",  
 :xt/tx-id 1,  
 :xt/valid-time #inst "2022-08...",  
 :xt/content-hash #xt/db/id "a95f149636e0a10a78452298e2135791c0203529"  
 :xt/doc {:xt/id 2 :entity/attribute "foobar"}}
```

Once we know how the history is represented, it is easy to see how the library can revert an entity: it is doing it by grabbing the history item right before the input time.

Nonetheless, this comes with a twist: how does the library handle an entity that was created? How does it handle a deleted entity?

It does it, roughly, this way:

- If the history item before **as-of-time** exists it means the entity was either *modified* or *deleted* (the resulting history item is the zeroed one). It submits a **::xt/put** transaction for updating the current state.
- If it does not exist, the entity was *newly added*. Therefore, it needs to **::xt/delete** the entity in order to revert it.

In both instances we also send a **::xt/match** against the latest document in history to guards us against concurrent changes.