

## Contents

- [Overview](#)
- [Entity cache](#)
- [Setting properties](#)
- [EntityState and EntityAspect](#)

[Create](#), [modify](#), and [delete](#) are things you do to entities while they are in memory.

## Overview

You saw how to retrieve entities from the database in the [Query](#) topic.

Now that entities are in memory, you may want to modify them, [delete](#) them, and [create](#) new ones - the three most important activities covered in this topic.

As you take these actions, you are making changes **locally**. These are **pending** changes, visible only in the current client session. Even "deleted" entities are provisional; they are merely scheduled for deletion. Your changes become permanent when you save them to the database, a subject covered separately in the [Save](#) topic. Here we concentrate exclusively on what you can do to entity instances while they are in memory.

We know you want to get right down to business and start changing entities. But first we have to introduce a few fundamentals that run through the entire entity change discussion.

## Entity cache

The entities in memory usually reside in an [EntityManager's cache](#) cache. When you [query](#) for entities from the database, they go directly into cache. That's most likely how you'll get the entities that you modify and delete. You can remove entities from the cache and manipulate them while they are detached. Note that some entity features are available only when an entity is in cache ... which is where it should be unless you have a good reason to do otherwise.

## Setting properties

Most of your changes involve setting data properties of the entity. It seems simple enough to set a property with a statement like:

```
aCustomer.Name="Acme";
```

```
aCustomer.Name="Acme"
```

In fact, the setter of that innocent looking "Name" property is doing work under the hood that you should know about as we'll see in the [Set a property](#) topic.

## EntityState and EntityAspect

An entity's [EntityState](#) tells you whether an entity is in cache or not. If it is in cache, it tells you if the entity has been changed or not ("Unchanged"). If it has changed, you'll want to know if it is new ("Added"), modified ("Modified"), or scheduled for deletion ("Deleted"). Many of the things you do to an entity - actions described in this section - will change its EntityState. Keep your eye on that EntityState.

Every entity has an EntityState but you won't see it listed among the entity's properties. You will see an [EntityAspect](#) property that returns an EntityAspect object. The EntityAspect holds the EntityState ... and exposes a goodly number of other members that concern the intersection between the entity and the DevForce infrastructure. You'll need the EntityAspect to get to some of the features discussed in this section.