

Node has [neuron::container::Node::owning_handle](#) member

Prop has [neuron::container::Mechanism::owning_handle](#) member



[Constructor of owning_handle](#) calls [acquire_owning_identifier](#) for both

By calling `for_all_vectors<detail::may_cause_reallocation::Yes>()` we append to the `m_storage` of `field_data` an element. We also set the `m_ptr` (`std::unique_ptr<std::size_t, deleter>`) of the `owning_identifier` to the `old_size`. This identifier will be used as offset to access the corresponding element from other parts of the code. Due to the fact that we use the `detail::may_cause_reallocation::Yes` template, when we call `for_all_vectors()` the `m_data_ptrs` member of the `field_data` for mechanisms will be updated to use the up to date pointers to the `m_storage` vectors