nx::next(3) 2.0 next ""

NAME

nx::next - Skip to the next most specific method implementation

TABLE OF CONTENTS

SYNOPSIS

next ?*arguments*?

DESCRIPTION

`next` ? *`arguments`* ?

This command is invoked inside a method body to call the next most specific method implementation in the list of available methods. This list of available methods is specific to the current method-call context. This context is set by the usage context of `nx::next` (method combination vs. method-call interception; see below). The optional *`arguments`* are the argument values to be passed into the next most specific method implementation. If omitted, the arguments of the current method call are automatically forwarded. To call the next most specific method implementation without arguments (or to suppress argument forwarding), *`arguments`* must be set to an empty string. To pass an empty string as a (single) argument value, protect it as a list. The result of a call to `nx::next` is the result of the next most specific method implementation. If there are no more further applicable methods, the result of `nx::next` will depend on its usage context: method combination or method-call interception. If `nx::next` is used in a method body for method combination, the result will be an empty string. If `nx::next` is used in the body of a filter method for method-call interception, the result will be an error.

When executing a method call, the NX dispatch mechanism computes a list of applicable method implementations for the method name requested from a given object receiving the call; in support of method combination and method-call interception.

For *method combination*, the computed list contains any object-local method implementation and any method implementations inherited by the object from the classes in its precedence list. Examples are overloading method implementations in the class hierarchy of the object, as well as from mixin classes of the object. For *method-call interception*, the computed list contains the applicable filter methods, again ordered by their definition order according to the precedence list of the called object.

To retrieve the next most specific method implementation to be invoked by `nx::current` from the internally computed list, if any, use `nx::current`.

COPYRIGHT