



nx::current(3) 2.0 current ""

NAME

nx::current - Return information about the method callstack

TABLE OF CONTENTS

Table Of Contents
Synopsis
Description
Copyright

SYNOPSIS

current ?*option*?

DESCRIPTION

current ? *option* ?

This introspection command provides information about various details, to be identified using *option*, on the callstack. The command is invoked from a method body. If *option* is not provided, **nx::current** will default to option *object* (see below). **nx::current** operates on the Tcl callstack and is aware of NX-specific callstack and stackframe details. *option* can be any of the following:

- **activelevel** returns the actual callstack level which calls into the currently executing method directly. This **activelevel** might correspond the **callinglevel**, but this is not necessarily the case. The **activelevel** also includes intermediate calls, such as **nx::next** invocations. The level is reported as an absolute level number (# followed by a digit) to be directly used as the first argument to **uplevel** or **upvar**.
- **args** returns the list of argument values passed into the currently executing method implementation.
- **calledclass** returns the name of the class that provides the method implementation to which the intercepted method call is to be redirected (only available from within filter methods).
- **calledmethod** returns the original method name requested by intercepted method call (only available from within filter methods).
- **callingclass** returns the name of the class which provides the method implementation calling into the currently executing method. See also **callingobject**.
- **callinglevel** resolves the callstack level of the originating invocation of the currently executing method implementation. Callstack levels introduced by method interception (e.g., filters) and by method combination (**nx::next**) are ignored. The level is reported as an absolute level number (# followed by a digit) to be directly used as the first argument to **uplevel** or **upvar**. See also **activelevel**.

- `callingobject` returns the name of the object which is calling into the currently executing method. See also `callingclass`.
- `class` returns the name of the class providing the currently executing method implementation. The returned method-providing class may be different to the class of the current object. If called from within a method implementation provided by the current object itself, an empty string is returned.
- `filterreg` returns the object (class) on which the currently executing method was registered as a filter method (only available from within filter methods).
- `isnextcall` will return 1, if the currently executing method implementation was invoked via `nx::next`; 0 otherwise.
- `method` returns the name of the currently executing method. If an ensemble-method call, the name of the bottom-most ("leaf") method is returned.
- `methodpath` returns the combined name of the currently executing method (including all ensemble levels) in an ensemble-method call. Otherwise, for a regular method call, the result corresponds to the result of option `method`.
- `nextmethod` returns the name of the next most specific method implementation to be called when invoking `nx::next`.
- `object` gives the name of the object on which the currently executing method implementation is evaluated.

COPYRIGHT

Copyright © 2014 Stefan Sobernig <stefan.sobernig@wu.ac.at>, Gustaf Neumann <gustaf.neumann@wu.ac.at>; available under the Creative Commons Attribution 3.0 Austria license (CC BY 3.0 AT).