

MiscIfStack

Inherits From: MiscStack
Declared In: misckit/MiscIfStack.h

Class Description

A MiscIfStack is a specialized stack which may be used in a dynamic interpreter to implement if/then/else/endif constructs. The syntax is expected to have an `^endif^` match every single `^if^` token. The `^else^` token is optional. The `^if^`, `^else^`, and `^endif^` divide the interpreted code into blocks. The MiscIfStack can tell you whether or not the current block should be executed. This includes correct handling of nested ifs.

To use a `MiscIfStack`, first send a **-reset** message. Then, whenever an `^ifo` token is encountered in the parsing, send a **-startIf:** message to the `MiscIfStack`. If the conditional evaluated true, use YES as the argument. Use NO if it evaluated falsely. When (and if) an `^elseo` token is encountered, send a **-startElse** message. Finally, when the `^endifo` token is encountered, send a **-endif** message.

To determine if the current block should be executed, simply query the MiscIfStack with a **-currentConditionalsActive** message. If YES is returned, then the code should be executed.

Instance Variables

Method Types

Set up - reset

Handling statements	<ul style="list-style-type: none"> - startIf: - startElse - endIf - currentConditionalsActive
Error notification	<ul style="list-style-type: none"> - endWithoutIfError - elseWithoutIfError - doubleElseIfError

Instance Methods

currentConditionalsActive

- (BOOL)**currentConditionalsActive**

Returns YES if the `if` or `else` block on top of the stack evaluates true and should be executed. Returns NO otherwise.

doubleElseIfError

- (void)**doubleElseIfError**

Prints a diagnostic error message to the console if two `else` tokens are found in a row. This is syntactically the same as having an `else` without a matching `if`.

elseWithoutIfError

- (void)**elseWithoutIfError**

Prints a diagnostic error message to the console if an `else` token is found that does not have a matching `if` token.

endIf

- **endIf**

Ends an `if-else` block, returning the `MiscIfStack` to the status of the block before the `if` to be cleared was encountered. Returns **self** if successful and **nil** if there was an error, such as an `endif` without a matching `if`.

endWithoutIfError

- (void)**endWithoutIfError**

Prints a diagnostic error message to the console if an `endif` token is found that does not have a matching `if` token.

reset

- **reset**

Clears the MiscIfStack. This should be called whenever a new program is started. Returns **self**.

startElse

- **startElse**

Begins an `else` block, changing the status to be the opposite of the `if` block this is paired to. Returns **self** if successful and **nil** if an error occurs. Errors include two `else` tokens in a row or an `else` without an accompanying `if` token.

startIf:

- **startIf:**(BOOL)*isActive*

Starts an `if` block. If the `if` evaluates to true, then *isActive* should be YES, NO otherwise. This will be used to determine the current status of the MiscIfStack. Returns **self**, or **nil** if an error occurs.