MiscClassDecoder

Inherits From: Object

Declared In: <misckit/MiscClassDecoder.h>

Class Description

The MiscClassDecoder object acquires the definition of a given class from the Objective-C run-time system. The class definition is decoded, translated into a form similar to an @interface definition and displayed in a window.

Bugs and Limitations

The syntax of array variables is not correct. The array brackets are printed before the variable name.

I don't know how to find the return type of a method. Therefore no return type is shown in the output.

Most struct names are not known to the decoder and their contents are shown even in method definitions.

Category methods are defined in objc_method_list-s and shown in separate sections. But I don't know how to find the name of a category.

Class names are not sorted.

There should be a method to analyze the class whose name is selected in **text**.

Usage

This class is most useful when you want to get information about undocumented classes. It can only get information about classes in its own application, but there is a bundle in the MiscClassDecoder example application that can be loaded into any application running on your system!

In order to analyze a class in another person's application, you have to run that application from the gdb debugger and load the MiscClassDecoder bundle from the example application.

```
> gdb SomeApp.app/SomeApp
(gdb) run
Ctrl-C
(gdb) print [[[[NXBundle alloc] initForDirectory:
           "/LocalDeveloper/Examples/MiscKit/MiscClassDecoder.app/MiscClassDeco
           der.bundle"] classNamed: "MiscClassDecoderSetup"] new]
(gdb) continue
```

You will see a new ClassDecoder window pop up in the application. :-)

Instance Variables

id **classNameField**;
id **text**;
int **indent**;
BOOL **bol**;
int **lineCount**;

classNameField	The TextField where the name of a class is entered.
text	The Text in a ScrollView where the class definition is displayed.
indent	Current indentation depth. Temporary variable.
bol	Indicates whether we are at the beginning of a line. If this is YES then a number of blanks is written before the next text. Temporary variable.
lineCount	The number of lines output so far. Temporary variable.

Method Types

Analyzing a class definition	- analyze:
Listing all known classes	- list:
Interface for Services	- analyzeClass:

Instance Methods

analyze:

- **analyze:***sender*

Reads a class name from **classNameField** and displays its interface definition in **text**. The definition is given in a format suitable for copying into a .h header file. Returns **self**.

See also: - **analyzeClass:**, - **list:**

analyzeClass:

- **analyzeClass:**(const char *)*className*

Enters *className* into the **classNameField** and calls analyze:. Returns **self**.

See also: - **analyze:**

list:

- **list:***sender*

Displays in **text** the names of all known classes in the application. Returns **self**.

See also: - analyze: