Release 1.0 Copyright ©1993, 1994, 1996 by Don Yacktman. All Rights Reserved.

MiscTree

Inherits From: Object

Declared In: misckit/MiscTree.h

Class Description

The MiscTree implements a tree data structure. Each MiscTree contains a List object which points to the branch objects, which should be other MiscTree objects. You can create a new node for the tree using ±**alloc** and ±**init** and you can assign a label to the node when you initialize it with the ±**initLabel:** and ±**initLabelString:** methods. To add a branch object, simply use the ±**addbranch:** method. If you need to manipulate the branches in some way, you can obtain the List object containing the branches via the ±**branches** method. The ±**depth** and ±**width** methods return the depth and width of the tree below the messaged node.

Each node in the tree can have a label and a value, both of which are arbitrary character strings. You can set and retrieve the character strings with the ±**label**, ±**value**, ±**setLabel:**, and ±**setValue:** methods.

The tree may be dumped to an NXStream, with all its branches, by using the ±**dumpTree:level:indent** method. You can avoid having the branches below a MiscTree node printed when dumping the tree by collapsing the node. Use ±**collapse** to collapse the node and ±**uncollapse** to uncollapse it. The ±**collapsed** method will tell you if a particular node is collapsed. If you wish to exert more specific control over whether or not a node will be dumped, you can override the ±**moreData:level:indent:** method to add the required functionality. Overriding this method also allows you to add extra text to a node's line when a node is being dumped.

Note that this documentation is incomplete. It will be finished soon.

Method Types

Initializing the MiscTree	± init ± initLabel: ± initLabelString:
Dealing with branches	± addBranch: ± branches ± depth ± width
Labels and node values	± label ± setLabel:

	± setValue: ± value
Printing	± collapse ± collapsed ± dumpTree:level:indent: ± moreData:level:indent: ± uncollapse

Instance Methods

addBranch:

- **addBranch:**child

Adds the MiscTree node *child* to the end of the ªbranchesº List object. Returns *self*.

See also: -**branches**

branches
- **branches**

Returns a List object containing all of the ªchildrenº of this MiscTree node.

See also: -**addBranch:**

collapse
- **collapse**

This method is used to avoid having the branches below a MiscTree node printed when dumping the tree by collapsing the node. Returns *self*.

See also: -**collapsed** and -**uncollapse**

collapsed
- (BOOL)**collapsed**

This methods returns YES if the MiscTree node is collapsed.

See also: -**collapse** and -**uncollapse**

depth
- (int)**depth**

Returns the depth of the MiscTree. Depth means the longest distance from the receiver to the farthest leaf node.

See also: -**width**

dumpTree:level:indent:

- **dumpTree:**(NXStream *)*file* **level:**(int)*lev* **indent:**(const char *)ind

Dumps the MiscTree to an NXStream. This method should be sent to the top level of the tree with *lev* set to zero and *ind* being a string used for indenting. Returns *self*.

free

- **free**

Frees the children beneath the MiscTree node sent to. Returns *self*.

See also: **-initLabel:** and **-initLabelString:**

init

- **init**

Initializes a new MiscTree node with an empty label. Returns *self*.

See also: **-initLabel:** and **-initLabelString:**

initLabel:

- **initLabel:**(const char *)*newLabel*

Initializes a new MiscTree node with a label of *newLabel*. Returns *self*.

See also: **-init** and **-initLabelString::**

initLabelString:

- **initLabelString:**(MiscString *)*newLabel*

Initializes a new MiscTree node with a the value of the MiscString object *newLabel*. Returns *self*.

See also: **-initLabel:** and **-init**

label
- (const char *)**label**
Returns the a string containing the label.

See also: **-setLabel:**

moreData:level:indent:
- (BOOL)**moreData:**(NXStream *)*file* **level:**(int)*lev* **indent:**(const char *)*ind*

This method is up to the subclass to define. You can dynamically control collapsing (for example, cut off at a certain level, etc.) and also add info to the end of a dumped node's line from here. Be sure to message super

when you override this method; if the superclass method returns a NO then you should return a NO, regardless. Don't just use the notCollapsed instance variable, since it may change in the future; look at the return value from super!

Here is an example of how you might override to keep from printing levels deeper than level 2 (remember that the root level is zero):

```
- (BOOL)moreData:(NXStream *)file level:(int)level indent:(const char *)indent
{
   if ((level > 2) || ![super moreData:file level:level indent:indent]) {
      return NO;
   }
   return YES;
}
```

By default returns **![*self* collapased]**.

See also: **-dumpTree:level:indent:**

setLabel:

- **setLabel:**(const char *)*newLabel*

Sets the value of the label of the MiscTree node to *newLabel*. Returns *self*.

See also: **-label**

setValue:

- **setValue:**(MiscString *)*newValue*

Sets the value held by the MiscTree node to *newValue*. Frees the current value. (Maybe this should return the old value instead of returning self?) Returns *self*.

See also: **-value**

uncollapse
- **uncollapse**

Does the opposite of **-collapse**. Returns *self*.

See also: **-collapse** and **-collapsed**

value
- (const char *)**value**

Returns a pointer to the string holding the value of the MiscTree node.

See also: **-setValue:**

width
- (int)**width**

Returns the total width of the MiscTree node and it's children. Width means the total number of leaf nodes.

See also: **-depth**