

# SortedList

<b>Inherits From:</b>	List : Object
<b>Declared In:</b>	SortedList.h

## Class Description

The SortedList class is a subclass of List that inserts in sorted order, according to a method in it's agent. In addition, the SortedList class implements Browser delegation, so that it can automatically communicate with a browser in such a way as to display itself properly when requested to.

## Instance Variables

id agent;

agent

An object repsonsible for specific tasks

relating to the objects kept in the list - namely displaying, comparing, and determining leaf and subdirectory status for particular objects. This functionality is described by the SortedListAgent protocol.

## Method Types

Sorting

addObject:  
sort:

Object-specific info

displayStringForObjectAt:  
compareObjectAt:with:  
isLeafAt:  
subdirectoryForObjectAt:

Browser delegation

browser:fillMatrix:inColumn:

Archiving/Unarchiving

read:

write:

## Instance Methods

**addObject:**

- **addObject:***anObject*

Binarily determines where to insert *anObject* into the receiver, using **insertObject:at:** or **addObject:** when the correct location is found. Uses the **compareObjectAt:with:** method to determine pivots in the binary search.

**agent**

- **agent**

Returns the receiver's *agent*, an object that should respond to the SortedListAgent protocol.

**browser:fillMatrix:inColumn:**

- **browser:sender** **fillMatrix:matrix** **inColumn:(int)column**

Browser's delegate routine; fills *matrix* in *browser* with information from the receiver, or, the subdirectory of the receiver corresponding to *column*. Subdirectories are obtained by calls to **subdirectoryForObjectAt:** in the receiver. Setting **isLeaf:** on the browser cells in *matrix* is determined by calls to **isLeafAt:** in the receiver. Returns the number of cells created and added to *matrix*.

**compareObjectAt:with:**

- (int)**compareObjectAt:(int)at** **with:object**

Using the agent method **compare:with:sender:**, returns an integer which, if negative, indicates the *at* object is less than *object*; if zero, indicates the *at* object is equal to *object*; if positive, indicates the *at* object is greater than *object*.

### **displayStringForObjectAt:**

- (const char \*)**displayStringForObjectAt:**(int)*at*

Returns a read-only string that is typically used to display the *at* object in a browser. The *agent* **displayStringFor:sender:** method is used to obtain this string.

### **isLeafAt:**

- (BOOL)**isLeafAt:**(int)*at*

Returns essentially [agent isLeaf:[self objectAt:at] sender:self].

**read:**

- **read:**(NXTypedStream \*)*stream*

Reads an instance of the timer from *stream*.

**setAgent:**

- **setAgent:***anAgent*

Set's the *agent* instance variable, an object that should respond to the SortedListAgent protocol.

**sort:**

- **sort:***sender*

Sorts the receiver according to the **compareObjectAt:with:** method. Implementation details: copies list, empties self, iteratively calls [self addObject:].

**subdirectoryForObjectAt:**

- **subdirectoryForObjectAt:**(int)*at*

Returns the SortedList object, or nil, which is the subdirectory of the object at *at* in the receiver. Determines this by calling the agent method

**subdirectoryFor:sender:**.

**write:**

- **write:**(NXTypedStream \*)*stream*

Writes the timer to *stream*.