

# MiscValueField

**Inherits From:** TextField : Control : View : Responder : Object

**Declared In:** MiscValueField.h

## Class Description

**MiscValueField** is a simple subclass of **TextField** which defaults to using the **MiscValueCell** to display its contents. It also implements a set of methods to allow you to send messages that get passed through to the **MiscValueCell** for various configuration needs. This allows exactly the same interface when the **MiscValueCell** is used in other places (like a **Matrix**). A **MiscStringArray** or a **StringList** (found as a **MiniExample**) can be attached to provide text strings that get shown instead of numbers.

A valid string list object responds to:

-(char \*) **StringAt**:(int)item;  
-(unsigned int) **count**;

The value sent to **StringAt** will be in the range 0 - **count**. When the **stringList** instance variable is set, the ranges as set in IB (or wherever) are ignored and the string list defines the range of its values. Of course, **StringAt** can create its return value any way it likes.

## Instance Variables

id          stringList;

stringList

An object to provide strings in place of values.

## Method Types

Initializing

+ initWithFrame:

Manipulating a MiscValueField

± setMinValue:

± setMaxValue:

± setMinBoundary:

± setMaxBoundary:

Querying values

± **setStepSize:**  
± **setAltStepSize:**  
± **setExpandMin:**  
± **setExpandMax:**  
± **setStringList:**

± **minValue**  
± **maxValue**  
± **minBoundary**  
± **maxBoundary**  
± **stepSize**  
± **altStepSize**  
± **expandMin**  
± **expandMax**  
± **stringList**

## Instance Methods

### **altStepSize**

±(float) **altStepSize**

Returns the amount the value will change when a button is clicked while Alternate is held down.

See also: ± **setAltStepSize:**, ± **setStepSize:**, ± **stepSize**

## **expandMax**

±(BOOL) **expandMax**

Returns YES if **maxBoundary** is in effect stopping the arrow buttons from reaching **maxValue**.

See also: ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

## **expandMin**

±(BOOL) **expandMin**

Returns YES if **minBoundary** is in effect stopping the arrow buttons from reaching **minValue**.

See also: ± **expandMax**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

## **initWithFrame:**

± **initWithFrame:**(const NXRect \*)*frame*

Initializes and returns the receiver with a MiscValueCell inside it. Returns **self**.

## **maxBoundary**

±(double) **maxBoundary**

Returns the current limit that the value will stop at when the up arrow button is being used to change the value. This limit can be changed by entering a larger value using the keyboard. This limit will be set the highest value entered that is still within the bounds of **maxValue**. It cannot be reduced except programmatically.

See also: ± **expandMax**, ± **expandMin**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

**maxValue**

±(double) **maxValue**

Returns the highest value this field is allowed to reach. This limit cannot be exceeded in any way. **maxBoundary** can match this value and then have no effect.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

**minBoundary**

±(double) **minBoundary**

Returns the current limit that the value will stop at when the down arrow button is being used to change the value. This limit can be changed by entering a smaller value using the keyboard. This limit will be set the lowest value entered that is still within the bounds of **minValue**. It cannot be increased except programmatically.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minValue**, ± **setExpandMax**, ±

€setExpandMin, ± setMaxBoundary, ± setMaxValue, ± setMinBoundary, ± setMinValue

### **minValue**

±(double) **minValue**

Returns the lowest value this field is allowed to reach. This value cannot be exceeded in any way. **minBoundary** can match this value and then have no effect.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **setExpandMax**, ± **€setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

### **setAltStepSize:**

± **setAltStepSize:**(float)*size*

Sets the amount the value will change when a button is clicked while Alternate is being held down. Returns **self**.

See also: ± **altStepSize:**, ± **setStepSize:**, ± **stepSize**

### **setExpandMax:**

± **setExpandMax:**(BOOL)*flag*

If *flag* is YES then **maxBoundary** has an effect on the upper limit of the field value when it's adjusted with the arrow buttons. Returns **self**.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

### **setExpandMin:**

± **setExpandMin**:(**BOOL**)*flag*

If *flag* is YES then **minBoundary** has an effect on the lower limit of the field value when it's adjusted with the arrow buttons. Returns **self**.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

### **setMaxBoundary**

± **setMaxBoundary**:(**double**)*value*

Sets the highest value the field will allow when using the arrow buttons to adjust the value. This value can be exceeded and altered by typing a value greater than this limit. This value will be adjusted to match the entered amount with an additional limitation that it will stop at **maxValue**. Returns **self**.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxValue**, ± **setMinBoundary**, ± **setMinValue**

### **setMaxValue**

± **setMaxValue**:(double)*value*

Sets the highest value the field will allow. There is no way to exceed this limit from the interface. **maxBoundary** will stop expanding when it gets to this value. Returns **self**.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMinBoundary**, ± **setMinValue**

### **setMinBoundary**

± **setMinBoundary**:(double)*value*

Sets the lowest value the field will allow when using the arrow buttons to adjust the value. This value can be exceeded and altered by typing a value lower than this limit. This value will be adjusted to match the entered amount with an additional limitation that it will stop at **minValue**. Returns **self**.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinValue**

### **setMinValue**

± **setMinValue**:(double)*value*

Sets the lowest value the field will allow. There is no way to exceed this limit from the interface. **minBoundary** will stop expanding when it gets to this value. Returns **self**.

See also: ± **expandMax**, ± **expandMin**, ± **maxBoundary**, ± **maxValue**, ± **minBoundary**, ± **minValue**, ± **setExpandMax**, ± **setExpandMin**, ± **setMaxBoundary**, ± **setMaxValue**, ± **setMinBoundary**

### **setStepSize:**

± **setStepSize:**(float)*size*

Sets the amount the value will change when a button is clicked. Returns **self**.

See also: ± **altStepSize:**, ± **setAltStepSize:**, ± **stepSize**

### **setStringList:**

± **setStringList:***anObject*

Sets the object to be asked for display strings. *anObject* should respond to **stringAt:** and **count** messages. When a string list is set the range limits are all ignored and the field is set to non-editable. The value of the field will only fit within a range defined by *anObject*'s **count** method. Returns **self**.

*anObject*'s **stringAt:** method should take an **int** as its argument and return a **char \*** with the correct string to be displayed for the given value. The value of the argument will range from 0 to **count** - 1. This arrangement has been created to allow the MiscValueField to be connected to a StringList (found in the MiniExamples) object in Interface Builder so lists may be created, displayed and dealt with with no additional code.

See also: ± **stringList**

### **stepSize**

$\pm$ (float) **stepSize**

Returns the amount the value will change when a button is clicked.

See also:  $\pm$  **altStepSize:**,  $\pm$  **setAltStepSize:**,  $\pm$  **setStepSize:**

**stringList**

$\pm$  **stringList**

Returns the object that is taking the task of providing the display strings.

See also:  $\pm$  **setStringList:**