

# MiscUserPath

**Inherits From:** NSObject  
**Declared In:** MiscUserPath.h

## Class Description

The MiscUserPath class facilitates the use of Postscript user paths. It handles automatic allocation and growth of the operator and operand arrays, as well as the calculation of the bounding box. The method **send:cached:** sends the user path to the window server. A user path may be optionally cached down at the window server, or repeatedly sent down. The methods **moveto::**, **rmoveto::**, **lineto::**, **rlineto:**, **curveto::::**, **rcurveto::::**, **arc::::**, **arcn::::**, **arct::::** fill the operator and operand arrays enlarging the arrays if necessary and recalculating the bounding box.

Because the functionality of a user path cannot be fully encapsulated in a class (for example the arrays once set cannot be modified easily to reflect a translation or rotation) the MiscUserPath class provides methods for querying and modifying all instance variables. The method **resetFill** empties the arrays but keeps the allocated memory.

## Instance Variables

```
unsigned int maxLengthParams;  
unsigned int maxLengthOps;  
BOOL ping;  
float *bboxParams;  
float *bbox;
```

```
float *params;  
unsigned int lengthParams;  
unsigned int lengthOps;  
char *ops;  
char *bboxOps;  
NSPoint cp;
```

maxLengthParams	Size of operand array.
maxLengthOps	Size of operator array.
ping	True if calling NSPing() after sending path to window server.
bboxParams	Pointer to operand+bounding box array.
bbox	Pointer to bounding box (bbox == bboxParams).
params	Pointer to operand array (params == &bboxParams [4]).
lengthParams	Size of operand fill.
lengthOps	Size of operator fill.
ops	Pointer to operator array (ops == &bboxOps[2]).
bboxOps	Pointer to operator+bounding box operators array.
cp	Current point.

## Method Types

Allocating and initializing

- initCountParams:countOps:

Accessing instance variables

- bounds  
- currentPoint  
- setCurrentPoint:  
- params  
- bbox

- bboxParams
- ops
- bboxOps
- lengthParams
- lengthOps
- maxLengthParams
- maxLengthOps
- increaseLengthParamsBy:
- increaseLengthOpsBy:
- doubleLengthParams
- doubleLengthOps
- setLengthParams:
- setLengthOps:
- setMaxLengthParams:
- setMaxLengthOps:

Defining user path

- resetFill
- moveto::
- rmoveto::
- lineto::
- rlineto::
- curveto:::::
- rcurveto:::::
- arc:::::
- arcn:::::
- arct:::::
- closepath

Sending to window server

- debugSend:cached:
- send:cached:

## Instance Methods

**arc:::::**

- (void)**arc**:(float)x :(float)y :(float)r :(float)ang1 :(float)ang2

Adds the arc operator (*dps\_arc*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

#### **arcn:::::**

- (void)**arcn**:(float)x :(float)y :(float)r :(float)ang1 :(float)ang2

Adds the arcn operator (*dps\_arcn*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

#### **arct:::::**

- (void)**arct**:(float)x1 :(float)y1 :(float)x2 :(float)y2 :(float)r

Adds the arct operator (*dps\_arct*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

#### **bbox**

- (float \*)**bbox**

Returns a pointer to the bounding box of the user path. The pointer points to the start of the **bboxParams** array. The first four float values of this array represent the bounding box of the user path. Don't reallocate the memory pointed to by this pointer. If you want to change the memory size use the methods **doubleLengthParams**, **increaseLengthParamsBy**: and then ask again for the pointer with this method because the reallocating methods invalidate pointers previously returned by **bbox**.

#### **bboxOps**

- (char \*)**bboxOps**

Returns a pointer to the **bbox** operator array. Don't reallocate the memory pointed to by this pointer.

#### **bboxParams**

- (float \*)**bboxParams**

Convenience method. Returns the same pointer as **bbox**. Same restriction apply.

### **bounds**

- (NSRect)**bounds**

Returns a NSRect with the bounding box of the user path.

### **closepath**

- (void)**closepath**

Adds the closepath operator (*dps\_closepath*) to the operator array.

### **currentPoint**

- (NSPoint)**currentPoint**

Returns the current point of the user path.

### **curveto:::::**

- (void)**curveto**:(float)x1 :(float)y1 :(float)x2 :(float)y2 :(float)x3 :(float)y3

Adds the curveto operator (*dps\_curveto*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

### **debugSend:cached:**

- (void)**debugSend**:(int)*op* **cached**:(BOOL)*cache*

Sends the user path with the rendering operation *op* to the window server and calls an NSPing() afterwards. If *cached* is YES the user path will be cached at the window server. *op* should be one of the following values: *dps\_ustroke*, *dps\_ufill*, *dps\_ueofill*, *dps\_ustrokepath*, *dps\_uappend*, *dps\_inufill*, *dps\_inueofill*, *dps\_inustroke*.

**doubleLengthOps**

- (void)**doubleLengthOps**

Doubles the size of the operator array.

**doubleLengthParams**

- (void)**doubleLengthParams**

Doubles the size of the operand array.

**increaseLengthOpsBy:**

- (void)**increaseLengthOpsBy:(unsigned int)extraLength**

Increases the size of the operator array by *extraLength*.

**increaseLengthParamsBy:**

- (void)**increaseLengthParamsBy:(unsigned int)extraLength**

Increases the size of the operand array by *extraLength*.

**initCountParams:countOps:**

- **initCountParams:(int)numParams countOps:(int)numOps**

Initializes a newly allocated user path object so that the operator array has the size *numOps* and the operand array has the size *numParams*.

**lengthOps**

- (unsigned int)**lengthOps**

Returns the amount of fill in the operator array.

### **lengthParams**

- (unsigned int)**lengthParams**

Returns the amount of fill in the operand array.

### **lineto::**

- (void)**lineto**:(float)x :(float)y

Adds the lineto operator (*dps\_lineto*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

### **maxLengthOps**

- (unsigned int)**maxLengthOps**

Returns the size of the operator array.

### **maxLengthParams**

- (unsigned int)**maxLengthParams**

Returns the size of the operand array.

### **moveto::**

- (void)**moveto**:(float)x :(float)y

Adds the moveto operator (*dps\_moveto*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

### **ops**

- (char \*)**ops**

Returns a pointer to the operator array. Don't reallocate the memory pointed to by this pointer. If you want to change the memory size use the methods **doubleLengthOps**, **increaseLengthOpsBy**: and then ask again for the pointer with this method because the reallocating methods invalidate pointers previously returned by **ops**.

### **params**

- (float \*)**params**

Returns a pointer to the operand array. Don't reallocate the memory pointed to by this pointer. If you want to change the memory size use the methods **doubleLengthParams**, **increaseLengthParamsBy**: and then ask again for the pointer with this method because the reallocating methods invalidate pointers previously returned by **params**.

### **rcurveto:.....**

- (void)**rcurveto**:(float)*dx1* :(float)*dy1* :(float)*dx2* :(float)*dy2* :(float)*dx3* :(float)*dy3*

Adds the rcurveto operator (*dps\_rcurveto*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

### **resetFill**

- (void)**resetFill**

Empties the operator and operand arrays. Size of the arrays remains constant.

### **rlineto::**

- (void)**rlineto**:(float)*x* :(float)*y*

Adds the rlineto operator (*dps\_rlineto*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

### **rmoveto::**

- (void)**rmoveto**:(float)*x* :(float)*y*

Adds the *rmoveto* operator (*dps\_rmoveto*) to the operator array and the arguments to the operand array. Enlarges arrays if necessary by doubling the current size. Recalculates the bounding box of the user path.

**send:cached:**

- (void)**send:(int)op cached:(BOOL)cache**

Sends the user path with the rendering operation *op* to the window server and calls an *NSPing()* afterwards. If *cached* is YES the user path will be cached at the window server. *op* should be one of the following values: *dps\_ustroke*, *dps\_ufill*, *dps\_ueofill*, *dps\_ustrokepath*, *dps\_uappend*, *dps\_inufill*, *dps\_inueofill*, *dps\_inustroke*.

**setCurrentPoint:**

- (void)**setCurrentPoint:(NSPoint)aPoint**

Sets the current point of the user path to *aPoint*.

**setLengthOps:**

- (void)**setLengthOps:(unsigned int)length**

Sets the amount of current operator fill to *length*.

**setLengthParams:**

- (void)**setLengthParams:(unsigned int)length**

Sets the amount of current operand fill to *length*.

**setMaxLengthOps:**

- (void)**setMaxLengthOps:(unsigned int)length**

Reallocates the operator array to size *length*. Invalidates pointers previously returned by **ops**.

**setMaxLengthParams:**

- (void)setMaxLengthParams:(unsigned int)*length*

Reallocates the operand array to size *length*. Invalidates pointers previously returned by **bbox** and **params**.