

# MiscSavePanel

**Inherits From:** SavePanel : Panel : Window : Responder : Object

**Declared In:** MiscSavePanel.h

## Class Description

The MiscSavePanel class adds an append option to the SavePanel. It is particularly useful when the user may want to save data into a pre-existing file without altering the original contents (e.g. saving mail messages).

When appending is enabled (i.e. appendOK == YES) and a pre-existing file is selected by the user, an alert panel gives the user the choice of replacing the current file contents, appending to the current file contents, or canceling the selection. Note that when in append mode, all panels that MiscSavePanel may display (e.g. the file browser or alert panel) are titled "Save/Append" instead of the usual "Save."

If appending is disabled and a pre-existing file is selected by the user, an alert panel gives the user the choice of

replacing the current file contents or canceling the selection (this behavior is identical to that of the `SavePanel`). Since appending is disabled by default, the default behavior of the `MiscSavePanel` is to be like a `SavePanel`.

Each application has only one instance of the `MiscSavePanel`; that object is shared by all references. The **new** method returns the `MiscSavePanel` instance (creating it if necessary). As a result, some generic panel features are disabled in the `MiscSavePanel` class. The allocation, deallocation, initialization, and copying methods have been overridden either to return **nil** or to generate an error and terminate.

Note that, as with the `SavePanel`, no data is actually saved, appended, or replaced by `MiscSavePanel`'s methods.

See the class description for `SavePanel` for more information.

Typical usage:

```
miscSavePanel = [[MiscSavePanel new] allowAppend:YES];
if ([miscSavePanel runModal]) {
    file = (char *) [miscSavePanel filename];
    if ([miscSavePanel appending]) {           // if append requested
        // open file and seek to end
    } else {
        // creat file (open w/truncation)
    }
    // write to file
}
```

## Instance Variables

BOOL  
BOOL

**appendOK;**  
**appMode;**

appendOK

A boolean that indicates whether appending is enabled.

appMode

A boolean that indicates whether appending was selected by the user for the most recent file selection.

## Method Types

Creation and freeing

+ alloc  
+ allocFromZone:(NXZone \*)zone  
+ new  
+ newContent:style:backing:buttonMask:defer:  
- init

Configuring the panel

± allowAppend:

Querying the panel

± appending

## Class Methods

**alloc**

+ **alloc**

Disables the inherited **alloc** method to prevent multiple MiscSavePanels from being created. There is only one instance of MiscSavePanel object for each application; you can access it through either of the **new...** methods. Returns **nil**.

**See also:** + **new**, + **newContent:style:backing:buttonMask:defer:**

**allocFromZone:**

- **allocFromZone:**(NXZone \*)*zone*

Disables the inherited **allocFromZone** method to prevent multiple MiscSavePanels from being created. There is only one instance of MiscSavePanel object for each application; you can access it through either of the **new...** methods. Returns **nil**.

**See also:** + **new**, + **newContent:style:backing:buttonMask:defer:**

**new**

+ **new**

Returns a MiscSavePanel object by invoking the **newContent:style:backing:buttonMask:defer:** method. An application has no more than one instance of MiscSavePanel, so this method either returns the previously created object (if it exists) or creates a new one. This method is inherited.

**See also:** + new

**newContent:style:backing:buttonMask:defer:**

+ **newContent:**(const NXRect \*)*contentRect*  
  **style:**(int)*aStyle*  
  **backing:**(int)*bufferingType*  
  **buttonMask:**(int)*mask*  
  **defer:**(BOOL)*flag*

Returns a MiscSavePanel object. An application has no more than one instance of MiscSavePanel, so this method either returns the previously created object (if it exists) or creates a new one. The arguments are ignored. This is the designated **new...** method of the MiscSavePanel class.

**See also:** + new

## Instance Methods

**\_validateNames:checkBrowser:**

± (BOOL)\_**validateNames:**(char \*)*name* **checkBrowser:**(BOOL)*check*

This method overrides the method in SavePanel that checks for pre-existing files and pops up an alert panel when one is specified. As far as I know it is not documented so it may change thereby breaking MiscSavePanel. Caveat emptor and all. If an existing file is specified by the arguments, an alert panel is displayed. The choices on the alert panel are as described under **allowAppend:**. This routine returns YES unless the user chooses Cancel in which case it returns NO.

### **allowAppend:**

± **allowAppend:**(BOOL)*flag*

Determines the MiscSavePanel's actions when the user selects a pre-existing file. If *flag* is YES then the MiscSavePanel will display an alert panel offering a three-way choice among: Append (default), Replace, and Cancel. If *flag* is NO, then the alert panel will only offer the Replace (default), and Cancel options. The setting of *flag* also controls whether the MiscSavePanel's file browser and alert panel will be titled "Save/Append" or just "Save."

Returns **self**.

### **appending**

-€(BOOL)**appending**

Returns YES if the path returned by the **filename** method (SavePanel) was specified as an append target by the user. Notice that the **appending** method, like the **filename** method, gives information about the last selection made, not the current configuration of the MiscSavePanel; thus its return value is not directly affected by the

**allowAppend:** method.

**See also:** `-filename` (SavePanel)

**free**

- **free**

Returns **self**, without freeing anything. The MiscSavePanel may not be freed once it has been created within an application.

**init**

- **init**

Returns **nil** and does nothing. The MiscSavePanel is initialized automatically when it is created.