

Version 1.0 Copyright ©1993, 1994, 1995 by Mike Ferris. All Rights Reserved.
Mike Ferris - February 16th, 1994. Modified for the MiscKit January 27, 1995 by Don Yacktman.

MiscDocType

Inherits From: Object

Declared In: misckit/MiscDocType.h

Class Description

MiscDocType objects are used by MiscDocController to keep track of the file types that a particular type of MiscDocController can read and/or save. MiscDocType objects are very data-centric—they don't do much besides manage their instance variables.

MiscDocType objects are created with the MiscDocController method

+addDocType:name:extension:openSelector:saveSelector:. Each subclass of MiscDocController should call that method at least once from its **+initialize** method. That's how the MiscDocController subclasses know what file types they can open and save and how to do it.

MiscDocType keeps track of several things about a particular file type. First, it has a tag which is not used by the MiscKit. You can use the tag for anything you want. The document type name and the file extension used for documents of that type are also stored. The name is used by the Save panel if your MiscDocController subclass supports multiple save types. The extension is used all over the place (in Open and Save panels especially). Finally, each MiscDocType stores two selectors, either one of which may be NULL (but not both). The openSelector tells what method of your MiscDocController subclass to call to read in a file of that particular type. If the MiscDocController cannot read files of that type, the selector is NULL. The saveSelector tells what method to call to save a file of that type. Both of the selectors should be set to methods that take a MiscDocType object as its single argument.

Instance Variables

Class	controllerClass;
int	typeTag;
MiscString	*typeName;
MiscString	*typeExtension;

SEL **openSelector;**
SEL **saveSelector;**

controllerClass	The MiscDocController which owns this type.
typeTag	an integer tag which can be used for any purpose you desire.
typeName	a full name for this file type (ie "Rich Text").
typeExtension	the file extension used by files of this type (ie ".rtf").
openSelector	the selector of the method to use to read files of this type. The method should take a single argument which will be this MiscDocType. If NULL, then the controller does not support reading this type.
saveSelector	the selector of the method to use to save files of this type. The method should take a single argument which will be this MiscDocType. If NULL, then the controller does not support saving this type.

Method Types

Initializing the class

+ initialize

Initializing instances

- init

- initWithControllerClass:tag:name:extension:openSelector:saveSelector:

- free

Querying the contents

- controllerClass

- typeTag

- typeName

- typeExtension

- canOpenType

- openSelector

- canSaveType

- saveSelector

Archiving

- read:

- write:

Class Methods

initialize

+ **initialize**

Sets the class version number.

Instance Methods

canOpenType

± (BOOL)**canOpenType**

Returns whether this document type can be opened. It just checks to see if the openSelector is non-NULL.

See also: ± **controllerClass**, ± **typeTag**, ± **typeName**, ± **typeExtension**, ± **openSelector**, ± **canSaveType**, ± **saveSelector**

canSaveType

± (BOOL)**canSaveType**

Returns whether this document type can be saved. It just checks to see if the saveSelector is non-NULL.

See also: `± controllerClass`, `± typeTag`, `± typeName`, `± typeExtension`, `± canOpenType`, `± openSelector`,
`± saveSelector`

controllerClass

`± (Class)controllerClass`

Returns the owning class.

See also: `± typeTag`, `± typeName`, `± typeExtension`, `± canOpenType`, `± openSelector`, `± canSaveType`, `± saveSelector`

free

`± free`

This method frees the MiscStrings.

See also: `± init`, `± initWithControllerClass:tag:name:extension:openSelector:saveSelector:`

init:

± init

Calls **initWithControllerClass:nil tag:-1 name:NULL extension:NULL openSelector:NULL saveSelector:NULL**. This method should not be used as it produces useless types and you can't change them.

See also: ± **initWithControllerClass:tag:name:extension:openSelector:saveSelector:**, ± **free**

initWithControllerClass:tag:name:extension:openSelector:saveSelector:

± **initWithControllerClass:(Class)controllerClass tag:(int)tag name:(const char *)name extension:(const char *)ext openSelector:(SEL)openSel saveSelector:(SEL)saveSel**

This is the designated initializer for the class. It allocates and initializes the two MiscString instance variables, and initializes the rest. No methods for changing the instance variables are provided, so the object must be completely configured through this method.

See also: ± **init**, ± **free**

openSelector

± (SEL)**openSelector**

Returns the selector of the method used to read files of this type. NULL means this type can't be read.

See also: ± **controllerClass**, ± **typeTag**, ± **typeName**, ± **typeExtension**, ± **canOpenType**, ± **canSaveType**, ± **saveSelector**

read:

± **read:**(NXTypedStream *)*stream*

Reads the instance from the typed stream. Only the frame name is archived.

See also: ± **awake**, ± **write:**

saveSelector

± (SEL)**saveSelector**

Returns the selector of the method used to save files of this type. NULL means this type can't be saved.

See also: ± **controllerClass**, ± **typeTag**, ± **typeName**, ± **typeExtension**, ± **canOpenType**, ± **openSelector**, ± **canSaveType**

typeExtension

± (const char *)**typeExtension**

Returns the type's file name extension.

See also: ± **controllerClass**, ± **typeTag**, ± **typeName**, ± **canOpenType**, ± **openSelector**, ± **canSaveType**, ± **saveSelector**

typeName

± (const char *)**typeName**

Returns the type's name.

See also: ± **controllerClass**, ± **typeTag**, ± **typeExtension**, ± **canOpenType**, ± **openSelector**, ± **canSaveType**, ± **saveSelector**

typeTag

± (int)**typeTag**

Returns the type's tag.

See also: ± `controllerClass`, ± `typeName`, ± `typeExtension`, ± `canOpenType`, ± `openSelector`, ± `canSaveType`, ± `saveSelector`

write:

± `write:(NXTypedStream *)stream`

Writes the instance to the typed stream. Only the frame name is archived.

See also: ± `awake`, ± `read`: