NS_DEV_DOCFOR:objc_class:Text;,NIDirectory

Inherits From: Object

Conforms To:

Declared In: NIDirectory.h

Class Description

The NIDirectory class defines an object that manages a NetInfo directory. Your application inherits these uses of the NIDirectory class when it incorporates any of the specific objects into its source. You never have to subclass this class directly. This class uses the structures defined in nikit/NIDomain.h and netinfo/ni.h.
This object may display several attention and warning panels. The strings displayed are fully localisable. They are defined in the file NIDirectory.strings.

Class Variables

None accessable from other units.

Instance Variables

id **delegate**;

```
char *domainName;
char *dirName;
char *baseName;
char *fullPath;
char *saveTitle;
char *user;
BOOL mayChangeName;
NIDomain *domain;
ni_id directory;
ni_proplist properties;
HashTable *hash;
List *list;
```

delegate	The object that's notified when the NIProperty object modifies a NetInfo entry.
domainName	The string with the domain or NULL.
dirName	The string with the root of the directory.
baseName	The string with the directory or NULL.
fullPath	A concattenation of domainName, dirName, and baseName or NULL.
saveTitle	The string with the title to use for the saveToDomain panel.
user	The name of the user for which was last authenticated.
mayChangeName	A flag to allow/disallow changing the name before saving.
domain	A domain object of the domain last authenticated.
directory	The id of the (possibly modified) NetInfo directory loaded with the NIDirectory object.
properties	The contents of the (possibly modified) NetInfo directory loaded with the NIDirectory object.
hash	A hashlist used to locate the registered properties of the directory by name.
list	The list of all registered properties.

Method Types

Creating a new NIDirectory object	+ new:root:directory:
Loading an NIDirectory object	+ open:root:withTitle:
Initializing an NIDirectory object	+ - init:dom:root:dir:errors:
Freeing a NIPropery object	- close
Transfering from NetInfo	- scan - reset
Saving	- save - saveToDomain - setAllowChangName:
Deleting	- delete
Setting the save panel's title	- setSaveTitle:
Setting the user to authenticate	- setAuthenticationUser:
Getting the domain and directory	- baseName - directory - directoryID - domain - domainName - handle
Adding a property	- addBool:outlet: - addBrowser:browser:text:add:remove: - addBrowser:browser:add:remove:mode:path:title:

	- addCall:displayAction: - addChar:outlet: - addInt:text:slider:zero: - addPopup:outlet: - addPopup:outlet:default: - addProperty: - addString:outlet: - addString:text:button:mode:path:title:
Getting a property	- property:
Assigning a delegate	- delegate - setDelegate:

Class Methods

NS_DEV_DOCFOR:objc_method:[Text-excludeFromServicesMenu:];,new:root:directory

+ **new:***sender* **root:**(const char *)rootPath **directory**:(const char *)*dirPath*

Creates a new empty directory object. *RootPath* is the root to the NetInfo directory into which the properties will be written to. If *dirPath* is non-null, this will be the mandatory (sub)directory to which to write the properties. Returns the created, empty NIDirectory object if successful or nil upon failure.

NS_DEV_DOCFOR:objc_method:[Text-getDefaultFont];,open:root:withTitle

+ **open:**sender **root**:(const char *)*rootPath* **withTitle**:(const char *)*title*

Creates a new directory object, shows the NetInfo open panel and reads the selected NetInfo directory. Returns the loaded NIDirectory object if successful or nil upon failure.

Instance Methods

NS_DEV_DOCFOR:objc_method:[Text-free];,**addBool:outlet:**

- **addBool:**(const char *)*label* **outlet**:*obj*

Adds a boolean property with the name *label* and assigns it the GUI object *obj*. Returns the NIProperty object if successful. The GUI object is expected to be a matrix of three radio buttons. The first button is taken as the default, the second the value for true, and the last represents a value of false.

See also: - **property, Delegates Implemented** (NIProperty), - **NS_DEV_DOCFOR:objc_method:[Text-initFrame:text:alignment:];,init:properties:name:outlet:** (NIBoolProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addBrowser:browser:text:add:remove:**

- **addBool:**(const char *)*label* **browser**:*bObj* **text**:*tObj* **add**:*aObj* **remove**:*dObj*

Adds a browser property with the name *label* and assigns it the GUI objects. A browser object consists of a browser (*bObj*), a single text field (*tObjs*), a button for addition (*aObj*), and a button for removal (*dObj*). Returns the NIProperty object if successful. The GUI object is expected to be a matrix of three radio buttons.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addBrowser:browser:add:remove:mode:path:title:**

- **addBool:**(const char *)*label* **browser**:*bObj* **add**:*aObj* **remove**:*dObj* **mode**:(int)*m* **path**:(const char *)*p* **title**:(const char *)*tString*

Same as addBrosser:browser:text:add:remove, but the GUI object consists of a browser (*bObj*), a button for addition (*aObj*), and a button for removal (*dObj*). When the user clicks the add button, a panel allowing to select

a file (NIPT_FILE), a directory (NIPT_DIR), or a NetInfo value (NIPT_NETINFO) will be shown, depending on what *mode* was set to. When specifying NIPT_NETINFO for *mode*, *path* must contain the root directory from which the value is to be taken (e.g. `/users'), in all other cases, *p* may be NULL. *Title* is a string shown in the panels, explaining the purpose of the selection. The values of the strings *p* and *tString* must not change during the lifetime of the object.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addCall:displayAction**

- **addCall:**(const char *)*label* **displayAction**:(SEL)*action*

Adds a calling property with the name *label*. Returns the NIProperty object if successful. A call object can have any corresponding GUI object or none. The method *action* is called whenver the property's value should be (re)displayed. The *action* takes only one argument: the sender.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addChar:outlet:**

- **addChar:**(const char *)*label* **outlet**:*obj*

Adds a character property with the name *label* and assigns it the GUI object *obj*. Returns the NIProperty object if successful. The GUI object is expected to be a single textfield element.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addInt:outlet:**

- **addInt:**(const char *)*label* **text**:*tObj* **slider**:*sObj* **zero**:(const char *)*string*

Adds an integer property with the name *label*. Returns the NIProperty object if successful. The GUI object is

expected to be at least a single textfield (*tObj*). An optional slider (sObj) may be specified. Neither the minimum nor the maximum values are set. The default value is assumed to be -1. *Zero* is an optional text to be displayed for the value 0. The value of the string *string* must not change during the lifetime of the object.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addPopup:outlet:**

- **addPopup:**(const char *)*label* **outlet**:*obj*

Adds a popup property with the name *label* and assigns it the GUI object *obj*. Returns the NIProperty object if successful. The GUI object is expected to be the target button of a popup menu.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addPopup:outlet:default:**

- **addPopup:**(const char *)*label* **outlet**:*obj* **default**:(conat char *)*defString*

Adds a popup property with the name *label* and assigns it the GUI object *obj*. Returns the NIProperty object if successful. The GUI object is expected to be the target button of a popup menu. *DefString* is the string to show if no property is defined. The value of the string *defString* must not change during the lifetime of the object.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addProperty:**

- **addString:**(const char *)*label*

Adds a property with the name *label*. Returns the NIProperty object if successful. There is no corresponding GUI object for this entry. The values of the property can be accessed with the methods given in the NIProperty class.

NS_DEV_DOCFOR:objc_method:[Text-free];,**addString:outlet:**

- **addString:**(const char *)*label* **outlet**:*obj*

Adds a string property with the name *label* and assigns it the GUI object *obj*. Returns the NIProperty object if successful. The GUI object is expected to be a single textfield. It may not be a textfield from an ALT-dragged matrix of textfields.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**addString:text:button:mode:path:title:**

- **addString:**(const char *)*label* **text**:*tObj* **button**:*bObj* **mode**:(int)*m* **path**:(const char *)p **title**:(const char *)*tString*

Adds a string property with the name *label*. Returns the NIProperty object if successful. The GUI object is expected to consist of a single textfield (*tObj*) and a button (*bObj*). When the user clicks the button, a panel allowing to select a file (NIPT_FILE), a directory (NIPT_DIR), or a NetInfo value (NIPT_NETINFO) will be shown, depending on what *mode* was set to. When specifying NIPT_NETINFO for *mode*, *path* must contain the root directory from which the value is to be taken (e.g. `/users'), in all other cases, *p* may be NULL. *Title* is a string shown in the panels, explaining the purpose of the selection. The values of the strings *p* and *tString* must not change during the lifetime of the object.

See also: - **property, Delegates Implemented** (NIProperty)

NS_DEV_DOCFOR:objc_method:[Text-free];,**baseName**

- (const char *)**basename**

Returns the name of the (sub)directory to which the values will be stored, if any was defined.

NS_DEV_DOCFOR:objc_method:[Text-free];,close

- **close**

Releases the storage for a NIDirectory object. It does not save any changed values back to NetInfo.

NS_DEV_DOCFOR:objc_method:[Text-delegate];,delegate

- **delegate**

Returns the NIProperty object's delegate.

See also: - **setDelegate:**

NS_DEV_DOCFOR:objc_method:[Text-delegate];,delete

- **delete**

Removes the directory from NetInfo. Does not close or free the NIDirectory object.

Returns self on sucess, nil on failure.

See also: - **setDelegate:**

NS_DEV_DOCFOR:objc_method:[Text-delegate];,directory

- (const char *)**directory**

Returns the root directory of a domain to which the values will be stored, if any was defined.

NS_DEV_DOCFOR:objc_method:[Text-delegate];,directoryID

- (ni_id)**directoryID**

Returns the directory ID of the directory last read/authenticated.

NS_DEV_DOCFOR:objc_method:[Text-getLocation:ofCell:];,NS_DEV_DOCFOR:objc_method:[Text-free];,domain

- (NIDomain *)**domain**

Returns the domain object of the domain last read/authenticated.

NS_DEV_DOCFOR:objc_method:[Text-free];,domainName

- (const char *)**domainName**

Returns the name of the domain to which the values will be stored, if any was defined.

handle

- (void *)**handle**

Returns the handle to the domain last read/authenticated.

init:dom:root:dir:errors:

- **init:**sender **dom:**(const char *)*domPath* **root:**(const char *)*basePath* **dir:**(const char *)*dirPath* **errors:** (BOOL)*warn*

Initializes an allocated NIDirectory object. This method should only be called if a directory is to be checked for or an existing directory is to be loaded without displaying an open panel. The class +new and +open methods call

this method, thus eliminating the neccessity to call this method after a +new or +open.

DomPath is the name of the domain to which to connect.

BasePath is the name of the root directory in which the subdirectory *dirPath* is to be read. If *warn* is true, a panel will be displayed if the subdirectory can't be found.

An alert panel will always be displayed if connecting to NetInfo fails.

Returns self.

property:

- **property:**(char *)*label*

Returns the property object with the given label, or nil if none is found.

See also: - **addBool:outlet:, addChar:outlet:, addInt:text:slider:zero:, addString:outlet: addString:text:button:mode:path:title:, addBrowser:browser:text:add:remove:, addBrowser:browser:add:remove:mode:path:title, addPopup:outlet:, addPopup:outlet:default:, addCall:fromNetInfo**

NS_DEV_DOCFOR:objc_method:[Text-windowChanged:];,reset

- **reset**

Reread NetInfo and redisplay all defined values. Returns self.

save

- **save**

Saves any changes back to NetInfo. If the operation is successful, returns self. If an error occurs, an appropriate error message will be displayed, the previously read data and all modifications remain unchanged and nil is returned. If save is called with no domainName and/or `name' property defined, a save panel will be shown. If the `name' property was changed and a directory with that name already exists, an attention panel with the option of overwriting or cancelling will be shown.

saveToDomain

- **saveToDomain**

Displays the NetInfo save panel and lets the user choose a new domain and, if the programmer allows, a new directory, then saves the values to NetInfo. If the operation is successful, returns self. If an error occurs, an appropriate error message will be displayed, the previously read data and all modifications remain unchanged and nil is returned. If a directory with that name already exists, an attention panel with the option of overwriting or cancelling will be shown.

scan

- **scan**

Displays all defined properties, using the previously loaded, and possibly modified NetInfo values. Returns self.

setAllowChangName

- **setAllowChangName:**(BOOL)*flag*

Enable or disable changing the name before saving (in the save panel). Returns self.

setAuthenticationUser

- **setAuthenticationUser:**(const char *)*userName*

Set the user to authenticate. If *userName* is NULL, then root will be the mandatory user. Returns self.

setSaveTitle

- **setSaveTitle:**(const char *)*title*

Set the title to be displayed in the NetInfo save panel. Returns self.

NS_DEV_DOCFOR:objc_method:[Text-setDelegate:];,setDelegate:

- **setDelegate:***anObject*

Sets the NIProperty object's delegate. In response to value changes, the NIProperty object can send the delegate any of several notification messages. See the introduction to this class specification for more information. Returns **self**.

See also: - **delegate**

Methods Implemented by the Delegate

NS_DEV_DOCFOR:objc_method:[Text-textDidChange:];,niDirOk:path::

- (BOOL)**niDirOk:**(const char *)*domain* **path:**(const char *)*directory*

Responds to a message sent to the delegate before opening a NetInfo domain and directory. If the delegate responds with NO, the open is aborted without an error and will return nil. This can be used to prevent reopening an already open domain/directory.