

NS_DEV_DOCFOR:objc_class:Text;,NIBrowserProperty

Inherits From: NIProperty : Object

Declared In: NIProperty.h

Class Description

The NIBrowserProperty class defines an object that manages the connection between a GUI item and a NetInfo entry. NIBrowserProperty objects link a NetInfo value list to a GUI element. The GUI element must be a single browser, two buttons, and an optional single textfield.

199500_browser.tiff ↪

browsertext.tiff ↪

In the first case, the add button, when pressed, will present a selection panel for selecting a file, directory, or a property from NetInfo. In the second case, the add button will attempt add the value entered in the textfield if it is not already present. The remove button will attempt to remove the *selected* entry in both cases. Both the addition and the removal can be prevented by letting the delegate react to the appropriate methods sent to it.

The GUI objects are sent the following methods:

TextField:

- **setTextDelegate**:*sender*
- **setStringValue**:(const char *)*value*
- (const char *)**stringValue**

is expected to send:

- **textDidEnd**:*textObject* **endChar**:(unsigned short)*whyEnd*
- **textDidChange**:*sender*

Browser:

- **setDelegate**:*sender*
- **setTarget**:*sender*
- **setAction**:*@selector(method:)*
- **matrixInColumn**:(int)*0*
- **selectCellAt**:(int)*i* :(int)*0*
- **reloadColumn**:(int)*0*
- **loadColumnZero**
- **selectedCell**

is expected to send:

- (int)**browser**:*sender* **fillMatrix**:*matrix* **inColumn**:(int)*column*

Button:

- **setTarget**:*sender*
- **setAction**:*@selector(method:)*

Instance Variables

```
id outlet;  
id textField;  
id addButton;  
id deleteButton;
```

outlet	The browser object.
textField	An optional text field.
addButton	The button for adding values.
deleteButton	The button for deleting values.

Method Types

Initializing the NIBrowserProperty object - init:properties:name:text:browser:mode:path:add:remove:title:

Instance Methods

```
NS_DEV_DOCFOR:objc_method:[Text-  
  initWithFrame:text:alignment:];,init:properties:name:text:browser:mode:path:add:remove:title:  
- init:sender  
  properties:(ni_proplist *)props  
  name:(const char *)label
```

text:*t* **browser:***b*
mode:(int)*m*
path:(const char *)*p*
add:*a* **remove:***r*
title:(const char *)*tString*

Initializes a new NIBrowserProperty object. This is the designated initializer for NIBrowserProperty objects: If you subclass NIProperty, your subclass's designated initializer must maintain the initializer chain by sending a message to **super** to invoke this method. See the introduction to the class specifications for more information.

The ten arguments specify the delegate (*sender*) to which the **propertyWillChange** and **propertyDidChange** methods will be sent, the property list as read from NetInfo (*props*), the string containing the name of the NetInfo entry (*label*), a text object (*t*), the browser (*b*), the mode for querying existing values (*m*), the path to an existing NetInfo value (*p*), a button for adding values (*a*), a button for removing values (*r*), and the title to be displayed in the panels for querying existing data (*tString*).

The text (*t*) is optional and may be nil. In this case the add button will cause a panel to be presented with which one can select a file, directory or arbitrary NetInfo property. Before any panel is presented, the delegate will be sent a **willSelect:what:path:title:state:** method. If the delegate returns NIPT_NONE, no panel is shown and the addition is aborted. If it returns any other value (NIPT_FILE, NIPT_DIR, or NIPT_NETINFO) the appropriate selection panel will be shown.

The mode (*m*) determines which panel is to be shown. This can be changed by the return value of the delegate's **willSelect:what:path:title:state:** method. The value must be one of NIPT_FILE, NIPT_DIR, or NIPT_NETINFO, for a panel for selecting a file, a directory, or an arbitrary value from NetInfo, respectively.

The path passed in *p* is the NetInfo directory from which a value is to be selected.

The add (*a*) and remove (*r*) buttons are optional.

The title passed in *tString* will be displayed in the selection panels.

Returns **self**.

See also: - **addBrowser:browser:text:add:remove:**, - **addBrowser:browser:add:remove:mode:path:title** (NIDirectory), **Delegates Implemented** (NIProperty)

Methods Implemented by the Delegate

NS_DEV_DOCFOR:objc_method:[Text-textWillChange:];,didSelect:what:value:domain:state:

- (int)didSelect:sender what:(int)kind value:(char **)selection domain:(const char *)niDomain state:(int)anInt

Responds to a message sent to the delegate after making a selection. The delegate receives a **willSelect:what:path:title:state:** message immediately before receiving a **didSelect:what:value:domain:state:** message unless the former returned NIPT_NONE.

Kind is the mode selected or returned by **willSelect:what:path:title:state:**.

Selection will hold a copy of the selection. The copy was allocated in [NXApp zone] and may be freed (must set *selection to NULL) or reallocated.

The value in *niDomain* is the domain from which the value was selected when selecting from NetInfo, or NULL if a file or directory was selected.

State is the value passed back from willSelect.

If the return value is greater than zero, the method **willSelect:what:path:title:state:** is resent and the selection will be presented again. If the return value is zero, the returned value in *selection* is used. If the return value is

less than zero, the selection is discarded.

If the delegate doesn't implement this method, the selection is accepted as is.

NS_DEV_DOCFOR:objc_method:[Text-textDidChange:];willSelect:what:path:title:state:

- (int)**willSelect:sender what:(int)kind path:(char **)ni_path
title:(char **)panel_title state:(int *)anInt**

Responds to a message sent before displaying any selection panel which was activated by pressing the button. The delegate's **willSelect:what:path:title:state:** method can prevent any panel from being presented by returning NIPT_NONE. If the delegate allows a selection, it presents a panel for selection a file (*kind* is NIPT_FILE), a directory (*kind* is NIPT_DIR), or a value from NetInfo (*kind* is NIPT_NETINFO). After making the selection, the delegate immediately receives a **propertyDidChange:value:** message. If the selection is aborted, no message is sent to the delegate. If the delegate doesn't implement this method, the selection panel is presented using the values *kind*, *ni_path*, and *ni_panel_title*.

Path is a directory in a NetInfo domain from which the value (a property) is to be read. It may be NULL when selecting a file or directory. *Panel_title* is the title shown for the appropriate panel.

The state passed in *anInt* is may be anything and is not used by the NIStringProperty object, but is passed as such to the **didSelect:what:value:domain:state:** method.