



CNS Menu

CNSMenu_Configure	CNSMenu_SetLayoutPartHeights
CNSMenu_DefineMenu	CNSMenu_ShowMenu
CNSMenu_DefineMenuFile	CNSMenu_Version
CNSMenu_DefineQuickMenu	CNSMenu_VersionAutoUpdate
CNSMenu_DeleteMenu	XML_CNSMenu_Tag
CNSMenu_File_GetPath	XML_deletemenuitem_Tag
CNSMenu_GetMenuChoice	XML_insertmenu_Tag
CNSMenu_GetMenuName	XML_insertmenuitem_Tag
CNSMenu_GetMenuValue	XML_menu_Tag
CNSMenu_GetMouseButton	XML_menuitem_Tag
CNSMenu_GetMouseCoordinates	XML_submenu_Tag
CNSMenu_ListMenus	XML_updatemenuitem_Tag
CNSMenu_PopupMenuErrors	
CNSMenu_Register	

CNSMenu_Configure

Description

Calling this function with no parameters will open the CNS Menu Configuration Dialog. You can optionally open to a specific tab by specifying the name of the tab as the "Option" parameter. This function also allows you to get or set any preference found in the Configuration Dialog.

Valid PrefNames:

"AddHelp" - If True, Function specific Help will be added to Calculations when inserting the Plug-in's Functions.

"PopupMenuErrors" - If True, an Error Dialog will popup upon selecting a menu item that is not setup correctly.

"AutoSetValue" - If True, any menu item that does not have a value set and does not inherit a default value will use the menu item name as the value.

"DisableMenuScaling" - If True, the menus will not scale in proportion to the current layout's zoom factor.

"UseOldFunctions" - If True, the old-style External("Scrp-XXX") functions will be available.

"UseNewFunctions" - If True, the new-style CNSMenu_XXX functions will be available.

Return Type

Varies

Format

`CNSMenu_Configure (Option ; PrefName ; PrefValue)`

Optional Parameters

Option

Specify the name of a Tab in the Configuration Dialog to show it opened to that tab.

Specify "Get" with the PrefName parameter to get a preference value. (If not found, and PrefValue is defined, PrefValue will be returned.)

Specify "Set" with the PrefName and PrefValue parameters to set a preference value.

PrefName

The Name of the Preference to Get or Set. (See the Function Description for a list of valid PrefNames.)

PrefValue

The Value of the Preference to Set. (See the Function Description for some possible values.)

Examples

Example 1

Code:

```
CNSMenu_Configure
```

Result:

Opens the CNS Menu Configuration Dialog. (Because the "Option" parameter is not used, the dialog will open to the "Basics" tab.)

Example 2

Code:

```
CNSMenu_Configure( "About" )
```

Result:

Opens the CNS Menu Configuration Dialog to the "About" tab.

Example 3

Code:

```
CNSMenu_Configure( "Get" ; "PopupMenuErrors" )
```

Result:

Returns the value of the 'Pop up Menu Errors' setting from the Configuration Dialog.

Example 4

Code:

```
CNSMenu_Configure( "Set" ; "DisableMenuScaling" ; "True" )
```

Result:

Sets the 'Disable Menu Scaling' setting in the Configuration Dialog to true.

CNSMenu_DefineMenu

Description

This is the main menu definition function. This function takes your XML menu definitions, parses them, and sets up menus to be displayed. For a complete understanding of this function, please see Appendix B. This function does not require that you use the root `<CNSMenu>` and `</CNSMenu>` tags unless you are using an internal DTD for entity definitions. If you do not use one of the root tags, you cannot use the other. In other words, both the opening `<CNSMenu>` and closing `</CNSMenu>` tags must be there, or both of the tags must not be there. You cannot have just the opening or just the closing tag.

Return Type

Text

Format

CNSMenu_DefineMenu (**XML**)

Required Parameters

XML

The XML that Defines your Menu(s).

Related Items

CNSMenu_DefineMenuFile, CNSMenu_DefineQuickMenu

Example

Code:

```
CNSMenu_DefineMenu( "<menu name='test' script='menu click'><menuitem name='a menu item' /><"
```

Result:

Creates a menu named "test" that calls the "menu click" script when an item is chosen.

CNSMenu_DefineMenuFile

Description

This function is similar to the CNSMenu_DefineMenu function, except that instead of defining the actual XML text, you specify a file on the hard drive containing the XML text. The XML file you specify must be a "well-formed" XML document. Unlike the CNSMenu_Define function, the XML file you specify must contain the root <CNSMenu> and </CNSMenu> tags.

Return Type

Text

Format

CNSMenu_DefineMenuFile (**File**)

Required Parameters

File

The Path and File Name of the XML File that Defines the Menu(s).

Related Items

CNSMenu_DefineMenu, CNSMenu_DefineQuickMenu, CNSMenu_File_GetPath

Examples

Example 1

Code:

```
CNSMenu_DefineMenuFile ( "c:\xmldocs\my_menu.xml" )
```

Result:

Specifies the xml file with a Windows path.

Example 2

Code:

```
CNSMenu_DefineMenuFile ( "/xmldocs/my_menu.xml" )
```

Result:

Specifies the xml file with a Mac OSX path.

Example 3

Code:

```
CNSMenu_DefineMenuFile( CNSMenu_File_GetPath( "Database" ) & If( Abs( Get( SystemPlatform
```

Result:

Specifies the xml file location using the CNSMenu_File_GetPath function to dynamically get the path to the database. This assumes the XML file is located in the same folder as the database file.

CNSMenu_DefineQuickMenu

Description

This function allows you to easily create a menu without using xml.

Return Type

Text

Format

CNSMenu_DefineQuickMenu (**Name** ; **Items** ; **Script** ; **DB** ; **FailScript** ; **FailDB** ; **Persistent** ; **IgnoreMetaTags** ; **BGColor** ; **FGColor**)

Required Parameters

Name

The Name for the Menu.

Items

The Return-Separated list of Items for the Menu. Each line represents one item on the menu. You can specify a value for a menu item by appending a semi-colon and the value onto the end of the item name. For example, if you had "Blue;0000FF" on a line, your menu would have an item with the name "Blue" and when a user selected that item, the value "0000FF" would be returned to your script. For sub-menus, specify the name of the sub-menu, followed by a greater-than sign (">") followed by the next sub-menu name or menu item. For example, if you had "Colors>Red" on a line, your menu would have a sub-menu named "Colors" with an item inside named "Red". Every sub-menu and menu item can be preceded with one or more special "meta tags" to specify different styles and marks for the sub-menu or item name as explained below:

! - the item will appear disabled.

\$ - the item will require the shift key to be pressed in order to show up.

^ - the item will require the control key to be pressed in order to show up.

~ - the item will require the alt/option key to be pressed in order to show up. (this is the tilde character, which is shifted version of the key to the left of the 1 on English keyboards) (only available on Mac)

@ - the item will require the command key to be pressed in order to show up. (only available on Mac)

- the item will appear bold.

% - the item will appear italicized.

_ - the item will appear underlined.

` - the item will appear with a check mark. (this is the backtick character, which is the key to the left of the 1 on English keyboards) (on Mac, you can also use opt-v to type an actual check mark)

* - the item will appear with a bullet mark. (On Mac, you can also use opt-8 to type an actual bullet mark)

+ - the item will appear with a diamond mark. (On Mac, you can also use shift-opt-v to type an actual diamond mark)

= - disable the meta tag processing for the rest of the sub-menu or item name.

You can specify more than one of the above for any item to set multiple attributes at once, with the exception of the check mark, bullet mark, and diamond mark. You can only have one mark defined for a menu item. For example, if you had "!#%Delete Record" on a line, your menu would have an item with the name "Delete Record" that appeared disabled, bold, and italicized. If you need to display a menu item that starts with one of these "meta tags", you can either set the "IgnoreMetaTags" parameter to

true, or you can use the equals ("=") meta tag to skip the meta tag processing for the item. For example, if you had "=\$4.00" on a line, your menu would have an item with the name "\$4.00" instead of a menu named "4.00" that required you to press the shift key in order to make the item visible.

Optional Parameters

Script

The Script to call when an Item is selected.

DB

The Database that contains the Script to call. (The database that calls this function is used by default.)

FailScript

The Script to call if the user clicks off the menu without selecting anything. (Default is to do nothing.)

FailDB

The DB that contains the FailScript. (Default is the same as the DB parameter.)

Persistent

The QuickMenu will be saved between FileMaker Pro sessions. (Default is false.)

IgnoreMetaTags

Disables processing of the special characters that define modifier keys, style, and marks. (Default is false.)

BGColor

Defines the background color of items on the menu. See the menuitem XML Tag definition for an explanation of how to define the color.

FGColor

Defines the foreground/font color of items on the menu. See the menuitem XML Tag definition for an explanation of how to define the color.

Related Items

CNSMenu_DefineMenu, CNSMenu_DefineMenuFile

Example

Code:

```
CNSMenu_DefineQuickMenu( "ColorsMenu" ; "Red;1¶Blue;2¶Yellow;3" )
```

Result:

Creates a menu named "ColorsMenu" with a Red item with a value of 1, Blue item with a value of 2, and a Yellow item with a value of 3.

CNSMenu_DeleteMenu

Description

This function will delete a previously defined menu. Specify the name of the menu to delete as the parameter.

Return Type

Text

Format

CNSMenu_DeleteMenu (**Name**)

Required Parameters

Name

The Name of the Menu to Delete.

CNSMenu_File_GetPath

Description

You can use this function to return the complete path to a special folder on the local hard drive.

Return Type

Text

Format

CNSMenu_File_GetPath (**Type**)

Required Parameters

Type

Specify one of the following:

"Database" - The Path to the Folder containing the Current Database.

"FileMaker" - The Path to the Folder containing the FileMaker or Runtime Application.

"Root" - The Root Volume for the System ("/" for Mac and "c:\" for Windows).

"System" - The System folder for the System ("/System/" for Mac and "c:\Windows\" for Windows).

"Desktop" - The user's Desktop Folder.

"Preferences" - The user's Preferences Folder.

"Temporary" - The Temporary Folder for storing Temporary files.

"Chewable" - Another Temporary Folder for storing Temporary files. On Mac OSX, the contents of this folder are deleted every time the computer is started. On Windows, this path is the same as when using the "Temporary" type.

"Applications" - The Folder where Applications are stored ("/Applications/" on Mac and "c:\Program Files\" on Windows).

"Documents" - The user's Documents Folder.

CNSMenu_GetMenuChoice

Description

This function will return the name or caption of the last menu item the user selected. If you have more than one menu item calling the same script in your database, you can use this function to determine which menu item was actually selected. If you specify a True value for the parameter, it will return the "full menu path" to the menu item the user selected. In other words, if you have a menu named "MyMenu", which contains a sub menu named "Colors", and the user selected the "Red" item in that sub menu, specifying True as the parameter will make this function return "MyMenu/Colors/Red" instead of just "Red".

Return Type

Text

Format

CNSMenu_GetMenuChoice ([GetFullPath](#))

Optional Parameters

GetFullPath

If True, the full menu path to the chosen item will be returned. In other words, if you have a menu named "MyMenu", which contains a sub menu named "Colors", and the user selected the "Red" item in that sub menu, specifying True as the parameter will make this function return "MyMenu/Colors/Red" instead of just "Red".

Related Items

CNSMenu_GetMenuName, CNSMenu_GetMenuValue, CNSMenu_GetMouseButton,
CNSMenu_GetMouseCoordinates

Example

Code:

```
CNSMenu_GetMenuChoice
```

Result:

Returns the name of the last menu item the user selected

CNSMenu_GetMenuName

Description

This function will return the name of the menu that contains the menu item that the user last selected. If you have more than one menu set up to call the same script in your database, you can use this function to determine which menu was actually used.

Return Type

Text

Format

CNSMenu_GetMenuName

Related Items

CNSMenu_GetMenuChoice, CNSMenu_GetMenuName, CNSMenu_GetMouseButton, CNSMenu_GetMouseCoordinates

Example

Code:

```
CNSMenu_GetMenuValue
```

Result:

Returns the name of the menu that contains the menu item the user last selected.

CNSMenu_GetMenuValue

Description

This function will return the user-defined value of the last selected menu item. When defining your menu items, you can specify any data you want, and then that menu item is selected, you can use this function in the script that is called to retrieve that data. If no value is specified for a menu item, this function will return the menu item name just as the CNSMenu_GetMenuChoice function does. FileMaker's Get (ScriptParameter) function can also be used to retrieve the menu item value.

Return Type

Text

Format

CNSMenu_GetMenuValue

Related Items

CNSMenu_GetMenuChoice, CNSMenu_GetMenuName, CNSMenu_GetMouseButton,
CNSMenu_GetMouseCoordinates

Example

Code:

```
CNSMenu_GetMenuValue
```

Result:

Returns the value that was assigned to the menu item the user last selected.

CNSMenu_GetMouseButton

Description

This function will return a number to indicate which mouse button was last clicked. If you click on a button with the left mouse button, this function will return "1". If you click with the right mouse button, it will return "2". If you click with a third or middle button, it will return "3". This allows your mouse buttons to act independently of each other, such as entering a field with the left mouse button and showing a menu with the right mouse button. You can also use the constant values "CNSMenu_LeftMouseButton", "CNSMenu_RightMouseButton", and "CNSMenu_ThirdMouseButton" to test for which button was pressed. For example, you could have the calculation "If(CNSMenu_GetMouseButton = CNSMenu_RightMouseButton ; CNSMenu_ShowMenu("RightClickMenu") ; CNSMenu_ShowMenu("LeftClickMenu"))" to show either a Left Click Menu or a Right Click Menu depending on which mouse button was pressed.

Return Type

Number

Format

CNSMenu_GetMouseButton

Related Items

CNSMenu_GetMenuChoice, CNSMenu_GetMenuName, CNSMenu_GetMenuValue,
CNSMenu_GetMouseCoordinates

Example

Code:

```
CNSMenu_GetMouseButton
```

Result:

Returns "1" if the left mouse button was clicked.

CNSMenu_GetMouseCoordinates

Description

This function returns the coordinates of the mouse. The optional parameters are "layout" and "screen". The "layout" parameter returns the coordinates of the mouse in relation to the layout, and the "screen" parameter is in relation to the whole screen. If no parameter is defined, the default setting is "layout".

Return Type

Text

Format

`CNSMenu_GetMouseCoordinates (Scope)`

Optional Parameters

Scope

Use either "layout" or "screen". The "layout" parameter returns the coordinates of the mouse in relation to the layout, and the "screen" parameter is in relation to the whole screen. If no parameter is defined, the default setting is "layout".

Related Items

CNSMenu_GetMenuChoice, CNSMenu_GetMenuName, CNSMenu_GetMenuValue,
CNSMenu_GetMouseButton

Example

Code:

```
CNSMenu_GetMouseCoordinates( "screen" )
```

Result:

Returns the coordinates of the mouse based on the whole screen.

CNSMenu_ListMenus

Description

This function returns a list of all the currently defined menus.

Return Type

Text

Format

CNSMenu_ListMenus

CNSMenu_PopupMenuErrors

Description

This function enables or disables Popup dialogs that display menu errors. If a menu item does not have a database or script associated with it, and none of its parent items do, then when the user chooses that menu item, nothing will happen. If PopupMenuErrors has been turned on, then when the user chooses that menu item, an error dialog will pop up indicating the error. This function is useful when troubleshooting your menus and determining why a script is not being called for any specific menu item.

Return Type

Text

Format

CNSMenu_PopupMenuErrors (**TrueOrFalse**)

Required Parameters

TrueOrFalse

To turn on PopupMenuErrors, use any of the following values for the parameter: 1, True, "T", "True", "Y", "Yes", or "On". To turn off PopupMenuErrors, use any of the following values for the parameter: 0, False, "F", "False", "N", "No", or "Off".

Examples

Example 1

Code:

```
CNSMenu_PopupMenuErrors( "Yes" )
```

Result:

Turns on popup dialogs to display menu errors

Example 2

Code:

```
CNSMenu_PopupMenuErrors( "Off" )
```

Result:

Turns off popup dialogs to display menu errors

CNSMenu_Register

Description

You can use this function to Register the plug-in from a script instead of through the Configuration Dialog. This is useful when the plug-in is being distributed to many computers, allowing you to install and register the plug-in without having to physically visit each computer. This function also allows you to check if the plug-in is already registered or clear the current registration. The plug-in always requires you to accept the License Agreement to use the plug-in. This is usually done by presenting the License Agreement Dialog, but that can be suppressed by using the special option value "I Accept the License Agreement".

Return Type

Text

Format

CNSMenu_Register (**FirstName** ; **LastName** ; **LicenseKey** ; **Option**)

Required Parameters

FirstName

The Registration First Name you specified when you ordered. (See your Receipt.)

LastName

The Registration Last Name you specified when you ordered. (See your Receipt.)

LicenseKey

The License Key from your Receipt.

Optional Parameters

Option

Specify "Dialog" to enter your Registration Information in a dialog.

Specify "Check" to see if the plug-in is already Registered.

Specify "Clear" to remove the Registration.

Specify "I Accept the License Agreement" to automatically accept the License Agreement dialog without showing it to the end user.

Notes: The "Dialog", "Check", and "Clear" options can also be specified as the first parameter. If you have a Developer License, do not use the "I Accept the License Agreement" value here. See your Developer Instructions file for more information.

Examples

Example 1

Code:

```
CNSMenu_Register( "My First Name" ; "My Last Name" ; "My License Key" )
```

Result:

Registers the plug-in with the provided registration information (obviously the above is not valid registration information; please see your Receipt).

Example 2

Code:

```
CNSMenu_Register( "Dialog" )
```

Result:

Displays a dialog for you to enter your First Name, Last Name, and License Key as it appears on your Receipt.

Example 3

Code:

```
CNSMenu_Register( "Check" )
```

Result:

Returns "Not Registered." or "Registered to <Name> for a <License>."

Example 4

Code:

```
CNSMenu_Register( "My Company Name" ; "My Company Name" ; "My Site License Key" ; "I Accept the License Agreement" )
```

Result:

Registers the plug-in and uses the "I Accept the License Agreement" option to keep the License Agreement dialog from appearing.

CNSMenu_SetLayoutPartHeights

Description

To properly display menus at specific pixel locations on layouts set to "List View", you need to let the plug-in know the heights of the Layout Parts of your layout. Once you have set these values, showing menus that appear to be attached to buttons in the list of records on your layout is as easy as specifying the pixel locations of the buttons on your layout in layout mode. See the CNSMenu_ShowMenu function for more information about showing menus at specific locations.

Return Type

Text

Format

CNSMenu_SetLayoutPartHeights (**Header** ; **Body** ; **Footer**)

Required Parameters

Header

The Height of the Header Part.

Body

The Height of the Body Part.

Footer

The Height of the Footer Part.

CNSMenu_ShowMenu

Description

Use this function to show a previously defined menu. You can have the menu popup at the current mouse coordinates, or you can specify where the popup menu should appear. If you want the menu to popup at the current mouse coordinates, just specify the name of the menu (ie. "My Menu"). If you want the menu to popup up at specific coordinates on your layout, you specify the values in the "Left" and "Top" parameters. For the X (vertical, column) pixel coordinate, specify it in the "Left" parameter. For the Y (horizontal, row) pixel coordinate, specify it in the "Top" parameter. An example: "My Menu"; "100"; "200". In all likelihood, if you want to specify coordinates, you will be wanting to line the menu up with some object on your layout. To know the coordinates for the menu to popup at, switch to Layout mode, change the Top and Left margins in the Layout Setup dialog to 0, select the object on the layout, and use the Object Info palette to get the pixel coordinates. The coordinates must be in pixels, so when looking at the Object Info palette, click on the measurement indicator to toggle between inches (in), centimeters (cm), and pixels (px).

Return Type

Text

Format

CNSMenu_ShowMenu (**Name** ; **Left** ; **Top** ; **Align** ; **MinWidth**)

Required Parameters

Name

The Name of the Menu to Show.

Optional Parameters

Left

The Horizontal Layout Coordinate for where the Menu should appear.

Top

The Vertical Layout Coordinate for where the Menu should appear.

Align

Use "Left" [the default], "Center", or "Right" to specify how the Menu Aligns to the Mouse or the above Coordinates.

MinWidth

Specify, in pixels, the Minimum Width the menu will always appear.

CNSMenu_Version

Description

This function returns the current version of CNS Menu. This function is useful for testing whether or not the plug-in is installed and enabled. If you call this function and a question mark ("?") is returned, then the plug-in is either not installed or not enabled.

Return Type

Text

Format

CNSMenu_Version

Related Items

CNSMenu_VersionAutoUpdate

Example

Code:

```
CNSMenu_Version
```

Result:

Returns the CNS Menu version like "CNS Menu v.1.1.10".

CNSMenu_VersionAutoUpdate

Description

This function returns an Auto Update friendly Version number of CNS Menu. The format of this version number is always exactly 8 digits long. The first two digits represent the major version of the plug-in (zero-filled). The third and fourth digits represent the minor version of the plug-in (zero-filled). The fifth and sixth digits represent the update portion of the version (zero-filled). The final two digits represent a special build number or a beta version number and will usually be zeros.

As an example for CNS Menu 1.1.10, the major version is 1, the minor version is 1, the update number is 10, and there is no special build or beta version defined. So, the resulting Auto Update friendly version number would be 01011000.

Return Type

Number

Format

`CNSMenu_VersionAutoUpdate`

Related Items

CNSMenu_Version

Example

Code:

```
CNSMenu_VersionAutoUpdate
```

Result:

Returns 01011000 for CNS Menu version 1.1.10.

