

uFileBrowser Documentation

Configuration

Default path:

The first parameter to be configured in the browser is the default path. You can leave this parameter and it will be automatically set based on the running platform.

The default path for each supported platform is:

- Windows: Drive partitions list
- Linux / OSX: /home
- Android: /storage



Selection mode:

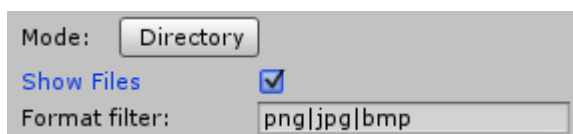
The next parameter is selection mode, you can set the browser to select a directory or a file. When in file selection mode you have the option to specify a file extension filter;

You can specify multiple extensions to be allowed by separating them with a | character. The filter is not case sensitive.



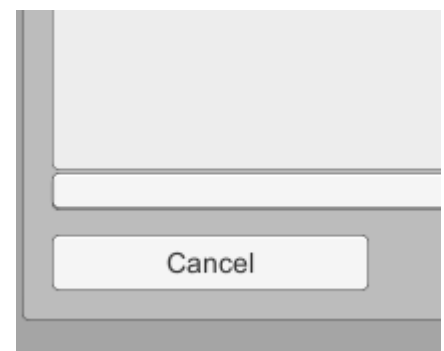
Do not include spaces or dots in the extension.

When in directory selection mode, files will not be displayed by default but you have the option to show them.



Can cancel:

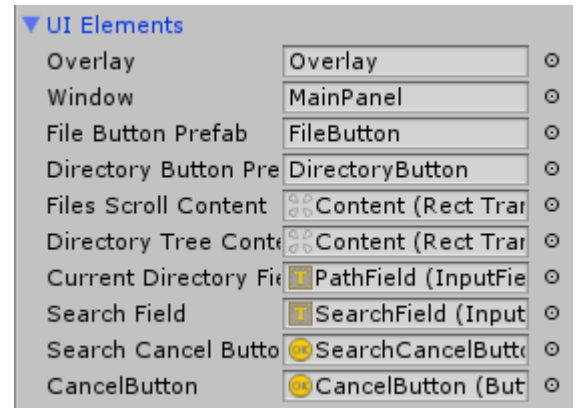
This toggle will allow the user to cancel the selection and close the browser when checked using the Cancel button, if the option is not checked then the button will be hidden.



UI Elements:

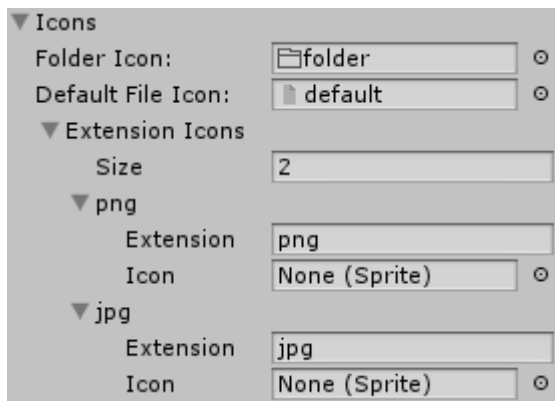
This foldout contains UI element variables that the browser needs, if you are using a prefab of the browser they are already set up.

Most elements can be disabled or removed without breaking functionality of the browser.



Icons:

In the icons section you can set up the folder icon, the default file icon and additional icons for any file extension.



As before you shouldn't put dots in the extension names.

Integrating with scripts

To show the browser from script you can call the Show method, it requires that you provide a delegate that will get called when the browser closes.

For example we want to show the browser when the user clicks a button, first we make the click function:

```
public void ShowButtonClick()  
{  
    browser.Show(new FileBrowserCallback(BrowserClosed));  
}
```

Then we create the BrowserClosed function, the same name that we specified in the delegate parameter.

```
public void BrowserClosed(string path)  
{  
    if (string.IsNullOrEmpty(path))  
    {  
        // no selection  
        return;  
    }  
    // successful selection  
    Debug.Log("You selected: " + path);  
}
```

The name of the function doesn't need to be exactly the one I used, you can use any name as long as you also update the parameter in the Show function.

The BrowserClosed function will be called when the user selects a file and presses the Select button, if the user presses the Cancel button an empty path will be passed.