Threads

Generated by Doxygen 1.9.1

1 Threads

A light weight library to dispatch actions to the Unity3D main-thread. Threads can be used at both Editor-time and Run-time straight out of the box. No initialization is required. **When `playmode` is toggled in the editor, the**

actions in the queue will be purged, and thus will never complete! The dispatcher synchronizes with Unity during the `Update loop` or `EditorApplication.Update` loop.

1.1 API

The dispatcher API can be accessed through the static `Dispatcher` class.

Method	Description
void Dispatcher.Post	Post an action on the main-thread
Task Dispatcher.PostAsync	Post an awaitable action on the main-thread
IDispatchResult Dispatcher.PostAsync<T>	Post an awaitable Func<T> on the main-thread. Results can be retrieved through the `IDispatchResult`
Dispatcher.ToMainThread()	An awaiter to synchronize to the main-tread

See the [Docs](Documentation~) for more information or Tests for how to use the Dispatcher and we have provides some `Samples`.

1.1.1 Synchronize to the main-thread

A very useful utility in this Dispatcher is the fact that code is easily synchronized to the main-thread. See the example below:

```
Task.Run(async () => {
    // We are on some other thread depending on scheduling.
    await Dispatcher.ToMainThread();
    // We are back on the main-thread now
    // so we can access UnityEngine code
    // without getting an exception!
    var myGameObject = Object.Instantiate(myPrefab);
    // F*cking Awesome XD!!!

    // We can also await a background thread
    await Dispatcher.ToBackgroundThread();
    // We are on a bachground thread again
    // so we can do expensive processing
    // without blocking the main-thread
});
```

2 CHANGELOG

3 LICENSE

4 Namespace Index

4.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

5 Hierarchical Index

5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

6 Class Index

6.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

7 Namespace Documentation

7.1 HamerSoft Namespace Reference

7.2 HamerSoft.Threads Namespace Reference

Classes

- class Dispatcher

  Main-thread dispatcher which can be used to post actions on the mainthread both Editor-time and Run-time
- interface IDispatchResult

  Result object of a function executed on the main-thread
- class MainThreadAwaiter

  An awaitable to synchronize to the main thread
- class MainThreadSync

  Object that facilitates synchronization to the main-thread

7.3 HamerSoft.Threads.Editor Namespace Reference

7.4 HamerSoft.Threads.Tests Namespace Reference

7.5 HamerSoft.Threads.Tests.Editor Namespace Reference

7.6 HamerSoft.Threads.Tests.Runtime Namespace Reference

8 Class Documentation

8.1 HamerSoft.Threads.Dispatcher Class Reference

Main-thread dispatcher which can be used to post actions on the mainthread both Editor-time and Run-time

Static Public Member Functions

- static void Post (Action action)

  Post an action on the main-thread
- static Task PostAsync (Action action)

  Post an action on the main-thread, and allow for waiting for its completion
- static Task< IDispatchResult< TResult > > PostAsync< TResult > (Func< TResult > function)

  Post a function on the main-thread, and allow for waiting for its completion
- static MainThreadSync ToMainThread ()

  Synchronize a thread to the main-thread
- static BackgroundThreadSync ToBackgroundThread ()

  Synchronize a thread to some background-thread

8.1.1 Detailed Description

Main-thread dispatcher which can be used to post actions on the mainthread both Editor-time and Run-time

8.1.2 Member Function Documentation

8.1.2.1 Post() `static void HamerSoft.Threads.Dispatcher.Post ( Action action ) [inline], [static]`

Post an action on the main-thread

Parameters

action	The function to be excecuted on the main-thread

8.1.2.2 PostAsync() `static Task HamerSoft.Threads.Dispatcher.PostAsync ( Action action ) [inline], [static]`

Post an action on the main-thread, and allow for waiting for its completion

Parameters

action	The function to be excecuted on the main-thread

Returns

An awaitable task

8.1.2.3 PostAsync< TResult >() `static Task<IDispatchResult<TResult> > HamerSoft.Threads.Dispatcher.PostAsync TResult > ( Func< TResult > function ) [inline], [static]`

Post a function on the main-thread, and allow for waiting for its completion

Parameters

function	The function to be excecuted on the main-thread

Template Parameters

TResult	Result type of the function

Returns

IDispatchResult<TResult> that contains the result of the function, or the exception thrown when it was not successfull

8.1.2.4 ToBackgroundThread() `static BackgroundThreadSync HamerSoft.Threads.Dispatcher.ToBackgroundThread ( ) [inline], [static]`

Synchronize a thread to some background-thread

This will mostly be used to sync from the main-thread to some background thread for further processing that doesn't need UnityEngine access.

Returns

an awaiter to synchronize to some background thread

8.1.2.5 ToMainThread() `static MainThreadSync HamerSoft.Threads.Dispatcher.ToMainThread ( ) [inline], [static]`

Synchronize a thread to the main-thread

Returns

an awaiter to synchronize to the main thread

The documentation for this class was generated from the following file:

- Runtime/Dispatcher.cs

8.2 HamerSoft.Threads.IDispatchResult< out out TResult > Interface Template Reference

Result object of a function executed on the main-thread

Properties

- TResult Result `[get]`
  The return value of the function
- bool Succeeded `[get]`
  Completion flag of the function
- Exception Exception `[get]`
  Optional exception throw, when unsuccessful

8.2.1 Detailed Description

Result object of a function executed on the main-thread

Template Parameters

TResult	result of the Func<T>

8.2.2 Property Documentation

8.2.2.1 Exception `Exception HamerSoft.Threads.IDispatchResult< out out TResult >.Exception [get]`

Optional exception throw, when unsuccessful

8.2.2.2 Result `TResult HamerSoft.Threads.IDispatchResult< out out TResult >.Result [get]`

The return value of the function

8.2.2.3 Succeeded `bool HamerSoft.Threads.IDispatchResult< out out TResult >.Succeeded [get]`

Completion flag of the function

The documentation for this interface was generated from the following file:

- Runtime/IDispatchResult.cs

8.3 HamerSoft.Threads.MainThreadAwaiter Class Reference

An awaitable to synchronize to the main thread

Inherits INotifyCompletion.

Public Member Functions

- void GetResult ()
  Get the result of the awaitable
- void Complete (Exception e)
  Complete the awaitable
- void OnCompleted (Action continuation)
  Function invoked when completed

Properties

- bool IsCompleted `[get]`

  Flag if the awaitable is complete

8.3.1 Detailed Description

An awaitable to synchronize to the main thread

8.3.2 Member Function Documentation

8.3.2.1 Complete() `void HamerSoft.Threads.MainThreadAwaiter.Complete ( Exception e ) [inline]`

Complete the awaitable

Parameters

e	optional exception

8.3.2.2 GetResult() `void HamerSoft.Threads.MainThreadAwaiter.GetResult ( ) [inline]`

Get the result of the awaitable

8.3.2.3 OnCompleted() `void HamerSoft.Threads.MainThreadAwaiter.OnCompleted ( Action continuation ) [inline]`

Function invoked when completed

Parameters

continuation	continuation action

8.3.3 Property Documentation

8.3.3.1 IsCompleted `bool HamerSoft.Threads.MainThreadAwaiter.IsCompleted [get]`

Flag if the awaitable is complete

The documentation for this class was generated from the following file:

- Runtime/MainThreadAwaiter.cs

8.4 HamerSoft.Threads.MainThreadSync Class Reference

Object that facilitates synchronization to the main-thread

Public Member Functions

- MainThreadAwaiter GetAwaiter ()
  Get the awaiter

8.4.1 Detailed Description

Object that facilitates synchronization to the main-thread

8.4.2 Member Function Documentation

8.4.2.1 GetAwaiter() `MainThreadAwaiter HamerSoft.Threads.MainThreadSync.GetAwaiter ( ) [inline]`

Get the awaiter

Returns

Main-thread Awaiter

The documentation for this class was generated from the following file:

- Runtime/MainThreadSync.cs

Index