Namespace WinForge

Interfaces

IModule

Interface IModule

Namespace: WinForge

Assembly: WinForge.Common.dll

```
public interface IModule
```

Properties

Name

```
string Name { get; }
```

Property Value

string

Status

```
int Status { get; }
```

Property Value

int

Version

```
string Version { get; }
```

Property Value

string

Methods

Initialize()

```
void Initialize()
```

Namespace WinForge.Base

Classes

ModuleLoader

Class ModuleLoader

Namespace: WinForge.Base

Assembly: WinForge.Base.dll

```
public static class ModuleLoader
```

Inheritance

object ← ModuleLoader

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

LoadModules(string)

```
public static List<IModule> LoadModules(string path = "./modules")
```

Parameters

`path` string

Returns

List<IModule>

Namespace WinForge.Common

Classes

Logger

Enums

Logger.LogLevel

Class Logger

Namespace: WinForge.Common

Assembly: WinForge.Common.dll

```
public static class Logger
```

Inheritance

object ← Logger

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

Debug(string)

```
public static void Debug(string message)
```

Parameters

`message` string

EnableFileLogging(string)

```
public static void EnableFileLogging(string filePath)
```

Parameters

`filePath` string

Error(string)

```
public static void Error(string message)
```

Parameters

`message` string

Info(string)

```
public static void Info(string message)
```

Parameters

`message` string

InitializeAsync()

```
public static Task InitializeAsync()
```

Returns

Task

Log(string, LogLevel, string?, bool)

Write a log message to console and to file

```
public static void Log(string message, Logger.LogLevel level = LogLevel.Info, string? tag =
null, bool includeTimestamp = true)
```

Parameters

`message` string

Log Message.

`level` Logger.LogLevel

Log Level. Default: LogLevel.Info

`tag` string

Tag of the logging module. Default: Null

`includeTimestamp` bool

Should the log line include a timestamp. Default: True

Warn(string)

```
public static void Warn(string message)
```

Parameters

`message` string

Enum Logger.LogLevel

Namespace: WinForge.Common

Assembly: WinForge.Common.dll

```
public enum Logger.LogLevel
```

Fields

`Debug = 4`

`Error = 3`

`Info = 0`

`Plugin = 1`

`Warning = 2`

Namespace WinForge.IPC

Classes

Client

HTTPManager

IPCMessage

MessageReceivedEventArgs

PipeMessenger

Class Client

Namespace: WinForge.IPC

Assembly: WinForge.Common.dll

```
public static class Client
```

Inheritance

object ← Client

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

RegisterListener(string, EventHandler<MessageReceivedEventArgs>)

```
public static void RegisterListener(string pipeName, EventHandler<MessageReceivedEventArgs>
onMessageReceived)
```

Parameters

`pipeName` string

`onMessageReceived` EventHandler<MessageReceivedEventArgs>

SendMessageAsync(IPCMessage)

```
public static Task SendMessageAsync(IPCMessage message)
```

Parameters

`message` IPCMessage

Returns

Task

Shutdown(string)

```
public static void Shutdown(string pipeName)
```

Parameters

`pipeName` string

UnregisterListener(string, EventHandler<MessageReceivedEventArgs>)

```
public static void UnregisterListener(string pipeName,
EventHandler<MessageReceivedEventArgs> handler)
```

Parameters

`pipeName` string

`handler` EventHandler<MessageReceivedEventArgs>

Class HTTPManager

Namespace: WinForge.IPC

Assembly: WinForge.Common.dll

```
public class HTTPManager : IDisposable
```

Inheritance

object ← HTTPManager

Implements

IDisposable

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

ConnectAsync(IPEndPoint, string, int)

Connect to server and send our pipe name as JSON handshake.

```
public static Task<TcpClient> ConnectAsync(IPEndPoint serverEndPoint, string clientPipeName,
int timeoutMs = 3000)
```

Parameters

`serverEndPoint` IPEndPoint

`clientPipeName` string

`timeoutMs` int

Returns

Task<TcpClient>

Dispose()

```
public void Dispose()
```

GetConnectedTCPClientPipeNames()

```
public static IReadOnlyCollection<string> GetConnectedTCPClientPipeNames()
```

Returns

IReadOnlyCollection<string>

ListenForBeaconAsync(int)

Listen once for a UDP beacon and return its IPEndPoint.

```
public static Task<IPEndPoint?> ListenForBeaconAsync(int timeoutMs = 35000)
```

Parameters

`timeoutMs` int

Returns

Task<IPEndPoint>

SendToClient(string, IPCMessage)

```
public static bool SendToClient(string pipeName, IPCMessage message)
```

Parameters

`pipeName` string

`message` IPCMessage

Returns

bool

StartServer(string)

Starts UDP beacon + TCP server in background.

```
public static void StartServer(string serverPipeName)
```

Parameters

`serverPipeName` string

WaitForBeaconAsync(int, CancellationToken)

```
public static Task<IPEndPoint?> WaitForBeaconAsync(int timeoutMs = 35000, CancellationToken
ct = default)
```

Parameters

`timeoutMs` int

`ct` CancellationToken

Returns

Task<IPEndPoint>

Class IPCMessage

Namespace: WinForge.IPC

Assembly: WinForge.Common.dll

```
public class IPCMessage
```

Inheritance

object ← IPCMessage

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

IPCMessage(string, string, string, object[]?)

```
public IPCMessage(string to, string from, string message, object[]? data = null)
```

Parameters

`to` string

`from` string

`message` string

`data` object[]

Properties

Data

```
public object[]? Data { get; set; }
```

Property Value

object[]

From

```
public string From { get; set; }
```

Property Value

string

Message

```
public string Message { get; set; }
```

Property Value

string

Timestamp

```
public DateTime Timestamp { get; set; }
```

Property Value

DateTime

To

```
public string To { get; set; }
```

Property Value

string

Class MessageReceivedEventArgs

Namespace: WinForge.IPC

Assembly: WinForge.Common.dll

```
public class MessageReceivedEventArgs : EventArgs
```

Inheritance

object ← EventArgs ← MessageReceivedEventArgs

Inherited Members

EventArgs.Empty , object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

From

```
public required string From { get; set; }
```

Property Value

string

Message

```
public required IPCMessage Message { get; set; }
```

Property Value

IPCMessage

Class PipeMessenger

Namespace: WinForge.IPC

Assembly: WinForge.Common.dll

```
public class PipeMessenger : IDisposable
```

Inheritance

object ← PipeMessenger

Implements

IDisposable

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Constructors

PipeMessenger(string)

```
public PipeMessenger(string pipeName)
```

Parameters

`pipeName` string

Methods

Dispose()

Performs application-defined tasks associated with freeing, releasing, or resetting unmanaged resources.

```
public void Dispose()
```

Events

OnMessageReceived

```
public event EventHandler<MessageReceivedEventArgs>? OnMessageReceived
```

Event Type

EventHandler <MessageReceivedEventArgs>

Namespace WinForge.Settings

Classes

Application

Persistence

Class Application

Namespace: WinForge.Settings

Assembly: WinForge.Common.dll

```
public static class Application
```

Inheritance

object ← Application

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Properties

LogFilePath

```
public static string LogFilePath { get; set; }
```

Property Value

string

MaxLogFiles

```
public static int MaxLogFiles { get; set; }
```

Property Value

int

ModuleDirectory

```
public static string ModuleDirectory { get; set; }
```

Property Value

string

Class Persistence

Namespace: WinForge.Settings

Assembly: WinForge.Common.dll

```
public static class Persistence
```

Inheritance

object ← Persistence

Inherited Members

object.Equals(object) , object.Equals(object, object) , object.GetHashCode() , object.GetType() , object.MemberwiseClone() , object.ReferenceEquals(object, object) , object.ToString()

Methods

Load()

```
public static bool Load()
```

Returns

bool

Save()

```
public static bool Save()
```

Returns

bool