

DarkRift

Unity API Reference

Introduction

The DarkRift Unity API is your game's gateway to DarkRift. It manages the connection to the server and passes data back and forth to your game.

The API is designed to be very small and very lightweight but still provide all the functionality you need to make it easy to incorporate, this reference will tell you what everything does.

Reference

DarkRiftConnection

This object represents a connection to the server instead of using the DarkRift.DarkRiftAPI static system. By default you should use DarkRift.DarkRiftAPI but if you need more than one connection you should use this.

Variables

public bool isConnected

Desc: Are we connected to a server?

Determines if the API has connected to a server or not, use Connect() to connect.

Constructors

public DarkRiftConnection()

Desc:

A blank constructor, use Connect() to connect.

public DarkRiftConnection(string ip)

Desc:

A constructor that connects to the server at IP.

public DarkRiftConnection(string ip, int port)

Desc:

A constructor that connects to the server at IP using port port.

Methods

public bool Connect(string ip)

Desc: Connect to the specified IP.

Tries to connect to the sever at IP on port 4296 (the default) returning true if sucessful or false if not.

public bool Connect(string ip, int port)

Desc: Connect to the specified IP through the specified port.

Tries to connect to the server at IP on the specified port returning true if successful or false if not.

public void Recieve()

Desc: Processes all the data received and fires the events.

This is the function that processes, decodes and distributes any messages received from the server. Usually this would be called from the Update routine or from the DarkRiftReceiver.cs script but if you need it, it's here.

public void SendMessageToServer(byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to the server.

Sends a message to the server only with tag, subject and data. This is used to talk to plugins on the server.

public void SendMessageToID(ushort targetID, byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to a specific ID.

Sends a message to the client with the specified ID with tag, subject and data.

public void SendMessageToAll(byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to all clients and the server.

Sends a message everyone (including the server) with tag, subject and data.

public void SendMessageToOthers(byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to all other clients.

Sends a message everyone but the sender (and the server) with tag, subject and data.

public void Disconnect()

Desc: Disconnect from the server.

Disconnects from the specified server, this should always be called in OnApplicationQuit() to stop the server trying to send data to it.

Delegates

public delegate void DataEvent(byte tag, ushort subject, object data);

Desc:

This is used in the onData event, it is used for basic data transmission.

public delegate void DetailedDataEvent(ushort sender, byte tag, ushort subject, object data);

Desc:

This is used in the onDataDetailed event, it's similar to the DataEvent delegate but also passes the sender's ID

public delegate void ConnectionEvent(ushort id);

Desc:

This is used in the onPlayerDisconnected event to give details about which ID disconnected. It will also be used in the onPlayerConnected event [future].

Events

public event DataEvent onData;

Desc: Occurs when data is recieved but only gives tag, subject and data.

This event is fired when data is received; it passes the tag, subject and data to the function.

public event DetailedDataEvent onDataDetailed;

Desc: Occurs when data is recieved but also passes the sender ID.

Like onData this is called when data received but also passes the ID of the sending client.

public event ConnectionEvent onPlayerDisconnected;

Desc: Occurs when a player has disconnected.

This is called when a player disconnects from the server allowing your game to remove their objects/data/etc; you will be passed their ID.

DarkRiftAPI.DarkRiftAPI

This is the static way to connect to servers, this can easily be called from any script. This should really be used by default.

Variables

public static bool isConnected

Desc: Are we connected to a server?

Determines if the API has connected to a server or not, use Connect() to connect.

Methods

public static bool Connect(string ip)

Desc: Connect to the specified IP.

Tries to connect to the sever at IP on port 4296 (the default) returning true if sucessful or false if not.

public static bool Connect(string ip, int port)

Desc: Connect to the specified I through the specified port.

Tries to connect to the sever at IP on the specified port returning true if successful or false if not.

public static void Recieve()

Desc: Processes all the data received and fires the events.

This is the function that processes, decodes and distributes any messages received from the server. Usually this would be called from the Update routine or from the DarkRiftReciever.cs script but if you need it, it's here.

public static void SendMessageToServer(byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to the server.

Sends a message to the server only with tag, subject and data. This is used to talk to plugins on the server.

public static void SendMessageToID(ushort targetID, byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to a specific ID.

Sends a message to the client with the specified ID with tag, subject and data.

public static void SendMessageToAll(byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to all clients and the server.

Sends a message everyone (including the server) with tag, subject and data.

public static void SendMessageToOthers(byte tag, ushort subject, object data)

Desc: Sends data, tag and subject to all other clients.

Sends a message everyone but the sender (and the server) with tag, subject and data.

public static void Disconnect()

Desc: Disconnect from the server.

Disconnects from the specified server, this should always be called in OnApplicationQuit() to stop the server trying to send data to it.

Delegates

public delegate void DataEvent(byte tag, ushort subject, object data);

Desc:

This is used in the onData event, it is used for basic data transmission.

public delegate void DetailedDataEvent(ushort sender, byte tag, ushort subject, object data);

Desc:

This is used in the onDataDetailed event, it's similar to the DataEvent delegate but also passes the sender's ID

public delegate void ConnectionEvent(ushort id);

Desc:

This is used in the onPlayerDisconnected event to give details about which ID disconnected. It will also be used in the onPlayerConnected event [future].

Events

public static event DataEvent onData;

Desc: Occurs when data is recieved but only gives tag, subject and data.

This event is fired when data is received; it passes the tag, subject and data to the function.

public static event DetailedDataEvent onDataDetailed;

Desc: Occurs when data is recieved but also passes the sender ID.

Like onData this is called when data received but also passes the ID of the sending client.

public static event ConnectionEvent onPlayerDisconnected;

Desc: Occurs when a player has disconnected.

This is called when a player disconnects from the server allowing your game to remove their objects/data/etc; you will be passed their ID.

Exceptions

ConnectionFailedException

The server couldn't be connected to, possibly because of a wrong IP or the server rejected us. See details in the error for more information.

NotConnectedException

You're trying to do something that requires you to be connected but you're not connected.

InvalidDataException

Data can't be sent because it was invalid. Usually because you are using 255 as a tag (it's reserved for inside use).