

# NtpClient Class Technical Documentation

## Overview

The **NtpClient** class is a simple implementation of a Network Time Protocol (NTP) client. It is designed to connect to an NTP server, request the current network time, and return it as a **DateTime** object. The class also supports proper resource management through the **IDisposable** interface.

## Namespace and Dependencies

```
using System;  
using System.Net;  
using System.Net.Sockets;  
using UnityEngine;
```

## Class Definition

### Declaration

```
public class NtpClient : IDisposable
```

### Private Fields

- **\_server**: Stores the address of the NTP server.
- **\_socket**: Represents the socket used for communication with the NTP server.
- **disposedValue**: Indicates whether the object has been disposed.

## Constructors

### Default Constructor

```
public NtpClient()
```

Initializes a new instance of the **NtpClient** class with the default NTP server address ("pool.ntp.org").

## Parameterized Constructor

```
public NtpClient(string server)
```

Initializes a new instance of the **NtpClient** class with the specified NTP server address.

## Methods

### Private Method: SwapEndianness

```
private uint SwapEndianness(ulong x)
```

Converts the endianness of a 64-bit unsigned integer.

### Public Method: GetNetworkTime

```
public DateTime GetNetworkTime()
```

Retrieves the current network time from the NTP server. Returns a **DateTime** object representing the network time.

### Exceptions:

- **InvalidOperationException:** Thrown when the network time cannot be retrieved.

### Implementation Details:

1. Creates a byte array for the NTP request.
2. Resolves the IP address of the NTP server.
3. Connects to the NTP server using the socket.
4. Sends the NTP request and receives the response.
5. Extracts and processes the time from the response.
6. Converts the NTP time to **DateTime**.
7. Closes the socket.

### Protected Method: Dispose

```
protected virtual void Dispose(bool disposing)
```

Releases the unmanaged resources used by the **NtpClient** class and optionally disposes of the managed resources.

### Public Method: Dispose

```
public void Dispose()
```

Releases all resources used by the current instance of the **NtpClient** class.

## Usage Example

```
using System;
using UnityEngine;

public class Example : MonoBehaviour
{
    void Start()
    {
        using (NtpClient ntpClient = new NtpClient())
        {
            try
            {
                DateTime networkTime = ntpClient.GetNetworkTime();
                Debug.Log("Network Time: " +
                    networkTime.ToString("yyyy-MM-dd HH:mm:ss"));
            }
            catch (InvalidOperationException ex)
            {
                Debug.LogError("Error retrieving network time: " +
                    ex.Message);
            }
        }
    }
}
```

## Remarks

- Ensure that the **Dispose** method is called to release the resources used by the **NtpClient** instance.
- The **GetNetworkTime** method sets a 3-second timeout for the socket receive operation, which can be adjusted if necessary.

## Exception Handling

The **GetNetworkTime** method throws an **InvalidOperationException** if it fails to retrieve the network time. This exception is logged using Unity's **Debug.LogException** method for debugging purposes.

## Conclusion

The **NtpClient** class provides a straightforward way to fetch the current network time from an NTP server. By implementing the **IDisposable** interface, it ensures that network resources are properly released after use. This class can be easily integrated into Unity projects for applications requiring accurate network time synchronization.