

Bit Helper

```
//GNU Library General Public License (LGPL)
//http://dac.codeplex.com/license
using System;
using System.Security;

namespace RaisingStudio.Collections.Generic
{
    internal class BitHelper
    {
        private const byte IntSize = 0x20;
        private int[] m_array;
        private unsafe int* m_arrayPtr;
        private int m_length;
        private const byte MarkedBitFlag = 1;
        private bool useStackAlloc;

        #if (PocketPC || Smartphone)
        #else
        [SecurityCritical]
        #endif
        internal unsafe BitHelper(int* bitArrayPtr, int length)
        {
            this.m_arrayPtr = bitArrayPtr;
            this.m_length = length;
            this.useStackAlloc = true;
        }

        internal BitHelper(int[] bitArray, int length)
        {
            this.m_array = bitArray;
            this.m_length = length;
        }

        [SecurityCritical]
        internal unsafe bool IsMarked(int bitPosition)
```

```

{
if (this.useStackAlloc)
{
int num = bitPosition / 0x20;
return (((num < this.m_length) && (num >= 0)) &&
((this.m_arrayPtr[num] & (((int)1) <= 0)) &&
((this.m_array[index] & (((int)1) <= 0))
{
int* numPtr1 = this.m_arrayPtr + num;
numPtr1[0] |= (((int)1) <= 0))
{
this.m_array[index] |= (((int)1) <

```