Issues in mscorlib.dll

# Dead code

* Method:
  + #17767
* Name:
  + System.Runtime.Serialization.Formatters.Binary.BinaryMethodCall.ReadArray(System.Object[],System.Object)
* Example:

internal IMethodCallMessage ReadArray(object[] callA, object handlerObject)

{

    if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsIsArray))

    {

        this.args = callA;

    }

    else

    {

        int num = 0;

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsInArray))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

            this.args = (object[]) callA[num++];

        }

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.GenericMethod))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

# Out of bounds exception

* Method:
  + #17767
* Name:
  + System.Runtime.Serialization.Formatters.Binary.BinaryMethodCall.ReadArray(System.Object[],System.Object)
* Example:

internal IMethodCallMessage ReadArray(object[] callA, object handlerObject)

{

    if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsIsArray))

    {

        this.args = callA;

    }

    else

    {

        int num = 0;

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsInArray))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

            this.args = (object[]) callA[num++];

        }

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.GenericMethod))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

# Dead code

* Method:
  + #17773
* Name:
  + System.Runtime.Serialization.Formatters.Binary.BinaryMethodReturn.ReadArray(System.Object[],System.Object)
* Example:

internal IMethodCallMessage ReadArray(object[] callA, object handlerObject)

{

    if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsIsArray))

    {

        this.args = callA;

    }

    else

    {

        int num = 0;

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsInArray))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

            this.args = (object[]) callA[num++];

        }

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.GenericMethod))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

# Out of bounds exception

* Method:
  + #17773
* Name:
  + System.Runtime.Serialization.Formatters.Binary.BinaryMethodReturn.ReadArray(System.Object[],System.Object)
* Example:

internal IMethodCallMessage ReadArray(object[] callA, object handlerObject)

{

    if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsIsArray))

    {

        this.args = callA;

    }

    else

    {

        int num = 0;

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.ArgsInArray))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

            this.args = (object[]) callA[num++];

        }

        if (IOUtil.FlagTest(this.messageEnum, MessageEnum.GenericMethod))

        {

            if (callA.Length < num)

            {

                throw new SerializationException(string.Format(CultureInfo.CurrentCulture, Environment.GetResourceString("Serialization\_Method"), new object[0]));

            }

# Dead code

* Method:
  + #18847
* Name:
  + System.Reflection.Emit.ModuleBuilder.GetTypes
* Example: Filtered is never modified

public override Type[] GetTypes()

{

int size = m\_TypeBuilderList.Count;

Type[] moduleTypes = new Type[size];

TypeBuilder tmpTypeBldr;

int filtered = 0;

for (int i = 0; i < size; i++)

{

EnumBuilder enumBldr = m\_TypeBuilderList[i] as EnumBuilder;

tmpTypeBldr = m\_TypeBuilderList[i] as TypeBuilder;

if (enumBldr != null)

tmpTypeBldr = enumBldr.m\_typeBuilder;

if (tmpTypeBldr != null)

{

if (tmpTypeBldr.m\_hasBeenCreated)

moduleTypes[i] = tmpTypeBldr.UnderlyingSystemType;

else

moduleTypes[i] = tmpTypeBldr;

}

else

{

// RuntimeType case: This will happen in TlbImp.

moduleTypes[i] = m\_TypeBuilderList[i];

}

}

if (filtered > 0)

{

Type[] filteredTypes = new Type[size - filtered];

int src, dst;

for (src = 0, dst = 0; src < size; src++)

{

if (moduleTypes[src] != null)

{

filteredTypes[dst] = moduleTypes[src];

dst++;

}

}

moduleTypes = filteredTypes;

}

return moduleTypes;

}