

Re: C# Generics: Why int[] doesn't implement IList<int>?

Source:

<http://www.tech-archive.net/Archive/DotNet/microsoft.public.dotnet.languages.csharp/2004-05/0492.html>

From: Nicholas Paldino [.NET/C# MVP] (mvp_at_spam.guard.caspershouse.com)

Date: 05/03/04

Date: Mon, 3 May 2004 12:10:27 -0400

Kevin,

This is true when you declare any array. Your array type is T[]. The base of T[] is System.Array.

--

```
    - Nicholas Paldino [.NET/C# MVP]
    - mvp@spam.guard.caspershouse.com
"Kevin P. Fleming" <kpfleming@backtobasicsmgmt.com> wrote in message
news:uUlh1WSMEHA.556@tk2msftngp13.phx.gbl...
> Nicholas Paldino [.NET/C# MVP] wrote:
>
> > You misunderstood the implication that I was making. Basically,
because
> > System.Array (the general base for all arrays) implements IList, and
> > IEnumerable, the specialized arrays (int[], string[], etc, etc) should
also
> > implement the specialized versions (IList<int> and IEnumerable<int>) as
> > well.
> >
> > Also, array1 and array2 are not of type System.Array. Rather, they
> > derive from System.Array. Because of this, it is easy for the
derivation to
> > implement IList<T>.
>
> OK, I did not realize that array1 and array2 would be instances of a
> type derived from System.Array. Is this true whenever I declare any
> array? I was not aware that the C# compiler created hidden types to be
> used to implement arrays. I guess it must, otherwise array1[0] would be
> a System.Object, not a System.Int32.
>
> Given this information, implementing the IList and IEnumerator generics
> would be relatively easy and quite useful.
>
```