**string是value type還是reference type?**

**前言:**

最近在複習記憶體管理Stack與Heap發現了一個有趣的現象，MSDN在文件中記載，表中除了 object 和 string 以外的所有型別，都視為傳值型別(https://docs.microsoft.com/zh-tw/dotnet/articles/csharp/language-  
reference/keywords/built-in-types-table)，令我有點意外，因為一直以來，我都是把string當作是value type在使用好像也沒發生甚麼事…，這次就來探究究竟是文件記載錯誤還是string真的是參考型別。

**GetType().IsValueType判斷是否是value type:**

**string** StringTest = "GARY";

Console.Write(StringTest.GetType().IsValueType);

//結果為True，證明String型別為reference type

**以Object來測試reference type的存取方式:**

Person P1 = **new** Person();

P1.name = "Gary";

P1.age = 24;

//將P2指向P1的位址

Person P2 = P1;

//改變P2的物件資料

P2.name = "Andy";

P2.age = 26;

//打印出P1執行至此的資料

Console.WriteLine("我是" + P1.name + "今年" + P1.age + "歲");

Console.Read();

//執行結果為:我是Andy今年26歲

//因P2及P1使用的是同一位址，所以當改變P2物件的值，P1物件的質也會跟著改變

//不出所料，這就是典型參考型別(Reference type)資料存取的結果

**以String來測試reference type的存取方式:**

**string** String1 = "Gary";

//將String2指向String1的位址

**string** String2 = String1;

//重新指定String2的值

String2 = "Andy";

Console.WriteLine(String2);

Console.Read();

//執行結果為:Gary

//原以為String2及String1使用的為同一位址，

//所以當改變String2物件的值，String1物件的質也會跟著改變。

//但結果和我們預想的不一樣，這樣string的存取方式似乎別叫接近傳值型別(value type)

**Google找答案!**

在StackOverflow馬上就找到了這問題，似乎有很多人跟我依樣有這個疑惑啊!以下擷取其中一個我覺得比較完整回答。

Strings aren't value types since they can be huge, and need to be stored on the heap. Value types are (in all implementations of the CLR as of yet) stored on the stack. Stack allocating strings would break all sorts of things: the stack is only 1MB, you'd have to box each string, incurring a copy penalty, you couldn't intern strings, and memory usage would balloon, etc...  
(出處: http://stackoverflow.com/questions/636932/in-c-why-is-string-a-  
reference-type-that-behaves-like-a-value-type)

**結語:**

因為String的容量通常比較大，需要被存在記憶體Heap中，而通常Value type是被存在Stack中。若直接將string當成value type存在stack會有問題(因為Stack的容量只有1MB)，但又為了讓我們操作方便，所以才將string的存取方式設計成類似value type的存取方式，算是一種比較特例的設計。

**參考資料:**

<http://blog.sanc.idv.tw/2012/01/c-equals.html>  
<http://stackoverflow.com/questions/636932/in-c-why-is-string-a-reference-type-that-behaves-like-a-value-type>  
<https://docs.microsoft.com/zh-tw/dotnet/articles/csharp/language-reference/keywords/built-in-types-table>