

# How do I make a clone of a Windows Runtime vector in C++/WinRT?

[devblogs.microsoft.com/oldnewthing/20191122-00](https://devblogs.microsoft.com/oldnewthing/20191122-00)

November 22, 2019



Raymond Chen

Suppose you have a Windows Runtime vector, either an `IVector<T>` or an `IVectorView<T>`, and you want to clone it so that you can operate on the clone without affecting the original. For example, maybe you have an `IVectorView<T>`, which is read-only, and you want to clone it so you can make changes.

The naïve way would be to copy the vector:

```
// Code in italics is wrong.  
IVector<Thing> original = GetTheThings();  
IVector<Thing> clone = IVector<Thing>{ original };
```

This doesn't work because `IVector<T>` and `IVectorView<T>` are interfaces, and copying an interface merely copies a reference to the same underlying object.

To get a brand new object, you need to create a brand new object.

```
IVector<Thing> original = GetTheThings();  
std::vector<Thing> temp{ original.Size() };  
original.GetMany(0, temp);  
IVector<Thing> clone = single_threaded_vector(std::move(temp));
```

First, we create a temporary vector into which we will copy the contents of the original vector.

Next, we use the `GetMany` method to read the entire contents of the original `IVector` into our temporary vector. Note that if the vector's size can change asynchronously, then there's a race condition if the size changes between the time we create the temporary vector and the time we fill it with goodies. Fixing that is left as an exercise.

Finally, we create a brand new `IVector` from our temporary vector.

Raymond Chen

**Follow**



