Setting the contents of a Windows Runtime Vector from C++/WinRT in one call

devblogs.microsoft.com/oldnewthing/20240524-00

May 24, 2024

Raymond Chen

We saw earlier that you can build the initial contents of a Windows Runtime `IVector` in a `std::vector` (which is usually far more convenient with better performance), and then convert it to a Windows Runtime `IVector` as a final step. Or you can create the Windows Runtime `IVector` from raw materials without having to use an explicit `std::vector` at all.

But what if somebody gave you an existing Windows Runtime `IVector`, and you want to overwrite its current contents with new content? This happens in many parts of the Windows Runtime, such as the `FileOpenPicker`, which gives you a `FileTypeFilter` that you can fill with the file types you want to filter for. You can't provide your own `IVector`; you have to fill the existing one.

The naïve way would be to clear the vector and then fill it with items, one at a time:

```
namespace winrt
{
    using namespace winrt::Windows::Storage::Pickers;
}

winrt::FileOpenPicker CreatePickerForSupportedImages()
{
    winrt::FileOpenPicker picker;
    auto filter = picker.FileTypeFilter();
    filter.Clear();
    filter.Append(L".jpg");
    filter.Append(L".png");
    filter.Append(L".bmp");
    filter.Append(L".gif");
    filter.Append(L".tif");
    return picker;
}
```

But there's a one-stop way of doing this: The `ReplaceAll` method.

```
winrt::FileOpenPicker CreatePickerForSupportedImages()
{
    winrt::FileOpenPicker picker;
    auto filter = picker.FileTypeFilter();
    filter.ReplaceAll({ L".jpg", L".png", L".bmp", L".gif", L".tif" });
    return picker;
}
```

The `ReplaceAll` method replaces the entire contents of the vector with the values you provide. You can think of it as a combination `Clear` and `Append`, but it all happens at once.

The `ReplaceAll` method takes a `winrt::array_view`, so you can pass anything that an `array_view` can construct from. In this example, we used an `initializer_list`, but you can look at the other constructors to see all the options you have available.