

# The COM static store, part 4: Aggregating into a single object

---

 [devblogs.microsoft.com/oldnewthing/20210211-00](https://devblogs.microsoft.com/oldnewthing/20210211-00)

February 11, 2021



Raymond Chen

So far, we've been looking at how you can use the COM static store to hold a singleton which will be run down when COM uninitializes. But what if you have a lot of things you want to save? I mean, you could put each one individually in the COM static store, but that gets quite cumbersome. And some of the things might not even be COM objects, or they may be COM objects that do not support `IInspectable`. What can you do about those?

You can wrap all of your global state into a single object that supports `IInspectable`, and put that single object into the COM static store. That way, you can just grab one thing out of the COM static store instead of having to go back to the COM static store for each one.

```

// C++/WinRT
struct SharedState :
    winrt::implements<SharedState,
                    winrt::Windows::Foundation::IInspectable>
{
    int some_value = 0;
    winrt::com_ptr<IStream> stream;
    std::vector<winrt::com_ptr<IStorage>> storages;
};

winrt::com_ptr<SharedState>
GetSingletonSharedState()
{
    auto props = CoreApplication::Properties();
    if (auto found = props.TryLookup(L"SharedState")) {
        return as_self<SharedState>(found);
    }
    auto value = winrt::make_self<SharedState>();
    static winrt::slim_mutex lock;
    winrt::slim_lock_guard const guard{ lock };
    if (auto found = props.TryLookup(L"SharedState")) {
        return as_self<SharedState>(found);
    }
    props.Insert(name, *value);
    return value;
}

```

Now, when you need stuff, you can ask for the singleton `SharedState` and access everything, rather than having to get each one individually.

But wait, we can do even better still.

Next time.

Raymond Chen

**Follow**

