Service Serialization

Service serialization is the serialization of a Tapestry IOC service.

Every once in a while you may need to serialize a service. For example, you may store an object into the HttpSession that holds a reference to a service. In a clustered environment, that object will be serialized and broadcast to other servers in the cluster.

Services in Tapestry are serializable. Specifically, service *proxies* are serializable. However, your service implementations *do not* have to be serializable.

Serialization works as follows:

- When a proxy is serialized, it instead serializes a *token* object.
- The token object is what's stored in the output stream.
- When the token is de-serialized, it locates the service proxy in the current Registry and returns that.

The end result is very efficient: just the tiny tokens are serialized, not the services with their proxies, configurations, implementations, dependencies, internal state and so forth.

Again, note that the actual service implementation is not serialized. Due to Tapestry's lazy creation policy, the service implementation may not even exist. Since outside code only sees the proxy, there's no difference.

Registry Resolution

The one trick here is locating the service proxy. Tapestry uses a *weak reference* to the Registry to do this. When a Registry starts up, it is stored in the reference, so that de-serialization can work.

The reference is cleared when you shut down the Registry. If you stop using the Registry, but fail to shut it down, the weak reference ensures that it will be released to the garbage collector anyway. Still, you should shutdown a Registry when done with it.

This all makes one BIG assumption: that there's just one Registry. That's normal for a web application, especially when the tapestry-ioc JAR is included as part of the web application's WAR.

If you are running multiple Registries you will likely see errors in your console:

```
[ERROR] SerializationSupport Setting a new service proxy provider when there's already
an existing provider. This may indicate that you have multiple IoC Registries.
```