ServicePreloadProposal

Problem Description

(April 16 2004, before 1.0-alpha-4)

Much of the HiveMind architecture is predicated upon the idea that no resource should be created until it is needed. Thus service proxies may be created early, but the underlying service implementations (including interceptors) are only created when a method of the service interface is first invoked.

This is not always appropriate:

- During development, it is desirous to create services early. This ensures that construction directives (in the module deployment descriptor) are valid, that service initialization code is valid. It is also helpful for services to be in a fully instantiated state prior to debugging.
- Event driven services should be created early, so that they can register for event notifications. Event notification is an aspect of the implementation, and event notification registration occurs when the service is created, so an event-driven service implementation would be unaware of any events that occured prior to its construction (on first invocation of a service method).

Attribute Solution

In this solution, the <service-point> element is extended to include a new attribute, `create`, which would accept the following values:

- invoke (default) - create the service implementation when a service method is first invoked
- reference - create the service implementation when the service is first referenced
- startup - create the service implementation when the registry is first constructed
- dev-startup - at startup in development mode, otherwise as invoke

Development mode could be defined by one of:

- A system property, perhaps `org.apache.hivemind.development-mode`
- A substitution symbol
- A constructor parameter for RegistryBuilder
- A method of RegistryBuilder or Registry (that is only invoked if the application is in development mode)

There are some conflicts between the `create` attribute and the service model for the implementation. Today, the primitive service model uses the equivalent of `reference`, and the other models use the equivalent of `invoke`.

For services with the threaded or pooled service model, it is (or should it be) necessary that ThreadEventNotifier be invoked after early bird services are created?

Is the *order* in which services are started relevant? I would hope note.

Configuration Point Solution

In this solution, a new configuration point is defined. Let's call it `hivemind.StartupServices`. Contributions look like `<create-service id="...">`.

After the registry is constructed, this extension point is used to force the creation of the services.

This solution works best if the ConditionalContributionsProposal is implemented, and applies to <contribution> elements as well as <implementation> elements.

Again, there's the question of whether order is important.

Discussion

The attribute solution is less verbose but I still find it troubling; I hate all ambiguities and this discord between create and the service model is troubling.

The configuration point solution is more typing, but (if ConditionalContributionsProposal is implemented) seems less ambigous.

Implementation

In 1.0-alpha-5, as configuration hivemind.EagerLoad