

CentralizedErrorReportingProposal

Problem Description

[HowardLewisShip](#), Jun 5 2004, 1.0-alpha-5

Some discussion has been going around about how HiveMind reports errors (errors when parsing deployment descriptors, and errors at runtime).

I prefer a *lenient* system, where early errors are reported and then the system continues on as best it can (with the expectation that early errors will blossom into later failures).

Others prefer a *strict* system, where early errors result in thrown exceptions and, potentially, no Registry or a shutdown of the application entirely.

Tapestry has traditionally used a strict system and, to date, HiveMind has used a lenient system.

This proposal is to make this *strategy* pluggable, so that it can be controlled by individual developer's taste and situation. It may also help Geoff (or whoever) writes an Eclipse plugin.

Proposed Solution

Part of the solution is already in place: `org.apache.hivemind.impl.ErrorHandler` is used by the module deployment descriptor parser; the default implementation is lenient but a strict implementation can be provided.

This interface should be moved to package `org.apache.hivemind` and should be extended:

```
public interface ErrorHandler
{
    public void error(Log log, String message, Location location, Throwable cause);
}
```

The default behavior would be to use the `Log` instance to log the error, as:

```
Error at <location>: <message>
```

or

```
Error: <message>
```

Depending on whether `location` is null or not. The exception would be provided (if not null).

Obtaining the [ErrorHandler](#)

The `Module` and `RegistryInternal` interfaces will be extended to provide access to the `ErrorHandler`. Possibly, they will implement the `ErrorHandler` interface and delegate out to a user-provided instance.

The [BuilderFactory](#) can now set a property, by default `errorHandler`, to the `Module`'s `ErrorHandler` instance (which, in fact, will be the global `ErrorHandler`).

Providing the [ErrorHandler](#)

The `ErrorHandler` will be provided to `RegistryBuilder`, which will use it when parsing module descriptors and provide it to the `Registry` as it is constructed.

Changes to error messages

Because the location will be uniformly displayed, it will not be necessary to include the location in error messages, which (I think) will make many of the existing error messages more readable. It will also make the messages more palatable to, for example, an Eclipse plugin (which will have other means of identifying the location of the error, rather than emedding that location as text within the message). Locations will be part of error messages only when discussion *two* locations, such as when an interceptor is contributed more than once to a service point.

Discussion

This is currently implemented in CVS and should debut formally whenever we release 1.0-beta-1. There's probably a few error messages that still explicitly include the location (and should be changed to let the `ErrorHandler` report the error).

The `DefaultErrorHandler` is lenient. We should probably provide an implementation of a strict `ErrorHandler`.