

HiveMind2 Backward Compatibility

Backward Compatibility

How far an application is backward compatible depends on the way it uses hivemind. For example a web application that uses the [HiveMindFilter](#) for loading the registry and defines service and configuration points without any complex schemas or factories has a good chance to be 100% compatible.

Things that have changed

The registry build and construction process

The [RegistryBuilder](#) and related classes (`ModuleDescriptorProvider`) are not used in the same way as before. See [NewAndNoteworthyFeaturesInAnnotationBranch](#) for examples how the registry is build.

Unique attributes and key-attributes

Configurations no longer limited to lists

These interfaces have changed. Items that were of type `java.util.List` before are now of type `Object`:

- [ServiceInterceptorFactory](#)
`public void createInterceptor(InterceptorStack stack, Module invokingModule, Object parameters);`