

# RoadMap

Here's the first list of objectives (aka roadmap) identified on the developer mailing list:

- We would like to set on a common implementation as a matter of economy of scale.
- There is a desire to remove the current PXE dependency on Hibernate, which is licensed under the LGPL. The abstraction layer developed for BPE could be useful for a common Ode implementation. The BPE implementation relies on J2EE CMP for stateful persistence. Stateless usage scenarios do not require persistence. EJB3 might be a suitable candidate for state persistence where needed.
- A data abstraction layer developed for the BPE could be useful for isolating the BPEL engine from a specific content model. This probably has implications to the entry point API mentioned above. There is an existing ODE thread of discussion on the topic.
- Two deployment models will be supported, one through pre-compilation at design time (model currently supported by PXE), and the other through pre-compilation at deployment time (model currently supported by BPE). The prior scenario is useful for tools that rely on early error detection specific to BPEL and the latter is useful when BPEL is not the originating language.
- The common Ode implementation should support BPEL 2.0 within the constraints of the WS-BPEL 2.0 specification timeframe.
- We will support optimizations for stateless business processes within the implementation
- We desire to supply run-time engine debugging support that is capable of referring back to the originating BPEL markup for purposes of tooling support
- The management interfaces represented in the PXE implementation are compelling
- The Jacob engine is interesting in itself because it appears to be analogous to the focused approach taken in BPE of walking the object model. We need to examine this further and consider the thread and container isolation issues.
- We need to determine the most capable and useful entry point API for any common Ode implementation. It's probably productive to avoid introducing any external dependencies to any bus architecture or external protocols so that the core engine can be used by as many external projects as possible while avoiding unnecessary dependencies. This implies that the proprietary JBI-like bus currently used by the PXE contribution would need to be abandoned.
- The implementation should support both a transactionally contiguous invoke-to-endpoint transaction model as well as an asynchronous coupling (callback-enable implementation) to the endpoint from the engine (the latter requirement is specific to current PXE usage scenarios). The transactional requirements imply some isolation of the thread semantics, currently present in the PXE contribution, would be necessary.
- Testing
  - a) Unit test suites can be contributed for BPEL 1.1 and BPEL 2.0 initially based on the PXE and BPE contributions and the final set will be derived from the intersection of the two test suite contributions.
  - b) A benchmarking framework should be created to assess typical messaging scenarios and stress tests