

# AvalonAssemblyStatusReportOne

h2. Status Report [AvalonAssembly] 07-DEC-02 Over the past couple of months work has focussed on the assembly, meta and Merlin packages as part of the transition out of Excalibur and into the avalon-sandbox project. The transition involved a number of changes that are listed below: \#refactoring of the classloading mechanisms (following Berin's suggestions from way back) taking into consideration the discussions concerning a profile driven container \#incorporation of support for the concepts and direction established in the "context discussions" including the ability to customize your contextualization interface (Leo's process) \#clean separation of component deployment (under Appliance) as distinct from development templates (under Profile) (Gary's comments on separation of concerns) \#addition of a appliance customization layer that makes is drop-dead simple to set up an component management engine and launch customized components(Steve) \#addition of test cases \#elimination of the necessity to explicitly declare components in jar file (files are now auto scanned) (Steve) \#improvements to the logging system (Vinay Chandran patches) \#addition of proxy support for \{ \{ [ComponentManager] and [ComponentSelector] \} \} (content from Berin in framework) Structurally speaking - the excalibur/assembly (Merlin 2.0) package has been split into two the distinct packages - avalon-sandbox/assembly and avalon-sandbox/merlin (refer [AvalonMerlin]). The updated sandbox assembly package is independent of a containment strategy - it basically handles all of the stuff to do with component deployment. Operationally speaking there are some changes that need to be considered. Aside from documentation updating, there are still a couple of more things to be done before Assembly 1.0 stabilizes - namely some more work on the context area to integrate an abstract location implementation model and handling of alias declarations, and secondly - based on the success of the context implementation approach, I would like to see the assembly package move closer to the deployment state model described on [AvalonFiveLifecycleProposal]. To go from where we are now to that model is not a big jump in terms of the code - but I thing it will be a massive jump in terms of overall flexibility in that we will be able to support arbitrary component models. [StephenMcConnell]