WishList

The purpose for this page is to capture (**in brief**) wish list ideas for future releases of the Apache Beehive project. It is **not intended for bugs or minor enhancements to existing features**, which have more appropriate venues (JIRA), but new things that would improve Beehive.

NOTE: This page is for recording ideas. The *discussion* of these ideas should take place on the beehive-user mailing list.

Please explain the idea and benefit **in brief** below. Longer descriptions can live on their own wiki pages, with a link from the short description below.

Contents

Integration with J2EE App Server(s)

Supporting Tomcat alone is too limiting, as it is only a Servlet Container. Many capabilities can't be effectively demonstrated or tested, since they may need JMS, or EJBs, or ... Geronimo is a good target candidate since it also lives in Apache, but the build/configuration model should be flexible enough to support others, since a key goal of Beehive is to enable app portability across app servers.

Business Logic Abstraction

A hole in the current Beehive app story is the ability to write some basic business logic and then add transaction or security semantics to it. This doesn't imply that Beehive needs a business logic container; the right answer might be to write the logic as a control (or annotated POJO) and provide a runtime mapping so that it runs inside of an EJB, Spring Bean, or other existing container that can provide the desired semantics.

Better Support for Building Tests

Beehive shouldn't just help a Beehive developers author a new abstraction such a pageflow, web service, or control, it should also make it easy to bootstrap testing of the abstractions. We should look at both in-container and standalone (unit/TDD) testing.

Flexible Package Configuration

A JSP page should be able to specify the package that it is in. For example, a JSP page may be www.somecompany.com/coolapp/somepage.jsp and it seems reasonable that a developer would like to use com.somecompany.coolapp for the Controller package.

Dynamic Updates of Data

BEEHIVE-276 was the start of this feature request. Trees use the XML HTTP Request in some cases to get some details from the server, but it is only after a user action. I would like to see an option on tree and data grid tags to poll their state on the server so they may be dynamically updated. The polling should be async and it would be nice to have a configurable timer value.

Support a Single Source Tree for WSM Compile

To compile a Web service that uses custom types (i.e. Service.jws uses Address in wsm-addressbook-enhanced) you have to put the Web service and types in separate source trees. I like that as an option, but it appears to be required. More concerning than types is that an endpoint interface currently has to be in a separate source tree for the implementation to compile. It would be nice to allow any or all of the Web service, its interface, and its types to share a single source tree. I added an attachment call SrcTreeWish.tar to demonstrate the source tree models that I'd like to see; README.txt included in the bundle.

Support for JSTL

When working with WebLogic Workshop 8.1 and the netui tags, I'm using JSTL because they offer a lot of useful tags and they are - well standard. But in the pre-beehive setting I had to use the <netui-data:getData> tag to retrieve pageflow scope based values and then use {pageScope value} in the EL. What I would like is to use JSTL and still be able to access pageflow scoped values - maybe using {pageflowScope value} in the EL or some other solution. I would like the new Eclipse based workshop to be able to use JSTL with the same tool support as the netui tags.

Support for Reusable Dialog Components

Jakarta taglibs just annonced that Reusable Dialog Components (RDC) Tag Library are released as supported tag library. Maybe beehive should provide tight integration with these very useful tag libraries?

Support for Workflow Engines

A facility for mapping a workflow of an application to its forms, reports and entries. By using this facility my PageFlowController is my business process description.

Support for JSR 168

Provide the ability to create a "beehive portlet" that can be deployed in a Portal container that supports JSR 168

Struts 2.0/WebWork based

The next version should take into account that Struts 2.0 is supposed to based on WebWork. Beehive should also take advantage of this so we don't have to use FormBeans and the required inheritance from Action classes. Beehive 2.0 should not require that PageFlowControllers must inherit from any base class, but focus more on conventions instead of inheritance.

Easy Eclipse IDE integration

Provide an easily installable and usable "out of the box" plugin for development of beehive applications in Eclipse.