



Admin Console GlassFish V3

<https://glassfish.dev.java.net>

Ken Paulsen
Anissa Lam

June 18, 2008
Sun Microsystems, Inc.

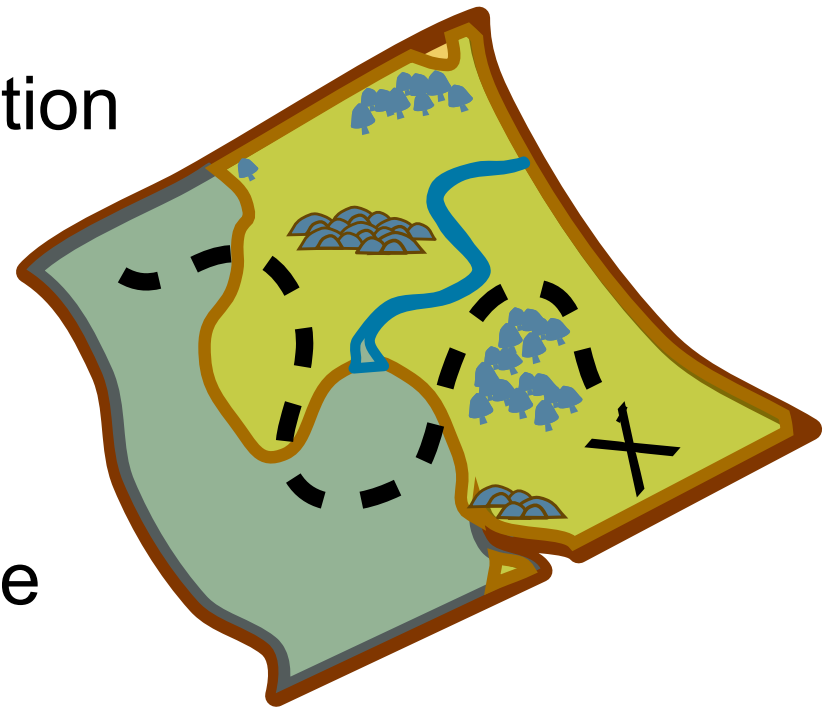


Agenda

- **v3 Admin Console Strategy**
- Console Plugins

v3 Admin Console Strategy

- Infrastructure
 - > Plugin API
 - > Update Center Integration
 - > Theme Support
 - > Preferences Support
- Features
 - > Scripting Support
 - > Enhanced Web Service
 - > Charting
 - > etc.



Infrastructure: Plugin API

- Plugin
 - > Tree Nodes (in TP2)
 - > Tabs
 - > Pages
 - > Themes
 - > Help
- Issues:
 - > ClassLoader Issues. Only resources can be in the plugin module

Branding

- Provides at least
 - > GlassFish Community Server
 - > Sun GlassFish Enterprise Server
- Questions:
 - > should we do this short term effort ?
 - > Can be achieved with Theme Plugin.

Preferences Support

- Build the infrastructure to support user preference.
- Allows user to customize their page, eg common task page, inline text etc.
- Providing the actual implementation is NTH for V3 Lite.
- May use Preference API in JDK6
- **Question:** whats the min. requirement of JDK version in V3 Lite ?

Update Center Integration

- Provide ISP packages for both Admin Console War and Plugin Modules
- Vehicle for initial download and updates
- New page showing installed modules and available updates
- Include other modules besides GlassFish modules if UC 2.0 provides necessary API

Update Center Integration (cont..)

Issues/Dependencies:

- specifying range of dependencies
- UC API
- UC to support downloading additional modules based on dependencies. eg. a new web plugin will also bring in the corresponding core module.

Monitoring

- Simple monitoring page
- May provide charting

Dependencies

- backend monitoring support
- JMaki charting
- AMX API

Scripting Support

- Deployment screen
- Listing of deployed app

Dependencies

- AMX API to list out apps

Fix & Finish

- > **Navigation**
 - > **Dynamic Sub Nodes**
 - > **Tree Node refresh**
 - > **Breadcrumbs**
- > **Core**
 - > **Logger Settings**
 - > **Log Levels Settings**

Fix & Finish (cont..)

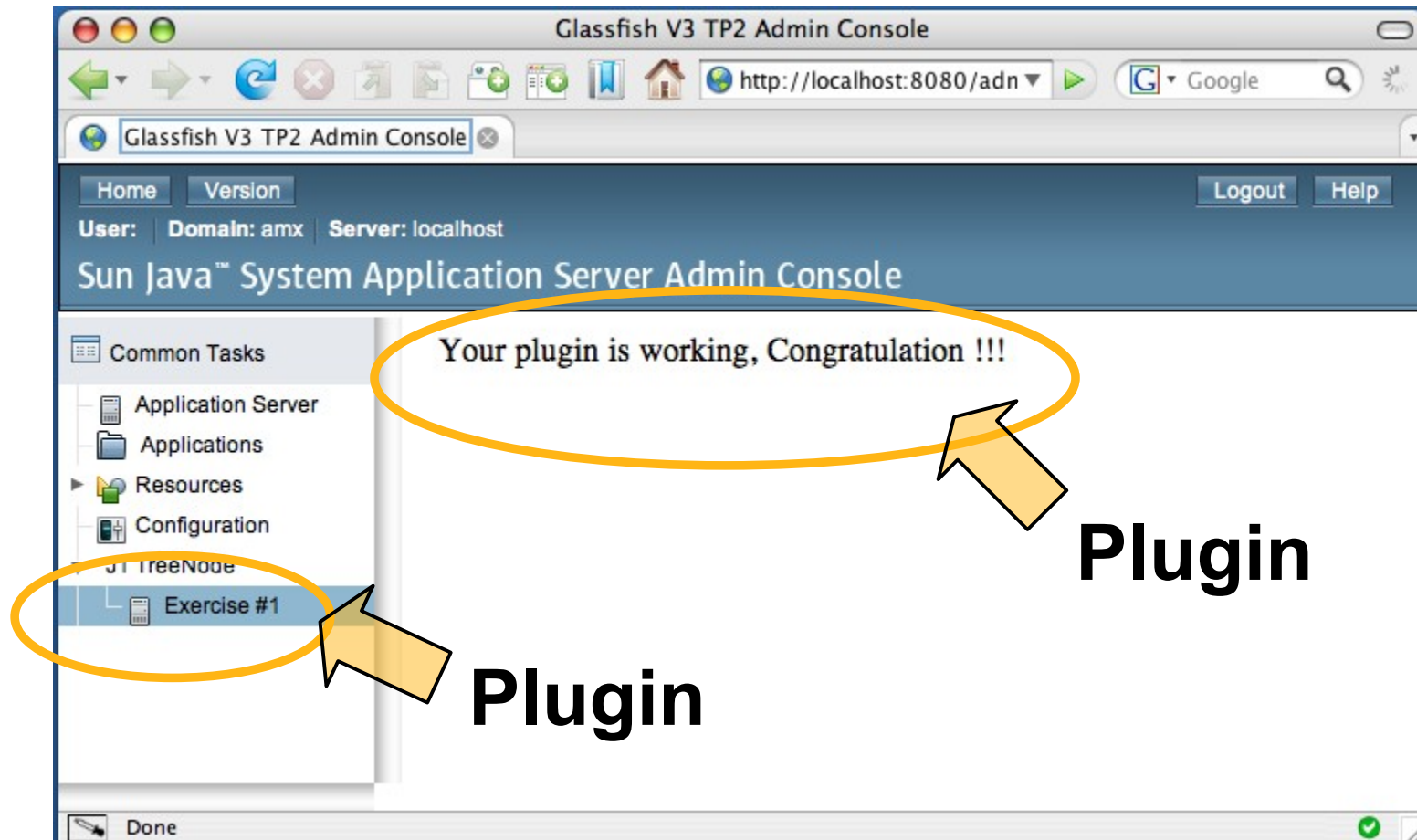
- Web module
 - > List/Create/Delete security Realms
 - > Manage user and group
- Woodstock Login component
- upgrade to Woodstock 4.2 GA release

Agenda

- V3 Admin Console Strategy
- **Console Plugins**

Admin Console v3 Plugins

What is an Admin Console Plugin?



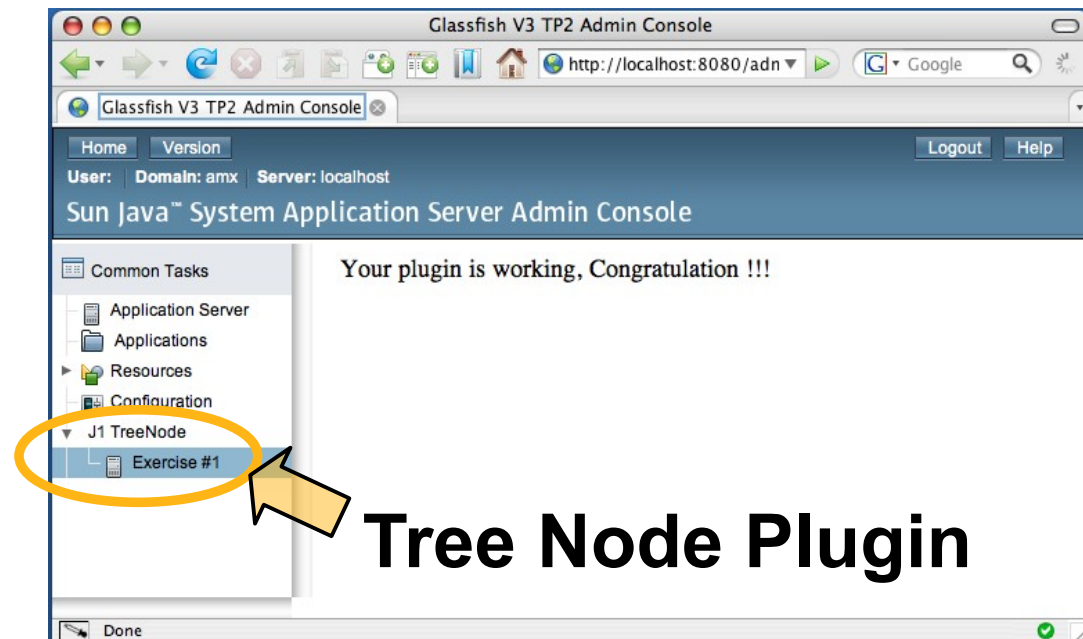
Plugin Examples

What is an Admin Console Plugin?

- Tree Nodes
- Pages
- Tabs
- Help
- Page-specific
- Theme

Integration Points

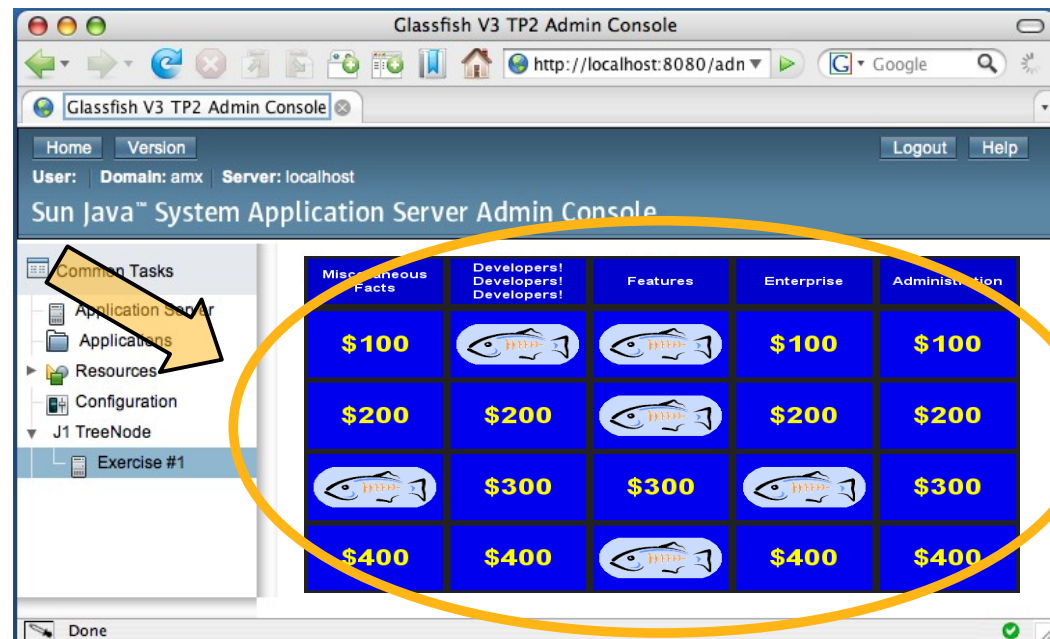
- Tree Nodes
 - > Allows plugin bundles to declare tree nodes
 - > Example: SIP and JBI specify new tree nodes allowing the user to navigate to SIP / JBI functionality



Integration Points

- Pages
 - > Allows plugin bundles to add pages
 - > Example: GlassFish Java One Jeopardy Game

Plugin Page



Integration Points

- Tabs
 - > Allows plugin bundles to add additional tabs to existing tab sets
 - > Example: A new “JBI” tab for clusters



The screenshot shows a web interface for managing a cluster. The breadcrumb path is "Clusters > cluster8". The navigation tabs are "General", "Applications", "Instances", "Resources", and "JBI". The "JBI" tab is highlighted with a red circle. Below the tabs, the "General Information" section is visible, containing a "Save" button and a note: "To start or stop any server instance, its node agent must be running. Refer to the online help for more information." Below this note are three buttons: "Start Cluster", "Stop Cluster", and "Migrate EJB Timers ...". At the bottom, the "Name:" is "cluster8" and the "Configuration:" is "cluster8-config".

Clusters > cluster8

General Applications Instances Resources **JBI**

General Information Save

To start or stop any server instance, its node agent must be running. Refer to the online help for more information.

Start Cluster Stop Cluster Migrate EJB Timers ...

Name: cluster8

Configuration: cluster8-config

Integration Points

- Online Help
 - > Allows for custom context sensitive help
 - > Provide a way for plugin bundles to integrate the help set so that it is:
 - > Searchable
 - > Merged with the help index
 - > Works with our context-sensitive help button
 - > Can link between plugin bundles
 - > Localized

Integration Points

- Page-specific Integration
 - > Plugin module provides additional content or actions
 - > Example: Application filter list includes types from the plugin module

Application Server Admin Console

Clusters > cluster8

General Applications Instances Resources JBI

Applications

Enable, disable, and remove selected applications associated with the cluster. Removing the application will only remove the application from this cluster, application will not be undeployed.

Applications (2)

| Deploy... Remove | -- More Actions -- | Filter: -- Show All --

	Name	Enabled	Type
<input checked="" type="checkbox"/>	clusterjsp	true	Enterprise A
<input type="checkbox"/>	hello	true	Web Applica

-- Show All --

-- Show All --

Enterprise Applications

Web Applications

EJB Modules

Lifecycle Modules

Connector Resources

Application Client Modules

Custom MBeans

EasyCrop Demo

20

Integration Points

- Page-specific Integration (continued)
 - > Example: Log Levels specific to the plugin show up in list of configurable log levels (not just a Property).

Util (javax.enterprise.system.util;)

Verifier (javax.enterprise.system.tools.verifier;)

Web Container (javax.enterprise.system.container.web; org.apache.catalina; org.apache.coyote; org.apache.jasper;)

Additional Properties (1)

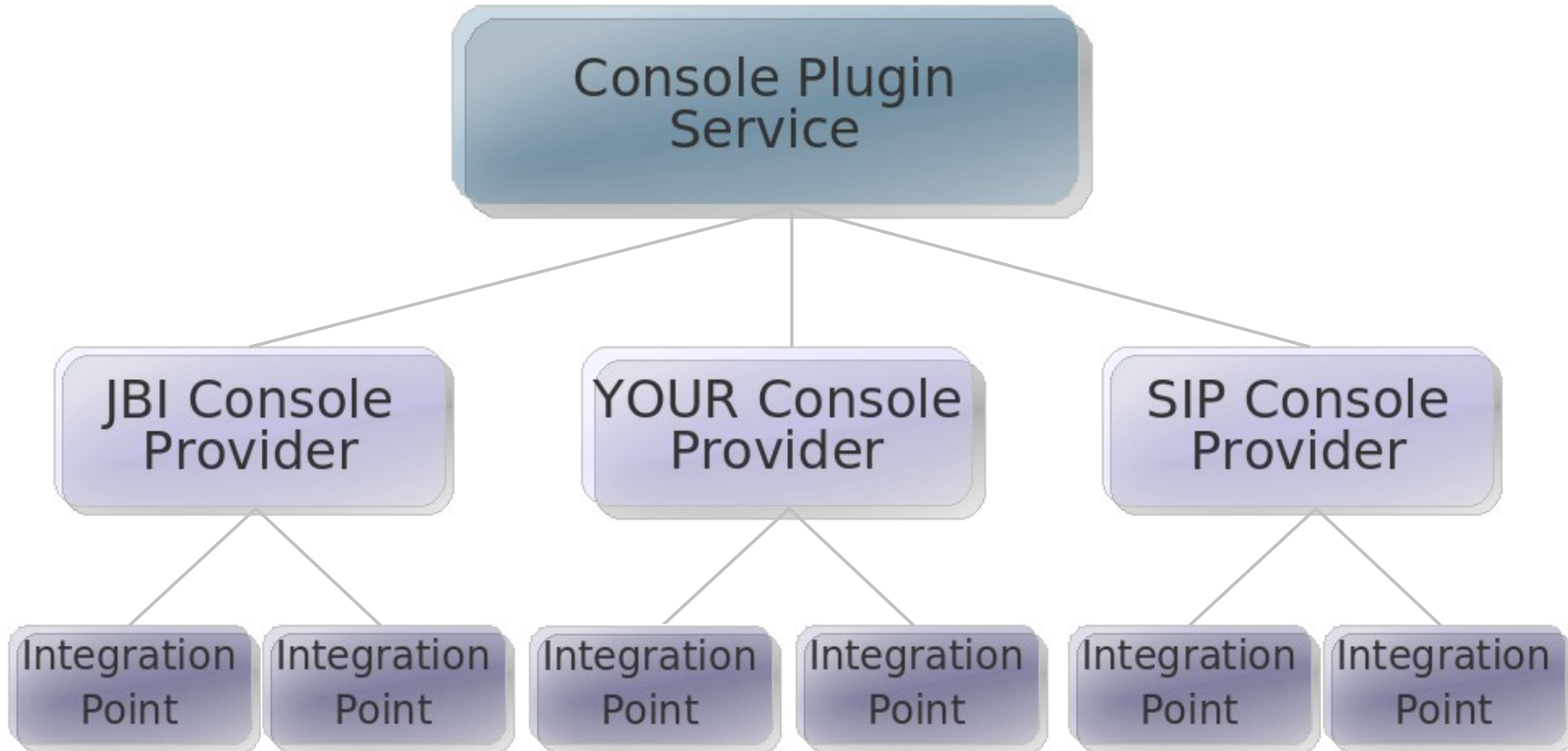
|

	Name	Value
<input type="checkbox"/>	com.sun.jbi.jsf	FINE

Integration Points

- Theme
 - > Ability to add or replace styles, images, and text that define the Look & Feel of the application.
 - > Example: Login Screen
 - > Example: Header Image / content
 - > Example: Page Colors

Plugin Design



Integration Design

- Console Plugin Service
 - > Admin GUI's access point for getting Integration Points from all plugin bundles

Console Provider

- Marker Service
- Provides console-config.xml URL
- Default: META-INF/admingui/console-config.xml

```
@Service
public class JavaOnePlugin implements
    ConsoleProvider {
    public URL getConfiguration() {
        return null;
    }
}
```

Integration Points

- Defined inside console-config.xml:

```
<?xml version="1.0" encoding="UTF-8"?>
<console-config id="javaone">
  <integration-point
    id="JavaOneNode"
    type="tree"
    priority="210"
    parentId="tree"
    content="treenode-ex1.jsf" />
</console-config>
```

Integration Points

- The “id” for the plugin bundle
- Visible in urls to plugin bundle resources

```
<?xml version="1.0" encoding="UTF-8"?>  
<console-config id="javaone">  
  <integration-point  
    id="JavaOneNode"  
    type="tree"  
    priority="210"  
    parentId="tree"  
    content="treenode-ex1.jsf" />  
</console-config>
```

Integration Points

- The “id” for the plugin bundle
- Visible in urls to plugin bundle resources

```
<?xml version="1.0" encoding="UTF-8"?>
<console-config id="javaone">
  <integration-point
    id="JavaOneNode"
    type="tree"
    priority="210"
    parentId="tree"
    content="treenode-ex1.jsf" />
</console-config>
```

Integration Points

- The integration-point “type” targets where an integration should occur

```
<?xml version="1.0" encoding="UTF-8"?>
<console-config id="javaone">
  <integration-point
    id="JavaOneNode"
    type="tree"
    priority="210"
    parentId="tree"
    content="treenode-ex1.jsf" />
</console-config>
```

Integration Points

- “priority” can be used for ordering
- “parentId” is used for fine-grained targeting

```
<?xml version="1.0" encoding="UTF-8"?>
<console-config id="javaone">
  <integration-point
    id="JavaOneNode"
    type="tree"
    priority="210"
    parentId="tree"
    content="treenode-ex1.jsf" />
</console-config>
```

Integration Points

- “content” specifies the contents of the integration, typically a JSF page fragment.

```
<?xml version="1.0" encoding="UTF-8"?>
<console-config id="javaone">
  <integration-point
    id="JavaOneNode"
    type="tree"
    priority="210"
    parentId="tree"
    content="treenode-ex1.jsf" />
</console-config>
```

OSGi

- Plugin Bundles need to be able to:
 - > Define new Java code that is invoked from JSFTemplating and does not implement any interface
 - > Provide other Java code / libraries
 - > Invoke Java code in the admin console web application, other plugin bundles, and in the admin console dependency bundles (i.e. glassfish-api, etc.)
 - > Provide faces-config.xml file entries which are scanned at application startup, JSF finds these via the context classloader (getResources).

OSGi

- Basically...

I want plugin bundles to behave as if they're in the WEB-INF/lib directory of the web application.

OSGi

- Current attempt to make this work:
 - > Provide common OSGi bundle for the Admin Console Web Application and all plugin bundles to depend on.

