

Here are the steps to implement the solution:

1. Download the request_attribute_binder.jar file that's attached to the case and put it in your <jasperserver-home>/WEB-INF/lib folder
2. Either download the attached applicationContext_semanticLayer.xml file or edit the current one in <jasperserver-home>/WEB-INF. Add the following bean entry to the binderList property of the groovyRunner bean:

```
<bean class="com.jaspersoft.commons.semantic.impl.RequestAttributesBinder">
  <property name="name" value="requestAttributes"/>
</bean>
```

```
180 - <!--
181 -   factory used to run stuff in groovy
182 -   centralize configuration here for use in security evaluation and sql generation
183 - -->
184 - <bean id="groovyRunner" class="com.jaspersoft.commons.groovy.GroovyRunnerFactory">
185 -   <property name="binderList">
186 -     <list>
187 -       <bean class="com.jaspersoft.commons.semantic.impl.AuthenticationBinder">
188 -         <property name="name" value="authentication"/>
189 -       </bean>
190 -       <bean class="com.jaspersoft.commons.semantic.impl.TenantBinder">
191 -         <property name="name" value="tenant"/>
192 -         <property name="tenantService" ref="tenantService"/>
193 -       </bean>
194 -       <bean class="com.jaspersoft.commons.semantic.impl.RequestAttributesBinder">
195 -         <property name="name" value="requestAttributes"/>
196 -       </bean>
197 -     </list>
198 -   </property>
199 -   <property name="defaultBindings">
200 -     <map>
201 -       <entry key="now">
202 -         <bean class="java.util.Date"/>
203 -       </entry>
```

3. Restart your application server
4. In the domain, create a calculated field. Leave the default type "String" selected and in the Expression textbox, enter:

```
groovy('requestAttributes.request.locale.toString()')
```

Available Fields:	Calculated Field
<div>Constants</div> <div>current_locale</div> <div>public_account</div>	<div>Field Name:</div> <div>current_locale</div> <div>Type:</div> <div>String</div> <div>Expression:</div> <div>groovy('requestAttributes.request.locale.toString()')</div>

5. Create another calculated field. This time, change the type to Boolean, double-click on the locale field in your domain to add it to the calculated field, then double-click to add the constant field you created in step 4 and use the == symbol to set them to be equal.

Available Fields:	Calculated Field
<div>Constants</div> <div>current_locale</div> <div>public_account</div> <div>account_description</div> <div>account_id</div> <div>account_parent</div> <div>account_rollup</div> <div>account_type</div> <div>custom_members</div> <div>locale</div> <div>locale_test</div>	<div>Field Name:</div> <div>locale_test</div> <div>Type:</div> <div>Boolean</div> <div>Expression:</div> <div>public_account.locale == constant_fields_level.current_locale</div>

6. Create a filter. Choose the calculated field you created in step 5 and set the filter condition to equals true

The image shows a software interface with two main panels: 'Fields' on the left and 'Filters' on the right.

Fields Panel: It has a header 'Fields' and a sub-header 'Constants'. Below this is a list of fields: 'public_account', 'account_description', 'account_id', 'account_parent', 'account_rollup', 'account_type', 'custom_members', 'locale', and 'locale_test'. The 'locale_test' field is highlighted with a green icon.

Filters Panel: It has a header 'Filters'. Below this is a configuration area for a filter. It shows 'locale_test' followed by a dropdown menu set to 'equals' and another dropdown menu set to 'true'. Below these are 'OK' and 'Cancel' buttons.

A red arrow points from a red-bordered box containing the text 'Use the field that's in the table, not the constant field' to the 'locale_test' field in the 'Fields' list.

7. Save the domain
8. Create a test Ad Hoc report using the domain. You should notice the locales other than the currently logged in user's are filtered out.

Note: This solution currently only works if the locales in the database are of the form lang or lang_Country (e.g. "en" or "en_US"). If you're using a different format for the locales, please let me know and I'll tweak the code to match what you're using.