Integrating an existing account system by using CAS

Adding account type field

- Adding a column in the user application system account, field is called usertype (it can also be named as other names), setting the value of the field for each account, each value is used to correspond to the role of information in the iPortal, the value of the field can be set as cas_ADMIN, cas_USER, cas_PUBLISHER here. According to the different level of each user, the value of usertype field is set respectively, cas_ADMIN is corresponding to the administrator role in iPortal, cas_USER is corresponding to the common user roles, cas_PUBLISHER is corresponding to the publisher roles.
- The SQL statement can be used to set usertype batches conveniently and quickly.

Configuring CAS Server

- Configuring CAS Server makes the user application system account (adding usertype column) log in CAS Server.
- Proceed as follows:
- Taking cas-server-webapp-3.4.12.war as an example here.

Preparing cas-server-webapp-3.4.12 package

- Unzipping the cas-server-webapp-3.4.12.war, add some jar packages in cas-server-webapp-3.4.12\WEB-INF\lib to connect to the user account database:

- mysql-connector-java-5.1.6-bin.jar
- commons-dbcp-1.4.jar
- commons-pool-1.6.jar

- In addition, users need to download the cas server-3.4.12-release.zip, then add the jar package (in addition to the cas-server-core-3.4.12. Jar, because the cas server-webapp-3.4.12 have had) of modules folder in the zip packege into the cas-server- webapp-3.4.12 \ WEB - INF \ lib folder.

Configuring data source connect to CAS

- Altering the configuration file WEB-INF/deployerConfigContext.xml：
- Adding the following configuration:

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
<bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource">
<property name="driverClassName">
  <!--Depending on the type of database that user account preserves-->
 <value>com.mysql.jdbc.Driver</value>
</property>
<property name="url">
  <!--Depending on the connecting way of database of user account-->
 <value> jdbc:mysql://localhost:3306/iportalusers</value>
</property>
<property name="username"><value>root</value></property>
<property name="password"><value></value></property>
</bean>
```

- Commenting out the following line of code:

```
<!-- <bean
class="org.jasig.cas.authentication.handler.support.SimpleTestUsernamePasswordAuthentic
 ationHandler" /> -->
```

- In the < property name = "authenticationHandlers" > node's < list > node, add the following content:

```
<bean class="org.jasig.cas.adaptors.jdbc.QueryDatabaseAuthenticationHandler">
<property name="dataSource" ref="dataSource" />
<property name="sql" value="select password from users where lower
(user)=lower(?)"/>
</bean>
```

- The meanings of the bold field in the code above are as follows:

- Password: variable {password}, users should write the actual account password field
- Users: variable {users}, users should write the table name saved in the database of actual user application system account.
- User: variable {user}, users should write the name field of the actual account

Configuring the returned attribute information after logging CAS

- Altering the configuration file WEB-INF/deployerConfigContext.xml：
- Commenting out the following line of code:

```
<!--   <bean id="attributeRepository"
class="org.jasig.services.persondir.support.StubPersonAttributeDao">
<property name="backingMap">
<map>
<entry key="uid" value="uid" />
<entry key="eduPersonAffiliation" value="eduPersonAffiliation" />
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
<entry key="groupMembership" value="groupMembership" />
<map>
</property>
</bean> -->
```

- Adding the following configuration:

```
<bean class="org.jasig.services.persondir.support.jdbc.SingleRowJdbcPersonAttributeDao" id="attributeRepository">
<!--Specifying the used data sources, here the dataSource is the data source which has been configured-->
<constructor-arg index="0" ref="dataSource"/>
<!-- Querying information of SQL statements from the database, the users is the table name of the user account-->
<constructor-arg index="1" value="select * from**users** where {0}"/>
<property name="queryAttributeMapping">
<map>
<!-- user is the name field of the account table-->
<entry key="username" value="**user**"/>
<map>
</property>
<property name="queryAttributeMapping">
<map>
<!-- Configuring the returned field information: usertype is the account type, id is the account id, roletype is used in iPortal configuration, namely the CAS user attributes field-->
<entry key="**usertype**'" value="**roletype**"/>
<entry key="id" value="guid" />
<map>
</property>
</bean>
```

- The meanings of the bold field in the code above are as follows:

- Users: variable {users}, users should write the table name of the actual user account
- User: variable {user}, users should write the name field of the the actual account
- Usertype: variables {usertype} , users should write account type field in the actual account table
- roletype: variables {roletype}, users should write the cas user attribute field configured by the casRealm. attributeRuleMapping in the shiro.ini file (located in the 【SuperMap iPortal support catalogue 】 / webapps/iPortal/WEB-INF directory) of iPortal

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

- Finding class as bean of org.jasig.cas.authentication.AuthenticationManagerImpl, and finding <property name="credentialsToPrincipalResolvers"> node's <list> node, altering the <list> node:

```
<bean class="org.jasig.cas.authentication.principal.UsernamePasswordCredentialsToPrincipalResolver" />
```

- To:

```
<bean class="org.jasig.cas.authentication.principal.UsernamePasswordCredentialsToPrincipalResolver" />
<property name="attributeRepository" ref="attributeRepository" />
</bean>
```

- Adding ignoreAttributes attribute in bean whose id is serviceRegistryDao, as follows:

```
<bean id="serviceRegistryDao" class="org.jasig.cas.services.InMemoryServiceRegistryDaoImpl">
    <property name="registeredServices">
        <list>
            <bean>
 class="org.jasig.cas.services.RegexRegisteredService">
                <property>
 name="id" value="0" />
                <property
 name="name" value="HTTP and IMAP" />
                <property
 name="description" value="Allows HTTP(S) and IMAP(S) protocols"
 />
                <property
 name="serviceId" value="^(https?|imaps?)://.*" />
                <property
 name="evaluationOrder" value="10000001" />
  <property name="ignoreAttributes"
 value="true" />
</bean>
```

- Returning attribute information, users also need to alter WEB-INF\view\jsp\protocol\2.0\casServiceValidationSuccess.jsp file, The specific method is to add after <cas:user>${fn:escapeXml(assertion.chainedAuthentications[fn:length(assertion.chainedAuthentications)-1].principal.id)}</cas:user>:

```
<cas:attributes>
<cas:test>test_value</cas:test>
<cas:test>test_value</cas:test>
<c:forEach var="attr" items="${auth.principal.attributes}">
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
<cas:${fn:escapeXml(attr.key)}>${fn:escapeXml(attr.value)}</cas:${fn:escapeXml(attr.key)}>
</c:forEach>
</c:forEach>
<cas:attributes>
```

Allowing to login CAS Server in the terms of HTTP

- Alter WEB-INF\spring-configuration\ticketGrantingTicketCookieGenerator.xml file to login cas using HTTP protocol: Set p:cookieSecure of CookieRetrievingCookieGenerator as "false".
- After the above configuration are completed, Place the cas-server-webapp--3.4.12 folder into the tomcat's webapps directory, renaming it as cas, and starting the tomcat, users can visit http://casserver:port/cas at this time, and login the cas using the user account of above configured iportalusers database.

Customizing the CAS Server login page

- Customizing the CAS Server login page to make CAS Server and iPortal use the same login page.
- Proceed as follows:

1. Customizing the CAS Server login page, users need to alter ui file, including casLoginView.jsp，casLogoutView.jsp，includes/top.jsp, includes/bottom.jsp under the cas/WEB-INF/view/jsp/default/ui directory, users need to customize to develop the login and logout page according to the requirements of application system. For example, users can alter jsp file to make visiting http://casserver:port/cas(/login) jump to iPortal home page, to make it jump to the iPortal home page after CAS Server quitting.
2. Opening iPortal manage page->Safety->Single login-on configuration, deselecting the check box "Retained the built-in account login";At this time opening iportal/web/login and the page will jump into http://casserver:port/cas/login?service=http://iportalAddress:port/iportal/shiro-cas, which is CAS Server login page. This step is not going to set, the changing of this step will be completed in altering the shiro.ini file part later.

Configuring the cas login settings of iPortal

- Before configuring iPortal, users firstly need to start CAS Server service.
- It is assumed that the CAS Server's address is http://192.168.120.58:8080/cas, iPortal's address is http://192.168.120.58:8090/iportal.
- Altering webapps\iportal\WEB-INF\shiro.ini file(users need to start the iPortal before, then the shiro.ini file is obtained) in iPortal package. Altering following parts

```
casRealm = com.supermap.services.security.CasRealm
casRealm.attributeRuleMapping = iserver_att={cas_SYSTEM=[SYSTEM, ADMIN], cas_PUBLISHER=[PUBLISHER], cas_USER=[USER]}
```


Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com

```
casRealm.iniFilePath = shiro.ini
casRealm.enabled = false
casRealm.reserveSystemAccount = true
casRealm.casServerUrlPrefix = http://{ip}:{port}/cas
casRealm.casService = http://{ip}:{port}/{contextPath}/shiro-cas
```

- Aa:

```
casRealm = com.supermap.iportal.security.IportalCasRealm
casRealm.backRealm = $usernamepasswordrealm
casRealm.attributeRuleMapping = roletype={cas_ADMIN=[ADMIN], cas_PUBLISHER=[PUBLISHER], cas_USER=[USER]}
casRealm.iniFilePath = shiro.ini
casRealm.enabled = true
casRealm.reserveSystemAccount = false
casRealm.casServerUrlPrefix = http://192.168.120.58:8080/cas
casRealm.casService = http://192.168.120.58:8090/iportal/shiro-cas
```

- The meanings of the bold field in the code above are as follows:

- roletype: variables {roletype}, users should write resultAttributeMapping returned attribute configured by the bean in the attributeRepository of WEB-INF/deployerConfigContext.xml file in the cas server

- **Note** : When a user application system account logins iPortal by the way of CAS, the user is recorded in iPortal user table, so, if a certain account in the user application system is deleted, users need to synchronously delete the user recorded in the iPortal, otherwise, the user can also occupy a user license of iPortal.

iPortal and users' application system achieving single sign-on

- After configuring according to the above steps, the user application system account can log on to the iPortal, and can make a map and other operations, but the user application system and the iPortal do not realize a single sign-on, now it only realizes logging iPortal using the account of the user application system. To realize single sign-on between the user application system and iPortal, users need to alter the user application system's web.xml, which makes logging in user application system need CAS login authentication. But it only realize login authentications, if the user application system needs more careful authorization, such as some url accessed only by some users, the users need to extend and develop CAS.

Login iPortal by CAS, personal center - > my information page will be blocked, if the user need to change the account information, such as changing the nickname, users can open the shield "my information" page, or the users themselves extend and develop page, add the page of changing the user's nickname, and then the changing of nickname resource by calling the iPortal. Note: cas is needed to be synchronized to connect the nickname in user account information and the user nickname recorded in the iPortal user table, making them consistent.

Add: 6/F, Building 107, No. A10, Jiuxianqiao North Road, Chaoyang District, Beijing, 100015, CHINA, 100015
E-mail: request@supermap.com Website: www.supermap.com