

,pm-diffs.PATCH

```
diff -r 763e4be166aa
edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/delegate/ServicesHandler.java
--- a/edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/delegate/ServicesHandler.java      Wed
Aug 11 15:54:18 2010 -0500
+++ b/edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/delegate/ServicesHandler.java    Fri Aug
27 09:34:30 2010 -0500
@@ -69,7 +69,7 @@
    }
```

```
-     private UserType validateSuppliedPassword (String method, String username, String
password, String domainController, String domain) throws Exception
+     protected UserType validateSuppliedPassword (String method, String username, String
password, String domainController, String domain) throws Exception
    {
        PMDbDao pmDb = new PMDbDao();
```

```
@@ -408,15 +408,13 @@
```

```
        UserType user = validateSuppliedPassword(method,
rmt.getUsername(), rmt.getPassword().getValue(), domainController, domain);
        userType.setFullName(user.getFullName());
+        userType.setUserName(user.getUserName());
+        userType.setDomain(rmt.getDomain());
    } catch (Exception e)
    {
        throw new Exception (e.getMessage());
    }

-        //if password was good then set info and generate a new session
key
-        userType.setUserName(rmt.getUsername());
-        userType.setDomain(rmt.getDomain());
-
        //SessionKey newKey;

        //build the date for the expiration of the key
@@ -436,7 +434,7 @@
        int timeout = 1800000;
        if (rmt.getPassword().getTokenMsTimeout() != null)
            timeout = rmt.getPassword().getTokenMsTimeout();
-        int result = pmDb.setSession(rmt.getUsername(), encryptedKey,
timeout);
+        int result = pmDb.setSession(userType.getUserName(),
encryptedKey, timeout);

        if (result < 1)
            throw new Exception("Failed to save session");
```

@@ -480,7 +478,7 @@
{

PMUtil.getInstance().convertToUppercaseStrings(userConfigurationType.getDataNeede
d());

PMUtil.getInstance().convertToUppercaseStrings(userConfigurationType.getProject());
- return runGetUserConfiguration(pmDb, userConfigurationType,
rmt.getUsername(), rmt.getDomain(), cType, uType);
+ return runGetUserConfiguration(pmDb, userConfigurationType,
uType.getUserName(), uType.getDomain(), cType, uType);
}

log.debug("Working on Rest of services: 1");

@@ -493,81 +491,81 @@

name = ((JAXBElement) obj).getName().getLocalPart();
log.debug("Element name is: " + name);
if (name.equals("set_user"))
- return runSetUser(pmDb, project, rmt.getUsername(),
(UserType) ((JAXBElement) obj).getValue());
+ return runSetUser(pmDb, project, uType.getUserName(),
(UserType) ((JAXBElement) obj).getValue());
else if (name.equals("get_all_user"))
- return runGetAllUser(pmDb, project, rmt.getUsername());
+ return runGetAllUser(pmDb, project, uType.getUserName()
);
else if (name.equals("set_project"))
- return runSetProject(pmDb, project, rmt.getUsername(),
(ProjectType) ((JAXBElement) obj).getValue());
+ return runSetProject(pmDb, project,
uType.getUserName(), (ProjectType) ((JAXBElement) obj).getValue());
else if (name.equals("get_all_project"))
- return runGetAllProject(pmDb, project, rmt.getUsername()
);
+ return runGetAllProject(pmDb, project,
uType.getUserName());
else if (name.equals("set_cell"))
- return runSetCell(pmDb, project, rmt.getUsername(),
(CellDataType) ((JAXBElement) obj).getValue());
+ return runSetCell(pmDb, project, uType.getUserName(),
(CellDataType) ((JAXBElement) obj).getValue());
else if (name.equals("delete_cell"))
- return runDeleteCell(pmDb, project, rmt.getUsername(),
(CellDataType) ((JAXBElement) obj).getValue());
+ return runDeleteCell(pmDb, project,
uType.getUserName(), (CellDataType) ((JAXBElement) obj).getValue());
else if (name.equals("get_cell"))
- return runGetCell(pmDb, project, rmt.getUsername(),
(CellDataType) ((JAXBElement) obj).getValue());
+ return runGetCell(pmDb, project, uType.getUserName(),
(CellDataType) ((JAXBElement) obj).getValue());

```

        else if (name.equals("get_all_cell"))
-           return runGetAllCell(pmDb, project, rmt.getUsername() );
+           return runGetAllCell(pmDb, project, uType.getUserName()
);
        else if (name.equals("set_global"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (GlobalDataType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (GlobalDataType) ((JAXBElement) obj).getValue() );
        else if (name.equals("set_project_param"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (ProjectType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (ProjectType) ((JAXBElement) obj).getValue() );
        else if (name.equals("set_project_user_param"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (ProjectType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (ProjectType) ((JAXBElement) obj).getValue() );
        else if (name.equals("get_all_project_user_param"))
-           return runGetAllParam(pmDb, project, rmt.getUsername(),
(ProjectType) ((JAXBElement) obj).getValue() );
+           return runGetAllParam(pmDb, project,
uType.getUserName(), (ProjectType) ((JAXBElement) obj).getValue() );
        else if (name.equals("set_user_param"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (UserType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (UserType) ((JAXBElement) obj).getValue() );
        else if (name.equals("set_cell_param"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (CellDataType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (CellDataType) ((JAXBElement) obj).getValue() );
        else if (name.equals("get_all_cell_param"))
-           return runGetAllParam(pmDb, project, rmt.getUsername(),
(CellDataType) ((JAXBElement) obj).getValue() );
+           return runGetAllParam(pmDb, project,
uType.getUserName(), (CellDataType) ((JAXBElement) obj).getValue() );
        else if (name.equals("set_hive_param"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (ConfigureType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (ConfigureType) ((JAXBElement) obj).getValue() );
        else if (name.equals("set_role"))
-           return runSetParam(pmDb, project, name,
rmt.getUsername(), (RoleType) ((JAXBElement) obj).getValue() );
+           return runSetParam(pmDb, project, name,
uType.getUserName(), (RoleType) ((JAXBElement) obj).getValue() );
        else if (name.equals("get_all_role"))

```

```

-                                     return runGetAllParam(pmDb, project, rmt.getUsername(),
(RoleType) ((JAXBElement) obj).getValue() );
+                                     return runGetAllParam(pmDb, project,
uType.getUserName(), (RoleType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("delete_role"))
-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
(RoleType) ((JAXBElement) obj).getValue() );
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), (RoleType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("get_role"))
-                                     return runGetParam(pmDb, project, rmt.getUsername(),
(RoleType) ((JAXBElement) obj).getValue() );
+                                     return runGetParam(pmDb, project, uType.getUserName(),
(RoleType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("set_hive"))
-                                     return runSetParam(pmDb, project, name,
rmt.getUsername(), (ConfigureType) ((JAXBElement) obj).getValue() );
+                                     return runSetParam(pmDb, project, name,
uType.getUserName(), (ConfigureType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("get_all_hive"))
-                                     return runGetAllParam(pmDb, project, rmt.getUsername(),
new ConfigureType() );
+                                     return runGetAllParam(pmDb, project,
uType.getUserName(), new ConfigureType() );
                                     else if (name.equals("delete_hive"))
-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
(ConfigureType) ((JAXBElement) obj).getValue() );
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), (ConfigureType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("get_hive"))
-                                     return runGetParam(pmDb, project, rmt.getUsername(),
(ConfigureType) ((JAXBElement) obj).getValue() );
+                                     return runGetParam(pmDb, project, uType.getUserName(),
(ConfigureType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("delete_project"))
-                                     return runDeleteProject(pmDb, rmt.getUsername(),
(ProjectType) ((JAXBElement) obj).getValue() );
+                                     return runDeleteProject(pmDb, uType.getUserName(),
(ProjectType) ((JAXBElement) obj).getValue() );
                                     else if (name.equals("get_project"))
-                                     return runGetProject(pmDb, rmt.getUsername(),
(ProjectType) ((JAXBElement) obj).getValue() );
+                                     return runGetProject(pmDb, uType.getUserName(),
(ProjectType) ((JAXBElement) obj).getValue() );

log.debug("working on value");
value = ((String) ((JAXBElement) obj).getValue()).trim();
log.debug("Element is single and is: " + value);
if (name.equals("delete_user"))

```



```

-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
global);
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), global);
}
else if (name.equals("delete_project_user_param"))
{
@@ -592,7 +590,7 @@
ProjectType global = new ProjectType();
global.setUserName("SET");
global.getParam().add(param);
-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
global);
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), global);
}
else if (name.equals("delete_cell_param"))
{
@@ -600,7 +598,7 @@
param.setld(Integer.valueOf(value));
CellDataType global = new CellDataType();
global.getParam().add(param);
-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
global);
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), global);
}
else if (name.equals("delete_hive_param"))
{
@@ -608,7 +606,7 @@
param.setld(Integer.valueOf(value));
ConfigureType global = new ConfigureType();
global.getParam().add(param);
-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
global);
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), global);
}
else if (name.equals("delete_user_param"))
{
@@ -616,26 +614,26 @@
param.setld(Integer.valueOf(value));
UserType global = new UserType();
global.getParam().add(param);
-                                     return runDeleteParam(pmDb, project, rmt.getUsername(),
global);
+                                     return runDeleteParam(pmDb, project,
uType.getUserName(), global);
}
}
else if (name.equals("get_all_hive_param"))

```

```

        {
            ConfigureType lcType = new ConfigureType();
            lcType.setDomainId(value);
            return runGetAllParam(pmDb, project,
- rmt.getUsername(),lcType);//(CellDataType) ((JAXBElement) obj).getValue() );
+ return runGetAllParam(pmDb, project,
uType.getUsername(),lcType);//(CellDataType) ((JAXBElement) obj).getValue() );
        }
        else if (name.equals("get_all_project_param"))
        {
            ProjectType pType = new ProjectType();
            pType.setId(value);
            return runGetAllParam(pmDb, project, rmt.getUsername(),
- pType );
+ return runGetAllParam(pmDb, project,
uType.getUsername(), pType );
        }
        else if (name.equals("get_all_user_param"))
        {
            UserType pType = new UserType();
            pType.setUsername(value);
            return runGetAllParam(pmDb, project, rmt.getUsername(),
- pType );
+ return runGetAllParam(pmDb, project,
uType.getUsername(), pType );
        }
        else if (name.equals("get_project_user_param"))
        {
@@ -645,7 +643,7 @@
            global.setUsername("USER");
            global.getParam().add(param);

            return runGetParam(pmDb, project, rmt.getUsername(),
- global );
+ return runGetParam(pmDb, project, uType.getUsername(),
global );

            // return runGetParam(pmDb, project, rmt.getUsername(),
(ProjectType) ((JAXBElement) obj).getValue() );
        }
@@ -656,7 +654,7 @@
            ProjectType global = new ProjectType();
            global.getParam().add(param);

            return runGetParam(pmDb, project, rmt.getUsername(),
- global );
+ return runGetParam(pmDb, project, uType.getUsername(),
global );
        }
        else if (name.equals("get_cell_param"))
        {

```

```

@@ -665,7 +663,7 @@
                                CellDataType global = new CellDataType();
                                global.getParam().add(param);

-                                return runGetParam(pmDb, project, rmt.getUsername()),
global );
+                                return runGetParam(pmDb, project, uType.getUserName()),
global );
                                }
                                else if (name.equals("get_user_param"))
                                {
@@ -674,7 +672,7 @@
                                UserType global = new UserType();
                                global.getParam().add(param);

-                                return runGetParam(pmDb, project, rmt.getUsername()),
global );
+                                return runGetParam(pmDb, project, uType.getUserName()),
global );
                                }
                                else if (name.equals("get_hive_param"))
                                {
@@ -683,7 +681,7 @@
                                ConfigureType global = new ConfigureType();
                                global.getParam().add(param);

-                                return runGetParam(pmDb, project, rmt.getUsername()),
global );
+                                return runGetParam(pmDb, project, uType.getUserName()),
global );
                                }
                                }
}

```

```

diff -r 763e4be166aa
edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/delegate/ServicesHandlerCAS.java
--- /dev/null Thu Jan 01 00:00:00 1970 +0000
+++ b/edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/delegate/ServicesHandlerCAS.java Fri Aug
27 09:34:30 2010 -0500
@@ -0,0 +1,99 @@
+/**
+ * Copyright (c) 2010 University of Kansas Medical Center (kumc.edu).
+ * @todo: determine license terms.
+ */
+package edu.harvard.i2b2.pm.delegate;
+
+import java.io.BufferedReader;
+import java.io.InputStreamReader;
+import java.net.URLConnection;
+import java.net.URLEncoder;
+import java.util.Iterator;
+import java.util.List;

```



```

+
+import edu.harvard.i2b2.common.exception.I2B2DAOException;
+import edu.harvard.i2b2.common.exception.I2B2Exception;
+import edu.harvard.i2b2.pm.dao.PMDao;
+import edu.harvard.i2b2.pm.datavo.pm.UserType;
+import edu.harvard.i2b2.pm.delegate.ServicesHandler;
+import edu.harvard.i2b2.pm.ws.ServicesMessage;
+
+/**
+ * ServicesHandlerCAS validates users using
+ * JA-SIG Central Authentication Service (CAS).
+ *
+ * @author: Dan Connolly
+ *
+ * @param: service: CAS service URL is passed in place of a username
+ * @param: ticket: CAS ticket is passed in place of a password
+ *
+ * @todo: Change hard-coded CAS server address to a deploy-time option.
+ */
+public class ServicesHandlerCAS extends ServicesHandler {
+    protected static final String casadr = "https://cas.kumc.edu/cas/"; //HARDCODED
+    protected static Exception fail = new Exception("Username or password does not exist");
+
+    public ServicesHandlerCAS(ServicesMessage servicesMsg) throws I2B2Exception{
+        super(servicesMsg);
+    }
+
+    protected UserType validateSuppliedPassword (String method,
+                                                String service,
+                                                String ticket,
+                                                String domainController,
+                                                String domain)
+        throws Exception {
+
+        // support password-based accounts too for OBFSC_SERVICE_ACCOUNT
+        if (! (service.startsWith("http:")
+            || service.startsWith("https:"))){
+            return super.validateSuppliedPassword(method, service,
+                                                ticket, domainController,
+                                                domain);
+        }
+
+        String addr = casadr + "validate?"
+            + "service=" + URLEncoder.encode(service)
+            + "&ticket=" + URLEncoder.encode(ticket);
+        log.debug("CAS validation address: " + addr);
+
+        BufferedReader body = URLOpener.open(addr);
+        if (body.readLine().equals("yes") == false){
+            log.debug("CAS authentication result negative");
+            throw fail;
+        }
+    }
+}

```

```

+     }
+
+     String username = body.readLine();
+     log.debug("CAS authenticated user:" + username);
+
+     PMDbDao pmDb = new PMDbDao();
+     List answers;
+     try {
+         answers = pmDb.getUser(username, null);
+     } catch (I2B2DAOException dberr) {
+         log.debug(dberr.toString());
+         throw fail;
+     }
+
+     Iterator users = answers.iterator();
+     if (!users.hasNext()) {
+         log.debug("No such user record: " + username);
+         throw fail;
+     }
+
+     return (UserType)users.next();
+ }
+}
+
+/**
+ * a la python's urloper
+ * Note: assumes utf-8
+ */
+class URLOper {
+    public static BufferedReader open(String addr)
+        throws java.net.MalformedURLException, java.io.IOException {
+        URLConnection conn = new java.net.URL(addr).openConnection();
+        return new BufferedReader(new InputStreamReader(conn.getInputStream(),
+            "utf-8"));
+    }
+}
+}
diff -r 763e4be166aa edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/ws/PMSERVICE.java
--- a/edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/ws/PMSERVICE.java    Wed Aug 11 15:54:18
2010 -0500
+++ b/edu.harvard.i2b2.pm/src/edu/harvard/i2b2/pm/ws/PMSERVICE.java    Fri Aug 27 09:34:30
2010 -0500
@@ -15,6 +15,7 @@
import edu.harvard.i2b2.pm.datavo.i2b2message.ResponseMessageType;
import edu.harvard.i2b2.pm.delegate.RequestHandler;
import edu.harvard.i2b2.pm.delegate.ServicesHandler;
+import edu.harvard.i2b2.pm.delegate.ServicesHandlerCAS;
import edu.harvard.i2b2.pm.ws.ExecutorRunnable;
import edu.harvard.i2b2.pm.ws.MessageFactory;

@@ -94,6 +95,8 @@
 * @throws PortletServiceNotFoundException

```

```

    * @throws PortletServiceUnavailableException
    * @throws Exception
+
    *
+
    * @todo: make ServicesHandler vs ServicesHandlerCAS a run-time choice.
    */
    public OMElement getServices(OMElement getPMDDataElement)
        throws I2B2Exception {
@@ -166,7 +169,8 @@
        //er.setInputString(requestElementString);
        log.debug("begin setRequestHandler, my servicesMsg: " + servicesMsg);

-
        er.setRequestHandler(new ServicesHandler(servicesMsg));
+
        //@@er.setRequestHandler(new ServicesHandler(servicesMsg));
+
        er.setRequestHandler(new ServicesHandlerCAS(servicesMsg));
        log.debug("middle setRequestHandler");

```