ADD A PERMANENT LINK BETWEEN TWO SERVICES

A permanent link can be used to give a more permanent structure to a network. It is created between two services, through only one call and is not removed until the request is made. This is different from a dynamic link that can be updated dynamically, based on reinforcement fed into the system, and manages its own link weight from that. To use a permanent link:

- Use the Service class's `createPermanentLinkTo` method to create a permanent link from one service to another service.
- Use the Service class's `removePermanentLinkTo` method to remove a permanent link from one service to another service.

The link remains in existence until it is directly removed. The admin key for the service that makes the request is required, because this is a permanent change in the service's and network's structure. This means that only the administrator of the service would be able to add a permanent link, although the service itself could do so internally. The following code shows how the commands can be invoked remotely.

CREATE A PERMANENT LINK

```
//some variables that might be required
String serviceFrom;                 //id of service linking from
String serviceTo;                   //id of service linking to
Element permanentLinkFromPath;      //path to service linking from
Element permanentLinkToPath;        //path to service linking to


permanentLinkFromPath = //get the path for the service to link from as a Handle 'Element'
permanentLinkToPath = //get the path for the service to link to as a Handle 'Element'

if ((permanentLinkFromPath != null) && (permanentLinkToPath != null))
{
    serviceFrom = Handle.getLastHandle(permanentLinkFromPath);
    serviceTo = Handle.getLastHandle(permanentLinkToPath);

    //retrieve the password for the service linking to as it is required as a parameter
    //the service linking from needs to invoke a method on the service linking to as well
    try
    {
        servicePassword = passwordHandler.getServicePassword(serviceTo);
    }
    catch (PasswordException exp)
    {
```

```
            //can ask for the password if it is not yet known
            methodInfo = new MethodInfo();
            methodInfo.setName(MethodConst.GETPASSWORD);
            methodInfo.setRtnType(TypeConst.STRING);
            methodInfo.setServiceURI(permanentLinkToPath);
            methodInfo.addParam(Const.ANON);
            methodInfo.addParam(XMLFactory.getInstance().createElement("abc", "abc"));
            servicePassword = (String)call.call(methodInfo);

            //add the password if it is missing
            passwordHandler.addServicePassword(serviceTo, servicePassword);
        }

        //invoke the method on the service linking from to try and create the permanent link
        if (servicePassword != null)
        {
            methodInfo = new MethodInfo();
            methodInfo.setName(MethodConst.CREATEPERMANENTLINKTO);
            methodInfo.setRtnType(TypeConst.BOOLEAN);
            methodInfo.setServiceURI(permanentLinkFromPath);
            methodInfo.setServerPassword(getServicePassword(serverUri));
            methodInfo.setPassword(getServicePassword(serviceFrom));
            methodInfo.getParams().add(getAdminKey(serviceFrom));
            methodInfo.getParams().add(serviceTo);
            methodInfo.getParams().add(getServicePassword(serviceTo));
            methodInfo.getParams().add(permanentLinkToPath);
            isOK = ((Boolean)call.call(methodInfo)).booleanValue();
        }
}
```

REMOVE A PERMANENT LINK

This is the same as above, except that the method call now looks as follows:

```
if (servicePassword != null)
{
    methodInfo = new MethodInfo();
    methodInfo.setName(MethodConst.REMOVEPERMANENTLINKTO);
    methodInfo.setRtnType(TypeConst.BOOLEAN);
    methodInfo.setServiceURI(permanentLinkFromPath);
    methodInfo.setServerPassword(passwordHandler.getServicePassword(serverUri));
    methodInfo.setPassword(getServicePassword(serviceFrom));
    methodInfo.getParams().add(new ParamInfo(getAdminKey(serviceFrom)));
    methodInfo.getParams().add(new ParamInfo(serviceTo));
    methodInfo.getParams().add(new ParamInfo(getServicePassword(serviceTo)));
    methodInfo.getParams().add(new ParamInfo(permanentLinkToPath));
    isOK = ((Boolean)call.call(methodInfo)).booleanValue();
}
```