

How to Call From Java Proxy Framework Without Using JCo

Document Information

- XI Area: Proxy Framework
- Release: XI 1.0
- Keywords: JCo, Java Proxy Framework

Problem Description

Before a Java application can send XML messages using HTTP to the Integration Server (IS), it must know which (technical) user it must use to log on to the IS. This information is located in the *exchange profile* and is saved in an R/3 database. In the current *all in one* installation this is the IS database. You can access the exchange profile externally using RFC. From the JPF the JCo is used for this purpose.

To access the exchange profile you also require a user/password that is recognized by the Java application. This initial user is normally accessible locally in the property file *dbconnect.properties*.

Therefore, calling Java applications and the IS communicate using two different protocols, namely RFC and HTTP.

If a Java application now wants to avoid this JCo-RFC call, for example, for security reasons, then it must locally save the data from the exchange profile that is required by the JPF in the file *jpf.properties*, and delete *dbconnect.properties* locally.

Procedure

The following entries are required by the JPF. If *dbconnect.properties* is not available, then they must be entered in *jpf.properties*:

```
com.sap.aii.server.name.lcr           the hostname of the LCR server
com.sap.aii.server.httpport.lcr      the HTTP port number of the LCR server
com.sap.aii.proxy.framework.serviceuser.name=XIAPPLUSER
com.sap.aii.proxy.framework.serviceuser.pwd=XIPASS
com.sap.aii.proxy.framework.serviceuser.language=EN
```

(The following entries are also required, but, if necessary, can be replaced by JPF internal program defaults:

```
com.sap.aii.proxy.framework.messageLoggingFilename
com.sap.aii.proxy.framework.messageTraceLevel
com.sap.aii.proxy.framework.messageTraceFilename
com.sap.aii.proxy.framework.serverURL
com.sap.aii.proxy.framework.serverIntervallInMSec
com.sap.aii.proxy.framework.flagFilename
com.sap.aii.proxy.framework.registryFilename)
```

Notes

1. The *inbound* Java runtime requires neither the system landscape directory (SLD) nor the service user above. Therefore, it does not matter whether the exchange profile can be read or not.
2. The disadvantage of the local solution described above is the separation from the central system landscape administration (for example, when changing the SLD host or port), which, if necessary, may require you to make manual adjustments. This is because the entries in *jpf.properties* override the exchange profile.