**JCO程式清除Meta Cache方法**

1. JCO建立連線的共用元件 tcc-jco.jar 中的JCoConnection.java，

新增下列 method。

/\*\*

\* 清除特定 Destination 的 Repository (用於RFC有變更規格時)

\* @param destinationName

\* @throws JCoException

\*/

public void clearRepository(String destinationName) throws JCoException{

if( provider.getDestinationProperties(destinationName)!=null ){

clearRepository(JCoDestinationManager.getDestination(destinationName));

}

}

/\*\*

\* 清除 Destination 的 Repository (用於RFC有變更規格時)

\* @param jcoDestination

\* @throws JCoException

\*/

public void clearRepository(JCoDestination jcoDestination) throws JCoException{

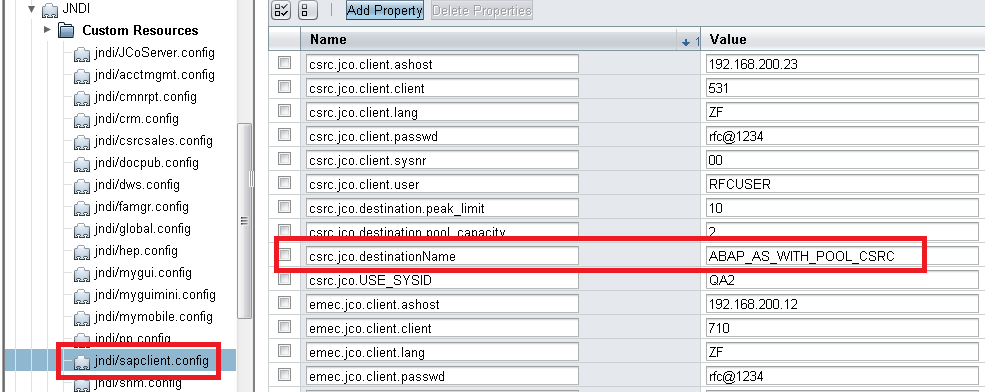
if( jcoDestination!=null ){

**jcoDestination.getRepository().clear();**

}

}

1. 上述 destinationName 指的JNDI中JCO連線設定中[sapClientCode].jco.destinationName 設定的值。如下圖紅框處。



1. 若JCO程式以proxy & proxyFactory方式撰寫，使用方式範例程式如下
2. 於proxy 實作程式新增下列method

@Override

public void clearRepository(){

if( destination!=null ){

try {

**destination.getRepository().clear();**

} catch (JCoException ex) {

logger.error("clearRepository exception :\n", ex);

}

}

}

1. Controller 執行

**@Resource(mappedName = "jndi/sapclient.config")**

private Properties jndiConfig;

public void doJCoClear(){

SapClientEnum[] sapClients = SapClientEnum.values();

for(int i=0; sapClients!=null && i<sapClients.length; i++){

String sapClientCode = sapClients[i].getSapClientCode();

Properties jcoProp = JCoUtils.getJCoProp(jndiConfig, sapClientCode); //取得相關Jco連結參數

PpProxy ppProxy = PpProxyFactory.createProxy(jcoProp);

**ppProxy.clearRepository();**

}

}

1. 測試步驟：

* SAP上新RFC 前，先通知啟動 GlassFish 及 測試AP, 並用WEB舊程式呼叫舊版 RFC，讓 JCO 產生舊版 RFC 的 Meta Cache.
* 通知 SAP上新版RFC
* 更新新版 WEB程式，但不重啟 WEB AP Server。執行新版WEB程式呼叫RFC，此時理論上會有 JCO ERROR。
* 在不重起GlassFish 狀況下，執行 repository.clear。之後再執行一次新版WEB程式呼叫RFC，看看是否就能成功。