

PROJECT

SV7QCARD JAVA Application

Version 1.0

September 4, 2005

## Version History

<b>Date</b>	<b>Version #</b>	<b>Description</b>	<b>Author</b>
09/05/2005	1.0a	Final Version	Vincent Taylor

## Introduction

This work request was submitted by SM.

## Scope

Write a Java Application that will connect to the Local IT Hub, extract data, transfer data to the local mainframe, and generate emails to SMEs.

## Issues

None foreseen at this time.

## Dependencies

N/A.

```

/* ****
*
* Confidential and Proprietary.
* Copyright 2006
* All rights reserved.
* Author: Vincent Taylor
*
* Created Sept 2005
*
* Description: A class to handle Extraction of Cycle Closes from Local Hub.
*
* Contents: class SV7QCard
*
****
*
* PVCS Revision History:
* -----
*
* $Log: //vmfs/vcs/local/src/java/com/xxxxxx/lmrs/file/SV7QCard.j!va $
*/

```

```
package com.xxxxxx.business.assembly.vip;

import com.dovetail.jzos.*;

import java.io.*;
import javax.xml.soap.*;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.Transformer;
import javax.xml.transform.dom.DOMResult;
import javax.xml.transform.stream.StreamSource;
```

```

public class SV7QCard
{
    public static void main(String[] args) throws Exception
    {
        java.util.Iterator it1 = null;
        java.util.Iterator it2 = null;
        java.util.Iterator it3 = null;

        String DMA = "0";
        String SSC = "0";
        String DMASSC = "0";
        String SDMA = " ";

    try
    {
        /*-----*
        /*Get the endpoint address from the ENDPOINT DDNAME control card.  *
        /*-----*
        char[] buf = new char[100];

        ZFile endPointFile = new ZFile("//DD:ENDPOINT","r");

        InputStreamReader endPointStreamReader =
            new InputStreamReader(endPointFile.getInputStream(),
ZFile.DEFAULT_EBCDIC_CODE_PAGE);
        int length = endPointStreamReader.read(buf);
        if (length == -1)
        {
            throw new RuntimeException("ENDPOINT DDNAME cannot be opened or contains
nothing.");
        }

        endPointStreamReader.close();

        String endpoint = new String(buf).trim();

        /*-----*
        /* Create an empty SOAP message.  *
        /*-----*
        MessageFactory mf = MessageFactory.newInstance();
        SOAPMessage sm = mf.createMessage();

        /*-----*
        /* Accessing 4 Elements of a Message. Its Part, Envelope, Header and Body *
        /* Either to Add Contents or to Retrieve Contents *
        /*-----*

```

```

/*-----*
SOAPPart sp = sm.getSOAPPart();
SOAPEnvelope se = sp.getEnvelope();
SOAPHeader sh = se.getHeader();
SOAPBody sb = sm.getSOAPBody();

// Open the input request file.
ZFile inFile = new ZFile("//DD:REQUEST","r");

InputStreamReader inputStreamReader =
    new InputStreamReader(inFile.getInputStream(),
ZFile.DEFAULT_EBCDIC_CODE_PAGE);

StreamSource streamSource = new StreamSource(inputStreamReader);

// Convert the input request file to a SOAP Element and
// add it to the SOAP Body.

TransformerFactory transformerFactory = TransformerFactory.newInstance();
Transformer transformer = transformerFactory.newTransformer();
DOMResult domResult = new DOMResult(sb);
transformer.transform(streamSource, domResult);

/*-----*
/* Get the name of the input XML request type to set the SOAPAction. *
/* Per Local Development standards, the SOAPAction is always the *
/* name of the service/operation. The XML request type name is always *
/* the service/operation name with "Request" added to the end. *
/* Therefore, strip off "Request" to get the SOAPAction. *
/*-----*

it1 = sb.getChildElements();

SOAPElement request = null;

while (it1.hasNext())
{
    if (request != null)
    {
        throw new RuntimeException("Input request should contain only one top level request
type.");
    }
    request = (SOAPElement)it1.next();
}

String requestName = request.getNodeName();

```

```

int serviceNameLength = requestName.indexOf("Request");

if (serviceNameLength == -1 ||
    requestName.length() != (serviceNameLength + 7))
{
    throw new RuntimeException("The name of the input request has to end with
\"Request\".");
}

String SoapAction = requestName.substring(0, serviceNameLength);

// Set the SOAPAction.
MimeHeaders hds = sm.getMimeHeaders();
hds.addHeader("SOAPAction", "\"" + SoapAction + "\"");

// Set additional SOAP Message properties and save changes.

sm.setProperty(SOAPMessage.WRITE_XML_DECLARATION, "true");
sm.saveChanges();

// Call the service.
SOAPConnectionFactory factory = SOAPConnectionFactory.newInstance();
SOAPConnection con = factory.createConnection();
SOAPMessage responseMessage = con.call(sm, endpoint);

/*-----*
/* Get the response from the service. *
/*-----*
SOAPBody responseBody = responseMessage.getSOAPBody();

/*-----*
/* Check if SOAP Fault received. *
/*-----*
if (responseBody.hasFault())
{
    SOAPFault soapFault = responseBody.getFault();
    System.err.println("SOAP Fault received from service:");
    System.err.println("  Fault Actor : " + soapFault.getFaultActor());
    System.err.println("  Fault Code : " + soapFault.getFaultCode());
    System.err.println("  Fault String: " + soapFault.getFaultString());
    throw new RuntimeException("SOAP Fault received from service.");
}

```

```

/*-----*
/* Get the response.  *
/*-----*
it1 = responseBody.getChildElements();

SOAPElement response = null;

while (it1.hasNext())
{
    if (response != null)
    {
        throw new RuntimeException("Service response should contain only one top level
response type.");
    }
    response = (SOAPElement)it1.next();

}
// Returns an iterator over all the immediate child nodes of this element
it1 = response.getChildElements();

/*-----*
/* Set up write variable *
/*-----*
ZUtil.setDefaultOutputEncoding(ZFile.DEFAULT_EBCDIC_CODE_PAGE);
ZFile outFile = new ZFile("//DD:RECORDS","wb,type=record,noseek");

while (it1.hasNext())
{
    SOAPElement record = (SOAPElement)it1.next();
    String name = record.getElementName().getLocalName();

    it2 = record.getChildElements();
    while (it2.hasNext())
    {
        SOAPElement child2 = (SOAPElement)it2.next();
        name = child2.getElementName().getLocalName();

        if (name.equals("geographyID"))
        {
            DMA = child2.getValue();
        }
        if (name.equals("samplingServiceCollectionMethodID"))
        {
            SSC = child2.getValue();
        }
    }
}

```

```

it3 = child2.getChildElements();
while (it3.hasNext())
{
    SOAPElement child3 = (SOAPElement)it3.next();
    name = child3.getElementName().getLocalName();

    if (name.equals("geographyID"))
    {
        DMA = child3.getValue();
    }
    if (name.equals("samplingServiceCollectionMethodID"))
    {
        SSC = child3.getValue();
        if (SDMA != DMA)
        {
            DMASSC = DMA + " " + SSC;
            outFile.write(DMASSC.getBytes());
        }
    }
} //Terminate while it3
} //Terminate while it2
//Save DMA value
SDMA = DMA;
} //Terminate while it1
outFile.close();
} //Terminate Try
catch(Exception e)
{
    System.err.println("Caught exception: " + e);
    e.printStackTrace();
    System.exit(16);
}
System.err.println("Version 2.0.");
} /* terminate public static void logic

} /* terminate getRecord program

```

**# This is a shell script which configures # any environment variables for the Java JVM.  
# Variables must be exported to be seen by the launcher.**

```

export JZOS_HOME=/u/home/nsdvpt/lib/jzos_1.1.2
export JAVA_HOME=/usr/lpp/java/J1.4
export LIB_HOME=/u/home/nsdvpt/lib

export PATH="${JAVA_HOME}"/bin:/bin

```



```

LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin
LIBPATH="$LIBPATH":"${JAVA_HOME}"/bin/classic
LIBPATH="$LIBPATH":"${JZOS_HOME}"
export LIBPATH="$LIBPATH":

# SV7Q jar file.
CLASSPATH="${LIB_HOME}"/SV7Q.jar

# Add JZOS required jars to end of CLASSPATH
for i in "${JZOS_HOME}"/*.jar; do
    CLASSPATH="$CLASSPATH":"$i"
done

# Add jwsdp required jars to end of CLASSPATH
for i in "${LIB_HOME}/jwsdp_1.4"/*.jar; do
    CLASSPATH="$CLASSPATH":"$i"
done

export CLASSPATH="$CLASSPATH":

# Set JZOS specific options
# Use this variable to specify encoding for DD STDOUT and STDERR
export JZOS_OUTPUT_ENCODING=Cp1047
# Use this variable to prevent JZOS from handling MVS operator commands
export JZOS_ENABLE_MVS_COMMANDS=false
# Use this variable to supply additional arguments to main
#export JZOS_MAIN_ARGS=""

# Configure JVM options
IJO="-Xms16m -Xmx128m "
IJO="$IJO -Djzos.home=${JZOS_HOME}"
IJO="$IJO -Djava.endorsed.dirs=${LIB_HOME}/jwsdp_1.4"

export IBM_JAVA_OPTIONS="$IJO "

export JAVA_DUMP_HEAP=false
export JAVA_PROPAGATE=YES
export IBM_JAVA_ZOS_TDUMP=NO

SV7QD01G
com.XXXXXX.business.assembly.vip.SV7QCard

SV7QD11G

# This is a shell script which configures

```

```

# any environment variables for the Java JVM.
# Variables must be exported to be seen by the launcher.

export JZOS_HOME=/u/home/nsdvpt/lib/jzos_1.1.2
export JAVA_HOME=/usr/lpp/java/J1.4
export LIB_HOME=/u/home/nsdvpt/lib

export PATH="${JAVA_HOME}"/bin:/bin

LIBPATH="$LIBPATH:${JAVA_HOME}/bin
LIBPATH="$LIBPATH:${JAVA_HOME}/bin/classic
LIBPATH="$LIBPATH:${JZOS_HOME}"
export LIBPATH="$LIBPATH":

# SV7Q jar file.
CLASSPATH="${LIB_HOME}"/SV7Q.jar

# Add JZOS required jars to end of CLASSPATH
for i in "${JZOS_HOME}"/*.jar; do
    CLASSPATH="$CLASSPATH":$i
done

# Add jwsdp required jars to end of CLASSPATH
for i in "${LIB_HOME}/jwsdp_1.4"/*.jar; do
    CLASSPATH="$CLASSPATH":$i
done

export CLASSPATH="$CLASSPATH":

# Set JZOS specific options
# Use this variable to specify encoding for DD STDOUT and STDERR
export JZOS_OUTPUT_ENCODING=Cp1047
# Use this variable to prevent JZOS from handling MVS operator commands
export JZOS_ENABLE_MVS_COMMANDS=false
# Use this variable to supply additional arguments to main
#export JZOS_MAIN_ARGS=""

# Configure JVM options
IJO="-Xms16m -Xmx128m "
IJO="$IJO -Djzos.home=${JZOS_HOME}"
IJO="$IJO -Djava.endorsed.dirs=${LIB_HOME}/jwsdp_1.4"

export IBM_JAVA_OPTIONS="$IJO "

export JAVA_DUMP_HEAP=false
export JAVA_PROPAGATE=YES

```

export IBM\_JAVA\_ZOS\_TDUMP=NO

## SV7QD21G

http://p60loc.XXXXXXX.com:8080/BusinessStandardReference/Services

## SV7QD31G

```
<?xml version="1.0" encoding="IBM-1047"?>
<getReportingGeographiesBySamplingServiceCollectionMethodRequest
  xmlns="local:XXXXXX:business:standard:reference">
  <samplingServiceCollectionMethodIDs>
<samplingServiceCollectionMethodID>1</samplingServiceCollectionMethodID>
  </samplingServiceCollectionMethodIDs>
  <samplingServiceCollectionMethodIDs>
<samplingServiceCollectionMethodID>2</samplingServiceCollectionMethodID>
  </samplingServiceCollectionMethodIDs>
  <samplingServiceCollectionMethodIDs>
<samplingServiceCollectionMethodID>3</samplingServiceCollectionMethodID>
  </samplingServiceCollectionMethodIDs>
  <samplingServiceCollectionMethodIDs>
<samplingServiceCollectionMethodID>4</samplingServiceCollectionMethodID>
  </samplingServiceCollectionMethodIDs>
  <samplingServiceCollectionMethodIDs>
<samplingServiceCollectionMethodID>5</samplingServiceCollectionMethodID>
  </samplingServiceCollectionMethodIDs>
  <samplingServiceCollectionMethodIDs>
<samplingServiceCollectionMethodID>6</samplingServiceCollectionMethodID>
  </samplingServiceCollectionMethodIDs>
  <effectiveDates>
<effectiveStartDate>&STARTDT</effectiveStartDate>
<effectiveEndDate>&ENDDT</effectiveEndDate>
  </effectiveDates>
</getReportingGeographiesBySamplingServiceCollectionMethodRequest>
```

## SV7QD41G

```
%LET NSIDSN = &Q7CARD..VIP&YYMM..CARD7Q;
%LET NHSIDSN = &Q7CARD..NHSI&YYMM..CARD7Q;
%LET DATACLAS = DCKFBLAR;
%LET MGMTCLAS = MCADEFLT;
%LET DISP = (REP,KEEP);
```

## SV7QD51G

HELO MOTHER

MAIL FROM:<vincent.taylor@XXXXXXX.com>  
RCPT TO:<Media-LocalProduction7QNotify@XXXXXXX.com>  
DATA  
FROM: Nokomis Processing Center - VIPC7SAS  
TO: Media - Local Production 7Q Notify  
SUBJECT: &CYCLE - LPM MU's Deleted

**SV7QD52**

HELO MOTHER  
MAIL FROM:<vincent.taylor@nielsen.com>a.com>  
RCPT TO:<Media - LocalProduction7QNotify@nielsen.com>om>  
DATA  
FROM: CIPO VIP Processors - VIPC7SASPC7SAS  
TO: Media - Local Production 7Q Notify  
SUBJECT: &CYCLE - LPM MU's Deleted

**SV7QD53G**

HELO MOTHER  
MAIL FROM:<peter.bauck@nielsen.com>  
RCPT TO:<Media-LocalProduction7QNotify@nielsen.com>  
DATA  
FROM: CIPO\_VIP\_Processors-VIPC7SAS  
TO: Media-LocalProduction7QNotify  
SUBJECT: &CYCLE - LPM MU's Deleted