import java.util.Calendar;

import java.util.Date;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import com.eviware.soapui.support.XmlHolder;

import org.apache.commons.lang.StringUtils;

import com.jcraft.jsch.\*;

public String getJulianDateForCurrentDate(){

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMdd");

Date date = dateFormat.parse(dateFormat.format(new Date()));

StringBuilder sb = new StringBuilder();

Calendar cal = Calendar.getInstance();

cal.setTime(date);

return sb.append(cal.get(Calendar.YEAR)% 100).append(String.format("%03d", cal.get(Calendar.DAY\_OF\_YEAR))).toString();

}

public String getCurrentDateForCurrentDate(){

SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMdd");

String date = dateFormat.format(new Date());

return date;

}

public String getCurrentTimeForCurrentDate(){

SimpleDateFormat dateFormat = new SimpleDateFormat("HHmmSS");

String date = dateFormat.format(new Date());

return date;

}

public void createInputFileDirectory(){

// CREATE INPUT FILES FOLDER IF DOES NOT EXISTS

String ProjectPath = context.expand('${projectDir}')

File inputFileDirectory = new File(ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local}'));

if (! inputFileDirectory.exists()){

inputFileDirectory.mkdir();

log.info("Input file Directory Created..!!!");

}else{

log.info("Input file Directory already Exists..!!!");

}

}

public void generateInputFile(){

def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)

BufferedWriter myWriter1;

StringBuilder builder1;

BufferedWriter myWriter2;

StringBuilder builder2;

createInputFileDirectory();

String ProjectPath = context.expand('${projectDir}')

def outputFilePath1 = ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local}') + "/LLA-"+context.expand('${#Project#AriaClientNo}')+"-"+getCurrentDateForCurrentDate()+"-"+getCurrentTimeForCurrentDate()+".csv";

def outputFilePath2 = ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local1}') + "/AT&T-"+context.expand('${#Project#AriaClientNo}')+"-"+getCurrentDateForCurrentDate()+"-"+getCurrentTimeForCurrentDate()+".csv";

log.info("Input File Path : " + outputFilePath1)

log.info("Input File Path : " + outputFilePath2)

testRunner.testCase.testSuite.setPropertyValue("InputFilePath\_Local", outputFilePath1)

testRunner.testCase.testSuite.setPropertyValue("InputFilePath\_Local", outputFilePath2)

myWriter1 = new BufferedWriter(new FileWriter(outputFilePath1, true))

myWriter2 = new BufferedWriter(new FileWriter(outputFilePath2, true))

builder1 = new StringBuilder()

builder2 = new StringBuilder()

def header = "";

def newLineText = "";

def AriaAcctId;

def AriaATTAcctNo;

def AriaStatementNo;

def AriaInvoiceNo;

def AriaInvoiceAmt;

def AriaPaymentAmt;

def AriaPaymentDate;

def header1 = "";

def newLineText1= "";

def ATTAcctNo;

def ATTInvoiceNo;

def ATTInvoiceAmt;

def ATTPaymentAmt;

def ATTPaymentDate;

for(int i=1; i<=4; i++){

testRunner.runTestStepByName("DS\_Account")

testRunner.runTestStepByName("create\_acct\_complete\_m")

testRunner.runTestStepByName("create\_order\_m")

testRunner.runTestStepByName("apply\_cash\_credit\_m")

testRunner.runTestStepByName("get\_acct\_statement\_history\_m")

header = "AriaAcctId"+","+ "AriaAT&TAcctNo"+","+"AriaStatementNo"+","+"AriaInvoiceNo"+","+"AriaInvoiceAmt"+","+"AriaPaymentAmt"+","+"AriaPaymentDate"

if(i==1){

myWriter1.write(header)

}

AriaAcctId = context.expand( '${create\_acct\_complete\_m#Response#//\*:out\_acct[1]/\*:client\_acct\_id[1]/text()}')

AriaATTAcctNo = context.expand( '${create\_acct\_complete\_m#Response#//\*:out\_acct[1]/\*:acct\_no[1]/text()}' )

AriaStatementNo = context.expand( '${get\_acct\_statement\_history\_m#Response#//\*:statements\_history[1]/\*:statement\_no[1]/text()}' )

AriaInvoiceNo = context.expand( '${create\_order\_m#Response#//\*:invoice\_no[1]}' )

AriaInvoiceAmt = context.expand( '${create\_order\_m#Response#//\*:total\_charges\_after\_tax[1]}' )

AriaPaymentAmt= context.expand( '${apply\_cash\_credit\_m#Response#//\*:amount[1]}' )

AriaPaymentDate = context.expand( '${apply\_cash\_credit\_m#Response#//\*:created\_date[1]}' )

for(int invoiceCount=1;invoiceCount<=1;invoiceCount++){

newLineText = AriaAcctId+","+AriaATTAcctNo+","+AriaStatementNo+","+AriaInvoiceNo+","+AriaInvoiceAmt+","+AriaPaymentAmt+","+AriaPaymentDate

builder1.append("\n")

builder1.append(newLineText)

myWriter1.newLine()

myWriter1.write(newLineText)

}

header1 = "AT&TAcctNo"+","+ "AT&TInvoiceNo"+","+"AT&TInvoiceAmt"+","+"AT&TPaymentAmt"+","+"AT&TPaymentDate"

if(i==1){

myWriter2.write(header1)

}

ATTAcctNo = context.expand( '${#TestCase#AcctNo}')

ATTInvoiceNo = context.expand( '${DS\_Account#AT&T\_InvoiceNo}' )

ATTInvoiceAmt = context.expand( '${#TestCase#InvAmt}' )

ATTPaymentAmt = context.expand( '${#TestCase#Payment\_Amt}')

ATTPaymentDate = context.expand( '${DS\_Account#AT&T\_Payment date}' )

for(int invoiceCount=1;invoiceCount<=1;invoiceCount++){

newLineText1 = ATTAcctNo+","+ATTInvoiceNo+","+ATTInvoiceAmt+","+ATTPaymentAmt+","+ATTPaymentDate

builder2.append("\n")

builder2.append(newLineText1)

myWriter2.newLine()

myWriter2.write(newLineText1)

}

}

log.info("File generated Successfully !!!");

myWriter1.flush();

myWriter1.close();

myWriter2.flush();

myWriter2.close();

}

// GENENRATE INPUT FILE

generateInputFile();

//Upload file to SFTP

//uploadFileToFTP();

# Script to run an XML Master file (combine all the Testcases in the Project)

import java.text.ParseException;

import java.text.SimpleDateFormat;

import com.eviware.soapui.support.XmlHolder;

import org.apache.commons.lang.StringUtils;

import com.jcraft.jsch.\*;

public String getCurrentDate(){

    SimpleDateFormat dateFormat = new SimpleDateFormat("yyyyMMddhhmmss");

    String date = dateFormat.format(new Date());

    return date;

}

public void createInputFileDirectory(){

    // CREATE INPUT FILES FOLDER IF DOES NOT EXISTS

    String ProjectPath = context.expand('${projectDir}')

    File inputFileDirectory = new File(ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local}'));

    if (! inputFileDirectory.exists()){

       inputFileDirectory.mkdir();

       log.info("Input file Directory Created..!!!");

    }else{

     log.info("Input file Directory already Exists..!!!");

    }

    }

public void generateInputFile(){

    def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)

    BufferedWriter myWriter1;

    StringBuilder builder1;

    createInputFileDirectory();

    String ProjectPath = context.expand('${projectDir}')

    String statement\_content\_1;

    String statement\_content\_2;

    String statement\_content\_3;

//def acctno = context.expand( '${create\_acct\_complete\_m\_Acct\_1#Response#//\*:acct\_no[1]}' )

    def outputFilePath1 = ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local}') + "/Statements\_"+getCurrentDate()+".txt";

    log.info ("ProjectPath : " + ProjectPath)

    log.info("Input File Path1 : " + outputFilePath1)

    myWriter1 = new BufferedWriter(new FileWriter(outputFilePath1, true))

    builder1 = new StringBuilder()

    def header = "";

    def newLineText = "";

    def newLineText1 = "";

    def maxNumberOfIterationsPerGroup = 12;

    def numberOfGroups = 1;

    def currentAccountNumber;

    def currentStatementNumber;

    def currentTestName;

    def currentStatementNumber2;

    def secondStatment = false;

    header = "<file><file\_summary><run\_date> 11-APR-23 </run\_date><record\_count> 40 </record\_count><client> LLA\_Dev </client></file\_summary>"

    builder1.append(header)

    myWriter1.write(header)

    // Loop until total groups

    for(int  cntNumberOfGroups = 1; cntNumberOfGroups <= numberOfGroups; cntNumberOfGroups++){

        // Loop through each group X times.

        for(int i = 1; i <= maxNumberOfIterationsPerGroup; i++){

            currentTestName = "AcctCreation\_Group\_" +i

            log.info("currentTestName : " + currentTestName)

            // Run each test case.

            context.getTestCase().getTestSuite().getTestCaseByName(currentTestName).run(new com.eviware.soapui.support.types.StringToObjectMap(), false);             // Store Account number and statement number in a test suite variable

  currentAccountNumber = testRunner.testCase.testSuite.getPropertyValue('CurrentAccountNumber')

            log.info "currentAccountNumber : " + currentAccountNumber

            currentStatementNumber = testRunner.testCase.testSuite.getPropertyValue("CurrentStatementNumber")

            log.info "currentStatementNumber : " + currentStatementNumber

            // Fetch the statement content for the given statement

            testRunner.runTestStepByName("get\_statement\_content\_m")

statement\_content\_1= context.expand('${get\_statement\_content\_m#Response#//\*:statement\_content[1]}').replaceAll('\\<\\?xml version=\"1.0\" encoding=\"UTF-8\"\\?\\>','')             // Add the statement content to the ouput XML.

            currentStatementNumber2 = testRunner.testCase.testSuite.getPropertyValue("CurrentStatementNumber2")

            log.info "currentStatementNumber2 : " + currentStatementNumber2;

            if(currentStatementNumber2 != 0 || currentStatementNumber2 != null){

                testRunner.testCase.setPropertyValue("Account", currentAccountNumber)

                testRunner.testCase.setPropertyValue("Statement", currentStatementNumber2)

                testRunner.runTestStepByName("get\_statement\_content\_m")

statement\_content\_2=context.expand('${get\_statement\_content\_m 2#Response#//\*:statement\_content[1]}').replaceAll('\\<\\?xml version=\"1.0\" encoding=\"UTF-8\"\\?\\>','')

            }

        }

        newLineText = statement\_content\_1+statement\_content\_2

        builder1.append("\n")

        builder1.append(newLineText)

        myWriter1.newLine()

        myWriter1.write(newLineText)

    }

    newLineText1 = "</file>"

    builder1.append("\n")

    builder1.append(newLineText1)

    myWriter1.newLine()

    myWriter1.write(newLineText1)

    log.info("File generated Successfully !!!");

    myWriter1.flush();

    myWriter1.close();

}

// GENENRATE INPUT FILE

generateInputFile();

# Script for generating XML statement into txt file - Generating Input file

import java.text.ParseException;

import java.text.SimpleDateFormat;

import com.eviware.soapui.support.XmlHolder;

import org.apache.commons.lang.StringUtils;

import com.jcraft.jsch.\*;

public void createInputFileDirectory(){

// CREATE INPUT FILES FOLDER IF DOES NOT EXISTS

String ProjectPath = context.expand('${projectDir}')

File inputFileDirectory = new File(ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local}'));

if (! inputFileDirectory.exists()){

inputFileDirectory.mkdir();

log.info("Input file Directory Created..!!!");

}else{

log.info("Input file Directory already Exists..!!!");

}

}

public void generateInputFile(){

def groovyUtils = new com.eviware.soapui.support.GroovyUtils(context)

BufferedWriter myWriter1;

StringBuilder builder1;

createInputFileDirectory();

String ProjectPath = context.expand('${projectDir}')

//def acctno = context.expand( '${create\_acct\_complete\_m#Response#//\*:acct\_no[1]}' )

def outputFilePath1 = ProjectPath + "/" + context.expand('${#TestSuite#InputFileFolder\_Local}') + "/Statements-"+context.expand('${create\_acct\_complete\_m#Response#//\*:acct\_no}')+".txt";

log.info ("ProjectPath : " + ProjectPath)

log.info("Input File Path1 : " + outputFilePath1)

myWriter1 = new BufferedWriter(new FileWriter(outputFilePath1, true))

builder1 = new StringBuilder()

def header = "";

def newLineText = "";

def newLineText1 = "";

header = "<file><file\_summary><run\_date> 11-APR-23 </run\_date><record\_count> 1 </record\_count><client> LLA\_Dev </client></file\_summary>"

myWriter1.write(header)

newLineText = context.expand('${get\_statement\_content\_m#Response#//\*:statement\_content[1]}').replaceAll('\\<\\?xml version=\"1.0\" encoding=\"UTF-8\"\\?\\>','')

builder1.append("\n")

builder1.append(newLineText)

myWriter1.newLine()

myWriter1.write(newLineText)

newLineText1 = "</file>"

builder1.append("\n")

builder1.append(newLineText1)

myWriter1.newLine()

myWriter1.write(newLineText1)

log.info("File generated Successfully !!!");

myWriter1.flush();

myWriter1.close();

}

// GENENRATE INPUT FILE

generateInputFile();