**turnaround1.**

**Design note**

Turnaround XREF lookup to find

* item.item.internalID
* LookupId: Receiver (Receiver being DC4 Company Name of OSR)
* LookupKey: PurchaseOrderNumber (read.tranId.data)
* Lookupkey2: read.itemList.item.vendorName.data
* Lookupkey3: BuyerPartNumber (read.itemList.item.itemName)
* LookupKey4: LineSequenceNumber (read.itemList.item.line.data)
* LookupValue: item.internalId

Help text :

read.itemList.item.itemRep.vendorName

LookupId: RaviTSender

LookupId2:RaviTReceiver

LookupKey: PurchaseOrderNumber (read.tranId.data)

LookupKey2: read.itemList.item.itemRep.vendorName.data

LookupKey3: BuyerPartNumber (read.itemList.item.itemName)

LookupKey4: LineSequenceNumber (read.itemList.item.line.data)

LookupValue: item.internalId

**Code for inserting lookup :**

**In environment of root :**

private void turnaroundInsert(String lookupValue, String lookupKey, String lookupKey2, String lookupKey3, String lookupKey4){

JString myVal = new JString();

myVal.initialize();

this.add(myVal);

myVal.setString(lookupValue);

myVal.setProperty("turnaroundLookup","Y");

myVal.setProperty("insert", "Y");

myVal.setProperty("lookupId", \_root().getProperty("sender"));

myVal.setProperty("lookupId2", \_root().getProperty("receiver"));

myVal.setProperty("lookupKey", lookupKey);

myVal.setProperty("lookupKey2", lookupKey2);

myVal.setProperty("lookupKey3", lookupKey3);

myVal.setProperty("lookupKey4", lookupKey4);

try {

Lookup lookupImpl = Lookup.getInstance(myVal);

lookupImpl.set(myVal, lookupImpl.lookup(myVal));

}

catch (Throwable myExp){

logger.info(String.valueOf(myExp));

}

this.delete(myVal);

}

// for fetching In root environment   
private String turnaroundLookup(String lookupKey, String lookupKey2, String lookupKey3, String lookupKey4){  
 String returnValue = "";  
 JString myVal = new JString();  
 myVal.initialize();  
 myVal.setJavaName("myVal");  
 myVal.setLabel("MyVal");  
 this.add(myVal);  
 myVal.setProperty("turnaroundLookup", "Y");  
 myVal.setProperty("lookupKey", lookupKey);  
 myVal.setProperty("lookupId", \_root().getProperty("sender"));  
 myVal.setProperty("lookupId2", \_root().getProperty("receiver"));  
 myVal.setProperty("lookupKey2", lookupKey2);  
 myVal.setProperty("lookupKey3", lookupKey3);  
 myVal.setProperty("lookupKey4", lookupKey4);  
  
 try {  
 Lookup lookupImpl = Lookup.getInstance(myVal);  
 lookupImpl.set(myVal, lookupImpl.lookup(myVal));  
 }  
 catch (Throwable myExp){  
 logger.info(String.valueOf(myExp));  
 }  
 returnValue = myVal.getString();  
 this.delete(myVal);  
 return returnValue;  
}

**Code to insert validation and init for inserting turnaround**

String internalId = \_source().item\_1.internalId.getString();

if(Str.HasData(internalId)) {

root.turnaroundInsert(

internalId,

\_root().header.orderHeader.purchaseOrderNumber.getString(),

"read.itemList.item.vendorName.data",

orderLine.buyerPartNumber.getString(),

orderLine.lineSequenceNumber.getString());

//code for fectching lookup :

item\_3.internalId.setString(root.turnaroundLookup(\_root().\_source().header.orderHeader.purchaseOrderNumber.getString(),

"read.itemList.item.itemRep.vendorName.data",

\_source().orderLine.buyerPartNumber.getString(),

\_source().orderLine.lineSequenceNumber.getString()));

**2. Source first ten characters as YYYY-MM-DD.**

**String date = me.getString();**

**if(date.length() > 10){**

**me.setString(date.substring(0,10));**

**}**

**root.formatDate(me, "yyyy-MM-dd", "yyyy-MM-dd");**

**// in root environment   
 private void formatDate(JString field, String readFormat, String writeFormat) {**

**String date = field.getString();**

**if (Str.HasData(date)) {**

**SimpleDateFormat dfRead = new SimpleDateFormat(readFormat);**

**SimpleDateFormat dfWrite = new SimpleDateFormat(writeFormat);**

**dfWrite.setLenient(false);**

**try {**

**field.setString(dfWrite.format(dfRead.parse(date)));**

**} catch (Exception e) {**

**logger.info("Unable to format the date \"" + date + "\" in " +**

**field.getFullyQualifiedJavaName().replaceAll("\\.[a-zA-Z]+Rep(?a-zA-Z])", "") +**

**": " + e.getMessage());**

**}**

**}**

**}**

**3.** **Convert value.internalId to RSX value:**

**1 > 05**

**2 > 01**

**Set="00" if customField not present.**

**//code in environment**

**/\*\***

**\* A class to hold a regex and a convertTo value pair.**

**\*/**

**private static final class FindReplacePair{**

**private String regex;**

**private String convertTo;**

**public FindReplacePair(String find, String replace){**

**this.regex = find;**

**this.convertTo = replace;**

**}**

**}**

**// code in environment**

**/\*\***

**\* A method to help make converting values easier.**

**\* <br><b>Required classes:</b>**

**\* <ul>**

**\* <li>FindReplacePair()</li>**

**\* </ul>**

**\* @param field The JElement that you want converted.**

**\* @param pairs Any number of FindReplacePairs that represent your conversions.**

**\*/**

**private void convertValues(JElement field, FindReplacePair... pairs) {**

**String code = field.getString();**

**for(FindReplacePair pair : pairs) {**

**if(code.matches(pair.regex)) {**

**field.setString(pair.convertTo);**

**return;**

**}**

**}**

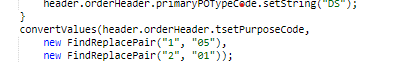
**}**

**// code**

**convertValues(header.orderHeader.tsetPurposeCode,**

**new FindReplacePair("1", "05"),**

**new FindReplacePair("2", "01"));**

****

**4.FOR checking TP id**

**//code in root validations(document finalize)  
checkTpid(write.header.orderHeader.tradingPartnerId);**

**//code in root environment**

**/\*\***

**\* Checks to see if a lookup was found for the given Element. Should only be used for Vendor Xrefs**

**\* Will fail the map if the vendor xref was not found.**

**\* @param tpid This will typically be the TradingPartnerId field on the RSX side of the map.**

**\*/**

**private void checkTpid(JElement tpid) {**

**if(!tpid.hasData()){**

**tpid.reportError(tpid, "TradingPartnerId lookup failed for Lookup\_key (Old Value): " + tpid.getPropertyString("lookupKey") + " , Lookup\_id (Sender): " + \_root().getPropertyString("sender"));**

**}**

**}**

**5. replaceAll method**

**createdFrom.name.data.setString(createdFrom.name.data.getString().replaceAll("(?i)sales order #", ""))**

**6.Deactivate /activate code**

**//activateAll() in pre-validation and deactivateAll() in validation**

**// code in root environment**

**/\*\***

**\* Goes through the whole map and activates all JOptionalGroups. Must be called on PreValidation.**

**\* @param ent Any JEntity, should probably be the root.**

**\*/**

**private void activateAll(Entity entity){**

**for (int i = 0; i < entity.getTemplateChildrenCount(); i++) {**

**if(!(entity.getTemplateChild(i) instanceof JElement)) {**

**activateAll(entity.getTemplateChild(i));**

**}**

**}**

**if (entity instanceof JOptionalGroup) {**

**((JOptionalGroup)entity).setActivated(true);**

**}**

**}**

**/\*\***

**\* Recursively goes through the map and turns off any JOptionalGroups that don't have data.**

**\* Must be called on DocFinalize.**

**\* @param jEntity Any JEntity, should probably be the root.**

**\*/**

**private boolean deactivateOptional(JEntity jEntity){**

**boolean hasChildData = false;**

**if (jEntity instanceof JOptionalGroup){**

**for (int i=0; i<jEntity.getChildrenCount(); i++){**

**JEntity child = (JEntity)jEntity.getJChild(i);**

**if (child != null && deactivateOptional(child)){**

**hasChildData = true;**

**}**

**}**

**if (!hasChildData){**

**((JOptionalGroup)jEntity).setActivated(false);**

**}**

**return hasChildData;**

**} else if (!(jEntity instanceof JElement)){**

**for (int i=0; i<jEntity.getChildrenCount(); i++){**

**JEntity child = (JEntity)jEntity.getJChild(i);**

**if (child != null && deactivateOptional(child)){**

**hasChildData = true;**

**}**

**}**

**return hasChildData;**

**} else if (jEntity instanceof JElement){**

**if (jEntity != null && jEntity.getProperty("defaultValue") == null && ((JElement)jEntity).hasData()){**

**return true;**

**}**

**}**

**return hasChildData;**

**}**

**7.** //**Order/LineItems/LineItem/ChargesAllowances/AllowChrgAmt**

**//(When AllowChrgIndicator = 'A' source as negative value.)**

**if(\_source().allowChrgIndicator.getString().equals("A")){**

**if(\_source().allowChrgAmt.hasData()){**

**amount.data.setString("-" + amount.data.getString());**

**}**

**else if(\_source().allowChrgPercent.hasData()){**

**calculation.rateNumeric.data.setString("-" + calculation.rateNumeric.data.getString());**

**}**

**}**

**8. Condition If one field is present among all given then all are required**

**If ShipmentWeight, ShipmentWeightUnitCode1, or ShipmentWeightQualifier1 are present, then all are required**

**// codes**

**checkCondition(Condition.PAIRED, "shipmentWeight,shipmentWeightUnitCode1,shipmentWeightQualifier1");**

**9. mandatoryCheck method   
//code to put in environment**

**private void mandatoryCheck(JEntity srcGroup,String javaName,String value){**

**if(srcGroup instanceof JRepeatableGroup){**

**if(!((JRepeatableGroup)srcGroup).hasRepetition(javaName,value)){**

**srcGroup.reportError(srcGroup,srcGroup.getFullyQualifiedJavaName().replaceAll("\\.[a-zA-Z]+Rep(?![a-zA-Z])", "") + " is missing the " + value + " repetition");**

**}**

**}**

**}**

**private void mandatoryCheck(JEntity en,String extraMsg){**

**boolean hasData=false;**

**if(!(en instanceof JOptionalGroup)){**

**hasData=hasChildData(en);**

**}else{**

**if(((JOptionalGroup)en).isActivated() == true){**

**hasData = hasChildData(en);**

**}**

**}**

**if(hasData == false){**

**en.reportError(en, "Missing mandatory data: " + makeXPath(en) + "; " +extraMsg);**

**}**

**}**

**private boolean hasChildData(JEntity en){**

**if(!(en instanceof JElement)){**

**for(int i=0; i<en.getChildrenCount(); i++){**

**if(hasChildData(en.getJChild(i)) == true){**

**return true;**

**}**

**}**

**}else{**

**return ((JElement)en).hasData();**

**}**

**return false;**

**}**

**private String makeXPath(JEntity en){**

**String path = en.getFullyQualifiedJavaName().replaceAll("\\.[a-zA-Z]+Rep(?![a-zA-Z])", "");**

**String[] parts = path.split("\\.");**

**String xpath = "/" + en.getJRoot().getPropertyString("name");**

**for (int i = 1; i < parts.length; i++) {**

**String part = parts[i];//ignore the root name**

**part = Character.toUpperCase(part.charAt(0)) + part.substring(1);**

**xpath += "/" + part;**

**}**

**return xpath;**

**}**

**//code in validation**

**mandatoryCheck(header.statusDetail.vesselName," vesselName required ");**

**10. When we have to write code for input in output**

**if(\_source().header.orderHeader.tsetPurposeCode.getString().matches("00")){**

**dataArea.sync.actionCriteria.actionExpression.actionCode.setString("Add");**

**}else{**

**dataArea.sync.actionCriteria.actionExpression.actionCode.setString("Update");**

**}**

**if(\_source().header.orderHeader.primaryPOTypeCode.getString().matches("BK")){**

**dataArea.sync.actionCriteria.actionExpression.actionCode.setString("Update");**

**}**

**12. Shipment code hl levels**

**// in root environment**

**String PONum="";**

**int itemCount = 0;**

**/\* Will populate all HLReps with the proper HL01 and HL02 values, ignoring the HL02 for the shipment HL**

**\* Required Methods:**

**\* shouldWrite()**

**\* @param entity the current JEntity**

**\* @param value the current value to use for the HL01 (when called from the root, use 0)**

**\*/**

**private int hlCount = 0;**

**private void setFEDSFields(JEntity entity, int parentHLCount) {**

**boolean foundField = false;**

**if(entity.getJChild("hL01") != null) {**

**foundField = true;**

**((JElement)entity.getJChild("hL01")).setString(String.valueOf(++\_root().hlCount));**

**}**

**if(entity.getJChild("hL02") != null) {**

**foundField = true;**

**((JElement)entity.getJChild("hL02")).setString(String.valueOf(parentHLCount));**

**//don't set HL02 for S loops**

**if(entity.getString("hL03") != null && entity.getString("hL03").equals("S")) {**

**((JElement)entity.getJChild("hL02")).setString("");**

**}**

**}**

**//always continue for the root, or repeatableGroups. Otherwise, only if we found at least 1 field.**

**if(entity instanceof JDocument || entity instanceof JRepeatableGroup || foundField) {**

**for(int i=0; i < entity.getChildrenCount(); i++) {**

**if(!(entity.getJChild(i) instanceof JElement)) {**

**//make sure this group is set to get written out.**

**if(shouldWrite(entity.getJChild(i))) {**

**//all repetitions share the same parent, so don't add to the count**

**if(entity.getJChild(i) instanceof JRepetition) {**

**setFEDSFields(entity.getJChild(i), parentHLCount);**

**}else {**

**//check the entity's HL01 value, if found, use for the parentHLCount for next call.**

**if(entity.getString("hL01") != null) {**

**setFEDSFields(entity.getJChild(i), Integer.parseInt(entity.getString("hL01")));**

**}else {**

**setFEDSFields(entity.getJChild(i), parentHLCount + 1);**

**}**

**}**

**}**

**}**

**}**

**}**

**}**

**/\*\***

**\* Check the dataType of the given entity, and return true if this will be written out**

**\* All JGroups, activated JOptionalGroups, and JRepeatableGroup with reps should return true.**

**\* @param entity the JEntity to check**

**\*/**

**private boolean shouldWrite(JEntity entity) {**

**if(entity instanceof JOptionalGroup) {**

**return ((JOptionalGroup)entity).isActivated();**

**}else if(entity instanceof JRepeatableGroup) {**

**return (entity.getChildrenCount() > 0);**

**}else if(entity instanceof JGroup || entity instanceof JRepetition) {**

**return true;**

**}**

**return false;**

**}**

**// in validation doc fin  
setFEDSFields(me, 0);**

**13. Mandatory rep and mandatorycheck code :**

**//In root environment**

/\*\*

\* Checks to see if the given JRepeatableGroup has a specific repetition, errors if not found.

\* <br><b>Requires the following Methods:</b>

\* <ul>

\* <li>makeXPath()</li>

\* </ul>

\* @param group The JRepeatableGroup you would like to check.

\* @param field javaName of the field you want to check

\* @param value value of field you want to check for.

\* @param extraMsg Any extra text you would like to add to the error message.

\*/

private void mandatoryRep(JRepeatableGroup group, String field, String value, String extraMsg){

if(!group.hasRepetition(field, value)) {

group.reportError(group, "Missing mandatory data: " + makeXPath(group) + " with " +

field + " = " + value + " is required;" + extraMsg);

}

}

private void mandatoryCheck(JEntity srcGroup, String javaName, String value) {

if (srcGroup instanceof JRepeatableGroup) {

if (!((JRepeatableGroup)srcGroup).hasRepetition(javaName, value)) {

srcGroup.reportError(srcGroup, srcGroup.getFullyQualifiedJavaName().replaceAll("\\.[a-zA-Z]+Rep(?![a-zA-Z])", "") + " is missing the " + value + " repetition");

}

}

}

private void mandatoryCheck(JEntity en, String extraMsg){

boolean hasData = false;

if(!(en instanceof JOptionalGroup)){

hasData = hasChildData(en);

}else{//this is an optional group

//no need to loop through non activated optional groups

if(((JOptionalGroup)en).isActivated() == true){

hasData = hasChildData(en);

}

}

if(hasData == false){

en.reportError(en, "Missing mandatory data: " + makeXPath(en) + "; " +extraMsg);

}

}

private boolean hasChildData(JEntity en){

if(!(en instanceof JElement)){

for(int i=0; i<en.getChildrenCount(); i++){

//if a child has data, then i have data

if(hasChildData(en.getJChild(i)) == true){

return true;

}

}

}else{//this is an element

return ((JElement)en).hasData();

}

return false;//to make compiler happy

}

private String makeXPath(JEntity en){

String path = en.getFullyQualifiedJavaName().replaceAll("\\.[a-zA-Z]+Rep(?![a-zA-Z])", "");

String[] parts = path.split("\\.");

String xpath = "/" + en.getJRoot().getPropertyString("name");

for (int i = 1; i < parts.length; i++) {

String part = parts[i];//ignore the root name

part = Character.toUpperCase(part.charAt(0)) + part.substring(1);

xpath += "/" + part;

}

return xpath;

}

**mandatoryCheck(header.shipmentHeader,"shipDate");**

**mandatoryRep(header.address,"addressTypeCode","ST","");**

**mandatoryRep(header.address,"addressTypeCode","SF","");**

**14.Set header for csv with code**

**// in root environment**

**/\*\***

**\* Sets the column headers for a CSV header group. These will be set as the name of each field.**

**\* @param jEntity The header group that will be printed before each item.**

**\*/**

**private void setHeader(JEntity jEntity) {**

**if (jEntity instanceof JElement) {**

**((JElement) jEntity).setString(jEntity.getLabel());**

**} else {**

**for (int i = 0; i < jEntity.getChildrenCount(); i++) {**

**setHeader(jEntity.getJChild(i));**

**}**

**}**

**}**

**// in doc fin  
setHeader(header);**

**15:quote wrap when only specific fields are required under quote rather than whole**

**//in root environment**

**//Quote wrap field, escapes double quotes if there are any**

**private void quoteWrapFields(JElement... fields){**

**for(JElement field : fields){**

**if(!field.getString().contains("\"")){**

**field.setString("\"" + field.getString() + "\"");**

**} else {**

**field.setString("\"" + field.getString().replaceAll("\"", "\"\"") + "\"");**

**}**

**}**

**}**

**//in linerep (write side)**

**public void documentFinalize(ValidationEvent e) {**

**root.quoteWrapFields(carrierInfo,**

**shipToName,**

**shipToCompanyName,**

**shipToStreet1,**

**shipToStreet2,**

**shipFromName,**

**shipFromCompanyName,**

**shipFromStreet1,**

**shipFromStreet2,**

**itemDescription);**

**}//end-method**

**16.**   
//**If weight is present then validate WeightQualifier=G**

**// code**

**if(weight.hasData() && (!weightQualifier.getString().equals("G") || !weightQualifier.hasData())){**

**reportError("WeightQualifier=G is required when Weight is present");**

**}**

**17. when there is only field in the read side (but it is repeated in input ) and we have to source that to many outputs based on field repetition**

**//in read pack level environment   
ArrayList<String> marksAndNumList = new ArrayList<String>();**

**//in read pack level Rep doc fin**

**for (int i = 0; i < marksAndNumbersCollection.getChildrenCount(); i++){**

**if (marksAndNumbersCollection.\_getRep(i).marksAndNumbersQualifier1.getString().equals("CA") &&**

**marksAndNumbersCollection.\_getRep(i).marksAndNumbers1.hasData()){**

**marksAndNumList.add(marksAndNumbersCollection.\_getRep(i).marksAndNumbers1.getString());**

**}**

**}**

**//IN write side item level rep environment**

**int trackNumCount = 2;**

**// in write side item level rep doc fin  
for (String trackNum : \_source().\_parent().\_parent().marksAndNumList){**

**if (trackNumCount < 6){**

**((JElement)getJChild("trackingNumber" + trackNumCount++)).setString(trackNum);**

**}**

**}**

**18. To set time   
//on that field doc fin  
DateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");**

**Calendar cal = Calendar.getInstance();**

**me.setString(dateFormat.format(cal.getTime()));  
  
//in root environment   
/\*\***

**\* Grab current date in the given Date format**

**\* @Param format the format you would like to get the date in.**

**\*/**

**private String getCurrentDate(String format) {**

**Date curDate = new Date();**

**DateFormat todayFormat = new SimpleDateFormat(format);**

**String todayDate = todayFormat.format(curDate);**

**return todayDate;**

**}  
  
19.** **SPS: do not output if value is 0  
  
 if(me.getString()){  
 me.setString(“”)  
 }  
  
  
20.** **SPS: If AddressLocationNumber is NUMERIC and 13 digits**

**THEN set LocationCodeQualifier = 'UL' ELSE set LocationCodeQualifier = '92'  
  
if(addressLocationNumber.getString().matches("\\d+") && addressLocationNumber.getString().length() == 13){**

**locationCodeQualifier.setString("UL");**

**}else{**

**locationCodeQualifier.setString("92");**

**}  
  
  
21.** **Each AllowChrgCode can only be sent a max of 1 time.  
(only one value can be sent of one field )**

**//root env  
ArrayList<String> addrCodeList=new ArrayList<String>();**

**//doc fin in root or respective parent group**

**for(int i=0; i<header.chargesAllowances.getChildrenCount(); i++) {**

**String addrCode = header.chargesAllowances.\_getRep(i).allowChrgCode.getString();**

**if (!addrCodeList.contains(addrCode)) {**

**addrCodeList.add(addrCode);**

**}**

**else {**

**reportError("Each AllowChrgCode can only be sent a max of 1 time.");**

**}**

**}**

**22.** **SPS to validate that TotalAmount is the sum of all items (Price \*Qty) +/- any allowances and charges**

**//in root environment   
String totalAmt = "0.0";**

**public String add(String total, String add){**

**if(NumberUtilities.IsFullyNumeric(add)){**

**return NumberUtilities.Calc.Add(total,add);**

**} else {**

**return total;**

**}**

**}  
  
//root doc fin  
 //go through LineItems to gather amounts**

**for(int detailIndex = 0; detailIndex < lineItems.lineItem.getChildrenCount(); detailIndex++) {**

**lineItems.lineItem.lineItemRep item = lineItems.\_lineItem(detailIndex);**

**String qty = item.invoiceLine.shipQty.getString();**

**String price = item.invoiceLine.purchasePrice.getString();**

**if (!qty.matches( "-?\\d+(\\.\\d+)?") || !price.matches( "-?\\d+(\\.\\d+)?")) return;**

**String detailTotal = NumberUtilities.Calc.Multiply(price,qty);**

**totalAmt = NumberUtilities.Calc.Add(detailTotal,totalAmt);**

**//gather lineItem/chargesAllowances**

**for(int sacIndex = 0; sacIndex < item.chargesAllowances.getChildrenCount(); sacIndex++) {**

**lineItems.lineItem.lineItemRep.chargesAllowances.chargesAllowancesRep sac = item.\_chargesAllowances(sacIndex);**

**String sacAmount = sac.allowChrgAmt.getString();**

**if(NumberUtilities.IsFullyNumeric(sacAmount))**

**{**

**String indicator = sac.allowChrgIndicator.getString();**

**if(indicator.matches("A"))**

**{**

**totalAmt = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,sacAmount);**

**} else if (indicator.matches("C")){**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,sacAmount);**

**}**

**}**

**} // done with lineItem/chargesAllowances**

**} // done with lineItem**

**//gather header/chargesAllowances**

**for(int sacHIndex = 0; sacHIndex < header.chargesAllowances.getChildrenCount(); sacHIndex++) {**

**header.chargesAllowances.chargesAllowancesRep sacH = header.\_chargesAllowances(sacHIndex);**

**String sacAmount1 = sacH.allowChrgAmt.getString();**

**if(NumberUtilities.IsFullyNumeric(sacAmount1))**

**{**

**String indicator1 = sacH.allowChrgIndicator.getString();**

**if(indicator1.matches("A"))**

**{**

**totalAmt = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,sacAmount1);**

**} else if (indicator1.matches("C")){**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,sacAmount1);**

**}**

**}**

**} // done with header/chargesAllowances**

**//gather header/taxes**

**for(int taxIndex = 0; taxIndex < \_source().header.taxes.getChildrenCount(); taxIndex++) {**

**read.header.taxes.taxesRep taxRep = \_source().header.\_taxes(taxIndex);**

**String taxAmount = taxRep.taxAmount.getString();**

**if(NumberUtilities.IsFullyNumeric(taxAmount))**

**{**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,taxAmount);**

**}**

**} // done with header/taxes**

**//gather line/taxes**

**for(int lineIndex = 0; lineIndex < \_source().lineItem.getChildrenCount(); lineIndex++){**

**read.lineItem.lineItemRep lineRep = \_source().\_lineItem(lineIndex);**

**for(int lineTaxIndex = 0; lineTaxIndex < lineRep.taxes.getChildrenCount(); lineTaxIndex++){**

**read.lineItem.lineItemRep.taxes.taxesRep lineTaxRep = lineRep.\_taxes(lineTaxIndex);**

**String lineTaxAmount = lineTaxRep.taxAmount.getString();**

**if(NumberUtilities.IsFullyNumeric(lineTaxAmount))**

**{**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,lineTaxAmount);**

**}**

**}**

**}**

**String value = "0.50";**

**String Decrement = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,value);**

**String Increment = NumberUtilities.Calc.Add(totalAmt,value);**

**if(summary.totalAmount.hasData()){**

**if(summary.totalAmount.getNumber().doubleValue() < Double.parseDouble(Decrement) || summary.totalAmount.getNumber().doubleValue() > Double.parseDouble(Increment)){**

**reportError("TotalAmount is (+/- 0.50) the sum of all items (Price\*Qty) +/- any allowances and charges.");**

**}else{**

**summary.totalAmount.setNumber(\_source().summary.totalAmount.getNumber());**

**}**

**}else{**

**summary.totalAmount.setNumber(Double.parseDouble(totalAmt));**

**}**

**When only throw error instead of difference and precision**

**// 2nd code**

**//go through LineItems to gather amounts**

**for(int detailIndex = 0; detailIndex < lineItems.lineItem.getChildrenCount(); detailIndex++) {**

**lineItems.lineItem.lineItemRep item = lineItems.\_lineItem(detailIndex);**

**String qty = item.invoiceLine.shipQty.getString();**

**String price = item.invoiceLine.purchasePrice.getString();**

**if (!qty.matches( "-?\\d+(\\.\\d+)?") || !price.matches( "-?\\d+(\\.\\d+)?")) return;**

**String detailTotal = NumberUtilities.Calc.Multiply(price,qty);**

**totalAmt = NumberUtilities.Calc.Add(detailTotal,totalAmt);**

**//gather lineItem/chargesAllowances**

**for(int sacIndex = 0; sacIndex < item.chargesAllowances.getChildrenCount(); sacIndex++) {**

**lineItems.lineItem.lineItemRep.chargesAllowances.chargesAllowancesRep sac = item.\_chargesAllowances(sacIndex);**

**String sacAmount = sac.allowChrgAmt.getString();**

**if(NumberUtilities.IsFullyNumeric(sacAmount))**

**{**

**String indicator = sac.allowChrgIndicator.getString();**

**if(indicator.matches("A"))**

**{**

**totalAmt = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,sacAmount);**

**} else if (indicator.matches("C")){**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,sacAmount);**

**}**

**}**

**} // done with lineItem/chargesAllowances**

**} // done with lineItem**

**//gather header/chargesAllowances**

**for(int sacHIndex = 0; sacHIndex < header.chargesAllowances.getChildrenCount(); sacHIndex++) {**

**header.chargesAllowances.chargesAllowancesRep sacH = header.\_chargesAllowances(sacHIndex);**

**String sacAmount1 = sacH.allowChrgAmt.getString();**

**if(NumberUtilities.IsFullyNumeric(sacAmount1))**

**{**

**String indicator1 = sacH.allowChrgIndicator.getString();**

**if(indicator1.matches("A"))**

**{**

**totalAmt = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,sacAmount1);**

**} else if (indicator1.matches("C")){**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,sacAmount1);**

**}**

**}**

**} // done with header/chargesAllowances**

**//gather header/taxes**

**for(int taxIndex = 0; taxIndex < \_source().header.taxes.getChildrenCount(); taxIndex++) {**

**read.header.taxes.taxesRep taxRep = \_source().header.\_taxes(taxIndex);**

**String taxAmount = taxRep.taxAmount.getString();**

**if(NumberUtilities.IsFullyNumeric(taxAmount))**

**{**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,taxAmount);**

**}**

**} // done with header/taxes**

**//gather line/taxes**

**for(int lineIndex = 0; lineIndex < \_source().lineItem.getChildrenCount(); lineIndex++){**

**read.lineItem.lineItemRep lineRep = \_source().\_lineItem(lineIndex);**

**for(int lineTaxIndex = 0; lineTaxIndex < lineRep.taxes.getChildrenCount(); lineTaxIndex++){**

**read.lineItem.lineItemRep.taxes.taxesRep lineTaxRep = lineRep.\_taxes(lineTaxIndex);**

**String lineTaxAmount = lineTaxRep.taxAmount.getString();**

**if(NumberUtilities.IsFullyNumeric(lineTaxAmount))**

**{**

**totalAmt = NumberUtilities.Calc.Add(totalAmt,lineTaxAmount);**

**}**

**}**

**}**

**if(summary.totalAmount.hasData()){**

**double totalAmt2 = Double.parseDouble(totalAmt);**

**if(totalAmt2 == summary.totalAmount.getDouble()){**

**summary.totalAmount.setNumber(totalAmt2);**

**}else{**

**reportError("TotalAmount should be the sum of all items (Price\*Qty) +/- any allowances and charges.");**

**}**

**}**

**23.** **Convert to a real number with decimal added before the last 2 digits**

**Ex. 12345 --> 123.45**

**if(me.hasData()){**

**String disAmt = me.getString();**

**if (disAmt.length() > 2) {**

**int endPt = disAmt.length();**

**int decPt = endPt - 2;**

**me.setString(disAmt.substring(0, decPt) +**

**"." + disAmt.substring(decPt, endPt));**

**} else {**

**int count = 2 - disAmt.length();**

**for (int i = 0; i < count; i++) {**

**disAmt = "0" + disAmt;**

**}**

**me.setString("0." + disAmt);**

**}**

**}  
  
  
  
24.multiple conditional sourcing  
  
//root envi  
  
private void FedsPER(JEntity pERLoop, JEntity contact){**

**LinkedHashMap <String, String> values = new LinkedHashMap <String, String>();**

**NumberFormat formatter = new DecimalFormat("00");**

**for(int i=3 ; i<=7 ; i=i+2){**

**values.put(pERLoop.getString("pER"+formatter.format(i)), pERLoop.getString("pER"+formatter.format(i+1)));**

**}**

**for (Map.Entry <String, String> value : values.entrySet()){**

**if (value.getKey().matches("TE")){**

**((JElement)contact.getJChild("primaryPhone")).setString(value.getValue());**

**}**

**if (value.getKey().matches("EM")){**

**((JElement)contact.getJChild("primaryEmail")).setString(value.getValue());**

**}**

**if (value.getKey().matches("FX")){**

**((JElement)contact.getJChild("primaryFax")).setString(value.getValue());**

**}**

**}**

**}   
//in doc fin of group  
  
root.FedsPER(\_source(),me);  
  
  
  
  
25. Populateline method for sourcing from feds to rsx  
  
// in root environment   
//code  
/\*\***

**\* Sources a number of Loop\_PO1 fields to a LIN group from a XML Loop\_PO1 group**

**\* @param loopPO1 the Loop\_PO1 group that contains line info (orderLine, shipmentLine, etc)**

**\* @param lineItem the RSX LIN group**

**\*/**

**private void populateLine(JEntity iT1Loop1\_Group, JEntity lineItem){**

**LinkedHashMap <String, String> values = new LinkedHashMap <String, String>();**

**NumberFormat formatter = new DecimalFormat("00");**

**for(int i=6 ; i<=20 ; i=i+2){**

**values.put(iT1Loop1\_Group.getString("iT1"+formatter.format(i)), iT1Loop1\_Group.getString("iT1"+formatter.format(i+1)));**

**}**

**for (Map.Entry <String, String> value : values.entrySet()){**

**if (value.getKey().matches("BP|IN|PI|SK|IT|CB|PN|ST")){**

**((JElement)lineItem.getJChild("buyerPartNumber")).setString(value.getValue());**

**}else if (value.getKey().matches("VA|VC|VN|VP|MG|MF")){**

**((JElement)lineItem.getJChild("vendorPartNumber")).setString(value.getValue());**

**}else if (value.getKey().matches("UP|UK|UI|UN|UA|UB")){**

**((JElement)lineItem.getJChild("consumerPackageCode")).setString(value.getValue());**

**}else if (value.getKey().matches("EN")){**

**((JElement)lineItem.getJChild("eAN")).setString(value.getValue());**

**}else if (value.getKey().matches("N1|N2|N3|N4|ND")){**

**((JElement)lineItem.getJChild("natlDrugCode")).setString(value.getValue());**

**}else if (value.getKey().matches("IB")){**

**((JElement)lineItem.getJChild("internationalStandardBookNumber")).setString(value.getValue());**

**}**

**}**

**}  
  
  
//in linitemRep  
  
root.populateLine(\_source().iT1\_Group, invoiceLine);**

**26. To count lineitem  
  
 if(!summary.totalLineItemNumber.hasData()){**

**summary.setActivated(true);**

**summary.totalLineItemNumber.setNumber(lineItems.lineItem.getChildrenCount());**

**}**

**27.to set current date   
//in hardcode  
SYSDATE**

**28. Multiple structure in shipment   
//at orderlevelrep  
if(itemLevel.getChildrenCount() == 0 && packLevel.getChildrenCount() == 0){**

**reportError(packLevel,"Atleast one of SOPI/SOTI or SOI structure is required");**

**}**

**if(itemLevel.getChildrenCount() > 0 && packLevel.getChildrenCount() > 0){**

**reportError(packLevel,"Only one of SOPI/SOTI or SOI or SOIP structure can be sent");**

**}  
  
29. Check mandatory method(If either A or B is present, then both are required.)  
  
 checkMandatory(locationCodeQualifier,addressLocationNumber);**

**checkMandatory(addressLocationNumber,locationCodeQualifier);**

**30.when concatenate sourcing with adding space   
//code**

**String noteInfo = "";**

**if(\_source().header.fhi\_note\_text1.hasData()){**

**noteInfo += \_source().header.fhi\_note\_text1.getString()+" ";**

**}**

**if(\_source().header.fhi\_note\_text2.hasData()){**

**noteInfo += \_source().header.fhi\_note\_text2.getString()+" ";**

**}**

**if(\_source().header.fhi\_note\_text3.hasData()){**

**noteInfo += \_source().header.fhi\_note\_text3.getString()+" ";**

**}**

**if(\_source().header.fhi\_note\_text4.hasData()){**

**noteInfo += \_source().header.fhi\_note\_text4.getString()+" ";**

**}**

**if(\_source().header.fhi\_note\_text5.hasData()){**

**noteInfo += \_source().header.fhi\_note\_text5.getString();**

**}**

**header.notes\_group\_11.note.setString(noteInfo.trim());**

**31 Set Time  
  
//in root**

**/\*\***

**\* Grab current date in the given Date format**

**\* @Param format the format you would like to get the date in.**

**\*/**

**private String getCurrentDate(String format) {**

**Date curDate = new Date();**

**DateFormat todayFormat = new SimpleDateFormat(format);**

**String todayDate = todayFormat.format(curDate);**

**return todayDate;**

**}**

**// in doc fin of field  
 DateFormat dateFormat = new SimpleDateFormat("HH:mm:ss");**

**Calendar cal = Calendar.getInstance();**

**me.setString(dateFormat.format(cal.getTime()));  
  
32.** **: If AddressLocationNumber is NUMERIC and 13 digits**

**THEN set LocationCodeQualifier = 'UL'**

**ELSE set LocationCodeQualifier = '92'  
  
 if(addressLocationNumber.hasData()){**

**if(addressLocationNumber.getString().length() == 13){**

**locationCodeQualifier.setString("UL");**

**}else{**

**locationCodeQualifier.setString("92");**

**}**

**}**

**33.drop date if it equals 00000000  
// read side pre-validation in init**

**if(me.getString().equals("00000000")){**

**me.setString("");**

**}**

**//read side validation in doc fin  
if (me.hasData()){**

**try {**

**SimpleDateFormat myFormat = new SimpleDateFormat("yyyyMMdd");**

**myFormat.setLenient(false);//Only allow valid dates**

**Date tempDate = myFormat.parse(me.getString());**

**} catch (Exception err) {**

**reportError("Invalid date \"" + me.getString() + "\" found in "+ DGEProperties.label.get(me));**

**}**

**}**

**//write validation init  
if(me.hasData()){**

**me.setString(root.formatDate(me.getString(), "yyyyMMdd","yyyy-MM-dd"));**

**}**

**34.when we need to get decimal output**

if(me.hasData()){

me.setString(NumberUtilities.Calc.Divide(me.getString(),"100",2));

}

**35. when we have to drop group when data is not there to avoid null o/p**

public boolean verify(JEntity ent){

if(ent.getString("schedule\_Notes").length()>0){

return true;

}

return false;

}

**36.Removing prefix with ignoring Case**

//remove prefix of sales order # (?i) = ignore case.

createdFrom.name.data.setString(createdFrom.name.data.getString().replaceAll("(?i)sales order #", ""));

**37.Ghost structure example1**

**//in root doc fin**

HashMap<String, ArrayList<read.itemList.item.itemRep>> lineHash = new HashMap<String, ArrayList<read.itemList.item.itemRep>>();

for(int i=0; i < itemList.item.getChildrenCount(); i++){

read.itemList.item.itemRep lineRep = itemList.item.\_getRep(i);

String key = "";

if (assemblyItem.name.data.hasData()) {

key = assemblyItem.name.data.getString();

}

if (Str.HasData(key)) {

if(lineHash.containsKey(key)){

lineHash.get(key).add(lineRep);

} else {

ArrayList<read.itemList.item.itemRep> lineArr = new ArrayList<read.itemList.item.itemRep>();

lineArr.add(lineRep);

lineHash.put(key,lineArr);

}

}

}

for(String key: lineHash.keySet()){

read.ghostLine.ghostLineRep ghostRep = ghostLine.\_jrepeat();

oneToOneCopy(assemblyItem, ghostRep.assemblyItem);

oneToOneCopy(quantity, ghostRep.quantity);

oneToOneCopy(units, ghostRep.units);

for(read.itemList.item.itemRep line: lineHash.get(key)){

read.ghostLine.ghostLineRep.ghostSubline.ghostSublineRep ghostSubRep = ghostRep.ghostSubline.\_jrepeat();

oneToOneCopy(line,ghostSubRep);

}

}

**//In root environment**

/\*\*

\* Copies the srcEntity to the tarEntity. Will call the built in OneToOneCopy() method only when it finds children with the same javaName.

\* This allows for very flexible ghost structures.

\* @param sourceEnt The entity to copy from

\* @param targetEnt The entity to copy to

\* @param exceptionList any javaNames you do not want to copy within the first level of children.

\*/

private void oneToOneCopy(JEntity sourceEnt, JEntity targetEnt, String... exceptionList){

for (int i = 0; i < sourceEnt.getChildrenCount(); i++){

String javaName = sourceEnt.getJChild(i).getPropertyString("javaName");

JEntity targetChild = targetEnt.getJChild(javaName);

boolean skipThisChild = false;

for(String ignoreThis : exceptionList) {

if(ignoreThis.equals(javaName)) {

skipThisChild = true;

}

}

if(targetChild != null && !skipThisChild){

try{

targetChild.OneToOneCopy(sourceEnt.getJChild(i));

}catch (Exception err){

logger.info("Two entities don't match - source: " + sourceEnt.getJChild(i) + " target: " + targetEnt.getJChild(javaName).getFullyQualifiedJavaName() + " " + "\n\t" + err);

}

}

}

}

**38.To validate integer   
//in root environment**

/\*\*

\* Checks a JElement if it is in integer

\* Errors if invalid.

\* @Param elmnt the JElement you want to validate

\*/

private void validateInt(JElement elmnt) {

String num = elmnt.getString();

if(!num.matches("\\d+")) {

elmnt.reportError(elmnt, "Invalid data: "+ num + " Located at: " + makeXPath(elmnt) + ". Data should contain a number.");

}

}

**39.Split length**

![Text

Description automatically generated

root.splitTargetByLength(mTX02, 256);

**//in root environment**

public void splitTargetByLength(JElement target, int maxTargetLength) {

String value = target.getString();

String regex = "(?<=\\G.{" + maxTargetLength + "})";

String[] splittedValues = value.split(regex);

if(splittedValues.length <= 1) return;

JEntity targetGroupRep = target.getJParent();

JEntity targetGroup = targetGroupRep.getJParent();

Entity groupRepCopy;

//get index of the current repetition in the repeatable group

//to insert newly created repetitions after the current one

//not at the end of the group

int lastIndex = targetGroupRep.getIndex();

target.setString(splittedValues[0]);

for (int i = 1; i < splittedValues.length; i++) {

groupRepCopy = targetGroupRep.copy();

target = (JElement)groupRepCopy.getChild(target.getIndex() );

target.setString(splittedValues[i]);

targetGroup.add(lastIndex + i, groupRepCopy);

}

}

**40.Regex code:** String regex = "(?<=\\G.{" + 50 + "})";

String [] code=data.split(regex);

if(code.length>=2){

header.address1.setString(code[0]);

header.address2.setString(code[1]);

}else if(code.length==1){

header.address1.setString(code[0]);

}

**41. when addrestype code ST’s addressname or contactname is required**  
if(address.hasRepetition("addressTypeCode","ST")){

if(!address.\_getRep("addressTypeCode","ST").addressName.hasData() && !address.\_getRep("addressTypeCode","ST").contacts.hasRepetition("contactTypeCode","RE")){

reportError("atleast contactname(contactTypeCode=RE) or addressname(addressTypeCode=ST) is required ");

}else if(!address.\_getRep("addressTypeCode","ST").addressName.hasData() && address.\_getRep("addressTypeCode","ST").contacts.hasRepetition("contactTypeCode","RE")){

if(!address.\_getRep("addressTypeCode","ST").contacts.\_getRep("contactTypeCode","RE").contactName.hasData()){

reportError("atleast contactname(contactTypeCode=RE) or addressname(addressTypeCode=ST) is required ");

}

}

}else if(!address.hasRepetition("addressTypeCode","ST")){

reportError("atleast contactname(contactTypeCode=RE) or addressname(addressTypeCode=ST) is required ");

}

**42.atleast addresslocationnumber or vendor required**

if(root.header.address.hasRepetition("addressTypeCode","VN") && !orderHeader.vendor.hasData()){

if(!root.header.address.\_getRep("addressTypeCode","VN").addressLocationNumber.hasData() && !orderHeader.vendor.hasData()){

reportError("atleast addresslocationnumber(addressTypeCode=VN) or vendor is required");

}

}else if(!root.header.address.hasRepetition("addressTypeCode","VN") && !orderHeader.vendor.hasData()){

reportError("atleast addresslocationnumber(addressTypeCode=VN) or vendor is required");

}

**43. Check if field is numeric**

if(!NumberUtilities.IsFullyNumeric(header.invoiceHeader.purchaseOrderNumber.getString())){

reportError("purchaseOrderNumber must be numeric");

**44. Remove leading zeros   
location.setString(location.getString().replaceAll("^0\*", ""));**

**45.index of method:**int i=str1.indexOf(" ");  
//It will return index of 1st space

**46. SPS to validate that TotalAmount is the sum of all items (Price \*Qty) +/- any allowances and charges +/- $.50**

Text

Description automatically generated

String totalAmt = "0.0";

//go through LineItems to gather amounts

for(int detailIndex = 0; detailIndex < lineItem.getChildrenCount(); detailIndex++) {

lineItem.lineItemRep item = \_lineItem(detailIndex);

String qty = item.invoiceLine.invoiceQty.getString();

String price = item.invoiceLine.purchasePrice.getString();

if (!qty.matches( "-?\\d+(\\.\\d+)?") || !price.matches( "-?\\d+(\\.\\d+)?")) return;

String detailTotal = NumberUtilities.Calc.Multiply(price,qty);

totalAmt = NumberUtilities.Calc.Add(detailTotal,totalAmt);

//gather lineItem/chargesAllowances

for(int sacIndex = 0; sacIndex < item.chargesAllowances.getChildrenCount(); sacIndex++) {

lineItem.lineItemRep.chargesAllowances.chargesAllowancesRep sac = item.\_chargesAllowances(sacIndex);

String sacAmount = sac.allowChrgAmt.getString();

if(NumberUtilities.IsFullyNumeric(sacAmount))

{

String indicator = sac.allowChrgIndicator.getString();

if(indicator.matches("A"))

{

totalAmt = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,sacAmount);

} else if (indicator.matches("C")){

totalAmt = NumberUtilities.Calc.Add(totalAmt,sacAmount);

}

}

} // done with lineItem/chargesAllowances

} // done with lineItem

//gather header/chargesAllowances

for(int sacHIndex = 0; sacHIndex < header.chargesAllowances.getChildrenCount(); sacHIndex++) {

header.chargesAllowances.chargesAllowancesRep sacH = header.\_chargesAllowances(sacHIndex);

String sacAmount1 = sacH.allowChrgAmt.getString();

if(NumberUtilities.IsFullyNumeric(sacAmount1))

{

String indicator1 = sacH.allowChrgIndicator.getString();

if(indicator1.matches("A"))

{

totalAmt = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,sacAmount1);

} else if (indicator1.matches("C")){

totalAmt = NumberUtilities.Calc.Add(totalAmt,sacAmount1);

}

}

} // done with header/chargesAllowances

String value = "0.50";

String Decrement = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmt,value);

String Increment = NumberUtilities.Calc.Add(totalAmt,value);

if(summary.totalAmount.hasData()){

if(summary.totalAmount.getNumber().doubleValue() < Double.parseDouble(Decrement) || summary.totalAmount.getNumber().doubleValue() > Double.parseDouble(Increment)){

reportError("TotalAmount:"+ summary.totalAmount.getString()+". Is (+/- 0.50) the sum Of all items (Price\*Qty) +/- any allowances and charges:"+totalAmt+".");

}else{

summary.totalAmount.setNumber(\_source().summary.totalAmount.getNumber());

}

}else{

summary.totalAmount.setNumber(Double.parseDouble(totalAmt));

}

**47.Condition**

Text, letter

Description automatically generated

if(addressTypeCode.getString().equals("ST")){

String addr = addressName.getString();

String addr4 = addr.substring(0,4);

if(NumberUtilities.IsFullyNumeric(addr4)){

if(addr4.equals("0565")||addr4.equals("0408")){

String add = address1.getString();

String add1 = add.substring(0,3);

String aln = addr4 +"-"+ add1;

addressLocationNumber.setString(aln);

}else{

addressLocationNumber.setString(addr4);

}

}

else{

addressLocationNumber.setString("");

}

}

else{

addressLocationNumber.setString(\_source().addressLocationNumber.getString());

}

**48. store line count of lines with Qty > 0 to make sure the 855 contains the same line count**

//lookup

for(int i=0;i<lineItem.getChildrenCount();i++){

write.lineItem.lineItemRep ItemRep = lineItem.\_getRep(i);

if(!ItemRep.orderLine.orderQty.getString().matches("0.0")){

lookupvalue = String.valueOf(root.counter);

root.counter++;

}

}

turnaroundInsert(lookupvalue,header.orderHeader.purchaseOrderNumber.getString(),

"write.lineItem.lineItemRep.orderLine.lineSequenceNumber","","");

49

if(\_source().primaryPhone.hasData() && \_source().primaryEmail.hasData() && \_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryPhone.getString());

contactQual1.setString("TE");

contactNumber2.setString(\_source().primaryEmail.getString());

contactQual2.setString("EM");

}else if(!\_source().primaryPhone.hasData() && \_source().primaryEmail.hasData() && \_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryEmail.getString());

contactQual1.setString("EM");

contactNumber2.setString(\_source().primaryFax.getString());

contactQual2.setString("FX");

}else if(\_source().primaryPhone.hasData() && !\_source().primaryEmail.hasData() && \_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryPhone.getString());

contactQual1.setString("TE");

contactNumber2.setString(\_source().primaryFax.getString());

contactQual2.setString("FX");

}else if(\_source().primaryPhone.hasData() && \_source().primaryEmail.hasData() && !\_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryPhone.getString());

contactQual1.setString("TE");

contactNumber2.setString(\_source().primaryEmail.getString());

contactQual2.setString("EM");

}else if(\_source().primaryPhone.hasData() && !\_source().primaryEmail.hasData() && !\_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryPhone.getString());

contactQual1.setString("TE");

}else if(!\_source().primaryPhone.hasData() && \_source().primaryEmail.hasData() && !\_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryEmail.getString());

contactQual1.setString("EM");

}else if(!\_source().primaryPhone.hasData() && !\_source().primaryEmail.hasData() && \_source().primaryFax.hasData()){

contactNumber1.setString(\_source().primaryFax.getString());

contactQual1.setString("FX");

}

**50.Startswith method(for removing zero from 2 digit number)**

if(\_source().addressLocationNumber.getString().length() == 2 && \_source().addressLocationNumber.getString().startsWith("0")) {

identificationCode.setString(\_source().addressLocationNumber.getString().substring(1));

}

**51. another method for removing leading zeros**

unitPrice.setString(NumberUtilities.Format.RemoveLeadingZeros(NumberUtilities.Format.EnsureDecimalDigits(unitPrice.getString(), 4)));

//ensuredecimaldigit is for ensuring it has 4 digits

**52. To set meta/dropship**

if(header.orderHeader.primaryPOTypeCode.getString().matches("DS")){

meta.setActivated(true);

meta.isDropShip.setString("true");

}else if(header.orderHeader.primaryPOTypeCode.getString().matches("SA|KN|SO")){

meta.setActivated(true);

meta.isDropShip.setString("false");

}

**53. SUM all of the OrderQty**

**//in env**Double totalqt=0.0;

**//in doc fin**

if(!quantity.hasData()){

for(int i=0;i<root.lineItem.getChildrenCount();i++){

totalqt=totalqt + root.lineItem.\_getRep(i).orderLine.orderQty.getDouble();

}

quantity.setNumber(totalqt);

}

**54.Map to refer for lookups**OSR067FairFax.orderChange.fromRSXv7m6r0.toRSXv7m7r4.write  
  
**55. To add spaces to their max length**   
//All fields, with the exception of fields that are marked in the design as Number, are padded to their max with spaces. Padding should be after the value. If no value is present, provide all spaces

**//in root doc fin**

addSpaces(me);

**//in root env**

/\*\*

\* Goes through the map and adds whitespace characters to the value

\* to fill the max field length

\* @param en Entity to start from. Usually the map root

\*/

private void addSpaces(JEntity en) {

if (en instanceof JElement) {

if (!(en instanceof JDate && en instanceof JDouble && ((JElement)en).hasData())){

int fieldLength = Integer.parseInt(((JElement)en).getPropertyString("maxLength"));

int currentLength = ((JElement)en).getString().length();

int spaceAmt = fieldLength - currentLength;

for (int i = 0; i < spaceAmt; i++){

((JElement)en).setString(((JElement)en).getString() + " ");

}

}

} else for (int i=0; i < en.getChildrenCount(); i++) {

addSpaces(en.getJChild(i));

}

}

**56 T&c code for turnaround**

**//in root env**

boolean isTnC = false;

/\*\*

\* Fires a turnaroundLookup to check for TnC override.

\* Return value should be set to a TnC variable for the map.

\* Requires turnaroundLookup() method

\* @param poNumber, the purchaseOrderNumber from the data

\*/

private boolean setTnC(String poNumber) {

String lookupValue = turnaroundLookup(poNumber, "Turnoff OSR Validation", "", "");

if(lookupValue.length() > 0) {

return true;

}

return false;

}

**//in root doc fin**

**isTnC = setTnC(\_source().header.invoiceHeader.purchaseOrderNumber.getString());**

**57. sequence lookup**

\*\*Sequence XREF\*\*

1.write.f47041.eDOC

LookupID: Sender

LookupID2: Receiver

LookupKey:AquaLung\_810\_eDOC

//lookup key format   
//companyname\_docname\_fieldname

**58.help text for data xref**  
\*\*Data Xref's\*\*1) read.header.chargesAllowances.chargesAllowancesRep.allowChrgCode  
LookupId: sender  
LookupId2: receiver  
LookupKey: value from read.f47016.kCOO  
LookupKey2: :read.header.chargesAllowances.chargesAllowancesRep.allowChrgCode"  
LookupValue: Return Value

**59. regex code**

String regex = "(?<=\\G.{" + 30 + "})";

String [] code=data.split(regex);

if(code.length>=2){

dSC1.setString(code[0]);

dSC2.setString(code[1]);

}else if(code.length==1){

dSC1.setString(code[0]);

}

**60.counter**

fieldname.setString(String.valueOf(me.getIndex() +1));

**61. trim function (remove leading and trailing spaces)**

String concat2 = concat.trim();

**62.to set date and time from one field**

if(\_source().header.invoiceHeader.invoiceDate.hasData()) {

if(\_source().header.invoiceHeader.invoiceTime.hasData()) {

formatDate(tranDate.data, "MM/dd/yyyyHH:mm:ss", "yyyy-MM-dd'T'HH:mm:ss");

}else {

formatDate(tranDate.data, "MM/dd/yyyy", "yyyy-MM-dd'T'HH:mm:ss");

}

}else {

tranDate.data.setString(getCurrentDate("yyyy-MM-dd'T'HH:mm:ss"));

}

**63.contains and replaceAll functions**

if(me.hasData() && me.getString().contains("-")){

me.setString(me.getString().replaceAll("-",""));

}

64.To validate time

**//in root environment**

public static boolean isValidRsxTime(String xmlTime){

if(xmlTime == null){

return false;

}else if(xmlTime.length() != 8){

return false;

}else {

return xmlTime.matches("([0|1|2]\\d):([0-5]\\d):([0-5]\\d)");

}

}

**//in root doc fin**

if(!root.isValidRsxTime(me.getString())){

reportError("Unparseable time in invoiceTime must be in the format HH:mm:ss.");

}

**64 when we have to remove a tag field from o/p**

**//root doc fin**

splitOutDocs(documents);

**//root env**

}

/\*\*

\* This Method will write out a new document for each repetition of a RepeatableGroup

\* @param docGroup Should be the only child of the root, with each repetition representing a new document

\*

\*/

private void splitOutDocs(JRepeatableGroup docGroup) {

ArrayList<JEntity> docList = new ArrayList<JEntity>();

//add each rep to a list so we can remove the group.

for(int i=0; i < docGroup.getChildrenCount(); i++) {

docList.add(docGroup.getJChild(i));

}

//remove the document group so it doesn't get written out.

docGroup.delete();

//write out each document rep as a new doc.

for(JEntity doc : docList) {

for(int i=0; i < doc.getChildrenCount(); i++) {

\_root().add(doc.getJChild(i));

}

\_root().getResolver().insert();

\_root().getResolver().endDocument();

//clear for next Doc

while(\_root().getChildrenCount() > 0) {

\_root().getJChild(0).delete();

}

}

}

**65. map to refer for creating new doc rather than new line item**

formats.AquaLung.JDE.itemRegistry.fromSemiColonDelimited.toRSXv7m7r0.write

**66. For ensuring leading zeros.**

me.setString(NumberUtilities.Format.EnsureLeadingZeros(me.getString(),3));

**67.** **Concatenate each occurrence of the source fields and truncate at 204 characters**

String str="";

for(int i=0;i<\_source().technicalErrors.technicalError.getChildrenCount();i++){

str=str+\_source().technicalErrors.technicalError.\_getRep(i).technicalErrorDescription.freeFormMessage.getString();

}

**68. To set header column field by code**

JRepetition firstDetRep = (JRepetition)me.\_orderLevel(0).\_packLevel(0).\_itemLevel(0);

for (int i = 0; i < firstDetRep.getChildrenCount(); i++) {

JElement elem = (JElement) firstDetRep.getJChild(i);

JString newField = new JString();

newField.setLabel(elem.getLabel());

newField.setJavaName(elem.getPropertyString("javaName"));

newField.initialize();

newField.setString(elem.getLabel());

header.add(newField);

}

**69.Maps to refer for roll up logic**  
  
OSR9IEAMDI.Netsuite.Inbound.shipment.fromRSXv7m7r1.toXML.write

OSREHLVitadepotDotcom.Netsuite.inboundShipment.fromRSXv7m7r1.toXML.write

OSRE6WGrandDesign.Netsuite.v2019.ItemFulfillment.shipment.v2.fromRSXv7m7r0.toXML.write

formats.Casper.warehouseShippingAdvice.fromRSXv7m7r0.toPipeDelimited.write

70 maps to refer for timestamp  
OSREDIFeverTree.Netsuite.v2019.orderAck.v2.fromRSXv7m7r0.toXML  
OSRAJ7DoorDash.Netsuite.v2019.orderAck.fromRSXv7m7r0.toXML

OSREDIFeverTree.Netsuite.v2019.WorkOrderCompletion.orderAck.fromRSXv7m7r4.toXML

**71. to add timestamp of T00:00:00**formatDate(record.tranDate.data, "MM/dd/yyyy", "yyyy-MM-dd'T'00:00:00");

**72. To check if data lookup exists or not**

**// in root environment**

private void convertReasonCode(JElement e){

if(e.hasData()){

if (e.getString().matches("NS")){

e.setString("");

return;

}

String returnValue = dataLookup(e.getString(), e.getFullyQualifiedJavaName(), "");

if (!Str.HasData(returnValue)){

e.reportError(e, String.format("Unable to find conversion for shipmentStatusReasonCode '%s'", e.getString()));

}

}

}

**73.Validate length   
//in root environment**

private void validateLength(JElement e, int min, int max){

if (e.hasData() && (e.getString().length() < min || e.getString().length() > max)){

e.reportError(e, String.format("Invalid data: %s located at %s. %s can have between %d and %d characters, but contains %d.",

e.getString(), makeXPath(e), e.getJavaName(), min, max, e.getString().length()));

}

}

**74. Find String**

**// in validation**

String sps\_so\_line = root.findString(\_source(), "customFieldList.customField.customFieldRep[scriptId=custcol\_sps\_so\_line].value.data");

**// in root env**

private String findString(JEntity parent, String expr) {

try{

JEntity temp = parent.get(expr);

if(temp instanceof JElement) {

return ((JElement)temp).getString();

}else {

return "";

}

}catch(NullPointerException e) {

logger.info("Could not find the expression: " + expr);

}

return "";

}

**75 glogXML maps to refer**   
formats.Funko.OTMGlog.carrierShipmentStatusMessage.fromRSXv7m7r0.toXML.write  
OSR8JTGrainCraft.OTMGlog.applicationAdvice.fromSIPv7.toXML.write

**76.Validate Date   
// root environment**

private void validateDate(JElement elmnt){

String dt = elmnt.getString();

try{

SimpleDateFormat myFormat = new SimpleDateFormat("yyyy-MM-dd");

myFormat.setLenient(false);//Only allow valid dates

Date tempDate = myFormat.parse(dt);

}

catch(Exception err){

elmnt.reportError(elmnt, "Invalid date \"" + dt + "\" found in "+ makeXPath(elmnt));

}

}

**77 To combine Date and Time**

**// in root read env**

private void combineDateAndTime(JDate date, JTime time){

if (date.hasData()){

Calendar dateCal = Calendar.getInstance();

dateCal.setTime(date.getDate());

int hour = 0, min = 0, sec = 0, milli = 0;

if (time != null && time.hasData()){

Calendar timeCal = Calendar.getInstance();

timeCal.setTime(time.getDate());

hour = timeCal.get(Calendar.HOUR);

min = timeCal.get(Calendar.MINUTE);

sec = timeCal.get(Calendar.SECOND);

milli = timeCal.get(Calendar.MILLISECOND);

}

dateCal.set(Calendar.HOUR, hour);

dateCal.set(Calendar.MINUTE, min);

dateCal.set(Calendar.SECOND, sec);

dateCal.set(Calendar.MILLISECOND, milli);

date.setDate(dateCal.getTime());

}

}

**// in root write side**

private String getLongDateString(Date d){

SimpleDateFormat formatter = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss");

return formatter.format(d);

**Example code**   
record.customFieldList.customField\_AcknowledgmentDate.platformCorevalue.data.setString(getLongDateString(\_source().header.orderHeader.acknowledgementDate.getDate()));

**78.Ghost structure Maps to Refer**

OSRC80MarcoRubber.shipment.fromFEDSv5010.toRSXv7m7r1

OSRFSCLuontoFurniture.shipment.fromFEDSv5010.toRSXv7m7r4.write

OSREGCRiverSupply.Orgill.shipment.fromFEDSv4010.toRSXv7m7r1.write

formats.SeymourOfSycamore.shipment.fromJSON.toRSXv7m7r0.write  
formats.EliteJewels.shipment.fromCSV.toRSXv7m7r1.write

formats.PureFishing.Hitachi.SAP.DELVRY03.shipment.fromFixedLength.toRSXv7m7r4.write

formats.Casper.shipment.fromPipeDelimited.toRSXv7m7r1.write

**79. To convert into Decimal up to 2 places**

if(me.hasData()){

me.setString(NumberUtilities.Calc.Divide(me.getString(),"100",2));

}

**80.Map to refer for Subline pull up logic**

OSR61YWSI.order.fromRSXv7m7r1.toFEDSv5010.write

**81 Complex code**

if(quantityQualifier.getString().equals("OOR")){

if(totalQty.hasData() || totalQtyUOM.hasData() || (dates.hasRepetition("dateTimeQualifier","018") && dates.\_getRep("dateTimeQualifier","018").date.hasData())){

root.mandatoryCheck(totalQty,"if either TotalQty, TotalQtyUOM or Date where QuantityQualifier=OOR is present then all are required");

root.mandatoryCheck(totalQtyUOM,"if either TotalQty, TotalQtyUOM or Date where QuantityQualifier=OOR is present then all are required");

root.mandatoryRep(dates,"dateTimeQualifier","018","if either TotalQty, TotalQtyUOM or Date where QuantityQualifier=OOR is present then all are required");

if(dates.hasRepetition("dateTimeQualifier","018") && !dates.\_getRep("dateTimeQualifier","018").date.hasData()){

reportError("if either TotalQty, TotalQtyUOM or Date where QuantityQualifier=OOR is present then all are required");

}

}

}

if(quantityQualifier.getString().equals("33")){

root.mandatoryCheck(totalQty,"totalQty is required when quantityQualifier is 33");

root.mandatoryCheck(totalQtyUOM,"totalQtyUOM is required when quantityQualifier is 33");

root.mandatoryRep(dates,"dateTimeQualifier","018","");

if(dates.hasRepetition("dateTimeQualifier","018") && !dates.\_getRep("dateTimeQualifier","018").date.hasData()){

reportError("Missing mandatory data: /ItemRegistry/Structure/LineItem/QuantitiesSchedulesLocations/Dates/Date when dateTimeQualifier=018");

}

}

**82. Code to add blank spaces**

for (int i = 0; i <30-len; i++){

adl=adl+ " ";

}

**83 Code to set count of line numbers// Counter**

**//in valid**

String linecount = Integer.toString(root.detailCount++);

//in root env

Integer detailCount = 1;

**84 Code to Remove excess fields in JSON O/P**  
dropBlankJSON(me);

**//In root env**

/\*\*

\* Recursively loops through each JElement. Change what you want to do for each field.

\* @param entity the current Entity

\*/

private void alterFieldsJSON(JEntity entity){

if(!(entity instanceof JElement)){//if entity is anything but an element, loop through it's children

for(int i=0; i < entity.getChildrenCount(); i++){

JEntity child = (JEntity)entity.getChild(i);

if(!(child instanceof JElement)){//not an Element, call again.

alterFieldsJSON(child);

}else{//child is an element.

if(!((JElement)child).hasData()){

deleteList.add(child);

}

}

}

}

}

/\*\*

\* Used for JSON maps to drop all blank fields and gorupsconsolidates

\* consolidates the call to deactivateOptionalJSON and alterFieldsJSON.

\* Must be called on Docfinalize

\* do not call consolidated methods separately

\* @param jEntity Any JEntity, should be the root.

\*/

List<JEntity> deleteList = new ArrayList<JEntity>();

private void dropBlankJSON(JEntity jEntity){

deactivateOptionalJSON(jEntity);

alterFieldsJSON(jEntity);

for (JEntity bye : deleteList){

bye.delete();

}

}

**85. Code to drop unnecessary Spaces**

**//alterfields**

\* Recursively loops through each JElement. Change what you want to do for each field.  
 \* [@Param](https://spscommerce.slack.com/team/UEHJKKYQG) entity the current Entity  
 \*/  
private void alterFields(JEntity entity){  
    if(!(entity instanceof JElement)){//if entity is anything but an element, loop through it's children  
        for(int i=0; i < entity.getChildrenCount(); i++){  
            JEntity child = (JEntity)entity.getChild(i);  
            if(!(child instanceof JElement)){//not an Element, call again.  
                alterFields(child);  
            }else{//child is an element.  
                if(((JElement)child).hasData()){  
                    ((JElement)child).setString(((JElement)child).getString().trim());  
                }  
            }  
        }  
    }  
}

**86.Grouplookup codes Map**

formats.LogisticsTurnaround.order.fromRSXv7m7r1.toRSXv7m7r1.ordRsxWrite  
formats.LogisticsTurnaround.warehouseStockTransferShipmentAdvice.fromRSXv7m7r0.toRSXv7m7r0.shiRsxWrite

https://atlassian.spscommerce.com/browse/FIBQ-59970

**87 Ghost structure (in write side )**  
formats.AdvancedDistribution.shipment.fromXML.toRSXv7m7r0.write

**88 SAP Maps**

formats.Agropur.SAP.orders05.orderAck.v2.fromXML.toRSXv7m7r0.write  
formats.CABEnterprises.SAP.delvry03.shipment.fromRSXv7m7r0.toFixedLength.write

OSRFRLOneLink.SAP.invoic02.invoice.fromRSXv7m7r4.toXML.write

formats.NiproMedical.SAP.DELVRY03.order.fromFixedLength.toRSXv7m7r0.write

formats.NordicNaturals.SAP.delvry07.shipment.fromXML.toRSXv7m7r0.write

**89. To validate maximum decimal**

/\*\*  
     \* Checks a JElement if it contains up to a number of decimal places  
     \* Errors if invalid.  
     \* [@Param](https://spscommerce.slack.com/team/UEHJKKYQG) elmnt the JElement you want to validate  
     \* [@Param](https://spscommerce.slack.com/team/UEHJKKYQG) decimalPlaces the max number of decimal places allowed in this element  
     \*/

    private void validateMaxDecimals(JElement elmnt, int decimalPlaces) {  
        String num = elmnt.getString();  
        String decPlaces = String.valueOf(decimalPlaces);  
        if(elmnt.hasData() && !num.matches("\\d+\\.?\\d{0," + decPlaces +"}")) {  
            elmnt.reportError(elmnt, "Invalid data: " + num + " Located at: " + makeXPath(elmnt) + ". Data should contain a number with at most " + decPlaces + " decimal places.");  
        }  
    }

**90. Subtract if allowchargeindicator is A and add if it is C**

for(JEntity chrgRep : header.chargesAllowances){

if(chrgRep.getString("allowChrgIndicator").equals("A")){

totalAmount = NumberUtilities.Calc.SubtractSecondFromFirst(totalAmount, chrgRep.getString("allowChrgAmt"));

}else if(chrgRep.getString("allowChrgIndicator").equals("C")){

totalAmount = NumberUtilities.Calc.Add(totalAmount, chrgRep.getString("allowChrgAmt"));

}

}

**91. Atleast addresslocationnumber or vendor is required**

if(header.address.hasRepetition("addressTypeCode","VN") && !header.invoiceHeader.vendor.hasData()){

if(!header.address.\_getRep("addressTypeCode","VN").addressLocationNumber.hasData()){

reportError("atleast addresslocationnumber(addressTypeCode=VN) or vendor is required");

}

}else if(!header.address.hasRepetition("addressTypeCode","VN") && !header.invoiceHeader.vendor.hasData()){

reportError("atleast addresslocationnumber(addressTypeCode=VN) or vendor is required");

}

**92 TNC code**

// in doc finalize  
isTnC = setTnC(header.invoiceHeader.purchaseOrderNumber.getString());

boolean isTnC = false;

/\*\*

\* Fires a turnaroundLookup to check for TnC override.

\* Return value should be set to a TnC variable for the map.

\* Requires turnaroundLookup() method

\* @param poNumber, the purchaseOrderNumber from the data

\*/

private boolean setTnC(String poNumber) {

String lookupValue = turnaroundLookup(poNumber, "Turnoff OSR Validation", "", "");

if(lookupValue.length() > 0) {

return true;

}

return false;

}

**pu93 add days to Data**

**lprivate String addDays(String date, int intDays){**

**Calendar cal = Calendar.getInstance();**

**SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd");**

**df.setLenient(false);**

**try{**

**Date dtOriginal = df.parse(date);**

**cal.setTime(dtOriginal);**

**cal.add(Calendar.DAY\_OF\_YEAR, intDays);**

**return String.valueOf(df.format(cal.getTime()));**

**}catch (Exception e){**

**logger.info("Error occurred in addDays() method. Could not parse date string: " + date);**

**}**

**return null;**

**}**

**p94 To set counter at linesequncenumber**

**orderLine.lineSequenceNumber.setString(String.valueOf(me.getIndex()+1));**

**p95 Map for turnaround on locations** OSRE7XVailResortsCanada.JDA.order.fromPipeDelimited.toRSXv7m7r4.write

**96 creating group via code**

notes.notesRep noteRep = notes.\_jrepeat();

noteRep.noteCode.setString("GEN");

noteRep.note.setString(note.substring(1,note.length()));

fOBRelatedInstruction.fOBRelatedInstructionRep fobRep = fOBRelatedInstruction.\_jrepeat();

fobRep.fOBPayCode.setString("DF");

fobRep.fOBLocationQualifier.setString(root.\_source().header.eDIPOH.eHFOBL.getString());

fobRep.fOBLocationDescription.setString(root.\_source().header.eDIPOH.eHFOBI.getString());

**97 setting field to receiver**

addressName.setString(\_root().getPropertyString("receiver"));

**98.Monster Map ,BODS XML INFORM3**

formats.Pompeian.InforM3.BODS.invoice.fromXML.toRSXv7m7r0.write

**99 how to check if a group is active and how to activate a group**

if(!header.address\_SF.isActivated()){

header.address\_SF1.setActivated(true);

}

**100. To check if the string starts with number and extract that number and store it**

if(Character.isDigit(desc1.charAt(0))){

String input1 = desc1;

int k=0;

k=root.findFirstLetterPosition(input1);

termsDiscountDueDays.setString(input1.substring(0,k).trim());

}

**101. To check if the string empty or not**

String ptdl =root.dataLookup(me.getString(),"write.header.paymentTerms.paymentTermsRep.paymentMethodCode","");

if(Str.HasData(ptdl)){

me.setString(ptdl);

}

**102. split in array**

if (adrs.contains(",")){

String[] adSplit = adrs.split(",");

}

**103. trigger lookup**

Boolean triggerFA = false;

triggerFA=Str.HasData(root.dataLookup("","write.header.fOBRelatedInstruction\_flat.fOBLocationDescription\_trigger",""));

if(!triggerFA){

me.delete();

}

**104. converting string to double**

Double.parseDouble(\_source().userArea.property.\_getRep(i).nameValue.data.getString()));

**105.If there are multiple references groups with referenceQual=SE SPS will create a CSV line for each instance of that references group with referenceQual=SE**

**// to create new records for refence group rather than lineitem**

OSRBX7ConvergeOne.shipment.fromRSXv7m7r0.toCSV.write

**106 Going through multiple loops**

for(int j=0; j<root.\_source().e2EDL20.getChildrenCount();j++){

read.e2EDL20.e2EDL20Rep EDL20rEP = root.\_source().e2EDL20.\_getRep(j);  
}

**107 Reverse engineering**   
formats.Pompeian.productActivity.fromRSXv7m7r0.toPipeDelimited.write

**108. When a script is needed**

[[FIBQ-59640] Hitachi | Pure Fishing | pipe delimited app - Jira (spscommerce.com)](https://atlassian.spscommerce.com/browse/FIBQ-59640)

**109. To validate if it is number (validatenumber)**

private void validateNumber(JElement elmnt) {  
        String num = elmnt.getString();  
        if(elmnt.hasData() && !num.matches("\\d+\\.?\\d\*$")) {  
            elmnt.reportError(elmnt, "Invalid data: " + num + " Located at: " + makeXPath(elmnt) + ". Data should contain a number.");  
        }  
    }

**110. FEDS CODE for SDQ (READ side SDQ write side LOCATION )(FEDS TO RSX)**

/\*\*

\* Populates a RSX LocationQuantities group from a SDQ FEDS group

\* @param fedsSDQGroup a FEDS SDQ group to source from

\* @param locationsGroup an RSX LocationQuantities group to source to

\*/

private void populateLocationQuantity(JEntity fedsSDQGroup, JEntity locationsGroup){

HashMap<String, String> values = new HashMap<String, String>();

for (int i = 0; i < fedsSDQGroup.getChildrenCount(); i++){

if (fedsSDQGroup.getJChild(i) instanceof JElement && fedsSDQGroup.getJChild(i).getPropertyString("referenceNum").equals("67") && ((JElement)fedsSDQGroup.getJChild(i)).hasData()){

values.put(((JElement)fedsSDQGroup.getJChild(i)).getString(), ((JElement)fedsSDQGroup.getJChild(i+1)).getString());

}

}

for (Map.Entry <String, String> value : values.entrySet()){

JRepetition newRep = ((JRepeatableGroup)locationsGroup.getJChild("locationQuantity")).jrepeat();

((JElement)newRep.getJChild("location")).setString(value.getKey());

((JElement)newRep.getJChild("qty")).setString(value.getValue());

}

}

**111.HL count**

formats.SimonSchuster.shipment.fromRSXv7m7r0.toFEDS.write

OSR067GolfTown.NLS.shipment.fromSIPv7.toXML.write

**112. FOR Julian date format**

formats.WarnIndustries.order.fromRSXv7m7r0.toJDEjson.write

formats.Casper.warehouseShippingAdvice.fromRSXv7m7r0.toPipeDelimited.write

formats.WarnIndustries.JDE.invoice.fromJSON.toRSXv7m7r0.write

//code

/\*\*

\* All JDates must be changed to JStrings.

\* Recursively loops through each JElement. When the fieldName contains 'Date' this will try and

\* format the date to the given format.

\* @param ent the current Entity

\* @param writeFormat the dateFormat you want written out.

\*/

private void formatAllDates(JEntity ent, String writeFormat){

if(!(ent instanceof JElement)){//if entity is anything but an element, loop through it's children

for(int i=0; i < ent.getChildrenCount(); i++){

JEntity child = (JEntity)ent.getChild(i);

if(!(child instanceof JElement)){//not an Element, call again.

formatAllDates(child, writeFormat);

}else{//child is an element.

if(!child.getPropertyString("name").matches("^DateTimeQualifier$") && child.getPropertyString("name").matches("^.\*[Dd]ate.\*$") && ((JElement)child).hasData()) {

String[] dateFormatList = {"1yyD"};

//change the value in this field to the writeFormat

for(int d=0; d < dateFormatList.length; d++) {

String value = ((JElement)child).getString();

try{

SimpleDateFormat dfRead = new SimpleDateFormat(dateFormatList[d]);

dfRead.setLenient(false);

SimpleDateFormat dfWrite = new SimpleDateFormat(writeFormat);

dfWrite.setLenient(false);

((JElement)child).setString(dfWrite.format(dfRead.parse(value)));

break; //this worked, skip the rest of the formats.

}catch(Exception e) {

if(d == dateFormatList.length - 1) {

logger.info("Unable to format the date: " + value + " in " + child.getFullyQualifiedJavaName());

}else {

logger.info("Trying next date format: " + dateFormatList[d] + " for field " + child.getFullyQualifiedJavaName());

}

}

}

}

}

}

}

}

/\*\*  
 \* Converts a date string to a Julian date. Reads in the date format yy-MM-dd (change as needed)  
 \* [@Param](https://spscommerce.slack.com/team/UEHJKKYQG) date String formatted date (yy-MM-dd) change format as needed.  
 \* @return a String formatted julian date  
 \*/  
private String convertToJulian(String date){  
    try {  
        SimpleDateFormat dateFormat = new SimpleDateFormat("MM/dd/yyyy");  
        Calendar calendar = Calendar.getInstance();  
        calendar.setTime(dateFormat.parse(date));  
        int year = calendar.get(Calendar.YEAR);  
        String syear = String.format("%04d",year).substring(2);  
        int century = Integer.parseInt(String.valueOf(((year / 100)+1)).substring(1));  
        int julian = Integer.parseInt(String.format("%d%s%03d",century,syear,calendar.get(Calendar.DAY\_OF\_YEAR)));  
        return String.valueOf(julian);  
    } catch (Throwable promiseDateError){  
        [logger.info](http://logger.info/)("\n\nDate Format did not succeed: "+promiseDateError+"\n\n");  
        return date;  
    }  
}

**113. 3 repeatable group priority sourcing to one group in write side**

**// code**

private void populateFedsLIN(JGroup sourceGroup,JEntity sourceGroup1, JEntity targetGroup) {

//sourceGroup is expected to be the group that contains the BPN, VPN, etc.

LinkedHashMap<String, String> fieldValues = new LinkedHashMap<String, String>();

for (int i = 0; i < sourceGroup.getJChild("productID").getChildrenCount(); i++){

fieldValues.put(sourceGroup.getJChild("productID").getJChild(i).getString("partNumberQual"),

sourceGroup.getJChild("productID").getJChild(i).getString("partNumber"));

}

for (int i = 0; i < sourceGroup1.getJChild("references").getChildrenCount(); i++){

fieldValues.put(sourceGroup1.getJChild("references").getJChild(i).getString("referenceQual"),

sourceGroup1.getJChild("references").getJChild(i).getString("referenceID"));

}

for (int i = 0; i < sourceGroup1.getJChild("productOrItemDescription").getChildrenCount(); i++){

fieldValues.put(sourceGroup1.getJChild("productOrItemDescription").getJChild(i).getString("productCharacteristicCode"),

sourceGroup1.getJChild("productOrItemDescription").getJChild(i).getString("productDescription"));

}

//source values from hash

int curChild = 16;

for(String key : fieldValues.keySet()) {

for(int i=curChild; i < 42; i++) {

if (targetGroup.getJChild(i) instanceof JElement){

if(fieldValues.get(key).length() > 0 ) {

((JString)targetGroup.getJChild(i)).setString(key);

((JString)targetGroup.getJChild(i+1)).setString(fieldValues.get(key));

curChild = i+2;//move the pointer to the next field

break;

}

}

}

}

}

**114. code for map to set a field to tpid for all records**

private void populateTpid(JEntity ent){

if(ent instanceof JElement){

if(((JElement)ent).getJavaName().matches("tUTPID")){

((JElement)ent).setString(\_root().\_source().header.planningHeader.tradingPartnerId.getString());

}

}else{

for(int i=0; i<ent.getChildrenCount(); i++){

populateTpid(ent.getJChild(i));

}

}

}

**115. code for map to set date fields 0 if data is not present**

private void defaultforDate(JEntity en) {

if (en instanceof JElement) {

if (en instanceof JDate && !((JElement)en).hasData()){

((JElement)en).setString("0");

}

} else{

for (int i=0; i < en.getChildrenCount(); i++) {

defaultforDate(en.getJChild(i));

}

}

}

**116 MAP to set key via code (ghost structure)(split logic)**

formats.PureFishing.Hitachi.warehouseStockTransferShipmentAdvice.fromRSXv7m7r0.toPipeDelimited.write

OSR3HSTownandCountryMarkets.shipment.fromRSXv7m7r1.toFixedLength.write

**117 Eaches to cases conversion**  
formats.MirMosaic.Fishbowl.itemRegistry.fromCSV.toRSXv7m7r0.write

**118 Map to repeat line item based on Location (where repeat entire order based on each location)(location lookup)**

format.AllSaints.order.fromRSXv7m7r0.toCommaDelimited.write

OSR565MarksWorkWarehouse.BobDale.order.fromRSXv7m7r4.toFEDSv4010.write

**119 special ghost for JSON 856**formats.WarnIndustries.JDE.shipment.fromJSON.toRSXv7m7r0.write

**120 APAC CASES**  
OSRFRPlagardereTravelRetailPacific.MSDynamics365BC.shipment.fromRSXv7m7r4.toXML.write

OSRFRPlagardereTravelRetailPacific.MSDynamics365BC.order.fromXML.toRSXv7m7r4.write