# **Configuration Section**

## **1 US#1234**

### **1.1 Configuration Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Variable Name** | **Description** | **Data Type** |
| Region | Region\_pcr | Used to show the region. | Float |
| Blades | blades\_quantity | Created to show the blades quantity. | Integer |

## **1.2 Recommendation Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Packaged Solutions [Table] | packagedSolutions | This rec rule is used to package the solutions. |
| Populate Blade Image [Easy Advanced] | populateBladeEnclosureImage | This rule is used to populate the image of blade in the page. |

## **1.3 Constraint Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Constrain Chassis by Performance [Table] | constrainChassisByPerformance | This rule constraints the chassis by performance levels. |

## **1.4 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide Blade Quantity/Enclosure [Simple] | hideBladeQty | This rule hides the blade quantity/enclosure |

## **1.5 Recommended Items rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Server Rack [Table] | serverRack | This rule sets the item & price on the quote from the new string. |

## **1.6 Configuration Flow rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Server Solutions | serverSolutionsProduct | This rule is created to group different vision servers attributes to display them on config screen. |

## **1.7 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Parse CSV | parseCSV | This util is used to parse the CSV file. |

## **Script Text**

arr = string[][];  
raw = urldatabyget(fileLocation, "", "ERROR");  
if(raw == "ERROR"){
return arr;  
}
goodLines = 0;  
knownColumns = 0;  
rawLines = split(raw,"\n");  
for rl in rawLines{
rawCols = split(rl,",");  
if(goodLines == 0){
knownColumns = sizeofarray(rawCols);  
}
if(sizeofarray(rawCols) <> knownColumns){
break;  
}
else{
goodLines = goodLines + 1;  
}
}
rows = range(goodLines);  
cols = range(knownColumns);  
for row in rows{
rawColumns = split(rawLines[row],",");  
for col in cols{
arr[row][col] = rawColumns[col];  
}
}
return arr;

## **1.8 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Quote Analysis PL | quoteAnalysisPL | This util analyze the quote PL. |

## **Script Text**

result="";  
monthlyRate = pow( 1 + (discountRate / 100.0), 1 / 12.0);  
term = range(6);  
decimalPlace = 2;  
 // round values to this many decimal places
for t in term{
years = t;  
months = 12 \* years;  
nominalRevenue = oneTimeRevenue + (months \* monthlyRevenue);  
nominalCost = oneTimeCost + (months \* monthlyCost);  
nominalPL = nominalRevenue - nominalCost;  
nominalReturn = 0.0;  
if(nominalCost <> 0.0){
nominalReturn = ((nominalRevenue / nominalCost) - 1.00) \* 100.0;  
nominalReturn = round(nominalReturn, decimalPlace);  
 // added on 2014.12.29 to resolve rounding error that emerged after the 2014R2 upgrade
}
counter = range(months);  
NPVRevenue = oneTimeRevenue;  
NPVCost = oneTimeCost;  
for c in counter{
NPVRevenue = NPVRevenue + monthlyRevenue / pow(monthlyRate, c + 1.0);  
NPVCost = NPVCost + monthlyCost / pow(monthlyRate, c + 1.0);  
}
NPVPL = NPVRevenue - NPVCost;  
NPVreturn = 0.0;  
if(NPVCost <> 0.0){
NPVreturn = ((NPVRevenue / NPVCost) - 1.00) \* 100.0;  
NPVreturn = round(NPVreturn, decimalPlace);  
 // added on 2014.12.29 to resolve rounding error that emerged after the 2014R2 upgrade
}
if(t == 0){
result = result + "1~nominalRevenue\_t~" + string (nominalRevenue) + "|" ;  
result = result + "1~nominalCost\_t~" + string (nominalCost) + "|";  
result = result + "1~nominalPL\_t~" + string (nominalPL) + "|";  
result = result + "1~nominalReturn\_t~" + string (nominalReturn) + "|";  
result = result + "1~nPVRevenue\_t~" + string (NPVRevenue) + "|";  
result = result + "1~nPVCost\_t~" + string (NPVCost) + "|";  
result = result + "1~nPVPL\_t~" + string (NPVPL) + "|";  
result = result + "1~nPVReturn\_t~" + string (NPVreturn) + "|";  
}
else{
result = result + "1~year" + string(t) + "NominalRevenue\_t~" + string (nominalRevenue) + "|";  
result = result + "1~year" + string(t) + "NominalCost\_t~" + string (nominalCost) + "|";  
result = result + "1~year" + string(t) + "NominalPL\_t~" + string (nominalPL) + "|";  
result = result + "1~year" + string(t) + "NominalReturn\_t~" + string (nominalReturn) + "|";  
result = result + "1~year" + string(t) + "NPVRevenue\_t~" + string (NPVRevenue) + "|";  
result = result + "1~year" + string(t) + "NPVCost\_t~" + string (NPVCost) + "|";  
result = result + "1~year" + string(t) + "NPVPL\_t~" + string (NPVPL) + "|";  
result = result + "1~year" + string(t) + "NPVReturn\_t~" + string (NPVreturn) + "|";  
}
}
return result;

## **2 US#5678**

### **2.1 Configuration Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Variable Name** | **Description** | **Data Type** |
| Enclosures Required | enclosuresRequired | Each Vision Blade 6000 has a maximum capacity of 10 Blades | Integer |
| Enclosure | enclosure | This attribute holds the enclosure details | Text |

## **2.2 Recommendation Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Set Enclosures Required [Easy Advanced] | setEnclosuresRequired | This rec rule is used to set the required enclosures. |
| Force Set Chassis [Simple] | forceSetChassis | This rule force set the chassis. |

## **2.3 Constraint Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Constrain Rack by Size [Simple/Table] | constrainRackBySize | This rule constraints the rack size if it crosses the limits. |

## **2.4 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide ServerQty/Rack [Simple] | hideServerQuantity | This rule hides the server quantity/racks |

## **2.5 Recommended Items rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Power Supply (selected) [Table] | powerSupply | This RI rule sets the item & price on the quote from the new string. |

## **2.6 Configuration Flow rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Display Pricing details | displayPricingDetails | This flow will display pricing related details on config screen. |

## **2.7 Pricing rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Calculate Price for Vision Servers | calculatePriceForVisionServers | This pricing rule calculates prices for vision servers based on combination of configured values. |

## **2.8 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| getBasicAuthCredentials | CN\_GetBasicAuthCredentials | This is used to retrieve the authorization credentials. |

## **Script Text**

//get credentials for JCS-SXs 2 legged OAuth client that was stored by you in secure columns of custom data tables
clientname = accountName;  
clients = bmql("select username, password from CN\_Credentials where integrationName=$clientname");  
//base 64 encode the credentials, per Basic Authorization standard
for client in clients {
encodedCreds= encodebase64( get(client, "username") + ":" + get(client, "password"));  
return encodedCreds;  
}
//allow appropriate message to user, if integration client information was not located
return defaultReturnValue;

## **2.9 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Get File Attachment Data | CN\_GetFileAttachmentData | This is used to retrieve the file attachment. |

## **Script Text**

response = dict("string");  
url = siteUrl+"/rest/v2"+
"/commerceProcesses/"+ processVarName+
"/transactions/"+ string(transactionId)+
"/attachments/"+ attachmentVarName+"/";  
headers = dict("string");  
put(headers, "Authorization", userAuth );  
put(headers, "Accept", "application/json");  
// call rest method
xlsxResponse = urldata( url, "GET", headers );  
if( get(xlsxResponse, "Status-Code") <> "200" ){
put(response, "error", get(xlsxResponse, "Error-Message"));  
return response;  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
}else{
mb = get(xlsxResponse, "Message-Body");  
jmb = json(mb);  
fileContent = jsonget(jmb, "fileContent");  
if( len(fileContent) < 5 ){
put(response, "error", "GetFileAttachment: Could not locate document in Response.");  
return response;  
}
put(response, "base64doc", fileContent);  
return response;  
};

## **3 US#7661**

### **3.1 Configuration Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Variable Name** | **Description** | **Data Type** |
| Power Supply | powerSupply | This menu attribute lists the power supply PSU's to choose. | Text |

## **3.2 Recommendation Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Set Remaining Blades [Easy Advanced] | setRemainingBlades | This rec rule helps to set the remaining blades slots. |

## **3.3 Constraint Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Inventory Check [Simple] | inventoryCheck | Filler Rule - To be reimplemented with real inventory management with EBS in the future. |

## **3.4 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide Adv Rack Configuration [Simple] | hideAdvancedRackConfiguration | This rule hides the advance rack configuration if server module equals UPS 1500 |

## **3.5 Recommended Items rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Operating System [Volume Table] | operatingSystemTable | This RI rule sets the item & price on the quote from the new string. |

## **3.6 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Get File Id From Response | CN\_GetFileIdFromResponse | This util is used to get the file ID from EBS response. |

## **Script Text**

mb = get(response, "Message-Body");  
jmb = json(mb);  
ids = jsonget(jmb, "fileId");  
if( isnumber(ids) ){
return atoi(ids);  
}
return -1;

## **3.7 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Update File Attachment | CN\_UpdateFileAttachment | This util is used to update the file attachment. |

## **Script Text**

response = dict("string");  
url = siteUrl+"/rest/v2"+
"/commerceProcesses/" + processVarName+
"/transactions/" + string(transactionId) +
"/attachments/" + attachmentVarname;  
headers = dict("string");  
put(headers, "Authorization", userAuth );  
put(headers, "Accept", "application/json");  
put(headers, "Content-Type", "application/json");  
// add json data
usePrintContext = "false";  
if(NOT hasBrowse){
usePrintContext = "true";  
}
jsonObj = json();  
jsonput(jsonObj, "mediaType", fileType);  
jsonput(jsonObj, "fileContent", fileContent);  
jsonput(jsonObj, "useTempTable", usePrintContext);  
jsonput(jsonObj, "usePrintContext", usePrintContext);  
jsonput(jsonObj, "fileName", fileName);  
// call rest method
xlsxResponse = urldata( url, "POST", headers, jsontostr(jsonObj) );  
if( get(xlsxResponse, "Status-Code") <> "200" ){
put(response, "error", get(xlsxResponse, "Error-Message"));  
return response;  
}else{
return xlsxResponse;  
};

## **4 US#8341**

### **4.1 Configuration Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Variable Name** | **Description** | **Data Type** |
| Processor | processors | This menu attribute shows the list of processors available | Text |

## **4.2 Recommendation Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Set Total Equipment [Easy Advanced] | setTotalEquipment | This rule sets the total equipment used. |

## **4.3 Constraint Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Contrain Processor\RAM\HDD\PSU\OS [Table] | contrainRAM | Based on performance this rule constraints processor/RAM/HDD/PSU/OS |

## **4.4 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide HDD Array [Simple] | hideHDDArray | This rule hides the HDD array if additional HDD's is false |

## **4.5 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Print Email Designer BML | CN\_PrintEmailDesigner | This util design the email format as approved by client. |

## **Script Text**

// Prints a Email Designer Template. Returns "error" or "subject" & "body".
response = dict("string");  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
// PRINT TEMPLATE
url = siteUrl+"/rest/v2/emailGenerator/";  
headers = dict("string");  
put(headers, "Authorization", userAuth );  
put(headers, "Accept", "application/json");  
put(headers, "Content-Type", "application/json");  
// add json data
jsonObj = json();  
jsonput(jsonObj, "processVarname", processVarName);  
jsonput(jsonObj, "templateName", templateName);  
jsonput(jsonObj, "transactionId", transactionId);  
jsonput(jsonObj, "languageCode", templateLanguage);  
// call rest method
xlsxResponse = urldata( url, "POST", headers, jsontostr(jsonObj) );  
// CHECK OUTPUT, UPDATE STATUS
statusCode = get(xlsxResponse, "Status-Code");  
printStatus = "";  
if( statusCode <> "204" ){
printStatus = "Print FAILED: " + templateName + "\n" +
"Error: " + get(xlsxResponse, "Error-Message");  
put(response, "error", printStatus);  
return response;  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
}else{
printStatus = "Print Completed: " + templateName + "\n";  
}
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
// GET PRINT OUTPUT
url = get(xlsxResponse, "Location");  
headers = dict("string");  
put(headers, "Authorization", userAuth);  
put(headers, "Accept", "application/json");  
// get print output call
xlsxResponse = urldata( url, "GET", headers );  
mb = get(xlsxResponse, "Message-Body");  
if(len(mb) > 0){
jmb = json(mb);  
printOutput = jsonget(jmb , "subject" );  
put(response, "subject", printOutput);  
printOutput = jsonget(jmb , "body" );  
put(response, "body", printOutput);  
}else{
put(response, "error", "Email Output was empty.");  
}
print("PRINT COMPLETE");  
print("");  
return response;

## **5 US#1003**

### **5.1 Configuration Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Name** | **Variable Name** | **Description** | **Data Type** |
| RAM | ram | This menu attribute shows the list of RAM available | Text |

## **5.2 Recommendation Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Rec Cloud Backup [Simple] | recCloudBackup | This rec rule helps to protect the data using cloud backup |

## **5.3 Constraint Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Open Source Software [Simple] | openSourceSoftware | This rule constraints OSS system if it requires Titanium support. |

## **5.4 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide Cloud Storage [Simple] | hideCloudBackup | This rule hides the cloud storage if cloud backup is false |

## **5.5 Recommended Items rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| SLA Parts | sLAParts | This RI rule sets the item & price on the quote from the new string. |

## **5.6 Configuration Flow rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Create Model from EBS for Modifications | createModelFromEBSForModifications | This flow is created to show the model from EBS with renewal attributes. |

## **5.7 Pricing rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Calculate Prices for renewal attributes from EBS | calculatePricesForRenewalAttributesFromEBS | This pricing rule calculates prices for renewal model created. |

## **5.8 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Compare Docx | CN\_CompareDocx | This util helps to compare the docx files. |

## **Script Text**

response = dict("string");  
// GENERATE DIFF REST CALL
url = siteUrl+"/rest/v2/docxCompare";  
jsonObj = json();  
jsonput(jsonObj, "processVarname", processVarName);  
jsonput(jsonObj, "transactionId", transactionId);  
jsonput(jsonObj, "oldDocAttachId", prevFileAttachId);  
jsonput(jsonObj, "newDocAttachId", curFileAttachId);  
headers = dict("string");  
put(headers, "Authorization", userAuth );  
put(headers, "Accept", "application/json");  
put(headers, "Content-Type", "application/json");  
// call rest method
xlsxResponse = urldata( url, "POST", headers, jsontostr(jsonObj) );  
if( get(xlsxResponse, "Status-Code") <> "200" ){
err = get(xlsxResponse, "Error-Message");  
if(len(err) <= 5){
err = get(xlsxResponse, "Message-Body");  
}
put(response, "error", err);  
return response;  
}else{
mb = get(xlsxResponse, "Message-Body");  
jmb = json(mb);  
diffs = jsonget(jmb , "diffs" );  
if(len(diffs) > 3){
put(response, "jsonString", mb);  
}
}
print("COMPARE COMPLETE");  
return response;

## **5.9 BML Util Libraries**

|  |  |  |
| --- | --- | --- |
| **Util Name** | **Variable Name** | **Description** |
| Print Doc Designer | CN\_PrintDocDesigner | This util prints the doc designer template. |

## **Script Text**

// Prints a Document Designer Document. Returns "error" or "base64doc".
response = dict("string");  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
// PRINT TEMPLATE
url = siteUrl+"/rest/v2/documentGenerator/";  
headers = dict("string");  
put(headers, "Authorization", userAuth );  
put(headers, "Accept", "application/json");  
put(headers, "Content-Type", "application/json");  
// add json data
jsonObj = json();  
jsonput(jsonObj, "processVarname", processVarName);  
jsonput(jsonObj, "templateName", templateName);  
jsonput(jsonObj, "transactionId", transactionId);  
jsonput(jsonObj, "languageCode", templateLanguage);  
jsonput(jsonObj, "outputFormat", outputFormat);  
// call rest method
xlsxResponse = urldata( url, "POST", headers, jsontostr(jsonObj) );  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
// CHECK OUTPUT, UPDATE STATUS
statusCode = get(xlsxResponse, "Status-Code");  
printStatus = "";  
if( statusCode <> "204" ){
printStatus = "Print FAILED: " + templateName + "\n" +
"Error: " + get(xlsxResponse, "Error-Message");  
put(response, "error", printStatus);  
return response;  
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
}else{
printStatus = "Print Completed: " + templateName + "\n";  
}
/\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/
// GET PRINT OUTPUT
url = get(xlsxResponse, "Location");  
headers = dict("string");  
put(headers, "Authorization", userAuth);  
put(headers, "Accept", "application/json");  
// get print output call
xlsxResponse = urldata( url, "GET", headers );  
mb = get(xlsxResponse, "Message-Body");  
if(len(mb) > 0){
jmb = json(mb);  
printOutput = jsonget(jmb , "document" );  
put(response, "base64doc", printOutput);  
}else{
put(response, "error", "Document Output was empty.");  
}
print("PRINT COMPLETE");  
print("");  
return response;

## **Commerce Section**

## **1 US#8341**

### **1.1 Commerce Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Label** | **Variable Name** | **Description** | **Data Type** |
| PO Number | pONumber\_t | This text attribute shows the customer's PO number in the quote. | TextAttribute |
| PO Date | pODate\_t | This date attribute displays the PO date in the quote. | DateAttribute |

## **1.2 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide Bottom Line Discount Amt | hideBottomLineDiscountAmt | This hiding rule hides the bottom line discount amount if BMDT equals amount. |

## **1.3 Validation rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Credit Check Validation | creditCheckValidation | This validation rule fires when credit check status is either denied or crossed limit. |

## **1.4 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| calculateDiscount\_BP | calculateDiscount\_BP | This function calculates the total discount applicable. |

### **Script Text**

startDate= "2018-02-01";  
endDate= "2018-02-15";  
startDate1=strtodate(startDate, "yyyy-mm-dd");  
endDate1=strtodate(endDate, "yyyy-mm-dd");  
discount=0.0;  
result = "";  
for line in transactionline
{
dateFromUI = "";  
if(line.orderDate <> "")
{
dateFromUI = line.orderDate;  
} elif (line.orderDateSS <> "") {
dateFromUI = line.orderDateSS;  
}
print startDate1;  
print dateFromUI ;  
if(dateFromUI <> "")
{
diffA=comparedates(strtodate(dateFromUI , "yyyy-mm-dd %H:%M:%S"), startDate1);  
diffB=comparedates(strtodate(dateFromUI , "yyyy-mm-dd %H:%M:%S"), endDate1);  
if((diffA == 1 OR diffA == 0)AND(diffB == -1 OR diffB == 0))
{
discount = 50.0;  
}
}
totalDiscountPrice1 = ((discount/100)\*(line.listPrice\_l));  
totalContractValue = line.listPrice\_l-totalDiscountPrice1;  
}
result = result + "1"+ "~totalDiscountPercent\_quote~"+ String(discount) + "|";  
//totalContractValue\_t = listPrice\_l-totalDiscountPrice1 ;  
result = result + "1"+ "~totalDiscount\_quote~"+ String(totalDiscountPrice1 ) + "|";  
//result=result+"1"+"
result = result + "1"+ "~totalContractValue\_t~"+ String(totalContractValue ) + "|";  
return result;

## **1.5 Commerce Actions**

|  |  |  |
| --- | --- | --- |
| **Action** | **Action Variable Name** | **Description** |
| Submit | submit\_t | This action is to submit the quote for approval, if required. If not required, the quote is auto approved. |

## **2 US#1003**

### **2.1 Commerce Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Label** | **Variable Name** | **Description** | **Data Type** |
| Attach PO | attachPO\_t | This file attachment attribute allows to browse & attach the PO file from local. | AttachmentAttribute |
| Project Cost | projectCost\_t | This currency attribute displays the expected project cost considering all the factors. | CurrencyAttribute |

## **2.2 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Hide Asset Fields | hideAssetFields | This rule hides the asset fields if user is "plohani". |

## **2.3 Validation rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Discount Percentage | discountPercentage | Validates that Discount amount is <100 if discount type is "Percent Off" |

## **2.4 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| Process JSON To Table | CN\_ProcessJsonToTable | This function converts JSON string into a table format. |

### **Script Text**

diffSize = 0;  
diffArray = jsonarray("[]");  
// turn json string to json object
if(find(jsonInput, "diffs")>0){
jobj = json(jsonInput);  
str = jsonget(jobj, "diffs");  
print("str");  
print(str);  
jary = jsonarray(str);  
print("DOBJ");  
print(jary);  
diffSize = jsonarraysize(jary);  
diffArray = jary;  
}
if(diffSize < 1){
return "";  
}
// table html builder
tableString = "<table class='cn\_compare\_table'>";  
  
rowCounter = 2;  
index = 0;  
strArray = String[diffSize];  
// for each diff returned in json
for str in strArray {
diffStr = jsonarrayget(diffArray, index);  
diffObj = json(diffStr);  
inserted = false;  
 updated = false;  
 removed = false;  
 moved = false;  
// Clause Label
clauseLabel = trim(jsonget(diffObj, "clauseLabel"));  
clauseLabel = replace(clauseLabel, "&", "&#38;  
");  
clauseLabel = replace(clauseLabel, "<", "&#60;  
");  
clauseLabel = replace(clauseLabel, ">", "&#62;  
");  
// Parent Clause Label
parentLabel = trim(jsonget(diffObj, "parentLabel"));  
parentLabel = replace(parentLabel, "&", "&#38;  
");  
parentLabel = replace(parentLabel, "<", "&#60;  
");  
parentLabel = replace(parentLabel, ">", "&#62;  
");  
// Are Changes Reviewed?
reviewed = trim(jsonget(diffObj, "reviewed"));  
// Change Type Label
action = trim(jsonget(diffObj, "action"));  
if(find(action, "Inserted")>-1){
inserted = true;  
}elif(find(action, "Updated")>-1){
updated = true;  
}elif(find(action, "Removed")>-1){
removed = true;  
}elif(find(action, "Moved")>-1){
moved = true;  
  
claz = "bgcolor-list-even";  
 if(rowCounter % 2 <> 0) {claz = "bgcolor-list-odd";  
}
if(inserted) {claz = claz+" "+"insert";  
} if(updated) {claz = claz+" "+"update";  
}
if(removed) {claz = claz+" "+"remove";  
} if(moved) {claz = claz+" "+"move";  
}
tableString = tableString + "<tr class='"+claz+"'>" + cellString + "</tr>";  
rowCounter = rowCounter + 1;  
index = index + 1;  
}
tableString = tableString + "</table>";  
return tableString;

## **2.5 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| Library to Get BOM | libraryToGetBOM | This function generates a JSON containing BOM details. |

### **Script Text**

retVal=json();  
bsId=atoi(bs\_id);  
lineFieldArr=String[];  
for line in transactionLine{
documentNumber=atoi(line.\_document\_number);  
if(line.\_model\_name<>""){
retVal=getbom(bsId, documentNumber,lineFieldArr, true, false, false);  
}
}
return retVal;

## **2.6 Commerce Actions**

|  |  |  |
| --- | --- | --- |
| **Action** | **Action Variable Name** | **Description** |
| Calculate Discounts | calculateDiscounts\_t | This action calculates the net price of either a part or entire model based on the discounts applied. |

## **2.7 Printer Friendly Documents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Name** | **Variable Name** | **Description** | **Commerce Process Linked** |
| Backup of Proposal Quote | backupOfProposalQuote | This document is the backup/copy of the quote proposal with the existing values. It can be used as reference for new proposal generation. | oraclecpqo |

## **3 US#1234**

### **3.1 Commerce Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Label** | **Variable Name** | **Description** | **Data Type** |
| Customer Type | customerType\_t | This menu attribute displays the type of customer. | MenuAttribute |
| Requested Price | requestedPrice\_quote | This currency attribute shows the requested price. | CurrencyAttribute |
| Approved Price | approvedPrice\_quote | This currency attribute displays the price approved by approvers. | CurrencyAttribute |

## **3.2 Constraint Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Test | test | This contraint rule requires the user to fill the title details. |

## **3.3 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Tax Exempt | taxExempt | This hiding rule hides the tax exempt reason attribute if tax exempt is false. |

## **3.4 Validation rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Quote without OPTY | quoteWithoutOPTY | This validation rule fires when no opportunity is linked with a particular quote. |

## **3.5 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| Discount Price BP | discountPrice\_Bp | This function calculates the discounted price based on the given discount on list price. |

### **Script Text**

for line in transactionline
{
totalDiscountPrice1 = ((totalDiscountPercent\_quote/100)\*(line.listPrice\_l));  
}
return totalDiscountPrice1;

## **3.6 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| Discount Percentage\_BP | discountPercentage\_BP | This function calculates the discount based on start date & end date. |

### **Script Text**

startDate= "2018-02-01";  
endDate= "2018-02-15";  
startDate1=strtodate(startDate, "yyyy-mm-dd");  
endDate1=strtodate(endDate, "yyyy-mm-dd");  
discount=0.0;  
for line in transactionline
{
dateFromUI = "";  
if(line.orderDate <> "")
{
dateFromUI = line.orderDate;  
} elif (line.orderDateSS <> "") {
dateFromUI = line.orderDateSS;  
}
print startDate1;  
print dateFromUI ;  
if(dateFromUI <> "")
{
diffA=comparedates(strtodate(dateFromUI , "yyyy-mm-dd %H:%M:%S"), startDate1);  
diffB=comparedates(strtodate(dateFromUI , "yyyy-mm-dd %H:%M:%S"), endDate1);  
if((diffA == 1 OR diffA == 0)AND(diffB == -1 OR diffB == 0))
{
discount = 50.0;  
}
}
}
return discount;

## **3.7 Commerce Actions**

|  |  |  |
| --- | --- | --- |
| **Action** | **Action Variable Name** | **Description** |
| Save | cleanSave\_t | This action saves the latest values in the quote. |
| Update Opportunity | saveAndSyncItems | This action updates the CRM opportunity with the values added/changed in CPQ. |

## **3.8 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| Discount More Than 10%-SS | discountMoreThan10SS | Discount More Than 10% |  |  |

### **Advanced Condition**

if(totalMonthlyDiscount\_t <> 0 AND totalMonthlyListAmount\_t <> 0)
{
discPercent = (totalMonthlyDiscount\_t/totalMonthlyListAmount\_t)\*100;  
if(discPercent > 10 AND discPercent <= 20){
return true;  
}
else
{
return false;  
}
}
else {
return false;  
};

## **3.9 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| Maverick Reason | maverickReason | Maverick Reason |  |  |

## **4 US#5678**

### **4.1 Commerce Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Label** | **Variable Name** | **Description** | **Data Type** |
| Promotional Discount | promotionalDiscount | This text attribute shows the promotional discount value. | TextAttribute |
| Discount Percentage | discountPercentage | This text attribute displays the discount in percentage. | TextAttribute |

## **4.2 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Line Item Grid: Discounting | lineItemGridDiscounting | Hiding rule for Line Item Grid attributes and columns related to item discounting. |

## **4.3 Validation rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Create Order Checklist | createOrderChecklist | This validation rule triggers when order checklist is incomplete. |

## **4.4 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| Total Contract Value BP | totalContractValueBP | This function calculates the total contract value after the discounts. |

### **Script Text**

for line in transactionline
{
totalContractValueV1 = (line.listPrice\_l-totalDiscount\_quote);  
}
return totalContractValueV1;

## **4.5 Commerce Actions**

|  |  |  |
| --- | --- | --- |
| **Action** | **Action Variable Name** | **Description** |
| Generate Proposal | generateProposal | This action generates the quote proposal which is sent to customer for review. |

## **4.6 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| 50% Discount | a50Discount\_bp | 50% Discount |  |  |

## **4.7 Printer Friendly Documents**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Name** | **Variable Name** | **Description** | **Commerce Process Linked** |
| Proposal | proposal | This document serves as the proposal to the customer. | oraclecpqo |

## **5 US#7661**

### **5.1 Commerce Attributes**

|  |  |  |  |
| --- | --- | --- | --- |
| **Attribute Label** | **Variable Name** | **Description** | **Data Type** |
| Total Amount | totalAmount | This attribute displays the total amount payable by customer after discount. | TextAttribute |
| Customer Contract | CN\_CustomerContract\_t | This file attachment attribute allows to browse & attach the file from local system. | AttachmentAttribute |

## **5.2 Hiding Rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Line Item Grid: Contract | lineItemGridPeriodicity | Hiding rule for Line Item Grid attributes and columns related to item pricing periods. |

## **5.3 Validation rule**

|  |  |  |
| --- | --- | --- |
| **Rule Name** | **Variable Name** | **Description** |
| Customer Details | customerDetails | This validation rule triggers when account number is empty. |

## **5.4 Commerce Libraries**

|  |  |  |
| --- | --- | --- |
| **Library Name** | **Variable Name** | **Description** |
| updateDiscountFields | updateDiscountFields | This function updates the discount fields based on contract dates. |

### **Script Text**

tempVal = "";  
discountP = 0;  
rs = recordset();  
price = "";  
door = "";  
discountAmount = 0;  
totalPrice = "";  
discountPrice = 0;  
result = "";  
startDate= "2018-02-01";  
endDate= "2018-02-15";  
startDate1=strtodate(startDate, "yyyy-mm-dd");  
endDate1=strtodate(endDate, "yyyy-mm-dd");  
for line in transactionLine{
if(line.\_model\_name<>"") {
tempVal = line.orderDate;  
door = getconfigattrvalue(line.\_document\_number,"trgBanMaColor");  
}
}
if(tempVal <> ""){
diffA=comparedates(strtodate(tempVal, "yyyy-mm-dd %H:%M:%S"), startDate1);  
diffB=comparedates(strtodate(tempVal, "yyyy-mm-dd %H:%M:%S"), endDate1);  
if((diffA == 1 OR diffA == 0)AND(diffB == -1 OR diffB == 0)){
discountP = 10;  
}
}
print discountP;  
result = result + "1" + "~discountPercentage~" + string(discountP)+ "|";  
rs = BMQL("select Pricing from parts where Filter = $door" );  
for record in rs{
price = get(record,"Pricing");  
}
print price;  
discountPrice = atof(price)\*discountP/100;  
result = result + "1"+ "~promotionalDiscount~" + string(discountPrice) + "|";  
discountAmount = atof(price)- (atof(price)\*discountP/100);  
print discountAmount;  
result = result + "1" + "~totalAmount~" + string(discountAmount)+ "|";  
return result;

## **5.5 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| Discount Above Maximum | discountAboveMaximum | Discount Above Maximum Reason | Manager User | Discount Approval |

### **Advanced Condition**

result=false;  
/\*\*
for line in transactionLine{
if(line.customDiscountType\_l=="Percent Off"){
if(line.customDiscountValue\_l>line.maxDiscount\_l){
result=true;  
}
}elif(line.customDiscountType\_l=="Amount Off"){
if(line.customDiscountValue\_l<>0 AND line.listPrice\_l<>0){
disc=(line.customDiscountValue\_l/line.listPrice\_l)\*100;  
if(disc>line.maxDiscount\_l){
result=true;  
}
}
}elif(line.customDiscountType\_l=="Price Override"){
if(line.customDiscountValue\_l<>0 AND line.customDiscountValue\_l<=line.listPrice\_l AND line.listPrice\_l<>0){
Off=line.listPrice\_l-line.customDiscountValue\_l;  
fDisc=(Off/line.listPrice\_l)\*100;  
if(fDisc>line.maxDiscount\_l){
result=true;  
}
}
}
}
\*\*/
return result;

## **5.6 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| Buy Price Adjustment | pRMBuyPriceAdjustment | Buy Price Adjustment Reason |  | PRM Buy Price Adj Approval |

### **Advanced Condition**

result=false;  
/\*\*
for line in transactionLine{
if(line.buyPriceDiscount\_l<>0.00){
result=true;  
}
}
\*\*/
return result;

## **5.7 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| Discount less than 10% | a10 | Discount less than 10 % |  |  |

### **Advanced Condition**

submitVal = true;  
if(totalMonthlyDiscount\_t <> 0 AND totalMonthlyListAmount\_t <> 0)
{
discPercent = (totalMonthlyDiscount\_t/totalMonthlyListAmount\_t)\*100;  
if(discPercent > 0 AND discPercent <= 10){
submitVal = true;  
}
else
{
submitVal = true;  
}
}
else
{
submitVal = true;  
}
return submitVal;

## **Data Table List**

|  |  |
| --- | --- |
| **Data Table Name** | **Data Table Columns** |
| Status | category , action , id , StatusOSC\_t , StatusLabel , StatusOSC\_l , HistoricalStatus , StatusID |
| osImages | category , action , id , operatingSystem , imageURL |
| monitoringPricing | category , action , id , Monitoring , pricetype , priceValue |
| ServerModules | category , action , id , Part , Module |
| ServerPerformance | category , action , id , Performance , Chassis , Processor , RAM , HDD , PSU , OperatingSystem |
| osVolumePricing | category , action , id , OS , Part , Comment , minQty , maxQty , Price |
| Processors\_ | category , action , id , Part , Processor |
| Packages | category , action , id , Package , Performance , Chassis , PSU , Processor , RAM , RamConfig , HD , OperatingSystem , NumProcessors |
| Support\_v | category , action , id , Part , SupportLevel , basePrice |
| SoftwarePackage | category , action , id , Part , Software , LicensePrice , MonthlyMaint |
| ServiceDetails | category , action , id , Part , Resource , Service , Rate , Price |
| PowerSupply | category , action , id , Part , PowerSupply |
| HardDrives | category , action , id , Part , HardDrive |
| Services | category , action , id , Service , ServiceDetail |
| CloudBackup | category , action , id , Part\_FeeType , FeeType , Part\_StorageAmt , CloudSpace , Comment |
| RAM | category , action , id , Part , RAM |
| RackSize | category , action , id , Part , Rack , MaxSize |
| INT\_SYSTEM\_DETAILS | category , action , id , System , Username , Endpoint , MaxLinesInPayload , Password |
| INT\_SYSTEM\_TEMPLATES | category , action , id , System , Operation , Template |
| LandingPages | category , action , id , Model , HTML , Mobile |
| PeriodConversations | category , action , id , custom\_part\_field4 , FusionValue , FusionLabel |
| PERIODIC\_CONVERSIONS | category , action , id , PartField , FusionValue , FusionLabel |
| MODEL\_CONVERSIONS | category , action , id , ModelLabel , ModelVarName , ItemId , OrgId , FeeType , Family , Cost |
| LandingPagesMobile | category , action , id , Model , HTML |
| RateFactor | category , action , id , Term , Rate , Yield |
| Financing | category , action , id , FinOptions , Term , Interest |
| pricing | category , action , id , entityId , state , entitytype , part , price |
| TestHAr | category , action , id , Test |
| CountryDetails | category , action , id , CountryName , State , Address , Phone , ZIpcode , Image , Answer , Question |
| DataToolAuth | category , action , id , UserName , Access |
| customerDetails | category , action , id , name , age , address , phoneNo |
| companyDetails | category , action , id , comapanyName , companyAddress , TelNumber , FaxNumber |
| employeeDetails | category , action , id , employeeName , Location , EmpId , ContactNumber |
| ModelFeatures | category , action , id , Frequency\_Band , Display\_Type , Mode |
| LanguageTranslations | category , action , id , English , French |
| ChicagoDataCenter | category , action , id , Name , Capacity , Serial\_Number |
| AryTable | category , action , id , Type , Country , State , City |
| Men\_Clothing | category , action , id , ModelNumber , Image |
| Customer | category , action , id , Name , LifeInsurance , Income , LoanRequested |
| Part | category , action , id , processor , operatingsystem , memory , software , Discount |
| PartPrice | category , action , id , Part , Value , Attribute , Price , Quantity |
| Parts | category , action , id , partNumber , Filter , Team , Quantity , Pricing |
| Employee\_Details | category , action , id , Name , PhoneNumber , DateofBirth , Salary |
| CN\_Credentials | category , action , id , integrationName , username , password , desc |
| PetPrices | category , action , id , Type , PartNumber , PetPrice |
| TestTable | category , action , id , Column1 |
| Oracle\_BomItemDef | category , action , id , VariableName , ItemId , Name , ItemType , PartNumber , Optional , SalesItem , ParentVariableName , RootVariableName , EffectiveFrom , EffectiveTo , ManufacturingItem , IncludedInBasePrice , SequenceNum , DefaultQuantity |
| Oracle\_BomItemMap | category , action , id , VariableName , BomItemVarName , ConfigAttrVarName , ConfigAttrValue , ParentBomMapVarName , EffectiveFrom , EffectiveTo , ConfigAttrOptionVal |
| POC\_Parts | category , action , id , Opportunity , Part\_Num |
| Vertiv\_InternalParts | category , action , id , BOMItemName , ParentBOMItemName , Attr1 , Op1 , Val1 , Attr2 , Op2 , Val2 , Qty |
| Contract\_IDS | category , action , id , Contract\_ID , Contract\_Name , Oppty\_ID |
| Oracle\_aboPart2Model | category , action , id , PartNumber , ProductLine , Segment , Model |
| Vertiv\_Documents | category , action , id , ProductID , DocumentName , DocumentPath |

## **Users List**

|  |  |
| --- | --- |
| **User ID** | **User Login Name** |
| 3 | superuser |
| 7 | superuser |
| 250519 | guest |
| 4119220 | superuser |
| 4119231 | guest |
| 36178341 | guest |
| 36257935 | Admin |
| 36257929 | inga.lang |
| 36279101 | mcadambi |
| 36271184 | laurent.dumont |
| 36394889 | chart |
| 36295001 | mandrews |
| 36295419 | demoadmin |
| 36379801 | luly |
| 36393264 | tferrie |
| 36295901 | hmak |
| 36295106 | tferrie2 |
| 36295111 | jbrinkman |
| 36298237 | cdetrych |
| 36302902 | lisa.jones |
| 36302908 | mateo.lopez |
| 36376201 | ggingerich |
| 36388213 | bpetrik |
| 36393846 | lsturz |
| 36442119 | ri.yang |
| 36442125 | sam.brown |
| 36442131 | takahiro.tayama |
| 36443102 | taro.hiraiwa |
| 36443120 | wei.wang |
| 36462029 | ejacobson |
| 36462035 | mmorman |
| 36403801 | mweinberg |
| 36442093 | liang.yang |
| 36442099 | lucy.harris |
| 36442106 | oliver.brunke |
| 36442112 | paul.dubois |
| 36488102 | owen.mann |
| 36481028 | lche |
| 36488076 | superuser |
| 36488082 | guest |
| 36497302 | provision |
| 36499241 | system\_user\_cpq\_server\_as\_client |
| 36547572 | colby.morris |
| 36554841 | david.wilson |
| 36742701 | ajledoux |
| 36948236 | dchang |
| 36995791 | yjiao |
| 36982277 | guest1 |
| 36997649 | stwinam |
| 37028560 | sraykar |
| 37028937 | gpulipati |
| 37163204 | rwest |
| 37163210 | sbahukhandi |
| 37163216 | speddireddy |
| 37163235 | spandhari |
| 37163240 | anandssingh |
| 37317499 | icsuser |
| 37320886 | snloya |
| 37320892 | amakuluri |
| 37321258 | sguo |
| 37029525 | julian.henderson |
| 37163228 | abinash |
| 37163247 | abkazi |
| 37163252 | mrinalk |
| 37163257 | rsharmas |
| 38441959 | bas |
| 37163262 | fosinubi |
| 37163267 | begeorge |
| 37163272 | shruti |
| 37163277 | NGupta |
| 37317207 | pagupta |
| 37336587 | rkunal |
| 37380763 | Prashant |
| 37327988 | subhdas |
| 37327998 | vepotti |
| 37324030 | vbelawade |
| 37327993 | jshaik |
| 37331965 | hrajupet |
| 37340252 | mkasturiwale |
| 37340259 | bracha |
| 37324138 | jswaminathan |
| 37324143 | bbhushan |
| 37321291 | hlee |
| 38442935 | rajakolli |
| 40770623 | rprabhakar |
| 41475905 | vsoni |
| 41586899 | Adwiti |
| 37422061 | pkrishnan |
| 37443657 | rfaigao |
| 37443662 | jbade |
| 38442892 | priyatam |
| 38442912 | ramyap |
| 38442969 | Nandita |
| 38442989 | Pallavi |
| 38882822 | kkim |
| 40514945 | traininguser |
| 40706542 | veramaswamy |
| 37386772 | adfGen |
| 40121332 | ogonzalez |
| 40688319 | plohani |
| 40706552 | bgong |
| 40779994 | Somya |
| 37387938 | ldesmet |
| 37494165 | sayanka |
| 37504721 | Test1 |
| 37504726 | Test2 |
| 37509124 | abc1 |
| 37509129 | mno1 |
| 37514283 | User2 |
| 37514288 | User3 |
| 37538971 | aaganesan |
| 37918267 | ctiwari |
| 38138007 | bflechsig |
| 38238630 | pvajjhala |
| 37388161 | pirathi |
| 37642896 | ssharma |
| 38280727 | spathade |
| 38659694 | ABhoi |
| 38714296 | rjadhwani |
| 38714306 | brjohnson |
| 40118527 | lsoto |
| 40350280 | ajain |
| 40752129 | Sirisha |
| 40975107 | aalibhai@deloitte.com |
| 41099875 | Shiwam |
| 41362736 | Teja |
| 40749551 | sradovic |

## **Groups List**

|  |  |
| --- | --- |
| **Group Label** | **Group Name** |
| Sales User | salesUser |
| Business Administrator | businssAdminstrator |
| Technical Administrator | technicalAdminstrator |
| Manager User | managerUser |
| VP Users | vPUsers |
| Tablets | tablets |
| Email Group | emailGroup |
| Data Manage | dataManagement |
| Federal Users | federalUsers |
| Alta UI Users | altaUIUsers |
| Green Dots Auto Parts | greenDotsAutoParts |
| Order Management | orderManagement |
| Executive Sales Manager | execSalesMgr |
| Vertiv UI Users | vertivUIUsers |