# **Table of Contents**

[**Table of Contents** 1](#_Toc532474813)

[**Document Overview and Description** 5](#_Toc532474814)

[**1.1 Login and Profiles** 5](#_Toc532474815)

[**1.2 Audience** 5](#_Toc532474816)

[**1.3 Key Terms** 5](#_Toc532474817)

[**1.4 Design Assumptions** 5](#_Toc532474818)

[**1.5 Key Dependencies** 5](#_Toc532474819)

[**1.6 Related Documentation** 5](#_Toc532474820)

[**Implementation Area** 7](#_Toc532474821)

[**Configuration Section** 7](#_Toc532474822)

[**1 US#1234** 7](#_Toc532474823)

[**1.1 Configuration Attributes** 7](#_Toc532474824)

[**1.2 Rules** 7](#_Toc532474825)

[**1.3 Recommendation Rule** 7](#_Toc532474826)

[**1.4 Constraint Rule** 8](#_Toc532474827)

[**1.5 Hiding Rule** 8](#_Toc532474828)

[**1.6 Recommended Items rule** 8](#_Toc532474829)

[**1.7 Configuration Flow rule** 8](#_Toc532474830)

[**1.8 BML Util Libraries** 8](#_Toc532474831)

[**2 US#5678** 11](#_Toc532474832)

[**2.1 Configuration Attributes** 11](#_Toc532474833)

[**2.2 Rules** 11](#_Toc532474834)

[**2.3 Recommendation Rule** 11](#_Toc532474835)

[**2.4 Constraint Rule** 11](#_Toc532474836)

[**2.5 Hiding Rule** 11](#_Toc532474837)

[**2.6 Recommended Items rule** 12](#_Toc532474838)

[**2.7 Configuration Flow rule** 12](#_Toc532474839)

[**2.8 Pricing rule** 12](#_Toc532474840)

[**2.9 BML Util Libraries** 12](#_Toc532474841)

[**3 US#7661** 14](#_Toc532474842)

[**3.1 Configuration Attributes** 14](#_Toc532474843)

[**3.2 Rules** 14](#_Toc532474844)

[**3.3 Recommendation Rule** 14](#_Toc532474845)

[**3.4 Constraint Rule** 14](#_Toc532474846)

[**3.5 Hiding Rule** 14](#_Toc532474847)

[**3.6 Recommended Items rule** 15](#_Toc532474848)

[**3.7 BML Util Libraries** 15](#_Toc532474849)

[**4 US#8341** 17](#_Toc532474850)

[**4.1 Configuration Attributes** 17](#_Toc532474851)

[**4.2 Rules** 17](#_Toc532474852)

[**4.3 Recommendation Rule** 17](#_Toc532474853)

[**4.4 Constraint Rule** 17](#_Toc532474854)

[**4.5 Hiding Rule** 17](#_Toc532474855)

[**4.6 BML Util Libraries** 18](#_Toc532474856)

[**5 US#1003** 20](#_Toc532474857)

[**5.1 Configuration Attributes** 20](#_Toc532474858)

[**5.2 Rules** 20](#_Toc532474859)

[**5.3 Recommendation Rule** 20](#_Toc532474860)

[**5.4 Constraint Rule** 20](#_Toc532474861)

[**5.5 Hiding Rule** 20](#_Toc532474862)

[**5.6 Recommended Items rule** 21](#_Toc532474863)

[**5.7 Configuration Flow rule** 21](#_Toc532474864)

[**5.8 Pricing rule** 21](#_Toc532474865)

[**5.9 BML Util Libraries** 21](#_Toc532474866)

[**Commerce Section** 24](#_Toc532474867)

[**1 US#8341** 24](#_Toc532474868)

[**1.1 Commerce Attributes** 24](#_Toc532474869)

[**1.2 Rules** 24](#_Toc532474870)

[**1.3 Hiding Rule** 24](#_Toc532474871)

[**1.4 Validation Rule** 24](#_Toc532474872)

[**1.5 Commerce Libraries** 25](#_Toc532474873)

[**1.6 Commerce Actions** 25](#_Toc532474874)

[**1.7 Commerce Steps** 26](#_Toc532474875)

[**1.8 Integration** 26](#_Toc532474876)

[**2 US#1003** 32](#_Toc532474877)

[**2.1 Commerce Attributes** 32](#_Toc532474878)

[**2.2 Rules** 32](#_Toc532474879)

[**2.3 Hiding Rule** 32](#_Toc532474880)

[**2.4 Validation Rule** 32](#_Toc532474881)

[**2.5 Commerce Libraries** 32](#_Toc532474882)

[**2.6 Commerce Actions** 34](#_Toc532474883)

[**2.7 Commerce Steps** 35](#_Toc532474884)

[**2.8 Printer Friendly Documents** 36](#_Toc532474885)

[**2.9 Integration** 36](#_Toc532474886)

[**3 US#1234** 39](#_Toc532474887)

[**3.1 Commerce Attributes** 39](#_Toc532474888)

[**3.2 Rules** 39](#_Toc532474889)

[**3.3 Constraint Rule** 39](#_Toc532474890)

[**3.4 Hiding Rule** 39](#_Toc532474891)

[**3.5 Validation Rule** 39](#_Toc532474892)

[**3.6 Commerce Libraries** 40](#_Toc532474893)

[**3.7 Commerce Actions** 40](#_Toc532474894)

[**3.8 Approval Sequence** 41](#_Toc532474895)

[**3.9 Commerce Steps** 41](#_Toc532474896)

[**3.10 Integration** 44](#_Toc532474897)

[**4 US#5678** 48](#_Toc532474898)

[**4.1 Commerce Attributes** 48](#_Toc532474899)

[**4.2 Rules** 48](#_Toc532474900)

[**4.3 Hiding Rule** 48](#_Toc532474901)

[**4.4 Validation Rule** 48](#_Toc532474902)

[**4.5 Commerce Libraries** 48](#_Toc532474903)

[**4.6 Commerce Actions** 49](#_Toc532474904)

[**4.7 Approval Sequence** 49](#_Toc532474905)

[**4.8 Commerce Steps** 49](#_Toc532474906)

[**4.9 Printer Friendly Documents** 50](#_Toc532474907)

[**4.10 Integration** 51](#_Toc532474908)

[**5 US#7661** 53](#_Toc532474909)

[**5.1 Commerce Attributes** 53](#_Toc532474910)

[**5.2 Rules** 53](#_Toc532474911)

[**5.3 Hiding Rule** 53](#_Toc532474912)

[**5.4 Validation Rule** 53](#_Toc532474913)

[**5.5 Commerce Libraries** 53](#_Toc532474914)

[**5.6 Approval Sequence** 54](#_Toc532474915)

[**5.7 Commerce Steps** 56](#_Toc532474916)

[**5.8 Integration** 57](#_Toc532474917)

[**File Manager** 62](#_Toc532474918)

[**1 US#8341** 62](#_Toc532474919)

[**1.1 WebServices : UpdateTransactionv2.0.xml** 62](#_Toc532474920)

[**1.2 Javascript : sitewide.js** 63](#_Toc532474921)

[**2 US#1003** 64](#_Toc532474922)

[**2.1 WebServices : updateTransactionFrame.xml** 64](#_Toc532474923)

[**3 US#1234** 65](#_Toc532474924)

[**3.1 WebServices : addtoTransaction.xml** 65](#_Toc532474925)

[**3.2 Javascript : commerce.js** 66](#_Toc532474926)

[**4 US#5678** 68](#_Toc532474927)

[**4.1 WebServices : getTransactionFrame.xml** 68](#_Toc532474928)

[**4.2 Javascript : config.js** 69](#_Toc532474929)

[**5 US#7661** 70](#_Toc532474930)

[**5.1 WebServices : addToTransactionFrame.xml** 70](#_Toc532474931)

[**5.2 WebServices : UpdateTransactionforQuoteLevelAttributes.xml** 70](#_Toc532474932)

[**5.3 Javascript : homepage.js** 72](#_Toc532474933)

[**Data Table List** 73](#_Toc532474934)

[**Users List** 76](#_Toc532474935)

[**Groups List** 81](#_Toc532474936)

# **Document Overview and Description**

### **1.1 Login and Profiles**

The purpose of this document is to provide detailed technical specifications of the CPQ Implementation in Oracle CPQ including details on core configuration, portal, web site and integrations.

### **1.2 Audience**

The intended audience of this document is all CPQ Implementation IT Team Members involved in requirement definition, design, configuration, or testing of the Oracle CPQ.

### **1.3 Key Terms**

The intended audience of this document is all CPQ Implementation IT Team Members involved in requirement definition, design, configuration, or testing of the Oracle CPQ.

|  |  |
| --- | --- |
| **Acronym** | **Description** |
| UAT | User Acceptance Testing |
| CPQ | Configure Pricing Quoting |
| DEV | Development Environment in CPQ |
| PROD | Production Environment in CPQ |
| BML | Big Machines Language |
| CSV | Comma Separated Value |
| API | Application Programming Interface |
| HTTP | Hypertext Transfer Protocol |
| WSDL | Web Services Description Language |

### **1.4 Design Assumptions**

### **1.5 Key Dependencies**

1) Quality of the migrated data will depend on the quality of the legacy data provided and the extent of any data cleansing.

2) Custom development and business workflow is dependent by CPQ limits. These limitations have been considered during design and can be monitored from within the Oracle CPQ.

### **1.6 Related Documentation**

The following table lists additional documents that contain related information and supporting program material

|  |  |
| --- | --- |
| **Title** | **Description** |
| Integration Strategy & Approach | Deliverable documenting the proposed integration with other boundary systems. |
| Environment Strategy & Approach | Deliverable that describes the approach to development, test and production environment management and release |

# **Implementation Area**

The following high-level section (configuration, commerce, integration, doc designer) defines the underline structure for the Oracle CPQ.

# **Configuration Section**

Configuration is the process of allowing users to customize a model to suit their needs.Using Configuration, the buyer can customize the items they are buying by specifying the values of configuration attributes.

## **1 US#1234**

### **1.1 Configuration Attributes**

Configuration attributes define the characteristics of Product Families that can be specified by the user. They are used in search flows, configuration flows, and every type of configuration rule. Attributes can be of float, integer, date, HTML or text data type, Menus.

Configuration attributes can be defined as stand-alone attributes or as arrays. Array attributes group multiple scalar attributes. Usually, the size of an array attribute is specified by a control attribute.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Rules enable you to set specific configuration information for your products. The user experience in configuration is controlled by the ability to fire the appropriate rules based on user input. Different types of rules enhance the selection and configuration experience.

For example, you can suggest configured values to buyers using recommendation rules or hide some configuration attributes from certain user types. Rules can also be used to recommend items to buyers, prevent them from selecting opposite values of configuration attributes and display additional price information.

### **1.3 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. |

### **1.4 Constraint Rule**

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

### **1.5 Hiding Rule**

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

### **1.6 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. |

### **1.7 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. |

### **1.8 BML Util Libraries**

BML libraries are extremely important in that they allow us to avoid duplicate code, and to easily maintain discrete logical operations. They are different than Advanced Modify functions and Advance Validation rules, and it may take some extra effort to reproduce errors.

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

**Script**

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;

**BML Util Libraries**

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

**Script**

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**

Configuration attributes define the characteristics of Product Families that can be specified by the user. They are used in search flows, configuration flows, and every type of configuration rule. Attributes can be of float, integer, date, HTML or text data type, Menus.

Configuration attributes can be defined as stand-alone attributes or as arrays. Array attributes group multiple scalar attributes. Usually, the size of an array attribute is specified by a control attribute.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Rules enable you to set specific configuration information for your products. The user experience in configuration is controlled by the ability to fire the appropriate rules based on user input. Different types of rules enhance the selection and configuration experience.

For example, you can suggest configured values to buyers using recommendation rules or hide some configuration attributes from certain user types. Rules can also be used to recommend items to buyers, prevent them from selecting opposite values of configuration attributes and display additional price information.

### **2.3 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. |

### **2.4 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. |

### **2.5 Hiding Rule**

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

### **2.6 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. |

### **2.7 Configuration Flow rule**

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

### **2.8 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. |

### **2.9 BML Util Libraries**

BML libraries are extremely important in that they allow us to avoid duplicate code, and to easily maintain discrete logical operations. They are different than Advanced Modify functions and Advance Validation rules, and it may take some extra effort to reproduce errors.

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

**Script**

//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;

**BML Util Libraries**

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

**Script**

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**

Configuration attributes define the characteristics of Product Families that can be specified by the user. They are used in search flows, configuration flows, and every type of configuration rule. Attributes can be of float, integer, date, HTML or text data type, Menus.

Configuration attributes can be defined as stand-alone attributes or as arrays. Array attributes group multiple scalar attributes. Usually, the size of an array attribute is specified by a control attribute.

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

### **3.2 Rules**

Rules enable you to set specific configuration information for your products. The user experience in configuration is controlled by the ability to fire the appropriate rules based on user input. Different types of rules enhance the selection and configuration experience.

For example, you can suggest configured values to buyers using recommendation rules or hide some configuration attributes from certain user types. Rules can also be used to recommend items to buyers, prevent them from selecting opposite values of configuration attributes and display additional price information.

### **3.3 Recommendation Rule**

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

### **3.4 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. |

### **3.5 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 |

### **3.6 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. |

### **3.7 BML Util Libraries**

BML libraries are extremely important in that they allow us to avoid duplicate code, and to easily maintain discrete logical operations. They are different than Advanced Modify functions and Advance Validation rules, and it may take some extra effort to reproduce errors.

|  |  |  |
| --- | --- | --- |
| **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**

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

**BML Util Libraries**

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

**Script**

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**

Configuration attributes define the characteristics of Product Families that can be specified by the user. They are used in search flows, configuration flows, and every type of configuration rule. Attributes can be of float, integer, date, HTML or text data type, Menus.

Configuration attributes can be defined as stand-alone attributes or as arrays. Array attributes group multiple scalar attributes. Usually, the size of an array attribute is specified by a control attribute.

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

### **4.2 Rules**

Rules enable you to set specific configuration information for your products. The user experience in configuration is controlled by the ability to fire the appropriate rules based on user input. Different types of rules enhance the selection and configuration experience.

For example, you can suggest configured values to buyers using recommendation rules or hide some configuration attributes from certain user types. Rules can also be used to recommend items to buyers, prevent them from selecting opposite values of configuration attributes and display additional price information.

### **4.3 Recommendation Rule**

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

### **4.4 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 |

### **4.5 Hiding Rule**

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

### **4.6 BML Util Libraries**

BML libraries are extremely important in that they allow us to avoid duplicate code, and to easily maintain discrete logical operations. They are different than Advanced Modify functions and Advance Validation rules, and it may take some extra effort to reproduce errors.

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

**Script**

// 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**

Configuration attributes define the characteristics of Product Families that can be specified by the user. They are used in search flows, configuration flows, and every type of configuration rule. Attributes can be of float, integer, date, HTML or text data type, Menus.

Configuration attributes can be defined as stand-alone attributes or as arrays. Array attributes group multiple scalar attributes. Usually, the size of an array attribute is specified by a control attribute.

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

### **5.2 Rules**

Rules enable you to set specific configuration information for your products. The user experience in configuration is controlled by the ability to fire the appropriate rules based on user input. Different types of rules enhance the selection and configuration experience.

For example, you can suggest configured values to buyers using recommendation rules or hide some configuration attributes from certain user types. Rules can also be used to recommend items to buyers, prevent them from selecting opposite values of configuration attributes and display additional price information.

### **5.3 Recommendation Rule**

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

### **5.4 Constraint Rule**

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

### **5.5 Hiding Rule**

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

### **5.6 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. |

### **5.7 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. |

### **5.8 Pricing rule**

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

### **5.9 BML Util Libraries**

BML libraries are extremely important in that they allow us to avoid duplicate code, and to easily maintain discrete logical operations. They are different than Advanced Modify functions and Advance Validation rules, and it may take some extra effort to reproduce errors.

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

**Script**

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;

**BML Util Libraries**

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

**Script**

// 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**

Commerce Processes are used to create templates for the selling processes used by your company. With Commerce Processes, you set up your quoting, ordering, approval, and other workflow processes. By creating ordered sets of Commerce document templates, along with associated attributes and actions, you enable buyers and supplier agents to conduct transactions on your customized CPQ Cloud application.

## **1 US#8341**

### **1.1 Commerce Attributes**

Commerce attributes are the foundation for input fields for Commerce documents. They apply to a Transaction as a whole, or its specific line items, depending on in which Commerce document the attribute is created.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Commerce rules allow you to do things like limit the values in one field based on the selections in another and hide attributes based on other selections.

### **1.3 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. |

### **1.4 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. |

### **1.5 Commerce Libraries**

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

**Script**

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;

### **1.6 Commerce Actions**

Commerce actions trigger events within documents to occur. The Commerce system creates a default set of actions for use in each document, and generates additional actions when Commerce documents include certain attribute types.

|  |  |  |
| --- | --- | --- |
| **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. |

### **1.7 Commerce Steps**

Workflow administration is the final step in setting up a Commerce Process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce Processes can have any number of workflow steps. Workflow steps use profiles to define access rights, transition notification, and transaction views.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step Name** | **Description** | **Variable Name** | **Participant Profile Name** | **Profile Description** | **Transition Rule** |
| Ordered | The order has been sent to the fulfillment system. | orderedBeingFulfilled | Admin | Default Participant Profile. | Next |
| Ordered | The order has been sent to the fulfillment system. | orderedBeingFulfilled | Admin | Default Participant Profile. | Previous |
| Rejected By Customer | The customer decided not to move forwards with the proposed transaction. | rejectedByCustomer | Admin | Default Participant Profile. |  |

### **1.8 Integration**

Integration for software applications, it typically refers to a process that tries to link, or share, data from different software applications. These applications include CRM and ERP software.

There are many benefits to integration, including:

• Streamlining the marketing and sales processes

• Organizing and updating customer information

• Preventing manual data entry

• Removing redundant, unnecessary or tedious everyday steps

The CPQ Cloud application can integrate with different types of applications, in different ways. Some integration needs can be handled by the default functionality within the software packages themselves, with some additional configuration. Other functionality may need to be built on a custom basis, depending on the business's needs and the limitations of the software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Integration Name** | **Variable Name** | **Description** | **ID Field** | **Endpoint URL** |
| Sales Cloud - Upsert Quote | oSCUpsertQuote | This integration update/insert quote attributes with current values from attached opportunity. | transaction.quoteID\_t | https://AAAA-FAPBBBB-crm.oracledemos.com/crmCommonSalesOrdersService/SalesOrderService |

**Integration Details**

SOAP Generator XSL

<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format" version="1.0">  
<xsl:output method="xml"/>  
<xsl:template name="sanitizeCurrency">  
<xsl:param name="val" />  
<xsl:choose>  
<xsl:when test="$val != ''">  
<xsl:choose>  
<xsl:when test="contains($val, ',')">  
<xsl:value-of select="substring-before($val, ',')" />  
<xsl:text>.</xsl:text>  
<xsl:value-of select="substring-after($val, ',')" />  
</xsl:when>  
<xsl:otherwise>  
<xsl:value-of select="$val"/>  
</xsl:otherwise>  
</xsl:choose>  
</xsl:when>  
<xsl:otherwise>0</xsl:otherwise>  
</xsl:choose>  
</xsl:template>  
<xsl:template match="/">  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@document\_number=1]"/>  
<xsl:variable name="documentAvailable">  
<xsl:choose>  
<xsl:when test="string-length($main\_doc/oSCFileAttachment) = 0">  
<xsl:value-of select="'false'"/>  
</xsl:when>  
<xsl:otherwise>  
<xsl:value-of select="'true'"/>  
</xsl:otherwise>  
</xsl:choose>  
</xsl:variable>  
<!-- Begin SOAP Envelope For Oracle Sales Cloud - Upsert Quote Export Integration -->  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://xmlns.oracle.com/apps/crmCommon/orders/salesOrdersService/types/" xmlns:sal="http://xmlns.oracle.com/apps/crmCommon/orders/salesOrdersService/">  
<soapenv:Body>  
<xsl:choose>  
<xsl:when test="string-length($main\_doc/oSCQuoteId) = 0">  
<typ:createSalesOrderHeader>  
<typ:salesOrderHeader>  
<sal:OptyId>  
<xsl:value-of select="$main\_doc/opportunityID\_t"/>  
</sal:OptyId>  
<sal:Name>  
<xsl:value-of select="$main\_doc/transactionName\_t"/>  
</sal:Name>  
<sal:ExternalReferenceNumber>  
<xsl:value-of select="$main\_doc/bs\_id"/>  
</sal:ExternalReferenceNumber>  
<sal:ExternalSystemReferenceCode>BMQUOTE</sal:ExternalSystemReferenceCode>  
<sal:ProposalExistFlag>  
<xsl:value-of select="$documentAvailable"/>  
</sal:ProposalExistFlag>  
<sal:WinStatusCode>  
<xsl:value-of select="$main\_doc/statusOSC\_t"/>  
</sal:WinStatusCode>  
<sal:VersionNumber>  
<xsl:value-of select="$main\_doc/version\_number\_versionTransaction\_t"/>  
</sal:VersionNumber>  
<sal:OrderTotal>  
<xsl:choose>  
<xsl:when test="($main\_doc/userType\_t) ='BuyAccess'">  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$main\_doc/totalExtBuyPrice\_t"/>  
</xsl:call-template>  
</xsl:when>  
<xsl:otherwise>  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$main\_doc/totalContractValue\_t"/>  
</xsl:call-template>  
</xsl:otherwise>  
</xsl:choose>  
</sal:OrderTotal>  
<sal:CurrencyCode>  
<xsl:value-of select="$main\_doc/currency\_t"/>  
</sal:CurrencyCode>  
<sal:SoldCustomerPartyId>  
<xsl:value-of select="$main\_doc/\_customer\_id"/>  
</sal:SoldCustomerPartyId>  
<sal:LastOptySyncDate>  
<xsl:value-of select="$main\_doc/oSCSyncTime"/>  
</sal:LastOptySyncDate>  
<sal:Status>  
<xsl:value-of select="$main\_doc/status\_t"/>  
</sal:Status>  
</typ:salesOrderHeader>  
</typ:createSalesOrderHeader>  
</xsl:when>  
<xsl:otherwise>  
<typ:updateSalesOrderHeader>  
<typ:salesOrderHeader>  
<sal:OptyId>  
<xsl:value-of select="$main\_doc/opportunityID\_t"/>  
</sal:OptyId>  
<sal:Name>  
<xsl:value-of select="$main\_doc/transactionName\_t"/>  
</sal:Name>  
<sal:ExternalReferenceNumber>  
<xsl:value-of select="$main\_doc/bs\_id"/>  
</sal:ExternalReferenceNumber>  
<sal:ExternalSystemReferenceCode>BMQUOTE</sal:ExternalSystemReferenceCode>  
<sal:ProposalExistFlag>  
<xsl:value-of select="$documentAvailable"/>  
</sal:ProposalExistFlag>  
<sal:WinStatusCode>  
<xsl:value-of select="$main\_doc/statusOSC\_t"/>  
</sal:WinStatusCode>  
<sal:VersionNumber>  
<xsl:value-of select="$main\_doc/version\_number\_versionTransaction\_t"/>  
</sal:VersionNumber>  
<sal:OrderTotal>  
<xsl:choose>  
<xsl:when test="($main\_doc/userType\_t) ='BuyAccess'">  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$main\_doc/totalExtBuyPrice\_t"/>  
</xsl:call-template>  
</xsl:when>  
<xsl:otherwise>  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$main\_doc/totalContractValue\_t"/>  
</xsl:call-template>  
</xsl:otherwise>  
</xsl:choose>  
</sal:OrderTotal>  
<sal:CurrencyCode>  
<xsl:value-of select="$main\_doc/currency\_t"/>  
</sal:CurrencyCode>  
<sal:SoldCustomerPartyId>  
<xsl:value-of select="$main\_doc/\_customer\_id"/>  
</sal:SoldCustomerPartyId>  
<sal:LastOptySyncDate>  
<xsl:value-of select="$main\_doc/oSCSyncTime"/>  
</sal:LastOptySyncDate>  
<sal:Status>  
<xsl:value-of select="$main\_doc/status\_t"/>  
</sal:Status>  
</typ:salesOrderHeader>  
</typ:updateSalesOrderHeader>  
</xsl:otherwise>  
</xsl:choose>  
</soapenv:Body>  
</soapenv:Envelope>  
<!-- End SOAP XML -->  
</xsl:template>  
</xsl:stylesheet>

Result Parser XSL

<?xml version="1.0" encoding="UTF-8"?>  
<!-- ################# Upsert Quote - Oracle Sales Cloud - Result Parser XSL ################# -->  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="/oracle/apps/sales/custExtn/views/common/">  
<xsl:output method="xml"/>  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@data\_type=0]"/>  
<xsl:template match="/">  
<data\_xml>  
<document document\_var\_name="transaction">  
<!-- ############## Set oracleCustomObject2QuoteInstanceID\_quote Value ########### -->  
<oSCQuoteId><xsl:value-of select="//\*[local-name()='OrderHeaderId']"/></oSCQuoteId>  
</document>  
</data\_xml>  
</xsl:template>  
</xsl:stylesheet>

## **2 US#1003**

### **2.1 Commerce Attributes**

Commerce attributes are the foundation for input fields for Commerce documents. They apply to a Transaction as a whole, or its specific line items, depending on in which Commerce document the attribute is created.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Commerce rules allow you to do things like limit the values in one field based on the selections in another and hide attributes based on other selections.

### **2.3 Hiding Rule**

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

### **2.4 Validation Rule**

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

### **2.5 Commerce Libraries**

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

**Script**

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'>";  
// HEADING ROW tableString = tableString + "<tr>" +//"<td>Clause Id</td>" + //"<td>Action Type</td>" + "<th>Clause</th>" + "<th>Section</th>" + "<th>Reviewed</th>" + "<th>Action</th>" + "</tr>";  
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;  
} cellString = "<td>"+clauseLabel+"</td>" + "<td>"+parentLabel+"</td>" + "<td>"+reviewed+"</td>" + "<td>"+action+"</td>";  
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;

**Commerce Libraries**

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

**Script**

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**

Commerce actions trigger events within documents to occur. The Commerce system creates a default set of actions for use in each document, and generates additional actions when Commerce documents include certain attribute types.

|  |  |  |
| --- | --- | --- |
| **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 Commerce Steps**

Workflow administration is the final step in setting up a Commerce Process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce Processes can have any number of workflow steps. Workflow steps use profiles to define access rights, transition notification, and transaction views.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step Name** | **Description** | **Variable Name** | **Participant Profile Name** | **Profile Description** | **Transition Rule** |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | Saved Transaction |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | submit - no approval needed |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | submit - approval needed |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | Delete Transaction |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | Previous |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | Alternate Order: Pending -> Alternate |
| Read-Only Step | The transaction is being edited prior to sending to the customer or ordering. | pending\_process\_bmClone\_1 | Admin | Admin | Next |

### **2.8 Printer Friendly Documents**

Printer Friendly/ History XSL views allow you to control the display of a transaction depending on the XSL template you select. You can create multiple XSL templates to produce different types of printer friendly proposal formats or transaction history outputs. These XSL templates can be predefined in the actions used to generate the printer friendly proposal or history output. Alternately, you can allow the user to select templates at run time to generate varied outputs.

You can generate output as XML, RTF, PDF, XLS, HTML, or DOCX.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |

### **2.9 Integration**

Integration for software applications, it typically refers to a process that tries to link, or share, data from different software applications. These applications include CRM and ERP software.

There are many benefits to integration, including:

• Streamlining the marketing and sales processes

• Organizing and updating customer information

• Preventing manual data entry

• Removing redundant, unnecessary or tedious everyday steps

The CPQ Cloud application can integrate with different types of applications, in different ways. Some integration needs can be handled by the default functionality within the software packages themselves, with some additional configuration. Other functionality may need to be built on a custom basis, depending on the business's needs and the limitations of the software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Integration Name** | **Variable Name** | **Description** | **ID Field** | **Endpoint URL** |
| Sales Cloud - Flip Opportunity | salesCloudFlipOpportunity | Changes the opportunity status to Closed Won upon clicking Create Order. Oracle Sales Cloud Release 9+ | quote\_process.none | https://AAAA-FAPBBBB-crm.oracledemos.com/opptyMgmtOpportunities/OpportunityService |

**Integration Details**

SOAP Generator XSL

<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format" version="1.0">  
<xsl:output method="xml"/>  
<xsl:template match="/">  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@document\_number=1]"/>  
<!-- Begin SOAP Envelope For Sales Cloud-->  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/types/" xmlns:opp="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/" xmlns:rev="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/" xmlns:not="http://xmlns.oracle.com/apps/crmCommon/notes/noteService" xmlns:not1="http://xmlns.oracle.com/apps/crmCommon/notes/flex/noteDff/" xmlns:rev1="http://xmlns.oracle.com/oracle/apps/sales/opptyMgmt/revenues/revenueService/" xmlns:act="http://xmlns.oracle.com/apps/crmCommon/activities/activitiesService/">  
<soapenv:Header/>  
<soapenv:Body>  
<typ:updateOpportunity>  
<xsl:choose>  
<xsl:when test="string-length($main\_doc/opportunityID\_t) != 0">  
<typ:opportunity>  
<opp:OptyId><xsl:value-of select="$main\_doc/opportunityID\_t"/></opp:OptyId>  
<opp:SalesStageId>300000000127579</opp:SalesStageId>  
<opp:ReasonWonLostCode>PRODUCT</opp:ReasonWonLostCode>  
<opp:StatusCode>WON</opp:StatusCode>  
</typ:opportunity>  
</xsl:when>  
<xsl:otherwise>  
<typ:opportunity>  
<opp:OptyId>300000066841395</opp:OptyId>  
</typ:opportunity>  
</xsl:otherwise>  
</xsl:choose>  
</typ:updateOpportunity>  
</soapenv:Body>  
</soapenv:Envelope>  
<!-- End SOAP XML -->  
</xsl:template>  
</xsl:stylesheet>

Result Parser XSL

<?xml version="1.0" encoding="UTF-8"?>  
<!-- ################# Upsert Quote - Oracle CRMOD Integration - Result Parser XSL ################# -->  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<xsl:output method="xml"/>  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@data\_type=0]"/>  
<xsl:template match="/">  
<data\_xml>  
<document document\_var\_name="quote\_process">  
<!-- ############## Set oracleCustomObject2QuoteInstanceID\_quote Value ########### -->  
<crmQuoteId\_quote><xsl:value-of select="//\*[local-name()='OrderHeaderId']"/></crmQuoteId\_quote>  
</document>  
</data\_xml>  
</xsl:template>  
</xsl:stylesheet>

## **3 US#1234**

### **3.1 Commerce Attributes**

Commerce attributes are the foundation for input fields for Commerce documents. They apply to a Transaction as a whole, or its specific line items, depending on in which Commerce document the attribute is created.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Commerce rules allow you to do things like limit the values in one field based on the selections in another and hide attributes based on other selections.

### **3.3 Constraint Rule**

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

### **3.4 Hiding Rule**

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

### **3.5 Validation Rule**

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

### **3.6 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**

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

**Commerce Libraries**

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

**Script**

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**

Commerce actions trigger events within documents to occur. The Commerce system creates a default set of actions for use in each document, and generates additional actions when Commerce documents include certain attribute types.

|  |  |  |
| --- | --- | --- |
| **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;  
 };

**Approval Sequence**

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

### **3.9 Commerce Steps**

Workflow administration is the final step in setting up a Commerce Process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce Processes can have any number of workflow steps. Workflow steps use profiles to define access rights, transition notification, and transaction views.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step Name** | **Description** | **Variable Name** | **Participant Profile Name** | **Profile Description** | **Transition Rule** |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Request Approval Submit Action |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Saved Transaction |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | submit - no approval needed |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | submit - approval needed |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Next |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Send Email |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Alternate Order: Pending -> Alternate |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Delete Transaction |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | In Progress --> Read-Only Step |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Submit Quote --> Waiting for Approval |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Email Users | Email Users | Previous |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | Saved Transaction |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | Next |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | Previous |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | Alternate Order: Pending -> Alternate |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | In Progress --> Read-Only Step |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | submit - approval needed |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | Delete Transaction |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | Submit Quote --> Waiting for Approval |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Admin | Admin | submit - no approval needed |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | Saved Transaction |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | submit - no approval needed |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | submit - approval needed |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | Alternate Order: Pending -> Alternate |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | Customer Rejected |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | Cancel Transaction |
| In Progress | The transaction is being edited prior to sending to the customer or ordering. | pending\_process | Manager | Manager | Delete Transaction |

**Advanced Forwarding Rule**

return true;

**Advanced Condition of Transition Rule:**

Profile Description: Manager, Transition Rule Name: Delete Transaction

Script:

return true;

### **3.10 Integration**

Integration for software applications, it typically refers to a process that tries to link, or share, data from different software applications. These applications include CRM and ERP software.

There are many benefits to integration, including:

• Streamlining the marketing and sales processes

• Organizing and updating customer information

• Preventing manual data entry

• Removing redundant, unnecessary or tedious everyday steps

The CPQ Cloud application can integrate with different types of applications, in different ways. Some integration needs can be handled by the default functionality within the software packages themselves, with some additional configuration. Other functionality may need to be built on a custom basis, depending on the business's needs and the limitations of the software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Integration Name** | **Variable Name** | **Description** | **ID Field** | **Endpoint URL** |
| Sales Cloud - Opportunity Import | oSCOppImport | This integration imports the opportunity details into CPQ. | transaction.opportunityID\_t | https://AAAA-FAPBBBB-crm.oracledemos.com/opptyMgmtOpportunities/OpportunityService |

**Integration Details**

SOAP Generator XSL

<?xml version="1.0" encoding="utf-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:typ="http://www.w3.org/1999/XSL/Format" version="1.0">  
<xsl:output method="xml"/>  
<xsl:template match="/">  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@document\_number=1]"/>  
<!-- Begin SOAP Envelope For Oracle Sales Cloud -->  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/types/" xmlns:typ1="http://xmlns.oracle.com/adf/svc/types/">  
<soapenv:Body>  
<ns1:getOpportunity xmlns:ns1="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/types/">  
<ns1:optyId><xsl:value-of select="$main\_doc/opportunityID\_t"/></ns1:optyId>  
</ns1:getOpportunity>  
</soapenv:Body>  
</soapenv:Envelope>  
<!-- End SOAP XML -->  
</xsl:template>  
</xsl:stylesheet>

Result Parser XSL

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sb1="urn:crmondemand/ws/account/10/2004" xmlns:ns3="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/" xmlns:ns2="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/" version="1.0">  
<xsl:output method="xml"/>  
<xsl:template match="/">  
<data\_xml>  
<document document\_var\_name="transaction">  
<opportunityName\_t>  
<xsl:value-of select="//ns3:Name"/>  
</opportunityName\_t>  
<\_customer\_t\_last\_name>  
<xsl:value-of select="//ns2:PrimaryContactLastName"/>  
</\_customer\_t\_last\_name>  
<\_customer\_t\_first\_name>  
<xsl:value-of select="//ns2:PrimaryContactFirstName"/>  
</\_customer\_t\_first\_name>  
<\_invoiceTo\_t\_first\_name>  
<xsl:value-of select="//ns2:PrimaryContactFirstName"/>  
</\_invoiceTo\_t\_first\_name>  
<\_invoiceTo\_t\_last\_name>  
<xsl:value-of select="//ns2:PrimaryContactLastName"/>  
</\_invoiceTo\_t\_last\_name>  
<opportunityNumber\_t>  
<xsl:value-of select="//ns2:OptyNumber"/>  
</opportunityNumber\_t>  
<oSCPrimaryOrgID\_t>  
<xsl:value-of select="//ns2:PrimaryOrganizationId"/>  
</oSCPrimaryOrgID\_t>  
<oSCOwnerResourcePartyId\_t>  
<xsl:value-of select="//ns2:OwnerResourcePartyId"/>  
</oSCOwnerResourcePartyId\_t>  
<currency\_t>  
<xsl:value-of select="//ns2:CurrencyCode"/>  
</currency\_t>  
<partnerLevel\_t>  
<xsl:value-of select="//ns3:PriceList\_c"/>  
</partnerLevel\_t>  
</document>  
</data\_xml>  
</xsl:template>  
</xsl:stylesheet>

## **4 US#5678**

### **4.1 Commerce Attributes**

Commerce attributes are the foundation for input fields for Commerce documents. They apply to a Transaction as a whole, or its specific line items, depending on in which Commerce document the attribute is created.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Commerce rules allow you to do things like limit the values in one field based on the selections in another and hide attributes based on other selections.

### **4.3 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. |

### **4.4 Validation Rule**

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

### **4.5 Commerce Libraries**

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

**Script**

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

### **4.6 Commerce Actions**

Commerce actions trigger events within documents to occur. The Commerce system creates a default set of actions for use in each document, and generates additional actions when Commerce documents include certain attribute types.

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

### **4.7 Approval Sequence**

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

### **4.8 Commerce Steps**

Workflow administration is the final step in setting up a Commerce Process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce Processes can have any number of workflow steps. Workflow steps use profiles to define access rights, transition notification, and transaction views.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step Name** | **Description** | **Variable Name** | **Participant Profile Name** | **Profile Description** | **Transition Rule** |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Email Users | Default Participant Profile. | Next |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Email Users | Default Participant Profile. | Alternate Order: Waiting for Approval -> Alternate |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Email Users | Default Participant Profile. | revise (submit) waiting for approval -> pending |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Email Users | Default Participant Profile. | Previous |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Admin | US#5678 Default Participant Profile. | Next |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Admin | US#5678 Default Participant Profile. | Alternate Order: Waiting for Approval -> Alternate |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Admin | US#5678 Default Participant Profile. | revise (submit) waiting for approval -> pending |
| Waiting for Approval | The transaction is locked awaiting approval from management. | waitingForInternalApproval | Admin | US#5678 Default Participant Profile. | Previous |

### **4.9 Printer Friendly Documents**

Printer Friendly/ History XSL views allow you to control the display of a transaction depending on the XSL template you select. You can create multiple XSL templates to produce different types of printer friendly proposal formats or transaction history outputs. These XSL templates can be predefined in the actions used to generate the printer friendly proposal or history output. Alternately, you can allow the user to select templates at run time to generate varied outputs.

You can generate output as XML, RTF, PDF, XLS, HTML, or DOCX.

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

### **4.10 Integration**

Integration for software applications, it typically refers to a process that tries to link, or share, data from different software applications. These applications include CRM and ERP software.

There are many benefits to integration, including:

• Streamlining the marketing and sales processes

• Organizing and updating customer information

• Preventing manual data entry

• Removing redundant, unnecessary or tedious everyday steps

The CPQ Cloud application can integrate with different types of applications, in different ways. Some integration needs can be handled by the default functionality within the software packages themselves, with some additional configuration. Other functionality may need to be built on a custom basis, depending on the business's needs and the limitations of the software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Integration Name** | **Variable Name** | **Description** | **ID Field** | **Endpoint URL** |
| Sales Cloud - Delete All Revenue Items | oSCDeleteAllRevenueItems | This CRM integration deletes all revenue items from CRM system. | transaction.none | https://AAAA-FAPBBBB-crm.oracledemos.com/opptyMgmtOpportunities/OpportunityService |

**Integration Details**

SOAP Generator XSL

<?xml version="1.0" encoding="utf-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format" version="1.0">  
<xsl:output method="xml"/>  
<xsl:template match="/">  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@document\_number=1]"/>  
<!-- Begin SOAP Envelope For Oracle Sales Cloud -->  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:typ="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/types/">  
<soapenv:Header/>  
<soapenv:Body>  
<typ:deleteAllRevenuesInOpportunity>  
<typ:optyId><xsl:value-of select="$main\_doc/opportunityID\_t"/></typ:optyId>  
</typ:deleteAllRevenuesInOpportunity>  
</soapenv:Body>  
</soapenv:Envelope>  
<!-- End SOAP XML -->  
</xsl:template>  
</xsl:stylesheet>

Result Parser XSL

<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:sb1="urn:crmondemand/ws/account/10/2004" xmlns:ns3="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/">  
<xsl:output method="xml"/>  
<xsl:template match="/">  
<data\_xml>  
<document document\_var\_name="transaction">  
<opportunityName\_t>  
<xsl:value-of select="//ns3:Name"/>  
</opportunityName\_t>  
</document>  
</data\_xml>  
</xsl:template>  
</xsl:stylesheet>

## **5 US#7661**

### **5.1 Commerce Attributes**

Commerce attributes are the foundation for input fields for Commerce documents. They apply to a Transaction as a whole, or its specific line items, depending on in which Commerce document the attribute is created.

|  |  |  |  |
| --- | --- | --- | --- |
| **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 Rules**

Commerce rules allow you to do things like limit the values in one field based on the selections in another and hide attributes based on other selections.

### **5.3 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. |

### **5.4 Validation Rule**

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

### **5.5 Commerce Libraries**

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

**Script**

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;

### **5.6 Approval Sequence**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Approval Reason Name** | **Approval Reason Variable Name** | **Approval Reason Description** | **Approver** | **Approval Template** |
| Discount Above Maximum | discountAboveMaximum | Discount Above Maximum Reason | VP Users | Discount Approval |

**Advanced Condition**

result=false;  
for line in transactionLine{ if(line.customDiscountType\_l=="Percent Off"){ if(line.customDiscountValue\_l>line.maxManagerDiscount\_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;  
if(disc>line.maxManagerDiscount\_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;  
if(fDisc>line.maxManagerDiscount\_l){ result=true;  
} } } } return result;

**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**

ret = false;  
for line in transactionLine{ if ((partnerLevel\_t == "Gold Partner Pricebook") AND (line.buyPriceDiscount\_l <> 10.0)){ ret = true;  
}elif ((partnerLevel\_t == "Silver Partner Pricebook") AND (line.buyPriceDiscount\_l <> 5.0)){ ret = true;  
}elif ((partnerLevel\_t == "Platinum Partner Pricebook") AND (line.buyPriceDiscount\_l <> 20.0)){ ret = true;  
} } return ret;

**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;

### **5.7 Commerce Steps**

Workflow administration is the final step in setting up a Commerce Process. Workflows are made up of steps, which define the states transactions can be in, as well as document permissions and routing. Commerce Processes can have any number of workflow steps. Workflow steps use profiles to define access rights, transition notification, and transaction views.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Step Name** | **Description** | **Variable Name** | **Participant Profile Name** | **Profile Description** | **Transition Rule** |
| Start | Default Step | start\_step | Email Users | Default Participant Profile. | Save -> In Progress |
| Start | Default Step | start\_step | Email Users | Default Participant Profile. | Save Quote |
| Start | Default Step | start\_step | Email Users | Default Participant Profile. | Previous |
| Start | Default Step | start\_step | Email Users | Default Participant Profile. | Next |
| Approved | The transaction has all necessary internal approvals. | approved | Email Users | Email Users | Next |
| Approved | The transaction has all necessary internal approvals. | approved | Email Users | Email Users | Previous |
| Approved | The transaction has all necessary internal approvals. | approved | Email Users | Email Users | Cancel Order |
| Approved | The transaction has all necessary internal approvals. | approved | Email Users | Email Users | revise (submit): approved -> In Progress |
| Approved | The transaction has all necessary internal approvals. | approved | Email Users | Email Users | Alternate Order: Approved -> Alternate |
| Approved | The transaction has all necessary internal approvals. | approved | Email Users | Email Users | Request Signature |
| Approved | The transaction has all necessary internal approvals. | approved | Admin | Admin | Next |
| Approved | The transaction has all necessary internal approvals. | approved | Admin | Admin | Previous |
| Approved | The transaction has all necessary internal approvals. | approved | Admin | Admin | Cancel Order |
| Approved | The transaction has all necessary internal approvals. | approved | Admin | Admin | revise (submit): approved -> In Progress |
| Approved | The transaction has all necessary internal approvals. | approved | Admin | Admin | Alternate Order: Approved -> Alternate |
| Approved | The transaction has all necessary internal approvals. | approved | Admin | Admin | Request Signature |

### **5.8 Integration**

Integration for software applications, it typically refers to a process that tries to link, or share, data from different software applications. These applications include CRM and ERP software.

There are many benefits to integration, including:

• Streamlining the marketing and sales processes

• Organizing and updating customer information

• Preventing manual data entry

• Removing redundant, unnecessary or tedious everyday steps

The CPQ Cloud application can integrate with different types of applications, in different ways. Some integration needs can be handled by the default functionality within the software packages themselves, with some additional configuration. Other functionality may need to be built on a custom basis, depending on the business's needs and the limitations of the software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Integration Name** | **Variable Name** | **Description** | **ID Field** | **Endpoint URL** |
| Sales Cloud - Create Revenue Items | oSCCreateRevenueItems | This integration creates revenue items in CRM system. | transaction.none | https://AAAA-FAPBBBB-crm.oracledemos.com/opptyMgmtOpportunities/OpportunityService |

**Integration Details**

SOAP Generator XSL

<?xml version="1.0" encoding="UTF-8"?>  
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:fo="http://www.w3.org/1999/XSL/Format" version="1.0">  
<xsl:output method="xml"/>  
<xsl:template name="sanitizeCurrency">  
<xsl:param name="val" />  
<xsl:choose>  
<xsl:when test="$val != ''">  
<xsl:choose>  
<xsl:when test="contains($val, ',')">  
<xsl:value-of select="substring-before($val, ',')" />  
<xsl:text>.</xsl:text>  
<xsl:value-of select="substring-after($val, ',')" />  
</xsl:when>  
<xsl:otherwise>  
<xsl:value-of select="$val"/>  
</xsl:otherwise>  
</xsl:choose>  
</xsl:when>  
<xsl:otherwise>0</xsl:otherwise>  
</xsl:choose>  
</xsl:template>  
<xsl:template match="/">  
<xsl:variable name="main\_doc" select="/transaction/data\_xml/document[@document\_number=1]"/>  
<!-- Begin SOAP Envelope For Oracle Sales Cloud -->  
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
<soap:Body>  
<ns1:mergeOpportunity xmlns:ns1="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/types/">  
<ns1:opportunity>  
<ns21:OptyId xmlns:ns21="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/">  
<xsl:value-of select="$main\_doc/opportunityID\_t"/>  
</ns21:OptyId>  
<xsl:for-each select="/transaction/data\_xml/document[@data\_type=2 or @data\_type=3]">  
<xsl:sort select="normalize-space(\_sequence\_number)" data-type="number"/>  
<xsl:variable name="sub\_doc" select="."/>  
<xsl:if test="normalize-space($sub\_doc/parentLineNumber\_l) = ''">  
<ns100:ChildRevenue xmlns:ns100="http://xmlns.oracle.com/apps/sales/opptyMgmt/opportunities/opportunityService/">  
<ns106:UnitPrice xmlns:ns106="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$sub\_doc/modelEstimatedPrice\_l"/>  
</xsl:call-template>  
</ns106:UnitPrice>  
<ns107:Quantity xmlns:ns107="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:value-of select="$sub\_doc/requestedQuantity\_l"/>  
</ns107:Quantity>  
<ns108:RevnAmount xmlns:ns108="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:choose>  
<xsl:when test="($main\_doc/userType\_t) ='BuyAccess'">  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$sub\_doc/modelTotalBuyPrice\_l"/>  
</xsl:call-template>  
</xsl:when>  
<xsl:otherwise>  
<xsl:call-template name="sanitizeCurrency">  
<xsl:with-param name="val" select="$sub\_doc/modelTotalContractValue\_l"/>  
</xsl:call-template>  
</xsl:otherwise>  
</xsl:choose>  
</ns108:RevnAmount>  
<ns109:RevnAmountCurcyCode xmlns:ns109="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:value-of select="$main\_doc/currency\_t"/>  
</ns109:RevnAmountCurcyCode>  
<ns111:InventoryOrgId xmlns:ns111="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:value-of select="$sub\_doc/orgId\_l"/>  
</ns111:InventoryOrgId>  
<ns110:InventoryItemId xmlns:ns110="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:value-of select="$sub\_doc/entityId\_l"/>  
</ns110:InventoryItemId>  
<!-- <ns112:UOMCode xmlns:ns112="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:value-of select="$sub\_doc/requestedUnitOfMeasure\_l"/>  
</ns112:UOMCode> -->  
<ns113:ResourcePartyId xmlns:ns113="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:value-of select="$main\_doc/oSCOwnerResourcePartyId\_t"/>  
</ns113:ResourcePartyId>  
</ns100:ChildRevenue>  
</xsl:if>  
</xsl:for-each>  
</ns1:opportunity>  
</ns1:mergeOpportunity>  
</soap:Body>  
</soap:Envelope>  
<!-- End SOAP XML -->  
</xsl:template>  
</xsl:stylesheet>

Result Parser XSL

<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="http://xmlns.oracle.com/apps/sales/opptyMgmt/revenues/revenueService/">  
<xsl:output method="xml"/>  
<xsl:variable name="ids" select="//ns1:RevnId"/>  
<xsl:template match="/">  
<response>  
<xsl:for-each select="$ids">  
<xsl:sort select="." data-type="number"/>  
<id\_field\_value>  
<xsl:value-of select="."/>  
</id\_field\_value>  
</xsl:for-each>  
</response>  
</xsl:template>  
</xsl:stylesheet>

# **File Manager**

File Manager is an integral part of CPQ Cloud. You can upload any kind of file to the File Manager, organize files into folders, and access files from anywhere on the Internet. The File Manager is used for:

• External images

• JavaScript files linked to various areas on the site

• CSS Stylesheets for Configurations Flow Templates

• Excel spreadsheets (to hold master data)

## **1 US#8341**

### **1.1 WebServices : UpdateTransactionv2.0.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Header>  
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">  
<wsse:UsernameToken wsu:Id="UsernameToken-2">  
<wsse:Username>traininguser</wsse:Username>  
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">kIN2\_sEE7\_</wsse:Password>  
</wsse:UsernameToken>  
</wsse:Security>  
</soapenv:Header>  
<soapenv:Body>  
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
<bm:transaction>  
<bm:id>{{transactionID\_t}}</bm:id>  
<bm:process\_var\_name>redStackDigitalMix</bm:process\_var\_name>  
<bm:buyer\_company\_name>cpq-p10-002</bm:buyer\_company\_name>  
<bm:supplier\_company\_name>cpq-p10-002</bm:supplier\_company\_name>  
<bm:data\_xml>  
<bm:transaction bm:bs\_id="{{transactionID\_t}}" bm:buyer\_company\_name="cpq-p10-002" bm:buyer\_user\_name="traininguser" bm:currency\_pref="USD" bm:data\_type="0" bm:document\_name="Transaction" bm:document\_number="1" bm:document\_var\_name="transaction" bm:process\_var\_name="redStackDigitalMix" bm:supplier\_company\_name="cpq-p10-002">  
<bm:\_document\_number>1</bm:\_document\_number>  
<bm:\_price\_book\_var\_name>\_default\_price\_book</bm:\_price\_book\_var\_name>  
</bm:transaction>  
</bm:data\_xml>  
<bm:action\_data>  
<bm:action\_var\_name>save</bm:action\_var\_name>  
<bm:\_bm\_cm\_new\_transaction\_currency/>  
<bm:performer\_comment/>  
<bm:reason\_var\_name/>  
<bm:performer\_name/>  
<bm:performer\_type/>  
<bm:performer\_company\_name/>  
</bm:action\_data>  
</bm:transaction>  
</bm:updateTransaction>  
</soapenv:Body>  
</soapenv:Envelope>

### **1.2 Javascript : sitewide.js**

/\*\*  
\* @param dependencies {Array} name of modules this code depends on. Can exclude ".js"  
\* @param callback {Function} function containing this module's functionality.  
\* Functions for sitewide context  
\* @version Fri Feb 25 18:44:56 2011  
\*/  
require([], function() {  
/\*  
\* Put all functions for sitewide context here  
\*/  
//this function runs when the page loads  
require.ready(function() {  
});  
});

## **2 US#1003**

### **2.1 WebServices : updateTransactionFrame.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Header>  
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:sessionId>SESSION\_ID</bm:sessionId>  
</bm:userInfo>  
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>  
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:schemaLocation>https://SITE\_NAME.bigmachines.com/bmfsweb/SITE\_NAME/schema/v1\_0/commerce/PROCESS\_VAR\_NAME.xsd</bm:schemaLocation>  
</bm:xsdInfo>  
</soapenv:Header>  
<soapenv:Body>  
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
<bm:transaction>  
<bm:id>TRANSACTION\_ID</bm:id>  
<bm:process\_var\_name>PROCESS\_VAR\_NAME</bm:process\_var\_name>  
<bm:buyer\_company\_name>SITE\_NAME</bm:buyer\_company\_name>  
<bm:supplier\_company\_name>SITE\_NAME</bm:supplier\_company\_name>  
<bm:data\_xml>  
<bm:transaction bm:bs\_id="TRANSACTION\_ID" bm:buyer\_company\_name="SITE\_NAME" bm:buyer\_user\_name="BMI\_API\_user" bm:currency\_pref="USD" bm:data\_type="0" bm:document\_name="Transaction" bm:document\_number="1" bm:document\_var\_name="transaction" bm:process\_var\_name="PROCESS\_VAR\_NAME" bm:supplier\_company\_name="SITE\_NAME">  
<bm:sub\_documents>  
LINE\_PROCESS  
</bm:sub\_documents>  
</bm:transaction>  
</bm:data\_xml>  
<bm:action\_data>  
<bm:action\_var\_name>\_update\_line\_items</bm:action\_var\_name>  
</bm:action\_data>  
</bm:transaction>  
</bm:updateTransaction>  
</soapenv:Body>  
</soapenv:Envelope>

## **3 US#1234**

### **3.1 WebServices : addtoTransaction.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Header>  
<wsse:Security xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd" xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">  
<wsse:UsernameToken wsu:Id="UsernameToken-2">  
<wsse:Username>traininguser</wsse:Username>  
<wsse:Password Type="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-username-token-profile-1.0#PasswordText">kIN2\_sEE7\_</wsse:Password>  
</wsse:UsernameToken>  
</wsse:Security>  
</soapenv:Header>  
<soapenv:Body>  
<bm:addToTransaction xmlns:bm="http://xmlns.oracle.com/cpqcloud/commerce/redStackDigitalMix" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
<bm:items>  
<bm:catalogItem>  
<bm:segment/>  
<bm:product\_line/>  
<bm:model/>  
</bm:catalogItem>  
<bm:partItem>  
<bm:part>{{\_part\_number}}</bm:part>  
<bm:quantity>1</bm:quantity>  
<bm:price\_book\_var\_name>\_default\_price\_book</bm:price\_book\_var\_name>  
</bm:partItem>  
</bm:items>  
<bm:transaction>  
<bm:process\_var\_name>redStackDigitalMix</bm:process\_var\_name>  
<bm:id>{{transactionID\_t}}</bm:id>  
<bm:action\_data>  
<bm:document\_var\_name>transaction</bm:document\_var\_name>  
</bm:action\_data>  
<bm:return\_specific\_attributes>  
<bm:documents>  
<bm:document>  
<bm:var\_name>transaction</bm:var\_name>  
<bm:attributes>  
<bm:attribute>\_document\_number</bm:attribute>  
</bm:attributes>  
</bm:document>  
</bm:documents>  
</bm:return\_specific\_attributes>  
</bm:transaction>  
<bm:include\_transaction>true</bm:include\_transaction>  
</bm:addToTransaction>  
</soapenv:Body>  
</soapenv:Envelope>

### **3.2 Javascript : commerce.js**

/\*\*  
\* @version Fri Dec 2 14:43:33 2011  
\*\*/  
require(["return\_to\_quote\_button", "commerce\_ids", "jquery\_cookie"], function(rtq) {  
var base\_url = base\_url || "/bmfsweb/" + \_BM\_HOST\_COMPANY + "/image/javascript";  
var commissionTotalURL = base\_url + "/commissionTotalsChart.js";  
$.getScript(commissionTotalURL, function(){  
});  
var commissionPriceTotalsChartURL = base\_url + "/commissionPriceTotalsChart.js";  
$.getScript(commissionPriceTotalsChartURL, function(){  
});  
var totalMarginChartjsURL = base\_url + "/totalMarginChartJS\_quote.js";  
$.getScript(totalMarginChartjsURL, function(){  
});  
var costDistributionjsURL = base\_url + "/costDistributionJS\_quote.js";  
$.getScript(costDistributionjsURL, function(){  
});  
var totalCostChartURL = base\_url + "/totalCostChart.js";  
$.getScript(totalCostChartURL, function(){  
});  
var analysisChartURL = base\_url + "/analysisChart.js";  
$.getScript(analysisChartURL, function(){  
});  
var marginChartURL = base\_url + "/marginChart.js";  
$.getScript(marginChartURL, function(){  
});  
var commissionURL = base\_url + "/commission.js";  
$.getScript(commissionURL, function(){  
});  
//this function runs when the page loads  
require.ready(function() {  
rtq.set\_cookie\_in\_commerce();  
});  
});

## **4 US#5678**

### **4.1 WebServices : getTransactionFrame.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Header>  
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:sessionId>SESSION\_ID</bm:sessionId>  
</bm:userInfo>  
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>  
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:schemaLocation>https://SITE\_NAME.bigmachines.com/bmfsweb/SITE\_NAME/schema/v1\_0/commerce/PROCESS\_VAR\_NAME.xsd</bm:schemaLocation>  
</bm:xsdInfo>  
</soapenv:Header>  
<soapenv:Body>  
<bm:getTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
<bm:transaction>  
<bm:id>TRANSACTION\_ID</bm:id>  
<bm:return\_specific\_attributes>  
<bm:documents>  
<bm:document>  
<bm:var\_name>transaction</bm:var\_name>  
<bm:attributes>  
<bm:attribute>\_document\_number</bm:attribute>  
</bm:attributes>  
</bm:document>  
<bm:document>  
<bm:var\_name>transactionLine</bm:var\_name>  
<bm:attributes>  
<bm:attribute>\_document\_number</bm:attribute>  
<bm:attribute>\_sequence\_number</bm:attribute>  
</bm:attributes>  
</bm:document>  
</bm:documents>  
</bm:return\_specific\_attributes>  
</bm:transaction>  
</bm:getTransaction>  
</soapenv:Body>  
</soapenv:Envelope>

### **4.2 Javascript : config.js**

/\*\*  
\* @param dependencies {Array} name of modules this code depends on. Can exclude ".js"  
\* @param callback {Function} function containing this module's functionality.  
\* @version Fri Feb 25 18:44:56 2011  
\*/  
require([], function(rtq) {  
/\*  
\* Put all functions for homepage here  
\*/  
// this function runs when the page loads  
require.ready(function() {  
$('#singleSelectArray .cell-select\_singleSelectArray input').click(function(){  
$("#selectedIndex\_singleSelectArray").val($(this).val());  
});  
$("#previous").click(  
function(){  
$("#previousFlag").val('Yes');  
$("#update").click();  
});  
$("#next").click(  
function(){  
$("#nextFlag").val('Yes');  
$("#update").click();  
});  
$("#partSeacrhFlag").change(  
function(){  
$("#update").click();  
}  
);  
});  
});

## **5 US#7661**

### **5.1 WebServices : addToTransactionFrame.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Header>  
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:sessionId>SESSION\_ID</bm:sessionId>  
</bm:userInfo>  
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>  
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:schemaLocation>https://SITE\_NAME.bigmachines.com/bmfsweb/SITE\_NAME/schema/v1\_0/commerce/PROCESS\_VAR\_NAME.xsd</bm:schemaLocation>  
</bm:xsdInfo>  
</soapenv:Header>  
<soapenv:Body>  
<bm:addToTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
<bm:items>  
ITEMS\_TO\_ADD  
</bm:items>  
<bm:transaction>  
<bm:process\_var\_name>PROCESS\_VAR\_NAME</bm:process\_var\_name>  
<bm:id>TRANSACTION\_ID</bm:id>  
<bm:action\_data>  
<bm:document\_var\_name>DOCUMENT\_VAR\_NAME\_MAIN</bm:document\_var\_name>  
</bm:action\_data>  
</bm:transaction>  
<bm:include\_transaction>true</bm:include\_transaction>  
</bm:addToTransaction>  
</soapenv:Body>  
</soapenv:Envelope>

### **5.2 WebServices : UpdateTransactionforQuoteLevelAttributes.xml**

<?xml version="1.0" encoding="UTF-8"?>  
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">  
<soapenv:Header>  
<bm:userInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:sessionId>{{SESSION\_ID}}</bm:sessionId>  
</bm:userInfo>  
<bm:category xmlns:bm="urn:soap.bigmachines.com">Commerce</bm:category>  
<bm:xsdInfo xmlns:bm="urn:soap.bigmachines.com">  
<bm:schemaLocation>https://cpq-p10-002.bigmachines.com/bmfsweb/cpq-p10-002/schema/v1\_0/commerce/oraclecpqo.xsd</bm:schemaLocation>  
</bm:xsdInfo>  
</soapenv:Header>  
<soapenv:Body>  
<bm:updateTransaction xmlns:bm="urn:soap.bigmachines.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">  
<bm:transaction>  
<bm:id>{{PTRANS\_ID}}</bm:id>  
<bm:process\_var\_name>oraclecpqo</bm:process\_var\_name>  
<bm:buyer\_company\_name>{{BUYER\_COMPANY\_NAME}}</bm:buyer\_company\_name>  
<bm:supplier\_company\_name>{{SUPPLIER\_COMPANY\_NAME}}</bm:supplier\_company\_name>  
<bm:step\_var\_name>pending\_process</bm:step\_var\_name>  
<bm:last\_document\_number>15</bm:last\_document\_number>  
<bm:date\_added>2016-07-18 03:13:51</bm:date\_added>  
<bm:date\_modified>2016-07-18 04:16:46</bm:date\_modified>  
<bm:data\_xml>  
<bm:transaction bm:bs\_id="37309049" bm:buyer\_company\_name="cpq-p10-002" bm:buyer\_user\_name="mrinalk" bm:currency\_pref="USD" bm:data\_type="0" bm:document\_name="Transaction" bm:document\_number="1" bm:document\_var\_name="transaction" bm:process\_var\_name="oraclecpqo" bm:supplier\_company\_name="cpq-p10-002">  
<bm:\_document\_number>1</bm:\_document\_number>  
<bm:createdBy\_t>{{CREATED\_BY}}</bm:createdBy\_t>  
<bm:\_price\_book\_var\_name>\_default\_price\_book</bm:\_price\_book\_var\_name>  
<bm:createdDate\_t>2016-07-18 00:00:00</bm:createdDate\_t>  
<bm:currency\_t>USD</bm:currency\_t>  
<bm:status\_t>{{NEW\_STATUS}}</bm:status\_t>  
<bm:bs\_id>{{PTRANS\_ID}}</bm:bs\_id>  
<!--<bm:version\_number\_versionTransaction\_t>2</bm:version\_number\_versionTransaction\_t>-->  
<bm:parent\_transaction\_id\_versionTransaction\_t/>  
<bm:currentTime>2016-07-18 03:14:05</bm:currentTime>  
<bm:obsolete\_attributes>  
<bm:\_price\_quantity>1</bm:\_price\_quantity>  
<bm:\_price\_book\_desc>Default Price Book (1.0 x base price)</bm:\_price\_book\_desc>  
<bm:\_price\_book\_name>Base Price</bm:\_price\_book\_name>  
</bm:obsolete\_attributes>  
<bm:approval\_history/>  
<bm:approval\_status/>  
<bm:sub\_documents/>  
<bm:rule\_data>  
<bm:bottomLineDiscountPercent\_quote bm:constrained="false" bm:hidden="true"/>  
<bm:taxExemptReason\_t bm:constrained="false" bm:hidden="true"/>  
<bm:commissionLost\_quote bm:constrained="false" bm:hidden="true"/>  
</bm:rule\_data>  
</bm:transaction>  
</bm:data\_xml>  
<bm:action\_data>  
<bm:action\_var\_name>{{ACTION\_VAR\_NAME}}</bm:action\_var\_name>  
<bm:\_bm\_cm\_new\_transaction\_currency/>  
<bm:performer\_comment/>  
<bm:reason\_var\_name/>  
<bm:performer\_name/>  
<bm:performer\_type/>  
<bm:performer\_company\_name/>  
</bm:action\_data>  
<bm:sub\_step\_name/>  
<bm:buyer\_user\_name>mrinalk</bm:buyer\_user\_name>  
<bm:currency\_pref>USD</bm:currency\_pref>  
<bm:status>1</bm:status>  
<bm:update\_count>10</bm:update\_count>  
<bm:language\_pref>English</bm:language\_pref>  
<bm:offline\_user\_id>-1</bm:offline\_user\_id>  
<bm:num\_transitions>2</bm:num\_transitions>  
</bm:transaction>  
</bm:updateTransaction>  
</soapenv:Body>  
</soapenv:Envelope>

### **5.3 Javascript : homepage.js**

/\*\*  
\* @param dependencies {Array} name of modules this code depends on. Can exclude ".js"  
\* @param callback {Function} function containing this module's functionality.  
\* @version Fri Feb 25 18:44:56 2011  
\*/  
require(["return\_to\_quote\_button", "jquery\_cookie"],function(rtq) {  
//Put all functions for homepage here  
//This function runs when the page loads  
require.ready(function() {  
rtq.add\_button\_to\_homepage();  
});  
});  
require(["qs\_homepage"]);

# **Data Table List**

Data Tables provide single-page administration which includes a spreadsheet-like user interface with easy editing capabilities, powerful search features that allow searches within a single table or across all tables, and enhanced performance and indexing.

|  |  |
| --- | --- |
| **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**

All users that require access to CPQ Cloud must have a specific user assigned to them. Users can be created and maintained manually by admins, through bulk upload /download, or through user integration with a CRM system.

Users can be created for the Host Company or Partner Organizations.

Users cannot be deleted, but they can be inactivated.

Both active and inactive Host Company users can be created, activated/inactivated, viewed, or modified on the User Administration List page.

|  |  |
| --- | --- |
| **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**

Groups are a collection of users and are useful for determining access rights for Commerce documents.

|  |  |
| --- | --- |
| **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 |