| Attribute Name | Variable Name | Data Type |
| --- | --- | --- |
| CRM Partner URL | cRMPartnerURL | Text |
| CRM Partner Session Id | cRMPartnerSessionId | Text |
| Tier | serverPerformance | Text |
| Server Module | serverChassis | Text |
| Servers | servers\_quantity | Integer |
| Blades | blades\_quantity | Integer |
| Server Rack | racksEnclosures | Text |
| Processor | processors | Text |
| RAM | ram | Text |
| RAM Configuration | ramConfiguration | Text |
| Number of Processors | numberOfProcessors | Integer |
| Hard Drive | hardDrives | Text |
| Optical Drive | opticalDrive | Text |
| Power Supply | powerSupply | Text |
| Floppy Drive | slimFloppyDrive | Boolean |
| Additional HDD's | additionalHD | Boolean |
| HD Control [Array Control Attribute] | raidControl | Integer |
| Additional Hard Drive | hardDrive\_raid\_array | Text |
| Redundant Power Supply | redundantPowerSupply | Boolean |
| Cloud Storage | cloudStorage | Text |
| Cloud Backup | cloudBackup | Boolean |
| Volume Price Information | volumePriceInformation | HTML |
| OS Licenses | oSLicenses | Integer |
| Software Package | softwarePackage\_array | Text |
| Application Counter [Array Control Attribute] | applicationSoftwareCounter | Integer |
| Licenses | licenses\_array | Integer |
| Maintenance | maintenance\_array | Text |
| Start Date | startDate\_array | Date |
| End Date | endDate\_array | Date |
| Qty | qty\_array | Integer |
| Support Level | supportLevel | Text |
| Rebate | rebate | HTML |
| Monitoring | monitoring | Text |
| Services Requested | servicesRequested | Text |
| Modify Service Details | modifyServiceDetails | Boolean |
| Resource | service\_servArray | Text |
| Service Detail | serviceDetail\_servArray | Text |
| Resources | resources\_servArray | Integer |
| Service Counter [Array Control Attribute] | serviceDetailController | Integer |
| Rate Level | rateLevel\_servArray | Integer |
| Flex | flex\_servArray | Text |
| Price | price\_servArray | Currency |
| Hours | hours\_servArray | Integer |
| Server | server\_advArray | Text |
| Memory | memory\_advArray | Text |
| Server Counter [Array Control Attribute] | serverCounter | Integer |
| RAM | ram\_advArray | Text |
| Primary HD | primaryHD\_advArray | Text |
| Power Supply | powerSupply\_advArray | Text |
| Enclosures Required | enclosuresRequired | Integer |
| Enclosure | enclosure | Text |
| Cloud Service | cloudServiceType | Text |
| Blade Enclosure Image [HTML] | bladeEnclosureImage | HTML |
| Remaining Blade Slots | remainingBladeSlots | Integer |
| Packaged Solutions | packagedSolutions | Text |
| Total Equipment FOR RULES | totalEquipmentFORRULES | Integer |
| SLA Server | sLAServer\_array | Text |
| SLA Processor | sLAProcessor\_array | Text |
| SLA Counter [Array Control Attrb] | sLACounter | Integer |
| SLA RAM | sLARAM\_array | Text |
| SLA HardDrive | sLAHardDrive\_array | Text |
| SLA OS | sLAOS\_array | Text |
| SLA Price | sLAPrice\_array | Currency |
| Campaign | campaign | HTML |
| Operating System | operatingSystem\_selector | Text |
| System | serverChassis\_selector | Text |
| Processor | processors\_selector | Text |
| Advanced Search Directions | advancedSearchDirections | HTML |
| Form Factor | formFactor\_selector | Text |
| I want to be able to | iWantToBeAbleTo\_selector | Text |
| Do multiple employees need to access storage appliances? | doYouHaveAnyStorageAppliances | Text |
| How much data do you currently manage and store? | howMuchDataDoYouCurrentlyManageAndStore | Text |
| How many employees does your company have? | howManyEmployeesDoYouHaveNow | Text |
| Override Low Quantity? | overrideLowQuantity | Boolean |
| Technical Override | technicalOverride | Text |
| Operating System | operatingSystemSelection | Text |
| Services Requested | servicesRequested\_mobile | Text |
| Operating System Image | operatingSystemImage\_html | HTML |
| Enclosure | enclosure\_mobile | Text |
| Do you plan to use Skype or Video Conferencing? | videoConferencing\_guidedSelling | Text |
| Do you plan to watch videos? | videos\_guidedSelling | Text |
| What are your connectivity needs? | connectivity\_guidedSelling | Text |
| What color would you like? | color\_guidedSelling | Text |
| Screen | screen | Text |
| CPU | cpu | Text |
| Memory | memory | Text |
| Storage | storage | Text |
| Connectivity | connectivity | Text |
| Camera | camera | Text |
| Battery | battery | Text |
| Tablet Accessories | tabletAccessories | Boolean |
| Standard Cases | standardCases | Text |
| Screen Protectors | screenProtectors | Text |
| Headphones | headphones | Text |
| Do you plan to store your video library on your device? | videoStorage\_guidedSelling | Text |
| What level of support do you need? | supportLevel\_guided | Text |
| Tablet Details | setLandingPageHTML | HTML |
| Part Upload ID | partUploadID | Text |
| Number of Rows [ARRAY CTRL] | numberOfRows | Integer |
| Part Number | partNumber | Text |
| Description | description | Text |
| Quantity | quantity | Text |
| Price | price | Text |
| Part Upload HTML | partUploadHTML | HTML |
| Service Promotion | servicePromotion | HTML |
| Selected Index | selectedIndex\_singleSelectArray | Integer |
| Single Select Array Control Attribute | singleSelectArrayControlAttribute | Integer |
| Item | item\_singleSelectArray | Text |
| Select | select\_singleSelectArray | Boolean |
| Additional Part Number | additionalPartNumber | Text |
| Additional Quantity | additionalQuantity | Integer |
| Body Color | bodyColor | Text |
| Engine Capacity | engineCapacity | Text |
| Test1\_IntTxt | test1 | Integer |
| Test2\_TxtSs | test2 | Text |
| Test3\_txtTxt | test3 | Text |
| Test1 | test1 | Integer |
| Test2 | test2 | Text |
| Test3 | test3 | Text |
| Test4 | test4 | Text |
| Array Set 1 Control Attribute | arraySet1ControlAttribute | Integer |
| Part Number | partNumber\_array1 | Text |
| Description | description\_array1 | Text |
| Page Size | pageNumber | Integer |
| Array Set 2 Control Attribute | arraySet2ControlAttribute | Integer |
| Select | select\_arrayset2 | Boolean |
| Part Number | partNumber\_arrayset2 | Text |
| Description | description\_array2 | Text |
| Qty | qty\_array2 | Integer |
| Test SSP | testSSP | Text |
| Previous | previous\_button | HTML |
| Next | next\_button | HTML |
| Previous Flag | previousFlag | Text |
| Next Flag | nextFlag | Text |
| Last Index of array 1 | lastIndexOfArray1 | Integer |
| Country | country | Text |
| Phone number | phoneNumber | Text |
| County | county | Text |
| State | state | Text |
| Zip Code | zipCode | Text |
| Address | address | Text |
| Question | question | Text |
| Array control | arrayControl | Integer |
| Answer List | answerList | Text |
| Base Model | BaseModel\_pcr | Text |
| Ultimate Destination Country | UltimateDestinationCountry\_pcr | Boolean |
| Region | Region\_pcr | Float |
| Part Number | partNumber | Text |
| Part Info String | partInfoString | Text |
| Part Seacrh Flag | partSeacrhFlag | Text |
| Ship to Country | SsshipToCountry | Text |
| Battery Type | SsbatteryType | Text |
| Battery Tech | SsbatteryTech | Text |
| Battery Capacity | SsbatteryCapacity | Text |
| Transactional Info | transactionalInfo | HTML |
| Test | test | Text |
| Office Type | officeType | Text |
| Number of chairs | numberOfChairs | Integer |
| Delivery | delivery | Boolean |
| Home Chair Test | homeChairTest | Text |
| school Chair test | schoolChairTest | Text |
| Base Model | BaseModel\_competencyBuilding | Text |
| Ultimate Destination Country | UltimateDestinationCountry\_competencyBuilding | Text |
| Business Geography | BusinessGeography\_competencyBuilding | Text |
| Number of Spares | NumberofSpares\_competencyBuilding | Integer |
| Spares Include | SparesInclude\_competencyBuilding | Boolean |
| Helplinks | Helplinks\_competencyBuilding | HTML |
| Region | Region\_competencyBuilding | Text |
| Order Type | OrderType\_competencyBuilding | Text |
| Co-termination Date | CoterminationDate\_competencyBuilding | Date |
| Price | Price\_competencyBuilding | Currency |
| Mounting Change | MountingChange\_competencyBuilding | Text |
| Assistance Type | AssistanceType\_competencyBuilding | Text |
| Country | country\_ary\_tbr | Text |
| State | state\_ary\_tbr | Text |
| City | city\_ary\_tbr | Text |
| Array Control Attribute | arrayControlAttribute | Integer |
| Type | type\_ary\_tbs | Text |
| Test1 | test1 | Text |
| Variable Pay | variablePay | Text |
| variable pay Float | variablePayFloat | Float |
| Integertextattributenonarray | integertextattributenonarray | Integer |
| HTMLattributenonarray | hTMLattributenonarray | HTML |
| Dateattributenonarraytextfield | dateattributenonarraytextfield | Date |
| Currencyattributenonarray | currencyattributenonarray | Currency |
| boolenattributenonarraytexttype | boolenattributenonarraytexttype | Boolean |
| floatattributenonattributessm | floatattributenonattributessm | Float |
| Integerssmnonarray | integerssmnonarray | Integer |
| TextMMAnonarray | textMMAnonarray | Text |
| Floatmsmnonarray | floatmsmnonarray | Float |
| IntegerMSMnonarray | integerMSMnonarray | Integer |
| textarraytype | textarraytype | Text |
| booleanarraytype | booleanarraytype | Boolean |
| currencytextattributearraytype | currencytextattributearraytype | Currency |
| Integertexttypearrayattribute | integertexttypearrayattribute | Integer |
| Floattextarraytype | floattextarraytype | Float |
| IntegerSingleSelectMenuArray | integerSingleSelectMenuArray | Integer |
| IntegerSSMarrayattribute | integerSSMarrayattribute | Integer |
| Continent | Continent\_competencyBuilding | Text |
| Demo Attr 1 | demoAttr1 | Text |
| Demo Attr 2 | demoAttr2 | Text |
| model | model\_demo | Text |
| Transaction ID | transactionID | Text |
| result | result | HTML |
| Transaction List | transactionList | HTML |
| jhghj | jhghj | Integer |
| Evaluation Period | evaluationPeriod | Text |
| test | test\_t | Text |
| Name | name\_t | Text |
| test234 | test234 | Text |
| Processor | processor | Text |
| Operating System | operatingSystem | Text |
| Sub Total | subTotal | Integer |
| Total | total | Integer |
| Bread Type | breadType | Text |
| Toppings | toppings | Text |
| Door | trgBanMaColor | Text |
| Quantity | trgBanMaquantity | Float |
| Area | trgBanMaArea | Text |
| Type | trgBanMaType | Boolean |
| Model | trgBanMMmodel | HTML |
| PhoneNumber | phoneNumber | Text |
| Area | trgBanMaArea1 | Text |
| HTML Body Style | trgBanMaHTML | HTML |
| Transaction Type | transactionType | Text |
| Transaction Date | transactionDate | Text |
| Account Type | accountType | Text |
| Windows | trgBanMWindows | Text |
| Date | trgBanMadate | Date |
| Cost | trgBanMacost | Currency |
| Multi\_Text | ban\_trg\_ss\_multi\_Text | Text |
| Transaction Amount | transactionAmount | Currency |
| Number | number | Integer |
| Laptop1 | ten\_ban\_ss\_laptop1 | HTML |
| Loan Type | loanType | Text |
| Interest Type | interestType | Text |
| Test Constraint | testConstraint | Text |
| HTML Series | hTMLSeries | HTML |
| Series | series | Text |
| Customer Name | customerName | Text |
| Number of months for repayment | numberOfMonthsForRepayment | Integer |
| Rate of Interest | rateOfInterest | Float |
| Engine And Transmission | engineAndTransmission | Text |
| Insurance Applicable | insuranceApplicable | Boolean |
| Colour | colour | Text |
| Select Memory | trgbanss\_memory | Text |
| Trim | bangtrtrim | Text |
| Options | options | Text |
| Select Processor | trnban\_ss\_processor | Text |
| Select Microsoft Operating System | trnban\_ss\_operationSystem | Text |
| Select Microsoft Office Software | trnban\_ss\_software | Text |
| Select Hard Disk | trnban\_ss\_hardDisk | Text |
| HTML Engine | hTMLEngine | HTML |
| HTML Color | hTMLColor | HTML |
| HTML Trim | hTMLTrim | HTML |
| HTML Options | hTMLOptions | HTML |
| HTML Full Details | hTMLFullDetails | HTML |
| Processor Image | trgbanss\_processorImage | HTML |
| OS Image | tegbanss\_oSImage | HTML |
| Memory Image | trgbanss\_memoryImage | HTML |
| Storage Image | trgbanss\_storageImage | HTML |
| Office Image | trgbanss\_OfficeImage | HTML |
| Alternate Phone Number | alternatePhoneNumber | Text |
| Customer Name | customer\_Name | Text |
| Salary | salary | Integer |
| Loan Amount Requested | loanAmountRequested | Integer |
| Life Insurance | lifeInsurance | Text |
| Memory | memory | Text |
| Software | software | Text |
| Discounts | discounts | Integer |
| Savings | savings | Integer |
| Headphones | trgbanss\_headphones | Text |
| Door Choosen | textDoor | Text |
| Colour Choosen | colourChoosed | Text |
| Trim Choosen | setTrim | Text |
| Series Choosen | seriesChoosen | Text |
| Engine Choosen | engineChoosen | Text |
| Options Choosen | optionsChoosen | Text |
| Options Choosen | optionsChoosen1 | Text |
| Options Choosen | optionsChoosen2 | Text |
| ArrayControlAttribute | bangtrarrayControlAttribute | Integer |
| OrderDate | orderDate | Date |
| Order Date | trgbanss\_orderDate | Date |
| Order Date | orderDate | Date |
| DateChoosen | dateChoosen | Text |
| Serial No | serialNo | Integer |
| Product | product | Text |
| Qty | qty | Integer |
| Array Size | arraySize | Integer |
| EmployeeName | employeeName | Text |
| Salary | Empsalary | Integer |
| Contact Number | EmpcontactNumber | Text |
| Date of Birth | EmpdateOfBirth | Text |
| Test Cat | testCat | Text |
| Type of Cat | typeOfCat | Text |
| Bearing | bearing | Text |
| Hub | hub | Text |
| Turbo | turbo | Text |
| Series | series | Text |
| Rotors | rotors | Text |
| WheelHub | wheelHub | HTML |
| Turbo | turboHTML | HTML |
| front brake rotor | frontBrakeRotor | HTML |
| Optical Drive MS | opticalDriveMS | Text |
| APS UPS Frame Type | selectedFrameType\_aPSModel | Text |
| APS UPS System KVA Rating | selectedSystemKVARating\_aPSModel | Text |
| Number of Charger Modules | numberofChargerModules\_aPSModel | Integer |
| System Type | selectedSystemType\_aPSModel | Text |
| Battery Type & No. of Strings | batteryType\_NumofStrings\_aPSModel | Integer |
| Distribution Slot 1 | selectedDistributionSlot1\_aPSModel | Text |
| Output Voltage & Frequency | selectedOutputVoltageFrequency\_aPSModel | Text |
| Distribution Slot 2 | selectedDistributionSlot2\_aPSModel | Text |
| APS SFA Product Type | selectedAPSSFAProductType\_aPSModel | Text |
| Communication | selectedCommunication\_aPSModel | Text |
| APS Label and Packaging | selectedAPSLabelAndPackaging\_aPSModel | Text |
| Json String Attribute | jsonStringAttribute | Text |
| Model Name | modelName | Text |
| Model | modelUps | Text |
| Form Factor | formFactor | Text |
| Power | power | Text |
| Default Options | defaultOptions | Text |
| Additional Accessories | additionalAccessories | Text |
| Comments | comments | Text |
| Shape | shape | Text |
| Size | size | Text |
| Cabinet Made | cabinetMade | Text |
| Maintenance Bypass Cabinet Model | maintenanceBypassCabinetModel | Text |
| Maintenance Bypass Cabinet Shape | maintenanceBypassCabinetShape | Text |
| Carrier | carrier | Text |
| Special Services | specialServices | Text |
| Reference 1 | reference1 | Text |
| Reference 2 | reference2 | Text |
| Reference 3 | reference3 | Text |
| CPU | cPU | Text |
| Spike | spike | Text |
| On Call Support | onCallSupport | Text |
| Freight Load/Unload | freightLoadUnload | Text |
| Most Selling | mostSelling | Text |
| Most Viewed | mostViewed | Text |
| Preferable Power Ranges | preferablePowerRanges | Text |
| Voltage Conditioning | voltageConditioning | Text |
| Processsor | processsor | Text |
| Operating System | operatingSystem | Text |
| Supported Browsers | supportedBrowsers | Text |
| Suggestion 1 | suggestion1 | Text |
| Suggestion 2 | suggestion2 | Text |
| Suggestion 3 | suggestion3 | Text |
| Carrier Service | carrierService | Text |
| Call Support | callSupport | Text |
| 551622G1 Flag | a551622G1Flag | Boolean |
| 552241G1 Flag | a552241G1Flag | Boolean |
| 551628G1 Flag | a551628G1Flag | Boolean |
| 551629G1 Flag | a551629G1Flag | Boolean |
| 552683G1 Flag | a552683G1Flag | Boolean |
| 552683G2 Flag | a552683G2Flag | Boolean |
| 551622G1 | a551622G1 | Text |
| 552241G1 | a552241G1 | Text |
| 551628G1 | a551628G1 | Text |
| 551629G1 | a551629G1 | Text |
| 552683G1 | a552683G1 | Text |
| 552683G2 | a552683G2 | Text |
| Test | test | Text |
| test1 | test1 | Text |
| test2 | test2 | Text |
| test3 | test3 | Text |
| test4 | test4 | Text |
| Contract Array Attribute | contractArrayAttribute | Integer |
| Contract Id | contractId | Text |
| Contract Name | contractName | Text |
| Select | select | Boolean |
| Part Number String | partNumberString | Text |
| QtyItem1 | qtyItem1 | Text |
| QtyItem2 | qtyItem2 | Text |
| qty | qty | Integer |
| WAF | wAFOffering | HTML |
| WAF Offering | wAFOfferingSelect | Text |
| WAF Subscription Additive Feature | wAFSubscriptionAdditiveFeature | HTML |
| WAF Additional Bandwidth | wAFAdditionalBandwidth | Text |
| WAF Additional FQDN | wAFAdditionalFQDN | Text |
| DDoS Offering | dDoSOffering | HTML |
| DDos Offering | dDosOfferingSelect | Text |
| DDoS Additional Bandwidth | dDoSAdditionalBandwidth | Text |
| Data Center | dataCenter | Text |
| Virtual IP | virtualIP | Text |
| Router Monitorning | routerMonitorning | Text |
| Deployment | deployment | Text |
| DDoS Subscription Additive Feature | dDoSSubscriptionAdditiveFeature | HTML |
| Configure WAF | configureWAF | Boolean |
| Configure DDoS | configureDDoS | Boolean |
| Configure Shared Services | configureSharedServices | Boolean |
| Select Model | vedanta\_SelectModel | Text |
| Quantity (in Metric Ton) | vedanta\_QuantityInMetricTon | Integer |
| Model | modelname | Text |
| Quantity (in Metric Ton) | quantityInMetricTon | Integer |

| Rule Name | Variable Name | Data Type |
| --- | --- | --- |
| Partner Info | partnerInfo | Recommendations |
| Set Enclosures Required [Easy Advanced] | setEnclosuresRequired | Recommendations |
| Force Set Chassis [Simple] | forceSetChassis | Recommendations |
| Adv Config Controller [Easy Advanced] | populateAdvConfigController | Recommendations |
| Force Config Servers [Medium - Advanced] | setArrayValues | Recommendations |
| Set Config Details [Medium - Advanced] | setConfigDetails | Recommendations |
| Populate Blade Image [Easy Advanced] | populateBladeEnclosureImage | Recommendations |
| OS Volume Pricing [Advanced HTML] | oSVolumePricing | Recommendations |
| Software Array Values [Easy Advanced] | softwarePackageArrayValues | Recommendations |
| Service Array [Hard Advanced] | serviceArray | Recommendations |
| Promotion [HTML] | promotionCampaign | Recommendations |
| Set Remaining Blades [Easy Advanced] | setRemainingBlades | Recommendations |
| Packaged Solutions [Table] | packagedSolutions | Recommendations |
| Rec Cloud Backup [Simple] | recCloudBackup | Recommendations |
| Service Details Price [Advanced] | serviceDetailsPrice | Recommendations |
| Set Total Equipment [Easy Advanced] | setTotalEquipment | Recommendations |
| SLA Array Size [1 Line Script] | sLAArraySize | Recommendations |
| SLA Array Fields [Advanced] | sLAArrayFields | Recommendations |
| Power Supply - Recommendation [Simple] | powerSupplyRecommendation | Recommendations |
| Campaign [HTML] | campaign | Recommendations |
| Entry Level Server | entryLevelServer | Recommendations |
| Mid Range Server | midRangeServer | Recommendations |
| High End Server | highEndServer | Recommendations |
| Green Starter Package | greenStarterPackage | Recommendations |
| Sentinel Standard Package | sentinelStandardPackage | Recommendations |
| Ultra Power Solution Package | ultraPowerSolutionPackage | Recommendations |
| Remove Override [Simple] | removeOverrideSimple | Recommendations |
| Set OS Image | setOSImage | Recommendations |
| Service Array [Hard Advanced] [Mobile] | serviceArrayMobile | Recommendations |
| CPU Rec Based on Screen Size | cpuScreenSize | Recommendations |
| Connectivity Needs | connectivityNeeds | Recommendations |
| Video Conferencing | videoConferencing | Recommendations |
| Video Playback | videoPlayback | Recommendations |
| Video Storage | videoStorage | Recommendations |
| Set Landing Page HTML | setLandingPageHTMl | Recommendations |
| Set Mobile Landing Page HTML | setMobileLandingPageHTML | Recommendations |
| Set Part Upload ID | setPartUploadID | Recommendations |
| Set Parts Array | setPartsArray | Recommendations |
| Set Promotion Image | setPromotionImage | Recommendations |
| Set Single Select Array | setSingleSelectArray | Recommendations |
| Set Array Set 1 | setArraySet1 | Recommendations |
| Set Array Set 2 | setArraySet2 | Recommendations |
| populate quetsion | populateQuetsion | Recommendations |
| Populate Parts Info String | populatePartsInfoString | Recommendations |
| Populate Geography Array | populateGeographyArray | Recommendations |
| model level | modelLevel | Recommendations |
| Demo Rule | demoRule | Recommendations |
| Display Result | displayResult | Recommendations |
| Display Transaction List | displayTransactionList | Recommendations |
| Calculate Subtotal | calculateSubtotal | Recommendations |
| INsurance Recommendation | iNsuranceRecommendation | Recommendations |
| HighendModelRecommendation | trgbanss\_highendModelRecommendation | Recommendations |
| Alternate Phone Number | bangtralternatePhoneNumber | Recommendations |
| Door | recommendation | Recommendations |
| Options | options | Recommendations |
| Colour | colour | Recommendations |
| Loan Recommendation | loanRecommendation1 | Recommendations |
| Life insurance recommendation | lifeInsuranceRecommendation | Recommendations |
| HidingHeadPhone | trgbanss\_hidingHeadPhone | Recommendations |
| SetColourValue | setColourValue | Recommendations |
| setDoorValue | setDoorValue | Recommendations |
| SetTrimValue | setTrimValue | Recommendations |
| setSeriesValue | setSeriesValue | Recommendations |
| setEngineValue | setEngineValue | Recommendations |
| setOptionsValue | setOptionsValue | Recommendations |
| setOptionsValue | setOptionsValue1 | Recommendations |
| Test | test | Recommendations |
| DateChoosentoStrind | dateChoosentoStrind | Recommendations |
| Populate Discount | populateDiscount | Recommendations |
| Calculate Savings | calculateSavings | Recommendations |
| Calculate Total | calculateTotal | Recommendations |
| Define Array Size | defineArraySize | Recommendations |
| Set Json Attribute | setJsonAttribute | Recommendations |
| Rec Rule to Set Helper Attributes | recRuleToSetHelperAttributes | Recommendations |
| Recommendation rule to Set attributes based on model is SMT100C | recommendationRuleToSetAttributesBasedOnModelIsSMT100C | Recommendations |
| Contract Recc Rule | contractReccRule | Recommendations |
| SetAttr | setAttr | Recommendations |

| Util Name | Variable Name | Script Text |
| --- | --- | --- |
| Print Doc Designer | CN\_PrintDocDesigner | // 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; |
| Compare Docx | CN\_CompareDocx | 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; |
| Print Email Designer BML | CN\_PrintEmailDesigner | // 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; |
| Update File Attachment | CN\_UpdateFileAttachment | 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; } |
| Quote Analysis PL | quoteAnalysisPL | 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; |
| Merge DOCX BML | CN\_MergeDocx | response = dict("string"); // GENERATE MERGE REST CALL url = siteUrl+"/rest/v2/docxMerge"; jsonObj = json(); jsonput(jsonObj, "processVarname", processVarName); jsonput(jsonObj, "transactionId", transactionId); jsonput(jsonObj, "sourceFileAttachId", sourceFileId); jsonput(jsonObj, "targetFileAttachId", targetFileId); jsonput(jsonObj, "approvedChanges", jsonarray(compareJsonString)); 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" ){ er = get(xlsxResponse, "Error-Message"); if(len(er) <= 5){ er = get(xlsxResponse, "Message-Body"); } put(response, "error", er ); return response; }else{ mb = get(xlsxResponse, "Message-Body"); jmb = json(mb); errors = jsonget(jmb , "errors" ); if(len(errors) > 5){ put(response, "errArray", errors); print("MERGE COMPLETE WITH ERRORS"); return response; } document = jsonget(jmb , "base64Doc" ); put(response, "base64doc", document); print("MERGE COMPLETE"); return response; } |
| Get File Id From Response | CN\_GetFileIdFromResponse | mb = get(response, "Message-Body"); jmb = json(mb); ids = jsonget(jmb, "fileId"); if( isnumber(ids) ){ return atoi(ids); } return -1; |
| Get File Attachment Data | CN\_GetFileAttachmentData | 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; } |
| abotester\_doUpdateAsset | abotester\_doUpdateAsset | //Function Name: abotester\_doUpdateAsset //input:txn\_json(json), lines\_json\_array(jsonarray) //output: response(json) //Description: // Entry level utility BML for header level 'Update Asset' test action. // For the given txn and list of lines in that txn as passed in the input, it will aggregate the lines for that // same product instance in order of request date to calculate the final project state and compare it with the // asset state to generate CRUD operation on the asset for all lines in an atomic RESTful call. //Logic: // 1. sort the line array by assetkey and request date. // 2. group the lines with the same asset key into the following structure. // { instace\_id:, final\_action:, final\_date, lines: { line\_seq, action, txn\_date}. // 3. For each group, // load base asset configuration and convert that to BOM struture // load BOM for each line in the group and place them as open orderline array // invoke abo\_apply with asset and open orderline array to caculated final projected BOM // use abo\_delta to caculate the delta state between original base asset BOM and projected BOM // convert the delta BOM to sync asset json format for local asset repository sync. // 4. put all the invidual sync asset json into an array and invoke asset servcie as single atomic call. //Step 1: Setup context abocontext = util.\_ORCL\_ABO.abo\_getContext(); bDiagnosisOff=jsonget(abocontext, "AboDiagnosticDisabled", "boolean", true); callContext="doUpdateAsset"; if(not bDiagnosisOff){//log the input dummy=util.\_ORCL\_ABO.abo\_addDiagnosticInfo(callContext+"\_input\_txnjson", txn\_json,"addToSummary"); lineJson=json(); jsonput(lineJson, "lineArray",lines\_json\_array); dummy=util.\_ORCL\_ABO.abo\_addDiagnosticInfo(callContext+"\_input\_linejson", lineJson,"addToSummary"); } localAssetSvc = jsonpathgetsingle(abocontext, "$.siteInfo.localAssetSvc"); //Step 2: Declare constants //Constants used for line data structure DOCUMENT\_NUMBER = "documentNumber"; CUSTOMER\_ID = "customer\_id"; TRANSACTION\_ID = "transactionId"; CURRENCY\_CODE = "currency\_code"; REQUEST\_DATE = "requestDate"; ACTION\_CODE = "actionCode"; ITEM\_INSTANCE\_ID = "itemInstanceId"; FINALACTION = "finalAction"; FINALDATE = "finalDate"; TERMINATE = "TERMINATE"; ORIGINALACTION = "OriginalAction"; ORIGINALREQUESTDATE = "OriginalRequestDate"; ADD = "ADD"; UPDATE = "UPDATE"; SUSPEND = "SUSPEND"; RESUME = "RESUME"; RENEW = "RENEW"; SUSPENDDATE = "suspendDate"; RESUMEDATE = "resumeDate"; RENEWDATE = "renewDate"; //Constants used for getBom query TRANSACTIONDATE = "requestDate\_l"; ACTION = "oRCL\_ABO\_ActionCode\_l"; ITEM\_INSTANCE\_NAME = "itemInstanceName\_l"; //Constants for internal logic LINES = "Lines"; //List of fields need to be retrieved in getBom call extraAssetFields = jsonget(abocontext, "extraLineFieldForAsset", "json", json()); lineFields = jsonKeys(extraAssetFields); sizeofExtraAssetFieldLst = sizeofarray(lineFields); lineFields[sizeofExtraAssetFieldLst] = ACTION; lineFields[sizeofExtraAssetFieldLst+1] = TRANSACTIONDATE; lineFields[sizeofExtraAssetFieldLst+2] = ITEM\_INSTANCE\_NAME; // Initialize the arrays jsonArrforSync = jsonarray(); syncAssetJson = json(); //Step 3: Sort the lines by assetKey and then date in ascending // If line.txnDate is empty, updateAsset action script will populate it with today's date during lineJson creation jsoncrit = jsonarray(); jsoncrit = jsonarray(); crit1= json("{\"field\":\"itemInstanceId\",\"order\":\"desc\"}");//type default to string crit2= json("{\"field\":\"requestDate\",\"order\":\"asc\",\"dateFormat\": \"yyyy-MM-dd\"}"); jsonarrayappend(jsoncrit,crit1); jsonarrayappend(jsoncrit,crit2); linesJsonArr = util.\_ORCL\_ABO.abo\_jsonSort(lines\_json\_array, jsoncrit); if(not bDiagnosisOff){//log the sorted array logJson=json(); jsonput(logJson, "sortedLines",linesJsonArr); jsonput(logJson, "crit",jsoncrit); dummy=util.\_ORCL\_ABO.abo\_addDiagnosticInfo(callContext+"\_sortingResult", logJson,""); } //Step 4: Since it has already been sorted, we just need to scan it in one pass sizeoflineArr = jsonarraysize(linesjsonarr); dummyTraverseArray = string[sizeoflineArr]; //retrieve transaction data transactionId = jsonget(txn\_json, TRANSACTION\_ID, "string", ""); bsId = atoi(transactionId ); jsonArrforSync = jsonarray(); curAssetkey=""; index = 0; finalActionforCurGrp = ""; grpJsonforProcessing = jsonarray(); lineGrpJsonArr = jsonarray(); curGrpJson = json(); // since lines\_json\_array is already sorted based on assetKey, requestDate, we just need to scan them once for oneItem in dummyTraverseArray{ oneLine = jsonarrayget(linesJsonArr, index, "json"); index = index + 1; nextAssetKey= jsonget(oneLine, ITEM\_INSTANCE\_ID, "string", ""); reqDate = jsonget(oneLine, REQUEST\_DATE, "string", ""); actionCode = jsonget(oneLine, ACTION\_CODE, "string", ""); if(curAssetKey <> nextAssetKey){ //now it is a new group, notice we initialize curAssetKey to "", so first line will trigger this logic as well curGrpJson = json(); lineGrpJsonArr = jsonArray(); jsonput(curGrpJson, ITEM\_INSTANCE\_ID, nextAssetKey); curGrpJson = jsonarrayappend(grpJsonforProcessing, curGrpJson); lineGrpJsonArr = jsonput(curGrpJson, LINES, lineGrpJsonArr); curAssetKey = nextAssetKey; finalActionforCurGrp = ""; jsonput(curGrpJson, ORIGINALACTION, actionCode); jsonput(curGrpJson, ORIGINALREQUESTDATE, reqDate); } //add line to current group jsonarrayappend(lineGrpJsonArr, oneLine); if(finalActionforCurGrp==TERMINATE){ // all lines after terminate will be igored. // PLACEHOLDER, you could choose to throw error in this case as well. // You could run into this situation if user does a Follow-On Order on d2, and then terminate on d1, // Assuming "d1" and "d2" are referring to dates // or additional logic can be added in abo\_prepareInstance to prevent this to happen continue; } if( actionCode == TERMINATE ){ //keep track of first terminate line finalActionforCurGrp = TERMINATE; jsonput(curGrpJson, FINALACTION , actionCode); jsonput(curGrpJson, FINALDATE, reqDate); } if (actionCode == SUSPEND) { jsonput(curGrpJson, SUSPENDDATE, reqDate); } if (actionCode == RESUME) { jsonput(curGrpJson, RESUMEDATE, reqDate); } if (actionCode == RENEW) { jsonput(curGrpJson, RENEWDATE, reqDate); } }// end of loop through lines to group them together by assetKey. if(not bDiagnosisOff){//log the output of group logJson=json(); jsonput(logJson, "arrGroups", grpJsonforProcessing); dummy=util.\_ORCL\_ABO.abo\_addDiagnosticInfo(callContext+"\_groupedLines", logJson,"addToSummary"); } //Step 5: Loop through each group, Call apply, delta and add the resultant json for sync in jsonArrforSync //calculate today's date in db format. If txnDate is emtpy in getBom, we will replace it with today //so that down stream procesing will not need to worry about this. tranDate = getDate(false); today = datetostr(tranDate, "yyyy-MM-dd HH:mm:ss"); sizeofgrpForProcessing = jsonarraysize(grpJsonforProcessing); dummyArr = string[sizeofgrpForProcessing]; index = 0; curGrpJson = json(); for oneItem in dummyArr{//loop thru groups oneJson = jsonarrayget(grpJsonforProcessing, index, "json"); index = index+1; //Step 5.1: Load base asset and convert to BOM structure assetId = jsonget(oneJson, ITEM\_INSTANCE\_ID, "string", ""); baseBom = util.\_ORCL\_ABO.abo\_loadAsset(assetId); //Step 5.2: Loop through order lines, and retrieve corresponding BOM. lineGrpJsonArray = jsonget(oneJson, LINES, "jsonarray"); numberofLines = jsonarraysize(lineGrpJsonArray); dummytraverseArr = string[numberofLines]; arrBomtoApply = jsonarray(); //prepare open quote BOM array docNum = ""; lineGrpJsonArrIndex = 0; actionCodeofOneQuoteBom = ""; for item in dummytraverseArr{ //loop through lines within a group oneLineJson = jsonarrayget(lineGrpJsonArray, lineGrpJsonArrIndex, "json"); lineGrpJsonArrIndex = lineGrpJsonArrIndex +1; //check if it is external process or internal - This value is set to true in updateExternal function ONLY isExternal=jsonget(oneLineJson, "isExternal", "string","false"); // intialize json oneQuoteBom = json(); // if it is an external process, retrieve the BOM from the config integration table if(isExternal == "true"){ // Get the config id from line config\_id = jsonget(oneLineJson, "\_document\_number", "string",""); // get flatten BOM oneQuoteBom = getconfigbom(atoi(config\_id),true); oneQuoteBom = jsonarrayappend(arrBomtoApply, oneQuoteBom); } else { // This is CPQ Transaction - hence retrieve data using the transactionId transactionId = jsonget(txn\_json, TRANSACTION\_ID, "string", ""); docNum = jsonget(oneLineJson, DOCUMENT\_NUMBER, "string"); lineNum = atoi(docNum); bsId = atoi(transactionId ); oneQuoteBom = getbom(bsId, lineNum, lineFields, false, //validate true, //flatten true); //isSalesBom oneQuoteBom = jsonarrayappend(arrBomtoApply, oneQuoteBom); } fieldJsonofOneQuoteBom = jsonget(oneQuoteBom, "fields", "json", json()); //adjust the action code for apply as current apply logic will throw error when trying to apply terminate lines //here we just leverage the apply logic to calculate the projected BOM state. This removes the child of terminate lines which is marked for delete. //Terminate line will be the last line to apply(as enforced in Step 4 grouping logic). actionCodeofOneQuoteBom = jsonget(fieldJsonofOneQuoteBom, ACTION, "string", ""); if(actionCodeofOneQuoteBom==TERMINATE){ jsonput(fieldJsonofOneQuoteBom, ACTION, UPDATE); } //if line did not specify transactionDate, put in the default date of today requestDateinOneQuoteBom = jsonget(fieldJsonofOneQuoteBom, TRANSACTIONDATE, "string", ""); if(requestDateinOneQuoteBom == ""){ jsonput(fieldJsonofOneQuoteBom, TRANSACTIONDATE, today); } reqDate = jsonget(oneLineJson, REQUEST\_DATE, "string",today); // for terminate, suspend resume case set the group summary if (actionCodeofOneQuoteBom == ADD) { jsonput(oneJson, ORIGINALACTION, actionCodeofOneQuoteBom); jsonput(oneJson, ORIGINALREQUESTDATE,today); } if (actionCodeofOneQuoteBom == TERMINATE) { jsonput(oneJson, FINALACTION,actionCodeofOneQuoteBom); jsonput(oneJson, FINALDATE,reqDate); } if (actionCodeofOneQuoteBom == SUSPEND) { jsonput(oneJson, SUSPENDDATE, reqDate); } if (actionCodeofOneQuoteBom == RESUME) { jsonput(oneJson, RESUMEDATE, reqDate); } if (actionCodeofOneQuoteBom == RENEW) { jsonput(oneJson, RENEWDATE, reqDate); } } //end of Step 5.2 loop through lines within a group to generate open order BOM array //Step 5.3: Apply open order on base asset to calculate the projected final asset state baseBomCopy = jsoncopy(baseBom); pacBom = util.\_ORCL\_ABO.abo\_apply(baseBom, arrBomtoApply); //Step 5.4: Compare projected state with base asset to get delta deltaBom = util.\_ORCL\_ABO.abo\_processDeltaBom(baseBomCopy, pacBom); jsonput(deltaBom, "groupSummary", oneJson); //Step 5.5: Convert delta BOM to sync asset json format syncAssetJson = util.\_ORCL\_ABO.abo\_convertDeltaBomtoAsset(deltaBom, txn\_json); jsonarrayappend(jsonArrforSync, syncAssetJson); }//end of Step 5, loop through groups of assets //Step 6: Invoke local asset repository sync servcie to apply the change on asset repository atomically. // //prepare final payload to invoke local asset repository itemsjson = json(); jsonput(itemsjson, "items", jsonArrforSync); jsonforpost = json(); jsonput(jsonforpost, "\_client\_driven\_action",true ); jsonput(jsonforpost, "documents",itemsjson ); finalUrl = localAssetSvc+"/actions/synchronize"; returnVal = json(); instructionJson=json(); jsonput(instructionJson,"Url",finalUrl); jsonput(instructionJson,"payload",jsonforpost); jsonput(instructionJson,"method","post"); //if (runService <> false) { if (true) {//updated by Piyush on 06/10/2018 for testing responseJson = util.\_ORCL\_ABO.abo\_invokeRESTService("post",finalUrl, jsonforpost); jsonput(instructionJson,"response",responseJson); jsonput(returnVal,"Successfully Invoked Asset Service with following details:",instructionJson); } else { jsonput(returnVal,"Please Invoke Asset Service with following details:",instructionJson); } return returnVal; |
| getBasicAuthCredentials | CN\_GetBasicAuthCredentials | //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; |
| Vertiv Rec Item BOM | recItemBOM | retString = ""; selectedAttrValueDict = selectedAttrDict; BOMItemDict = dict("string"); retStringDict = dict("string"); Quanitity = 0; keysOfSelectAttrValDict = keys(selectedAttrValueDict); results = bmql("Select BOMItemName, ParentBOMItemName, Qty, " + "Attr1, Op1, Val1, " + "Attr2, Op2, Val2 " + "FROM Vertiv\_InternalParts" + " WHERE" + " (Attr1 IN $keysOfSelectAttrValDict OR Attr1 IS NULL)" + " AND (Attr2 IN $keysOfSelectAttrValDict OR Attr2 IS NULL)"); for eachRec in results { flagCheck1 = false; finalFlag = false; Quantity1 = 0; BOMItemName = get(eachRec, "BOMItemName"); ParentBOMItemName = get(eachRec, "ParentBOMItemName"); Quantity = atoi(get(eachRec, "Qty")); Attr1 = get(eachRec, "Attr1"); Op1 = get(eachRec, "Op1"); Val1 = get(eachRec, "Val1"); Attr2 = get(eachRec, "Attr2"); Op2 = get(eachRec, "Op2"); Val2 = get(eachRec, "Val2"); if(Op1 == "=") { if(get(selectedAttrValueDict, Attr1) == Val1) { flagCheck1 = true; } } elif(Op1 == "<>") { if(get(selectedAttrValueDict, Attr1) <> Val1) { flagCheck1 = true; } } if(flagCheck1 == true) { if(isnull(Attr2)) { finalFlag = true; } elif(Op2 == "=") { if(get(selectedAttrValueDict, Attr2) == Val2) { finalFlag = true; } } elif(Op2 == "<>") { if(get(selectedAttrValueDict, Attr2) <> Val2) { finalFlag = true; } } } if(finalFlag == true) { /\*if(not containskey(BOMItemDict, BOMItemName)) { put(BOMItemDict, BOMItemName, string(Quantity)); Quantity1 = Quantity; } else { Quantity1 = Quanitity + atoi(get(BOMItemDict, BOMItemName)); put(BOMItemDict, BOMItemName, string(Quantity1)); }\*/ retString = retString + BOMItemName + "~" + string(Quantity) + "~" + BOMItemName + "~" + "0.0" + "|^|"; } } if(substring(retString,len(retString) - 3, len(retString)) == "|^|") { retString = substring(retString, 0, len(retString) - 3); } return retString; |
| Parse CSV | parseCSV | 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; |