|  |  |  |  |
| --- | --- | --- | --- |
| Module | AR15 – Commission Cash Receipt | | |
| Current Code | oerep/r-commcr.w  oerep/r-commcN.w  oerep/r-commcr.i (run-report logic is located here) | | |
| Current Screen | cid:image002.png@01D18F28.2EF88DC0 | | |
| New Directory Locations & Naming Conventions | 1. programs\aoa 2. programs\appServer 3. programs\includes 4. programs\param 5. programs\tempTable 6. resources\dashboards 7. resources\datFiles 8. resources\excel 9. resources\images 10. resources\publish 11. resources\reports 12. resources\vbScript | 1. stub program location 2. AppServer startups 3. auto generated code 4. parameter screens 5. temp-table includes 6. dashboard.edp files 7. AOA .dat files 8. excel output 9. AOA images 10. publish .bat & .exe 11. report.rpa files 12. vb script .dat files | 1. Procedure blocks begin and end should align 2. Place a comment on all END. statements indicating which procedure block it belongs to and that it is indented to match the starting statement 3. Use properly named variables that are descriptive 4. All Progress key words are capitalized 5. NO-LOCK appears on the same line as the table is accessed, ie, FOR EACH <TableName> NO-LOCK 6. Lock status should appear before the WHERE clause and not at the end of the statement 7. All local variables should begin with its data type “c Character, d Decimal, dt Date, i Interger, l Logical, r RowID, h Handle”, etc. 8. Use camel case naming conventions, ie, a decimal variable for Price Amount = dPriceAmount 9. Internal procedures start with a lower case “p<ProcedureName>” 10. Functions begin with “f<FunctionName>” 11. Temp-Tables begin with “tt<TempTable>” 12. INPUT PARAMETER for a CHARACTER would be ipc<VariableName> 13. OUTPUT PARAMETER would be opc<VariableName> 14. INPUT-OUTPUT PARAMETER would be iopc<VariableName> |
| New Startup | Create aoa/r-commcr.p   1. it should look like the code found below. 2. aoaID is extracted from DataPA Report when you publish a report and is found under Manage Published, do a right mouse click on the Report name … copy URL link… the value of the aoaID is contained in the clipboard… paste it into the code and remove everything but the DataPA ID. 3. setting the Scoped-Define test value to YES allows you to run this program from an editor without having to create a menu entry and running the menu program each time you want to test run.  Simply press F2. 4. aoaCustListForm is only used when using Defined Customer Lists 5. aoaTitle has to match title used in DataPA for naming “.rpa” file 6. Add &SCOPED-DEFINE excelOnly if the report should function as an Excel dump only (see aoa/ rd-fgexp.p as an example)   /\* r-commcr.p \*/  &SCOPED-DEFINE aoaID 9678E2EAD9294562812A613ADF175212  &SCOPED-DEFINE aoaProgramID r-commcr.  &SCOPED-DEFINE aoaTitle Commission Cash Receipt  &SCOPED-DEFINE aoaType Report  &SCOPED-DEFINE aoaParam YES  &SCOPED-DEFINE aoaColumns YES  &SCOPED-DEFINE aoaCustListForm AR15  &SCOPED-DEFINE test NO  {aoa/includes/aoaParam.i} | | |
| New Parameter Screen | Create aoa/param/r-commcr.w    Note:   1. Use aoa/param/template.w as a template for internal procedures, functions and object triggers 2. Load the template.w, perform a SAVE-AS to the new program desired. 3. Delete those objects not needed, add new objects as needed. 4. Remove any code found in procedure pInitialize and pPopulateOptions that prevents the parameter screen from compiling. 5. The include file “{aoa/includes/pCommissionCashReceipt.i}” used in the Business Logic program in aoa/appServer/r-commcr.p (see below) is auto generated doing a RIGHT-MOUSE-CLICK on the Parameter Screen “Cancel” button     Clicking “Yes” will launch Notepad so the Generated Include can be viewed   1. All Parameter Screen fields should begin “sv<FieldName>” 2. Use of All Customers, All Sales Reps, etc. should be named based on the range variable the ALL applies 3. svAllCustNo because the range fields are svStartCustNo and svEndCustNo 4. svAllSalesRep, range fields svStartSalesRep and svEndSalesRep 5. Screen Parameters should be aligned, clean and neat in appearance | | |
| Template Parameter Screen | Use aoa/param/template.p  Screen shot of the parameter screen template which can be used as a starting point for new parameter screens. | | |
| AppServer | Modify definitions section in aoa/appServer/aoaAR.p  /\*------------------------------------------------------------------------  File : aoa/appServer/aoaAR.p  Purpose : AppServer Functions and Procedures  Syntax :  Description : AppServer Functions and Procedures  Author(s) : Ron Stark  Created : 3.23.2016  Notes :  ----------------------------------------------------------------------\*/  /\* This .W file was created with the Progress AppBuilder. \*/  /\*----------------------------------------------------------------------\*/  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\* Commision Cash Receipt.rpa \*/  {aoa/tempTable/ttCommissionCashReceipt.i}  Note: any time changes are done to any aoa/appServer procedure, the asAOA AppServer needs to be stopped and restarted by either using the OpenEdge Explorer or use the following commands in ProEnv (may need to run this as Administrator)   1. asbman -name asAOA -stop 2. asbman -name asAOA -start 3. asbman -name asAOA -query | | |
| Temp-Table  Begins with “tt”  “tt<TempTable>” | Create aoa/tempTable/ttCommissionCashReceipt.i  /\* Commision Cash Receipt.rpa \*/  DEFINE TEMP-TABLE ttCommissionCashReceipt NO-UNDO  {aoa/tempTable/ttFields.i}  FIELD <field> AS <data-type> LABEL “<label>” FORMAT <”format”>  …  Notes:   1. Table Field names are to be in the same order as base DataPA report will be layed out. 2. Field names should not contain any special characters, such as #, $, %, & and/or slashes, etc. 3. Avoid using underscore “\_” or dash “-“. 4. Use “No” for “#”, “Amt” for “$”, “Pct” for “%”, “and” for “&” 5. Any fields needed but should not be selectable by end user should start “xx<Field>” 6. Any data-types of RECID or ROWID are ignored throughout the base code 7. Temp-Table name should be the same as the Report Title with blanks removed 8. If sorting needed use field names such as “xxSort” | | |
| Function  Begins with “f”  “f<Function>” | Add function to aoa/appServer/aoaAR.p  FUNCTION fCommissionCashReceipt RETURNS HANDLE ( {aoa/includes/fInputVars.i} ) :  /\*------------------------------------------------------------------------------  Purpose: Commission Cash Receipt.rpa  Notes:  ------------------------------------------------------------------------------\*/  EMPTY TEMP-TABLE ttCommissionCashReceipt.  /\* subject business logic \*/  RUN aoa/BL/r-commcr.p (OUTPUT TABLE ttCommissionCashReceipt, ipcCompany, ipiBatch, ipcUserID).  RETURN TEMP-TABLE ttCommissionCashReceipt:HANDLE .  END FUNCTION. | | |
| Function “fGetTableHandle” | Modify function fGetTableHandle in aoa/appServer/aoaAR.p  Add the new TEMP-TABLE to this function (alphabetically) with the aoa/BL name and the tt<TEMP-TABLE>:HANDLE.  FUNCTION fGetTableHandle RETURNS HANDLE ( ipcProgramID AS CHARACTER ) :  /\*------------------------------------------------------------------------------  Purpose:  Notes:  ------------------------------------------------------------------------------\*/  CASE ipcProgramID:  WHEN "r-commcr." THEN  RETURN TEMP-TABLE ttCommissionCashReceipt:HANDLE.  END CASE.  END FUNCTION. | | |
| Business Logic | Create aoa/BL/r-commcr.p  Create this program using the code found in “oerep/r-commcr.i”, striping all UI and output logic. All fields that were once output need to be placed into the AOA Temp-Table. Follow formatting and structure of existing aoa/BL programs already created.  /\*------------------------------------------------------------------------  File: r-commcr.p  Description: Business Logic  \*/  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  /\* Commision Cash Receipt.rpa \*/  {aoa/tempTable/ttCommissionCashReceipt.i}  {sys/ref/CustList.i NEW}  /\* Parameters Definitions --- \*/  DEFINE OUTPUT PARAMETER TABLE FOR ttCommissionCashReceipt.  {aoa/includes/pCommissionCashReceipt.i}  /\* local variables \*/  DEFINE VARIABLE ….  /\* subject business logic \*/  DEFINE BUFFER bARInvl FOR ar-invl.  FIND FIRST oe-ctrl NO-LOCK WHERE oe-ctrl.company EQ ipcCompany NO-ERROR.  IF NOT AVAILABLE oe-ctrl THEN RETURN.  FOR EACH ar-inv NO-LOCK  WHERE ar-inv.company EQ ipcCompany  ….  ….  END. /\* each ar-inv \*/  …  …  {aoa/BL/pBuildCustList.i}  Note: if the process calls for a Customer Range, the aoa/BL program will need both “{sys/ref/CustList.i NEW}” in the definitions section and “{aoa/BL/pBuildCustList.i}” at the end of the program.  Note: the include file “{aoa/includes/pCommissionCashReceipt.i}” is auto generated doing a RIGHT-MOUSE-CLICK on the new Parameter “Cancel” button    Clicking “Yes” will launch Notepad so the Generated Included can be viewed. | | |
| Business Logic (BL) Test program .p | For quicker development and testing, the following framework can be utilized. Follow the instructions for the values for the two Scop-Defs. Press F2 to Run and it will auto output the BL results to Excel.  /\* AR15.p \*/  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Instructions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*  1. populate scoped-define tempTable below without "tt" prefix  2. populate scoped-define BL below without ".p" extension  3. use blSample.p for BL program structure, remove CustList code if N/A  4. return aoa/BL/{&BL}.p via email as an attachment for code review  \* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  &SCOPED-DEFINE tempTable CommissionCashReceipt  &SCOPED-DEFINE BL r-commcr  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Definitions \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  /\* add additional fields to this include \*/  {aoa/tempTable/tt{&tempTable}.i}  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Main Block \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  DEFINE VARIABLE hBin AS HANDLE NO-UNDO.  RUN aoa/appServer/aoaBin.p PERSISTENT SET hBin.  SESSION:ADD-SUPER-PROCEDURE (hBin).  RUN p{&tempTable} ("001",0,USERID("nosweat")).  RUN aoa/aoaExcel.p (TEMP-TABLE tt{&tempTable}:HANDLE).  /\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Internal Procedures \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*/  PROCEDURE p{&tempTable}:  /\*------------------------------------------------------------------------------  Purpose:  Parameters: Company, Batch Seq, User ID  Notes:  ------------------------------------------------------------------------------\*/  {aoa/includes/aoaInputDefParams.i}  /\* subject business logic \*/  RUN aoa/BL/{&BL}.p (OUTPUT TABLE tt{&tempTable}, ipcCompany, ipiBatch, ipcUserID).  END PROCEDURE.  Note: For testing purposes, the developer can hard code the parameter values in the aoa/BL program immediately after “{aoa/includes/pCommissionCashReceipt.i}” … be sure to remove them before releasing the BL program. Otherwise, run the parameter screen so that it populates the user-print table, which is where this include gets it’s values. | | |
| New AOA Report | Create resources/reports/Commission Cash Receipt.rpa | | |
| Analytics Engine | Note:   1. Create a Subject 2. Subject Names should be identical to the AOA Report File Name “Commission Cash Receipt.rpa”. 3. <Parameter> = <Required Value Name> = <Required Value Description> 4. ipcCompany = svCompany = Company 5. ipiBatch = svBatch = Batch Seq 6. ipcUserID = svUserID = User ID 7. Select All Columns 8. Check Column Attributes to be sure field labels are named properly and the width values are large enough for the desired data | | |
| Create New AOA Report | Open “aoaReports/Template.rpa” to use as a base template for all AOA Reports    Steps:   1. Save-As to “Commission Cash Receipt.rpa” 2. Delete the fields and headers for RowType and Parameters 3. Modify the Query 4. Change the name and Query Description from “Template” to “Commission Cash Receipt”, click Next 5. Change Subject from “Template” to “Commission Cash Receipt”, click Next 6. Set Conditions if any (usually none are needed), click Next 7. Select Desired Columns in the order in which they appear by default in the report, usually all, except RECID or ROWID types, click Next 8. Select Sort Criteria, first field should always be “rowType”, if any xxSort fields exist, add them next, click Next 9. Click Finish | | |
| Design AOA Report | Steps:   1. Add selected fields to Detail Section (if added in order, select them all except rowType and parameter), drag them to the report form 2. Add Group Headers/Footers for SubTotals as needed 3. Add Grand Totals in ReportFooter as needed 4. Be sure all Sections are set to the desired size 5. Be sure the ReportHeader section is fully visible showing the Parameter objects 6. No need to name any objects, Parameters, Headers, Detail fields, Groups, Borders, etc. … a utility exists to properly name them according to desired standards of functionality (aoa/aoaRptMod.w) 7. When saving a DataPA report, make sure all the toggle boxes for Showing Sections are clicked and all Columns are selected (if not, you will leave the DataPA report in a state of change, especially the Parameters found in the ReportHeader section) 8. Save the report and run aoa/aoaRptMod.w to set the field names and column sizes | | |
| AOA Report Modifier | Run aoa/aoaRptMod.w    Steps:   1. Open desired program by double clicking the Prog ID [AR] [r-commcr.p] 2. Click “Set Names”… this will give all the AOA Report objects the correct naming convention 3. Click “Save” to apply the new names and just as importantly will place the scripts needed to allow selectable columns, show/hide sections and make the MetaData record containing values found in rowType and parameters fields found in “{aoa/tempTable/ttFields.i}, which is needed for all Temp-Tables used in Business Logic Subjects 4. Any Parameter fields not named are excess, return to DataPA Report Designer and remove them 5. Place the remaining Parameters into a desired layout and size the Report Header section into an appropriate height 6. Once the AOA Report is completed as desired set the Property values, within DataPA (Select File, select Properties)      1. Click “Publish” either in DatPA or from within AOA Report Modifier 2. Select Setup and then Managed Published      1. Select the AOA report name (Commission Cash Receipt) and do a right mouse click 2. Select “Copy Enterprise to URL Clipboard” which in this case generates <http://localhost:80/DataPA/Document.aspx?ID=9678E2EAD9294562812A613ADF175212>, the value after ?ID= is the “<DataPA PublishID>” coded into the aoa/r-commcr.p program. 3. Any of the scripts used can be accessed via the Script buttons located on the Modifier screen, careful consideration and complete testing needs to occur before releasing any script changes 4. Scripts are found in resources/vbScripts/Rpt.<section>.dat | | |
| Important Notes | Labels are everything with the DataPA world, any changes done to labels after the fact need to be done carefully and all aspects of the AOA Report need to be checked from A to Z to ensure nothing has been broken | | |