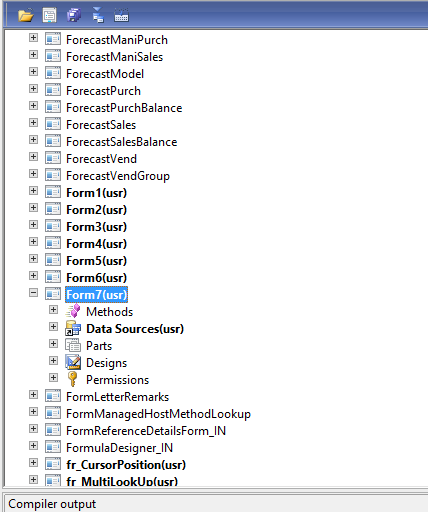
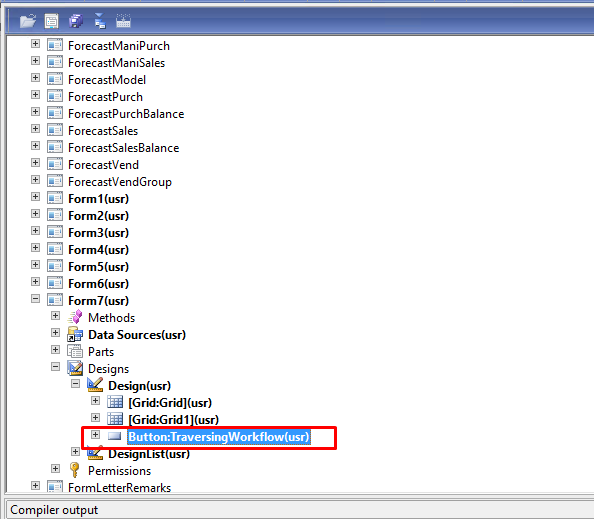
Tutorial for traversing all possible paths of workflow (binary tree like structure).

1. First of all open your form that you are using for workflow.



1. Add a button to this form and renamed it as TraversingWorkflow.



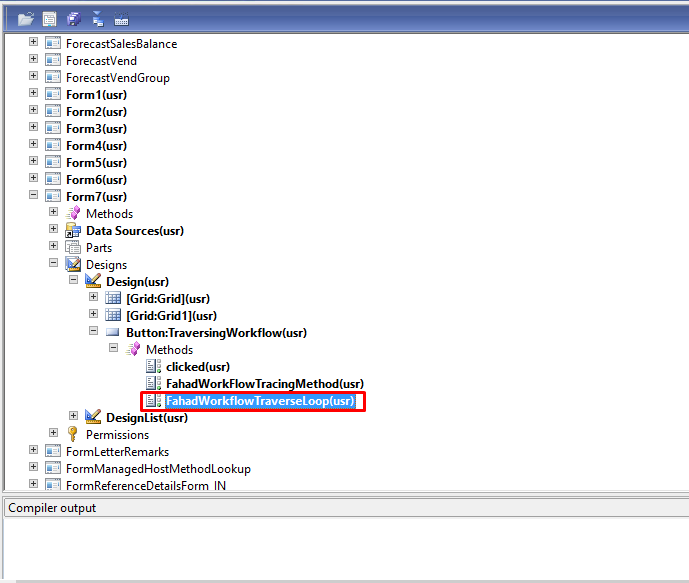
1. For traversing the path of the workflow we required four tables

WorkFlowTable -> WorkFlowVersionTable -> WorkFlowElementLinkTable -> WorkFlowElementTable.

1. Override the clicked method of the button that you have added in the last step with the below mentioned code.

|  |
| --- |
| **void** clicked()  {  **container** getDataFromTable,tempContainer,cont;  FWorkFlowTraverseClass object,tempObject,temp;  **str** myString="";  List listObj = **new** List(Types::Class);  ListIterator iterate;  **int** counter = **1**,i=**1**;  **int** j = **1**;  WorkflowTable workflowtable;  WorkflowVersionTable workflowversiontable;  WorkflowElementLinkTable workflowelementlinktable;  WorkflowElementTable workflowelementtable;  **container** firstPath,secondPath;  **super**();  *//info("Sardar");*  *//info(strFmt("%1",element.design().workflowType()));*  **while** **select** workflowtable **where** workflowtable.CategoryName == "fworkflowcategory" && workflowtable.Name == "fworkflowtypeType" && workflowtable.DefaultConfiguration == **true**  **join** workflowversiontable **where** workflowtable.RecId == workflowversiontable.WorkflowTable && workflowversiontable.Enabled == **true**  **join** workflowelementlinktable **where** workflowelementlinktable.WorkflowVersionTable == workflowversiontable.RecId  *//join workflowelementtable where workflowelementlinktable.SourceId == workflowelementtable.RecId*  {  tempContainer += [workflowelementlinktable.Level];  tempContainer += [workflowelementlinktable.SourceId];  tempContainer += [workflowelementlinktable.TargetId];  tempContainer += [**0**];  tempContainer += [**0**];  tempContainer += [**0**];  getDataFromTable+=tempContainer;  *// info(con2Str(tempContainer));*  */\**  *object = new FWorkFlowTraverseClass();*  *info("Printing");*  *object.setLevel(conPeek(getDataFromTable,counter));*  *info(int2str(object.getLevel()));*  *counter+=1;*  *object.setSrcID(conPeek(getDataFromTable,counter));*  *info(int642str(object.getSrcID()));*  *counter+=1;*  *object.setTrgtID(conPeek(getDataFromTable,counter));*  *info(int642str(object.getTrgtID()));*  *counter+=1;*  *info(strFmt("%1 %2 %3",object.getLeftVisited(),object.getRightVisited(),object.getNodeVisited()));*  *// listObj.addEnd(object);*  *//info(con2Str(getDataFromTable));*  *\*/*  tempContainer = **conNull**();  }  */\**  *iterate = new ListIterator(listObj);*  *while(iterate.more())*  *{*  *temp = iterate.value();*  *iterate.next();*  *//temp.setLeftVisited(true);*  *info(strfmt("%1 %2 %3 %4 %5 %6",temp.getLevel(),temp.getSrcID(),temp.getTrgtID(),temp.getLeftVisited(),temp.getRightVisited(),temp.getNodeVisited()));*  *}*  *\*/*  *//info(int2str(conlen(getDataFromTable)));*  **while**(i<**61**)  {  *//info(int2str(conPeek(getDataFromTable,i)));*  i+=**1**;  *//info(int642str(conPeek(getDataFromTable,i)));*  i+=**1**;  *//info(int642str(conPeek(getDataFromTable,i)));*  i+=**1**;  *//info(int2str(conPeek(getDataFromTable,i)));*  i+=**1**;  *// info(int642str(conPeek(getDataFromTable,i)));*  i+=**1**;  *// info(int642str(conPeek(getDataFromTable,i)));*  i+=**1**;  **break**;  }  this.FahadWorkFlowTracingMethod(**conPeek**(getDataFromTable,**1**),**conPeek**(getDataFromTable,**2**),**conPeek**(getDataFromTable,**3**),myString);  } |

1. Make a new method in the button



1. Add the following code into the method that you have created in the last step.

|  |
| --- |
| **public** **void** FahadWorkflowTraverseLoop(**int** LEVEL,**int64** SRCID,**int64** TRGTID)  {  *//Info(strFmt("%1 %2 %3",LEVEL,SRCID,TRGTID));*  **container** myCon,temp;  **int** lev;  **int64** src,trgt;  **container** tempCon;  **container** stepUser;  **str** conCatString="",travPath;  WorkflowElementTable workflowelementtable;  WorkflowElementLinkTable workflowelementlinktable;  WorkflowStepTable workflowsteptable;  WorkflowAssignmentTable workflowassignmenttable;  myCon+=[LEVEL];  myCon+=[SRCID];  myCon+=[TRGTID];  myCon+=[conCatString];  *//info(con2Str(myCon));*  *//info(int642str(conPeek(myCon,conLen(myCon))));*  **while**(**conLen**(myCon)>**0**)  {  travPath = **conPeek**(myCon,**conLen**(myCon));  temp = **conDel**(myCon,**conLen**(myCon),**conLen**(myCon));  myCon = temp;  trgt = **conPeek**(myCon,**conLen**(myCon));  temp = **conDel**(myCon,**conLen**(myCon),**conLen**(myCon));  myCon = temp;  src = **conPeek**(myCon,**conLen**(myCon));  temp = **conDel**(myCon,**conLen**(myCon),**conLen**(myCon));  myCon = temp;  lev = **conPeek**(myCon,**conLen**(myCon));  temp = **conDel**(myCon,**conLen**(myCon),**conLen**(myCon));  myCon = temp;  *//info("Con");*  *//info(int2str(conLen(myCon)));*  **if**(trgt==**0**)  {  stepUser = **conNull**();  **while** **select** workflowelementtable **where** workflowelementtable.RecId == src  {  *//info(workflowelementtable.Name);*  travPath+=workflowelementtable.Name;    **while** **select** workflowsteptable **where** workflowsteptable.ElementId == workflowelementtable.ElementId  **join** workflowassignmenttable **where** workflowsteptable.RecId == workflowassignmenttable.WorkflowStepTable  {  stepUser+=[workflowassignmenttable.UserValue];  }  travPath+="(";  travPath+=con2Str(stepUser);  travPath+=")";  travPath+="------->";  *//conCatString+=workflowelementtable.Name;*  *//conCatString+="------->";*  }  *// info("End");*  travPath+="End";  info(travPath);  travPath="";  }  **else**  {  **if**(src == **0**){  *//info("Start");*  travPath+="Start------->";  }  **else**{  stepUser = **conNull**();  **while** **select** workflowelementtable **where** workflowelementtable.RecId == src  {  *// info(workflowelementtable.Name);*  travPath+=workflowelementtable.Name;      **while** **select** workflowsteptable **where** workflowsteptable.ElementId == workflowelementtable.ElementId  **join** workflowassignmenttable **where** workflowsteptable.RecId == workflowassignmenttable.WorkflowStepTable  {  stepUser+=[workflowassignmenttable.UserValue];  }  travPath+="(";  travPath+=con2Str(stepUser);  travPath+=")";  travPath+="------->";  *//conCatString+=workflowelementtable.Name;*  *//conCatString+="------->";*  }  }  tempCon = **conNull**();  **while** **select** workflowelementlinktable **where** workflowelementlinktable.SourceId==trgt  {  tempCon+=[workflowelementlinktable.Level];  *// info("Level");*  *//info(int2str(conPeek(tempCon,1)));*  tempCon+=[workflowelementlinktable.SourceId];  *//info("Source ID");*  *//info(int642str(conPeek(tempCon,2)));*  tempCon+=[workflowelementlinktable.TargetId];  *//info("Target ID");*  *//info(int642str(conPeek(tempCon,3)));*  }  **if**(**conLen**(tempCon)>=**2**)  {  **if**(**conPeek**(tempCon,**1**)==**conPeek**(tempCon,**4**) && **conPeek**(tempCon,**2**)==**conPeek**(tempCon,**5**) && **conPeek**(tempCon,**3**)!=**conPeek**(tempCon,**6**))  {  *//this.FahadWorkFlowTracingMethod(conPeek(tempCon,1),conPeek(tempCon,2),conPeek(tempCon,3),conCatString);*  myCon+=[**conPeek**(tempCon,**1**)];  myCon+=[**conPeek**(tempCon,**2**)];  myCon+=[**conPeek**(tempCon,**3**)];  myCon+=[travPath];  *//this.FahadWorkFlowTracingMethod(conPeek(tempCon,4),conPeek(tempCon,5),conPeek(tempCon,6),conCatString);*  myCon+=[**conPeek**(tempCon,**4**)];  myCon+=[**conPeek**(tempCon,**5**)];  myCon+=[**conPeek**(tempCon,**6**)];  myCon+=[travPath];  }  **else**  {  *//this.FahadWorkFlowTracingMethod(conPeek(tempCon,1),conPeek(tempCon,2),conPeek(tempCon,3),conCatString);*  myCon+=[**conPeek**(tempCon,**1**)];  myCon+=[**conPeek**(tempCon,**2**)];  myCon+=[**conPeek**(tempCon,**3**)];  myCon+=[travPath];  }  }  **else**{  *// this.FahadWorkFlowTracingMethod(conPeek(tempCon,1),conPeek(tempCon,2),conPeek(tempCon,3),conCatString);*  myCon+=[**conPeek**(tempCon,**1**)];  myCon+=[**conPeek**(tempCon,**2**)];  myCon+=[**conPeek**(tempCon,**3**)];  myCon+=[travPath];  }  }  }  } |

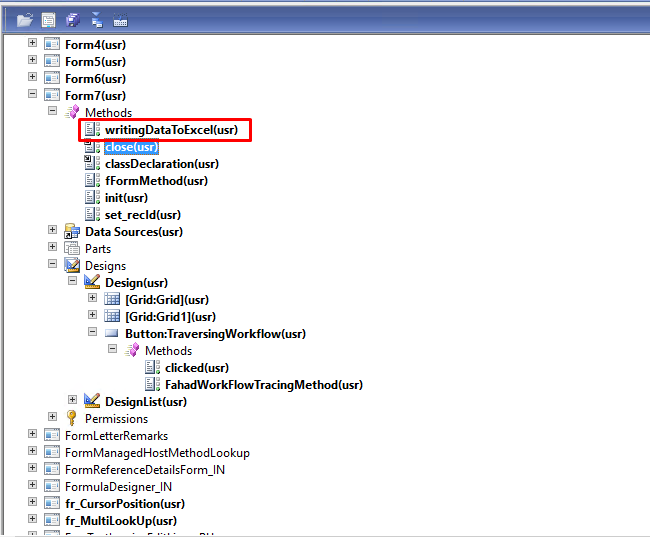
1. Override the class Declaration code with the following code

|  |
| --- |
| **public** **class** FormRun **extends** ObjectRun  {  **int64** RecID;  **container** storingPath;  } |

1. Override the close method of the form with the code below.

|  |
| --- |
| **public** **void** close()  {  *//info(con2Str(storingPath));*  this.writingDataToExcel(storingPath);  **super**();  } |

1. Make a new method “writingDataToExcel” into the form scope.

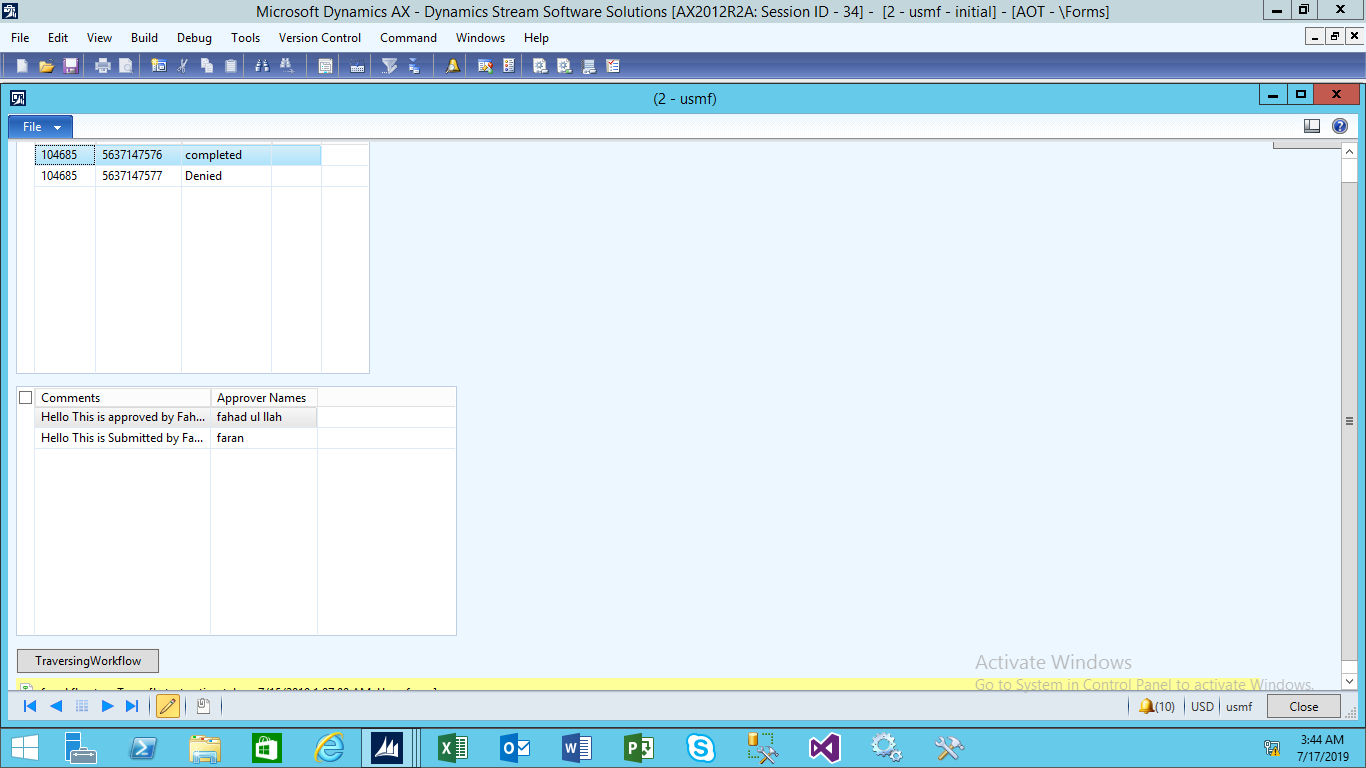


1. Add the following code in the method to write the traversed path string.

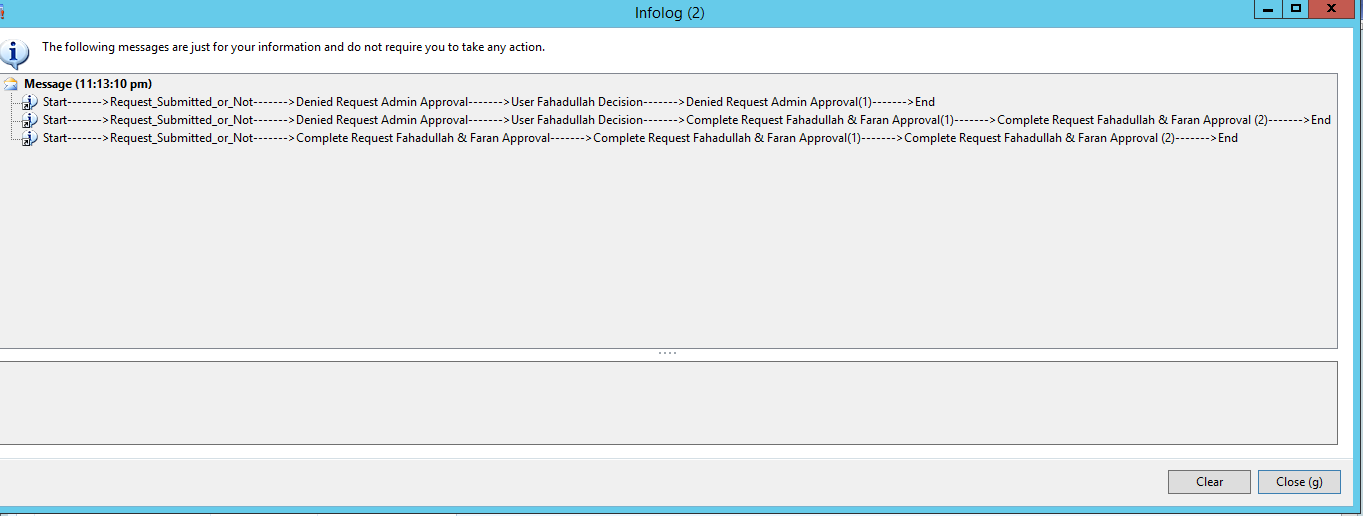
|  |
| --- |
| **public** **void** writingDataToExcel(**container** saveData)  {  *//info(conpeek(saveData));*  **int** i=**1**;  SysExcelApplication application;  SysExcelWorkbooks workbooks;  SysExcelWorkbook workbook;  SysExcelWorksheets worksheets;  SysExcelWorksheet worksheet;  SysExcelCells cells;  SysExcelCell cell;  **int** row;  ;  application = SysExcelApplication::construct();  workbooks = application.workbooks();  workbook = workbooks.add();  worksheets = workbook.worksheets();  worksheet = worksheets.itemFromNum(**1**);  cells = worksheet.cells();  cells.range('A:A').numberFormat('@');  cell = cells.item(**1**,**1**);  cell.value("Traveresed Path");  **while**(i<=**conLen**(saveData))  {  row++;  cell = cells.item(row, **1**);  cell.value(**conPeek**(saveData,i));  i++;  }  application.visible(**true**);  } |

1. Final output be like.

Snapshot: 1 (Open your form)



Snapshot: 2 (Click the traversing workflow button)



Snapshot: 3 (When you close this form)

