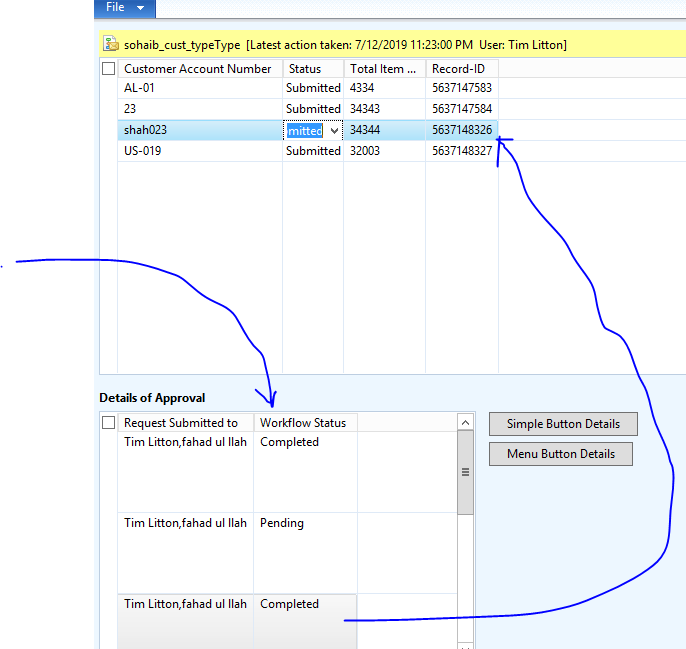
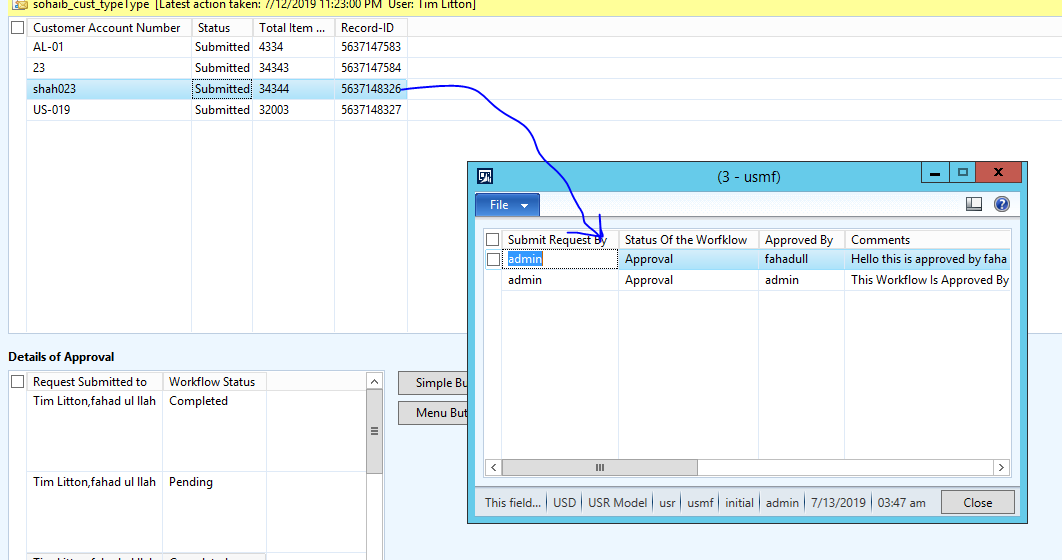
# Problem:

Now I am making a scenario that when ever I click on a record their will be a grid below that show the status of the workflow.

1. In the screen below when ever I am populating a grid by display method (Status of the Workflow and Users whom which workflow are assigned to Approve.  
   
2. Now I want to see that which user has approve the request of the following record. In this form I am getting the names of the user to whom request are submitted and in the next form I am viewing that who has accepted the workflow and who have not. **(Tim Litton is admin)** now when I click the button a new form populates and show the details of the form. (Both the buttons do the same things Just I am trying to pass the arguments and open another form through a simple button and Menu Item Button )(Function of both of the buttons are same)  
   

# Solution

Now to do all of this work. First of all, we have to write a display method into the data source which we are using to store our information of the workflow.

Go to the data Dictionary

Go to your data Source where you stored all the information of the workflow.

Go to the Methods of the data Source

Add new method.

## Data Dictionary Tables

### Code (Submitted to request of the Workflow Display Method)

*//Workflow Approvals Names*

[SysClientCacheDataMethodAttribute(**true**)]

**public** **display** **str** ApprovalNames() // Request Submmited to the User to accept the workflow

{

**container** daxCon;

**str** ResultSet;

workflowtrackingstatustable workflowtrackingstatustable;

WorkflowTrackingTable workflowtrackingtable;

WorkflowStepTable WorkflowStepTable;

WorkflowTrackingWorkItem workflowTrackingWorkItem;

WorkflowTrackingCommentTable workflowtrackingcommenttable;

WorkflowWorkItemTable workflowWorkItemTable;

**WHILE** **select** workflowtrackingstatustable

**join** workflowtrackingtable

**where** workflowtrackingstatustable.ContextRecId == this.RecId

&& workflowtrackingstatustable.ContextTableId == **tableNum**(Sohaib\_Workflow\_DB)

*//&& workflowtrackingstatustable.TrackingStatus == WorkflowTrackingStatus::Completed*

&& workflowtrackingtable.TrackingContext == workflowtrackingcontext::WorkItem

&& workflowtrackingtable.TrackingType == workflowtrackingtype::Creation

&& workflowtrackingtable.WorkflowTrackingStatusTable == workflowtrackingstatustable.RecId

{

daxCon += SysWorkflowHelper::getDirPartyNameFromUserId(workflowtrackingtable.USER);

}

ResultSet=con2Str (daxCon);

**if** (ResultSet=="0")

**return** "No Approved this workflow";

**else**{

**return** ResultSet;

}

}

### Code(Status of the Workflow Display Method)

*//Workflow Approvals Names*

[SysClientCacheDataMethodAttribute(**true**)]

**public** **display** **str** WorkflowStatus()

{

**container** daxCon;

**str** ResultSet;

workflowtrackingstatustable workflowtrackingstatustable;

**select** workflowtrackingstatustable

**where** workflowtrackingstatustable.ContextRecId == this.RecId

&& workflowtrackingstatustable.ContextTableId == **tableNum**(Sohaib\_Workflow\_DB);

ResultSet = **enum2str**(workflowtrackingstatustable.TrackingStatus);

**if** (ResultSet=="0")

**return** "Workflow Did not Exists";

**else**{

**return** ResultSet;

}

}

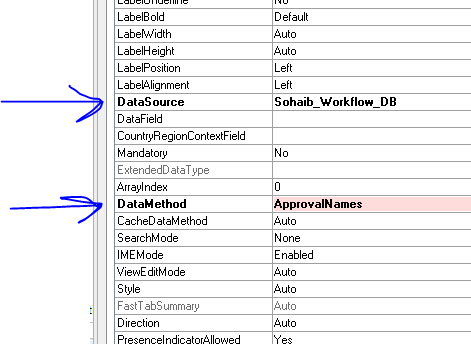
## Form

Now add Grid to you form

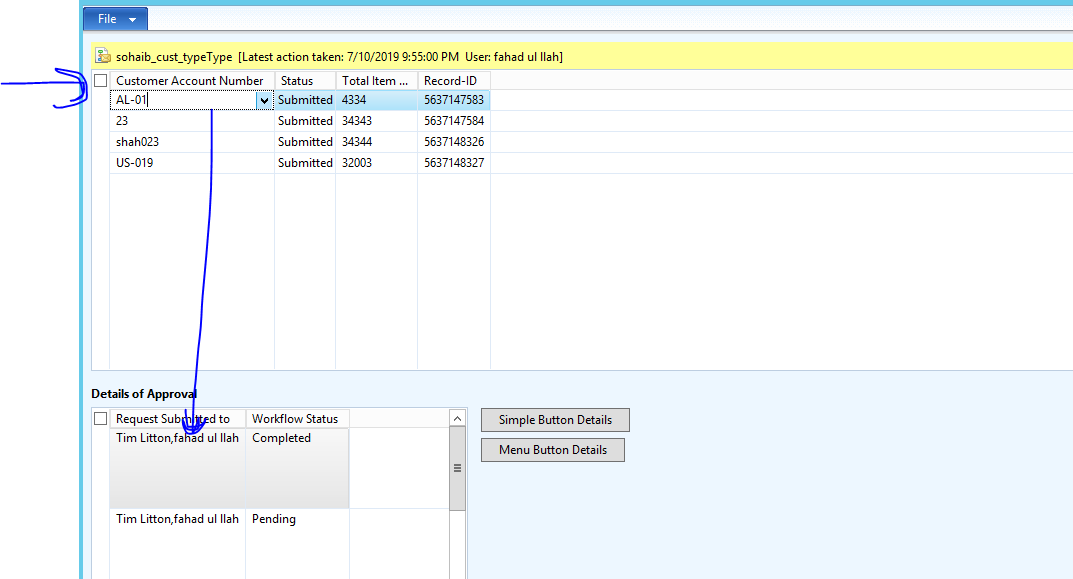
Add two String edits to the grid one for the User whom you send the workflow request to accept and other for the workflow status.

Now go to the string edit properties and specify Data Source and Data Method.

String edit will fetch the values from the display method that you have written on the Data Source method.



Now you when you click on different records detail of the following table will be highlighted in the grid below in which we specify methods.



### Second Form

Now what we have to doo we have to see all of the details of the following record on the next form. When ever we double click on the record a new form will open and show all of the details of that particular record. We store all the details in the temporary table because workflow status will be updated time by time so we need the latest status. When ever we click on record query will execute take values from 4 of the tables of the workflow and store it into the temporary table and populate that fields onto the grid.

#### Steps

Make a new form

Add grid on it

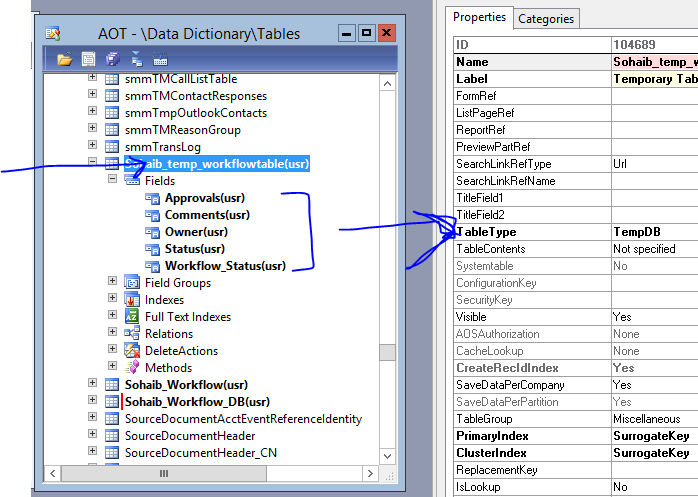
Go to the Data Dictionary

Add new table

Add all the fields you want to extract from the different table

Go to the properties of that table

Turn that table to temporary table by changing the (table type= None to table type= **TempDb**)



Now drag that table to the form data source

Drag fields of the table to the grid which you want to show onto the grid.

Add Data Source of the Grid and Data Fields.

Now declare Container and temp\_table object in the class decelartion method of the form.

##### Form Methods:

**public** **class** FormRun **extends** ObjectRun

{

**container** Ids\_In\_contain;

Sohaib\_temp\_workflowtable temp\_table; // temp table object

}

Now add One more method in which we will write query and get all the details of the record thorugh its recID.

**public** Sohaib\_temp\_workflowtable TempTable\_Insert(**container** \_Ids)

{

**container** Ids;

**int** iterator;

Sohaib\_Workflow\_DB \_Sohaib\_Workflow\_DB;

WorkflowTrackingStatusTable \_WorkflowTrackingStatusTable;

WorkflowTrackingTable \_WorkflowTrackingTable;

WorkflowTrackingCommentTable \_WorkflowTrackingCommentTable;

**int64** tableID;

**int64** recId;

Ids=\_Ids;

tableId=**conpeek**(Ids,**1**);

*// currently working for viewing only one workflow details*

**if**(Ids)

{

**for** (iterator=**2**;iterator<=**conLen**(Ids);iterator++) *// if more than 1 workflow are selected for viewing details*

{ recId=**conPeek**(Ids,iterator);

TableID=**tableNum**(Sohaib\_Workflow\_DB);

**While** **select** \* **from** \_WorkflowTrackingTable

**join** \_WorkflowTrackingStatusTable

**where** \_WorkflowTrackingStatusTable.ContextRecId == recId

&& \_WorkflowTrackingStatusTable.ContextTableId == TableID

&& \_WorkflowTrackingStatusTable.RecId == \_WorkflowTrackingTable.WorkflowTrackingStatusTable

&& \_WorkflowTrackingTable.TrackingContext == WorkflowTrackingContext::WorkItem

&& \_WorkflowTrackingTable.TrackingType ==WorkflowTrackingType::Approval

**join** \_WorkflowTrackingCommentTable

**where** \_WorkflowTrackingCommentTable.WorkflowTrackingTable == \_WorkflowTrackingTable.RecId

{

temp\_table.Owner=\_WorkflowTrackingStatusTable.Owner;

temp\_table.Status=**enum2str**(\_WorkflowTrackingTable.TrackingType);

temp\_table.Approvals=\_WorkflowTrackingTable.User;

temp\_table.Workflow\_Status=**enum2str**(\_WorkflowTrackingTable.TrackingType);

temp\_table.Comments=\_WorkflowTrackingCommentTable.Comment;

temp\_table.insert();

} *// while end loop*

}*// for end loop*

}*// if end loop*

**else**{

info("No Ids In Container");

}

**return** temp\_table;

}

In this code I am passng container contains (recID and tableID) of the Current Record which we double click on the previous form to see the details.

##### Data Source Method

Go the the form data source and override init method.

**public** **void** init()

{

**super**();

*// Check for passed arguments*

**if**( element.args() )

{

Ids\_In\_contain = str2con(element.args().parm());

Sohaib\_temp\_workflowtable.linkPhysicalTableInstance(element.TempTable\_Insert(Ids\_In\_contain));

}

**else**

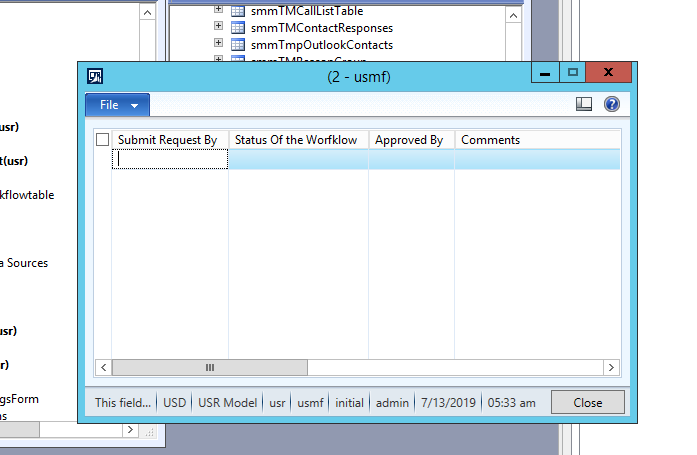
{

info("Arguments not Received");

}

}

In this method I am getting arguments from the previous form from which I have passed Table ID and recId. And pass that container to the this method I write in Form Method **public** Sohaib\_temp\_workflowtable TempTable\_Insert(**container** \_Ids)



## First Form:

Now we will pass recId when ever we press Button after selecting a particular record.

### Steps:

#### Pass Arguments through Simple Button:

Add a Button to the form and override the clicked method

**void** clicked()

{

MultiSelectionHelper selectionHelper = MultiSelectionHelper::construct();*//Create instance of class*

Set selectedRecords = **new** Set(Types::Record);*//Declare a set of type record*

Sohaib\_Workflow\_DB tableLines; *//Your table buffer*

Args args;

FormRun formRun;

RecId rec;

**container** conRecIds;

**str** temp\_string;

;

selectionHelper.parmDataSource(Sohaib\_Workflow\_DB\_DS); *//Set the datasource*

tableLines = selectionHelper.getFirst(); *//assign to table buffer the reference to selected record(s)*

**if** (tableLines.RecId)

{

conRecIds+=tableLines.TableId;

**while** (tableLines)

{

selectedRecords.add(tableLines);

conRecIds += **int642str**(tableLines.RecId);

tableLines = selectionHelper.getNext();

}

temp\_string=con2Str(conRecIds);

args = **new** args(**formstr**(Sohaib\_Cust\_details)); *// sending Args(record) to FormB*

args.parm(temp\_string);

formRun = classFactory.formRunClass(args);

formrun.run();

formRun.init();

formrun.wait();

}

**else**{

info("Didnt Get Rec ID");

}

**super**();

}

In this function I am selecting the RecID of the selected record and passing it to the next form.

Through Arguments.

#### Pass Arguments through Menu Item Button:

1. Now if you want to pass parameters through menu Item button.
2. Go the menu item button
3. Create menu Item button
4. Specify form which you want to open
5. Now add menu item button to the form
6. Specify menu Item in the properties of menuitem button
7. Override clicked method and write

**void** clicked()

{

MultiSelectionHelper selectionHelper = MultiSelectionHelper::construct();*//Create instance of class*

Set selectedRecords = **new** Set(Types::Record);*//Declare a set of type record*

Sohaib\_Workflow\_DB tableLines; *//Your table buffer*

Args args;

FormRun formRun;

RecId rec;

**container** conRecIds;

**str** temp\_string;

;

selectionHelper.parmDataSource(Sohaib\_Workflow\_DB\_DS); *//Set the datasource*

tableLines = selectionHelper.getFirst(); *//assign to table buffer the reference to selected record(s)*

**if** (tableLines.RecId)

{

conRecIds+=tableLines.TableId;

**while** (tableLines)

{

selectedRecords.add(tableLines);

conRecIds += **int642str**(tableLines.RecId);

tableLines = selectionHelper.getNext();

}

}

temp\_string=con2Str(conRecIds);

args = **new** Args();

args.caller(this);

args.parm(temp\_string);

**new** MenuFunction(**menuItemDisplayStr**(Sohaib\_Cust\_details), MenuItemType::Display).run(args);

}

# Final Forms

1st form will display all the workflows

2nd form display details

