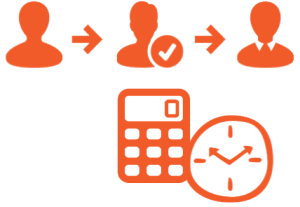
**WORKFLOW IN AX2012 FO:-  
  
Create custom workflow Microsoft Dynamics AX 2012**

workflow

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhYovNJCLzkBe9A7LAEiekZzhaM8P2VVouODY0JSZWH-flb-UMfzkDXqJDwjeAc6JnAl7C2MdjUbyLQOQbzRRcW75qfISON_igMxQZuBKOcLbR82Q7i_nJGZ9Uh-5t4z4rMnruvxypkaDw/s1600/Workflow.png)

Here we work on how to create custom workflow using x++ dynamics ax 2012. It’s simple and easy to do with the following steps in dynamics ax 2012.

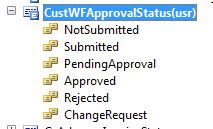
The following steps shows to create custom workflow for new created customer approval. Below are the objects needs to be created:

* Base Enum
* WorkflowApprovalStatus field
* Method on CustTable
* Query
* Workflow category
* Workflow type
* Workflow approval
* Drag workflow approval to workflow type
* Enable workflow on form

1. **BaseEnum:**

Create “**CustWFApprovalStatus**” base enum for the workflow approval status

* Not Submitted
* Submitted
* Pending Approval
* Change Request
* Approved
* Reject



1. **Create new Field on CustTable:**

Create Enum type field ‘**WorkflowApprovalStatus** ‘ on **CustTable** and assign **CustWFApprovalStatus** to enumtype

1. **Override Method on CustTable:**

Override **canSubmitToWorkflow** method on C**ustTable**

**public** **boolean** canSubmitToWorkflow(**str** \_workflowType = '')

{

**boolean** ret;

   ret = this.RecId != **0** && this.WorkflowApprovalStatus == CustWFApprovalStatus::NotSubmitted;

**return** ret;

}

**New Method on CustTable:**

Create new method **UpdateCustWorkflowState** on **CustTable** to update status

**public** **static** **void** UpdateCustWorkflowState(RefRecId \_recId, CustWFApprovalStatus \_state)

{

   CustTable custTable = CustTable::findRecId(\_recId, **true**);

**ttsBegin**;

   custTable.WorkflowApprovalStatus = \_state;

   custTable.update();

**ttsCommit**;

}

1. **Query:**

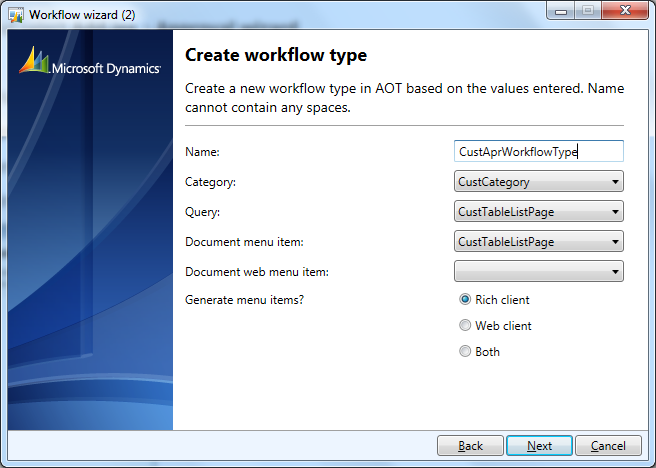
Here we use **CustTableListPage** query. Hence no need to create new query. Add **WorkflowApprovalStatus** field on query to display on customer list page.

1. **Workflow Category:**

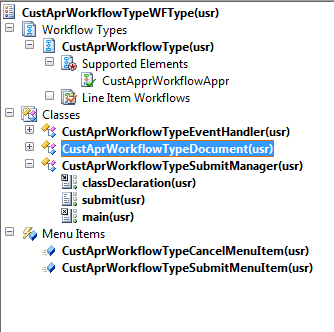
Here we use **CustCategory** as it has **customer module** attached with it.

1. **Workflow Type:**

1. In the **AOT**, expand the **Workflow node**.
2. Right-click the **Workflow Types** node, and then click **Add-Ins > Workflow type wizard**. The Workflow wizard is displayed. This wizard will help you create a new workflow type.
3. Click **Next**.
4. Set the following values for the wizard.



This will create a private project as below:



After the workflow type is created, you will add code for the workflow events.

Add below code to the **CustAprWorkflowTypeSubmitManager** class on workflow type project

**public** **static** **void** main(Args \_args)

{

*// Variable declaration.*

   CustTable                           CustTable;

   CustAprWorkflowTypeSubmitManager    submitManger;

   recId \_recId =                      \_args.record().RecId;

   WorkflowCorrelationId               \_workflowCorrelationId;

   workflowTypeName                    \_workflowTypeName = **workFlowTypeStr**("CustAprWorkflowType");

   WorkflowComment                     note = "";

   WorkflowSubmitDialog                workflowSubmitDialog;

   submitManger =                      **new** CustAprWorkflowTypeSubmitManager();

*//Opens the submit to workflow dialog.*

   workflowSubmitDialog = WorkflowSubmitDialog::construct(\_args.caller().getActiveWorkflowConfiguration());

   workflowSubmitDialog.run();

**if** (workflowSubmitDialog.parmIsClosedOK())

   {

       CustTable = \_args.record();

*// Get comments from the submit to workflow dialog.*

       note = workflowSubmitDialog.parmWorkflowComment();

**try**

       {

**ttsbegin**;

*// Activate the workflow.*

           \_workflowCorrelationId = Workflow::activateFromWorkflowType(\_workflowTypeName, CustTable.RecId, note, NoYes::No);

           CustTable.WorkflowApprovalStatus = CustWFApprovalStatus::Submitted;

           CustTable.update();

**ttscommit**;

*// Send an Infolog message.*

           info("Submitted to workflow.");

       }

**catch** (Exception::Error)

       {

           error("Error on workflow activation.");

       }

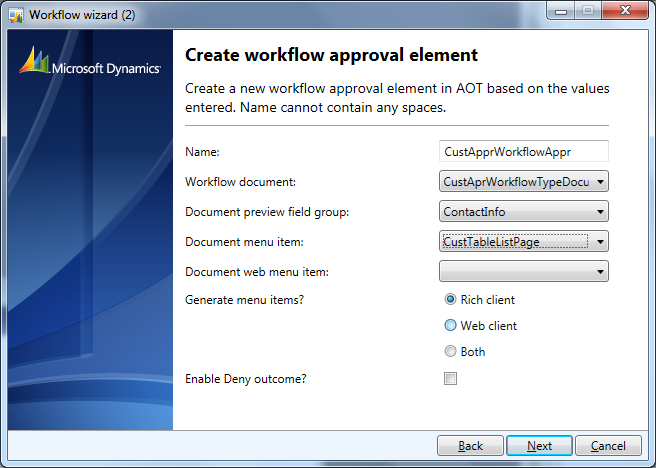
   }

   \_args.caller().updateWorkFlowControls();

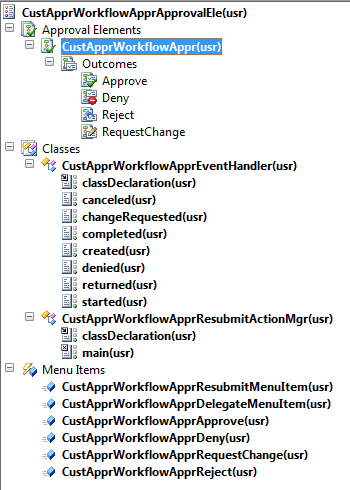
}

1. **Workflow Approval**

* 1. Open the **AOT**.
  2. Expand the **Workflow** node.
  3. Right-click on **Approvals** and select **Add-ins > Approval wizard**.
  4. Click **Next**.



This wizard will create the private project as below:



Now, Update code on CustApprWorkflowApprEventHandler for different events

**public** **void** started(WorkflowElementEventArgs \_workflowElementEventArgs)

{

   CustTable::UpdateCustWorkflowState(\_workflowElementEventArgs.parmWorkflowContext().parmRecId(),CustWFApprovalStatus::Submitted);

}

**public** **void** completed(WorkflowElementEventArgs \_workflowElementEventArgs)

{

   CustTable::UpdateCustWorkflowState(\_workflowElementEventArgs.parmWorkflowContext().parmRecId(), CustWFApprovalStatus::Approved);

}

**public** **void** changeRequested(WorkflowElementEventArgs \_workflowElementEventArgs)

{

   CustTable::UpdateCustWorkflowState(\_workflowElementEventArgs.parmWorkflowContext().parmRecId(),  CustWFApprovalStatus::ChangeRequest);

}

**public** **void** canceled(WorkflowElementEventArgs \_workflowElementEventArgs)

{

   CustTable::UpdateCustWorkflowState(\_workflowElementEventArgs.parmWorkflowContext().parmRecId(),  CustWFApprovalStatus::Rejected);

}

**Update Lables on Menuitems**

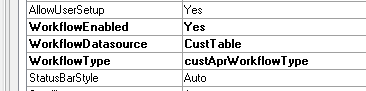
* CustApprWorkflowApprApprove  set label as **Approve**
* CustApprWorkflowApprReject set label as **Reject**
* CustApprWorkflowApprRequestChange set label as **Request change**

1. **Drag workflow approval to workflow type**

Drag workflow approval to workflow type under CustAprWorkflowType > Supported elements

1. **Enable workflow on form**

* Go to AOT > Forms > CustTableListPage
* Go to Designs > Design
* Right click > Properties

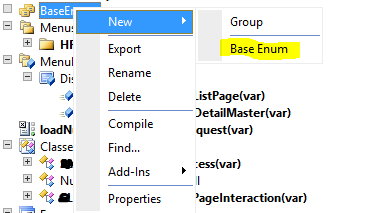


*That’s it now everything is setup and ready to go. But before start do* ***incremental CIL*** *and this is must. Every time you change the code, CIL is required.****WORKFLOW IN AX2012 FO:--***[**Microsoft Dynamics 365FO/AX Hub**](https://dynamicsaxtechnicalstuff.blogspot.com/)

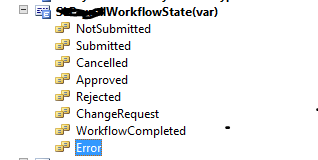
**Monday, 24 September 2018**

**Create Custom Workflow in AX 2012**

Today, I will be discussing about how to develop a custom workflow in AX 2012. Let's say we have a scenario in which we need to create a custom form for making any request and it requires approval from different authorities. Following are the steps for create custom workflow.  
  
Step 1) Create a new enum type. Enum type will be used for defining the state or status of the workflow. For this right on base enum group in your project and select Base Enum option.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEi-ufI2SCmLOXNb7dcBXFVIHkGC9KlhtL95ti_TwuUXVro6M0xEJyIH-z0aevAgZM80ddAw_qTy0dK1ahzvfR4Xcw39dny9OjYyEgKo6sap1yE5Vjx9qwweNmnRbvPS76oo27_QDXDr5aw/s1600/1.PNG)

Step 2) Add following elements in the enum.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiXLb9ntyyCBfxhTKqdmqoxvDWwJw3CsLfg4nQ2SUqgqNJexBmYoQrCFkwQqjgC7PunMWYxuOLkHo-NDYcdBIu9Dn_H56gevxfbWexFVK6PLS4G-grTLfkFKzylkod-H7OGOPQYnrKnB2Q/s1600/2.PNG)

Step 3) Add this enum in your custom table and override the canSubmitToWorkflow() on table. This method will be used for defining workflow submission criteria or condition on which workflow will be enabled.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEjKIq-zNOUBlQdV4IAl_fHOITPP57MS2F_9mWFQlwhZkRv7AsBti7gg4eQqXkuhED6iMVqFVvalPudWTwI9tyws8CwA7xoe8cCVgf24D8G2scs_9tqeAO2tGpy8tZgsvW7XijOSBB2oZls/s1600/3.PNG)

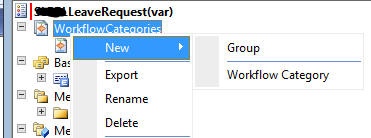
Step 4) Add another method for updating the status of the workflow.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgNm3SbJSTOJVOi8zyxgIOQu8B4xu6PzgFbUpySO8vZ1EhAKNUdXuNQ5DONkHn1Hbim2A0uQ3o5_KbJ4PwMu4JRsGzZ_Noga2-rHeglKvMLLdN9OyC1XEDgwULFr2XniQZN_8uLzUJ2rwA/s1600/4.PNG)

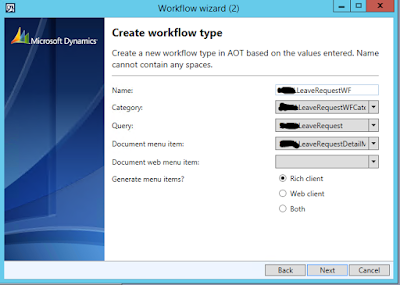
Step 5) Now create a custom query.  Workflow will be using this query to define the tables and fields in order to define the data set available. For this right click on queries group in your project and create a new query.  Add your custom table as data source in query and set dynamic field value to yes in order to show all fields.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEhlQ7CNfcdJqAJA5gYU8HGt25RawASBjHSSdImwBROykzvzm0_qo_piDKA8j2Xcxu49823PUJ52CnePCJJy4gcdZssvrJ0vBIEY9IQ0HFnLaZ1UuKoixv1zU364uURBHGhBznDPxrYwpvE/s1600/5.PNG)

Step 6) Create a new workflow category. It will determine the module in which workflow will be enabled. For this right click on workflow category group in your project and create a new category. Set its label and select the module in which you want your workflow to be enabled. In my case I have selected Human resource module.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEgVCXFVm6i-jleut45nnIjrX90XOHfLPSvqNaMCEJaNrZzxkpsBibrZdQ9JHnWLhLAOPdgf2pWLivHezHMtSDvrDcMNPLiFDpl53tyFlV6jvtR_rnanpxLDL8O0C_qSK7Zt6xR5jbjGpC4/s1600/6.PNG)

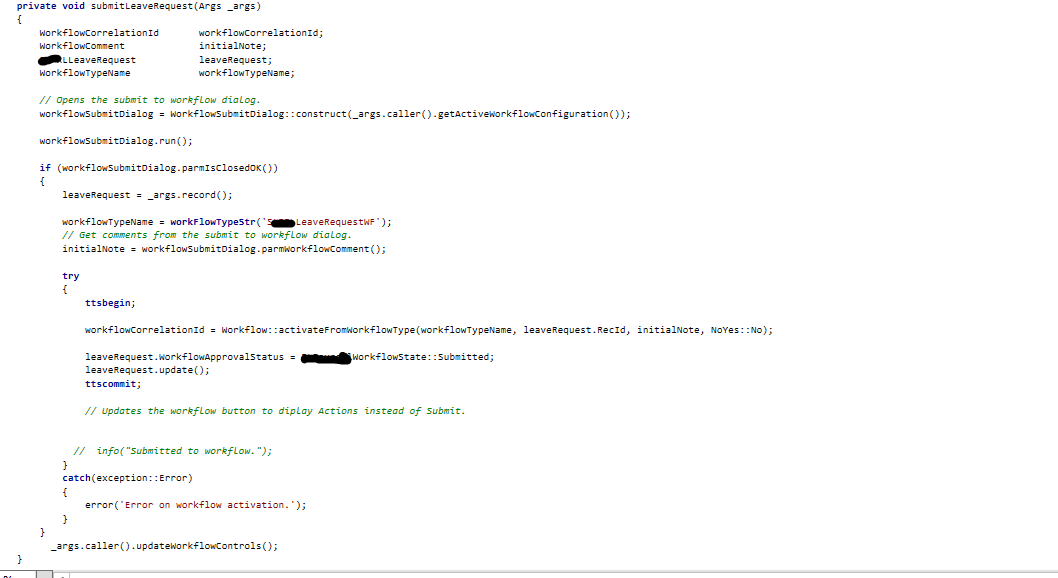
Step 7) Create a new workflow type using the wizard. For this go to AOT-> Workflows -> Workflow types -> Right click on it -> Addins -> Workflow type wizard.  You need to set out here workflow category, document menu item  and query.  
  
Category      - Name of workflow category  
Query           - Name of the query  
Document    - Name of the display menu item for document form  
  
On its generation following artifacts will be created.  
1. Workflow type  
2. Classes  
      - Document class it will extend workflow document  
      - Event handler class it will used for implementation of workflow events  
      - Submit Manager class  
3. Action menu item  
     - Submit menu item - Points to submit manager class  
     - Cancel menu item - Points to workflow cancel manager class

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiyAA-7wr85K3MQTij_q8BU1ZX8701j097a5WGotzp8sHKQ6Z5-frlIDCLVyCuKMqQnpNGTmgI6-69JiJ7RMoqB7hCGXQzC6hSHRoJKj8pueJtG1B5B9ouZjHl3inYBs-oeH9lrl6afMIA/s1600/7.PNG)

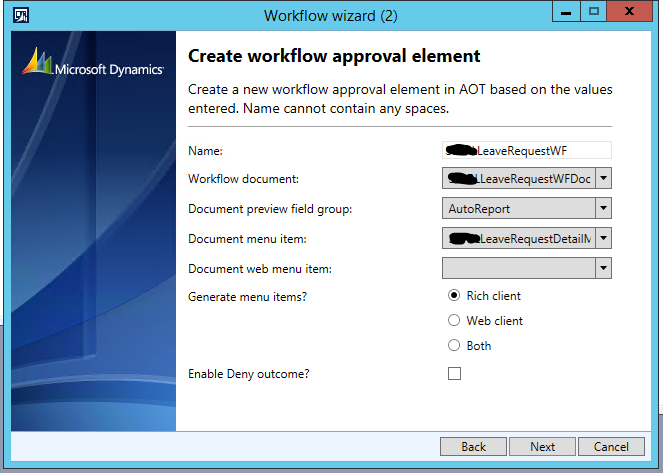
Step 8) Enable workflow on your custom detail form and custom list page. For this you need to set below mentioned properties on design of form.

[](https://blogger.googleusercontent.com/img/b/R29vZ2xl/AVvXsEiqe-AdDXupOk0kmd-4Y2KKkvFFzKtbpU1jcTgNrXRukFBlknX1nmE99dqI7uxTWMIyEQGUjoVN64Ky1VkgpVoBesweBqJsIxHOWmBcZ-TKquPOWjAbmbuMSuJzrdhIfm6ztAffkDLsLNQ/s1600/8.PNG)

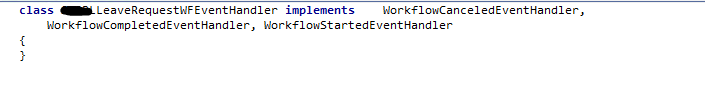
Step 9) Add logic in Submit Manager class. I have create a separate method with submitLeaveRequest naming and calling it from main method of submit manager class.

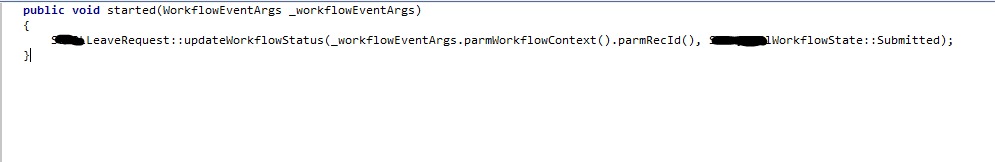


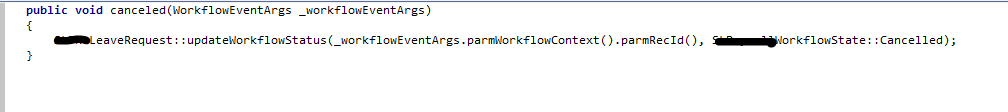
Step 10) Create a new workflow approval element using wizard. For this go to AOT -> Workflow -> Approvals -> Add-ins ->  Approval wizard  
Add following information on it and it will create list of artifacts  
- Name of workflow approval  
- Workflow document  
- Document menu item  
    -  Document field group

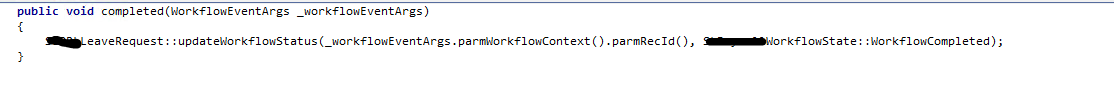


Step 11) Now add code in workflow event handler class and workflow approval event handler class.  
**Workflow event handler class code**

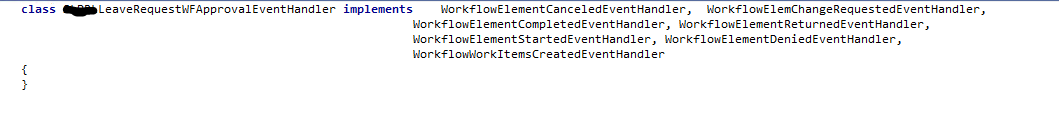


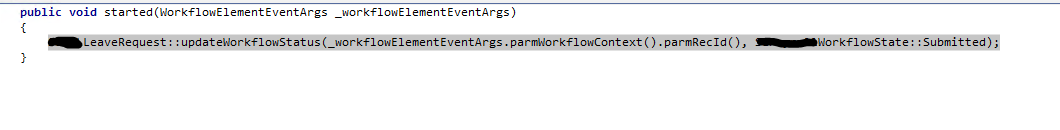


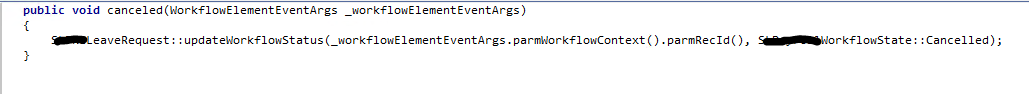


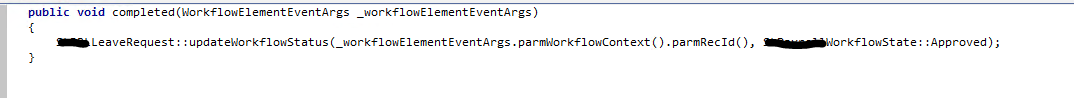


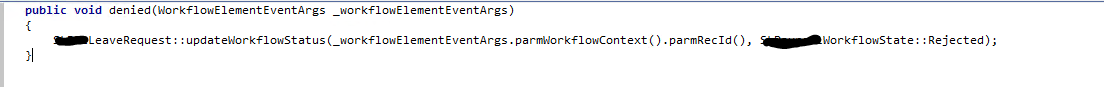
**Workflow approval event handler class**

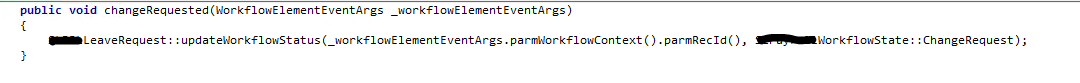


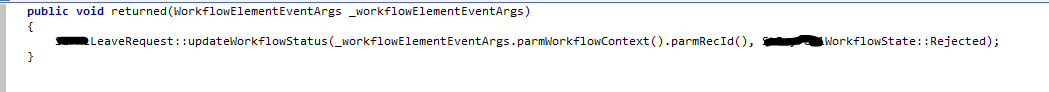






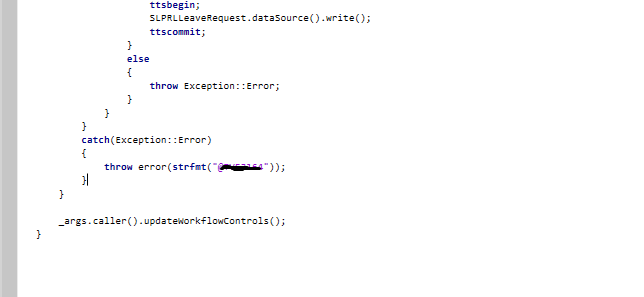




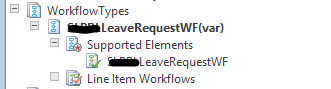


Step 12) Now add code in newly created Resubmit Manager class.

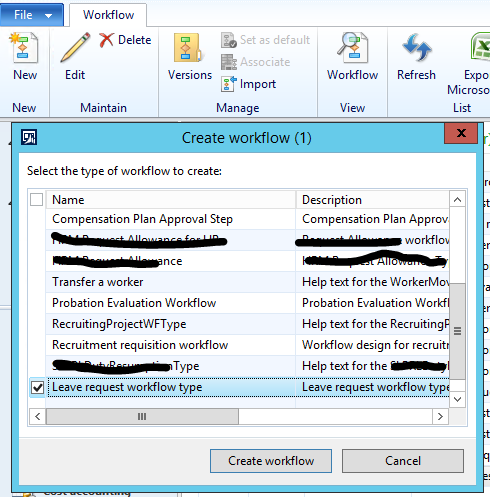




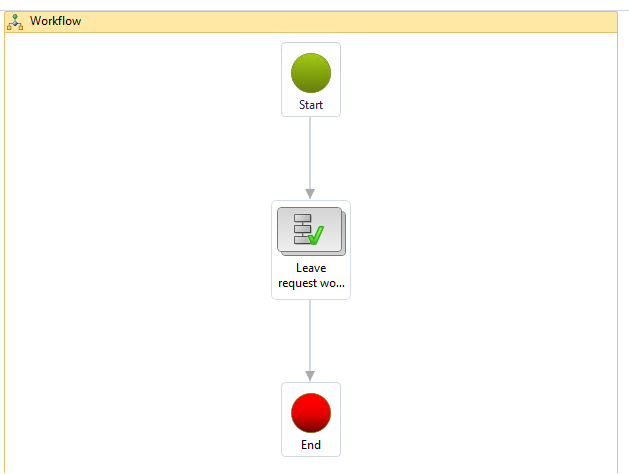
Step 13) Add workflow approval item in workflow type. For this expand your workflow type and add approval element in supported elements node.



Step 14) Now its time for designing the workflow. I am creating a workflow in human resource module. So for this Go to Human resource -> Setup -> Human resource workflows  
Click on new workflow button and select your custom workflow type then click on create  workflow button  
**Note: Before designing new workflow. Execute incremental CIL.**



Step 14) Define start and end states of workflow.  
For this you need to go through below mentioned steps  
- Drag your approval elements from Toolbox on the left of designer.  
- Connect the bottom of start state with top of approval element.  
- Connect the bottom of approval elements with top of the end state.  
- Resolve issue : For instance you may require to add step. For this double click on approval element. It will navigate you to step screen. Now right click on step and open its properties add assignments, workflow item subject and work flow item instructions if required.  
- Activate your workflow



Step 15) Go to your list page form and you will see your workflow bar there.

