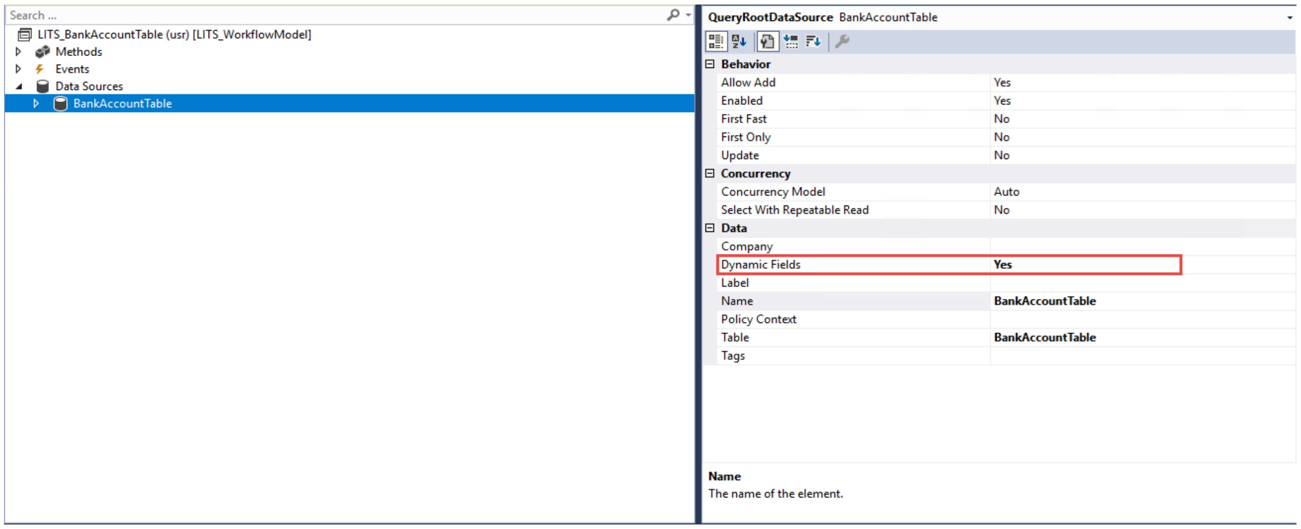
Custom workflow for Bank Account in Microsoft Dynamics 365 for Operations

Bilal Ahmed

- **Make a query and add BankAccountTable to datasource and set properties dynamic fields to Yes.**



Search ...

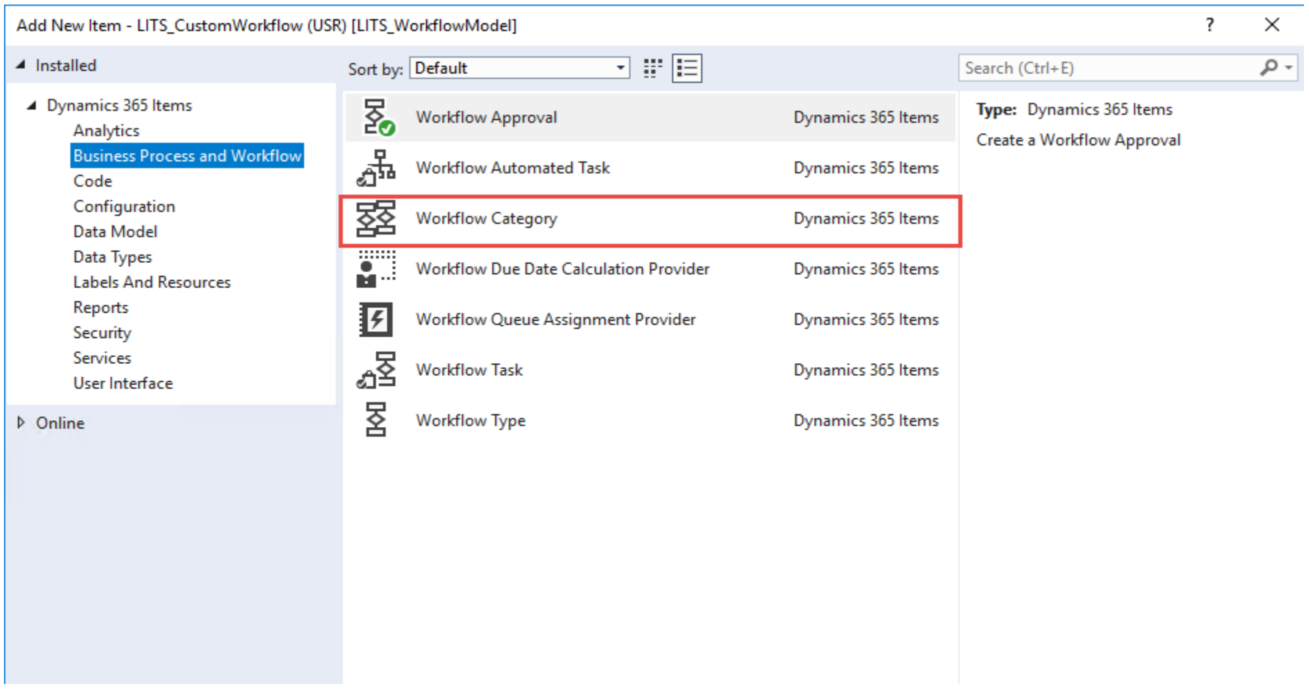
LITS_BankAccountTable (usr) [LITS_WorkflowModel]
- Methods
- Events
- Data Sources
  - BankAccountTable

QueryRootDataSource BankAccountTable

Behavior	
Allow Add	Yes
Enabled	Yes
First Fast	No
First Only	No
Update	No
Concurrency	
Concurrency Model	Auto
Select With Repeatable Read	No
Data	
Company	
Dynamic Fields	**Yes**
Label	
Name	**BankAccountTable**
Policy Context	
Table	**BankAccountTable**
Tags	

Name
The name of the element.

- **Add workflow category to your project.**



Add New Item - LITS_CustomWorkflow (USR) [LITS_WorkflowModel]

Installed
- Dynamics 365 Items
  - Analytics
  - Business Process and Workflow
  - Code
  - Configuration
  - Data Model
  - Data Types
  - Labels And Resources
  - Reports
  - Security
  - Services
  - User Interface

Online

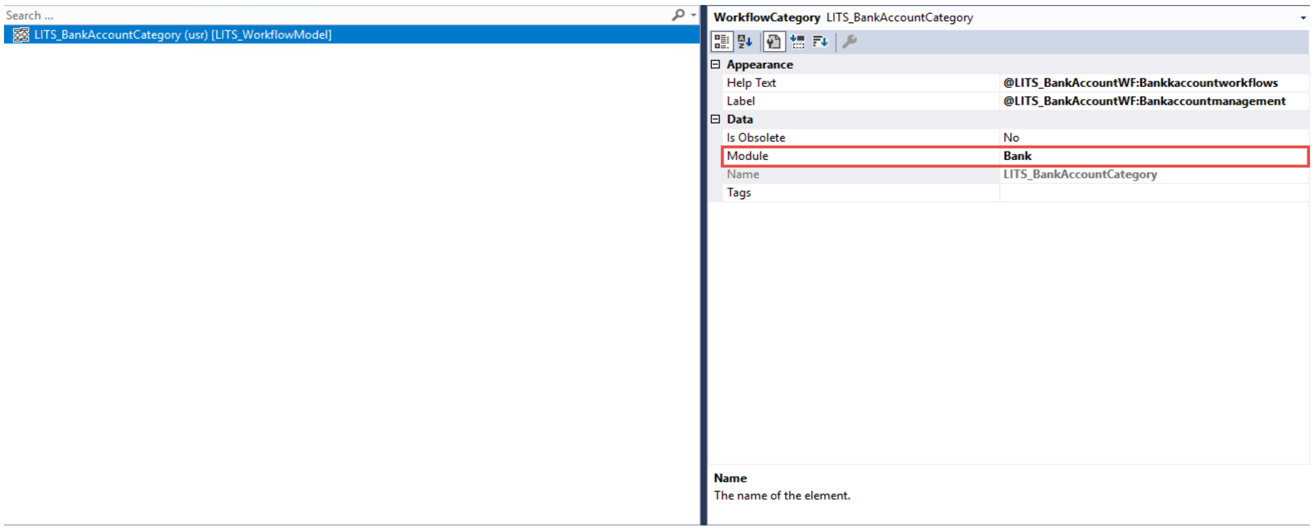
Sort by: Default

Search (Ctrl+E)

Workflow Approval	Dynamics 365 Items
Workflow Automated Task	Dynamics 365 Items
Workflow Category	Dynamics 365 Items
Workflow Due Date Calculation Provider	Dynamics 365 Items
Workflow Queue Assignment Provider	Dynamics 365 Items
Workflow Task	Dynamics 365 Items
Workflow Type	Dynamics 365 Items

Type: Dynamics 365 Items
Create a Workflow Approval

- Set module to **Bank**.



Search ...

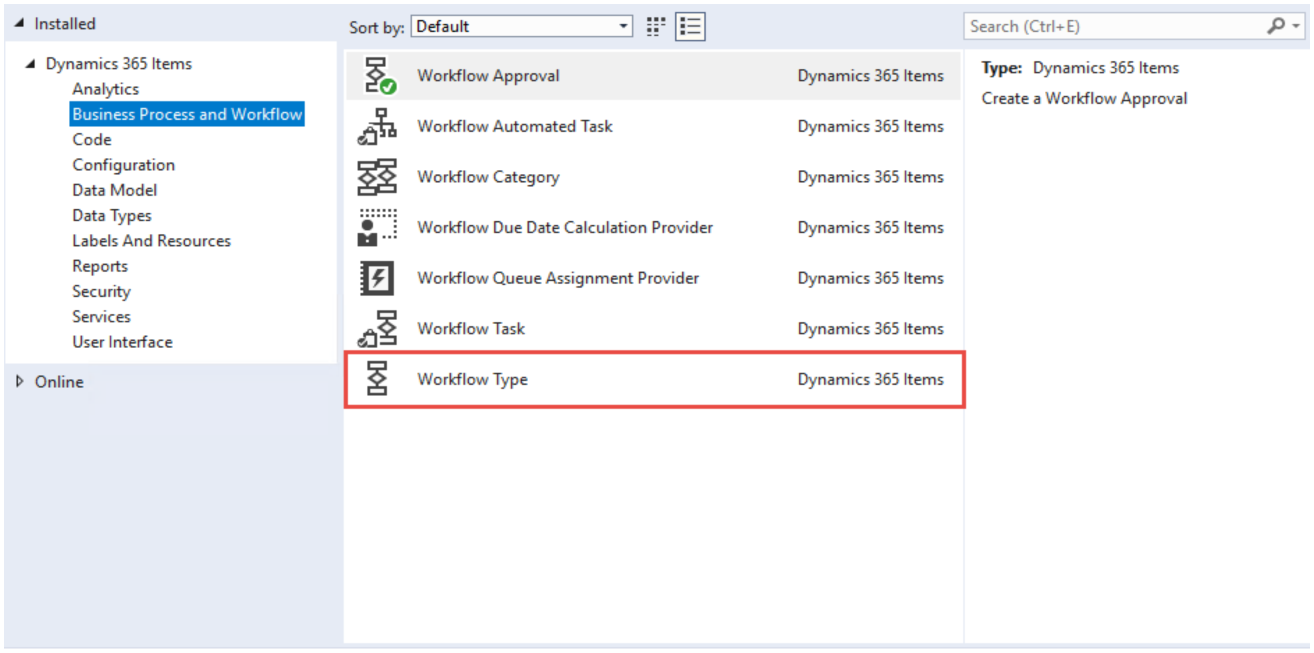
LITS_BankAccountCategory (usr) [LITS_WorkflowModel]

WorkflowCategory LITS_BankAccountCategory

Appearance	
Help Text	@LITS_BankAccountWF:Bankkaccountworkflows
Label	@LITS_BankAccountWF:Bankaccountmanagement
Data	
Is Obsolete	No
Module	Bank
Name	LITS_BankAccountCategory
Tags	

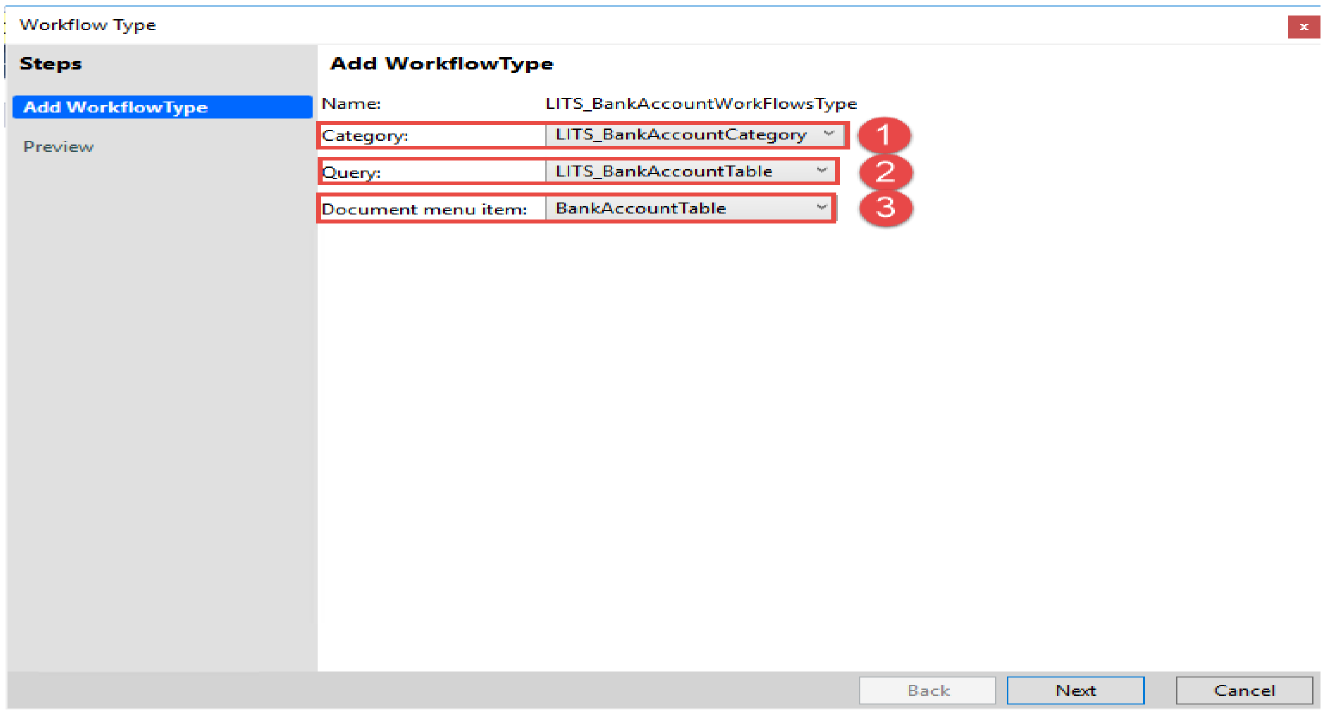
Name
The name of the element.

- **Add workflow type.**



Workflow Approval	Dynamics 365 Items
Workflow Automated Task	Dynamics 365 Items
Workflow Category	Dynamics 365 Items
Workflow Due Date Calculation Provider	Dynamics 365 Items
Workflow Queue Assignment Provider	Dynamics 365 Items
Workflow Task	Dynamics 365 Items
Workflow Type	Dynamics 365 Items

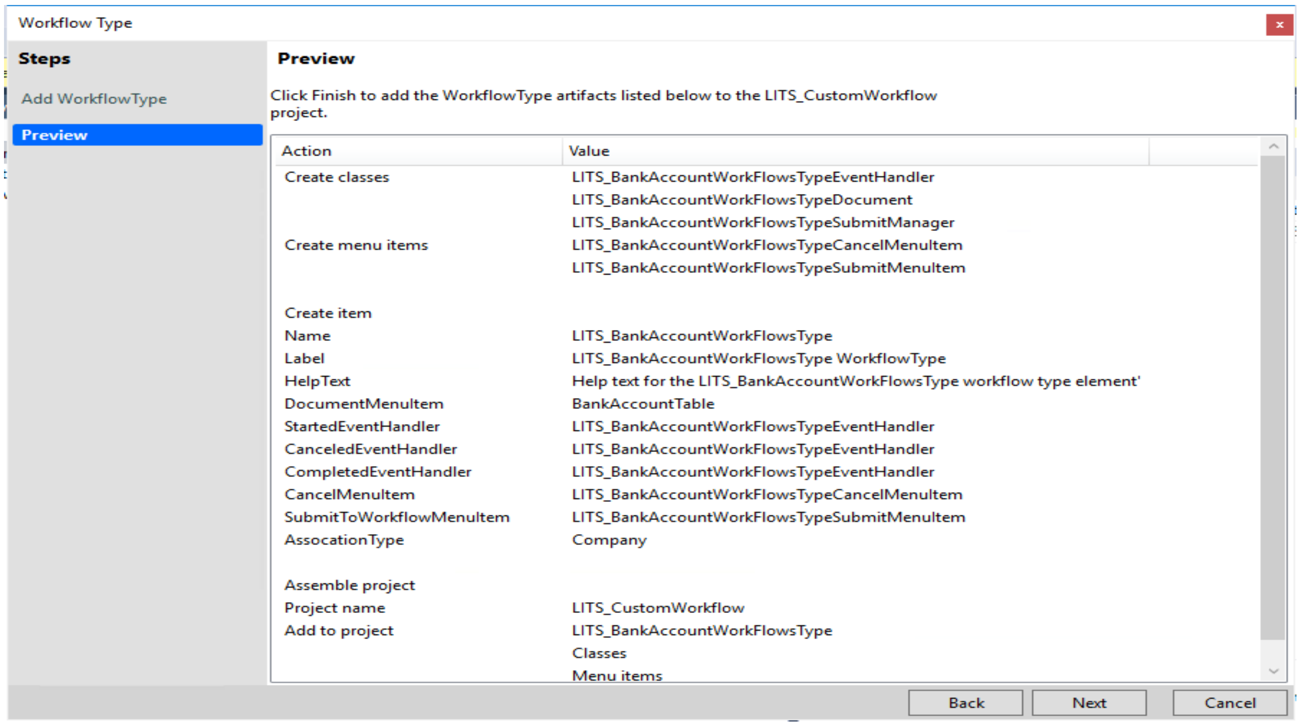
Type: Dynamics 365 Items
Create a Workflow Approval



Name:	LITS_BankAccountWorkFlowsType
Category:	LITS_BankAccountCategory
Query:	LITS_BankAccountTable
Document menu item:	BankAccountTable

1. Select workflow category.
2. Select the query.
3. Select form's menuitem.

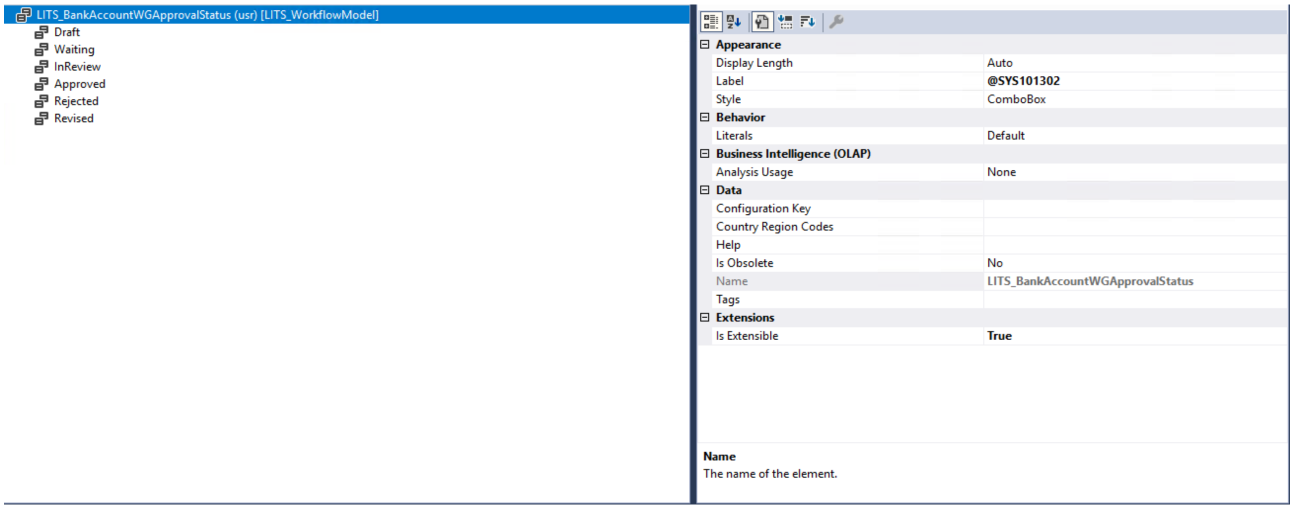
All these items will be added to your project.



Click Finish to add the WorkflowType artifacts listed below to the LITS_CustomWorkflow project.

Action	Value
Create classes	LITS_BankAccountWorkFlowsTypeEventHandler
	LITS_BankAccountWorkFlowsTypeDocument
	LITS_BankAccountWorkFlowsTypeSubmitManager
Create menu items	LITS_BankAccountWorkFlowsTypeCancelMenuItem
	LITS_BankAccountWorkFlowsTypeSubmitMenuItem
Create item	
Name	LITS_BankAccountWorkFlowsType
Label	LITS_BankAccountWorkFlowsType WorkflowType
HelpText	Help text for the LITS_BankAccountWorkFlowsType workflow type element'
DocumentMenuItem	BankAccountTable
StartedEventHandler	LITS_BankAccountWorkFlowsTypeEventHandler
CanceledEventHandler	LITS_BankAccountWorkFlowsTypeEventHandler
CompletedEventHandler	LITS_BankAccountWorkFlowsTypeEventHandler
CancelMenuItem	LITS_BankAccountWorkFlowsTypeCancelMenuItem
SubmitToWorkflowMenuItem	LITS_BankAccountWorkFlowsTypeSubmitMenuItem
AssocationType	Company
Assemble project	
Project name	LITS_CustomWorkflow
Add to project	LITS_BankAccountWorkFlowsType
	Classes
	Menu items

- **Add base enum and set the label for each element.**

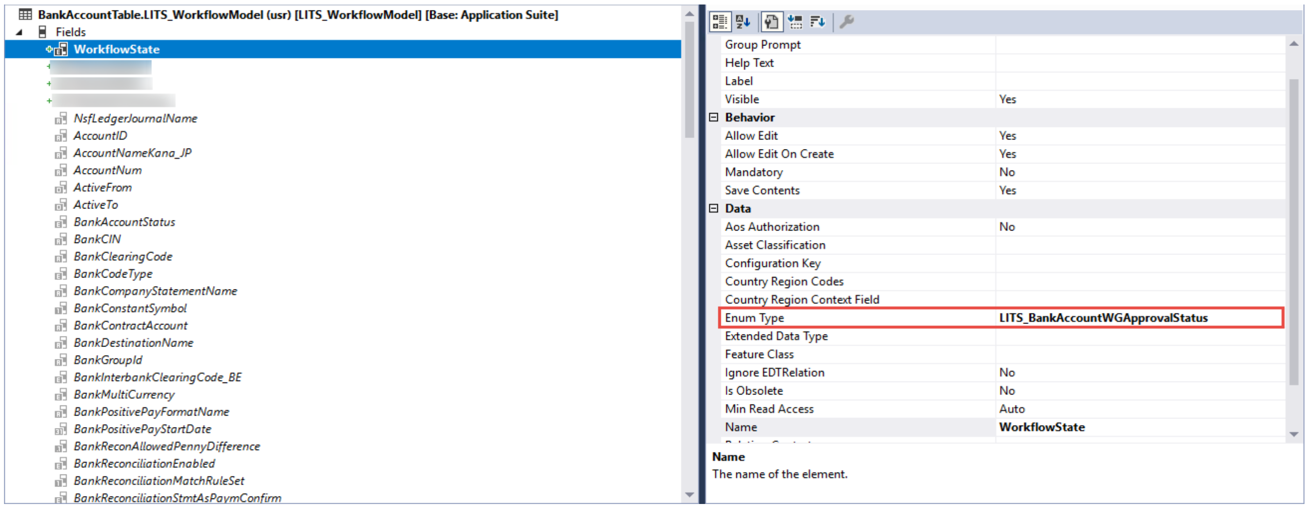


LITS_BankAccountWGApprovalStatus (usr) [LITS_WorkflowModel]
- Draft
- Waiting
- InReview
- Approved
- Rejected
- Revised

Appearance	
Display Length	Auto
Label	**@SYS101302**
Style	ComboBox
Behavior	
Literals	Default
Business Intelligence (OLAP)	
Analysis Usage	None
Data	
Configuration Key	
Country Region Codes	
Help	
Is Obsolete	No
Name	LITS_BankAccountWGApprovalStatus
Tags	
Extensions	
Is Extensible	**True**

Name
The name of the element.

- **Create extension of BankAccountTable and add fields.**

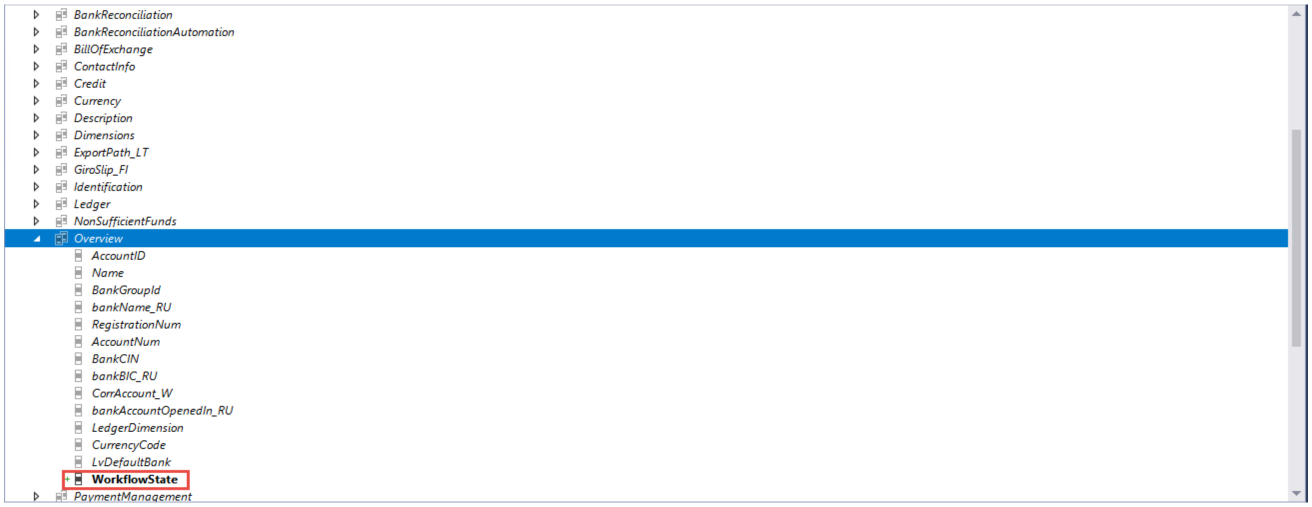


BankAccountTable.LITS_WorkflowModel (usr) [LITS_WorkflowModel] [Base: Application Suite]
- Fields
  - WorkflowState
  - [redacted]
  - *NsfLedgerJournalName*
  - *AccountID*
  - *AccountNameKana_JP*
  - *AccountNum*
  - *ActiveFrom*
  - *ActiveTo*
  - *BankAccountStatus*
  - *BankCIN*
  - *BankClearingCode*
  - *BankCodeType*
  - *BankCompanyStatementName*
  - *BankConstantSymbol*
  - *BankContractAccount*
  - *BankDestinationName*
  - *BankGroupId*
  - *BankInterbankClearingCode_BE*
  - *BankMultiCurrency*
  - *BankPositivePayFormatName*
  - *BankPositivePayStartDate*
  - *BankReconAllowedPennyDifference*
  - *BankReconciliationEnabled*
  - *BankReconciliationMatchRuleSet*
  - *BankReconciliationStmtAsPaymConfirm*

Group Prompt	
Help Text	
Label	
Visible	Yes
Behavior	
Allow Edit	Yes
Allow Edit On Create	Yes
Mandatory	No
Save Contents	Yes
Data	
Aos Authorization	No
Asset Classification	
Configuration Key	
Country Region Codes	
Country Region Context Field	
Enum Type	**LITS_BankAccountWGApprovalStatus**
Extended Data Type	
Feature Class	
Ignore EDTRelation	No
Is Obsolete	No
Min Read Access	Auto
Name	**WorkflowState**

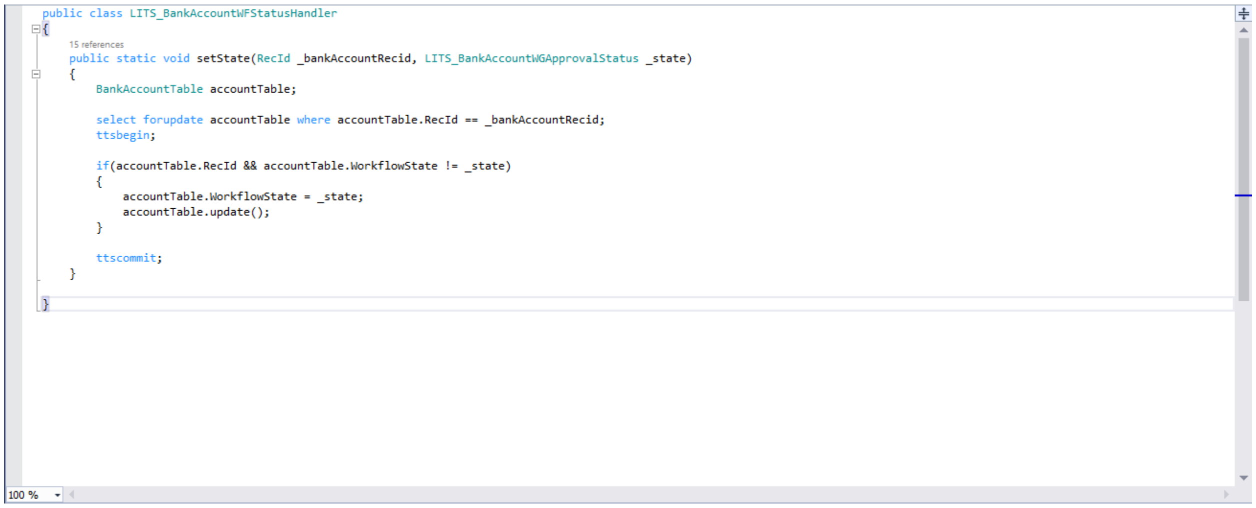
Name
The name of the element.

- **Add field in Overview field group.**



BankReconciliation
BankReconciliationAutomation
BillOfExchange
ContactInfo
Credit
Currency
Description
Dimensions
ExportPath_LT
GiroSlip_FI
Identification
Ledger
NonSufficientFunds
Overview
AccountID
Name
BankGroupId
bankName_RU
RegistrationNum
AccountNum
BankCIN
bankBIC_RU
CorrAccount_W
bankAccountOpenedIn_RU
LedgerDimension
CurrencyCode
LvDefaultBank
WorkflowState
PaymentManagement

- **Add status handler class.**



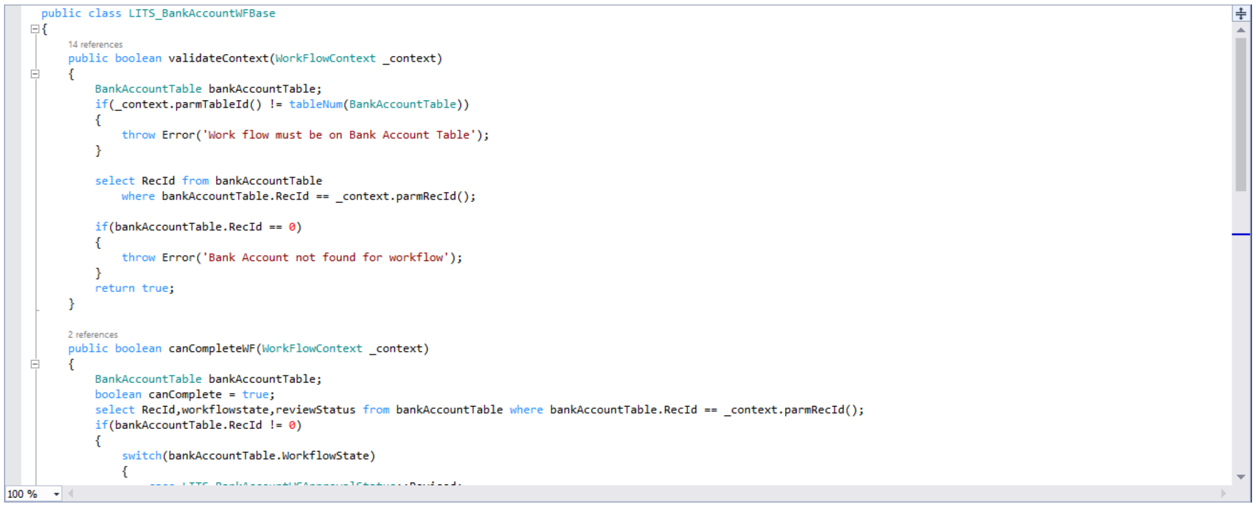
```
public class LITS_BankAccountWFStatusHandler
{
    15 references
    public static void setState(RecId _bankAccountRecid, LITS_BankAccountWGApprovalStatus _state)
    {
        BankAccountTable accountTable;

        select forupdate accountTable where accountTable.RecId == _bankAccountRecid;
        ttsbegin;

        if(accountTable.RecId && accountTable.WorkflowState != _state)
        {
            accountTable.WorkflowState = _state;
            accountTable.update();
        }

        ttscommit;
    }

}
```

100 %

Code for Status handler class:

```
public class LITS_BankAccountWFStatusHandler
{
    public static void setState(RecId _bankAccountRecid, LITS_BankAccountWGApprovalStatus
_state)
    {
        BankAccountTable accountTable;

        select forupdate accountTable where accountTable.RecId == _bankAccountRecid;
        ttsbegin;

        if(accountTable.RecId && accountTable.WorkflowState != _state)
        {
            accountTable.WorkflowState = _state;
            accountTable.update();
        }
        ttscommit;
    }

}
```

- **Add a base class**



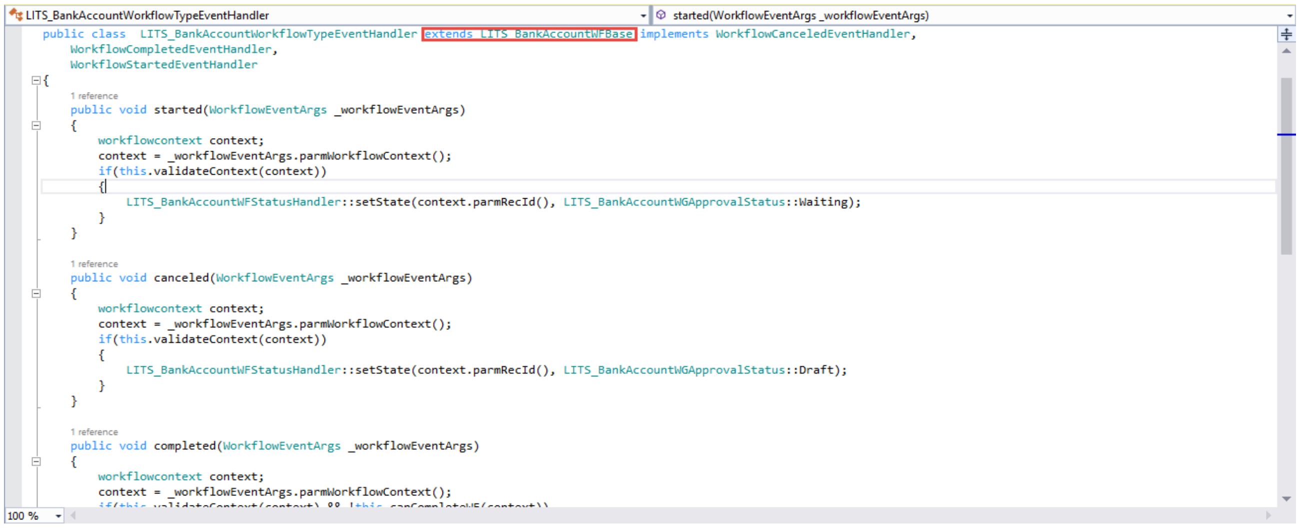
```
public class LITS_BankAccountWFBase
{
    14 references
    public boolean validateContext(WorkFlowContext _context)
    {
        BankAccountTable bankAccountTable;
        if(_context.parmTableId() != tableNum(BankAccountTable))
        {
            throw Error('Work flow must be on Bank Account Table');
        }

        select RecId from bankAccountTable
            where bankAccountTable.RecId == _context.parmRecId();

        if(bankAccountTable.RecId == 0)
        {
            throw Error('Bank Account not found for workflow');
        }
        return true;
    }

    2 references
    public boolean canCompleteWF(WorkFlowContext _context)
    {
        BankAccountTable bankAccountTable;
        boolean canComplete = true;
        select RecId,workflowstate,reviewStatus from bankAccountTable where bankAccountTable.RecId == _context.parmRecId();
        if(bankAccountTable.RecId != 0)
        {
            switch(bankAccountTable.WorkflowState)
            {
```

100 %

Code for base class:

```
public class LITS_BankAccountWFBase
{
    public boolean validateContext(WorkFlowContext _context)
    {
        BankAccountTable bankAccountTable;
        if(_context.parmTableId() != tableNum(BankAccountTable))
        {
            throw Error('Work flow must be on Bank Account Table');
        }

        select RecId from bankAccountTable
            where bankAccountTable.RecId == _context.parmRecId();

        if(bankAccountTable.RecId == 0)
        {
            throw Error('Bank Account not found for workflow');
        }
        return true;
    }

    public boolean canCompleteWF(WorkFlowContext _context)
    {
        BankAccountTable bankAccountTable;
        boolean canComplete = true;
              select RecId,workflowstate,reviewStatus from bankAccountTable where
      bankAccountTable.RecId == _context.parmRecId();
        if(bankAccountTable.RecId != 0)
        {
            switch(bankAccountTable.WorkflowState)
            {
                case LITS_BankAccountWGApprovalStatus::Revised:
                case LITS_BankAccountWGApprovalStatus::Waiting:
                case LITS_BankAccountWGApprovalStatus::InReview:

                    canComplete = false;
                    break;
                default:
                    canComplete = true;

            }
            switch(bankAccountTable.ReviewStatus)
            {
                case LITS_BankAccountReviewStatus::PendingApproval:
                case LITS_BankAccountReviewStatus::Waiting:
                    canComplete = false;
                    break;
            }
        }
        return canComplete;
    }

}
```

- **Open the class "LITS_BankAccountWorkflowTypeEventHandler" which was added by the system and extends this class with your base class.**


LITS_BankAccountWorkflowTypeEventHandler
started(WorkflowEventArgs _workflowEventArgs)
100 %

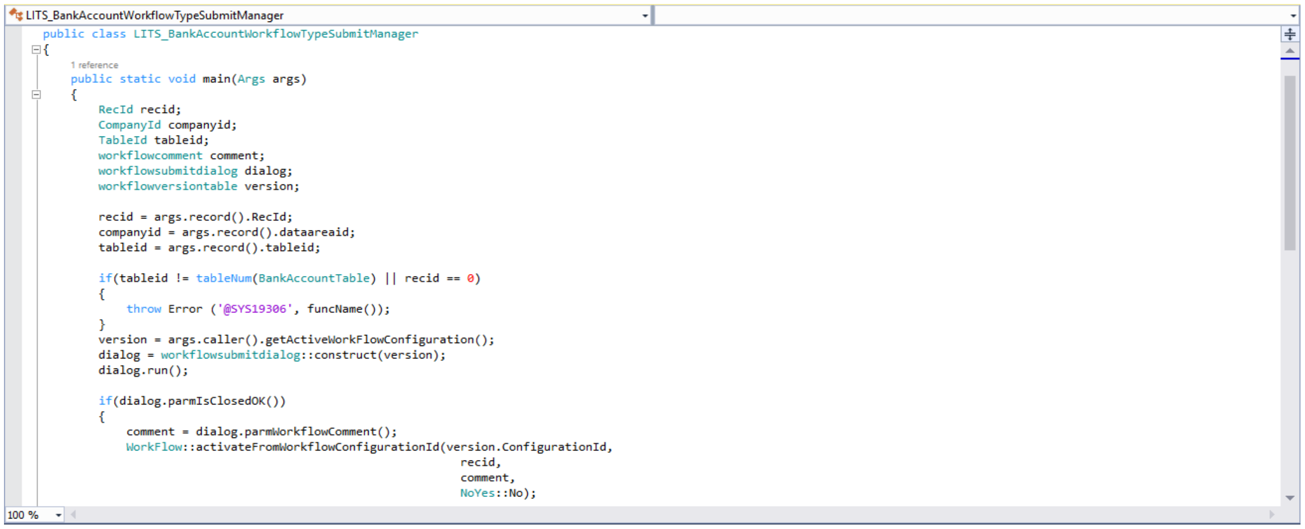
```
public class  LITS_BankAccountWorkflowTypeEventHandler extends LITS_BankAccountWFBase implements WorkflowCanceledEventHandler,
    WorkflowCompletedEventHandler,
    WorkflowStartedEventHandler
{
    1 reference
    public void started(WorkflowEventArgs _workflowEventArgs)
    {
        workflowcontext context;
        context = _workflowEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
            LITS_BankAccountWFStatusHandler::setState(context.parmRecId(), LITS_BankAccountWGApprovalStatus::Waiting);
        }
    }

    1 reference
    public void canceled(WorkflowEventArgs _workflowEventArgs)
    {
        workflowcontext context;
        context = _workflowEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
            LITS_BankAccountWFStatusHandler::setState(context.parmRecId(), LITS_BankAccountWGApprovalStatus::Draft);
        }
    }

    1 reference
    public void completed(WorkflowEventArgs _workflowEventArgs)
    {
        workflowcontext context;
        context = _workflowEventArgs.parmWorkflowContext();
        if(this.validateContext(context) && !this.canCompleteWF(context))
```

Code for LITS_BankAccountWorkflowTypeEventHandler class:

```
/// <summary>
/// The LITS_BankAccountWorkflowTypeEventHandler workflow event handler.
/// </summary>
```

```
public class  LITS_BankAccountWorkflowTypeEventHandler extends LITS_BankAccountWFBase
implements WorkflowCanceledEventHandler,
       WorkflowCompletedEventHandler,
       WorkflowStartedEventHandler
{
    public void started(WorkflowEventArgs _workflowEventArgs)
       {
        workflowcontext context;
        context = _workflowEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                   LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Waiting);
        }
       }

    public void canceled(WorkflowEventArgs _workflowEventArgs)
       {
        workflowcontext context;
        context = _workflowEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                   LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Draft);
        }
       }

    public void completed(WorkflowEventArgs _workflowEventArgs)
       {
        workflowcontext context;
        context = _workflowEventArgs.parmWorkflowContext();
        if(this.validateContext(context) && !this.canCompleteWF(context))
        {
                   LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Draft);
        }
       }

}
```

- **Open the class "LITS_BankAccountWorkflowTypeSubmitManager" which was added by the system and extends this class with your base class.**

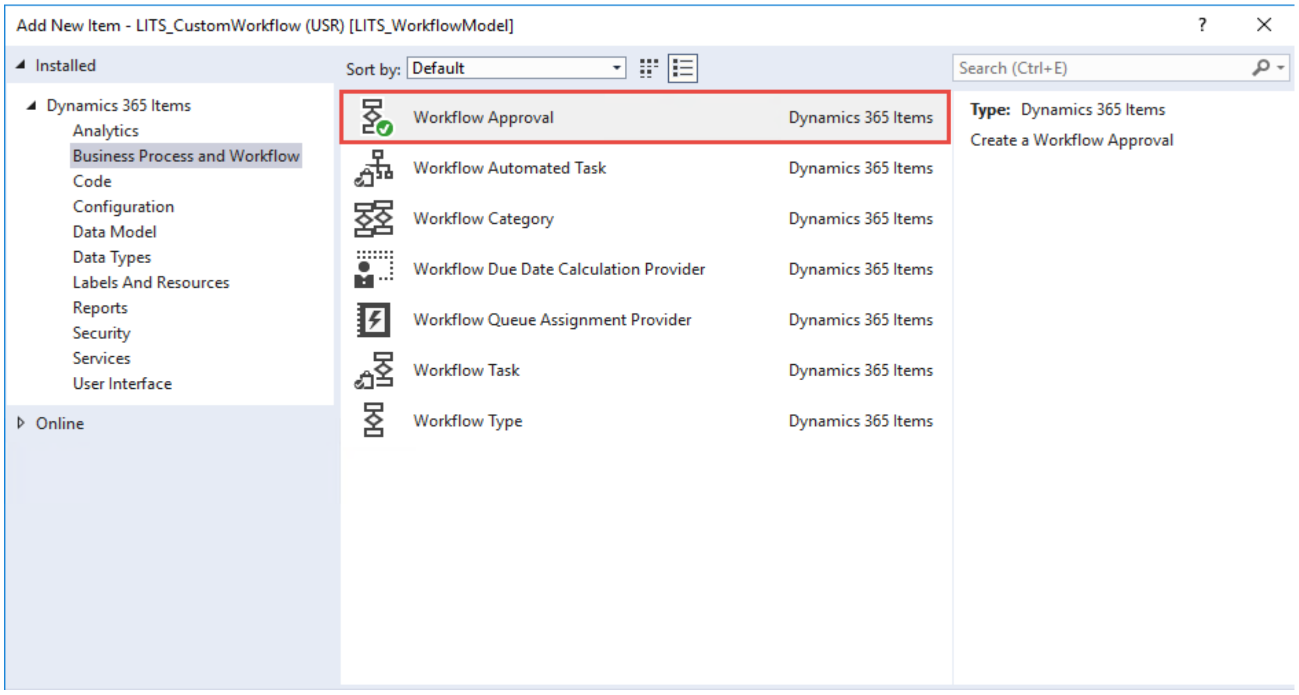


```
LITS_BankAccountWorkflowTypeSubmitManager
public class LITS_BankAccountWorkflowTypeSubmitManager
{
    1 reference
    public static void main(Args args)
    {
        RecId recid;
        CompanyId companyid;
        TableId tableid;
        workflowcomment comment;
        workflowsubmitdialog dialog;
        workflowversiontable version;

        recid = args.record().RecId;
        companyid = args.record().dataareaid;
        tableid = args.record().tableid;

        if(tableid != tableNum(BankAccountTable) || recid == 0)
        {
            throw Error ('@SYS19306', funcName());
        }
        version = args.caller().getActiveWorkFlowConfiguration();
        dialog = workflowsubmitdialog::construct(version);
        dialog.run();

        if(dialog.parmIsClosedOK())
        {
            comment = dialog.parmWorkflowComment();
            WorkFlow::activateFromWorkflowConfigurationId(version.ConfigurationId,
                                                          recid,
                                                          comment,
                                                          NoYes::No);
100 %
```

Code for LITS_BankAccountWorkflowTypeSubmitManager class:

```
/// <summary>
/// The LITS_BankAccountWorkflowTypeSubmitManager menu item action event handler.
/// </summary>
public class LITS_BankAccountWorkflowTypeSubmitManager
{
    public static void main(Args args)
       {
        RecId recid;
        CompanyId companyid;
        TableId tableid;
        workflowcomment comment;
        workflowsubmitdialog dialog;
        workflowversiontable version;

        recid = args.record().RecId;
        companyid = args.record().dataareaid;
        tableid = args.record().tableid;

        if(tableid != tableNum(BankAccountTable) || recid == 0)
        {
            throw Error ('@SYS19306', funcName());
        }
        version = args.caller().getActiveWorkFlowConfiguration();
        dialog = workflowsubmitdialog::construct(version);
        dialog.run();

        if(dialog.parmIsClosedOK())
        {
            comment = dialog.parmWorkflowComment();
            WorkFlow::activateFromWorkflowConfigurationId(version.ConfigurationId,
                                                        recid,
                                                        comment,
                                                        NoYes::No);
       LITS_BankAccountWFStatusHandler::setState(args.record().RecId,LITS_BankAccountWGAp
       provalStatus::Waiting);
        }
        if(FormDataUtil::isFormDataSource(args.record()))
        {
            FormDataUtil::getFormDataSource(args.record()).research(true);
        }
        args.caller().updateWorkFlowControls();
       }

}
```

- **Add workflow approval**



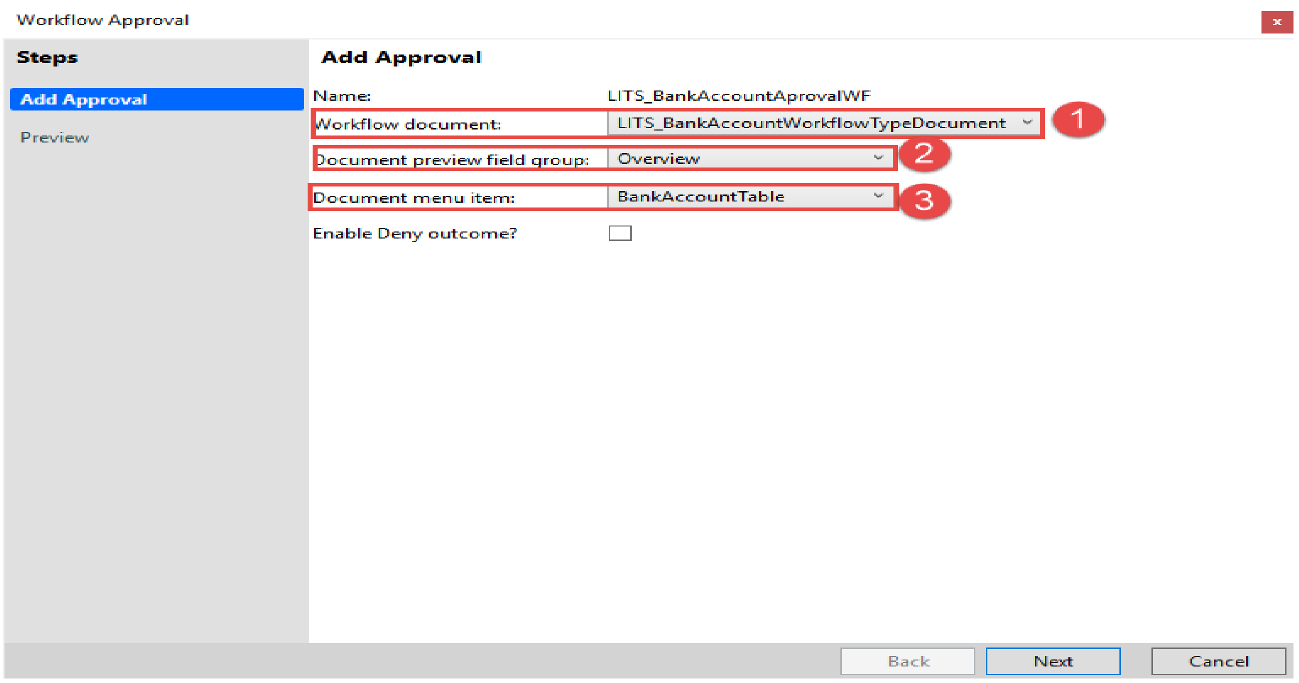
Add New Item - LITS_CustomWorkflow (USR) [LITS_WorkflowModel]

Installed
- Dynamics 365 Items
  - Analytics
  - Business Process and Workflow
  - Code
  - Configuration
  - Data Model
  - Data Types
  - Labels And Resources
  - Reports
  - Security
  - Services
  - User Interface
- Online

Sort by: Default

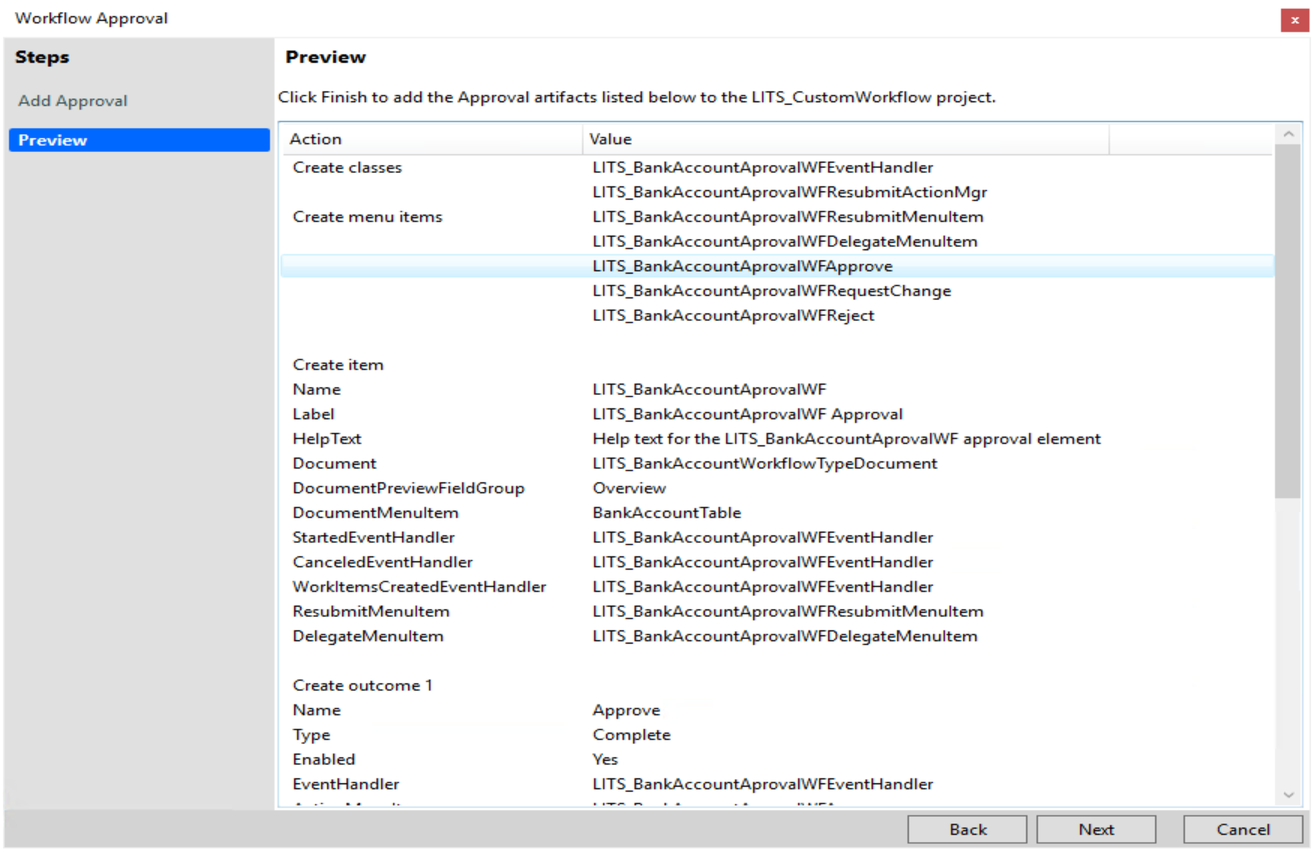
Workflow Approval	Dynamics 365 Items
Workflow Automated Task	Dynamics 365 Items
Workflow Category	Dynamics 365 Items
Workflow Due Date Calculation Provider	Dynamics 365 Items
Workflow Queue Assignment Provider	Dynamics 365 Items
Workflow Task	Dynamics 365 Items
Workflow Type	Dynamics 365 Items

Type: Dynamics 365 Items
Create a Workflow Approval



Name:	LITS_BankAccountAprovalWF
Workflow document:	LITS_BankAccountWorkflowTypeDocument
Document preview field group:	Overview
Document menu item:	BankAccountTable
Enable Deny outcome?	☐

1. Select workflow document class which was added by system.
2. Select Overview field group as workflowstate field was added in this field group.
3. Select form’s menuitem.

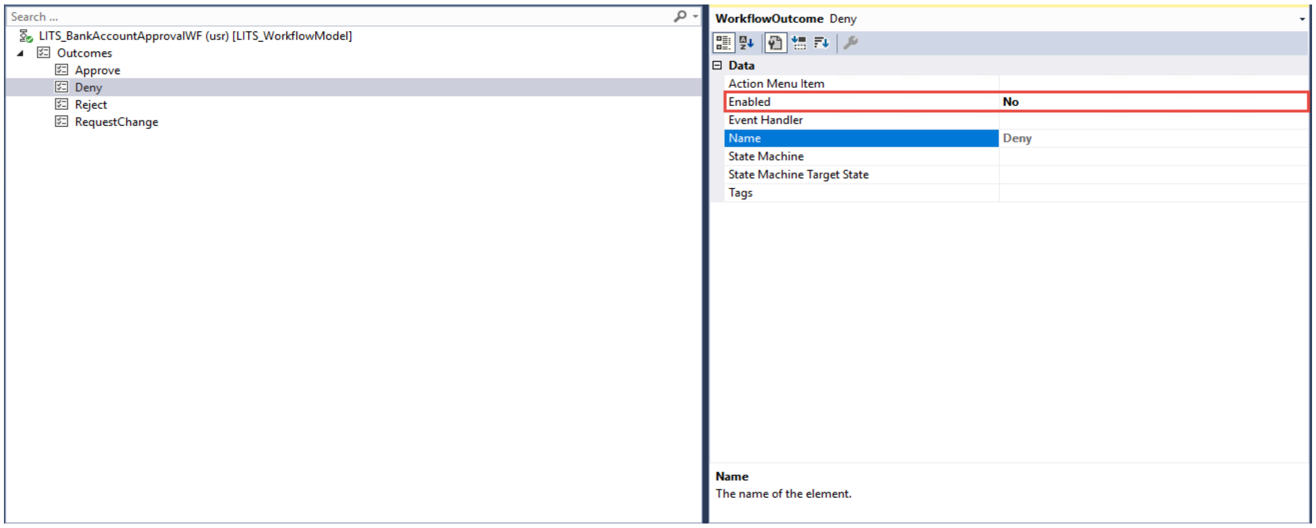


Click Finish to add the Approval artifacts listed below to the LITS_CustomWorkflow project.

Action	Value
Create classes	LITS_BankAccountAprovalWFEventHandler
	LITS_BankAccountAprovalWFResubmitActionMgr
Create menu items	LITS_BankAccountAprovalWFResubmitMenuItem
	LITS_BankAccountAprovalWFDelegateMenuItem
	LITS_BankAccountAprovalWFApprove
	LITS_BankAccountAprovalWFRequestChange
	LITS_BankAccountAprovalWFReject
Create item	
Name	LITS_BankAccountAprovalWF
Label	LITS_BankAccountAprovalWF Approval
HelpText	Help text for the LITS_BankAccountAprovalWF approval element
Document	LITS_BankAccountWorkflowTypeDocument
DocumentPreviewFieldGroup	Overview
DocumentMenuItem	BankAccountTable
StartedEventHandler	LITS_BankAccountAprovalWFEventHandler
CanceledEventHandler	LITS_BankAccountAprovalWFEventHandler
WorkItemsCreatedEventHandler	LITS_BankAccountAprovalWFEventHandler
ResubmitMenuItem	LITS_BankAccountAprovalWFResubmitMenuItem
DelegateMenuItem	LITS_BankAccountAprovalWFDelegateMenuItem
Create outcome 1	
Name	Approve
Type	Complete
Enabled	Yes
EventHandler	LITS_BankAccountAprovalWFEventHandler

All these elements will be added in your project automatically.

- **Open workflow approval**



Search ...

LITS_BankAccountApprovalWF (usr) [LITS_WorkflowModel]
- Outcomes
  - Approve
  - Deny
  - Reject
  - RequestChange

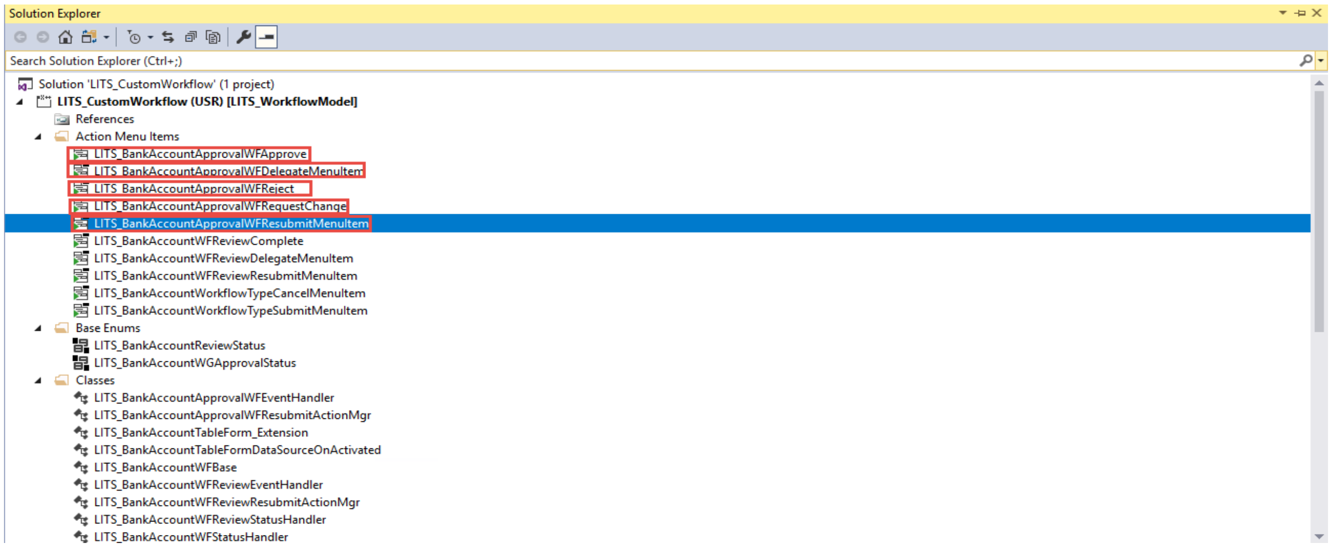
WorkflowOutcome Deny

Data

Action Menu Item	
Enabled	No
Event Handler	
Name	Deny
State Machine	
State Machine Target State	
Tags	

Name
The name of the element.

- **Rename label for these action menu items to "Approve", "Delegate", "Reject", "Request Change", "Resubmit".**

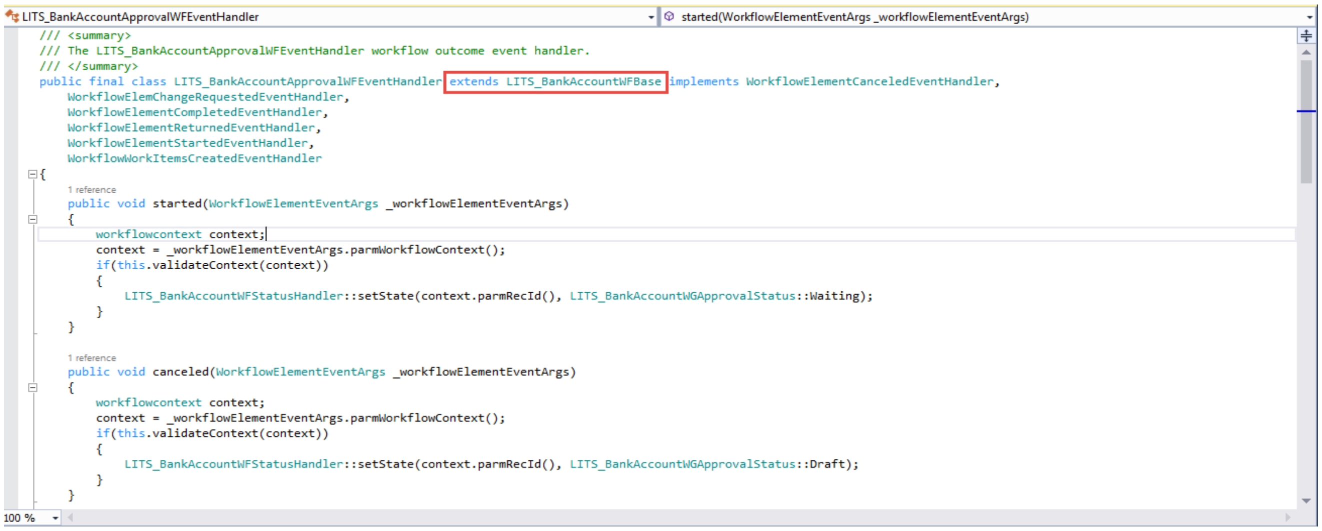


Solution Explorer

Search Solution Explorer (Ctrl+;)

```
Solution 'LITS_CustomWorkflow' (1 project)
LITS_CustomWorkflow (USR) [LITS_WorkflowModel]
  References
  Action Menu Items
    LITS_BankAccountApprovalWFApprove
    LITS_BankAccountApprovalWFDelegateMenuItem
    LITS_BankAccountApprovalWFReject
    LITS_BankAccountApprovalWFRequestChange
    LITS_BankAccountApprovalWFResubmitMenuItem
    LITS_BankAccountWFReviewComplete
    LITS_BankAccountWFReviewDelegateMenuItem
    LITS_BankAccountWFReviewResubmitMenuItem
    LITS_BankAccountWorkflowTypeCancelMenuItem
    LITS_BankAccountWorkflowTypeSubmitMenuItem
  Base Enums
    LITS_BankAccountReviewStatus
    LITS_BankAccountWGApprovalStatus
  Classes
    LITS_BankAccountApprovalWFEventHandler
    LITS_BankAccountApprovalWFResubmitActionMgr
    LITS_BankAccountTableForm_Extension
    LITS_BankAccountTableFormDataSourceOnActivated
    LITS_BankAccountWFBase
    LITS_BankAccountWFReviewEventHandler
    LITS_BankAccountWFReviewResubmitActionMgr
    LITS_BankAccountWFReviewStatusHandler
    LITS_BankAccountWFStatusHandler
```

- **Open "LITS_BankAccountApprovalWFEventHandler" class and extends it to base class**

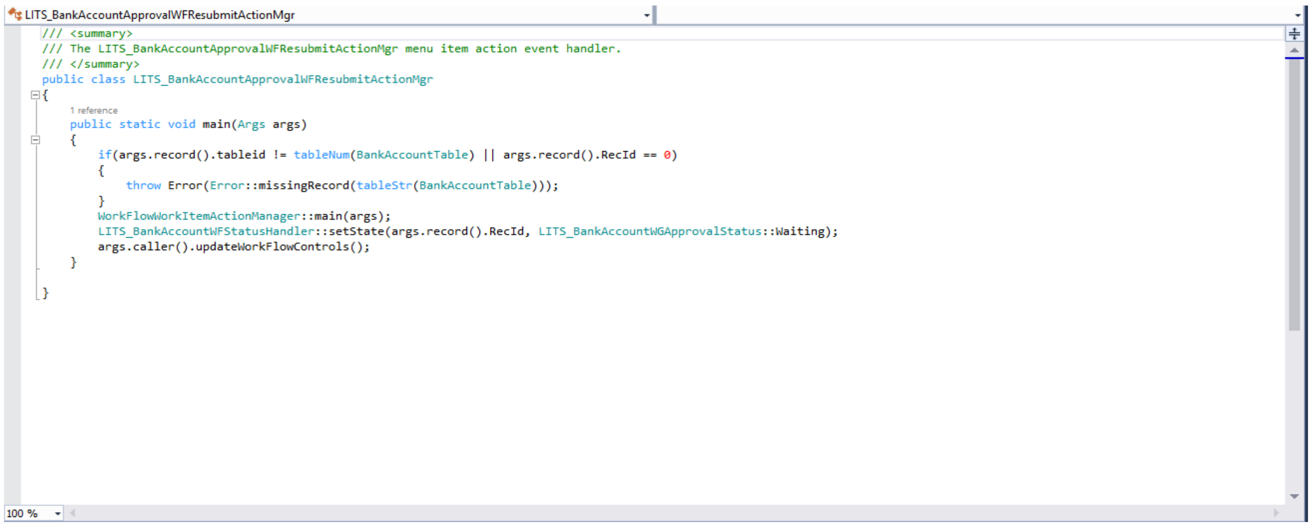

LITS_BankAccountApprovalWFEventHandler
started(WorkflowElementEventArgs _workflowElementEventArgs)
100 %

```
/// <summary>
/// The LITS_BankAccountApprovalWFEventHandler workflow outcome event handler.
/// </summary>
public final class LITS_BankAccountApprovalWFEventHandler extends LITS_BankAccountWFBase implements WorkflowElementCanceledEventHandler,
    WorkflowElemChangeRequestedEventHandler,
    WorkflowElementCompletedEventHandler,
    WorkflowElementReturnedEventHandler,
    WorkflowElementStartedEventHandler,
    WorkflowWorkItemsCreatedEventHandler
{
    1 reference
    public void started(WorkflowElementEventArgs _workflowElementEventArgs)
    {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
            LITS_BankAccountWFStatusHandler::setState(context.parmRecId(), LITS_BankAccountWGApprovalStatus::Waiting);
        }
    }

    1 reference
    public void canceled(WorkflowElementEventArgs _workflowElementEventArgs)
    {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
            LITS_BankAccountWFStatusHandler::setState(context.parmRecId(), LITS_BankAccountWGApprovalStatus::Draft);
        }
    }
```

Code for class "LITS_BankAccountApprovalWFEventHandler":

```
/// <summary>
/// The LITS_BankAccountApprovalWFEventHandler workflow outcome event handler.
/// </summary>
public final class LITS_BankAccountApprovalWFEventHandler extends LITS_BankAccountWFBase
implements WorkflowElementCanceledEventHandler,
       WorkflowElemChangeRequestedEventHandler,
       WorkflowElementCompletedEventHandler,
       WorkflowElementReturnedEventHandler,
       WorkflowElementStartedEventHandler,
       WorkflowWorkItemsCreatedEventHandler
{
    public void started(WorkflowElementEventArgs _workflowElementEventArgs)
       {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                   LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Waiting);
        }
       }

    public void canceled(WorkflowElementEventArgs _workflowElementEventArgs)
       {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                   LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Draft);
        }
       }
    public void completed(WorkflowElementEventArgs _workflowElementEventArgs)
       {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                    LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Approved);
        }
       }

    public void changeRequested(WorkflowElementEventArgs _workflowElementEventArgs)
       {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                    LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Revised);
        }
       }

    public void returned(WorkflowElementEventArgs _workflowElementEventArgs)
       {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                    LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::Rejected);
        }
       }

    public void created(WorkflowWorkItemsEventArgs _workflowWorkItemsEventArgs)
       {
        WorkflowElementEventArgs workFlowArgs =
       _workflowWorkItemsEventArgs.parmWorkFlowElementEventArgs();
        workflowcontext context;
        context = workFlowArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
                    LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
       LITS_BankAccountWGApprovalStatus::InReview);
        }
       }

}
```

- **Open the Approval Resubmit manager class**


LITS_BankAccountApprovalWFResubmitActionMgr

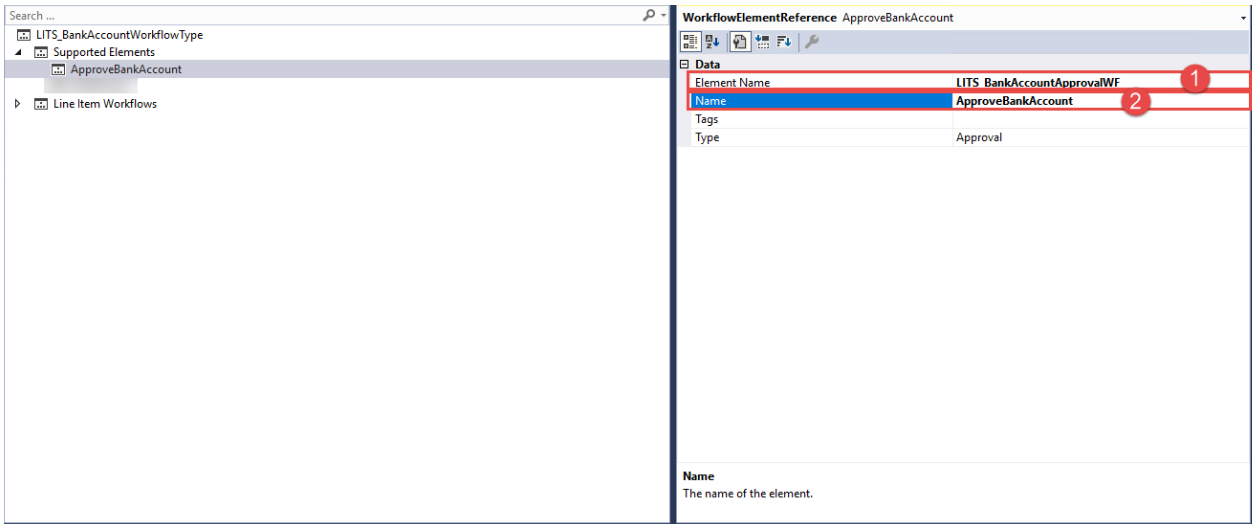
```
/// <summary>
/// The LITS_BankAccountApprovalWFResubmitActionMgr menu item action event handler.
/// </summary>
public class LITS_BankAccountApprovalWFResubmitActionMgr
{
    1 reference
    public static void main(Args args)
    {
        if(args.record().tableid != tableNum(BankAccountTable) || args.record().RecId == 0)
        {
            throw Error(Error::missingRecord(tableStr(BankAccountTable)));
        }
        WorkFlowWorkItemActionManager::main(args);
        LITS_BankAccountWFStatusHandler::setState(args.record().RecId, LITS_BankAccountWGApprovalStatus::Waiting);
        args.caller().updateWorkFlowControls();
    }

}
```

100 %

Code for "LITS_BankAccountApprovalWFResubmitActionMgr" class

```
/// <summary>
/// The LITS_BankAccountApprovalWFResubmitActionMgr menu item action event handler.
/// </summary>
public class LITS_BankAccountApprovalWFResubmitActionMgr
{
    public static void main(Args args)
       {
              if(args.record().tableid != tableNum(BankAccountTable) ||
       args.record().RecId == 0)
        {
            throw Error(Error::missingRecord(tableStr(BankAccountTable)));
        }
        WorkFlowWorkItemActionManager::main(args);
               LITS_BankAccountWFStatusHandler::setState(args.record().RecId,
       LITS_BankAccountWGApprovalStatus::Waiting);
        args.caller().updateWorkFlowControls();
       }

}
```

- **Open workflow type and add new workflow element reference in supported elements node.**


Search ...

LITS_BankAccountWorkflowType
Supported Elements
ApproveBankAccount
Line Item Workflows

WorkflowElementReference ApproveBankAccount

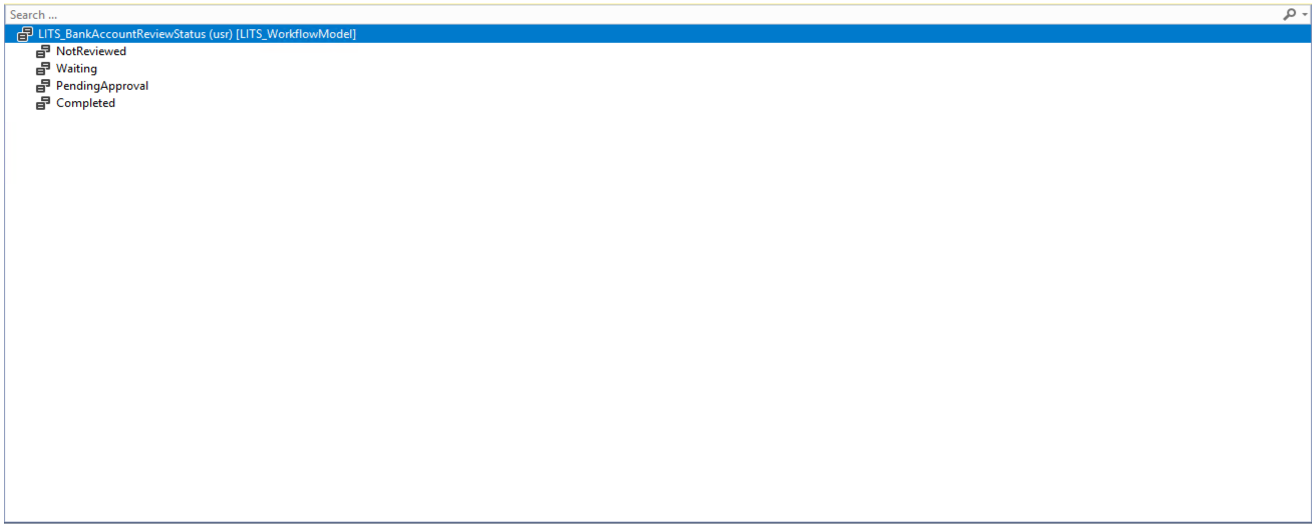
Data

Element Name	LITS_BankAccountApprovalWF
Name	ApproveBankAccount
Tags	
Type	Approval

Name
The name of the element.

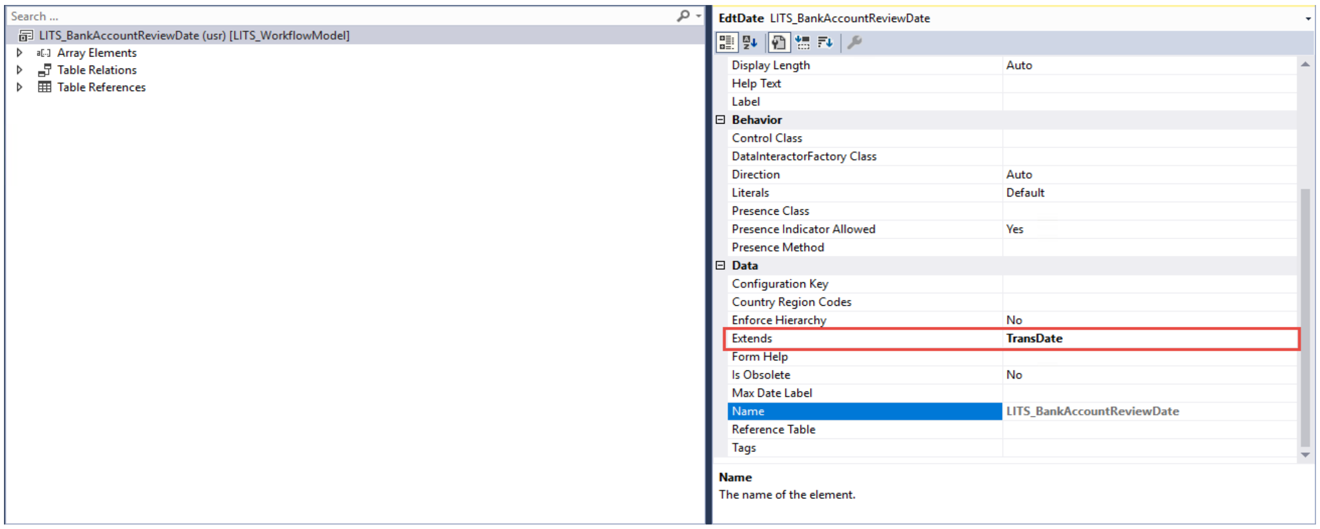
1. Element name should be your workflow approval name.
2. Rename element reference.

- **Create another base enum and set the label for each element.**


Search ...

LITS_BankAccountReviewStatus (usr) [LITS_WorkflowModel]
NotReviewed
Waiting
PendingApproval
Completed

- **Create an EDT Date and extends it with TransDate.**



Search ...

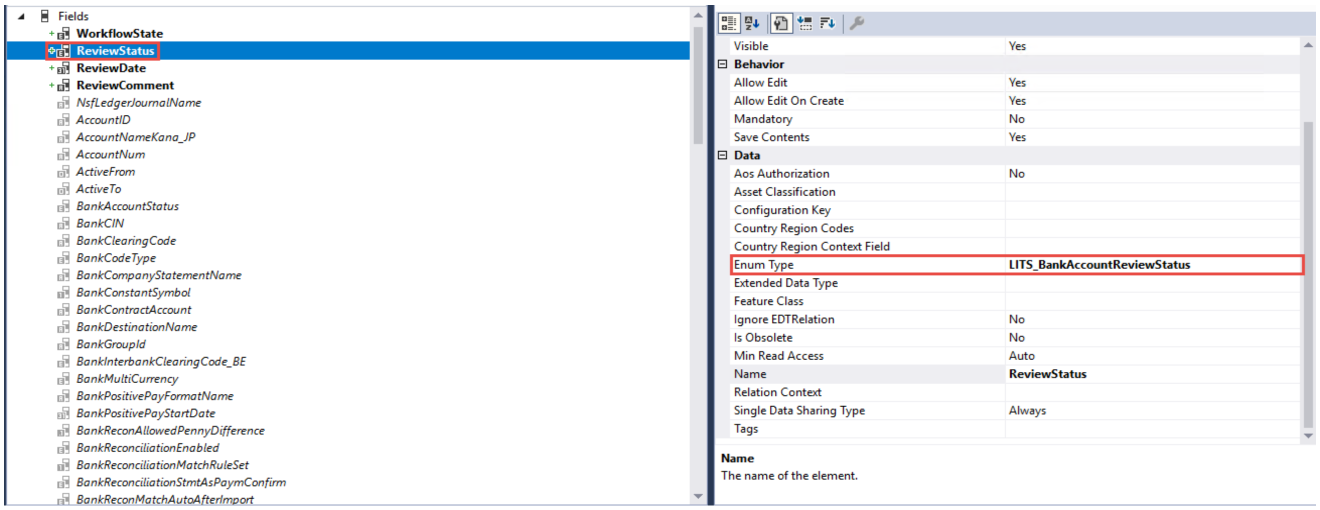
- LITS_BankAccountReviewDate (usr) [LITS_WorkflowModel]
  - Array Elements
  - Table Relations
  - Table References

EdtDate LITS_BankAccountReviewDate

Display Length	Auto
Help Text	
Label	
Behavior	
Control Class	
DataInteractorFactory Class	
Direction	Auto
Literals	Default
Presence Class	
Presence Indicator Allowed	Yes
Presence Method	
Data	
Configuration Key	
Country Region Codes	
Enforce Hierarchy	No
Extends	**TransDate**
Form Help	
Is Obsolete	No
Max Date Label	
Name	LITS_BankAccountReviewDate
Reference Table	
Tags	

Name
The name of the element.

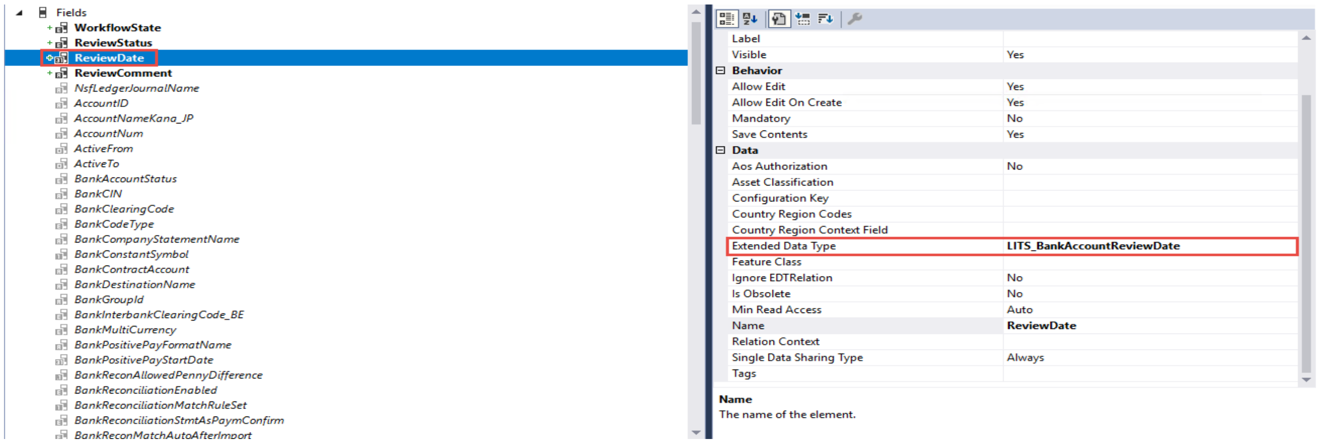
- **Add three more fields in Bank Account Table and set the properties.**



- Fields
  - **WorkflowState**
  - **ReviewStatus**
  - **ReviewDate**
  - **ReviewComment**
  - *NsfLedgerJournalName*
  - *AccountID*
  - *AccountNameKana_JP*
  - *AccountNum*
  - *ActiveFrom*
  - *ActiveTo*
  - *BankAccountStatus*
  - *BankCIN*
  - *BankClearingCode*
  - *BankCodeType*
  - *BankCompanyStatementName*
  - *BankConstantSymbol*
  - *BankContractAccount*
  - *BankDestinationName*
  - *BankGroupId*
  - *BankInterbankClearingCode_BE*
  - *BankMultiCurrency*
  - *BankPositivePayFormatName*
  - *BankPositivePayStartDate*
  - *BankReconAllowedPennyDifference*
  - *BankReconciliationEnabled*
  - *BankReconciliationMatchRuleSet*
  - *BankReconciliationStmtAsPaymConfirm*
  - *BankReconMatchAutoAfterImport*

Visible	Yes
Behavior	
Allow Edit	Yes
Allow Edit On Create	Yes
Mandatory	No
Save Contents	Yes
Data	
Aos Authorization	No
Asset Classification	
Configuration Key	
Country Region Codes	
Country Region Context Field	
Enum Type	**LITS_BankAccountReviewStatus**
Extended Data Type	
Feature Class	
Ignore EDTRelation	No
Is Obsolete	No
Min Read Access	Auto
Name	**ReviewStatus**
Relation Context	
Single Data Sharing Type	Always
Tags	

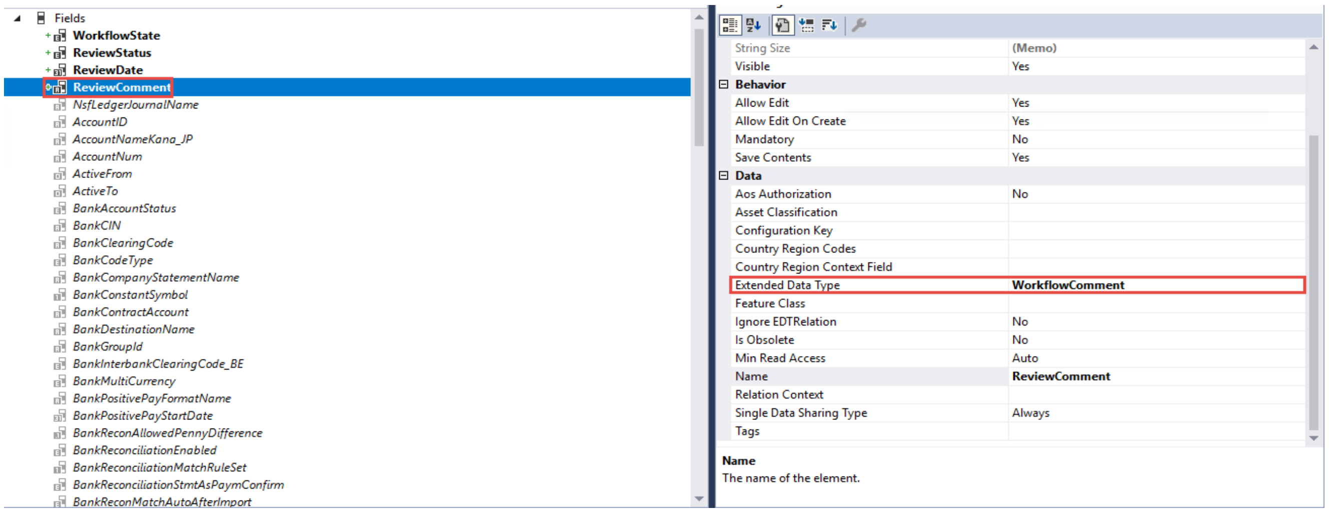
Name
The name of the element.



- Fields
  - **WorkflowState**
  - **ReviewStatus**
  - **ReviewDate**
  - **ReviewComment**
  - *NsfLedgerJournalName*
  - *AccountID*
  - *AccountNameKana_JP*
  - *AccountNum*
  - *ActiveFrom*
  - *ActiveTo*
  - *BankAccountStatus*
  - *BankCIN*
  - *BankClearingCode*
  - *BankCodeType*
  - *BankCompanyStatementName*
  - *BankConstantSymbol*
  - *BankContractAccount*
  - *BankDestinationName*
  - *BankGroupId*
  - *BankInterbankClearingCode_BE*
  - *BankMultiCurrency*
  - *BankPositivePayFormatName*
  - *BankPositivePayStartDate*
  - *BankReconAllowedPennyDifference*
  - *BankReconciliationEnabled*
  - *BankReconciliationMatchRuleSet*
  - *BankReconciliationStmtAsPaymConfirm*
  - *BankReconMatchAutoAfterImport*

Label	
Visible	Yes
Behavior	
Allow Edit	Yes
Allow Edit On Create	Yes
Mandatory	No
Save Contents	Yes
Data	
Aos Authorization	No
Asset Classification	
Configuration Key	
Country Region Codes	
Country Region Context Field	
Extended Data Type	**LITS_BankAccountReviewDate**
Feature Class	
Ignore EDTRelation	No
Is Obsolete	No
Min Read Access	Auto
Name	**ReviewDate**
Relation Context	
Single Data Sharing Type	Always
Tags	

Name
The name of the element.

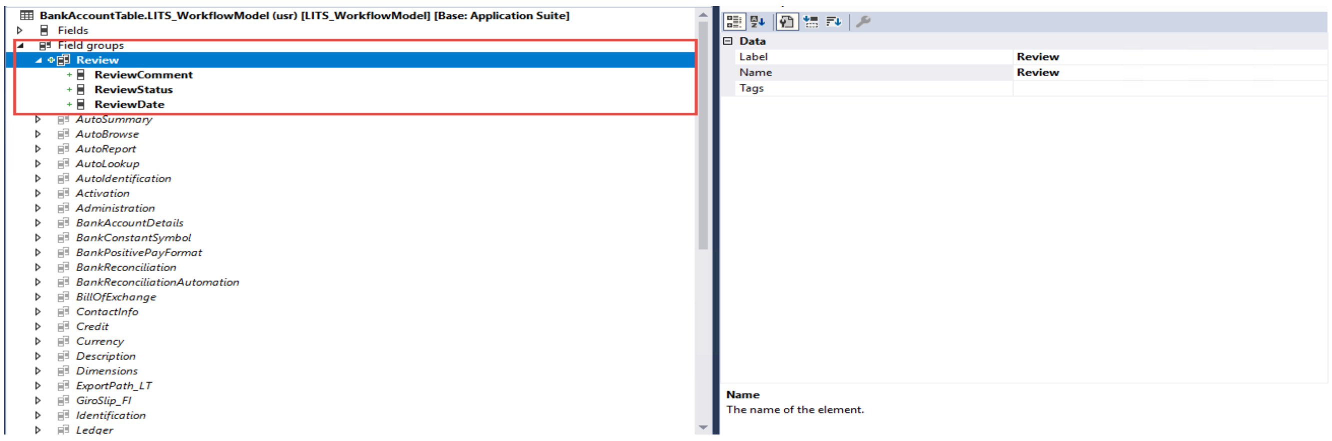


```
◢ Fields
   WorkflowState
   ReviewStatus
   ReviewDate
   ReviewComment
   NsfLedgerJournalName
   AccountID
   AccountNameKana_JP
   AccountNum
   ActiveFrom
   ActiveTo
   BankAccountStatus
   BankCIN
   BankClearingCode
   BankCodeType
   BankCompanyStatementName
   BankConstantSymbol
   BankContractAccount
   BankDestinationName
   BankGroupId
   BankInterbankClearingCode_BE
   BankMultiCurrency
   BankPositivePayFormatName
   BankPositivePayStartDate
   BankReconAllowedPennyDifference
   BankReconciliationEnabled
   BankReconciliationMatchRuleSet
   BankReconciliationStmtAsPaymConfirm
   BankReconMatchAutoAfterImport
```

String Size	(Memo)
Visible	Yes
Behavior	
Allow Edit	Yes
Allow Edit On Create	Yes
Mandatory	No
Save Contents	Yes
Data	
Aos Authorization	No
Asset Classification	
Configuration Key	
Country Region Codes	
Country Region Context Field	
Extended Data Type	**WorkflowComment**
Feature Class	
Ignore EDTRelation	No
Is Obsolete	No
Min Read Access	Auto
Name	**ReviewComment**
Relation Context	
Single Data Sharing Type	Always
Tags	

Name
The name of the element.

- **Create new field group and add these fields in this group.**

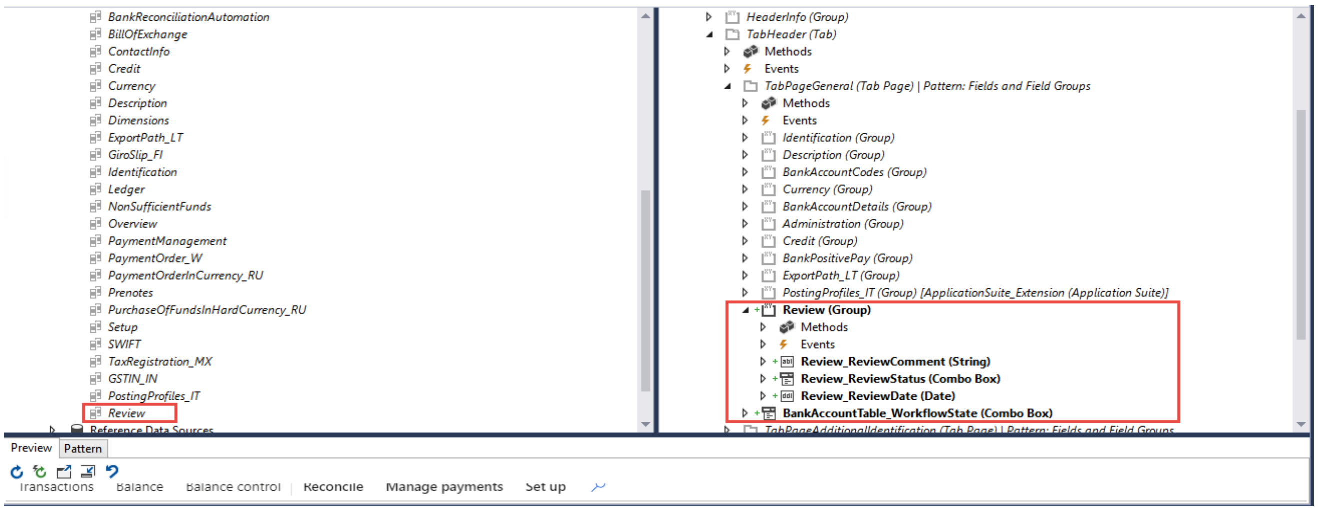


```
BankAccountTable.LITS_WorkflowModel (usr) [LITS_WorkflowModel] [Base: Application Suite]
 ▷ Fields
 ◢ Field groups
   ◢ Review
        ReviewComment
        ReviewStatus
        ReviewDate
   ▷ AutoSummary
   ▷ AutoBrowse
   ▷ AutoReport
   ▷ AutoLookup
   ▷ AutoIdentification
   ▷ Activation
   ▷ Administration
   ▷ BankAccountDetails
   ▷ BankConstantSymbol
   ▷ BankPositivePayFormat
   ▷ BankReconciliation
   ▷ BankReconciliationAutomation
   ▷ BillOfExchange
   ▷ ContactInfo
   ▷ Credit
   ▷ Currency
   ▷ Description
   ▷ Dimensions
   ▷ ExportPath_LT
   ▷ GiroSlip_FI
   ▷ Identification
   ▷ Ledger
```

Data	
Label	**Review**
Name	**Review**
Tags	

Name
The name of the element.

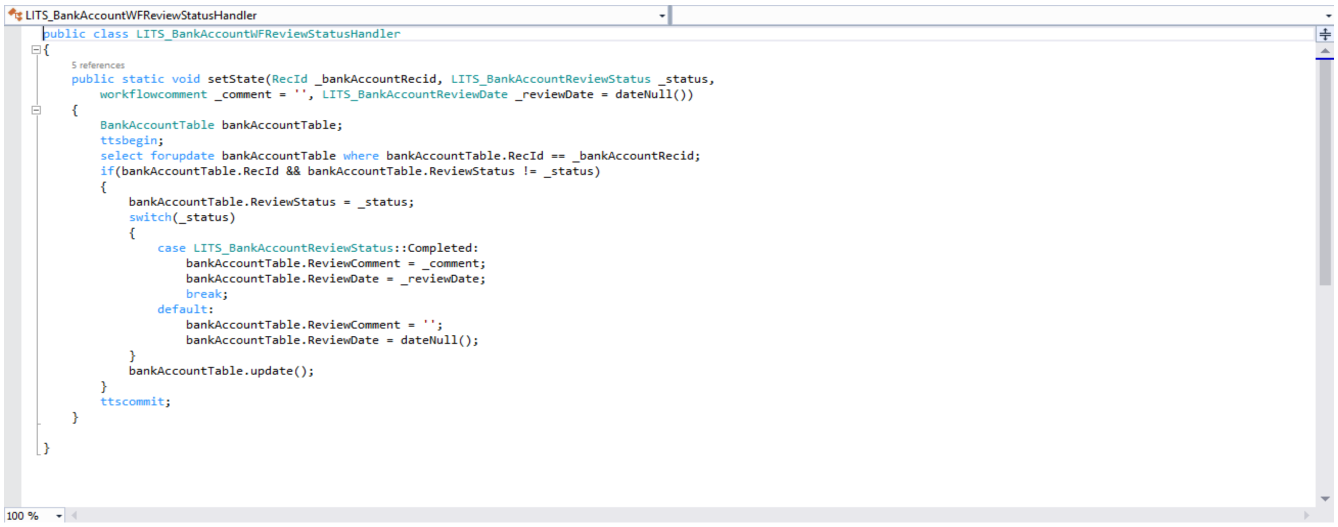
- **Create extension of BankAccountTable form and drag drop this field group.**



```
BankReconciliationAutomation
BillOfExchange
ContactInfo
Credit
Currency
Description
Dimensions
ExportPath_LT
GiroSlip_FI
Identification
Ledger
NonSufficientFunds
Overview
PaymentManagement
PaymentOrder_W
PaymentOrderInCurrency_RU
Prenotes
PurchaseOfFundsInHardCurrency_RU
Setup
SWIFT
TaxRegistration_MX
GSTIN_IN
PostingProfiles_IT
Review
Reference Data Sources
```

```
HeaderInfo (Group)
TabHeader (Tab)
  Methods
  Events
  TabPageGeneral (Tab Page) | Pattern: Fields and Field Groups
    Methods
    Events
    Identification (Group)
    Description (Group)
    BankAccountCodes (Group)
    Currency (Group)
    BankAccountDetails (Group)
    Administration (Group)
    Credit (Group)
    BankPositivePay (Group)
    ExportPath_LT (Group)
    PostingProfiles_IT (Group) [ApplicationSuite_Extension (Application Suite)]
    Review (Group)
      Methods
      Events
      Review_ReviewComment (String)
      Review_ReviewStatus (Combo Box)
      Review_ReviewDate (Date)
    BankAccountTable_WorkflowState (Combo Box)
  TabPageAdditionalIdentification (Tab Page) | Pattern: Fields and Field Groups
```

- **Create a new class for review status handler**



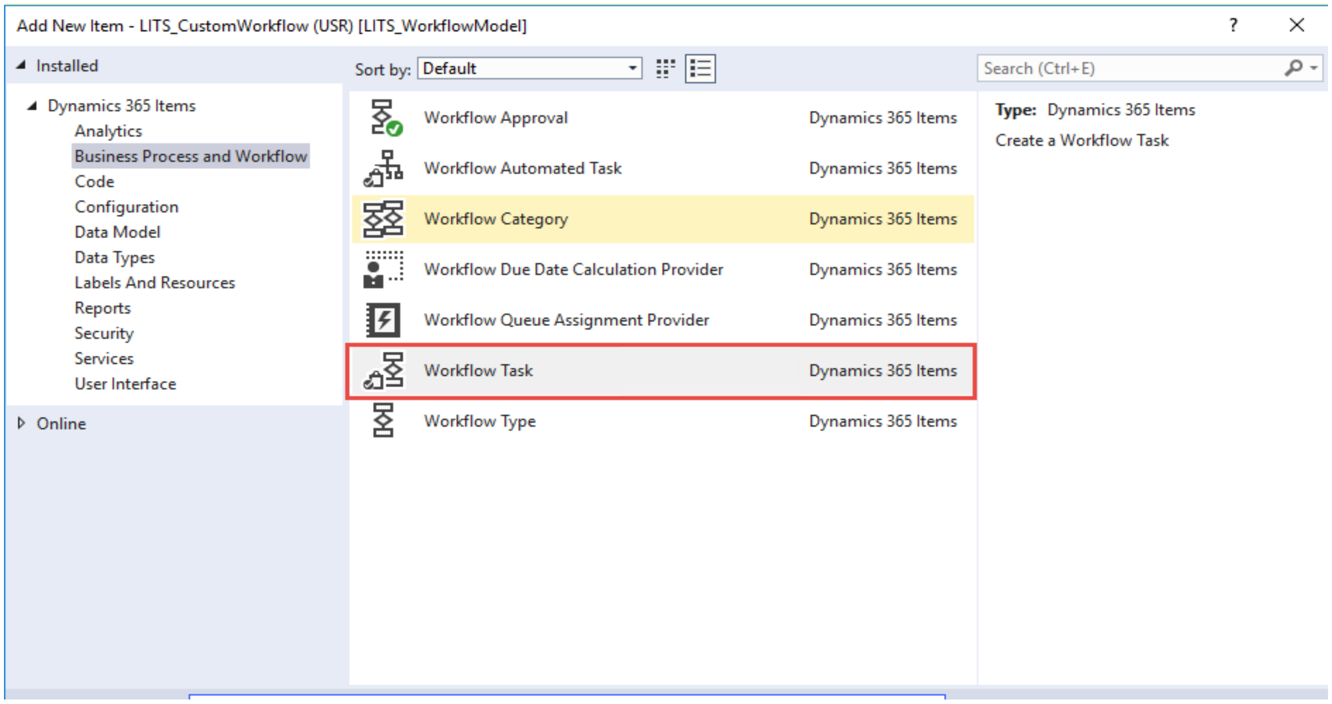
```
public class LITS_BankAccountWFReviewStatusHandler
{
    public static void setState(RecId _bankAccountRecid, LITS_BankAccountReviewStatus _status,
        workflowcomment _comment = '', LITS_BankAccountReviewDate _reviewDate = dateNull())
    {
        BankAccountTable bankAccountTable;
        ttsbegin;
        select forupdate bankAccountTable where bankAccountTable.RecId == _bankAccountRecid;
        if(bankAccountTable.RecId && bankAccountTable.ReviewStatus != _status)
        {
            bankAccountTable.ReviewStatus = _status;
            switch(_status)
            {
                case LITS_BankAccountReviewStatus::Completed:
                    bankAccountTable.ReviewComment = _comment;
                    bankAccountTable.ReviewDate = _reviewDate;
                    break;
                default:
                    bankAccountTable.ReviewComment = '';
                    bankAccountTable.ReviewDate = dateNull();
            }
            bankAccountTable.update();
        }
        ttscommit;
    }
}
```

Code for review status handler class:

```
public class LITS_BankAccountWFReviewStatusHandler
{
    public static void setState(RecId _bankAccountRecid, LITS_BankAccountReviewStatus
    _status,
        workflowcomment _comment = '', LITS_BankAccountReviewDate _reviewDate =
    dateNull())
    {
        BankAccountTable bankAccountTable;
        ttsbegin;
               select forupdate bankAccountTable where bankAccountTable.RecId ==
      _bankAccountRecid;
        if(bankAccountTable.RecId && bankAccountTable.ReviewStatus != _status)
```

```
            {
                bankAccountTable.ReviewStatus = _status;
                switch(_status)
                {
                    case LITS_BankAccountReviewStatus::Completed:
                        bankAccountTable.ReviewComment = _comment;
                        bankAccountTable.ReviewDate = _reviewDate;
                        break;
                    default:
                        bankAccountTable.ReviewComment = '';
                        bankAccountTable.ReviewDate = dateNull();
                }
                bankAccountTable.update();
            }
            ttscommit;
        }

    }
```

- **Create a Workflow task**



Add New Item - LITS_CustomWorkflow (USR) [LITS_WorkflowModel]

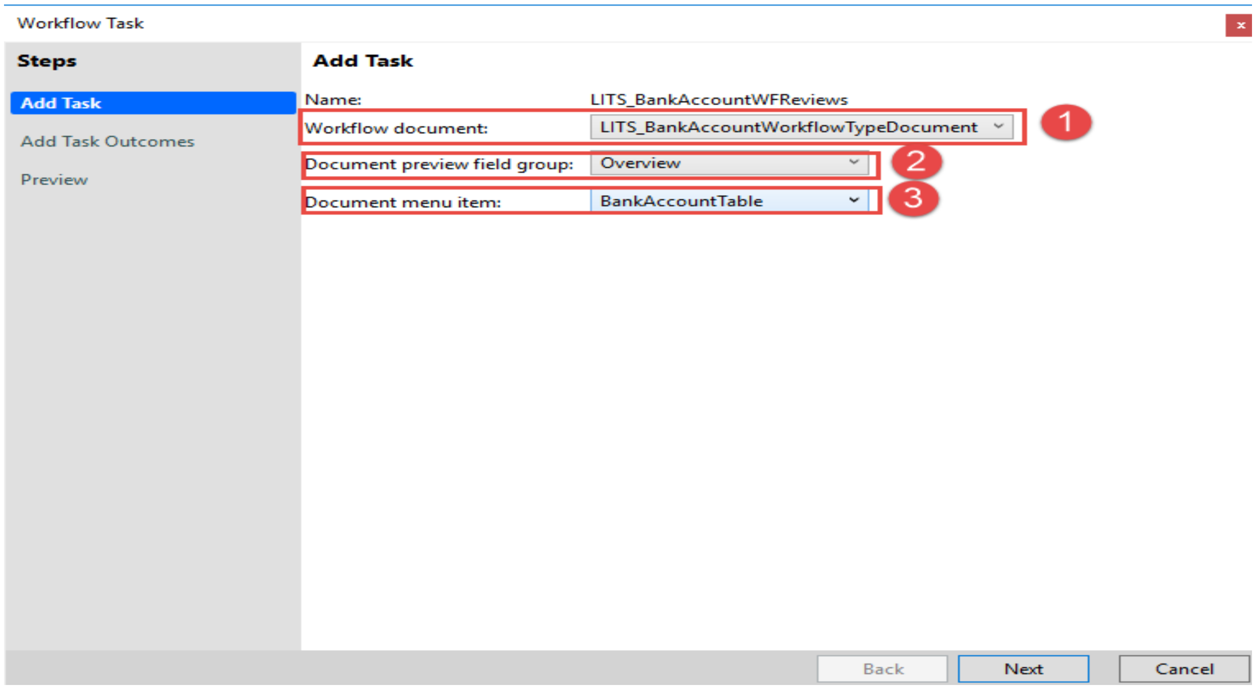
Sort by: Default

Search (Ctrl+E)

Installed
- Dynamics 365 Items
  - Analytics
  - Business Process and Workflow
  - Code
  - Configuration
  - Data Model
  - Data Types
  - Labels And Resources
  - Reports
  - Security
  - Services
  - User Interface
- Online

Workflow Approval	Dynamics 365 Items
Workflow Automated Task	Dynamics 365 Items
Workflow Category	Dynamics 365 Items
Workflow Due Date Calculation Provider	Dynamics 365 Items
Workflow Queue Assignment Provider	Dynamics 365 Items
Workflow Task	Dynamics 365 Items
Workflow Type	Dynamics 365 Items

Type: Dynamics 365 Items
Create a Workflow Task


Workflow Task

Steps

Add Task

Add Task Outcomes

Preview

Add Task

Name: LITS_BankAccountWFReviews

Workflow document: LITS_BankAccountWorkflowTypeDocument (1)

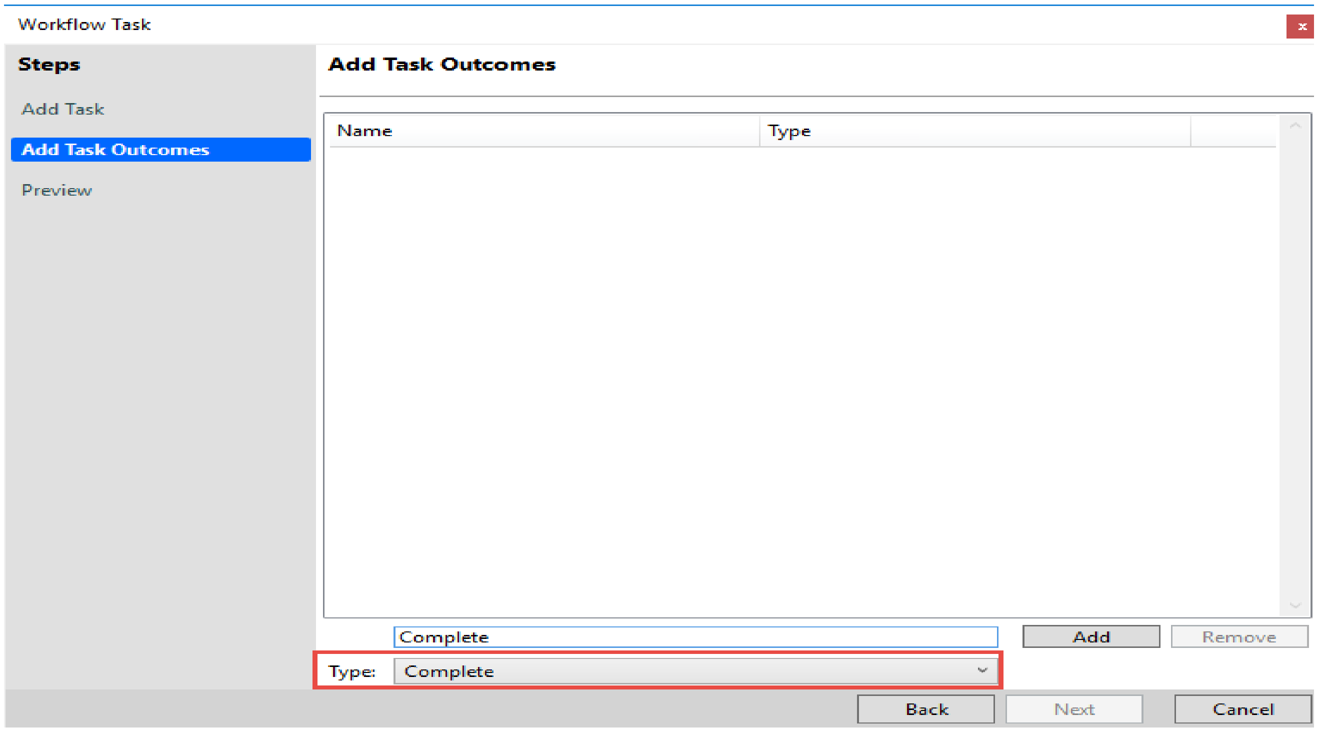
Document preview field group: Overview (2)

Document menu item: BankAccountTable (3)

Back | Next | Cancel

1. Select Document class.
2. Select Field group.
3. Select form’s menuitem.

Select Type Complete, Click on Add and then click on Next button.


Workflow Task

Steps

Add Task

Add Task Outcomes

Preview

Add Task Outcomes

Name	Type

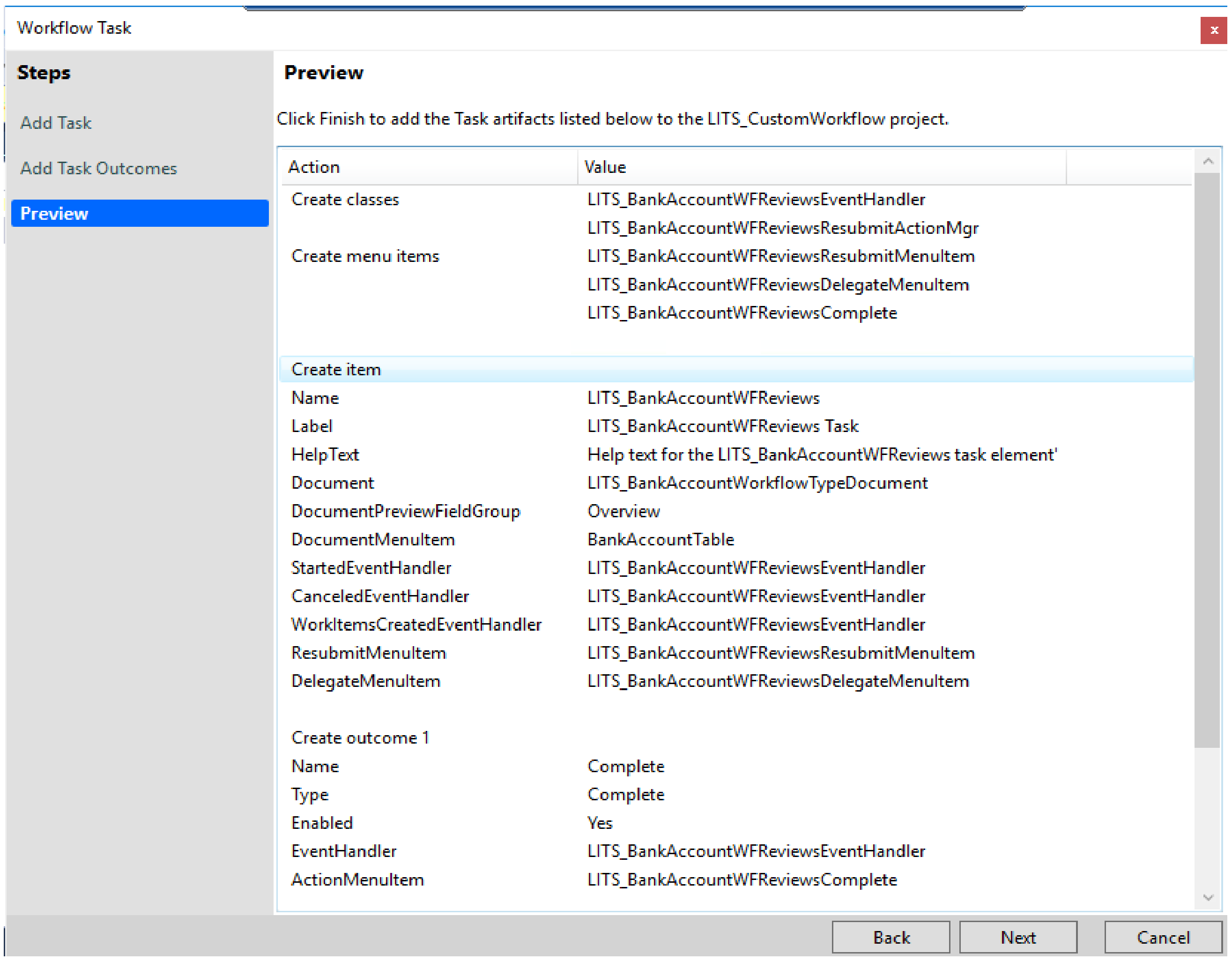
Complete

Add | Remove

Type: Complete

Back | Next | Cancel

All these elements will be added by system automatically.


Workflow Task

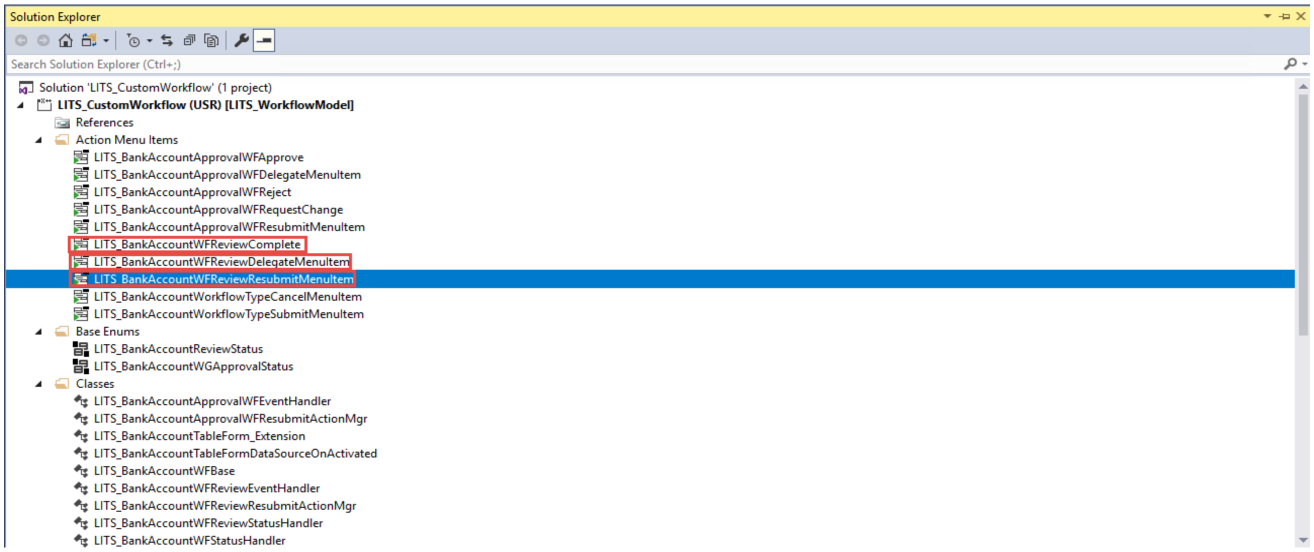
Steps
Add Task
Add Task Outcomes
Preview

Preview

Click Finish to add the Task artifacts listed below to the LITS_CustomWorkflow project.

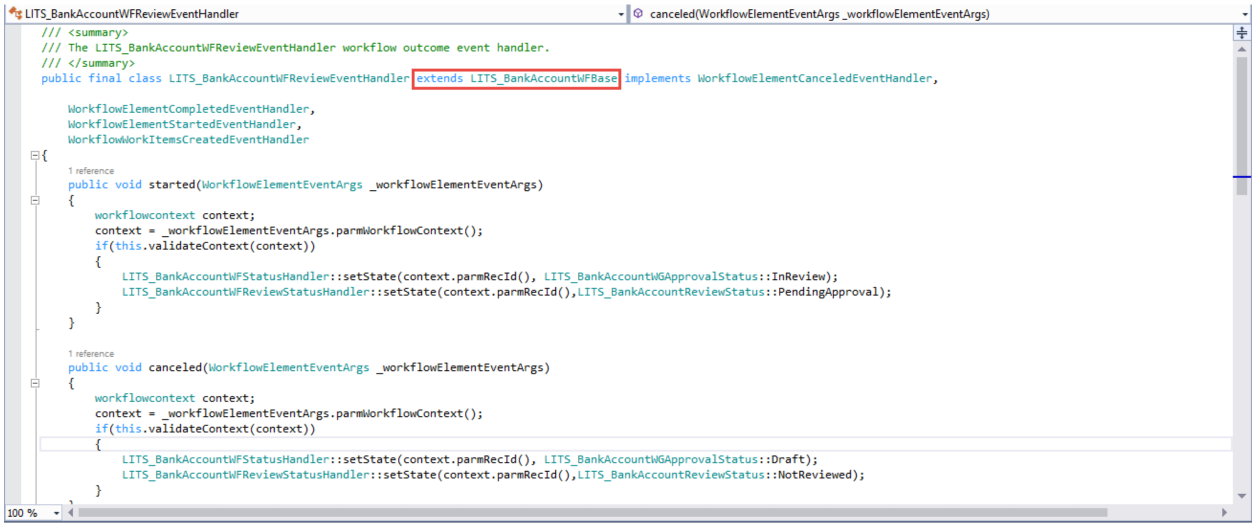
Action	Value
Create classes	LITS_BankAccountWFReviewsEventHandler
	LITS_BankAccountWFReviewsResubmitActionMgr
Create menu items	LITS_BankAccountWFReviewsResubmitMenuItem
	LITS_BankAccountWFReviewsDelegateMenuItem
	LITS_BankAccountWFReviewsComplete
Create item	
Name	LITS_BankAccountWFReviews
Label	LITS_BankAccountWFReviews Task
HelpText	Help text for the LITS_BankAccountWFReviews task element'
Document	LITS_BankAccountWorkflowTypeDocument
DocumentPreviewFieldGroup	Overview
DocumentMenuItem	BankAccountTable
StartedEventHandler	LITS_BankAccountWFReviewsEventHandler
CanceledEventHandler	LITS_BankAccountWFReviewsEventHandler
WorkItemsCreatedEventHandler	LITS_BankAccountWFReviewsEventHandler
ResubmitMenuItem	LITS_BankAccountWFReviewsResubmitMenuItem
DelegateMenuItem	LITS_BankAccountWFReviewsDelegateMenuItem
Create outcome 1	
Name	Complete
Type	Complete
Enabled	Yes
EventHandler	LITS_BankAccountWFReviewsEventHandler
ActionMenuItem	LITS_BankAccountWFReviewsComplete

- **Add label for these action menus to Complete, Delegate and ReSubmit.**



```
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'LITS_CustomWorkflow' (1 project)
  LITS_CustomWorkflow (USR) [LITS_WorkflowModel]
    References
    Action Menu Items
      LITS_BankAccountApprovalWFApprove
      LITS_BankAccountApprovalWFDelegateMenuItem
      LITS_BankAccountApprovalWFReject
      LITS_BankAccountApprovalWFRequestChange
      LITS_BankAccountApprovalWFResubmitMenuItem
      LITS_BankAccountWFReviewComplete
      LITS_BankAccountWFReviewDelegateMenuItem
      LITS_BankAccountWFReviewResubmitMenuItem
      LITS_BankAccountWorkflowTypeCancelMenuItem
      LITS_BankAccountWorkflowTypeSubmitMenuItem
    Base Enums
      LITS_BankAccountReviewStatus
      LITS_BankAccountWGApprovalStatus
    Classes
      LITS_BankAccountApprovalWFEventHandler
      LITS_BankAccountApprovalWFResubmitActionMgr
      LITS_BankAccountTableForm_Extension
      LITS_BankAccountTableFormDataSourceOnActivated
      LITS_BankAccountWFBase
      LITS_BankAccountWFReviewEventHandler
      LITS_BankAccountWFReviewResubmitActionMgr
      LITS_BankAccountWFReviewStatusHandler
      LITS_BankAccountWFStatusHandler
```

- **Open Review event handler class and extends it to base class.**



```
LITS_BankAccountWFReviewEventHandler          canceled(WorkflowElementEventArgs _workflowElementEventArgs)
/// <summary>
/// The LITS_BankAccountWFReviewEventHandler workflow outcome event handler.
/// </summary>
public final class LITS_BankAccountWFReviewEventHandler extends LITS_BankAccountWFBase implements WorkflowElementCanceledEventHandler,

    WorkflowElementCompletedEventHandler,
    WorkflowElementStartedEventHandler,
    WorkflowWorkItemsCreatedEventHandler
{
    1 reference
    public void started(WorkflowElementEventArgs _workflowElementEventArgs)
    {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
            LITS_BankAccountWFStatusHandler::setState(context.parmRecId(), LITS_BankAccountWGApprovalStatus::InReview);
            LITS_BankAccountWFReviewStatusHandler::setState(context.parmRecId(),LITS_BankAccountReviewStatus::PendingApproval);
        }
    }

    1 reference
    public void canceled(WorkflowElementEventArgs _workflowElementEventArgs)
    {
        workflowcontext context;
        context = _workflowElementEventArgs.parmWorkflowContext();
        if(this.validateContext(context))
        {
            LITS_BankAccountWFStatusHandler::setState(context.parmRecId(), LITS_BankAccountWGApprovalStatus::Draft);
            LITS_BankAccountWFReviewStatusHandler::setState(context.parmRecId(),LITS_BankAccountReviewStatus::NotReviewed);
        }
100 %
```

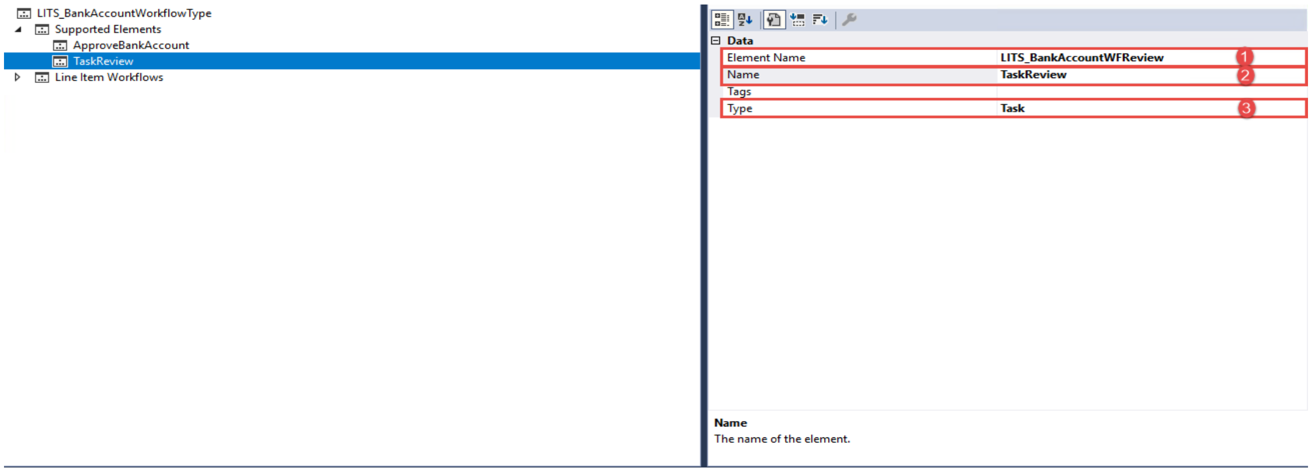
Code for Review Event Handler class:

```
/// <summary>
/// The LITS_BankAccountWFReviewEventHandler workflow outcome event handler.
/// </summary>
public final class LITS_BankAccountWFReviewEventHandler extends LITS_BankAccountWFBase
implements WorkflowElementCanceledEventHandler,

       WorkflowElementCompletedEventHandler,
       WorkflowElementStartedEventHandler,
       WorkflowWorkItemsCreatedEventHandler
{
```

```
public void started(WorkflowElementEventArgs _workflowElementEventArgs)
   {
    workflowcontext context;
    context = _workflowElementEventArgs.parmWorkflowContext();
    if(this.validateContext(context))
    {
               LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
   LITS_BankAccountWGApprovalStatus::InReview);

   LITS_BankAccountWFReviewStatusHandler::setState(context.parmRecId(),LITS_BankAccou
   ntReviewStatus::PendingApproval);
    }
   }

public void canceled(WorkflowElementEventArgs _workflowElementEventArgs)
   {
    workflowcontext context;
    context = _workflowElementEventArgs.parmWorkflowContext();
    if(this.validateContext(context))
    {
               LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
   LITS_BankAccountWGApprovalStatus::Draft);

   LITS_BankAccountWFReviewStatusHandler::setState(context.parmRecId(),LITS_BankAccou
   ntReviewStatus::NotReviewed);
    }
   }

public void completed(WorkflowElementEventArgs _workflowElementEventArgs)
   {
    workflowcontext context;
    context = _workflowElementEventArgs.parmWorkflowContext();
    workflowcorrelationid correlationId = context.parmWorkflowCorrelationId();
          workflowtrackingtable trackingTable =
   Workflow::findLastWorkflowTrackingRecord(correlationId);
    WorkflowTrackingCommentTable commentTable = trackingTable.commentTable();
    WorkflowComment comment = commentTable.Comment;
    Timezone timeZone = DateTimeUtil::getUserPreferredTimeZone();
    if(this.validateContext(context))
    {

   LITS_BankAccountWFReviewStatusHandler::setState(context.parmRecId(),LITS_BankAccou
   ntReviewStatus::Completed,comment,DateTimeUtil::getSystemDate(timeZone));
    }
   }

public void created(WorkflowWorkItemsEventArgs _workflowWorkItemsEventArgs)
   {
    WorkflowElementEventArgs workFlowArgs =
   _workflowWorkItemsEventArgs.parmWorkflowElementEventArgs();

    workflowcontext context;
    context = workFlowArgs.parmWorkflowContext();
    if(this.validateContext(context))
    {
               LITS_BankAccountWFStatusHandler::setState(context.parmRecId(),
   LITS_BankAccountWGApprovalStatus::InReview);
```

```
        LITS_BankAccountWFReviewStatusHandler::setState(context.parmRecId(),LITS_BankAccou
        ntReviewStatus::PendingApproval);
         }
        }

}
```

- **Again, open workflow type and add another new workflow element reference in Supported node.**



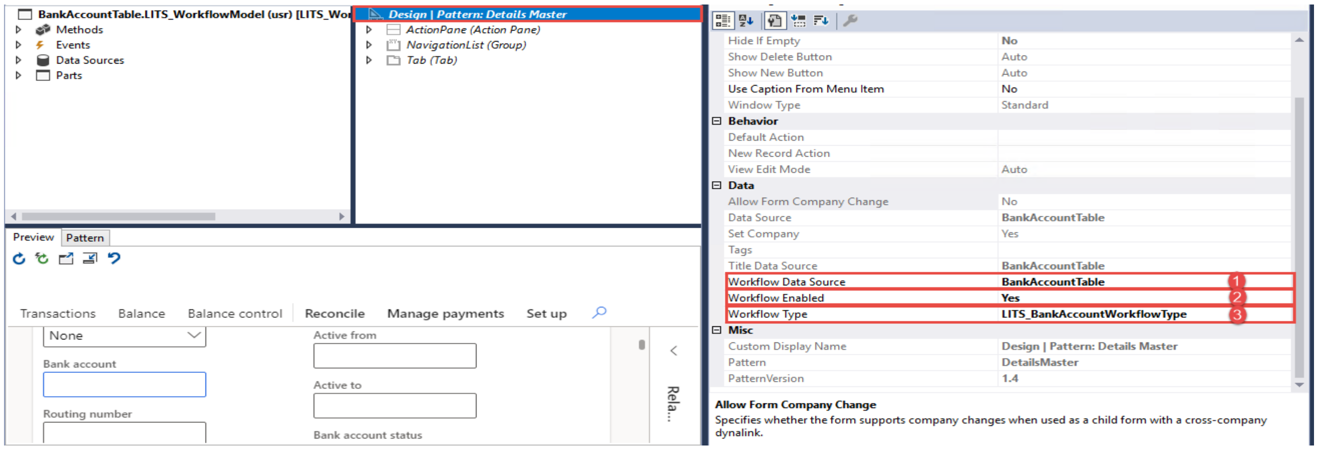
LITS_BankAccountWorkflowType
- Supported Elements
  - ApproveBankAccount
  - TaskReview
- Line Item Workflows

Data	
Element Name	LITS_BankAccountWFReview (1)
Name	TaskReview (2)
Tags	
Type	Task (3)

Name
The name of the element.

1. Element name should be your workflow task name.
2. Rename element reference.
3. Select Task.

- **Open BankAccountTable form extension**



BankAccountTable.LITS_WorkflowModel (usr) [LITS_Wo
- Methods
- Events
- Data Sources
- Parts

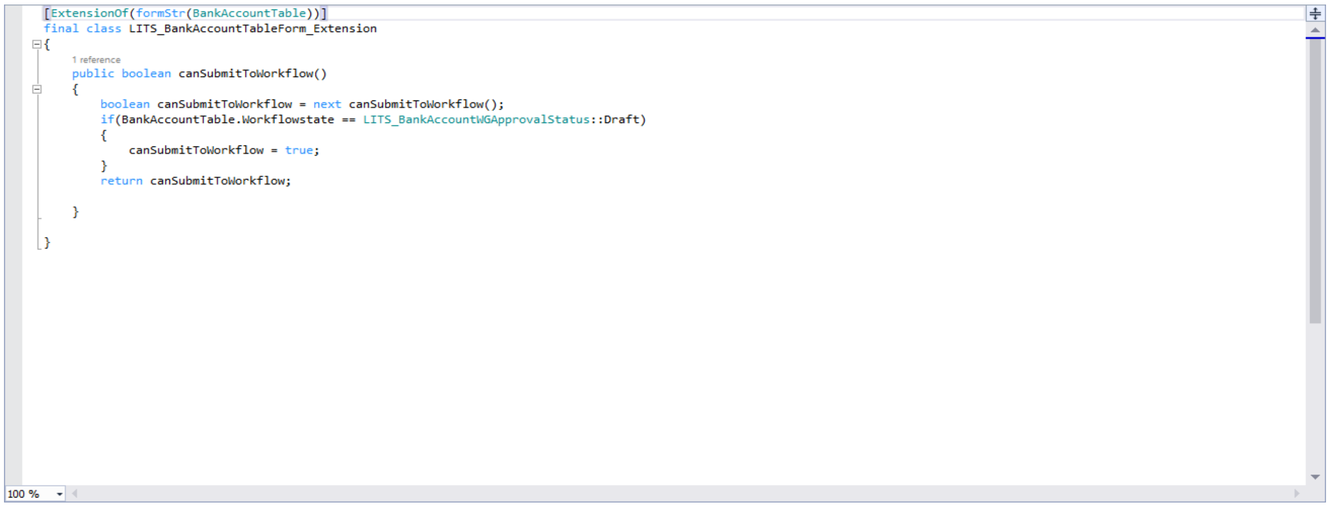
Design | Pattern: Details Master
- ActionPane (Action Pane)
- NavigationList (Group)
- Tab (Tab)

Hide If Empty	No
Show Delete Button	Auto
Show New Button	Auto
Use Caption From Menu Item	No
Window Type	Standard
Behavior	
Default Action	
New Record Action	
View Edit Mode	Auto
Data	
Allow Form Company Change	No
Data Source	BankAccountTable
Set Company	Yes
Tags	
Title Data Source	BankAccountTable
Workflow Data Source	BankAccountTable (1)
Workflow Enabled	Yes (2)
Workflow Type	LITS_BankAccountWorkflowType (3)
Misc	
Custom Display Name	Design | Pattern: Details Master
Pattern	DetailsMaster
PatternVersion	1.4

Allow Form Company Change
Specifies whether the form supports company changes when used as a child form with a cross-company dynalink.

1. Select datasource.
2. Set workflow enabled to Yes.
3. Select Workflow type you have created in your project.

- **Create an extension class for BankAccountTable form to add method.**



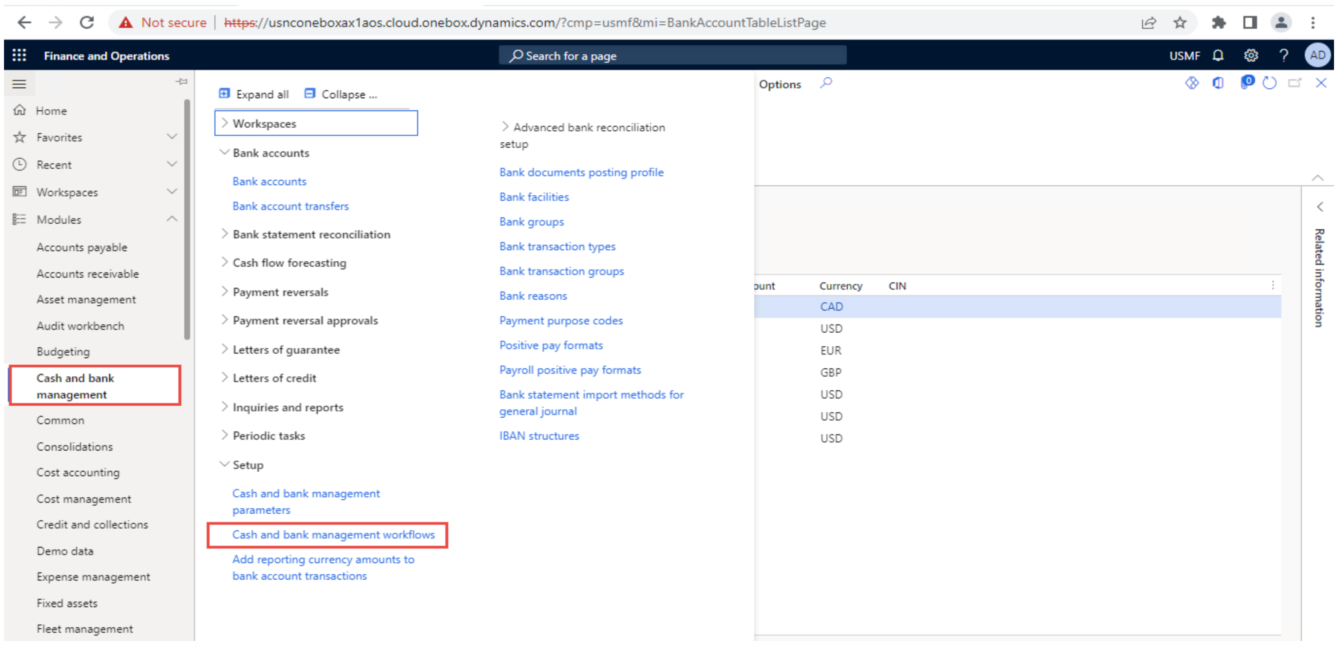
```
[ExtensionOf(formStr(BankAccountTable))]
final class LITS_BankAccountTableForm_Extension
{
    1 reference
    public boolean canSubmitToWorkflow()
    {
        boolean canSubmitToWorkflow = next canSubmitToWorkflow();
        if(BankAccountTable.Workflowstate == LITS_BankAccountWGApprovalStatus::Draft)
        {
            canSubmitToWorkflow = true;
        }
        return canSubmitToWorkflow;

    }

}
```

100 %

Code for Extension class:

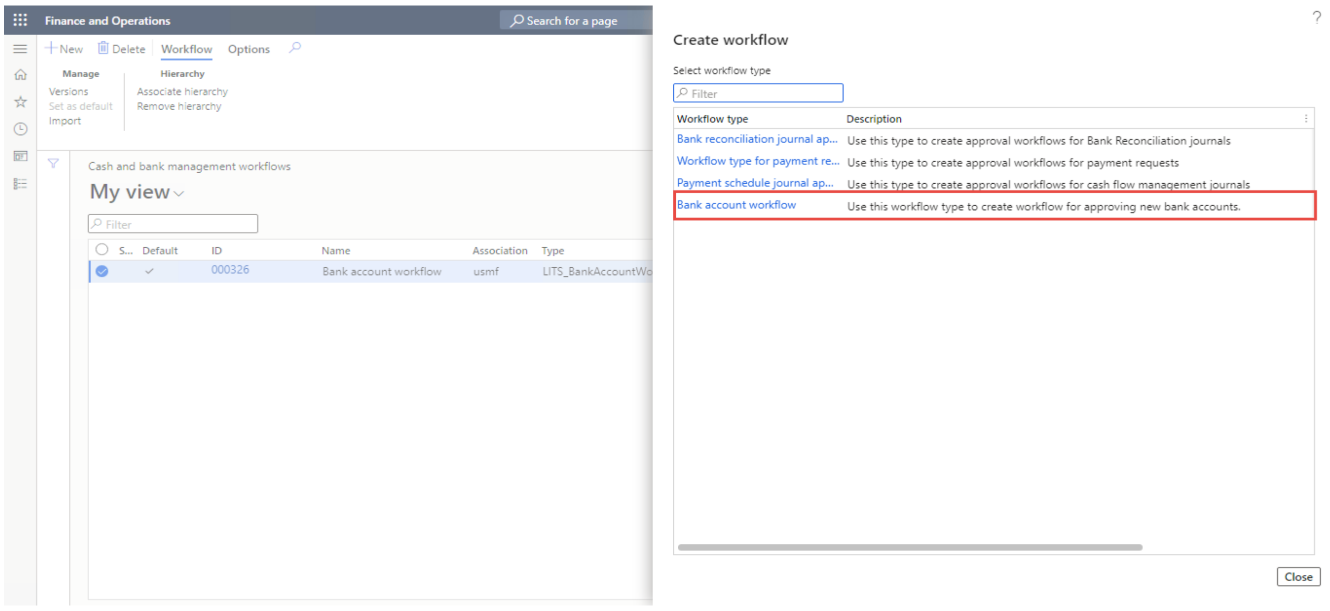
```
[ExtensionOf(formStr(BankAccountTable))]
final class LITS_BankAccountTableForm_Extension
{
    public boolean canSubmitToWorkflow()
    {
        boolean canSubmitToWorkflow = next canSubmitToWorkflow();
        if(BankAccountTable.Workflowstate == LITS_BankAccountWGApprovalStatus::Draft)
        {
            canSubmitToWorkflow = true;
        }
        return canSubmitToWorkflow;

    }

}
```

- **Now go to Cash and bank management workflows**



https://usnconeboxax1aos.cloud.onebox.dynamics.com/?cmp=usmf&mi=BankAccountTableListPage

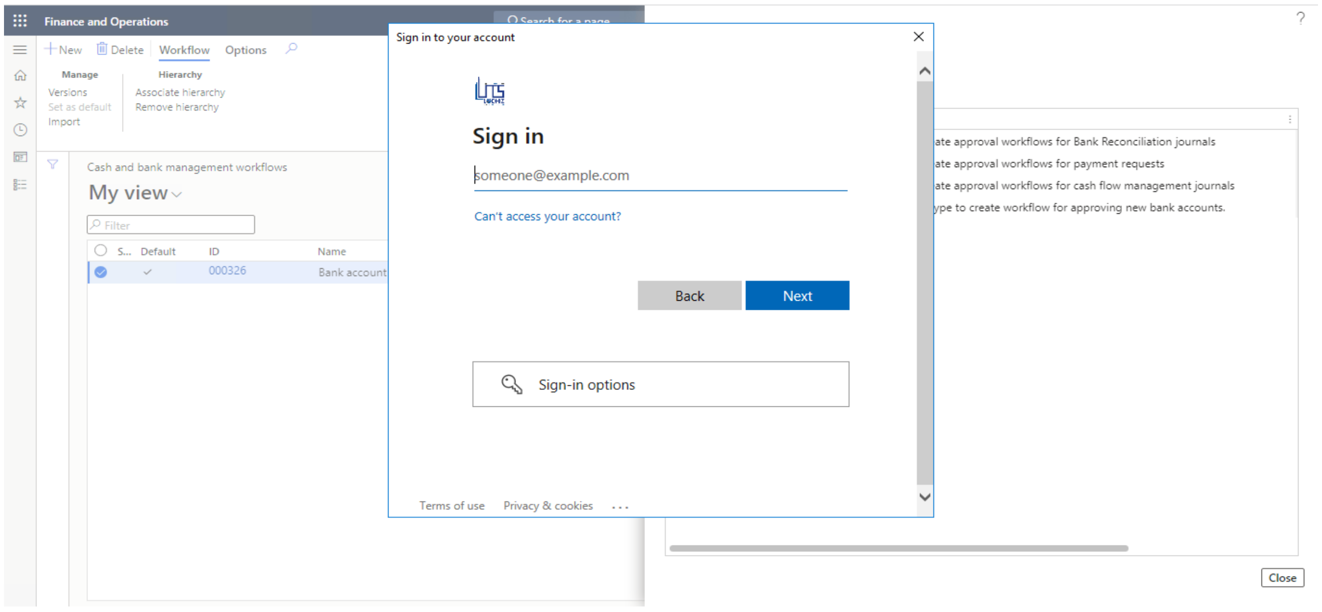
- **Click in new select BankAccountWorkflow**



S...	Default	ID	Name	Association	Type
	✓	000326	Bank account workflow	usmf	LITS_BankAccountWo

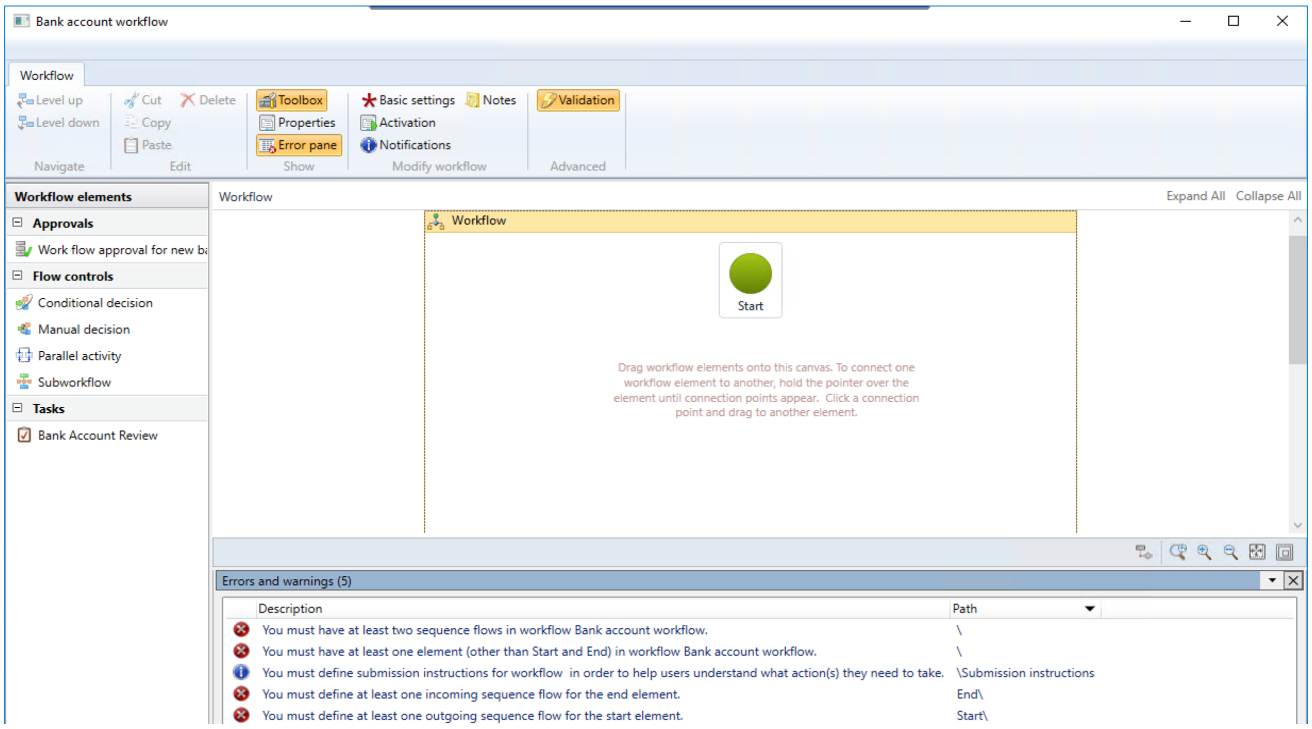
Workflow type	Description
Bank reconciliation journal ap...	Use this type to create approval workflows for Bank Reconciliation journals
Workflow type for payment re...	Use this type to create approval workflows for payment requests
Payment schedule journal ap...	Use this type to create approval workflows for cash flow management journals
Bank account workflow	Use this workflow type to create workflow for approving new bank accounts.

- **After clicking you must sign in with admin account.**



ate approval workflows for Bank Reconciliation journals
ate approval workflows for payment requests
ate approval workflows for cash flow management journals
ype to create workflow for approving new bank accounts.

- **After signing in this window will be appeared.**

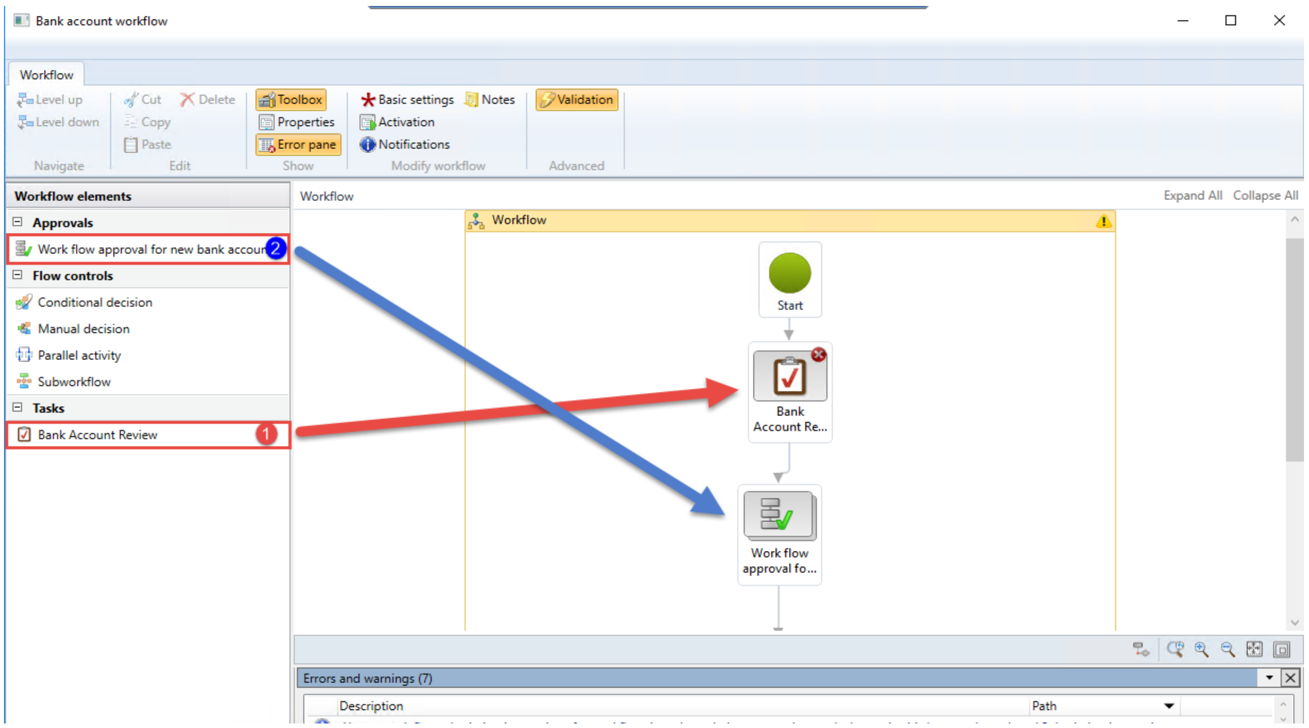


Drag workflow elements onto this canvas. To connect one workflow element to another, hold the pointer over the element until connection points appear. Click a connection point and drag to another element.

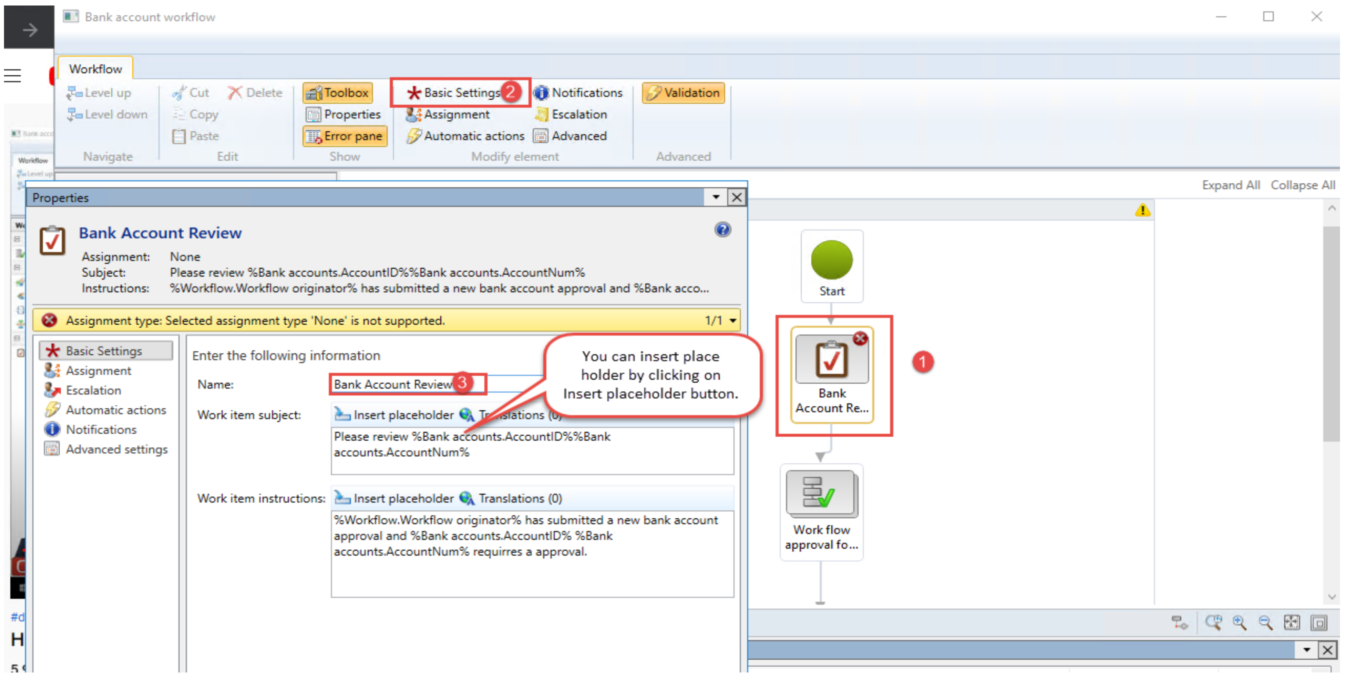
Errors and warnings (5)

Description	Path
You must have at least two sequence flows in workflow Bank account workflow.	\
You must have at least one element (other than Start and End) in workflow Bank account workflow.	\
You must define submission instructions for workflow in order to help users understand what action(s) they need to take.	\Submission instructions
You must define at least one incoming sequence flow for the end element.	End\
You must define at least one outgoing sequence flow for the start element.	Start\

- **Drag task and then approval in workflow area. You can link items by drawing a line through item nodes.**



- **Now add these information.**



Bank Account Review

Assignment: None
Subject: Please review %Bank accounts.AccountID%%Bank accounts.AccountNum%
Instructions: %Workflow.Workflow originator% has submitted a new bank account approval and %Bank acco...

Assignment type: Selected assignment type 'None' is not supported. 1/1

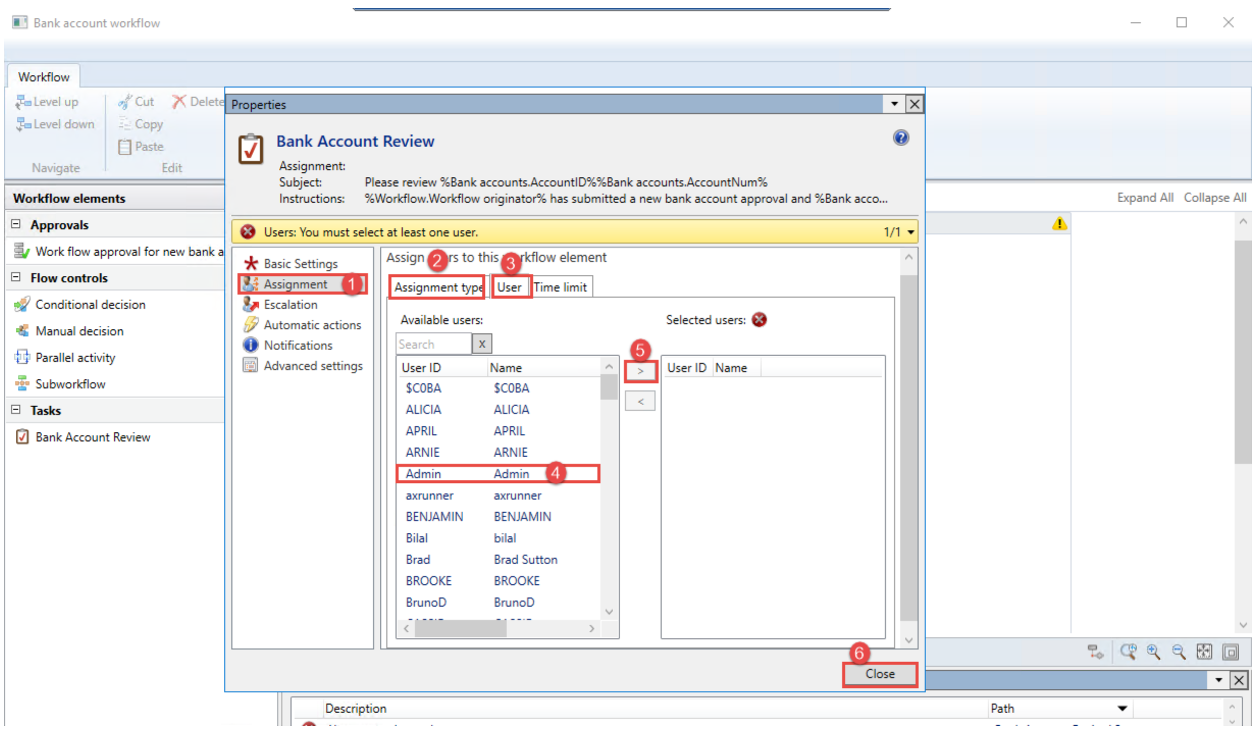
Enter the following information

Name: Bank Account Review

Work item subject: Please review %Bank accounts.AccountID%%Bank accounts.AccountNum%

Work item instructions: %Workflow.Workflow originator% has submitted a new bank account approval and %Bank accounts.AccountID% %Bank accounts.AccountNum% requirres a approval.

1. Click on Bank Account Review.
2. Click on Basic settings.
3. Rename name and complete work item subject and work item instructions.



Bank Account Review

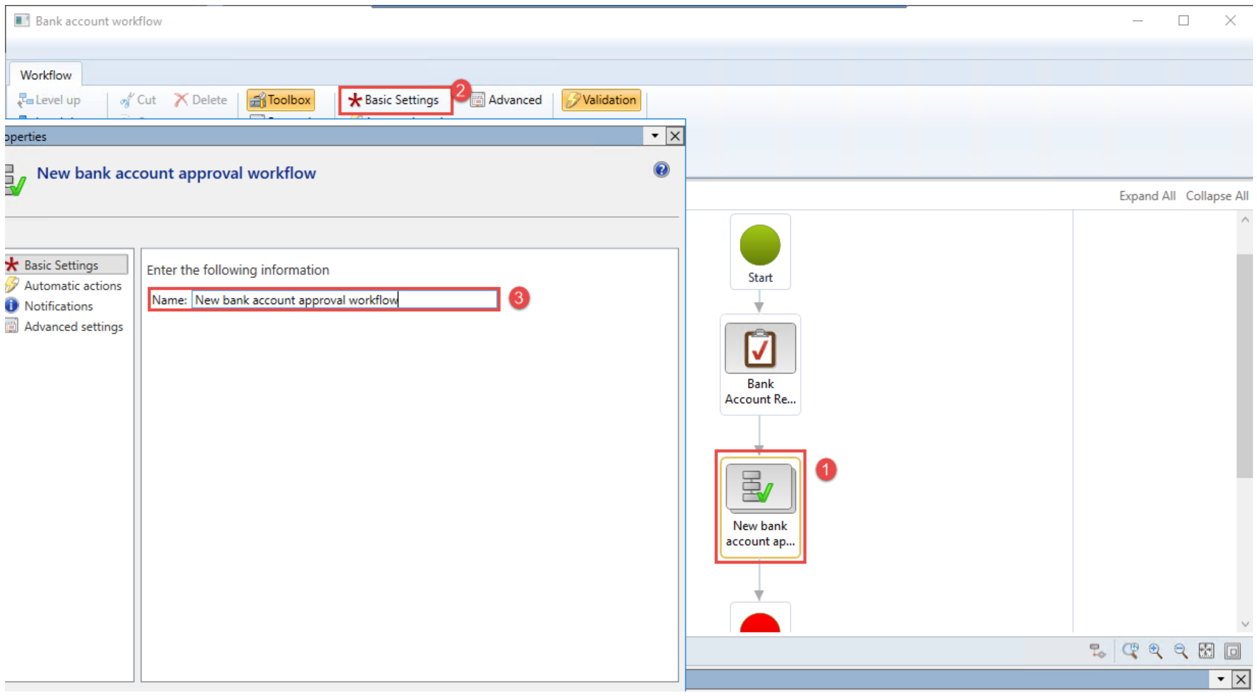
Assignment:
Subject: Please review %Bank accounts.AccountID%%Bank accounts.AccountNum%
Instructions: %Workflow.Workflow originator% has submitted a new bank account approval and %Bank acco...

Users: You must select at least one user. 1/1

Assign users to this workflow element

User ID	Name
$COBA	$COBA
ALICIA	ALICIA
APRIL	APRIL
ARNIE	ARNIE
Admin	Admin
axrunner	axrunner
BENJAMIN	BENJAMIN
Bilal	bilal
Brad	Brad Sutton
BROOKE	BROOKE
BrunoD	BrunoD

1. Now click on Assignment just next to Basic Settings.
2. Click on Assignment type and select **USER**.
3. Click on User Tab.
4. Select Admin.
5. Click on Right Arrow user will be added to right column.
6. Click on Close.



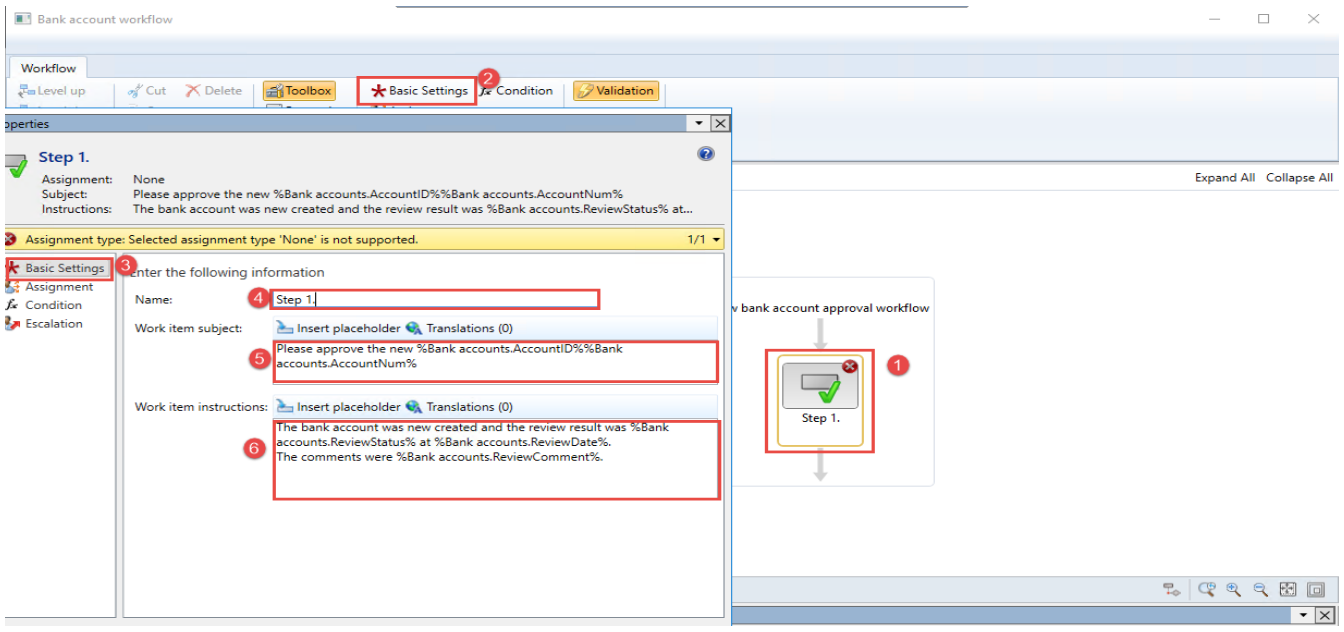
New bank account approval workflow

Enter the following information

Name: New bank account approval workflow

1. Click on Workflow Approval Task.
2. Click on Basic Settings.
3. Rename it.

- **Now double click on Workflow Approval Task it will take you to the Step1.**



Step 1.

Assignment: None

Subject: Please approve the new %Bank accounts.AccountID%%Bank accounts.AccountNum%

Instructions: The bank account was new created and the review result was %Bank accounts.ReviewStatus% at...

Assignment type: Selected assignment type 'None' is not supported. 1/1

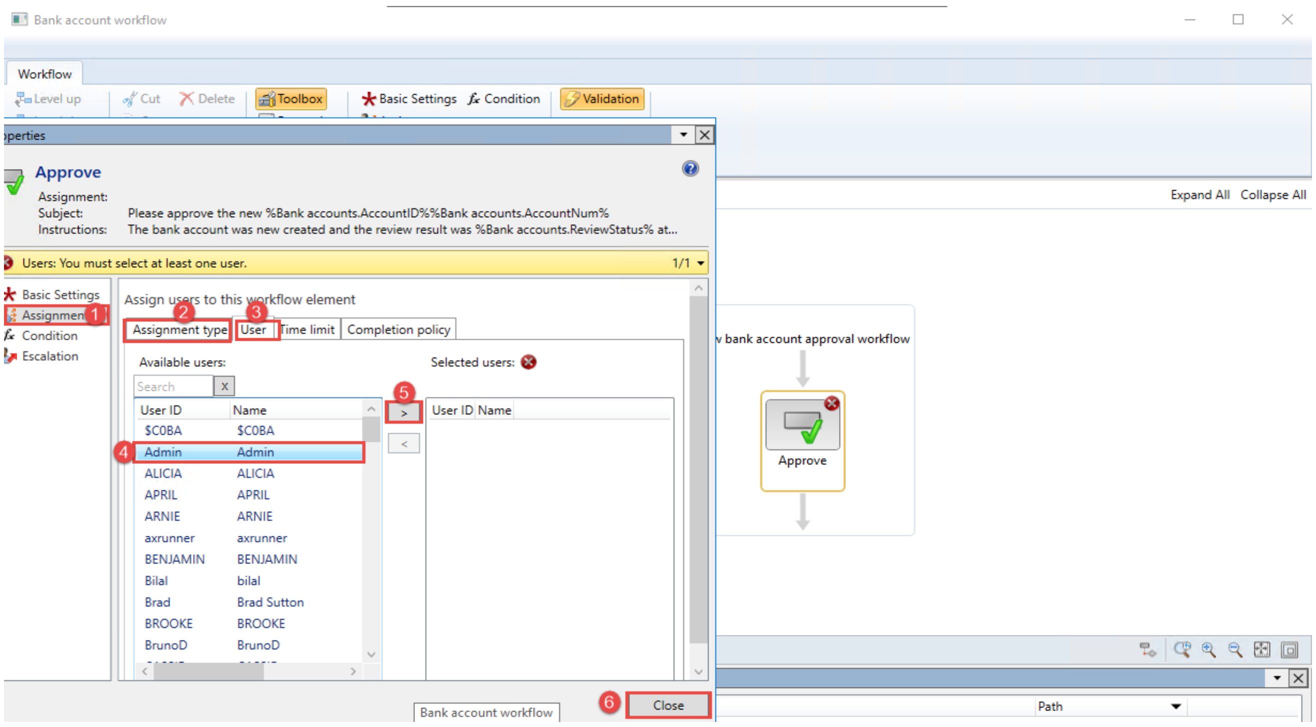
Enter the following information

Name: Step 1.

Work item subject: Please approve the new %Bank accounts.AccountID%%Bank accounts.AccountNum%

Work item instructions: The bank account was new created and the review result was %Bank accounts.ReviewStatus% at %Bank accounts.ReviewDate%.
The comments were %Bank accounts.ReviewComment%.

1. Click on step 1.
2. Click on Basic settings.
3. Select Tab.
4. Rename it to Approve.
5. Insert work item subject.
6. Insert work item instructions.

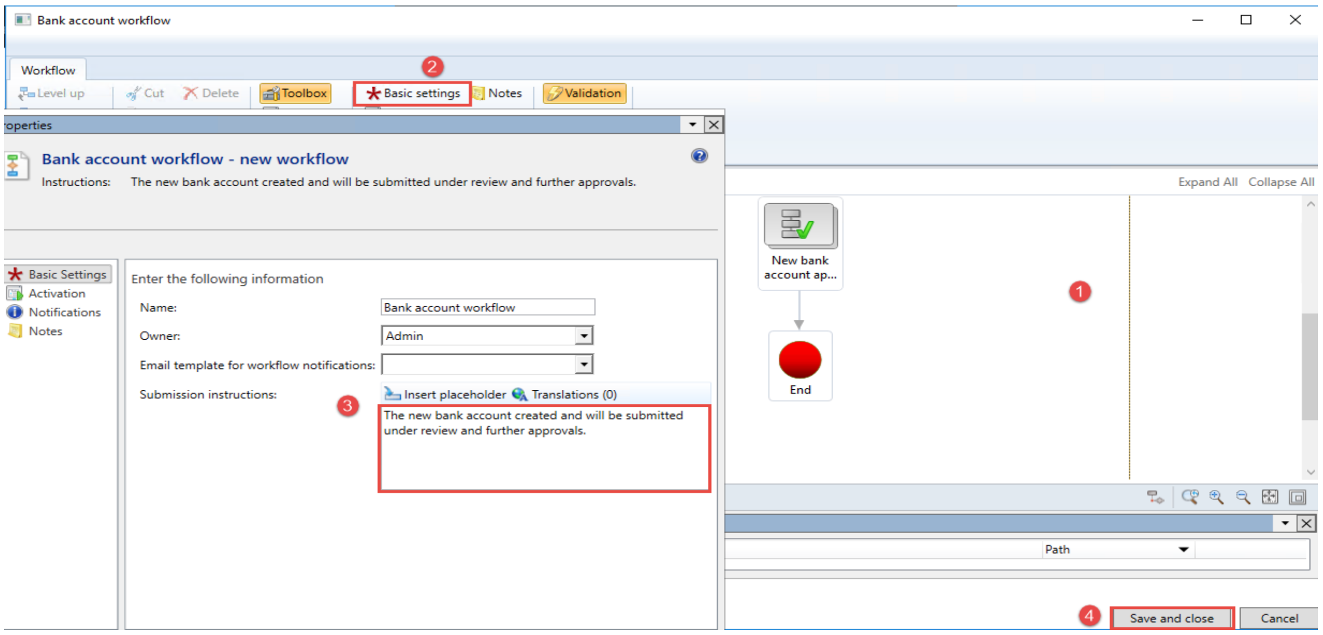


Assignment:
Subject: Please approve the new %Bank accounts.AccountID%%Bank accounts.AccountNum%
Instructions: The bank account was new created and the review result was %Bank accounts.ReviewStatus% at...

Users: You must select at least one user. 1/1

User ID	Name
$COBA	$COBA
Admin	Admin
ALICIA	ALICIA
APRIL	APRIL
ARNIE	ARNIE
axrunner	axrunner
BENJAMIN	BENJAMIN
Bilal	bilal
Brad	Brad Sutton
BROOKE	BROOKE
BrunoD	BrunoD

1. Click on Assignment.
2. Select Assignment type tab and select **USER.**
3. Click on User tab.
4. Select Admin.
5. Click on Right arrow admin user will be added to right hand column.
6. Click on Close and go to main page by clicking levelup.



Bank account workflow - new workflow

Instructions: The new bank account created and will be submitted under review and further approvals.

Enter the following information

Name: Bank account workflow

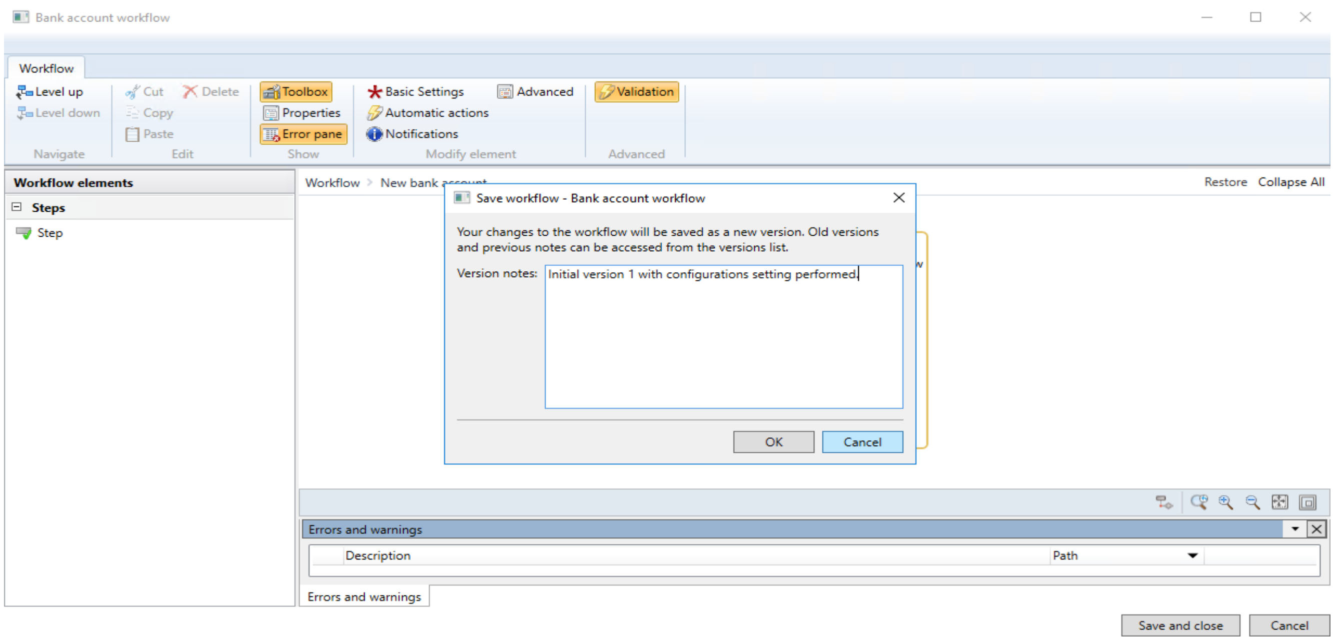
Owner: Admin

Email template for workflow notifications:

Submission instructions: The new bank account created and will be submitted under review and further approvals.

1. Click on empty space.
2. Select Basic settings.
3. Insert instructions.
4. Click save and close.

- **A dialog box will be appeared.**



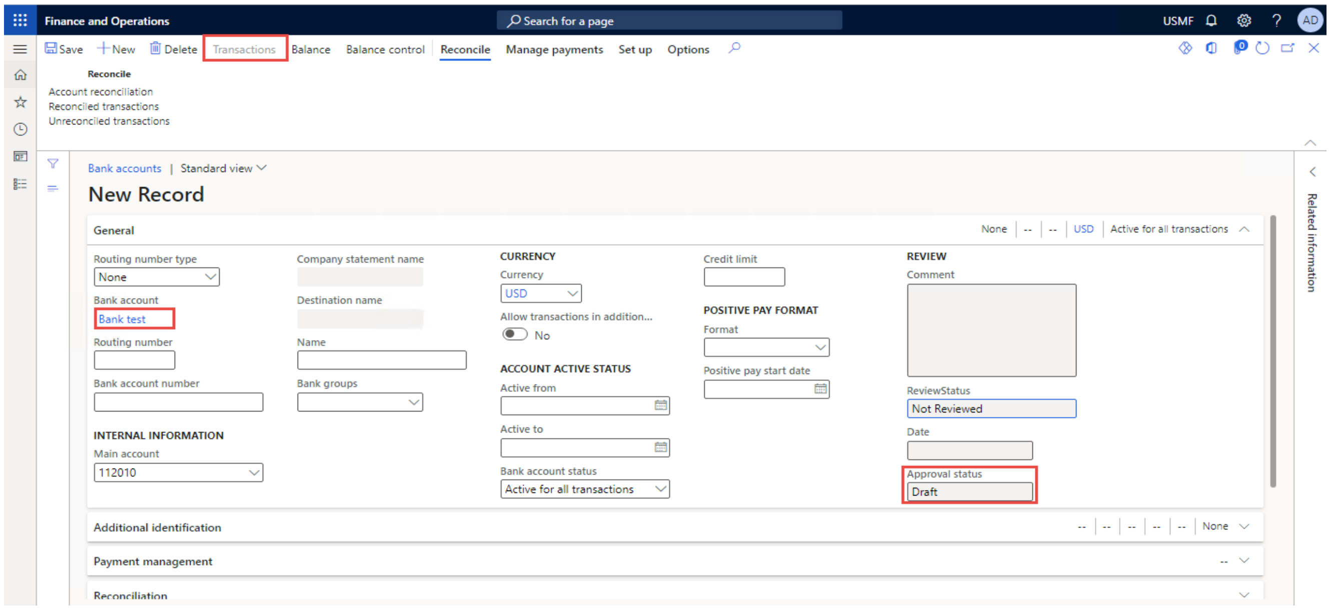
Save workflow - Bank account workflow

Your changes to the workflow will be saved as a new version. Old versions and previous notes can be accessed from the versions list.

Version notes: Initial version 1 with configurations setting performed.

After inserting version notes click on OK and select Activate the new version and select OK.

- Just for example I have disabled the Transaction button and it will be enabled when workflow status will be Approved.



New Record

General — None | -- | -- | USD | Active for all transactions

Routing number type: None
Bank account: Bank test
Routing number:
Bank account number:

INTERNAL INFORMATION
Main account: 112010

Company statement name:
Destination name:
Name:
Bank groups:

CURRENCY
Currency: USD
Allow transactions in addition...: No

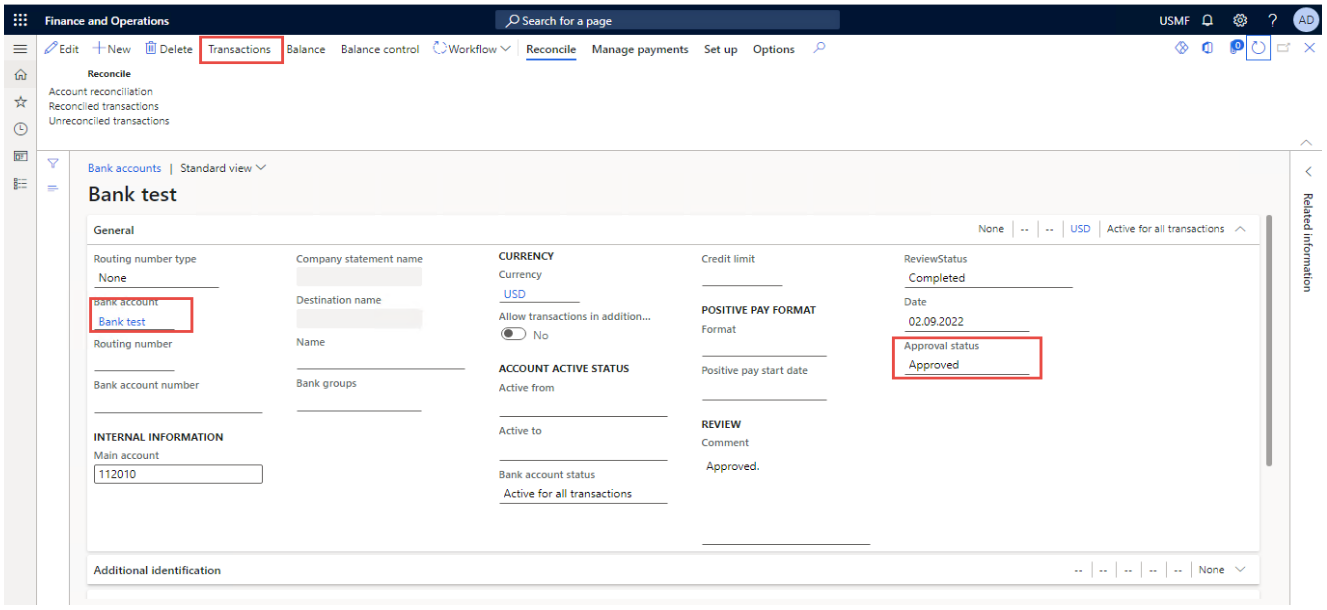
ACCOUNT ACTIVE STATUS
Active from:
Active to:
Bank account status: Active for all transactions

Credit limit:

POSITIVE PAY FORMAT
Format:
Positive pay start date:

REVIEW
Comment:
ReviewStatus: Not Reviewed
Date:
Approval status: Draft

Additional identification — -- | -- | -- | -- | -- | None
Payment management — --
Reconciliation



Bank test

General — None | -- | -- | USD | Active for all transactions

Routing number type: None
Bank account: Bank test
Routing number:
Bank account number:

INTERNAL INFORMATION
Main account: 112010

Company statement name:
Destination name:
Name:
Bank groups:

CURRENCY
Currency: USD
Allow transactions in addition...: No

ACCOUNT ACTIVE STATUS
Active from:
Active to:
Bank account status: Active for all transactions

Credit limit:

POSITIVE PAY FORMAT
Format:
Positive pay start date:

REVIEW
Comment: Approved.

ReviewStatus: Completed
Date: 02.09.2022
Approval status: Approved

Additional identification — -- | -- | -- | -- | -- | None

End of Document