ACUMATICA TEST SDK

DEVELOPER GUIDE

Acumatica

Contents

Copyright . . . 4
Part 1: Acumatica Test SDK—Certification Training . . . 5
Course Prerequisites . . . 6
Acumatica Test SDK Overview . . . 8
Configuring Tests Solution . . . 10
Implementing the Test . . . 15
Running the Test . . . 20
Part 2. Acumatica Test SDK—API Reference . . . 22
PX.QA.Tools.dll . . . 23
Check . . . 24
Config . . . 26
Log . . . 28
Core.dll . . . 29
Browser . . . 30
Button . . . 31
CheckBox . . . 33
DateSelector . . . 35
DropDown . . . 38
Selector . . . 40
TreeSelector . . . 43
Grid . . . 44
GroupBox . . . 47
ImageUploader . . . 50
Input . . . 51
Label . . . 53
RichTextEdit . . . 54
ToolBarButton . . . 55
TreeView . . . 57
ClassGenerator.exe . . . 60
ClassGenerator.exe.config . . . 61
Part 3: Acumatica Test SDK—Programming Tasks . . . 64
Known Issues . . . 65
How to Use Page Wrapper Generation Tool . . . 66
How to Generate Page Wrappers Using Test SDK API . . . 67
How to Work With Errors That Occur During Page Wrapper Generation . . . 69
How to Change the Settings of the Page Wrapper Generation Tool . . . 75
How to Change Browser Settings . . . 78
How to Change Chrome Settings . . . 79
How to Change Culture Settings . . . 80
How to Change Predefined Timeouts . . . 82
How to Manage Log Providers . . . 83
How to Work with Alerts . . . 85
How to Commit Changes in Rows of a Detail Table . . . 88
How to Navigate through the Rows of a Detail Table . . . 89
How to Show or Hide Columns in a Detail Table . . . 90
How to Use Column Filters of a Detail Table . . . 93
How to Work with Errors . . . 95
How to Work with Warnings . . . 97
How to Work with Windows . . . 99
How to Work with the Names of UI Elements . . . 100
How to Work with Drop-Down Menus on Toolbars . . . 101
How to Work with Pop-Up Dialog Boxes . . . 102
How to Work with Pop-Up Panels . . . 104
How to Verify a Note on a Form . . . 105
How to Add a Note to a Detail Line . . . 106
How to Add a Note to an Acumatica Form . . . 107
How to Upload a File . . . 108
How to Upload Data from an Excel File into a Detail Table . . . 109
How to Upload Data from a CSV File into a Detail Table . . . 110
How to Attach a File to an Acumatica Form . . . 111
How to Attach a File to a Detail Line . . . 112
How to Remove an Attached File from an Acumatica Form . . . 113
How to Remove an Attached File from a Detail Line . . . 114
How to Publish Customization Projects . . . 115
How to Work with Smart Delays . . . 116
How to Capture Screenshots . . . 118
How to Get Datetime in the Current User's Timezone . . . 119
How to use Comparator to compare .xml, .csv, .pdf, Excel files . . . 120
How to Verify string/long/int/DateTime returned by a method . . . 121
How to work with dynamic controls . . . 122

Test SDK

Acumatica Test SDK is a programming framework that provides an easy way to develop automated tests for applications built on top of Acumatica Framework. This document provides important information about Acumatica Test SDK, including its structure, API reference, programming guidelines, and installation and configuration instructions. The document is of particular interest to those who develop extensions or customizations by using Acumatica Framework and want to automate the testing of the functionality that they have developed.

Copyright

4030 Lake Washington Blvd NE, Suite 100

Kirkland, WA 98033

Restricted Rights

The product is provided with restricted rights. Use, duplication, or disclosure by the United States Government is subject to restrictions as set forth in the applicable License and Services Agreement and in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 or subparagraphs (c)(1) and (c)(2) of the Commercial Computer Software-Restricted Rights at 48 CFR 52.227-19, as applicable.

Disclaimer

Acumatica, Inc. makes no representations or warranties with respect to the contents or use of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Acumatica, Inc. reserves the right to revise this document and make changes in its content at any time, without obligation to notify any person or entity of such revisions or changes.

Trademarks

Acumatica is a registered trademark of Acumatica, Inc. All other product names and services herein are trademarks or service marks of their respective companies.

Part 1: Acumatica Test SDK—Certification Training

In This Part

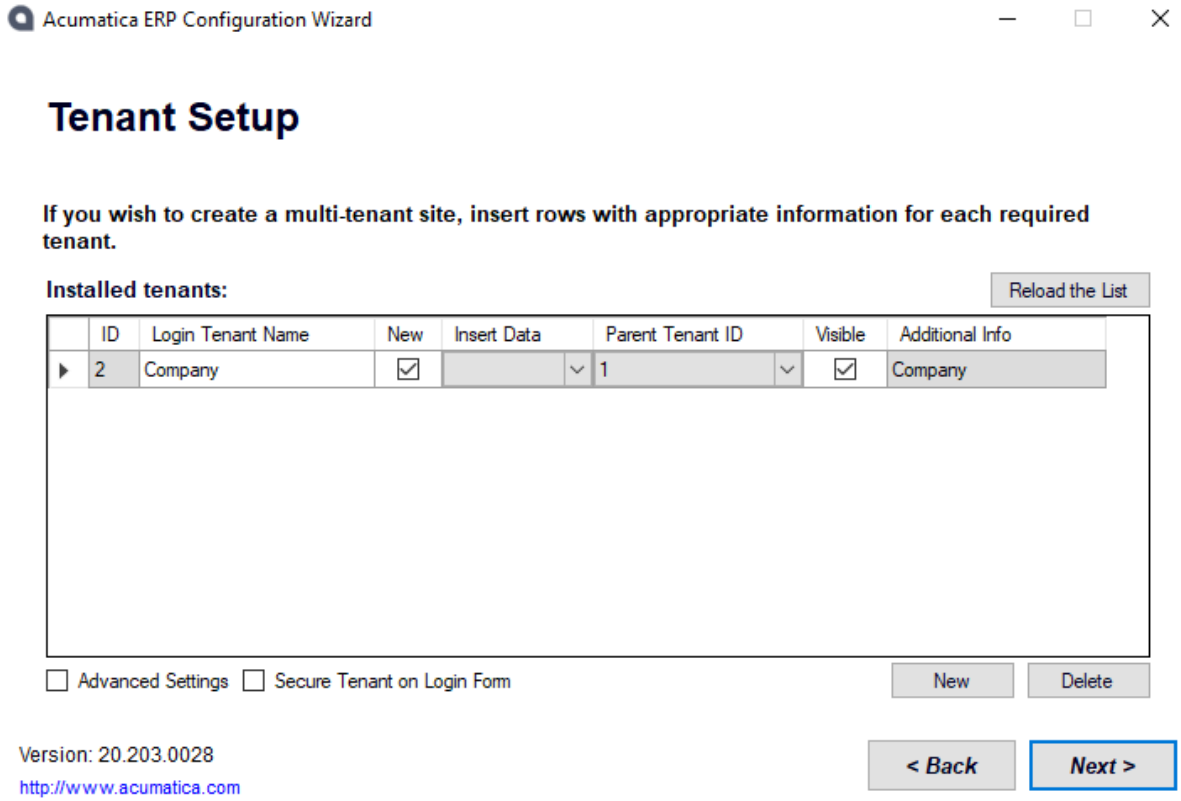
- Course Prerequisites
- Acumatica Test SDK Overview
- Configuring Tests Solution
- Implementing the Test
- Running the Test

Course Prerequisites

- Acumatica ERP Instance
- Downloading Acumatica Test SDK
- Creating Acumatica Test Solution

Acumatica ERP Instance

Acumatica ERP instance should be deployed with default tenant settings



The first login is admin, and the password to log in to the new company is setup; you should change the password before running your tests.

Downloading Acumatica Test SDK

You should download the preferred version of Acumatica Test SDK from http://acumatica-builds.s3.amazonaws.com/index.html?prefix=builds. Deploy the Acumatica Test SDK and unzip it to a folder on your local computer.

> You must use the same version of both Acumatica ERP and Acumatica Test SDK.

Creating Acumatica Test Solution

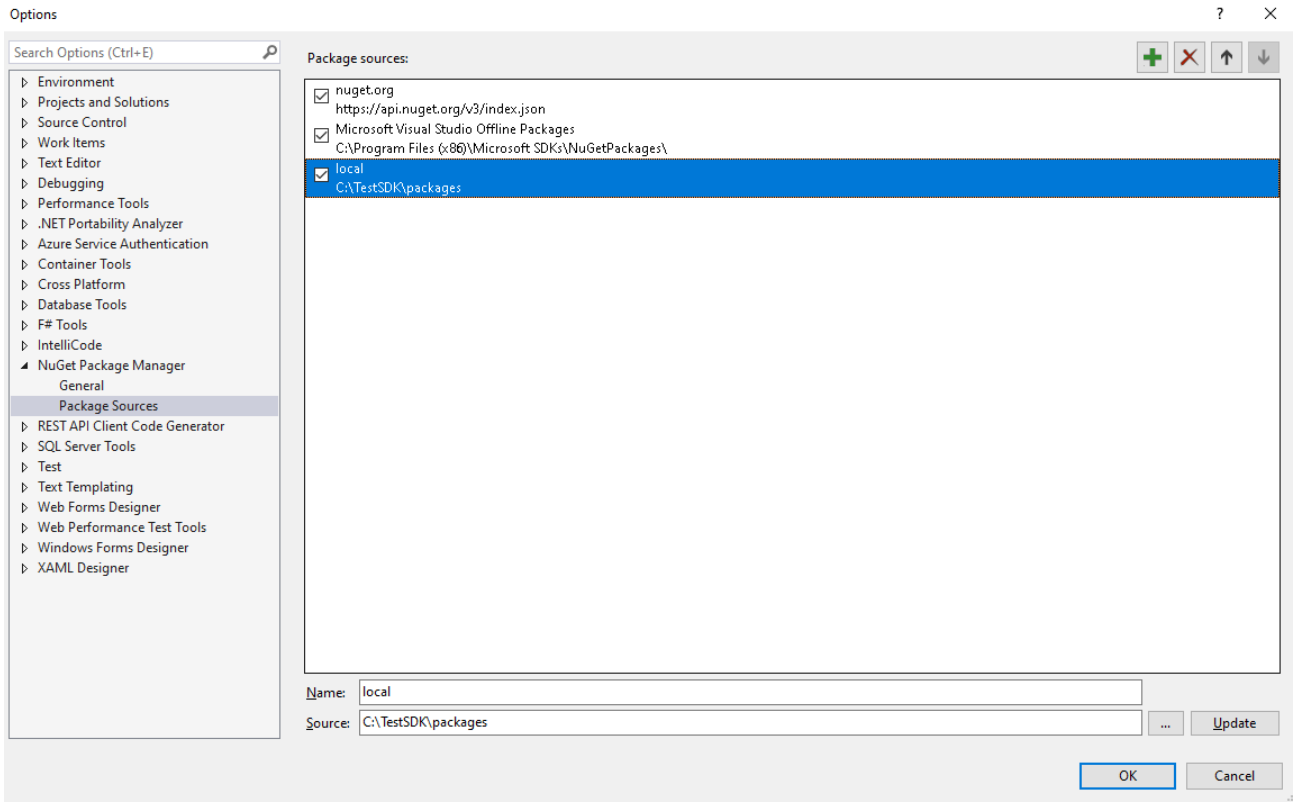
Create a test solution by using latest Visual Studio as follows:

1. In Visual Studio, create a new project Tests with the following parameters:

- Project template: Console App (.NET Core) C#
- Project name: Tests
- Solution name: Tests

> Project name must start from the word Tests.

2. In Visual Studio, add packages folder from your Test SDK as a nuget source



2. Add package references to the Tests project:

- Execution
- GeneratedWrappers.Acumatica (you may need to check "Enable prerelease" in NuGet Packages Manager and add version to the file name in the packages folder, e.g. GeneratedWrappers.Acumatica.21.102.63-P106163.nupkg)

At this point your Tests.csproj file should look like:

```
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>Exe</OutputType>
    <TargetFramework>net5.0</TargetFramework>
  </PropertyGroup>
  <ItemGroup>
    <PackageReference Include="Execution" Version="21.102.70.71" />
    <PackageReference Include="GeneratedWrappers.Acumatica" Version="21.102.63-P106163" />
  </ItemGroup>
</Project>
```

3. Open Program.cs file of the Tests project (or create a new one if it was not created automatically on step 1) and modify it as follows:

```
namespace Tests
{
    public class Program
    {
        public static int Main(string[] args) => Execution.Launcher.Main(args);
    }
}
```

Acumatica Test SDK Overview

- Understanding the Acumatica Test SDK Components
- Acumatica Test SDK Components

Understanding the Acumatica Test SDK Components

You will use the following components to create and run tests for Acumatica ERP or products based on Acumatica Framework:

- Browser: You use a browser to run and test an Acumatica-based product.
- Page wrapper generation tool: You use this tool to create wrappers for the pages of your Acumatica-based product.
- Test framework: You include the generated page wrappers in your test solution and create the code of your tests by using the classes available in the wrappers. You use the test runner to run the created tests in the browser.
- Selenium WebDriver: The test runner uses the WebDriver for interaction with the browser.

> The page wrapper generation tool creates object mapping model for every page developed by Acumatica Framework. You can access any UI element such as Forms, Grids, Toolbars, data-fields available on a page which you use operating as an ordinary user with Acumatica ERP or any other Acumatica based product. Test SDK uses built-in control wrapper to construct complex objects such as forms, grids and pages. You need to create wrappers for each page that you want to test.

The interaction between these components is illustrated in the following diagram.

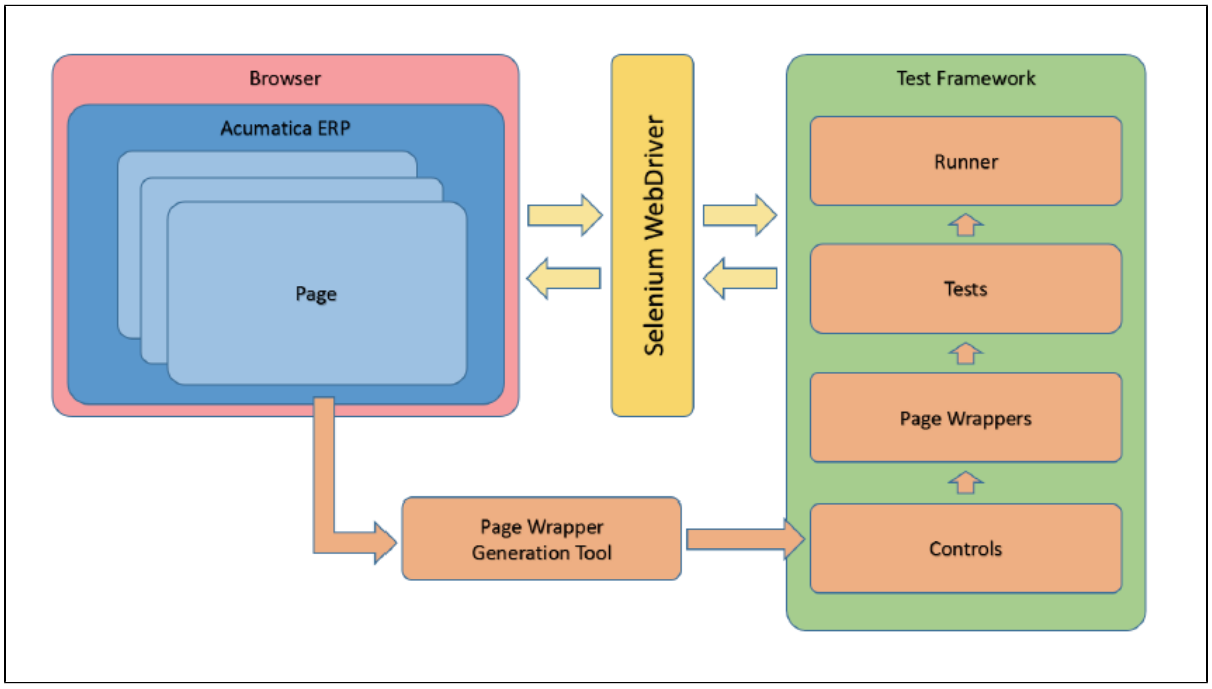


Figure: Acumatica Test SDK Components

Acumatica Test SDK Components

Acumatica Test SDK consists of the components in the following table.

	Folder	File Name or Subfolder	Description
1	Test SDK > Chrome	chrome.exe and other files/folders	The Google Chrome browser application
2	Test SDK > Firefox	firefox.exe and other files/folders	The Mozilla Firefox browser application
3	Test SDK > ClassGenerator	ClassGenerator.exe	The page wrapper generation tool. You use this console application to generate all your page wrappers

		ClassGenerator.exe.config	An example of the configuration file, which contains the wrapper generation settings that the page wrapper generation tool uses You can find more details on how to change these settings in How to Change the Settings of the Page Wrapper Generation Tool
		Host.dll	A service component that is used to create proxy objects to generate page wrappers
4	Test SDK > packages	*.nupkg	TestSDK nuget packages
5	Test SDK	Config.xml	An example of the configuration file that contains test settings used by the test runner
		SDK - README.pdf	The Test SDK documentation

Configuring Tests Solution

- Step 1: Creating Extension Classes for Generated Page Wrappers
- Step 2: Including Extension Classes Into Project
- Step 3: Creating Test

Step 1: Creating Extension Classes for Generated Page Wrappers

In this step, you will create extension classes for page wrappers from the GeneratedWrappers.Acumatica package.

- SM201010_AccessUsers
- SM201025_ModernAccess
- SM201060_PreferencesSecurityMaint
- SM208600_DashboardMaint
- CS100000_FeaturesMaint
- CS101500_OrganizationMaint
- CS102000_BranchMaint
- AP101000_APSetupMaint
- AR101000_ARSetupMaint
- GL102000_GLSetupMaint
- GL202500_AccountMaint
- SO101000_SOSetupMaint
- EP202500_EPLoginTypeMaint
- DB000038

> By means of modifications in extension classes you grant access to automatically generated protected properties of the page wrappers and can change their default behaviour. By default, the page wrapper generation tool marks all automatically generated properties that grant access to container properties as protected. To access these properties from the test, create an extension class.

Create extension classes (cs files with the appropriate names) for each automatically generated page wrapper from the list below:

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public class User : SM201010_AccessUsers
    {
        public c_userlist_form Summary => UserList_form;

        public c_allowedroles_gridroles Roles => AllowedRoles_gridRoles;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public class AccessRightsByRole : SM201025_ModernAccess
    {
        public c_roles_form Summary => Roles_form;

        public c_roleentities_griddet Details => RoleEntities_griddet;

        public c_parameters ParametersTree => Parameters;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class SecurityPreferences : SM201060_PreferencesSecurityMaint
    {
        public c_prefs_form GeneralSettings => Prefs_form;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public class Dashboards : SM208600_DashboardMaint
    {
        public c_dashboards_frmheader Summary => Dashboards_frmHeader;

        public c_entityroles_gridroles Details => EntityRoles_gridRoles;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class Features : CS100000_FeaturesMaint
    {
        public c_features_form Summary => Features_form;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public class Company : CS101500_OrganizationMaint
    {
        public c_baccount_pxformview1 Summary => BAccount_PXFormView1;

        public c_defaddress_defaddress DefaultAddress => DefAddress_DefAddress;

        public c_organizationview_pxformview1 OrganizationSettings => OrganizationView_PXFormView1;

        public c_organizationview_company BaseCurrency => OrganizationView_Company;

        public c_commonsetup_commonsettings CommonSettings => Commonsetup_commonsettings;

        public c_organizationledgerlinkwithledgerselect_grdledgerlinks Ledgers => OrganizationLedgerLinkWithLedgerSelect_grdLedgerLinks;

        public c_branchesview_grdbranches Branches => BranchesView_grdBranches;

        public c_employees_grdemployees Employees => Employees_grdEmployees;

        public c_organizationview_configurationsettings ConfigurationSettings => OrganizationView_ConfigurationSettings;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class Branch : CS102000_BranchMaint
    {
        public c_baccount_pxformview1 Summary => BAccount_PXFormView1;

        public c_defaddress_defaddress DefaultAddress => DefAddress_DefAddress;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class SetupAp : AP101000_APSetupMaint { }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class SetupAr : AR101000_ARSetupMaint { }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class SetupGl : GL102000_GLSetupMaint
    {
        public c_glsetuprecord_form GeneralSettings => GLSetupRecord_form;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class Account : GL202500_AccountMaint
    {
        public c_accountrecords_grid Details => AccountRecords_grid;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public partial class SetupSo : SO101000_SOSetupMaint { }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public class UserTypes : EP202500_EPLoginTypeMaint
    {
        public c_logintype_form Summary => LoginType_form;

        public c_allowedroles_allowsrolesgrid AllowedRoles => AllowedRoles_allowsRolesGrid;
    }
}
```

```
using GeneratedWrappers.Acumatica;

namespace Tests.Extensions
{
    public class Dashboard : DB000038 { }
}
```

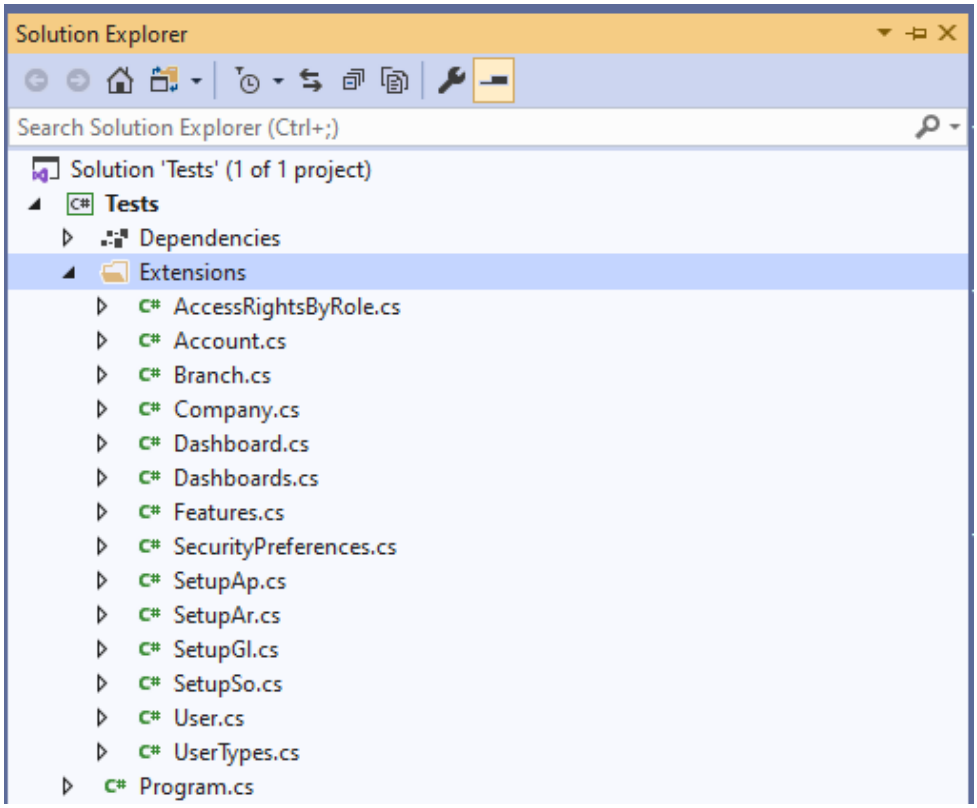
Step 2: Including Extension Classes Into Project

In this step, you will include extension classes for page wrappers into project.

To include extension classes for page wrappers into project, do the following:

1. Create an Extensions folder in the Tests project.
2. Include all extension classes from the previous step into it.

At this point your solution should look like shown on the screenshot below:



To simplify maintenance place extension classes in a separate folder of the project.

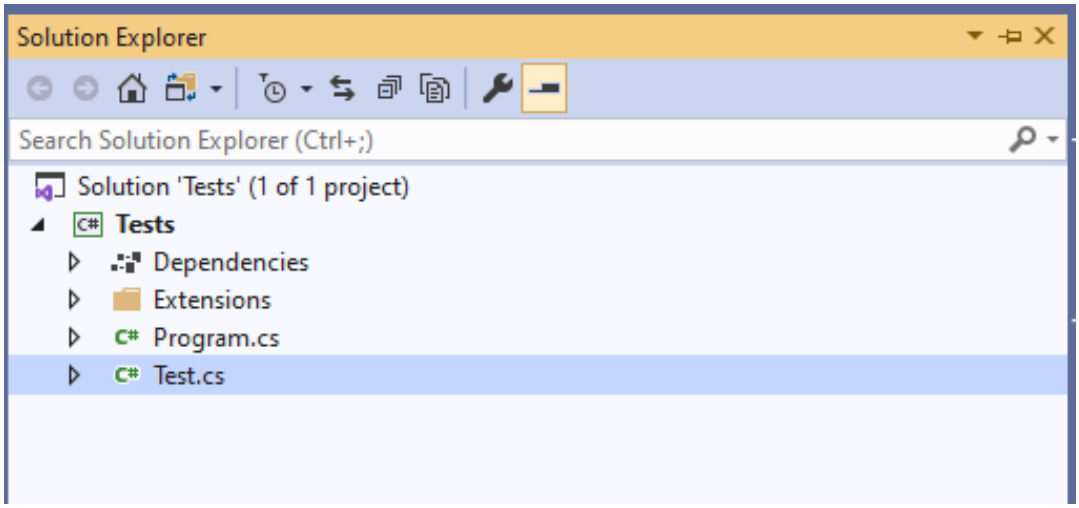
Step 3: Creating Test

In this step, you will create a template for your test.

Create Test class (Test.cs) as shown below and include it into the project:

```
using Controls.Dashboard;
using Core;
using Core.Config;
using Core.Login;
using Core.TestExecution;
using Tests.Extensions;

namespace Tests
{
    //Use the Check class as a parent for every test.
    public class Test : Check
    {
  public override void Execute()
        {
   //Add test-specific logic in this method.
        }
    }
}
```

At this point your solution should look like shown on the screenshot below:


Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Tests' (1 of 1 project)
Tests
Dependencies
Extensions
Program.cs
Test.cs

Implementing the Test

- 1. Adding wrapper extensions to the test
- 2. Adding steps to the test

1. Adding wrapper extensions to the test

Modify Test class according to:

```
using Controls.Dashboard;
using Core;
using Core.Config;
using Core.Login;
using Core.TestExecution;
using Tests.Extensions;

namespace Tests
{
    //Use the Check class as a parent for every test.
    public class Test : Check
    {
        private readonly SecurityPreferences SecurityPreferences = new SecurityPreferences();
        private readonly AccessRightsByRole AccessRightsByRole = new AccessRightsByRole();
        private readonly Dashboards Dashboards = new Dashboards();
        private readonly Dashboard Dashboard = new Dashboard();
        private readonly UserTypes UserTypes = new UserTypes();
        private readonly Features Features = new Features();
        private readonly Company Company = new Company();
        private readonly Account Account = new Account();
        private readonly SetupSo SetupSo = new SetupSo();
        private readonly SetupGl SetupGl = new SetupGl();
        private readonly SetupAp SetupAp = new SetupAp();
        private readonly SetupAr SetupAr = new SetupAr();
        private readonly Branch Branch = new Branch();
        private readonly User User = new User();

        public override void Execute()
        {
            //Add test-specific logic in this method.
        }
    }
}
```

2. Adding steps to the test

Modify Test class according to:

```
using Controls.Dashboard;
using Core;
using Core.Config;
using Core.Login;
using Core.TestExecution;
using Tests.Extensions;

namespace Tests
{
    //Use the Check class as a parent for every test.
    public class Test : Check
    {
        private readonly SecurityPreferences SecurityPreferences = new SecurityPreferences();
        private readonly AccessRightsByRole AccessRightsByRole = new AccessRightsByRole();
        private readonly Dashboards Dashboards = new Dashboards();
        private readonly Dashboard Dashboard = new Dashboard();
        private readonly UserTypes UserTypes = new UserTypes();
        private readonly Features Features = new Features();
        private readonly Company Company = new Company();
        private readonly Account Account = new Account();
        private readonly SetupSo SetupSo = new SetupSo();
        private readonly SetupGl SetupGl = new SetupGl();
        private readonly SetupAp SetupAp = new SetupAp();
        private readonly SetupAr SetupAr = new SetupAr();
        private readonly Branch Branch = new Branch();
        private readonly User User = new User();

        public override void Execute()
        {
            //Add test-specific logic in this method.
            PxLogin.LoginToDestinationSite();

            #region TestCase 1 - Operationg with few branches/companies
            using (TestExecution.CreateTestCaseGroup("Operationg with few branches/companies"))
            {
                #region Step 1 - Create first company
                using (TestExecution.CreateTestStepGroup("Create first company"))
                {
                    Features.OpenScreen();
                    Features.Insert();
                    Features.Summary.Branch.SetTrue();
                    Features.Summary.DistributionModule.SetTrue();
                    Features.RequestValidation();

                    Company.OpenScreen();
                    Company.Insert();
                    Company.Summary.AcctCD.Type("FIRSTCOMP");
                    Company.Summary.AcctName.Type("First company");
                    Company.OrganizationSettings.OrganizationType.Select("Without Branches");
                    Company.DefaultAddress.CountryID.Select("RU");
                    Company.BaseCurrency.BaseCuryID.Select("RUB");
```

```
    Company.CommonSettings.WeightUOM.Select("KG");
    Company.CommonSettings.VolumeUOM.Select("LITER");
    Company.Save();
    VerifyCompanyVisibility(1, 0, "First company");
}
#endregion

#region Step 2 - Create second company
using (TestExecution.CreateTestStepGroup("Create second company"))
{
    Company.Insert();
    Company.Summary.AcctCD.Type("SECONDCOMP");
    Company.Summary.AcctName.Type("Second company");
    Company.OrganizationSettings.OrganizationType.Select("Without Branches");
    Company.DefaultAddress.CountryID.Select("AL");
    Company.Save();
    VerifyCompanyVisibility(2, 0, "First company", "Second company");
}
#endregion

#region Step 3 - Remove first company
using (TestExecution.CreateTestStepGroup("Remove first company"))
{
    Company.Insert();
    Company.Summary.AcctCD.Select("FIRSTCOMP");
    Company.Delete();
    VerifyCompanyVisibility(1, 0, "Second company");
}
#endregion

#region Step 4 - Add new one with branches
using (TestExecution.CreateTestStepGroup("Add new one with branches"))
{
    Company.Insert();
    Company.Summary.AcctCD.Type("THIRDCOMP");
    Company.Summary.AcctName.Type("Third company");
    Company.OrganizationSettings.OrganizationType.Select("With Branches Not Requiring Balancing");
    Company.DefaultAddress.CountryID.Select("CA");
    Company.Save();
    VerifyCompanyVisibility(1, 0, "Second company");
}
#endregion

#region Step 5 - Create few branches
using (TestExecution.CreateTestStepGroup("Create few branches"))
{
    Branch.OpenScreen();
    Branch.Insert();
    Branch.Summary.AcctCD.Type("THBRANCH1");
    Branch.Summary.AcctName.Type("Third Cmp Branch 1");
    Branch.Summary.OrganizationID.Select("THIRDCOMP");
    Branch.DefaultAddress.CountryID.Select("AF");
    Branch.Save();
    VerifyCompanyVisibility(2, 1, "Second company", "Third company", "Third Cmp Branch 1");
    Branch.Insert();
    Branch.Summary.AcctCD.Type("THBRANCH2");
    Branch.Summary.AcctName.Type("Third Cmp Branch 2");
    Branch.Summary.OrganizationID.Select("THIRDCOMP");
    Branch.DefaultAddress.CountryID.Select("AL");
    Branch.Save();
    VerifyCompanyVisibility(2, 2, "Second company", "Third company", "Third Cmp Branch 1", "Third Cmp Branch 2");
}
#endregion

#region Step 6 - Add user and verify company visibility
using (TestExecution.CreateTestStepGroup("Add user and verify company visibility"))
{
    SecurityPreferences.OpenScreen();
    SecurityPreferences.GeneralSettings.IsPasswordMinLength.SetFalse();
    SecurityPreferences.GeneralSettings.PasswordComplexity.SetFalse();
    SecurityPreferences.Save();

    UserTypes.OpenScreen();
    UserTypes.Summary.LoginTypeName.Type("NewType");
    UserTypes.Summary.Description.Type("NewUserType");
    UserTypes.AllowedRoles.New();
    UserTypes.AllowedRoles.Row.Rolename.Select("Customizer");
    UserTypes.Save();

    User.OpenScreen();
    User.Insert();
    User.Summary.Username.Type("user1");
    User.Summary.LoginTypeID.Select("NewType");
    User.Summary.Email.Type("test@con.con");
    User.Roles.Row.Selected.SetTrue();
    User.Save();
    User.ResetPassword();
    User.Summary.NewPassword.Type(Config.SITE_DST_PASSWORD);
    User.Summary.ConfirmPassword.Type(Config.SITE_DST_PASSWORD);
    User.Summary.Ok();

    AccessRightsByRole.OpenScreen();
    AccessRightsByRole.Summary.Rolename.Select("Customizer");
    AccessRightsByRole.ParametersTree.Tree.Select("COMPANY");
    foreach (var workspace in AccessRightsByRole.Details.Columns.RoleDescr.GetValues())
    {
        AccessRightsByRole.Details.SelectRow(AccessRightsByRole.Details.Columns.RoleDescr, workspace);
        AccessRightsByRole.Details.Row.RoleRight.Select("Granted");
    }
    AccessRightsByRole.Save();

    PxLogin.LogIn("user1", Config.SITE_DST_PASSWORD);
    VerifyCompanyVisibility(2, 2, "Second company", "Third company", "Third Cmp Branch 1", "Third Cmp Branch 2");
}
#endregion

#region Step 7 - Set Access Role for companies
using (TestExecution.CreateTestStepGroup("Set Access Role for companies"))
{
    PxLogin.LoginToDestinationSite();
    Company.OpenScreen();
    Company.Summary.AcctCD.Select("SECONDCOMP");
    Company.ConfigurationSettings.RoleName.Select("Customizer");
    Company.Save();
    VerifyCompanyVisibility(1, 0, "Second company");

    PxLogin.LogIn("user1", Config.SITE_DST_PASSWORD);
    VerifyCompanyVisibility(1, 0, "Second company");
```

```
            PxLogin.LoginToDestinationSite();
            Company.OpenScreen();
            Company.Summary.AcctCD.Select("THIRDCOMP");
            Company.ConfigurationSettings.RoleName.Select("Administrator");
            Company.Save();
            Company.MessageBox.Ok();
            VerifyCompanyVisibility(2, 2, "Second company", "Third company", "Third Cmp Branch 1", "Third Cmp Branch 2");

            Company.Summary.AcctCD.Select("SECONDCOMP");
            Company.ConfigurationSettings.RoleName.Reset();
            Company.Save();
            VerifyCompanyVisibility(1, 2, "Third company", "Third Cmp Branch 1", "Third Cmp Branch 2");
            PxLogin.LogIn("user1", Config.SITE_DST_PASSWORD);
            Company.OpenScreen();
            Company.PageHeader.Branch.IsVisible().VerifyEquals(false);
        }
        #endregion
    }
    #endregion

    #region TestCase 2 - Widgets
    using (TestExecution.CreateTestCaseGroup("Widgets"))
    {
        PxLogin.LoginToDestinationSite();

        #region Step 1 - Create dashboard
        using (TestExecution.CreateTestStepGroup("Create dashboard"))
        {
            Account.OpenScreen();
            Account.New();
            Account.Details.Row.AccountCD.Type("111");
            Account.Details.Row.AccountClassID.Select("AP");
            Account.New();
            Account.Details.Row.AccountCD.Type("222");
            Account.Details.Row.AccountClassID.Select("AR");
            Account.Save();

            SetupGl.OpenScreen();
            SetupGl.GeneralSettings.YtdNetIncAccountID.Select("111");
            SetupGl.GeneralSettings.RetEarnAccountID.Select("222");
            SetupGl.Save();

            SetupAp.OpenScreen();
            SetupAp.Save();

            SetupAr.OpenScreen();
            SetupAr.Save();

            SetupSo.OpenScreen();
            SetupSo.Save();

            Dashboards.OpenScreen();
            Dashboards.Insert();
            Dashboards.Summary.Name.Type("Board");
            Dashboards.Summary.Visible.SetTrue();
            Dashboards.Summary.WorkspaceID.Select("Data Views");
            Dashboards.Summary.SubcategoryID.Select("Dashboards");
            Dashboards.Summary.DefaultOwnerRole.Select("DashboardDesigner");
            Dashboards.Details.SetFalse();
            Dashboards.Details.SelectRow(Dashboards.Details.Columns.RoleName, "Administrator");
            Dashboards.Details.Row.RoleRight.Select("Granted");
            Dashboards.Details.SelectRow(Dashboards.Details.Columns.RoleName, "Customizer");
            Dashboards.Details.Row.RoleRight.Select("Granted");
            Dashboards.Save();
            Dashboards.ViewDashboard();
        }
        #endregion

        #region Step 2 - Create datatable widget
        using (TestExecution.CreateTestStepGroup("Create datatable widget"))
        {
            Dashboard.Design();
            Dashboard.Dashboard.DynamicControl<EmptyWidget>(1, 1).AddWidget();

            Dashboard.Dashboard.Wizard.AddWidget.DataTabledisplaysTheTableWithDataFromAParticularView();
            Dashboard.Dashboard.Wizard.AddWidget.Next();
            Dashboard.Dashboard.Wizard.AddPXTableWidget.InquiryScreenID.Select("Sales Orders");
            Dashboard.Dashboard.Wizard.AddPXTableWidget.EdCaption.Type("Sales Orders table");

            Dashboard.Dashboard.Wizard.AddWidget.Finish();
            Dashboard.Design();

            Dashboard.Dashboard.DynamicControl<DataTableWidget>(1, 1).Grid.ColumnsCount().VerifyIsGreaterThan(1);
            Dashboard.Dashboard.DynamicControl<DataTableWidget>(1, 1).Grid.RowsCount().VerifyEquals(0);
        }
        #endregion

        #region Step 3 - Create chart widget
        using (TestExecution.CreateTestStepGroup("Create chart widget"))
        {
            Dashboard.Design();
            Dashboard.Dashboard.DynamicControl<EmptyWidget>(1, 2).AddWidget();

            Dashboard.Dashboard.Wizard.AddWidget.ChartdisplaysTheChartUsingDataFromAParticularView();
            Dashboard.Dashboard.Wizard.AddWidget.Next();
            Dashboard.Dashboard.Wizard.AddPXChartWidget.InquiryScreenID.Select("Order Types");

            Dashboard.Dashboard.Wizard.AddPXChartWidget.Configure();
            Dashboard.Dashboard.Wizard.ChartSettings.ChartType.Select("Doughnut");
            Dashboard.Dashboard.Wizard.ChartSettings.CategoryField.Select("Description");
            Dashboard.Dashboard.Wizard.ChartSettings.ValueField.Select("Order Type");
            Dashboard.Dashboard.Wizard.ChartSettings.ValueAggregate.Select("Count All");

            Dashboard.Dashboard.Wizard.ChartSettings.Ok1();
            Dashboard.Dashboard.Wizard.AddPXChartWidget.EdCaption.Type("Order Types");
            Dashboard.Dashboard.Wizard.AddWidget.Finish();
            Dashboard.Design();

            Dashboard.Dashboard.DynamicControl<PieChartWidget>(1, 2).GetSlice(1).GetValue().VerifyEquals("Cash Sale: 10.00% (1.00)");
        }
        #endregion

        #region Step 4 - Create scored kpi widget
        using (TestExecution.CreateTestStepGroup("Create scored kpi widget"))
        {
            Dashboard.Design();
            Dashboard.Dashboard.DynamicControl<EmptyWidget>(2, 1).AddWidget();
```

```
            Dashboard.Dashboard.Wizard.AddWidget.KeyPerformanceIndicatorKpiDisplaysAKpiAsAMeterOrScorecard();
            Dashboard.Dashboard.Wizard.AddWidget.Next();
            Dashboard.Dashboard.Wizard.AddKPIScoreWidget.InquiryScreenID.Select("Order Types");
            Dashboard.Dashboard.Wizard.AddKPIScoreWidget.AggregateField.Select("Order Type");
            Dashboard.Dashboard.Wizard.AddKPIScoreWidget.EdCaption.Type("Order Types KPI");

            Dashboard.Dashboard.Wizard.AddWidget.Finish();
            Dashboard.Dashboard.DynamicControl<CardKpiWidget>(2, 1).Edit.Click();
            Dashboard.Dashboard.Wizard.AddKPIScoreWidget.InquiryScreenID.Select("Order Types");
            Dashboard.Dashboard.Wizard.AddKPIScoreWidget.AggregateField.Select("Order Type");
            Dashboard.Dashboard.Wizard.AddWidget.Finish();
            Dashboard.Design();

            Dashboard.Dashboard.DynamicControl<CardKpiWidget>(2, 1).GetKpiStatus().VerifyEquals(CardKpiWidget.KpiStatuses.Error);
        }
        #endregion
    }
    #endregion
}

internal static void VerifyCompanyVisibility(int companiesCount, int branchesCount, params string[] companies)
{
    var wrapper = new Wrapper();
    wrapper.PageHeader.Branch.Click();
    wrapper.PageHeader.Branch.GetValues().VerifyEachOfValuesEquals(companies);
    wrapper.PageHeader.Branch.Status.GetValue().VerifyEquals($"Companies: {companiesCount}, Branches: {branchesCount}");
    wrapper.PageHeader.Branch.Click();
```

```
        }
    }
}
```

Running the Test

1. Modify Config.xml located in Test SDK package as follows:

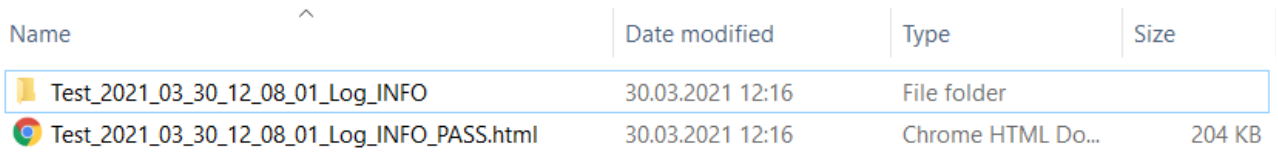
```
<?xml version="1.0" encoding="utf-8"?>
<config>
 <general>
      <browserbin>path to chrome.exe file from Test SDK package</browserbin>
      <browser_downloads_folder>c:\share\download\</browser_downloads_folder>
      <site_dst>
         <url>http://yourhost/yoursite</url>
         <login>admin</login>
         <pswd>123</pswd>
      </site_dst>
      <logging>
         <logStorage type="htmlfile" level="INFO" outputFolder="path to the folder where the log files will be located" screenshotActive="true" />
      </logging>
   </general>
   <testing>
      <Check Name="Test"/>
   </testing>
</config>
```

2. Right-click on Tests project in Tests solution and select Properties option
3. In Debug => Application arguments type arguments as follows:

   - /config "path to Config.xml file from Test SDK package" (example: /config "C:\TestSDK\Config.xml")

```
Command line arguments example to start your test:
.\Tests.exe /config "path to Config.xml file from Test SDK package"
```

4. Press F5 to start the test.
5. If you have done everything correctly the following log files will appear the outputFolder folder specified in Config.xml after test completes:



- Check: Test
    + Operation: Login to site: https://localhost/site/
    + Testcase 1: Operationg with few branches/companies
    - Testcase 2: Widgets
        + Operation: Login to site: https://localhost/site/
        + Step 1: Create dashboard
        + Step 2: Create datatable widget
        + Step 3: Create chart widget
        - Step 4: Create scored kpi widget
            + Operation: Click toolbar button: Design
            - Operation: Click link: Add a New Widget
            + Operation: Click toolbar button: Scorecard KPIDisplays the total of the selected column from the inquiry as a scorecard.
            + Operation: Click button: Next
            + Operation: Select in the selector: Inquiry Screen, value: Order Types
            + Operation: Select in the drop-down: Field to Aggregate, values: Order Type
            + Operation: Type into input: Caption, value: Order Types KPI
            + Operation: Click button: Finish
            + Operation: Click toolbar button: Edit
            + Operation: Select in the selector: Inquiry Screen, value: Order Types
            + Operation: Select in the drop-down: Field to Aggregate, values: Order Type
            + Operation: Click button: Finish
            - Operation: Click toolbar button: Design

            - Operation: Verify the "Kpi Status" property of the control "CardKpiWidget" equals: Error

Part 2. Acumatica Test SDK—API Reference

In This Part

- PX.QA.Tools.dll
- Core.dll
- ClassGenerator.exe

PX.QA.Tools.dll

- Check
- Config
- Log

Check

Every test created using Test SDK must have a Check class as a parent - this allows to use the following approach for every test with the following steps in the sequence:

1. You can do some preparation before running your test - for example you can login into the system first, restore a company snapshot or restore a database backup if required.
2. Test whatever is required to be tested by your test.
3. Finalize the test - for example you can sigh out from the system, stop browser, create company snapshot or create a database backup if required.

The following public methods are available for every test:

- BeforeExecute()
- Execute()
- AfterExecute()

BeforeExecute()

Use if something must be done before the test starts, e.g. site configuration can be placed here.

```
namespace Tests
{
    public class Test : Check
    {
        public override void BeforeExecute()
        {
   //Place configuration logic of the test here.
  }

        public override void Execute()
        {
        }

        public override void AfterExecute()
        {
        }
    }
}
```

Execute()

```
Place main logic of the test here.
```

```
namespace Tests
{
    public class Test : Check
    {
        public override void BeforeExecute()
        {
  }

        public override void Execute()
        {
   //Place main logic of the test here.
        }

        public override void AfterExecute()
        {
        }
    }
}
```

AfterExecute()

```
Use if something must be done after the test ends.
```

```
namespace Tests
{
    public class Test : Check
    {
        public override void BeforeExecute()
        {
  }

        public override void Execute()
        {
        }

        public override void AfterExecute()
        {
   //Place logic of the test here.
        }
    }
}
```

Config

- Example 1
- Example 2
- Node Specification

Config (Config.xml) is the configuration file that includes all required settings and options Test SDK users have to provide for every test, such as URL of the AUT, access credentials, web browser, logs format etc.

Example 1

```
<?xml version="1.0" encoding="utf-8"?>
<config>
    <general>
        <browserbin>C:\TestSDK\Chrome\chrome.exe</browserbin>
  <site_dst>
            <url>http://host/aut/</url>
            <login>username</login>
            <pswd>password</pswd>
        </site_dst>
        <logging>
        <logStorage type="htmlfile" level="INFO" outputFolder="output folder for log files" screenshotActive="true" />
        </logging>
   </general>
   <testing/>
</config>
```

Example 2

```
<?xml version="1.0" encoding="utf-8"?>
<config>
 <general>
      <browserbin>path to browser.exe</browserbin>
            <browser_downloads_folder>c:\share\download\</browser_downloads_folder>
      <default_culture_info>en-US</default_culture_info>
      <long_timeout_ms>global timeout in milliseconds</long_timeout_ms>
      <site_dst>
    <url>http://host/aut/</url>
    <login>username</login>
    <pswd>password</pswd>
      </site_dst>
      <site_prt>
    <url>http://host/portal/</url>
    <login>username</login>
    <pswd>password</pswd>
      </site_prt>
      <logging>
          <logStorage type ="htmlfile" level="INFO" outputFolder ="output folder for log files" screenshotActive="true" />
   </logging>
      <backup_restore Active="true" dbProvider="sql_server_name">
    <backup_folder>output folder for backups</backup_folder>
      </backup_restore>
 </general>
   <testing>
      <Check Name="Test name 1"/>
      <Check Name="Test name 2"/>
      <Check Name="..."/>
      <Check Name="Test name N"/>
   </testing>
</config>
```

Node Specification

Nodes	Parents	Attributes	Options	Description
`<?xml version="1.0" encoding="utf-8"?>`	-	-	-	Specifies the XML version and the encoding that should be used if the encoding is different from ASCII.
`<config/>`	-	-	-	Root node for all other tags.
`<general/>`	`<config/>`	-	-	Contains general configuration settings for your test.
`<browserbin/>`	`<general/>`	-	-	Specifies the path to the browser application launcher (required for Chrome).
`<browser_downloads_folder/>`	`<general/>`	-	-	Specifies the browser's downloads folder
<browserheadless/>	`<general/>`	-	`true/false`	Specifies whether to run the browser in headless mode

<default_culture_info/>	<general/>	-	-	Specifies the culture to be used in your test. The English (United States) culture (en-US) is used if the tag is not specified.
<long_timeout_ms/>	<general/>	-	-	Specifies the global timeout in milliseconds
<site_dst/>	<general/>	-	-	Contains information about the destination Acumatica ERP instance used in your test.
<site_prt/>	<general/>	-	-	Contains information about the Partner Portal instance used in your test.
<url/>	<site_dst/>	-	-	Specifies the URL of the Acumatica ERP instance to be used in your test.
	<site_prt/>			
<login/>	<site_dst/>	-	-	Specifies the login of the Acumatica ERP instance to be used in your test.
	<site_prt/>			
<pswd/>	<site_dst/>	-	-	Specifies the password of the Acumatica ERP instance to be used in your test.
	<site_prt/>			
<logging/>	<general/>	-	-	Contains log settings.
<logStorage/>	<logging/>	type	htmlfile	Specifies the log storage setting. Log storage settings are defined in the How to Configure Logging and Define Screenshot Settings topic.
		level	OFF	
			ERROR	
			WARN	
			INFO	
			DEBUG	
		outputFolder	-	
		screenshotActive	true/false	
<backup_restore/>	<general/>	Active	true/false	
		dbProvider	-	
<backup_folder/>	<backup_restore/>	-	-	
<testing/>	<config/>	-	-	Contains the list of your tests.
<check/>	<testing/>	Name	-	Specifies the name of your test.

Log

The test log can include the entries shown in the following table.

	Type	Description	Example
1	`Testcase`	A group of messages of a single test case. Contains the name of the test case.	`Testcase: Case 1`
2	`Step`	A group of messages of a single test step. Contains the name of the test step.	`Step: Step 1`
3	`Debug`	A debug message. Can include a nested `Screenshot` message.	`Debug: The page is not fully loaded yet`
4	`Screenshot`	A reference to a screenshot that has been saved to a file on your computer.	`Screenshot: c:\pic\localhost_27_05_2015_17_22_43_027.jpg`

5	`Infomation`	An informational message, which contains information that makes the log easier to read.	`Information: Verify error successful. Error message:`
			
			`Error #13: Inserting 'GL Batch' record raised one or more errors. Please review.`

6	`Warning`	A warning message. Can include a nested `Screenshot` message.	`Warning: Cannot show Acumatica Trace`
7	`Error`	An error message. Can include a nested `Screenshot` message.	`Error: Modal dialog present`

You can find more information on log files in How to Manage Log Providers.

Core.dll

- Browser
- Button
- CheckBox
- DateSelector
- DropDown
- Selector
- TreeSelector
- Grid
- GroupBox
- ImageUploader
- Input
- Label
- RichTextEdit
- ToolBarButton
- TreeView

Browser

To begin preparing your test environment, you need to make sure that you have the suitable versions of WebDriver and your browser. Please use dlls and browsers provided with the TestSDK package

Preparing for Testing in Chrome

Specify the browser-dependent settings in the Config.xml file, which is available in your Acumatica Test SDK folder, as described below.

- `<browserbin>path to *chrome.exe*</browserbin>`

Preparing for Testing in Firefox

Specify the browser-dependent settings in the Config.xml file, which is available in your Acumatica Test SDK folder, as described below.

- `<browserbin>path to *firefox.exe*</browserbin>`

Button

- UI

How to use Button

UI

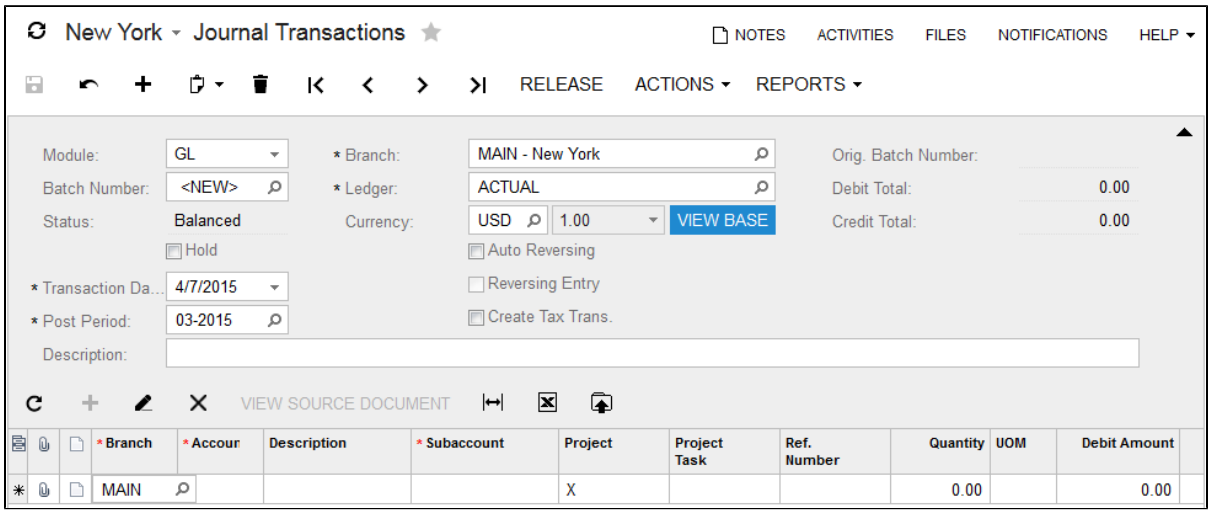


Figure: Button VIEW BASE on the Journal Transactions page

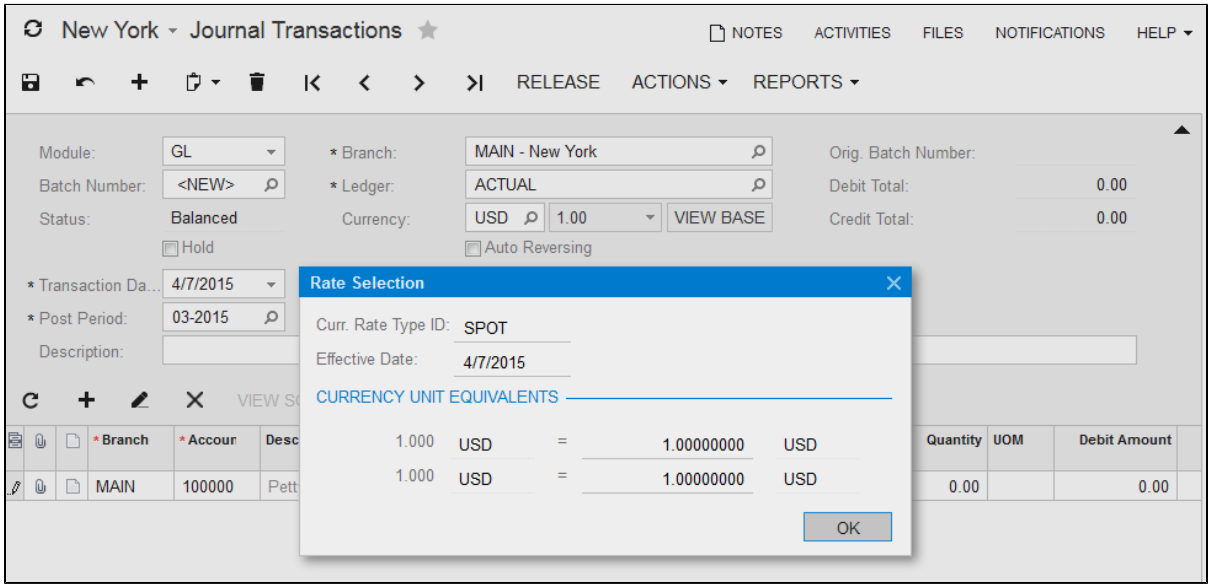


Figure: Button OK in the Rate Selection dialog

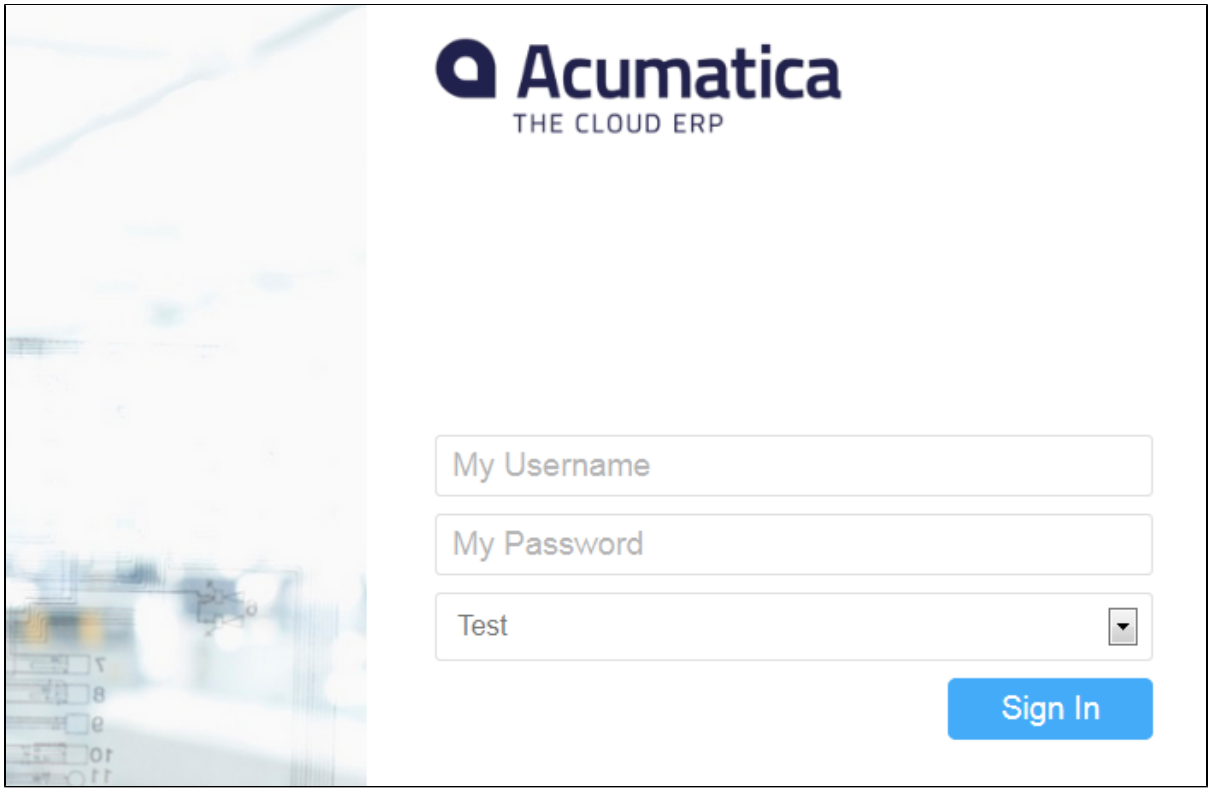


Figure: Button Sign In on the login page

How to use Button

```
Branch Branch = new Branch();
Branch.OpenScreen();
Branch.GeneralInfo_MainAddress.Buttons.ViewonMap.Click();
```

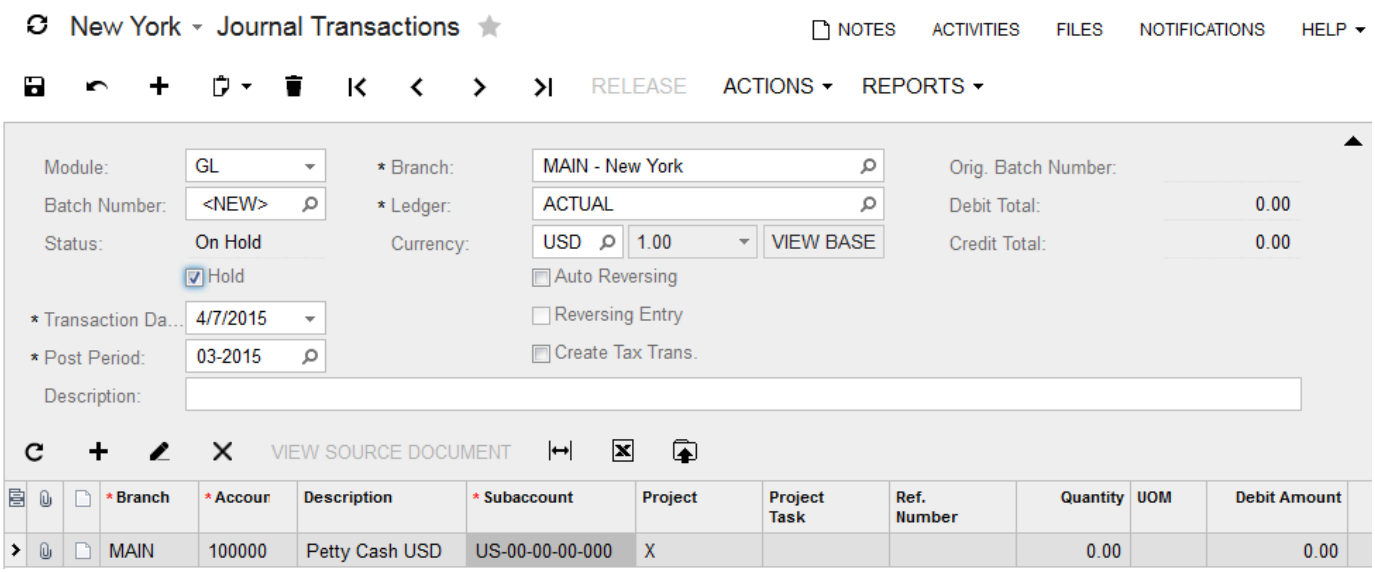
```
Branch Branch = new Branch();
Branch.OpenScreen();

if (Branch.GeneralInfo_MainAddress.Buttons.ViewonMap.IsEnabled())
{
 Branch.GeneralInfo_MainAddress.Buttons.ViewonMap.Click();
}
```

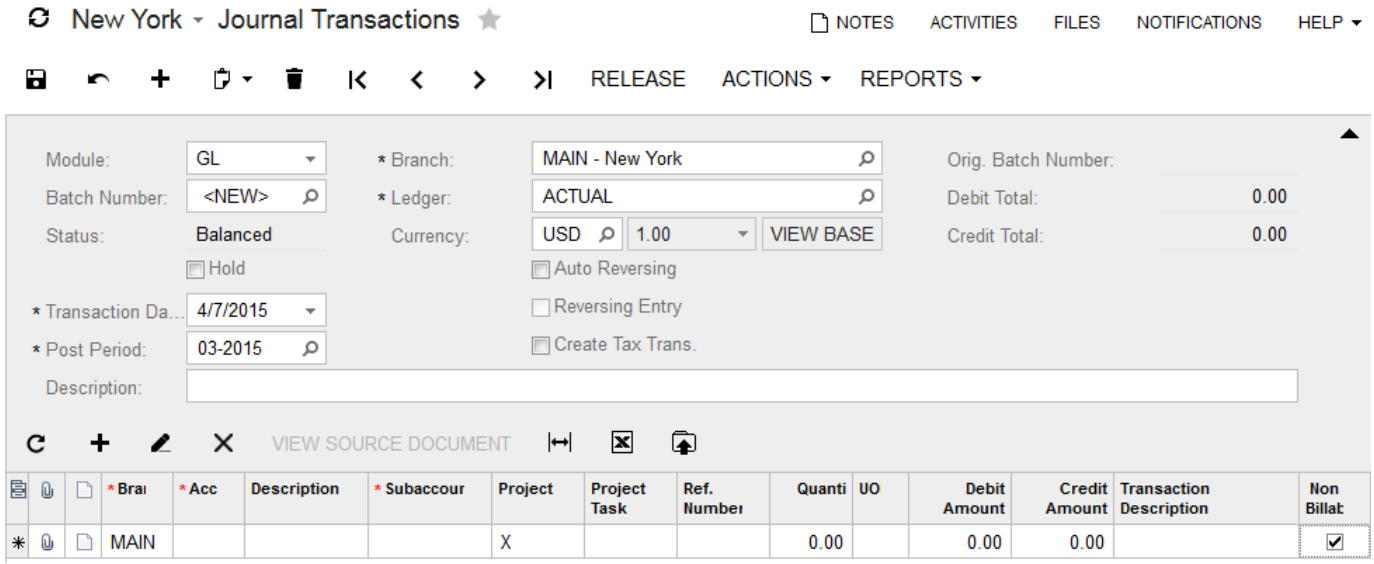
CheckBox

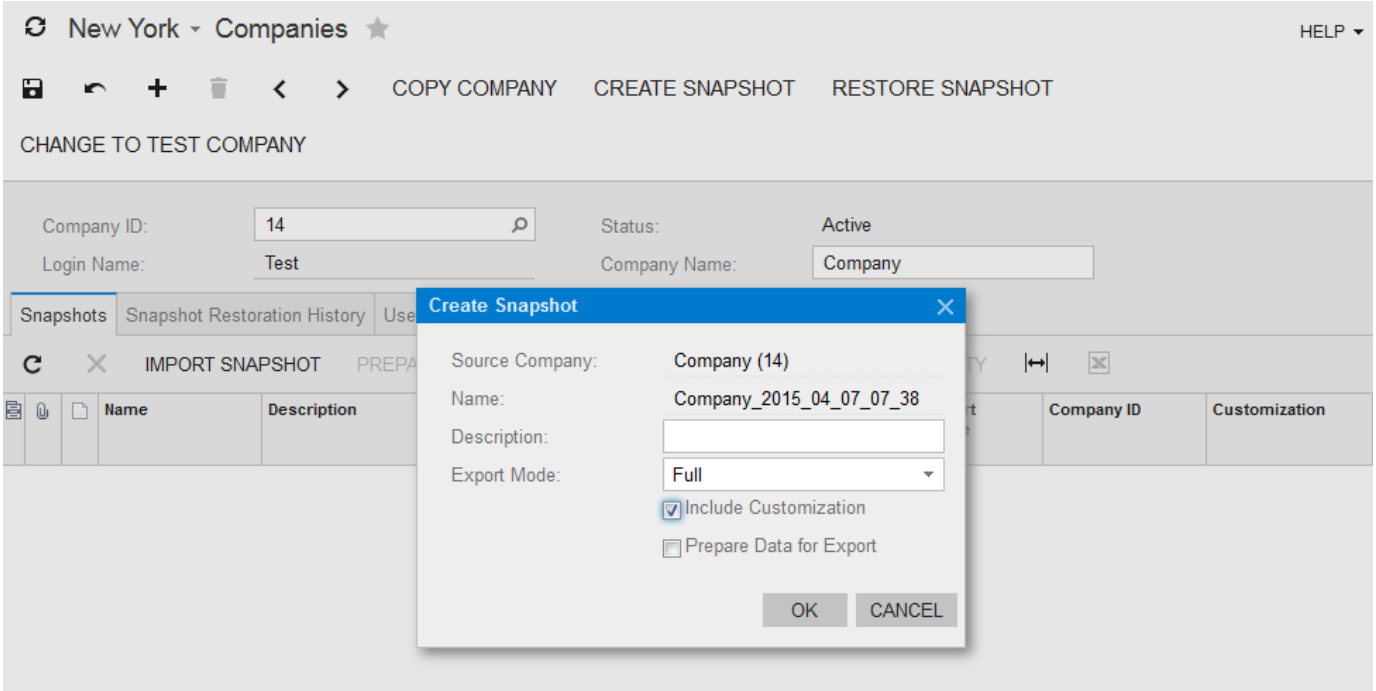
Description
Checkbox allows you to set one of the available boolean values: true, false.

- How a checkbox looks like in user interface
- How to use a checkbox

How a checkbox looks like in user interface







How to use a checkbox

The following code fragments set journal transactions batch on hold on "Journal Transactions" screen:

```
BatchBatch.Summary.Hold.Set(true);
```

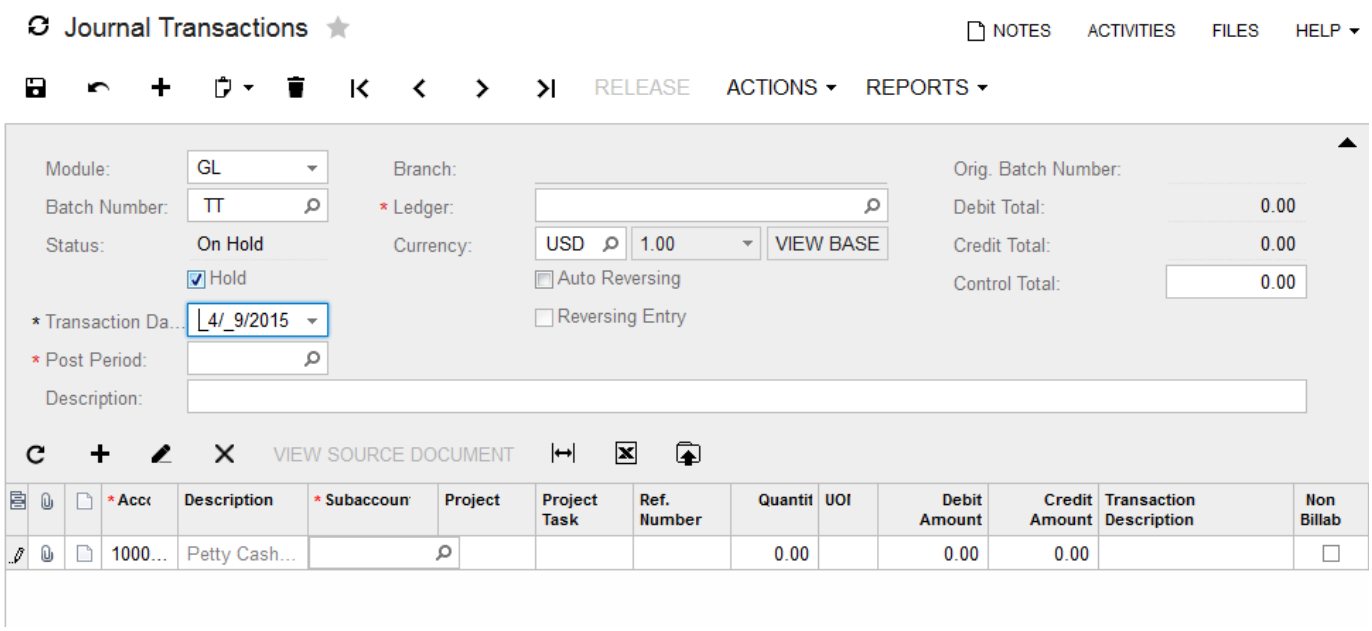
```
Batch.Summary.Hold.SetTrue();
```

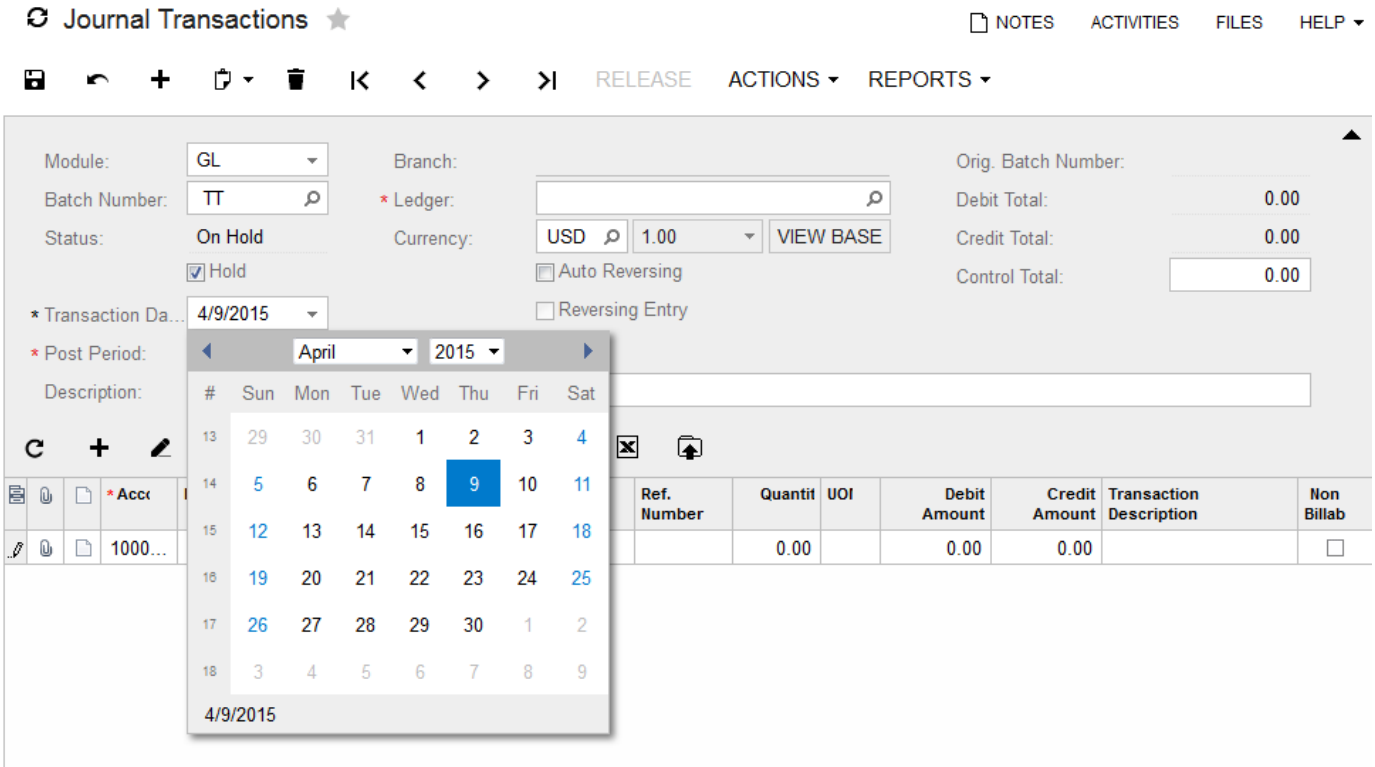
DateSelector

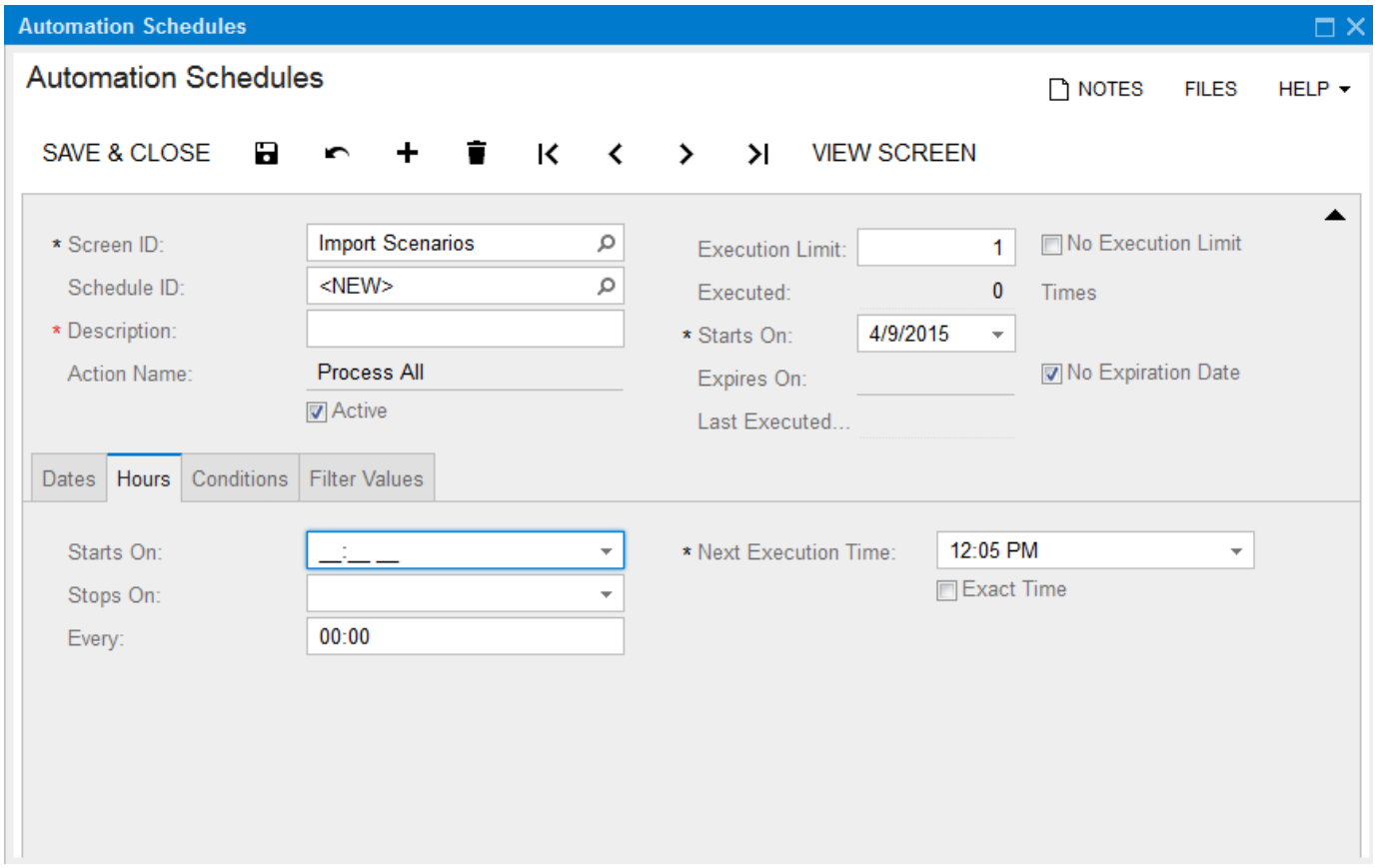
Description
Date-selector allows you to type or select string values in formats: Date (1900/1/1), Time (12:00 AM), Duration (00:10), Text (value is typed as is, no additional formatting applied).

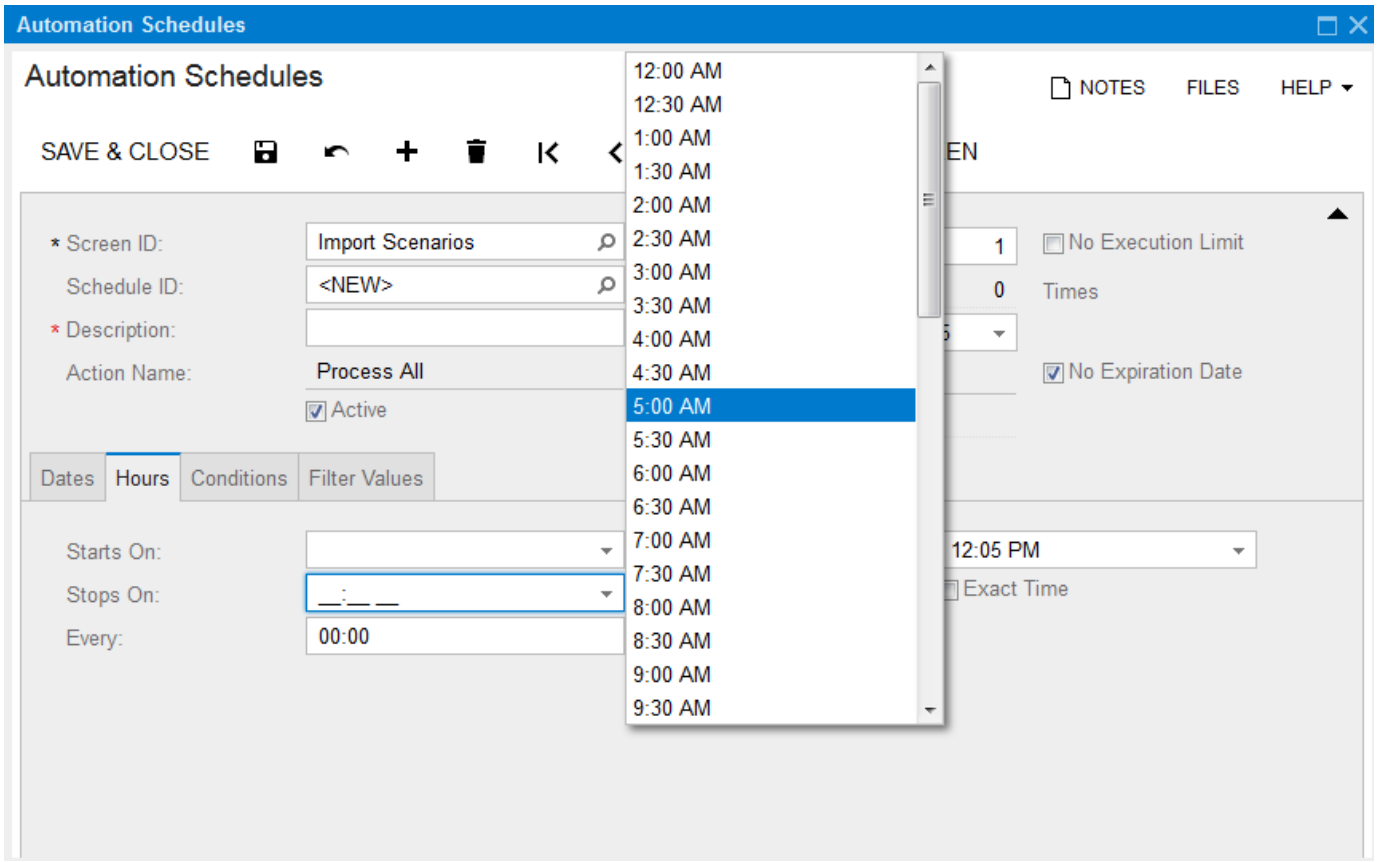
- How date-selector looks like in user interface
- How to use date-selector

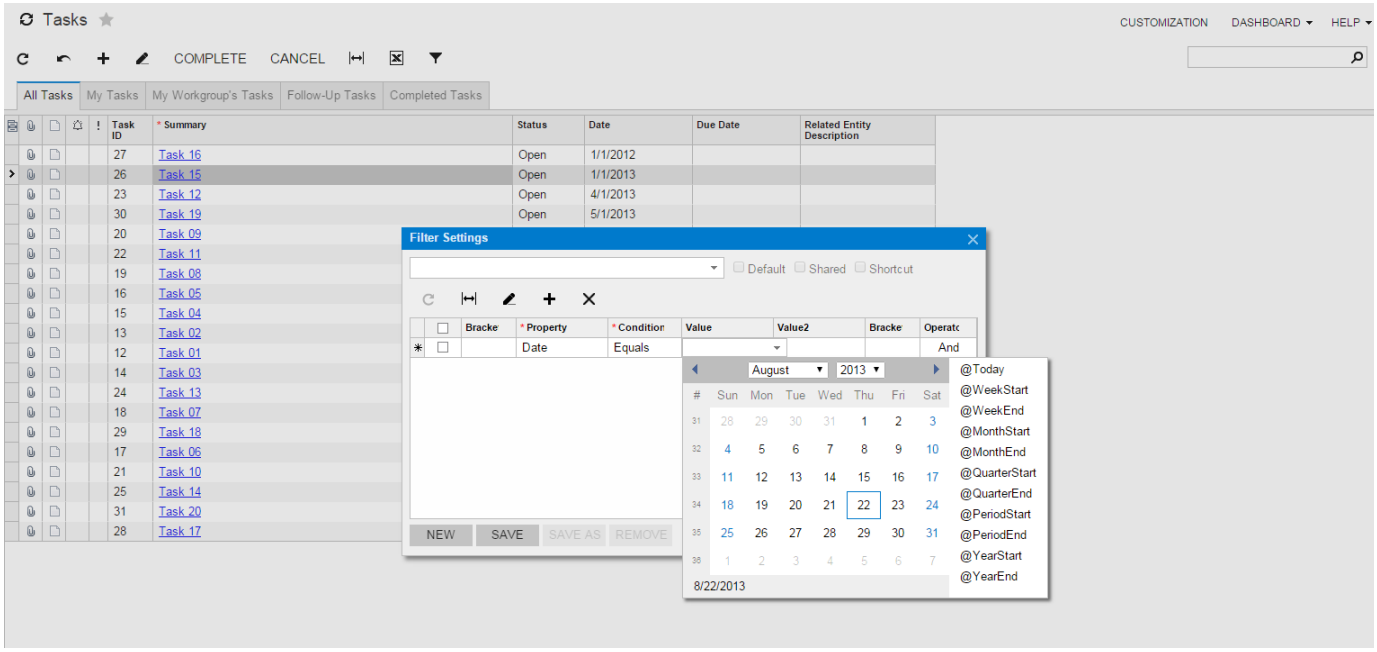
How date-selector looks like in user interface






Automation Schedules
Automation Schedules
NOTES
FILES
HELP
SAVE & CLOSE
VIEW SCREEN
* Screen ID:
Import Scenarios
Schedule ID:
<NEW>
* Description:
Action Name:
Process All
Active
Execution Limit:
1
No Execution Limit
Executed:
0
Times
* Starts On:
4/9/2015
Expires On:
No Expiration Date
Last Executed...
Dates
Hours
Conditions
Filter Values
Starts On:
__:__ __
Stops On:
Every:
00:00
* Next Execution Time:
12:05 PM
Exact Time


Automation Schedules
Automation Schedules
NOTES
FILES
HELP
SAVE & CLOSE
* Screen ID:
Import Scenarios
Schedule ID:
<NEW>
* Description:
Action Name:
Process All
Active
No Execution Limit
0
Times
No Expiration Date
Dates
Hours
Conditions
Filter Values
Starts On:
Stops On:
__:__ __
Every:
00:00
12:05 PM
Exact Time
12:00 AM
12:30 AM
1:00 AM
1:30 AM
2:00 AM
2:30 AM
3:00 AM
3:30 AM
4:00 AM
4:30 AM
5:00 AM
5:30 AM
6:00 AM
6:30 AM
7:00 AM
7:30 AM
8:00 AM
8:30 AM
9:00 AM
9:30 AM



How to use date-selector

The code examples below show how to type "Transaction Date" on "Journal Transactions" screen:

```
JournalEntry batch = new JournalEntry();
batch.Summary.DateEntered.Type(new DateTime(1900, 12, 31));
```

```
JournalEntry batch = new JournalEntry();
batch.Summary.DateEntered.Type("12/31/1900", DateSelector.Options.Text);
```

The code example below show how to select "Start Time" and "End Time" for schedule on "Automation Schedules" screen:

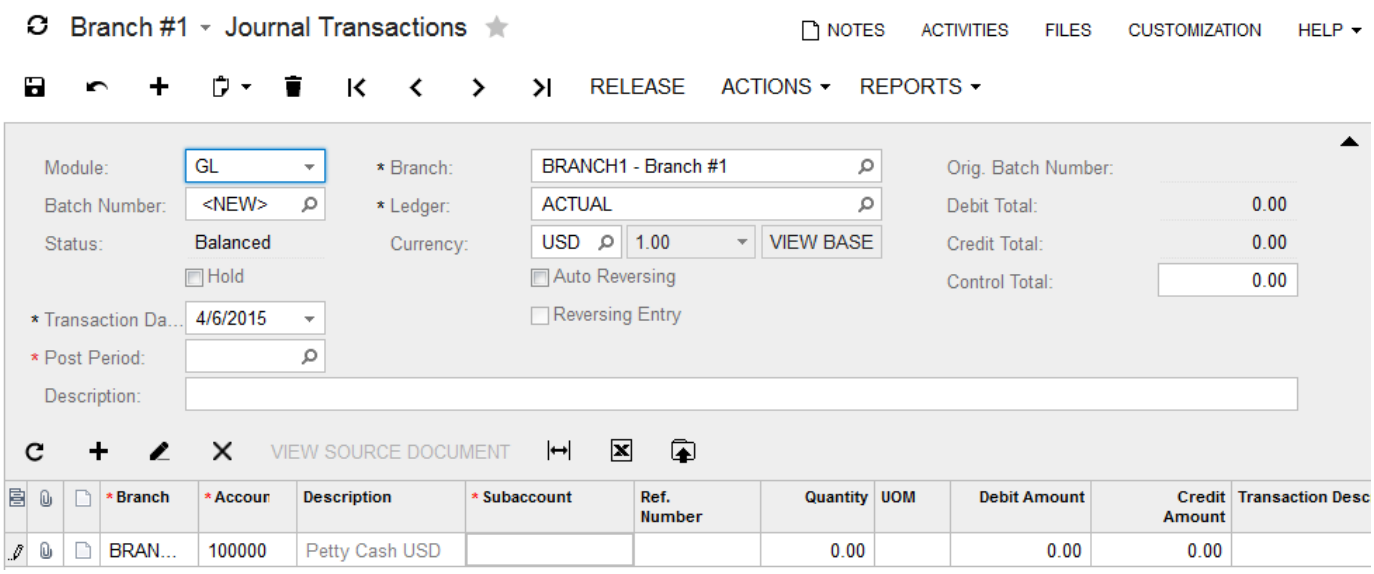
```
ProcessImportScenario processImportScenario = new ProcessImportScenario();
AutomationSchedule automationSchedule = new AutomationSchedule();
processImportScenario.OpenScreen();
processImportScenario.ScheduleAdd();
automationSchedule.Summary.Description.Type("Test");
automationSchedule.DatesHours.StartTime.Type(0, 0, DateSelector.Options.Time);
automationSchedule.DatesHours.EndTime.Type(23, 59, DateSelector.Options.Time);
automationSchedule.Save();
```

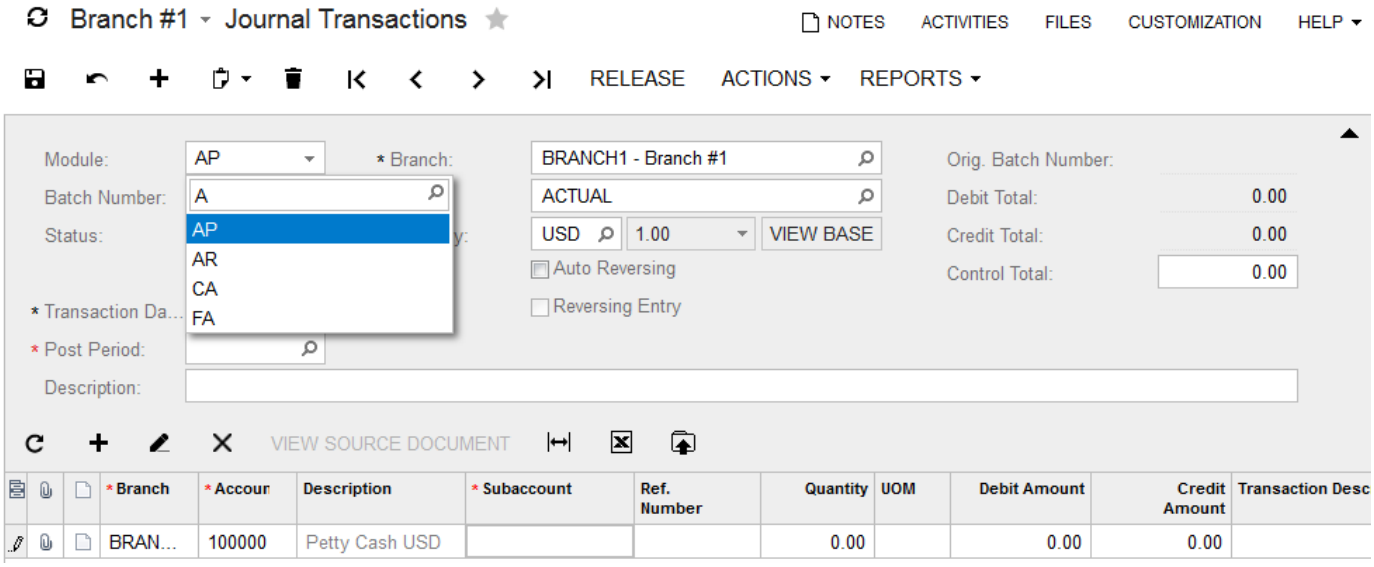
DropDown

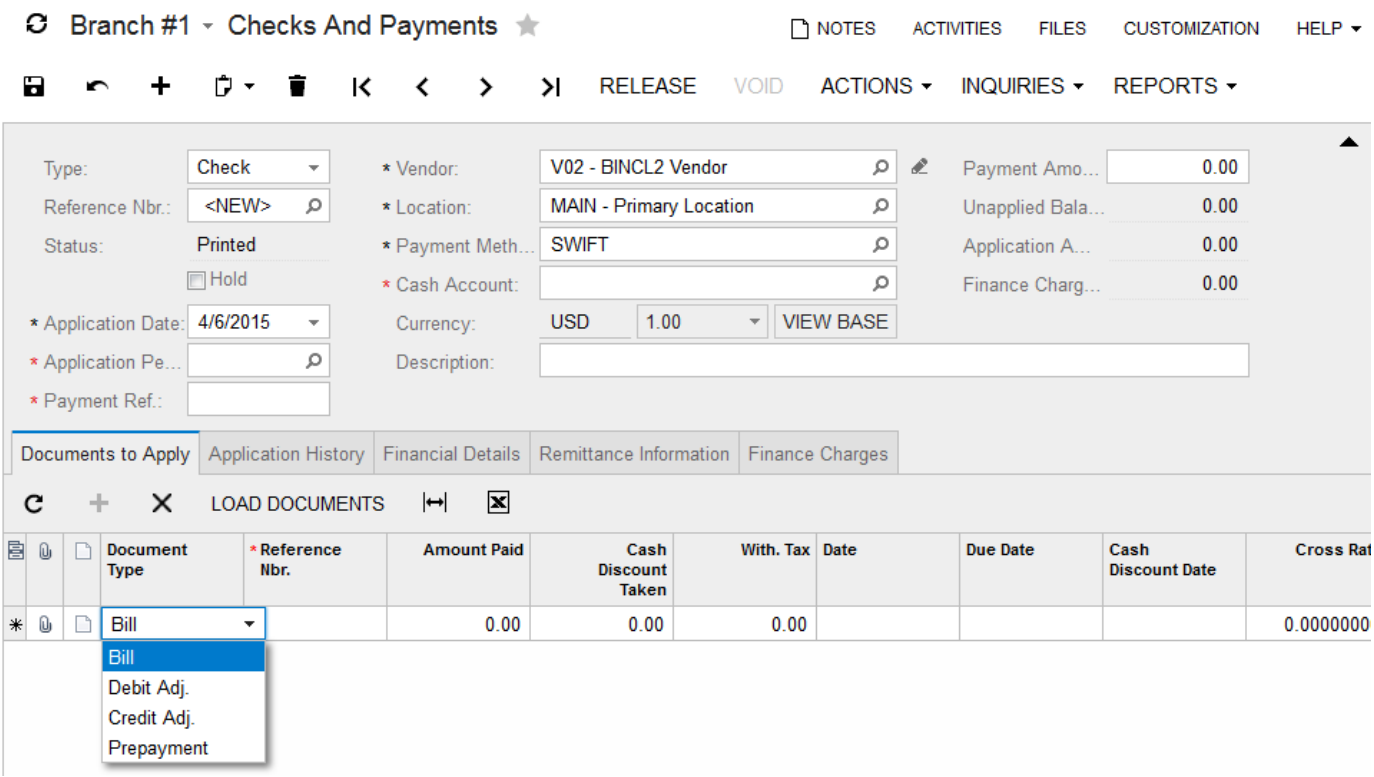
> Description
> Drop-down allows you to select one of the options availbale in the drop-down list.

- How drop-down looks like in user interface
- How to use drop-down

How drop-down looks like in user interface







How to use drop-down

The following code fragments select value "GL" in the drop-down "Module" on "Journal Transactions" screen:

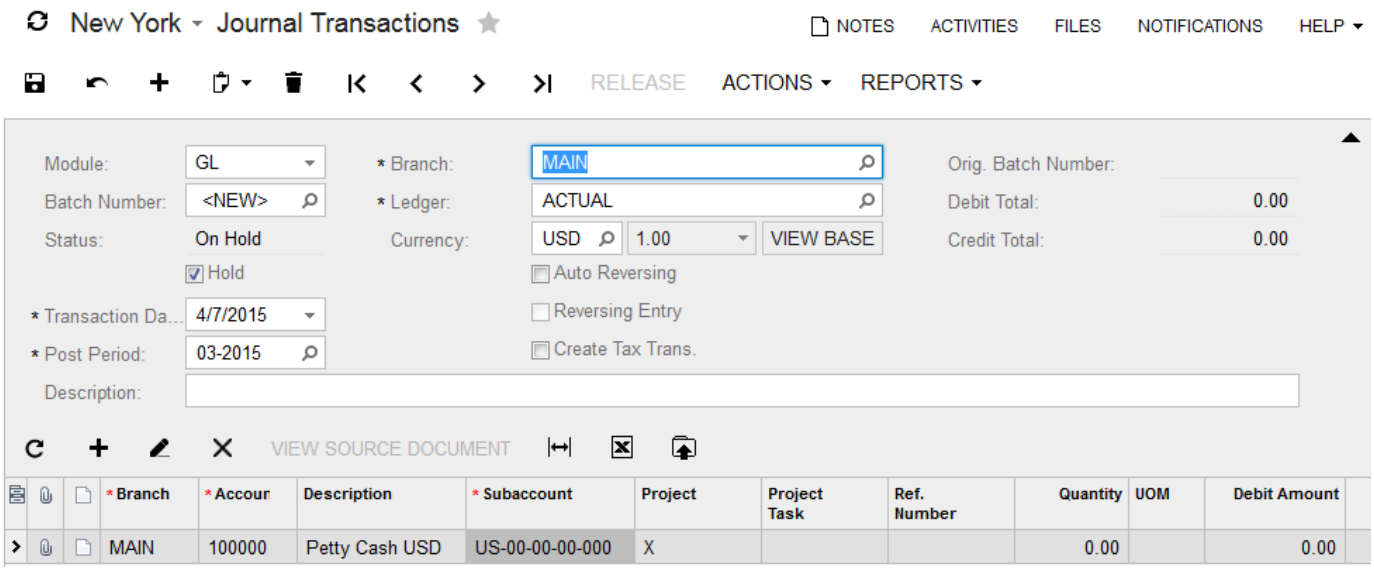
```
Batch.Summary.Module.Select("GL");
```

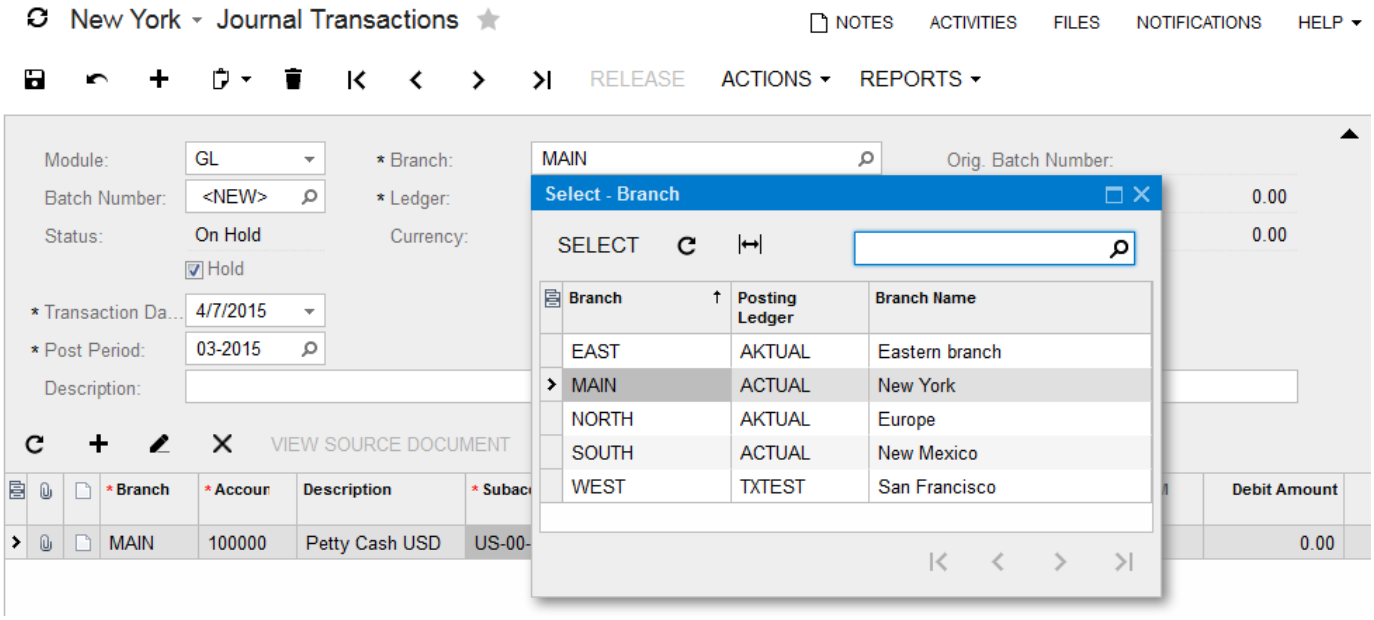
Selector

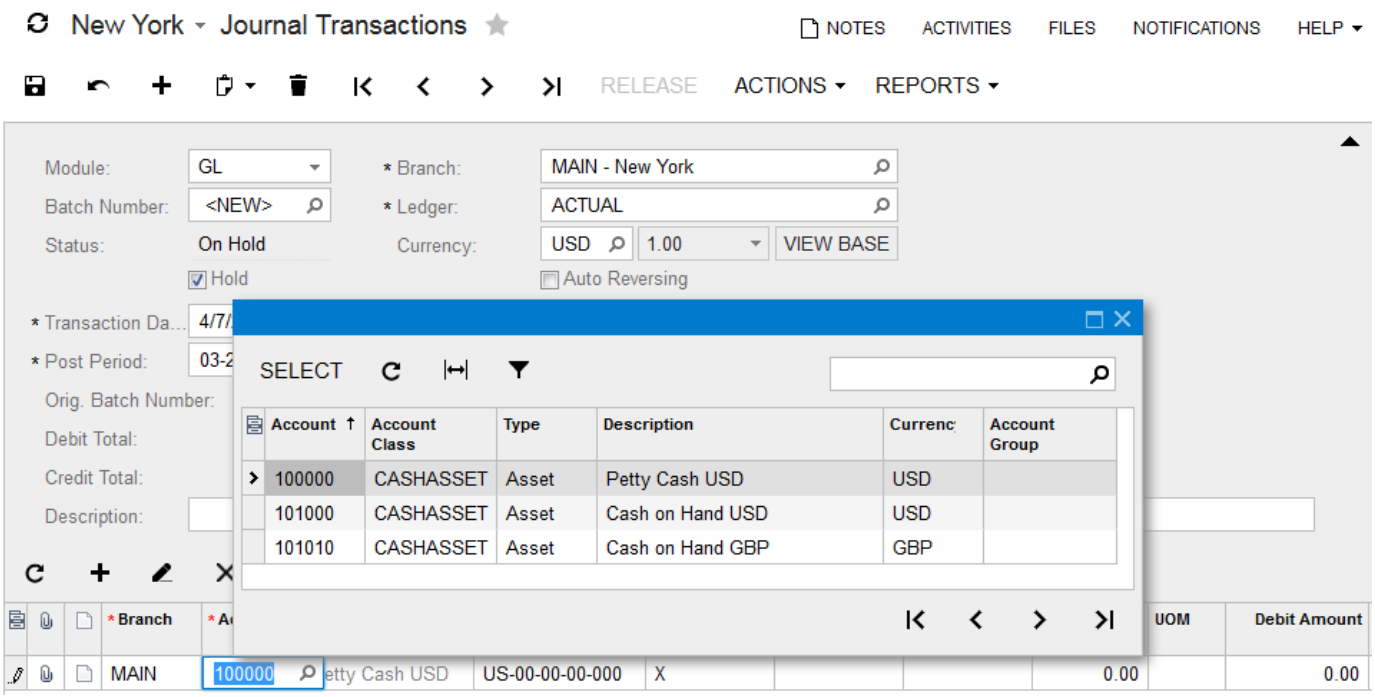
Description
Selector allows to select single record in pop-up table using differnt values available in table columns.

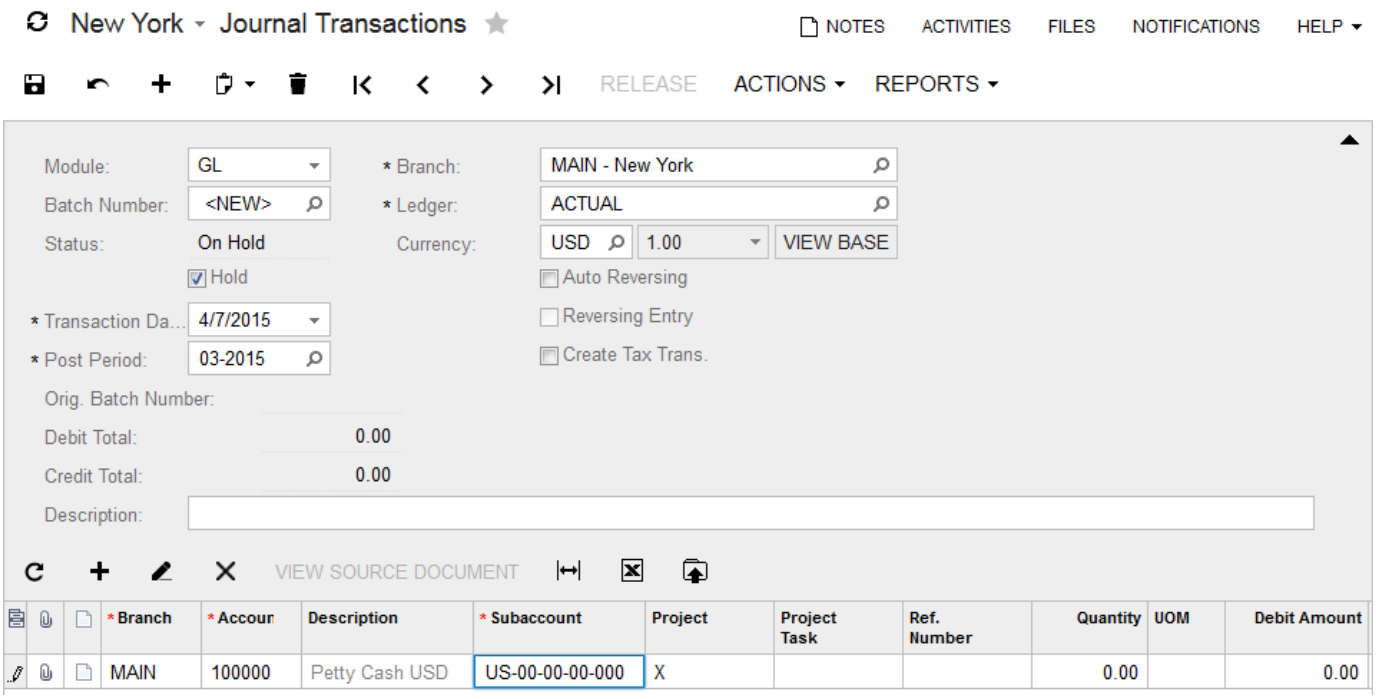
- How Selector Looks Like in User Interface
- How to Use Selector

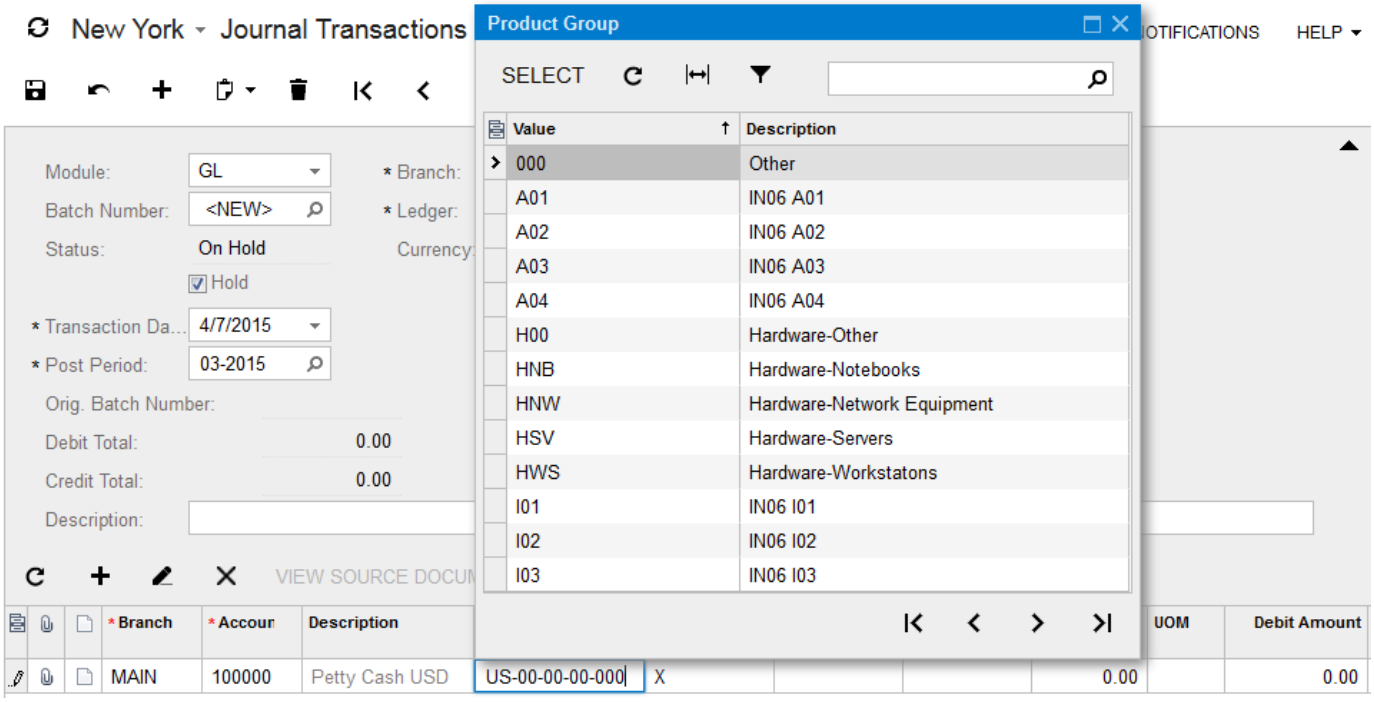
How Selector Looks Like in User Interface






New York
Journal Transactions
NOTES
ACTIVITIES
FILES
NOTIFICATIONS
HELP
RELEASE
ACTIONS
REPORTS
Module: GL
Batch Number: <NEW>
Status: On Hold
Hold
* Branch: MAIN - New York
* Ledger: ACTUAL
Currency: USD 1.00 VIEW BASE
Auto Reversing
* Transaction Da...
* Post Period:
Orig. Batch Number:
Debit Total:
Credit Total:
Description:
SELECT
Account
Account Class
Type
Description
Currenc
Account Group
100000 CASHASSET Asset Petty Cash USD USD
101000 CASHASSET Asset Cash on Hand USD USD
101010 CASHASSET Asset Cash on Hand GBP GBP
* Branch
UOM
Debit Amount
MAIN
100000
US-00-00-00-000
0.00
0.00


New York
Journal Transactions
NOTES
ACTIVITIES
FILES
NOTIFICATIONS
HELP
RELEASE
ACTIONS
REPORTS
Module: GL
Batch Number: <NEW>
Status: On Hold
Hold
* Branch: MAIN - New York
* Ledger: ACTUAL
Currency: USD 1.00 VIEW BASE
Auto Reversing
Reversing Entry
Create Tax Trans.
* Transaction Da... 4/7/2015
* Post Period: 03-2015
Orig. Batch Number:
Debit Total: 0.00
Credit Total: 0.00
Description:
VIEW SOURCE DOCUMENT
* Branch
* Accour
Description
* Subaccount
Project
Project Task
Ref. Number
Quantity
UOM
Debit Amount
MAIN
100000
Petty Cash USD
US-00-00-00-000
X
0.00
0.00



How to Use Selector

To select a value that is present in any column/row of the selector use:

```
User.OpenScreen();
User.Summary.LoginTypeID.Select("Employee"); //Specify value here
```

To select a value that is present in a specific column of the selector use:

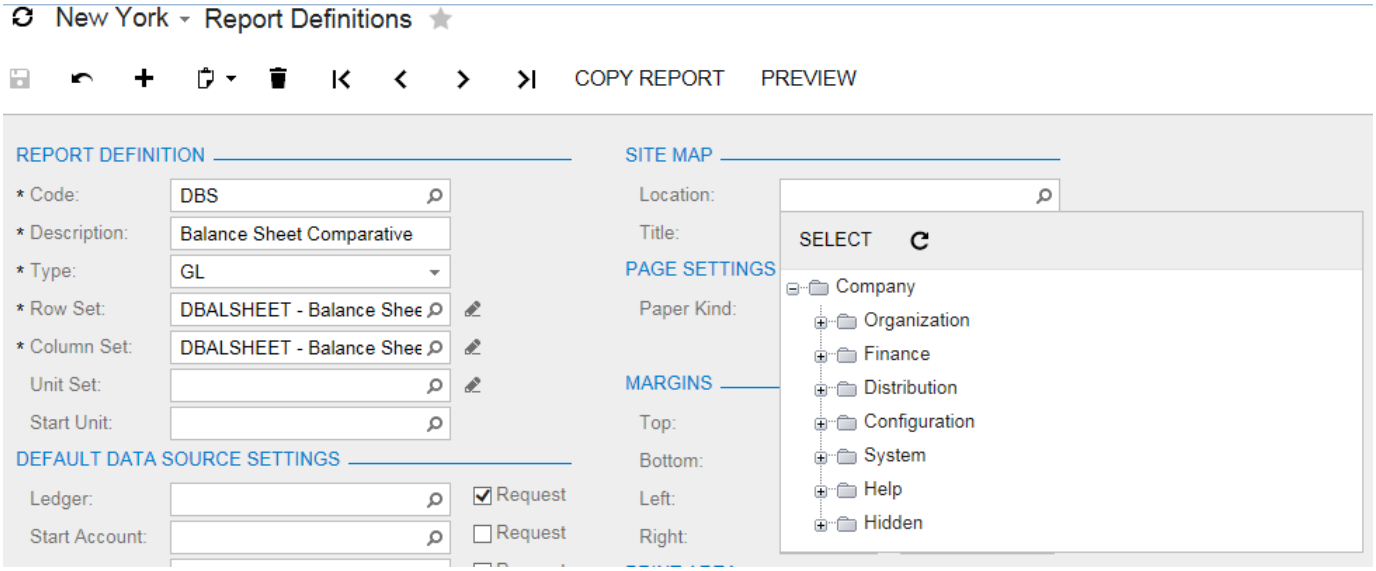
```
User.OpenScreen();
User.Summary.LoginTypeID.Select("Employee", "User Type"); //Specify value and column name here
```

TreeSelector

Description
Selector control with tree in popup panel

- How Tree-Selector Looks Like in User Interface
- How to Use Tree-Selector

How Tree-Selector Looks Like in User Interface



How to Use Tree-Selector

This control have several select options. Most commonly used option is "Path".

With this option, framework will select value in tree like real user.

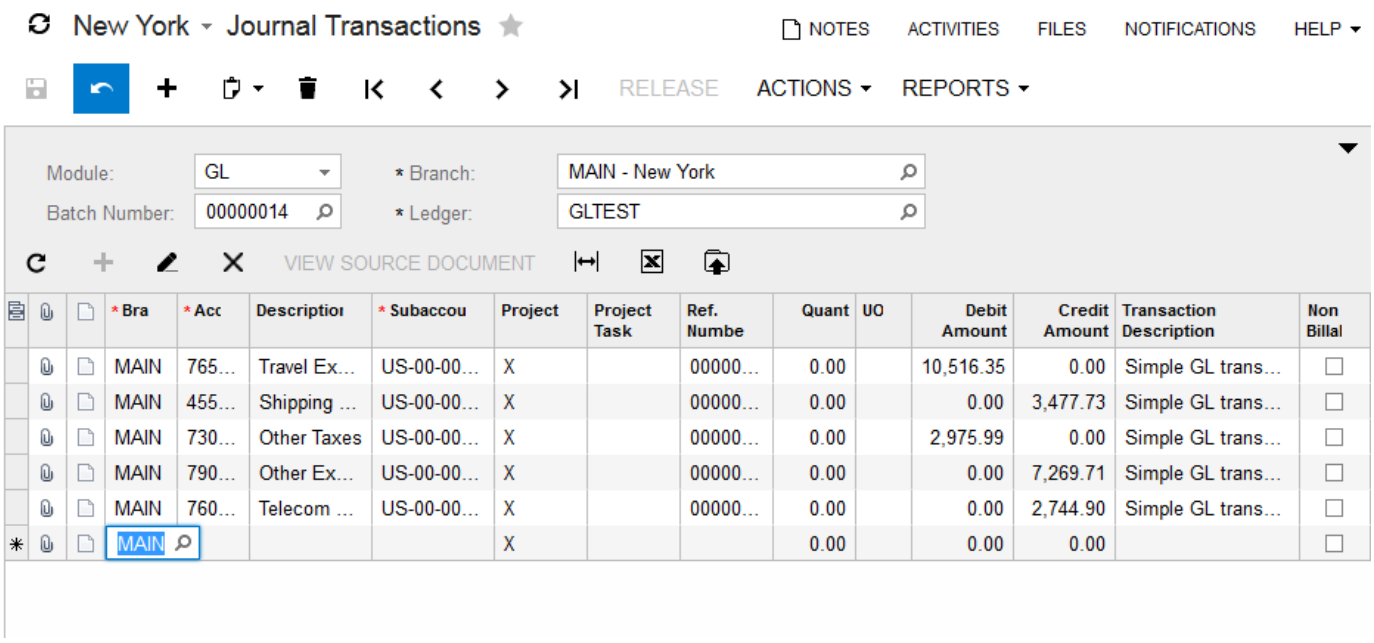
```
// Select sitemap location
AutomationNotification.Summary.ScreenID.Select("Company/Organization/Customer Management/Work Area/Enter/Leads");
```

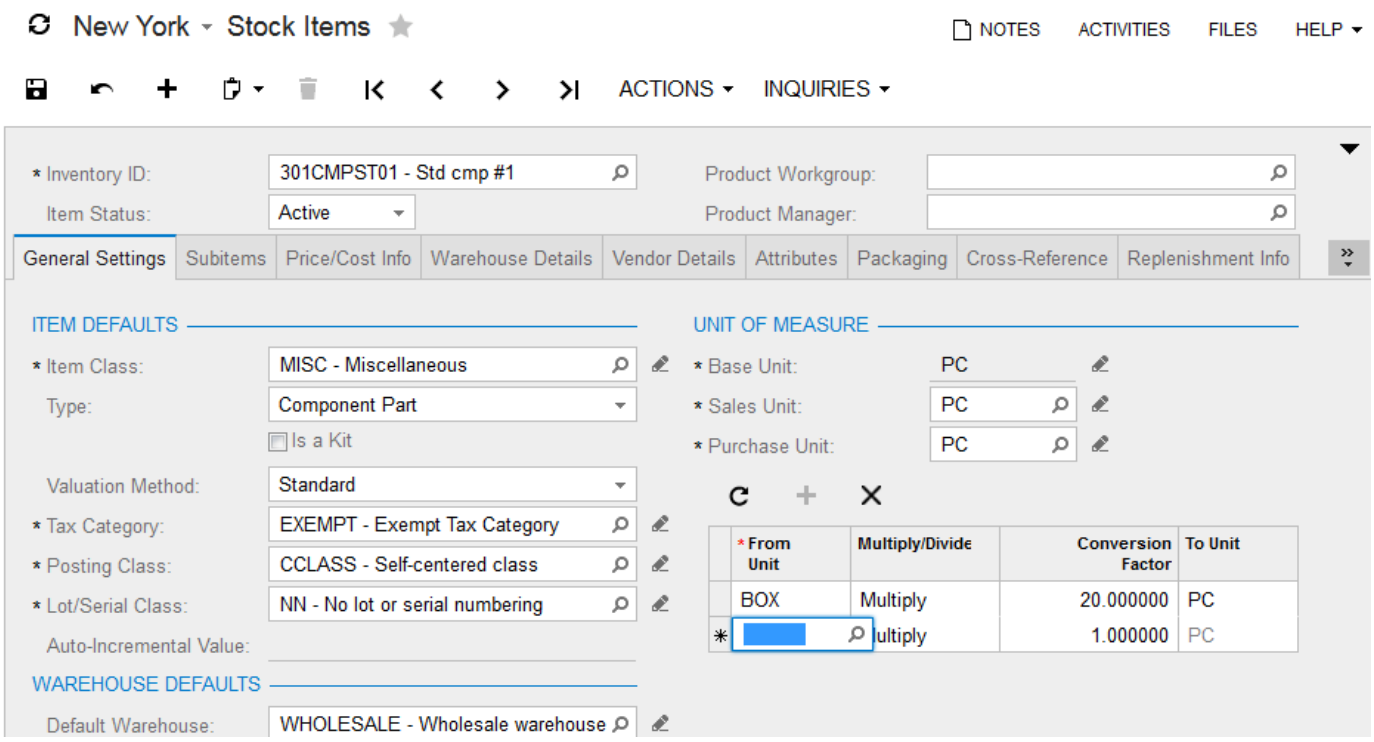
Grid

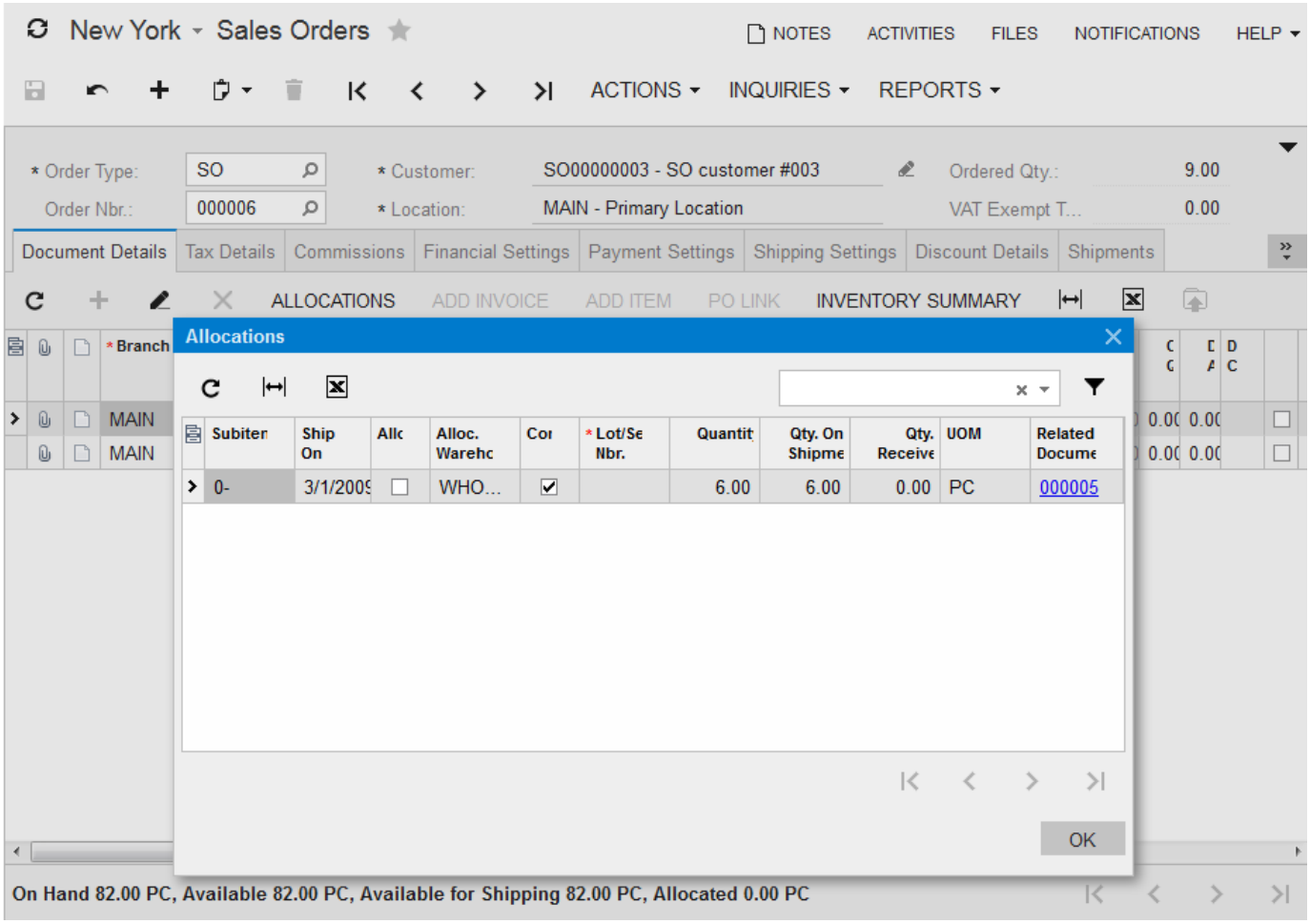
Description
Grid allows you to manipulate with records in table.

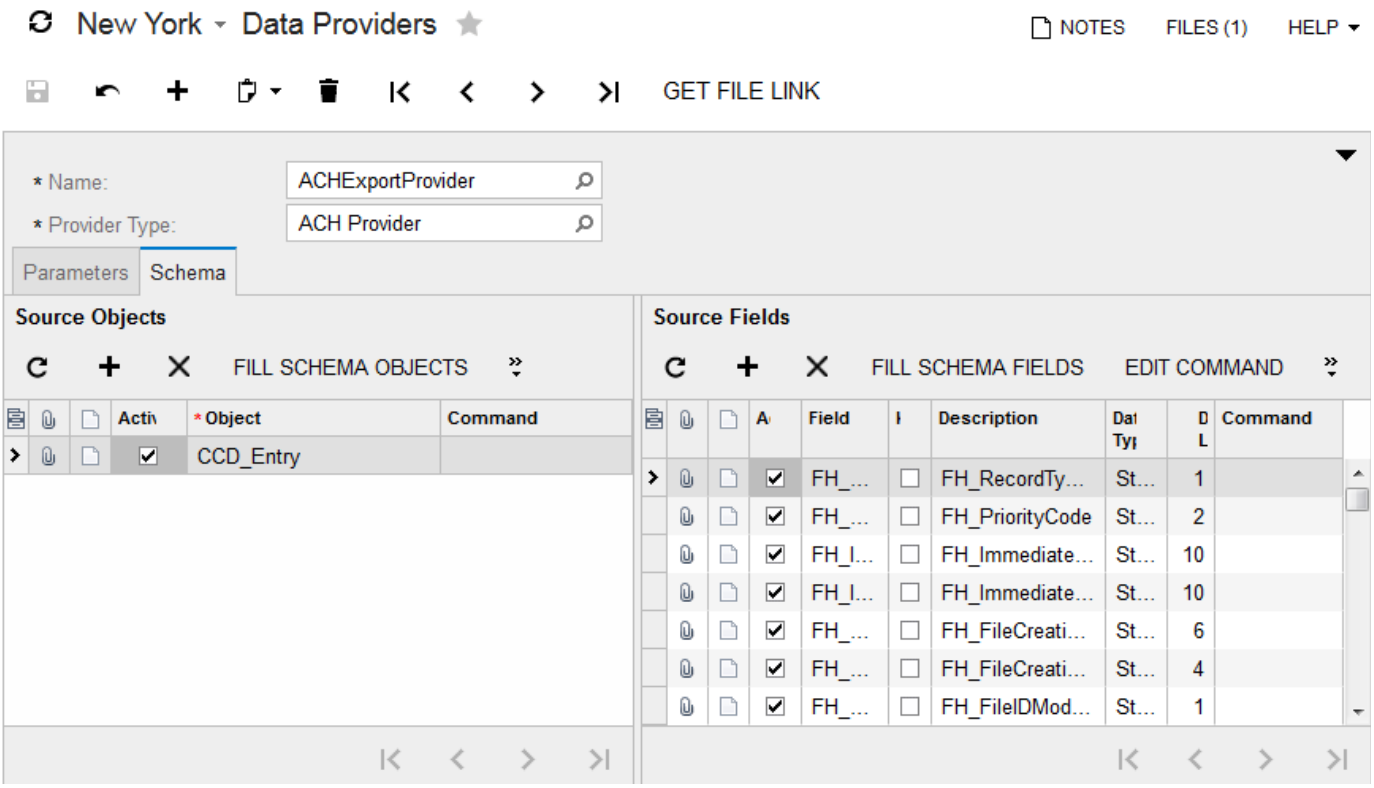
- How Grid looks like in user interface
- How to use Grid

How Grid looks like in user interface






New York
Sales Orders
NOTES
ACTIVITIES
FILES
NOTIFICATIONS
HELP
ACTIONS
INQUIRIES
REPORTS
* Order Type:
SO
* Customer:
SO00000003 - SO customer #003
Ordered Qty.:
9.00
Order Nbr.:
000006
* Location:
MAIN - Primary Location
VAT Exempt T...
0.00
Document Details
Tax Details
Commissions
Financial Settings
Payment Settings
Shipping Settings
Discount Details
Shipments
ALLOCATIONS
ADD INVOICE
ADD ITEM
PO LINK
INVENTORY SUMMARY
Allocations
Subiter
Ship On
Alloc.
Alloc. Wareho
Lot/Se Nbr.
Quantit
Qty. On Shipme
Qty. Receive
UOM
Related Docume
0-
3/1/2009
WHO...
6.00
6.00
0.00
PC
000005
OK
On Hand 82.00 PC, Available 82.00 PC, Available for Shipping 82.00 PC, Allocated 0.00 PC


New York
Data Providers
NOTES
FILES (1)
HELP
GET FILE LINK
* Name:
ACHExportProvider
* Provider Type:
ACH Provider
Parameters
Schema
Source Objects
FILL SCHEMA OBJECTS
Acti
* Object
Command
CCD_Entry
Source Fields
FILL SCHEMA FIELDS
EDIT COMMAND
Field
Description
Dat Typ
Command
FH_...
FH_RecordTy...
FH_PriorityCode
FH_Immediate...
FH_FileCreati...
FH_FileIDMod...

How to use Grid

Code fragment below shows how to add row in "Transaction Details" grid on "Journal Transactions" screen:

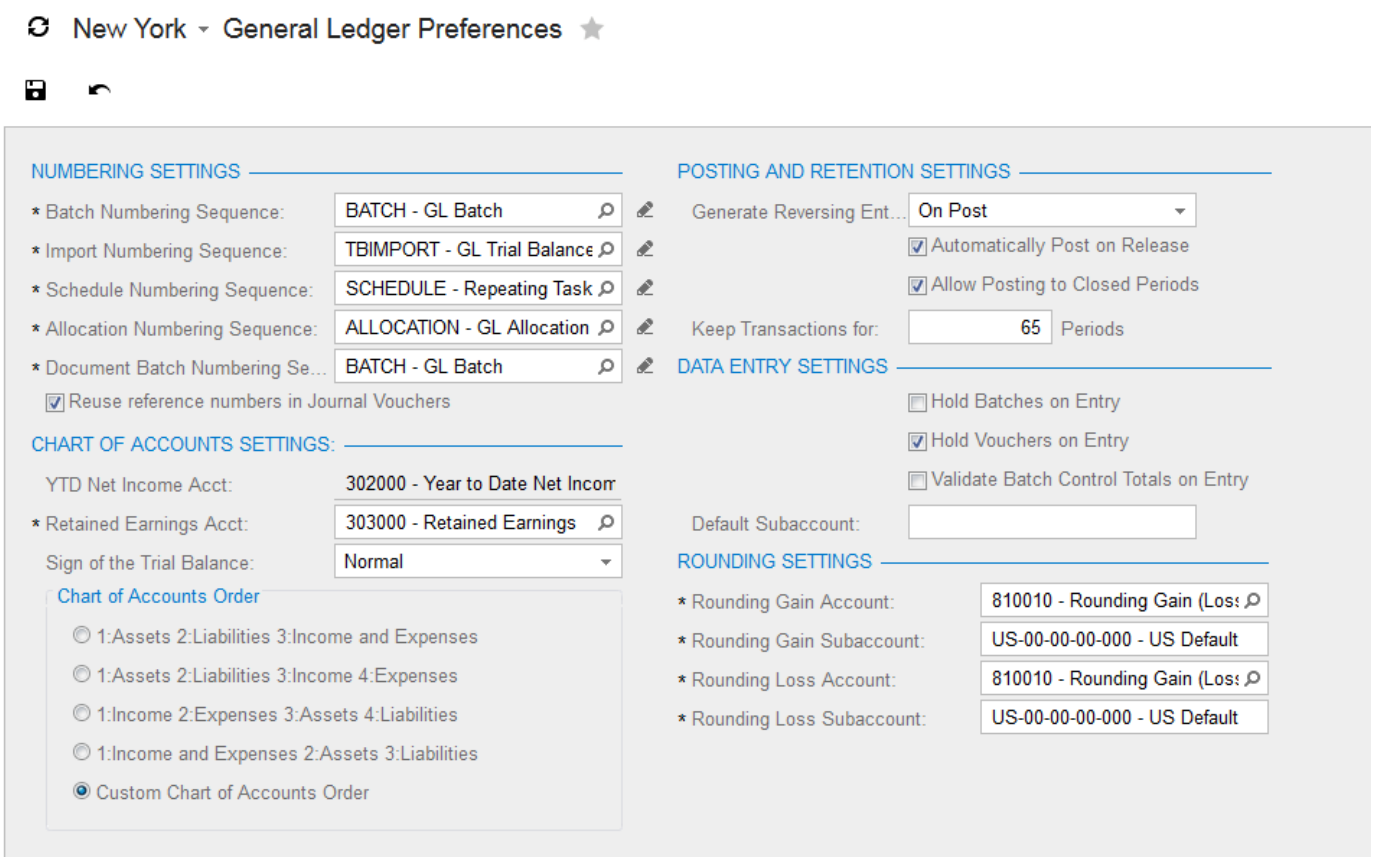
```
var batch = new JournalEntry();
batch.OpenScreen();
batch.Details.New();
batch.Details.Row.AccountID.Select("100000");
batch.Details.Row.SubID.Type("US000000000");
batch.Details.Row.CuryDebitAmt.Type(100);
batch.Details.Row.CuryCreditAmt.Type(0);
```

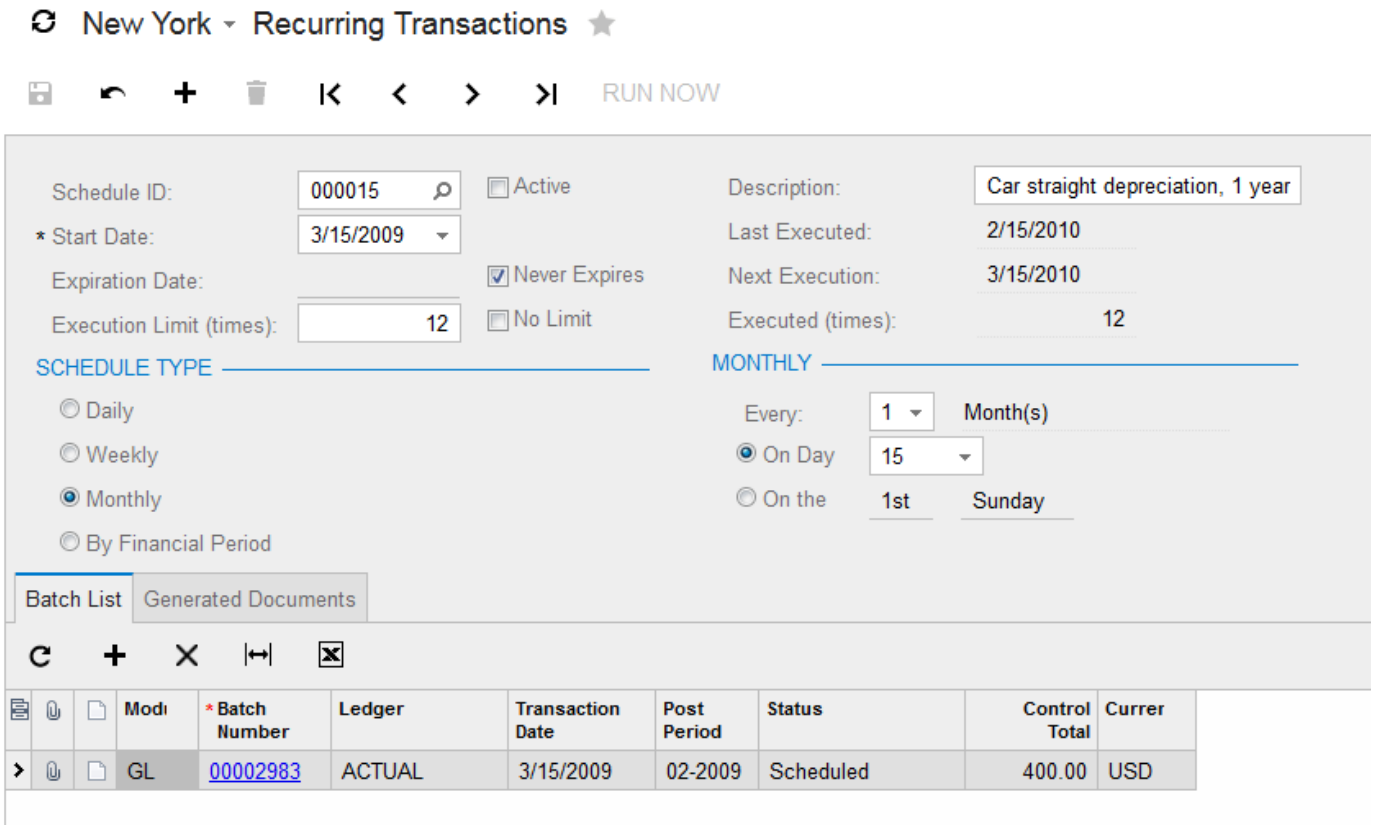
GroupBox

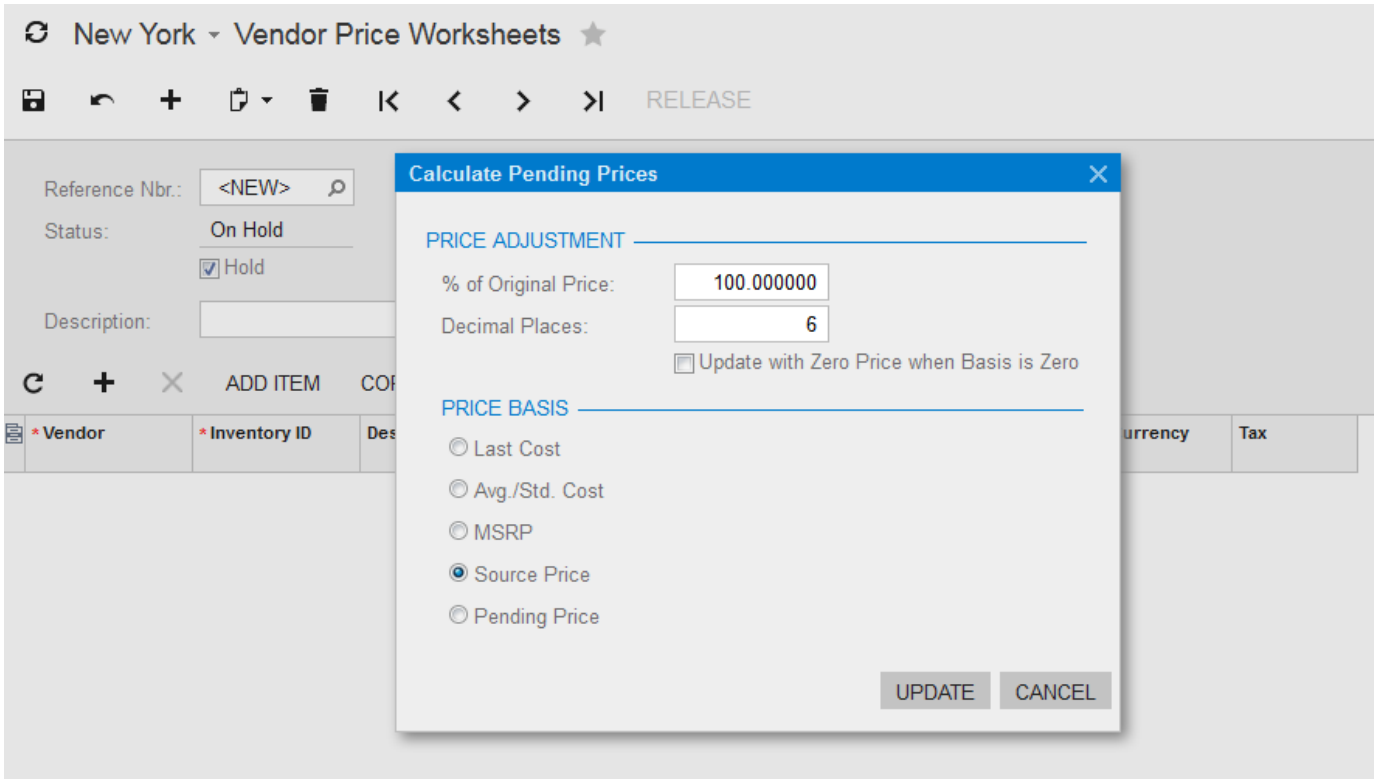
Description
Group-box allow you to select one of te options available.

- How group-box looks like in user interface
- How to use group-box

How group-box looks like in user interface







How to use group-box

Code fragments below show how to use group-box:

```
GenerateRecurringTransactionsGl.Summary.LimitTypeSel.Set("On this date");
```

Code below shows how to select differnet price basis options in pop-up panel "Calculate Pending Prices" on "Vendor Price Worksheets" screen:

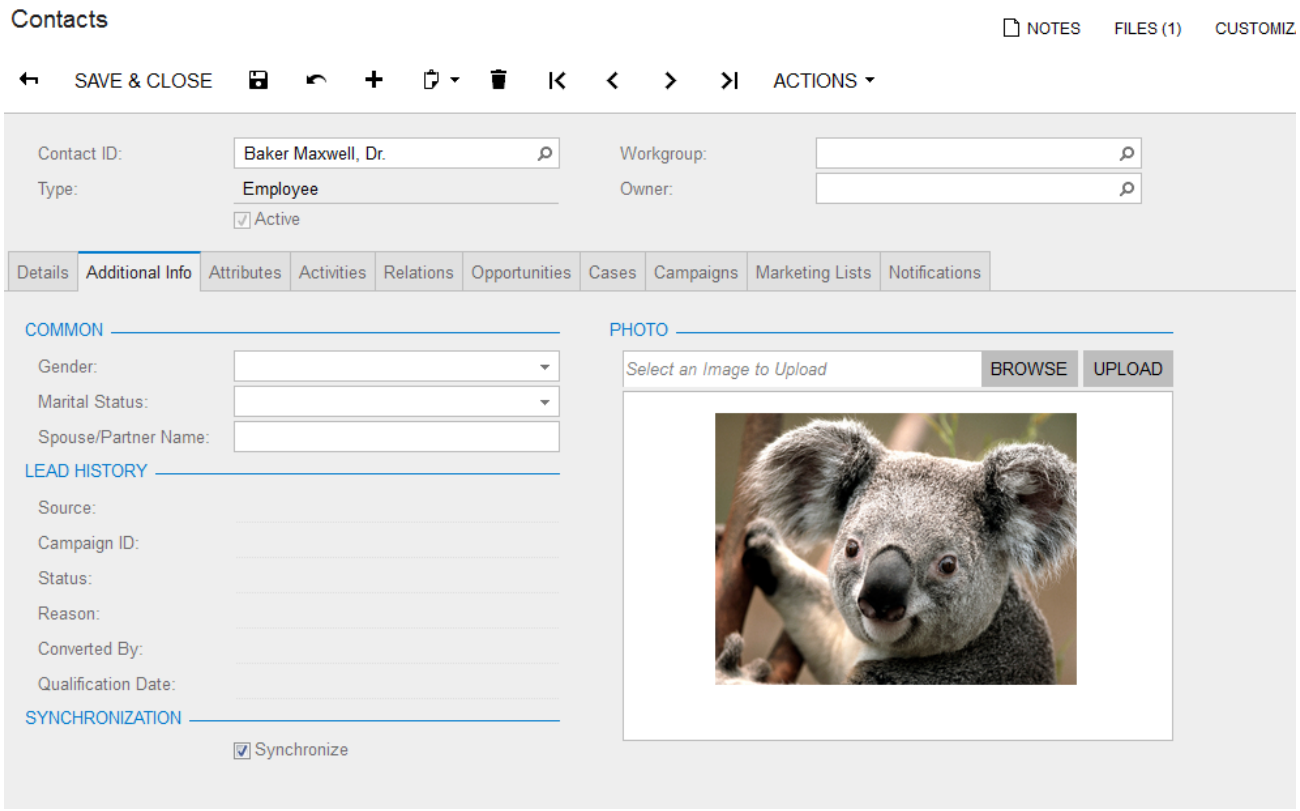
```
VendorPriceWorksheets vendorPriceWorksheets = new VendorPriceWorksheets();
vendorPriceWorksheets.OpenScreen();
vendorPriceWorksheets.Details.CmdCalculate();
vendorPriceWorksheets.CalculatePendingPrices.PriceBasis.Set("Last Cost");
vendorPriceWorksheets.CalculatePendingPrices.PriceBasis.Set("MSRP");
vendorPriceWorksheets.CalculatePendingPrices.PriceBasis.Set("Pending Price");
vendorPriceWorksheets.CalculatePendingPrices.Cancel();
```

ImageUploader

Description
Represents upload image control. This control used for upload and preview images in Acumatica.

- How Image Uploader looks like in user interface
- How to use Image Uploader

How Image Uploader looks like in user interface



How to use Image Uploader

This control extend functionality of FileUploader and adds image preview panel

Next example show upload image process

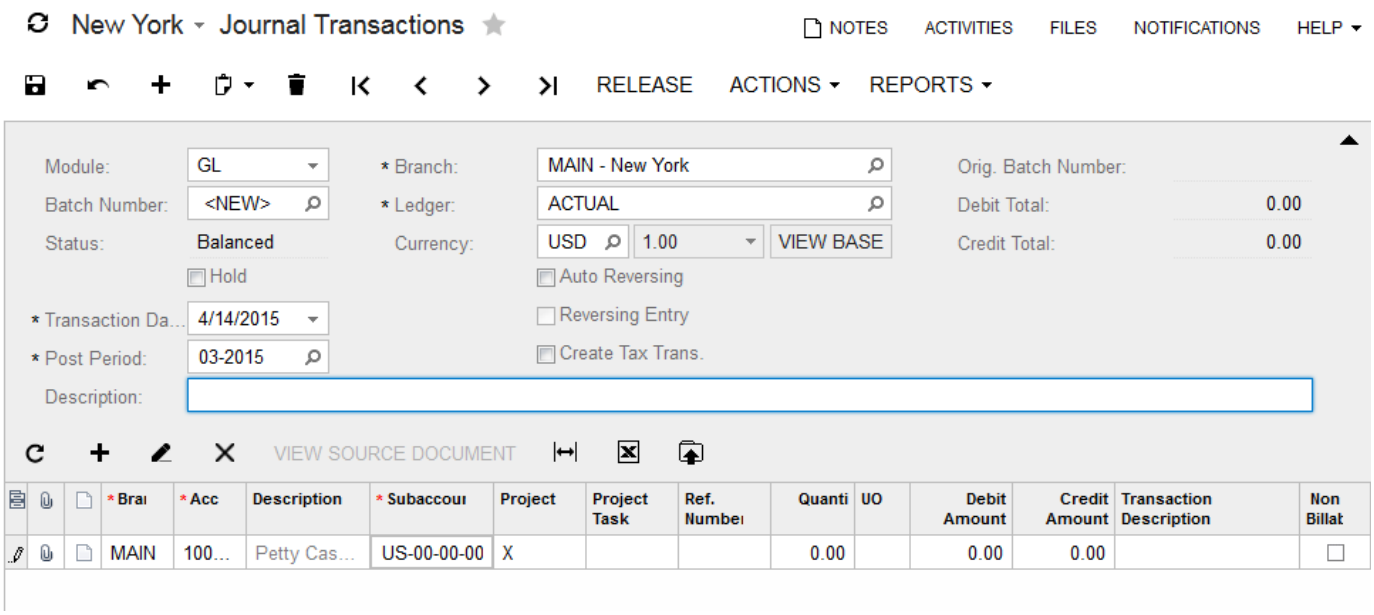
```
// Click BROWSE, type file path and click Open
Contact.Details.Img.SelectFile(@"\\qaserver\QACenterShare\testdata\Exchange\pomidor.png");
// Click UPLOAD button
Contact.Details.Img.UploadFile();
```

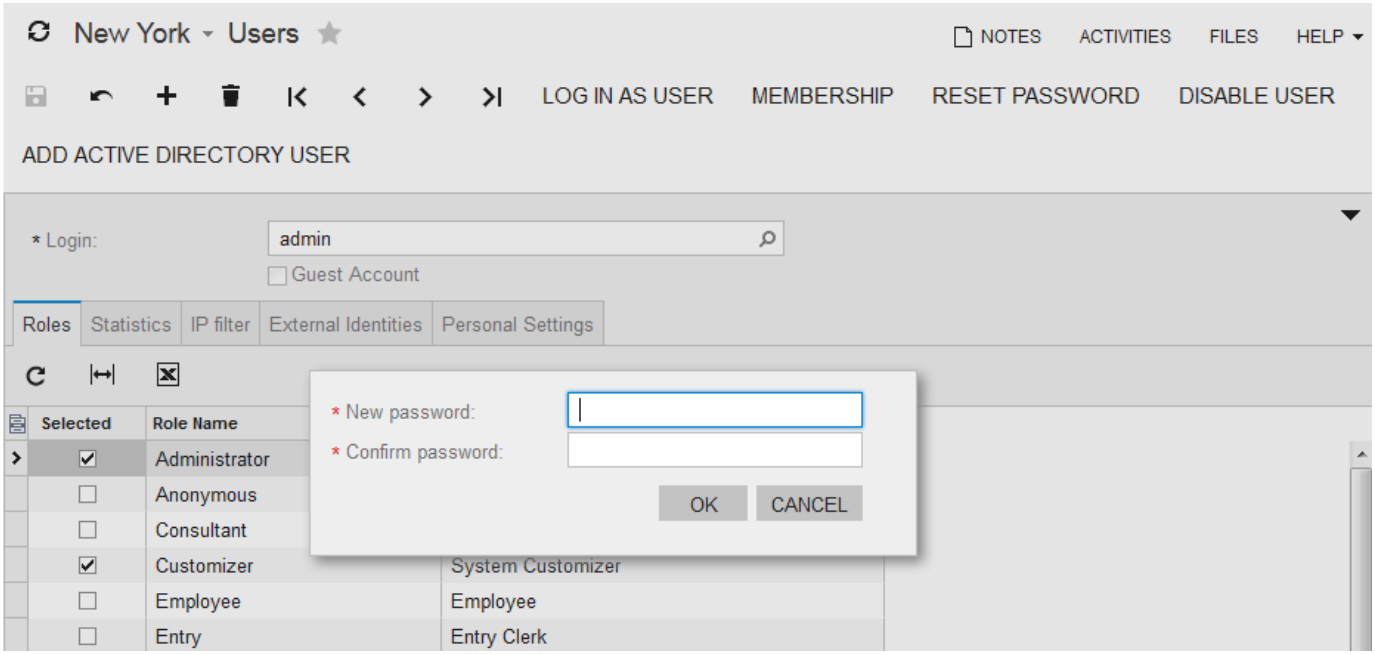
Input

Description
Input allows you to type text values.

- How input looks like in user interface
- How to use input

How input looks like in user interface





How to use input

The code fragments below type value into "Description" filed on "Journal Transactions" screen:

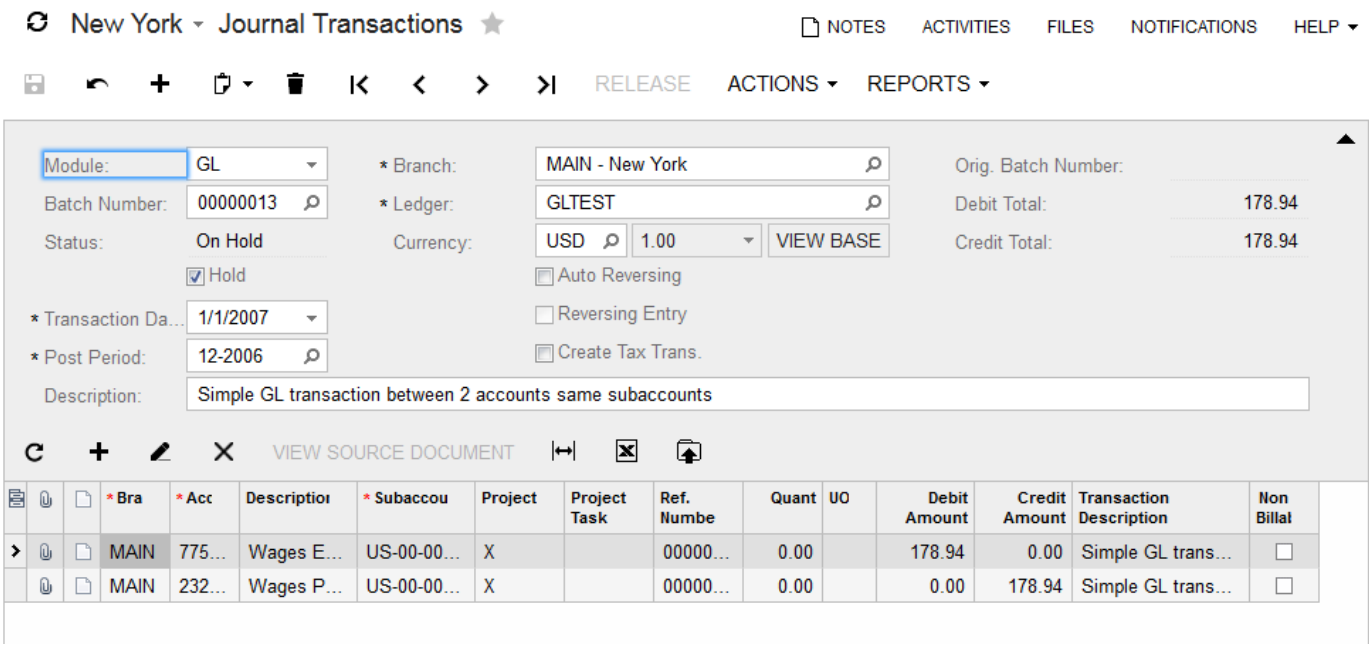
```
JournalEntry journalEntry = new JournalEntry();
journalEntry.OpenScreen();
journalEntry.Insert();
journalEntry.Summary.Description.Type("Test");
```

Label

Description
Label shows informational text, more often descriptive name of the respective field.

- How Label Looks Like in User Interface
- How to Use Label

How Label Looks Like in User Interface



How to Use Label

The code below opens "Journal Transaction" screen, gets text from label next to "Module" field and types it into console:

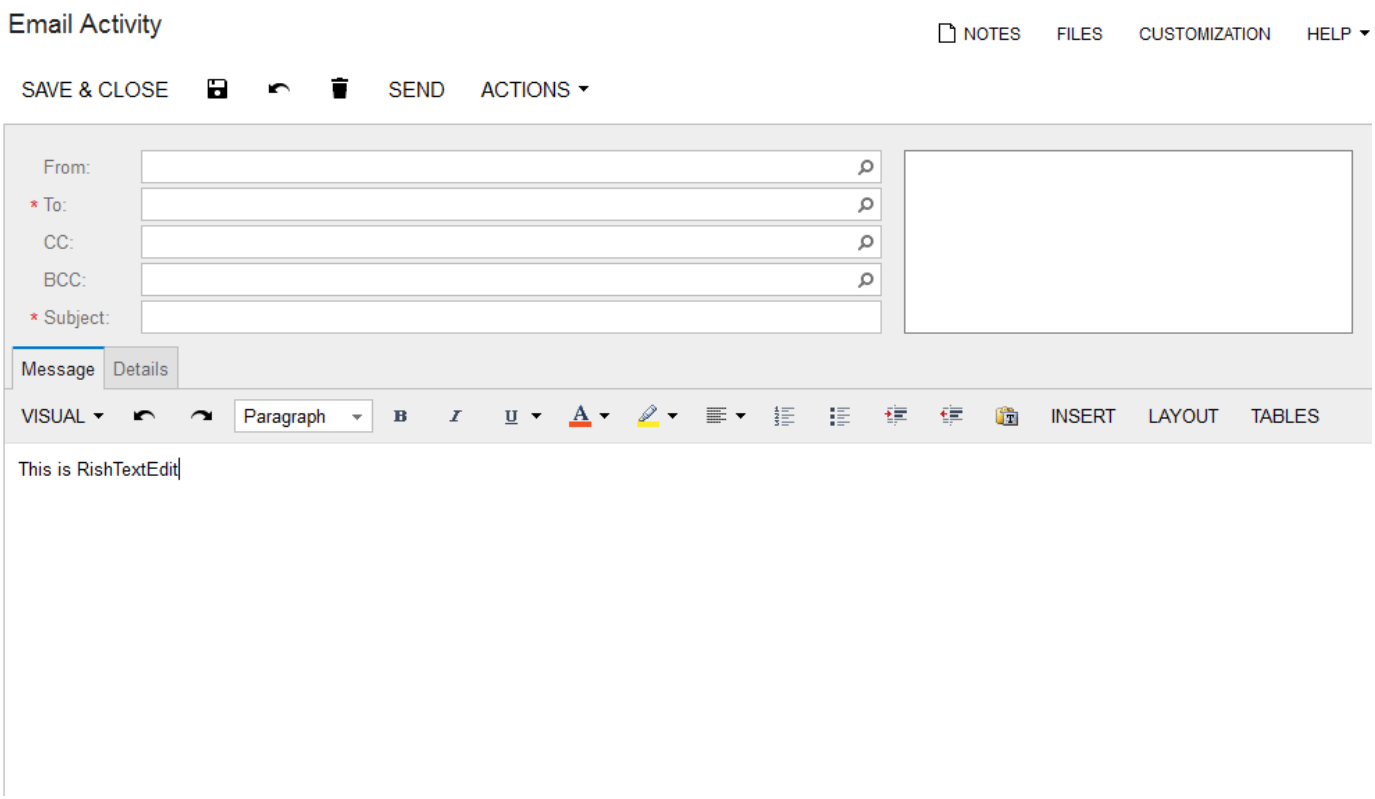
```
JournalEntry batch = new JournalEntry();
batch.OpenScreen();
var module = batch.Summary.ModuleLabel.GetValue();
```

RichTextEdit

Description
Represents complex text editor. Allows to edit text, press several editor buttons and perform text navigation

- How Rich Text Edit looks like in user interface
- How to use Rich Text Edit

How Rich Text Edit looks like in user interface



How to use Rich Text Edit

Next example demonstrate typical editor usage.

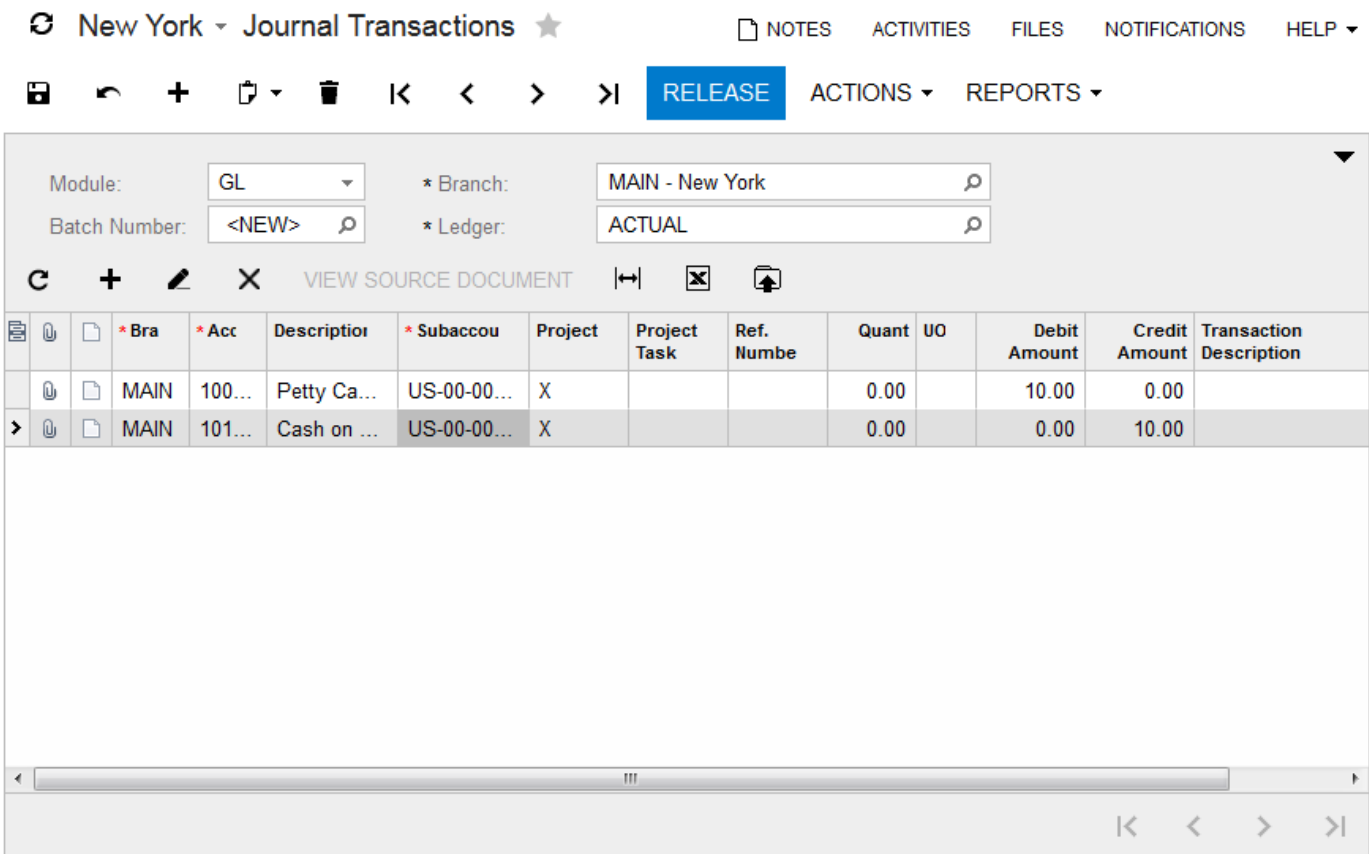
```
// Change editor mode to Visual
Email.Details.Body.Mode.Visual();
// Erase all content from editor
Email.Details.Body.Clear();
// Type new content
Email.Details.Body.Type("EX404 – Step 2");
```

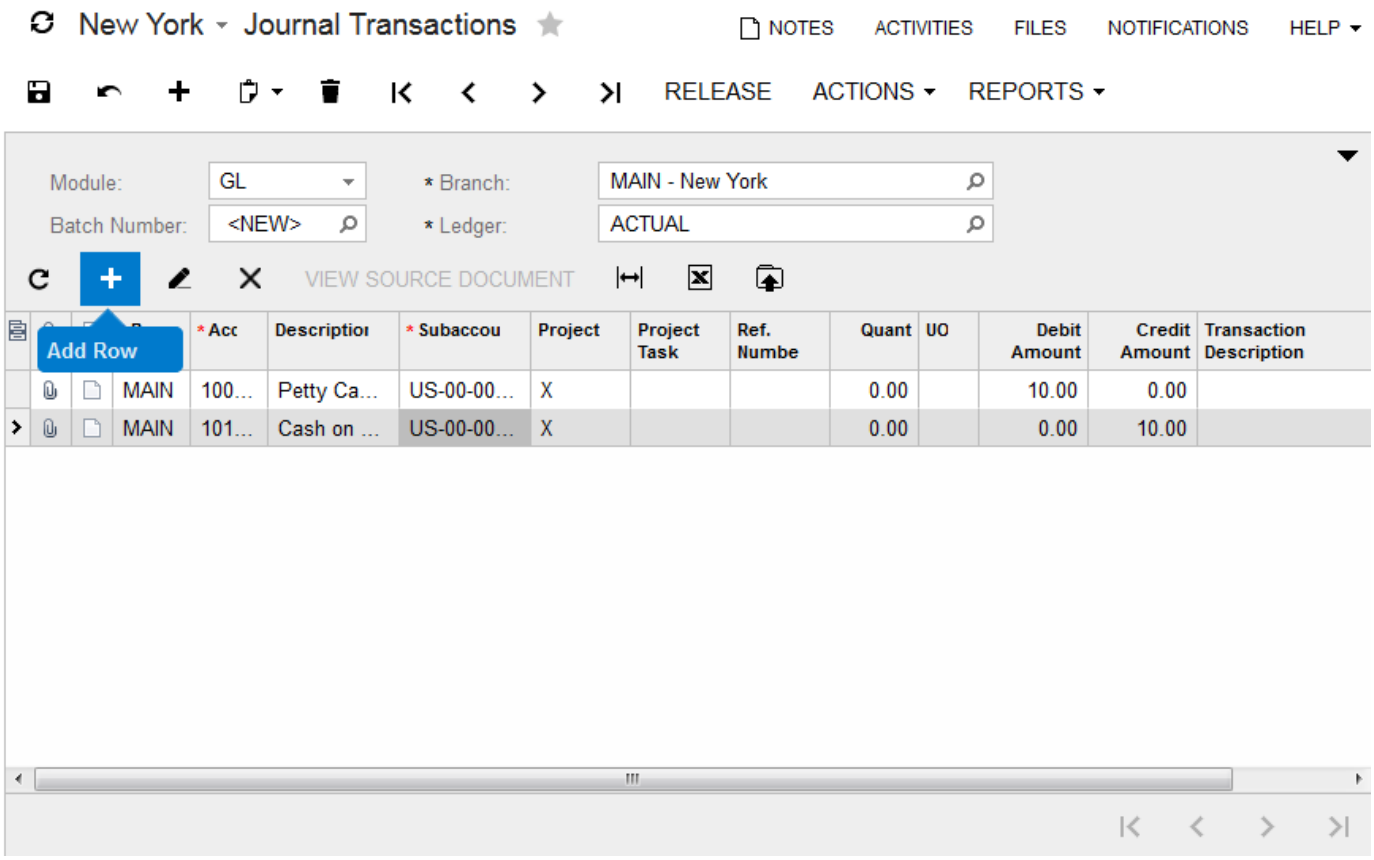
ToolBarButton

Description
ToolBarButton allows you to click on buttons in screen, report or grid toolbars.

- How ToolBarButton Looks Like in User Interface
- How to Use ToolBarButton

How ToolBarButton Looks Like in User Interface





How to Use ToolBarButton

The code below opens "Journal Transactions" screen and clicks "Insert" button:

```
JournalEntry batch = new JournalEntry();
batch.OpenScreen();
batch.Insert();
```

The code below opens "Journal Transactions" screen, skips to the last batch available and clicks "Release" button if it is enabled:

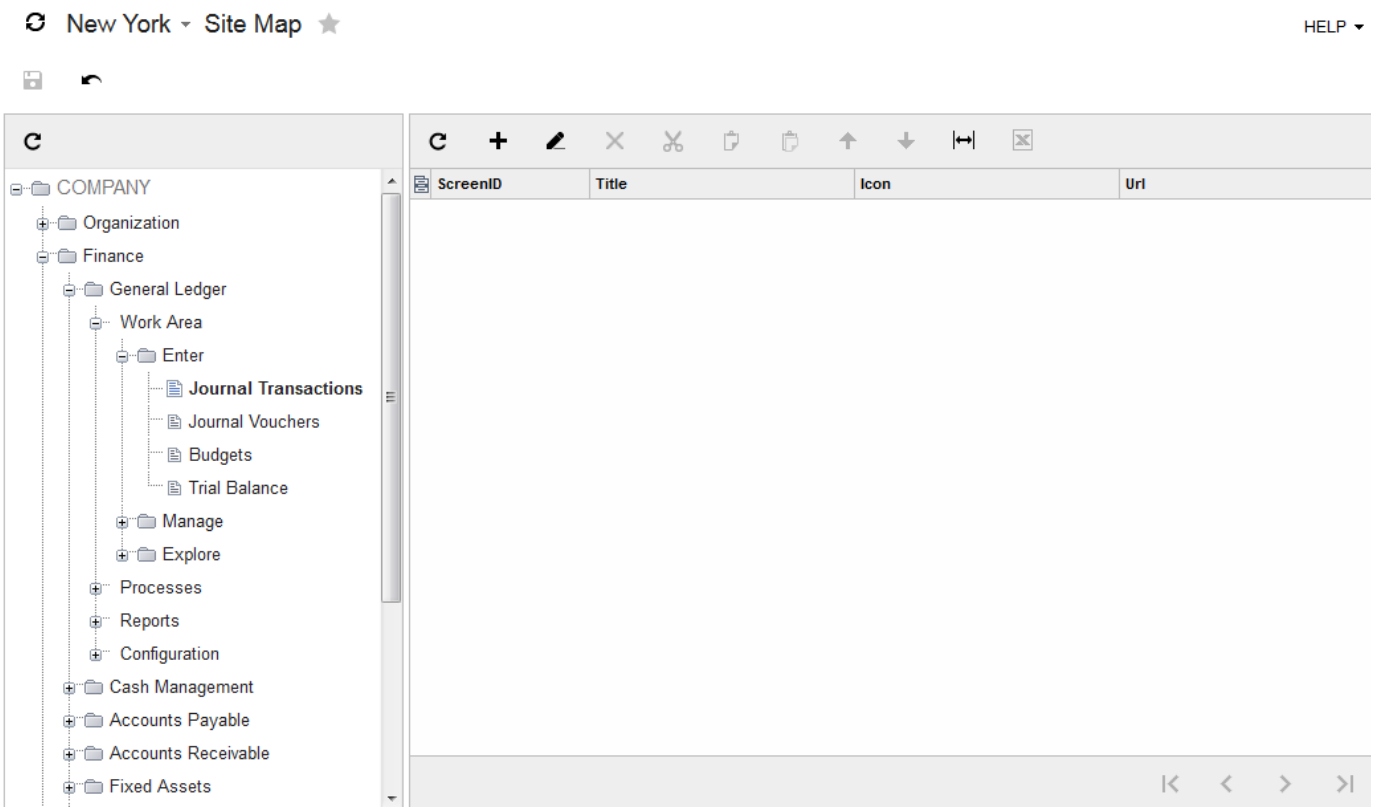
```
JournalEntry batch = new JournalEntry();
batch.ToolBar.Release.WaitAction = Wait.WaitForLongOperationToComplete;
batch.OpenScreen();
batch.Last();

if (batch.ToolBar.Release.IsEnabled())
{
 batch.Release();
}
```

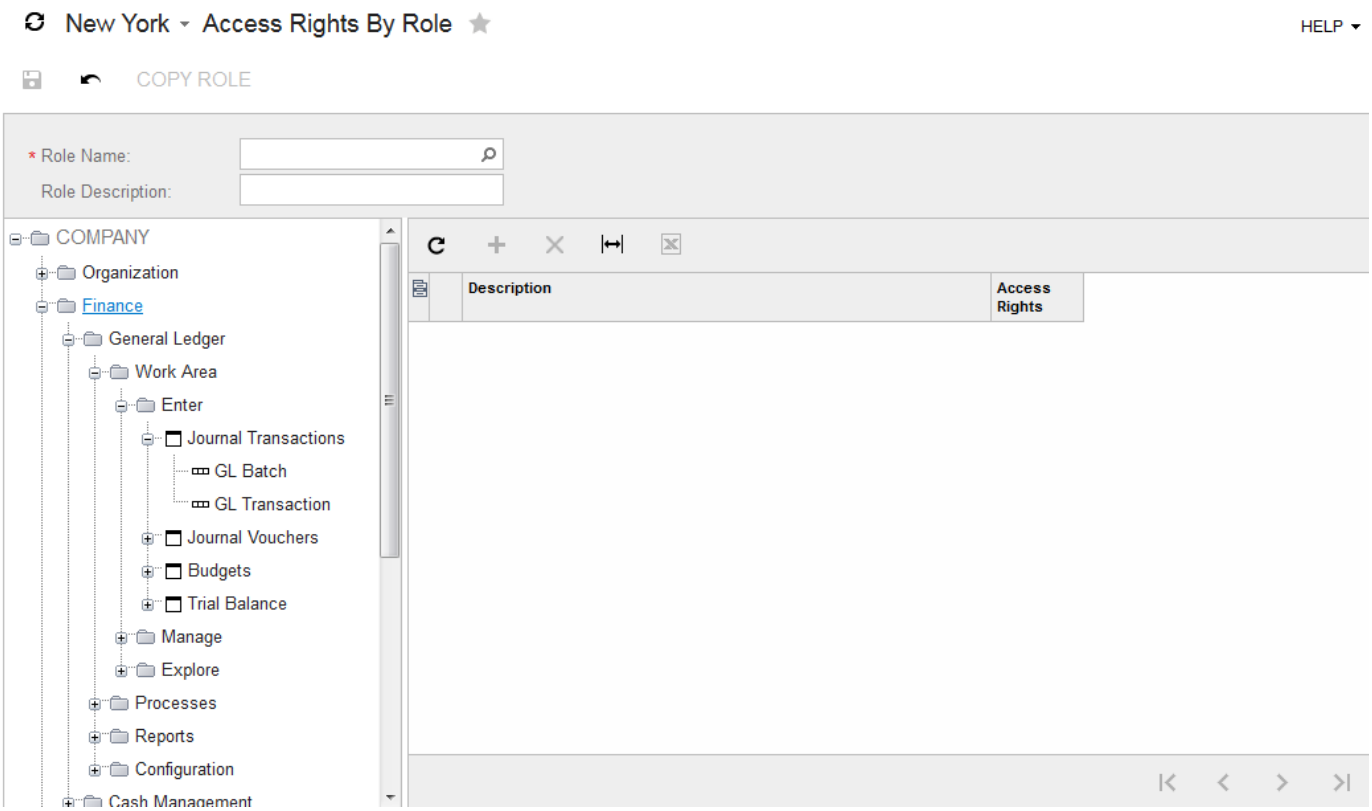
TreeView

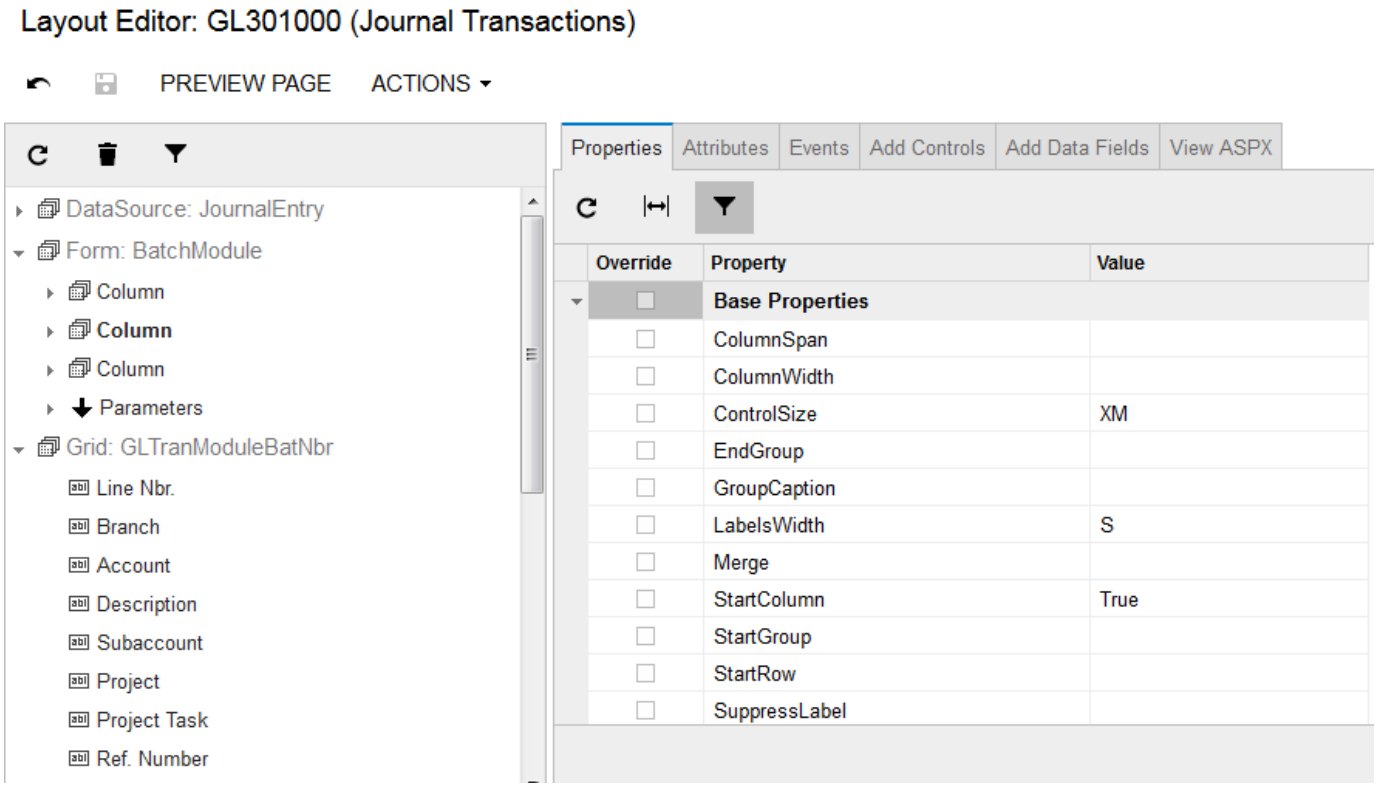
Description
Tree-view allows you to select nodes in tree.

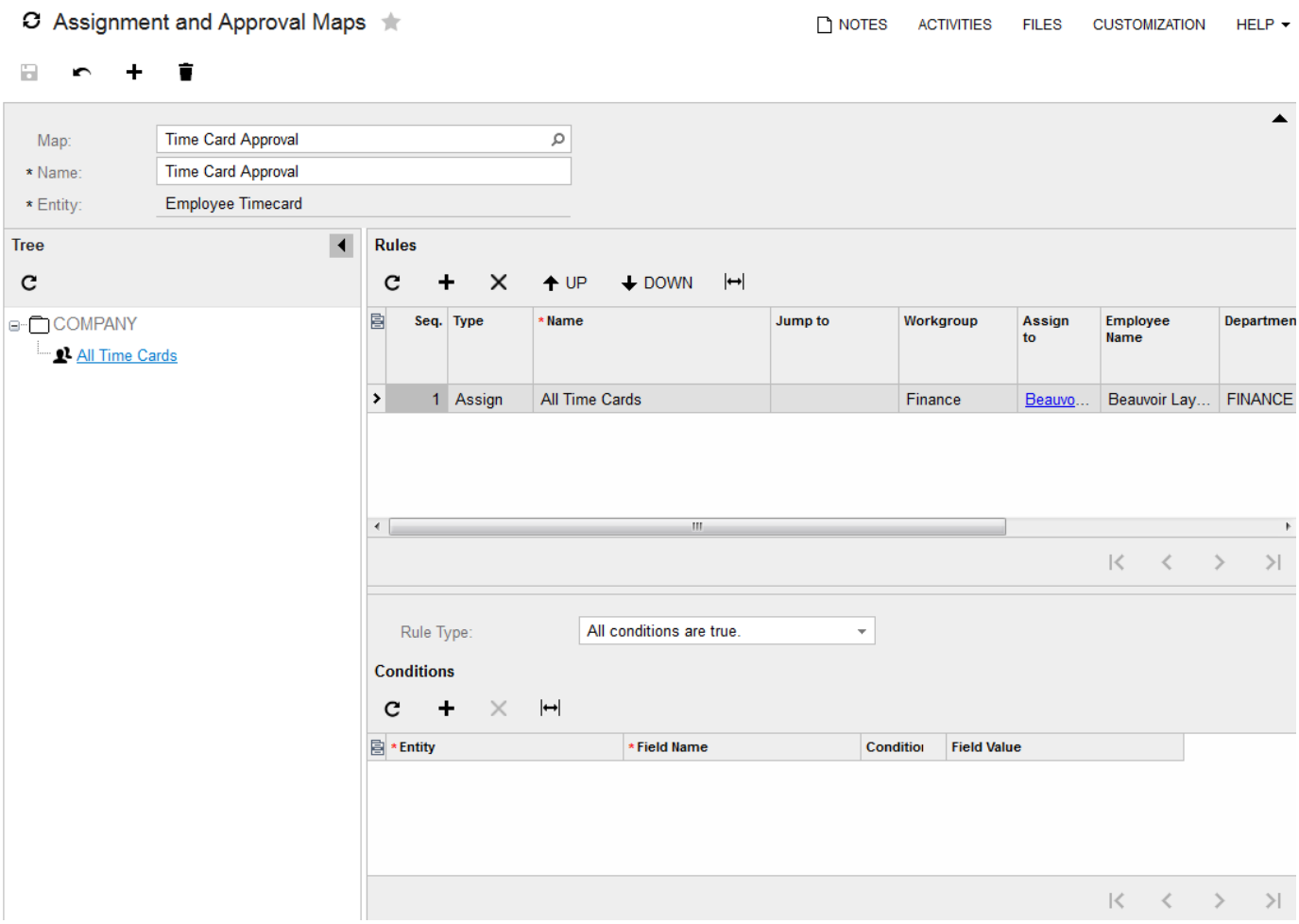
- How Tree-View Looks Like in User Interface
- How to Use Tree-View
- Known Issues & Helpful Tips

How Tree-View Looks Like in User Interface




New York
Access Rights By Role
HELP
COPY ROLE
* Role Name:
Role Description:
COMPANY
Organization
Finance
General Ledger
Work Area
Enter
Journal Transactions
GL Batch
GL Transaction
Journal Vouchers
Budgets
Trial Balance
Manage
Explore
Processes
Reports
Configuration
Cash Management
Description
Access Rights


Layout Editor: GL301000 (Journal Transactions)
PREVIEW PAGE
ACTIONS
DataSource: JournalEntry
Form: BatchModule
Column
Column
Column
Parameters
Grid: GLTranModuleBatNbr
Line Nbr.
Branch
Account
Description
Subaccount
Project
Project Task
Ref. Number
Properties
Attributes
Events
Add Controls
Add Data Fields
View ASPX
Override
Property
Value
Base Properties
ColumnSpan
ColumnWidth
ControlSize
XM
EndGroup
GroupCaption
LabelsWidth
S
Merge
StartColumn
True
StartGroup
StartRow
SuppressLabel



How to Use Tree-View

The code fragments below shows how to open "Journal Transactions" node in site-map tree:

```
SiteMap siteMap = new SiteMap();
siteMap.OpenScreen();
siteMap.Parameters.Tree.Select("COMPANY/Finance/General Ledger/Work Area/Enter/Journal Transactions");
```

Known Issues & Helpful Tips

If tree contains several nodes with identical names you can select any of them by index:

```
JournalEntry journalEntry = new JournalEntry();
CustomizationObjectExt customizationMenu = new CustomizationObjectExt();
LayoutEditor layoutEditor = new LayoutEditor();

journalEntry.OpenScreen();
customizationMenu.Custom();
customizationMenu.InspectelementCtrlAtlClick();
journalEntry.Summary.BranchID.Click();
customizationMenu.ElementProperties.Customize();
customizationMenu.SelectCustomizationProject.New();
customizationMenu.NewProject.NewProject.Type("Test");
customizationMenu.NewProject.Ok();
customizationMenu.SelectCustomizationProject.Ok();

layoutEditor.Properties.SelectRow(layoutEditor.Properties.FilterForm.Fields.Value.Value, "BranchID");
layoutEditor.ParametersTree.TreePageControls.Select("Form: BatchModule/Column", 3);
layoutEditor.ParametersTree.TreePageControls.Select("Form: BatchModule/Column", 1);
layoutEditor.ParametersTree.TreePageControls.Select("Form: BatchModule/Column", 2);
```

ClassGenerator.exe

ClassGenerator.exe is a command line tool that allows Test SDK users to generate page wrappers for applications built on top of Acumatica Framework.

- ClassGenerator.exe.config

ClassGenerator.exe.config

- ClassGenerator.exe.config
- PagesList.txt
- PagesWithParameters.txt
- GenericInquiriesWithParameters.txt

ClassGenerator.exe.config

You can change the settings of the page wrapper generation tool (ClassGenerator.exe) in the ClassGenerator.exe.config file. The structure of this file is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <appSettings>
        <!--Local path to the Acumatica ERP instance installation directory-->
        <add key="SitePhysicalPath" value="C:\Program Files (x86)\Acumatica ERP\yoursite"/>
        <!--Output directory to store the generated page wrappers-->
        <add key="GenResultPath" value="C:\share\output"/>
        <!--User to be used for page wrapper generation-->
        <add key="UserName" value="admin@Demo"/>
        <!--IDs of the pages you want to run wrapper generation for; the wildcard * is supported-->
        <add key="FileNameFilter" value="CS100000, CS102000, CM202000, GL201500, CS202000, GL202500, GL102000, GL101000, GL201000"/>
        <!--Deletes all files in output directory before running the page wrapper generation process-->
        <add key="ClearOutput" value="true"/>
        <!--Namespace where wrapper classes will be defined. Use the template "GeneratedWrappers.<PartnerName>".-->
        <add key="Namespace" value="GeneratedWrappers.Acumatica"/>
    </appSettings>
</configuration>
```

You can use the following keys in the ClassGenerator.exe.config file to configure the generation of page wrappers.

#	Key Name	Key Description	Mandatory/ Optional	Usage Example
1	`SitePhysicalPath`	Specifies the local path to the installation directory of an Acumatica ERP instance.	Mandatory	`<add key="SitePhysicalPath" value="C:\Program Files (x86)\Acumatica ERP\demo"/>`
2	`GenResultPath`	Specifies the directory where the generated page wrappers should be saved.	Mandatory	`<add key="GenResultPath" value="C:\Output"/>`
3	`UserName`	Specifies the user name to be used to log in to the Acumatica ERP instance. If you need to log in to a specific company, you should specify the user name in the following format: `UserName@CompanyName`, where `UserName` is replaced with the name of the user, and `CompanyName` is replaced with the name of the company to which you want to log in.	Mandatory	`<add key="UserName" value="admin"/>` `<add key="UserName" value="admin@Company"/>`

4	`FileNameFilter`	Specifies the IDs of the forms for which you want to generate page wrappers.
		You can use two-letter prefix of the Acumatica ERP module name as the value of this key to generate wrappers for all forms of the module.
		You can use * as the value of this key to generate wrappers for all pages.
			Mandatory	`<add key="FileNameFilter" value="GL301000, GL501000"/>`
				`<add key="FileNameFilter" value="GL301000, CR"/>`
				`<add key="FileNameFilter" value="*"/>`

5	`PagesList`	Specifies the file that contains the list of IDs of the forms that should be included in or excluded from the page wrapper generation.	Optional	`<add key="PagesList" value="PagesList.txt"/>`
6	`PagesListAttribute`	Specifies whether the forms that are specified in the `PagesList` key should be included in or excluded from the page wrapper generation. You can set the value of this key to `include` to include the forms in the page wrapper generation. You can set the value of this key to `exclude` to exclude the forms from the page wrapper generation.	Optional	`<add key="PagesListAttribute" value="include"/>` `<add key="PagesListAttribute" value="exclude"/>`
7	`ClearOutput`	Specifies whether all files in the output directory, which is specified in the `GenResultPath` key,	Mandatory	`<add key="ClearOutput" value="true"/>`

		should be removed before the page wrapper generation starts. You can set the value of this key to `true` to remove all files from the output folder before page wrapper generation. You can set the value of this key to `false` to not delete the files from the output folder before page wrapper generation.		`<add key="ClearOutput" value="false"/>`
8	`Namespace`	Specifies the namespace where page wrapper classes should be defined. We recommend that you use the following template for the namespace name: `GeneratedWrappers.<PartnerName>`.	Optional	`<add key="Namespace" value="GeneratedWrappers.Acumatica"/>`
9	`PagesParameters`	Specifies the file that contains the list of forms that require some parameters to open.	Optional	`<add key="PagesParameters" value="PagesWithParameters.txt"/>`
10	`GenericInquiryParameters`	Specifies the file that contains the list of generic inquiries that require some parameters to open.	Optional	`<add key="GenericInquiryParameters" value="GenericInquiriesWithParameters.txt"/>`

11	`CorrectCTL01`	Specifies whether the entries of `ctl01` should be replaced with `ctl00` in the generated page wrappers.
		
		You can set the value of this key to `true` to replaces entries of `ctl01` with `ctl00` in the generated page wrappers.
		
		You can set the value of this key to `false` to not replace entries of `ctl01` with `ctl00` in the generated page wrappers.
			Optional	`<add key="CorrectCTL01" value="true"/>`
				
				`<add key="CorrectCTL01" value="false"/>`

PagesList.txt

```
<add key="PagesList" value="PagesList.txt" />
```

Specifies the file that contains the list of IDs of the forms that should be included in or excluded from the page wrapper generation.

Example

```
CS100000
CS102000
GL201500
GL301000
GL501000
```

PagesWithParameters.txt

```
<add key="PagesParameters" value="PagesWithParameters.txt"/>
```

Specifies the file that contains the list of forms that require some parameters to open.

Example

```
CR306010 ?TaskID=29&RefNoteID=764
CR306015 ?TaskID=30&RefNoteID=764&NotificationID=4517FCC3-98A7-4521-B4C7-1BD80B2846E6
```

GenericInquiriesWithParameters.txt

```
<add key="GenericInquiryParameters" value="GenericInquiriesWithParameters.txt"/>
```

Specifies the file that contains the list of generic inquiries that require some parameters to open.

Example

```
GI000001 ?Name=Currency Rates History
CR3010PL ?ID=20e4ab0d-0631-4759-a48d-6ec20ac78291
CR3060PL ?ID=24d48139-cf11-4be6-9bd3-57ee86893778
CR3040PL ?ID=a95a50d6-6892-4052-b6aa-8e769efa2bfc
CR3080PL ?ID=6b673610-9cee-46c3-adb4-1e1569ddbb0b
CR3020PL ?ID=d345a840-d1cf-4d6f-a2df-a454b85b20d8
CR3030PL ?ID=df95d4e3-fb8c-4aff-ba9a-f8371e7b1908
CR2040PL ?ID=45c1d74f-f7b4-498d-bee2-416863b35196
CR2020PL ?ID=d7dfa28a-36ad-467c-addb-c2080eda3484
```

Part 3: Acumatica Test SDK—Programming Tasks

In this part, you can find descriptions of the programming tasks that you can perform by using Acumatica Test SDK.

In This Part

- Known Issues
- How to Use Page Wrapper Generation Tool
- How to Generate Page Wrappers Using Test SDK API
- How to Work With Errors That Occur During Page Wrapper Generation
- How to Change the Settings of the Page Wrapper Generation Tool
- How to Change Browser Settings
- How to Change Chrome Settings
- How to Change Culture Settings
- How to Change Predefined Timeouts
- How to Manage Log Providers
- How to Work with Alerts
- How to Commit Changes in Rows of a Detail Table
- How to Navigate through the Rows of a Detail Table
- How to Show or Hide Columns in a Detail Table
- How to Use Column Filters of a Detail Table
- How to Work with Errors
- How to Work with Warnings
- How to Work with Windows
- How to Work with the Names of UI Elements
- How to Work with Drop-Down Menus on Toolbars
- How to Work with Pop-Up Dialog Boxes
- How to Work with Pop-Up Panels
- How to Verify a Note on a Form
- How to Add a Note to a Detail Line
- How to Add a Note to an Acumatica Form
- How to Upload a File
- How to Upload Data from an Excel File into a Detail Table
- How to Upload Data from a CSV File into a Detail Table
- How to Attach a File to an Acumatica Form
- How to Attach a File to a Detail Line
- How to Remove an Attached File from an Acumatica Form
- How to Remove an Attached File from a Detail Line
- How to Publish Customization Projects
- How to Work with Smart Delays
- How to Capture Screenshots
- How to Get Datetime in the Current User's Timezone
- How to use Comparator to compare .xml, .csv, .pdf, Excel files
- How to Verify string/long/int/DateTime returned by a method
- How to work with dynamic controls

Known Issues

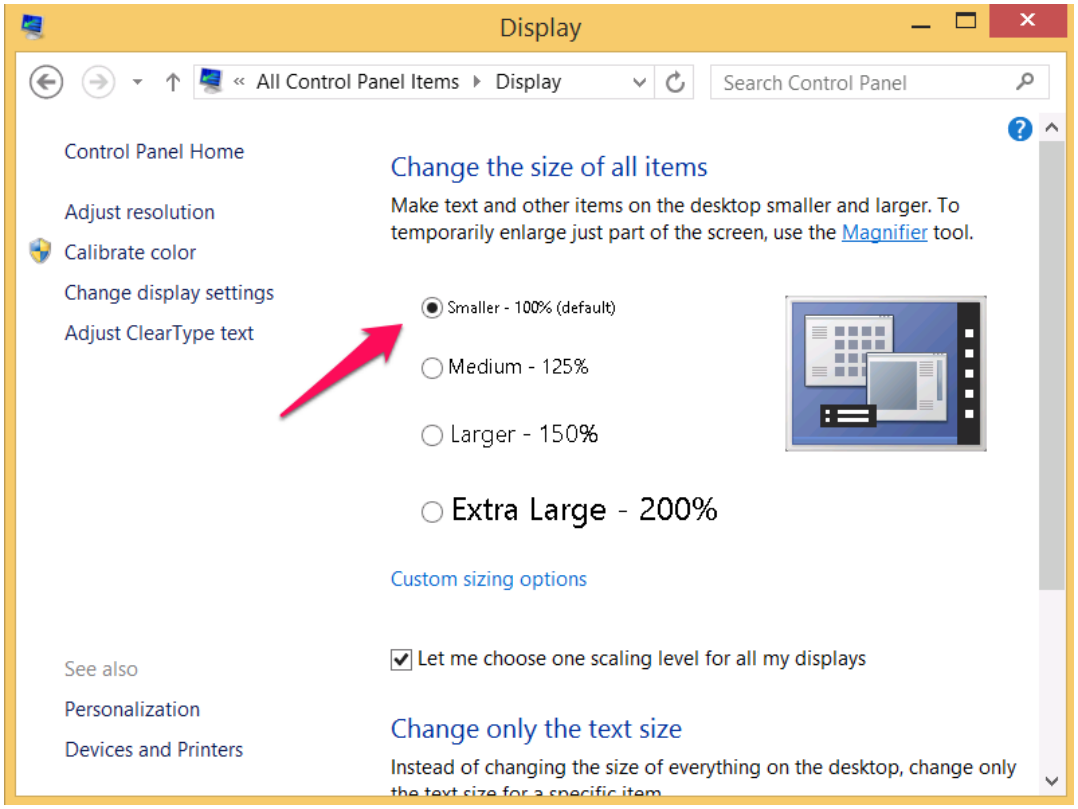
This topic describes the following known issue of Acumatica Test SDK:

- Tests Incorrectly Click or Don't Click Elements on a Form

Tests Incorrectly Click or Don't Click Elements on a Form

One of the possible reasons of such behavior is the display scale setting in Windows OS.

To fix the issue, in the Control Panel, select All Control Panel Items > Display, and set the display scale to 100%, as shown in the following screenshot.



How to Use Page Wrapper Generation Tool

The page wrapper generation tool is a tool that you can use to generate page wrappers for the Acumatica forms whose functionality you want to test.

Parts of the Page Wrapper Generation Tool

The page wrapper generation tool consists of two parts:

- ClassGenerator.exe: The assembly that you run to generate page wrappers. It searches for ASPX and RPX files in the directory that you specify by the conditions that you specify. Also, the assembly moves the Host.dll file from the working directory of ClassGenerator.exe to the Acumatica instance directory that you specify. The Acumatica instance directory must be accessible by the file system.
- Host.dll: The working assembly. It is moved to the Acumatica instance directory by ClassGenerator.exe during page wrapper generation. The working assembly processes requests from ClassGenerator.exe, uses the ASP.NET infrastructure to get instances of Acumatica page classes, parses them, generates the wrapper classes, and saves the classes in the directory that you specify.

ClassGenerator.exe uses requests, such as `<Instance path>/Pages/PM/PM503000.aspx`. You can find this request in the `ScreenUrl` property of the generated class. If something goes wrong during page wrapper generation, try to open the form by using this URL in a browser. You can find more information on page wrapper generation errors in Common Errors That Can Occur During Page Wrapper Generation.

Launch Modes

You can run page wrapper generation tool as a separate application or use it though the API. The following recommendations apply:

- If you run ClassGenerator.exe as a separate application, you need to specify its parameters in the configuration file. For details, see How to Change the Settings of the Page Wrapper Generation Tool.
- If you use ClassGenerator.exe through the API, you need to add a reference to it from your .Net Framework code and create an instance of the `ClassGenerator.ClassGenerator` class. For details, see How to Generate Page Wrappers Programmatically.

How to Generate Page Wrappers Using Test SDK API

You can use the page wrapper generation tool through the API. To do this, add a reference to ClassGenerator.exe to your project, create an instance of the `ClassGenerator.ClassGenerator` class and generate the wrappers, as shown in the code fragment below.

```
ClassGenerator.ClassGenerator WG = new ClassGenerator.ClassGenerator("C:\Program Files (x86)\Acumatica ERP\demo", "C:\Output"));
WG.Run("GL301000, GL501000");
```

To use the page wrapper generation tool (ClassGenerator.exe) through the API, do the following:

1. Add to your project a reference to ClassGenerator.exe.
2. Create an instance of the `ClassGenerator.ClassGenerator` class, as shown in the following code fragment. The constructor takes as input arguments the path to the Acumatica ERP instance and the path to the folder where the generated page wrappers should be saved.

   ```
   ClassGenerator.ClassGenerator WG = new ClassGenerator.ClassGenerator("C:\Program Files (x86)\Acumatica ERP\demo", "C:\Output");
   ```

3. Specify the username that should be used to log in to the Acumatica ERP instance, as shown in the following code fragment. You can omit this step if the user is admin.

   ```
   WG.Username = "Simpson";
   ```

4. Specify the namespace that should be used for generated classes, as shown in the following code fragment. By default, the *`GeneratedWrappers.Acumatica`* namespace is used. We recommend that you use the following format for the namespace name: GeneratedWrappers.<OEMName>.

   ```
   WG.Namespace = "GeneratedWrappers.Acumatica";
   ```

5. If you need to create multiple wrappers for one form, specify a postfix for wrapper files and class names to distinguish the wrappers, as shown in the following code.

   ```
   WG.Postfix = "INT";
   ```

6. If you need to specify the forms for which you do not need to generate wrappers, add these forms to the list of forms that should be excluded from page wrapper generation as follows.

   ```
   WG.ExceptedScreens.Add("SM204520");
   ```

7. If you need to use the specific URL of a page, you can use one of the approaches listed below:
   - If you need to specify particular entries for a form, do this as follows.

     ```
     WG.SpecificEntries.Add("CR306010", "?TaskID=29&RefNoteID=764");
     ```

   - If you need to generate wrappers for a generic inquiry, add the inquiry to the list of generic inquiries for generation.

     ```
     WG.GIs.Add("GI000001", "?Name=Currency Rates History");
     ```

   - In the situations described above or in other situations when you need to specify particular URL of a page, add the page for generation as follows. The example below adds analytical reports from demo data for generation. (This is the only way to generate wrappers for these pages.)

```
WG.AddPage("GL.63.40.00", "~/Frames/RMLauncher.aspx?ID=DBSP.rpx");
WG.AddPage("GL.63.45.00", "~/Frames/RMLauncher.aspx?ID=DBS.rpx");
WG.AddPage("GL.63.50.00", "~/Frames/RMLauncher.aspx?ID=DPL.rpx");
WG.AddPage("GL.63.55.00", "~/Frames/RMLauncher.aspx?ID=DPLP.rpx");
WG.AddPage("GL.63.60.00", "~/Frames/RMLauncher.aspx?ID=DPLQ.rpx");
WG.AddPage("GL.63.65.00", "~/Frames/RMLauncher.aspx?ID=DCF.rpx");
```

8. Start page wrapper generation in one of the following ways.

    - If you have added all forms for which you need to generate page wrappers by the specific URL, as described in the previous step, run page wrapper generation as follows.

        ```
        WG.Run();
        ```

    - If you want to generate wrappers for ASPX and RPX files in the folder of the Acumatica ERP instance, run page wrapper generation as follows.

        ```
        WG.Run("*.*px");
        ```

    - If you want to generate wrappers for particular forms, run page wrapper generation as follows.

        ```
        WG.Run("AU203002,SM204520");
        ```

How to Work With Errors That Occur During Page Wrapper Generation

In this topic, you can find the following common errors that can occur during page wrapper generation and recommendations on how to fix them:

- Error: Value cannot be null
- Error: Request is not available in this context
- Error: Object reference not set to an instance of an object.
- Error: The 'SkinId' property cannot be changed dynamically if Page has a stylesheet theme. For dynamic controls, set the property before calling ApplyStyleSheetSkin().
- Error: Unable to find assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e'.
- Error: Object reference not set to an instance of an object. for a Custom Application
- Error: Type 'PX.Data.PXNotEnoughRightsException' in Assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e' is not marked as serializable.

Error: Value cannot be null

The error Value cannot be null can occur during page wrapper generation. The stack trace of the error is shown below.

```
Process DONE - GL301000
```

Value cannot be null

```
System.Exception: Unable to get screen GL301000 with url /Pages/GL/GL301000.aspx --->
System.ArgumentNullException: Value cannot be null.

Parameter name: Unable to receive Request Handler: (Page)((HttpContext)ar.AsyncState).Handler is null,
verify screen existing.
   at ClassGenerator.Host.MyProxy.GetPage(IAsyncResult ar)
   at ClassGenerator.Host.MyProxy.<>c__DisplayClass7_0.<Process>b__0(IAsyncResult ar)
   at System.Web.HttpAsyncResult.Complete(Boolean synchronous, Object result, Exception error,
RequestNotificationStatus status)
   at System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(Exception error)
   at System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(HttpContext context,
AsyncCallback cb, Object extraData)
   at ClassGenerator.Host.MyProxy.ProcessRequest(HttpContext ctx, AsyncCallback cb)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user, String screenID, String genResultPath,
String postfix, String Namespace, Boolean corectCTL01)
   --- End of inner exception stack trace ---

Server stack trace:
   at ClassGenerator.Host.MyProxy.Process(String uri, String user, String screenID, String genResultPath,
String postfix, String Namespace, Boolean corectCTL01)
   at System.Runtime.Remoting.Messaging.StackBuilderSink._PrivateProcessMessage(IntPtr md, Object[] args,
Object server, Object[]& outArgs)
   at System.Runtime.Remoting.Messaging.StackBuilderSink.SyncProcessMessage(IMessage msg)

Exception rethrown at [0]:
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user, String screenID, String genResultPath,
String postfix, String Namespace, Boolean corectCTL01)
   at ClassGenerator.ScreenExtractor.GetScreen(ScreenMeta ScreenMeta) in
E:\Bld\AC-TESTSDK2018R101-JOB1\tests\Selenium\ClassGenerator\ClassGenerator\ScreenExtractor.cs:line 42
   at ClassGenerator.ClassGenerator.Run() in
E:\Bld\AC-TESTSDK2018R101-JOB1\tests\Selenium\ClassGenerator\ClassGenerator\ClassGenerator.cs:line 183
```

Solution

Remove Telemetry dlls from the site's bin folder, disable Request Profiler on SM205070

Error: Request is not available in this context

The error Request is not available in this context can occur during page wrapper generation. The stack trace of the error is shown below.

```
Process DONE - CS206000
```

Request is not available in this context

```
System.Exception: Unable to get screen cs206000 with url /Pages/CustomFolder/CS/cs206000.aspx --->
System.Web.HttpException: Request is not available in this context
   at System.Web.UI.Page.get_Request()
   at PX.Web.UI.PXSmartPanel.GetVisibleFromRequest() in
C:\BuildAgent\work\3942f517507dc821\code\NetTools\PX.Web.UI\Controls\Containers\SmartPanel.cs:line 2077
   at PX.Web.UI.PXSmartPanel.OnInit(EventArgs e) in
C:\BuildAgent\work\3942f517507dc821\code\NetTools\PX.Web.UI\Controls\Containers\SmartPanel.cs:line 1134
   at System.Web.UI.Control.InitRecursive(Control namingContainer)
   at System.Web.UI.Control.InitRecursive(Control namingContainer)
   at System.Web.UI.Control.InitRecursive(Control namingContainer)
   at System.Web.UI.Control.InitRecursive(Control namingContainer)
   at System.Web.UI.Control.AddedControl(Control control, Int32 index)
   at System.Web.UI.ControlCollection.Add(Control child)
   at PX.Web.UI.PXFormView.CreateChildControls(IEnumerable dataSource, Boolean dataBinding) in
C:\BuildAgent\work\3942f517507dc821\code\NetTools\PX.Web.UI\Controls\Containers\FormView.cs:line 627
   at System.Web.UI.WebControls.CompositeDataBoundControl.CreateChildControls()
   at PX.Web.UI.PXBoundPanel.CreateChildControls() in
C:\BuildAgent\work\3942f517507dc821\code\NetTools\PX.Web.UI\Controls\Containers\BoundPanel.cs:line 1757
   at System.Web.UI.Control.EnsureChildControls()
   at System.Web.UI.WebControls.CompositeDataBoundControl.get_Controls()
   at ClassGenerator.Host.ControlSearcher`1.FindControls(Control control)
   at ClassGenerator.Host.ControlSearcher`1.FindControls(Control control)
   at ClassGenerator.Host.ControlSearcher`1.FindControls(Control control)
   at ClassGenerator.Host.ControlSearcher`1.FindControls(Control control)
   at ClassGenerator.Host.ControlSearcher`1.FindControls(Control control)
   at ClassGenerator.Host.ControlSearcher`1..ctor(Control control)
   at ClassGenerator.Host.Helper.setToolbarUpdatable(Page page)
   at ClassGenerator.Host.MyProxy.GetPage(IAsyncResult ar)
   at ClassGenerator.Host.MyProxy.<>c__DisplayClass7_0.<Process>b__2(IAsyncResult ar)
   at System.Web.HttpAsyncResult.Complete(Boolean synchronous, Object result, Exception error,
RequestNotificationStatus status)
   at System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(Exception error)
   at System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(HttpContext context,
AsyncCallback cb, Object extraData)
   at ClassGenerator.Host.MyProxy.ProcessRequest(HttpContext ctx, AsyncCallback cb)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user, String screenID, String genResultPath,
String postfix, String Namespace, Boolean corectCTL01)
   --- End of inner exception stack trace ---

Server stack trace:
   at ClassGenerator.Host.MyProxy.Process(String uri, String user, String screenID, String genResultPath,
String postfix, String Namespace, Boolean corectCTL01)
   at System.Runtime.Remoting.Messaging.StackBuilderSink._PrivateProcessMessage(IntPtr md, Object[] args,
Object server, Object[]& outArgs)
   at System.Runtime.Remoting.Messaging.StackBuilderSink.SyncProcessMessage(IMessage msg)

Exception rethrown at [0]:
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user, String screenID, String genResultPath,
String postfix, String Namespace, Boolean corectCTL01)
   at ClassGenerator.ScreenExtractor.GetScreen(ScreenMeta ScreenMeta) in
E:\Bld\AC-TSDK60U1-JOB1\tests\Selenium\ClassGenerator\ClassGenerator\ScreenExtractor.cs:line 42
   at ClassGenerator.ClassGenerator.Run(List`1 ScreensMetadata) in
E:\Bld\AC-TSDK60U1-JOB1\tests\Selenium\ClassGenerator\ClassGenerator\ClassGenerator.cs:line 158
```

Solution

- You can not neither open the screen in browser nor generate page wrapper for it. Pleas fixe your application to let the browser open it and run wrapper generation again.
- You can face this error running Windows 8 OS - request maybe lost specifically on this OS due to related MS bug. Please change OS and run wrapper generation again.
- You have more than 1 company in your database but there is no company specified in the Config.xml you use while generating page wrappers. Please join user name with company name in your Config.xml file as follows: <add key="Username" value="admin@COMPANY_NAME"/>.

Error: Object reference not set to an instance of an object.

The error Object reference not set to an instance of an object. can occur during page wrapper generation. The stack trace of the error is shown below.

```
Process page /Pages/MODULE/XX000000.aspx
Process DONE - XX000000 in 4 ms
Process FAIL
Object reference not set to an instance of an object.

Server stack trace:
   at PX.Web.UI.PXBaseDataSource.GetPrimaryActions()
   at PX.Web.UI.PXBaseDataSource.CreateChildControls()
   at System.Web.UI.Control.EnsureChildControls()
   at PX.Web.UI.PXBaseDataSource.get_Controls()
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1 unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1 unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1 unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1 unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.GetDataControls(Page page, List`1 unboundControls, List`1 unboundIDs, List`1 graphViewOrder)
   at PX.Web.UI.ControlHelper.GetDataControls(Page page, List`1 unboundControls)
   at ClassGenerator.Host.MyProxy.Callback(IAsyncResult ar)
   at System.Web.HttpAsyncResult.Complete(Boolean synchronous, Object result, Exception error, RequestNotificationStatus status)
   at System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(Exception error)
   at System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(HttpContext context, AsyncCallback cb, Object extraData)
   at ClassGenerator.Host.MyProxy.process(String uri, String user, AsyncCallback cb)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at System.Runtime.Remoting.Messaging.StackBuilderSink._PrivateProcessMessage(IntPtr md, Object[] args, Object server, Object[]& outArgs)
   at System.Runtime.Remoting.Messaging.StackBuilderSink.SyncProcessMessage(IMessage msg)

Exception rethrown at [0]:
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at ClassGenerator.Program.Main(String[] args)
```

Solution

The form for which you are generating a page wrapper requires a parameter to open. You need to generate wrappers for such forms in a special way. If the form is a generic inquiry, see How to Generate Wrappers for Generic Inquiries with Parameters for more information. For information on generating page wrappers for other pages with parameters, see How to Generate Wrappers for Forms with Parameters.

Error: The 'SkinId' property cannot be changed dynamically if Page has a stylesheet theme. For dynamic controls, set the property before calling ApplyStyleSheetSkin().

The error The 'SkinId' property cannot be changed dynamically if Page has a stylesheet theme. For dynamic controls, set the property before calling ApplyStyleSheetSkin(). can occur during page wrapper generation. The stack trace of the error is shown below.

```
Process page/Pages/MODULE/XX000000.aspx
Process DONE - XX000000 in 10 ms
Process FAIL
The 'SkinId' property cannot be changed dynamically if Page has a stylesheet theme. For dynamic controls, set the property before calling ApplyStyleSheetSkin().

Server stack trace:
   at System.Web.UI.Control.set_SkinID(String value)
   at PX.Web.UI.PXToolBar.set_SkinID(String value)
   at PX.Web.UI.PXBaseDataSource.CreateChildControls()
   at System.Web.UI.Control.EnsureChildControls()
   at PX.Web.UI.PXBaseDataSource.get_Controls()
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1 unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1 unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
```

```
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.GetDataControls(Page page, List`1 unboundControls, List`1 unboundIDs, List`1
graphViewOrder)
   at PX.Web.UI.ControlHelper.GetDataControls(Page page, List`1 unboundControls)
   at ClassGenerator.Host.MyProxy.Callback(IAsyncResult ar)
   at System.Web.HttpAsyncResult.Complete(Boolean synchronous, Object result, Exception error,
RequestNotificationStatus status)
   at System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(Exception error)
   at System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(HttpContext context,
AsyncCallback cb, Object extraData)
   at ClassGenerator.Host.MyProxy.process(String uri, String user, AsyncCallback cb)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at System.Runtime.Remoting.Messaging.StackBuilderSink._PrivateProcessMessage(IntPtr md, Object[] args,
Object server, Object[]& outArgs)
   at System.Runtime.Remoting.Messaging.StackBuilderSink.SyncProcessMessage(IMessage msg)

Exception rethrown at [0]:
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at ClassGenerator.Program.Main(String[] args)
```

Solution

Check whether you can open the form in the user interface of your Acumatica application. This usually indicates that there is some error in your Acumatica application code, and that you neither can open this form in the user interface of the application nor generate a page wrapper for the form.

Error: Unable to find assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e'.

The error Unable to find assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e'. can occur during page wrapper generation. The stack trace of the error is shown below.

```
Process page /Pages/MODULE/XX000000.aspx
Process DONE - XX000000 in 6 ms
Process FAIL
Unable to find assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e'.

Server stack trace:
   at System.Runtime.Serialization.Formatters.Binary.BinaryAssemblyInfo.GetAssembly()
   at System.Runtime.Serialization.Formatters.Binary.ObjectReader.GetType(BinaryAssemblyInfo assemblyInfo,
String name)
   at System.Runtime.Serialization.Formatters.Binary.ObjectMap..ctor(String objectName, String[]
memberNames, BinaryTypeEnum[] binaryTypeEnumA, Object[] typeInformationA, Int32[] memberAssemIds,
ObjectReader objectReader, Int32 objectId, BinaryAssemblyInfo assembly
Info, SizedArray assemIdToAssemblyTable)
   at
System.Runtime.Serialization.Formatters.Binary.__BinaryParser.ReadObjectWithMapTyped(BinaryObjectWithMapTyp
ed record)
   at System.Runtime.Serialization.Formatters.Binary.__BinaryParser.ReadObjectWithMapTyped(BinaryHeaderEnum
binaryHeaderEnum)
   at System.Runtime.Serialization.Formatters.Binary.__BinaryParser.Run()
   at System.Runtime.Serialization.Formatters.Binary.ObjectReader.Deserialize(HeaderHandler handler,
__BinaryParser serParser, Boolean fCheck, Boolean isCrossAppDomain, IMethodCallMessage methodCallMessage)
   at System.Runtime.Serialization.Formatters.Binary.BinaryFormatter.Deserialize(Stream
serializationStream, HeaderHandler handler, Boolean fCheck, Boolean isCrossAppDomain, IMethodCallMessage
methodCallMessage)
   at System.Runtime.Remoting.Channels.CrossAppDomainSerializer.DeserializeObject(MemoryStream stm)
   at System.Runtime.Remoting.Messaging.SmuggledMethodReturnMessage.FixupForNewAppDomain()
   at System.Runtime.Remoting.Channels.CrossAppDomainSink.SyncProcessMessage(IMessage reqMsg)

Exception rethrown at [0]:
```

```
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at ClassGenerator.Program.Main(String[] args)
```

Solution

There are two possible reasons for this error:

- Something is wrong in your Acumatica application code, and you can neither open this form in the user interface of the application nor generate a page wrapper for the form. Check whether you can open the form in the user interface of your Acumatica application.
- A user that doesn't exist is specified in the configuration file of the page wrapper generation tool (ClassGenerator.exe.config). Specify an existing user of your Acumatica application in the configuration file. You can find information on how to change the login that is used by the page wrapper generation tool in How to Change the Settings of the Page Wrapper Generation Tool.

Error: Object reference not set to an instance of an object. for a Custom Application

The error Object reference not set to an instance of an object. can occur during page wrapper generation when a page wrapper is generated for a custom application. The stack trace of the error is shown below.

```
Process page/Pages/MODULE/XX000000.aspx
Process DONE - XX000000 in 7 ms
Process FAIL
Object reference not set to an instance of an object.

Server stack trace:
   ... custom aplication code...
   at PX.Data.PXGraph.CreateInstance(Type graphType, String prefix)
   at PX.Web.UI.PXBaseDataSource.CreateDataGraphAsSingleton(Type type)
   at PX.Web.UI.PXBaseDataSource.CreateDataGraph(Type type)
   at PX.Web.UI.PXBaseDataSource.get_DataGraph()
   at PX.Web.UI.PXBaseDataSource.get_ToolBar()
   at PX.Web.UI.PXBaseDataSource.CreateChildControls()
   at System.Web.UI.Control.EnsureChildControls()
   at PX.Web.UI.PXBaseDataSource.get_Controls()
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.EnumDataControls(ControlCollection collection, List`1 dataControls, List`1
unboundControls, List`1 unboundIDs, Boolean collectUnboundControls)
   at PX.Web.UI.ControlHelper.GetDataControls(Page page, List`1 unboundControls, List`1 unboundIDs, List`1
graphViewOrder)
   at PX.Web.UI.ControlHelper.GetDataControls(Page page, List`1 unboundControls)
   at ClassGenerator.Host.MyProxy.Callback(IAsyncResult ar)
   at System.Web.HttpAsyncResult.Complete(Boolean synchronous, Object result, Exception error,
RequestNotificationStatus status)
   at System.Web.HttpApplication.ApplicationStepManager.ResumeSteps(Exception error)
   at System.Web.HttpApplication.System.Web.IHttpAsyncHandler.BeginProcessRequest(HttpContext context,
AsyncCallback cb, Object extraData)
   at ClassGenerator.Host.MyProxy.process(String uri, String user, AsyncCallback cb)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at System.Runtime.Remoting.Messaging.StackBuilderSink._PrivateProcessMessage(IntPtr md, Object[] args,
Object server, Object[]& outArgs)
   at System.Runtime.Remoting.Messaging.StackBuilderSink.SyncProcessMessage(IMessage msg)

Exception rethrown at [0]:
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at ClassGenerator.Program.Main(String[] args)
```

Solution

There are two possible reasons for this error:

- The form for which you are generating a page wrapper requires a parameter to open. You need to generate wrappers for such forms in a special way. If the form is a generic inquiry, see How to Generate Wrappers for Generic Inquiries with Parameters for more information. For information on generating page wrappers for other pages with parameters, see How to Generate Wrappers for Forms with Parameters.
- Something is wrong in your custom Acumatica application code, and you neither can open this form in the user interface of the application nor generate a page wrapper for the form. Check whether you can open the form in the user interface of your Acumatica application.

Error: Type 'PX.Data.PXNotEnoughRightsException' in Assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e' is not marked as serializable.

The error Type 'PX.Data.PXNotEnoughRightsException' in Assembly 'PX.Data, Version=1.0.0.0, Culture=neutral, PublicKeyToken=3b136cac2f602b8e' is not marked as serializable. can occur during page wrapper generation. The stack trace of the error is shown below.

```
Process page /Pages/MODULE/XX000000.aspx
Process DONE - XX000000 in 5 ms
Process FAIL
Type 'PX.Data.PXNotEnoughRightsException' in Assembly 'PX.Data, Version=1.0.0.0, Culture=neutral,
PublicKeyToken=3b136cac2f602b8e' is not marked as serializable.

Server stack trace:
   at System.Runtime.Serialization.Formatters.Binary.WriteObjectInfo.InitSerialize(Object obj,
ISurrogateSelector surrogateSelector, StreamingContext context, SerObjectInfoInit serObjectInfoInit,
IFormatterConverter converter, ObjectWriter objectWriter, Serializati
onBinder binder)
   at System.Runtime.Serialization.Formatters.Binary.WriteObjectInfo.Serialize(Object obj,
ISurrogateSelector surrogateSelector, StreamingContext context, SerObjectInfoInit serObjectInfoInit,
IFormatterConverter converter, ObjectWriter objectWriter, SerializationBi
nder binder)
   at System.Runtime.Serialization.Formatters.Binary.ObjectWriter.Serialize(Object graph, Header[]
inHeaders, __BinaryWriter serWriter, Boolean fCheck)
   at System.Runtime.Serialization.Formatters.Binary.BinaryFormatter.Serialize(Stream serializationStream,
Object graph, Header[] headers, Boolean fCheck)
   at System.Runtime.Remoting.Channels.CrossAppDomainSerializer.SerializeMessageParts(ArrayList
argsToSerialize)
   at System.Runtime.Remoting.Messaging.SmuggledMethodReturnMessage..ctor(IMethodReturnMessage mrm)
   at System.Runtime.Remoting.Messaging.SmuggledMethodReturnMessage.SmuggleIfPossible(IMessage msg)
   at System.Runtime.Remoting.Channels.CrossAppDomainSink.DoDispatch(Byte[] reqStmBuff,
SmuggledMethodCallMessage smuggledMcm, SmuggledMethodReturnMessage& smuggledMrm)
   at System.Runtime.Remoting.Channels.CrossAppDomainSink.DoTransitionDispatchCallback(Object[] args)

Exception rethrown at [0]:
   at System.Runtime.Remoting.Proxies.RealProxy.HandleReturnMessage(IMessage reqMsg, IMessage retMsg)
   at System.Runtime.Remoting.Proxies.RealProxy.PrivateInvoke(MessageData& msgData, Int32 type)
   at ClassGenerator.Host.MyProxy.Process(String uri, String user)
   at ClassGenerator.Program.Main(String[] args)
```

Solution

Check whether the user that is used by the page wrapper generation tool for logging in to your Acumatica application has sufficient rights to view the form. Check whether the page is configured as visible in your Acumatica application.

How to Change the Settings of the Page Wrapper Generation Tool

- ClassGenerator.exe.config
- PagesList.txt
- PagesWithParameters.txt
- GenericInquiriesWithParameters.txt

ClassGenerator.exe.config

You can change the settings of the page wrapper generation tool (ClassGenerator.exe) in the ClassGenerator.exe.config file. The structure of this file is shown below.

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <appSettings>
        <!--Local path to the Acumatica ERP instance installation directory-->
        <add key="SitePhysicalPath" value="C:\Program Files (x86)\Acumatica ERP\yoursite"/>
        <!--Output directory to store the generated page wrappers-->
        <add key="GenResultPath" value="C:\share\output"/>
        <!--User to be used for page wrapper generation-->
        <add key="UserName" value="admin@Demo"/>
        <!--IDs of the pages you want to run wrapper generation for; the wildcard * is supported-->
        <add key="FileNameFilter" value="CS100000, CS102000, CM202000, GL201500, CS202000, GL202500, GL102000, GL101000, GL201000"/>
        <!--Deletes all files in output directory before running the page wrapper generation process-->
        <add key="ClearOutput" value="true"/>
        <!--Namespace where wrapper classes will be defined. Use the template "GeneratedWrappers.<PartnerName>".-->
        <add key="Namespace" value="GeneratedWrappers.Acumatica"/>
    </appSettings>
</configuration>
```

You can use the following keys in the ClassGenerator.exe.config file to configure the generation of page wrappers.

#	Key Name	Key Description	Mandatory/ Optional	Usage Example
1	`SitePhysicalPath`	Specifies the local path to the installation directory of an Acumatica ERP instance.	Mandatory	`<add key="SitePhysicalPath" value="C:\Program Files (x86)\Acumatica ERP\demo"/>`
2	`GenResultPath`	Specifies the directory where the generated page wrappers should be saved.	Mandatory	`<add key="GenResultPath" value="C:\Output"/>`
3	`UserName`	Specifies the user name to be used to log in to the Acumatica ERP instance. If you need to log in to a specific company, you should specify the user name in the following format: `UserName@CompanyName`, where `UserName` is replaced with the name of the user, and `CompanyName` is replaced with the name of the company to which you want to log in.	Mandatory	`<add key="UserName" value="admin"/>` `<add key="UserName" value="admin@Company"/>`

4	`FileNameFilter`	Specifies the IDs of the forms for which you want to generate page wrappers.
		You can use two-letter prefix of the Acumatica ERP module name as the value of this key to generate wrappers for all forms of the module.
		You can use * as the value of this key to generate wrappers for all pages.
			Mandatory	`<add key="FileNameFilter" value="GL301000, GL501000"/>`
				`<add key="FileNameFilter" value="GL301000, CR"/>`
				`<add key="FileNameFilter" value="*"/>`

5	`PagesList`	Specifies the file that contains the list of IDs of the forms that should be included in or excluded from the page wrapper generation.	Optional	`<add key="PagesList" value="PagesList.txt"/>`
6	`PagesListAttribute`	Specifies whether the forms that are specified in the `PagesList` key should be included in or excluded from the page wrapper generation. You can set the value of this key to `include` to include the forms in the page wrapper generation. You can set the value of this key to `exclude` to exclude the forms from the page wrapper generation.	Optional	`<add key="PagesListAttribute" value="include"/>` `<add key="PagesListAttribute" value="exclude"/>`
7	`ClearOutput`	Specifies whether all files in the output directory,	Mandatory	`<add key="ClearOutput" value="true"/>`

		which is specified in the `GenResultPath` key, should be removed before the page wrapper generation starts. You can set the value of this key to `true` to remove all files from the output folder before page wrapper generation. You can set the value of this key to `false` to not delete the files from the output folder before page wrapper generation.		`<add key="ClearOutput" value="false"/>`
8	`Namespace`	Specifies the namespace where page wrapper classes should be defined. We recommend that you use the following template for the namespace name: `GeneratedWrappers.<PartnerName>`.	Optional	`<add key="Namespace" value="GeneratedWrappers.Acumatica"/>`
9	`PagesParameters`	Specifies the file that contains the list of forms that require some parameters to open.	Optional	`<add key="PagesParameters" value="PagesWithParameters.txt"/>`
10	`GenericInquiryParameters`	Specifies the file that contains the list of generic inquiries that require some parameters to open.	Optional	`<add key="GenericInquiryParameters" value="GenericInquiriesWithParameters.txt"/>`
11	`CorrectCTL01`	Specifies whether the entries of `ctl01` should be replaced with `ctl00` in the generated page wrappers. You can set the value of this key to `true` to replaces entries of `ctl01` with `ctl00` in the generated page wrappers. You can set the value of this key to `false` to not replace entries of `ctl01` with `ctl00` in the generated page wrappers.	Optional	`<add key="CorrectCTL01" value="true"/>` `<add key="CorrectCTL01" value="false"/>`

PagesList.txt

```
<add key="PagesList" value="PagesList.txt" />
```

Specifies the file that contains the list of IDs of the forms that should be included in or excluded from the page wrapper generation.

Example

```
CS100000
CS102000
GL201500
GL301000
GL501000
```

PagesWithParameters.txt

```
<add key="PagesParameters" value="PagesWithParameters.txt"/>
```

Specifies the file that contains the list of forms that require some parameters to open.

Example

```
CR306010 ?TaskID=29&RefNoteID=764
CR306015 ?TaskID=30&RefNoteID=764&NotificationID=4517FCC3-98A7-4521-B4C7-1BD80B2846E6
```

GenericInquiriesWithParameters.txt

```
<add key="GenericInquiryParameters" value="GenericInquiriesWithParameters.txt"/>
```

Specifies the file that contains the list of generic inquiries that require some parameters to open.

Example

```
GI000001 ?Name=Currency Rates History
CR3010PL ?ID=20e4ab0d-0631-4759-a48d-6ec20ac78291
CR3060PL ?ID=24d48139-cf11-4be6-9bd3-57ee86893778
CR3040PL ?ID=a95a50d6-6892-4052-b6aa-8e769efa2bfc
CR3080PL ?ID=6b673610-9cee-46c3-adb4-1e1569ddbb0b
CR3020PL ?ID=d345a840-d1cf-4d6f-a2df-a454b85b20d8
CR3030PL ?ID=df95d4e3-fb8c-4aff-ba9a-f8371e7b1908
CR2040PL ?ID=45c1d74f-f7b4-498d-bee2-416863b35196
CR2020PL ?ID=d7dfa28a-36ad-467c-addb-c2080eda3484
```

How to Change Browser Settings

If you want to use custom browser configuration settings, you can change them as described in this topic.

Below you can find the following information on changing browser settings:

- Changing Browser Profile Preferences in Code
- Reseting Browser Settings to Default in Code

Changing Browser Profile Preferences in Code

You can change the browser profile preferences in code by using the `Browser.ProfileBuilder` property.

The code below shows an example of the browser settings being changed. After this code is executed, the browser restarts with the new profile.

```
Browser.ProfileBuilder.SetPreference("browser.download.dir", @"C:\download").ApplyProfile();
```

Reseting Browser Settings to Default in Code

The code below shows how to reset the browser profile to use the default settings. After this code is executed, the browser restarts with the new profile.

```
Browser.ProfileBuilder.ResetProfile().ApplyProfile();
```

How to Change Chrome Settings

If you want to use custom Chrome configuration settings, you can change them as described in this topic.

Below you can find the following information on changing Chrome settings:

- Changing Chrome Profile Preferences in Code
- Reseting Chrome Settings to Default in Code

Changing Chrome Profile Preferences in Code

You can change the Chrome profile preferences in code by using the `Browser.ProfileBuilder` property.

The code below shows an example of changing the Chrome settings. After this code is executed, the browser restarts with the new profile.

```
Browser.ProfileBuilder.SetPreference("download.default_directory", @"C:\download").ApplyProfile();
```

Reseting Chrome Settings to Default in Code

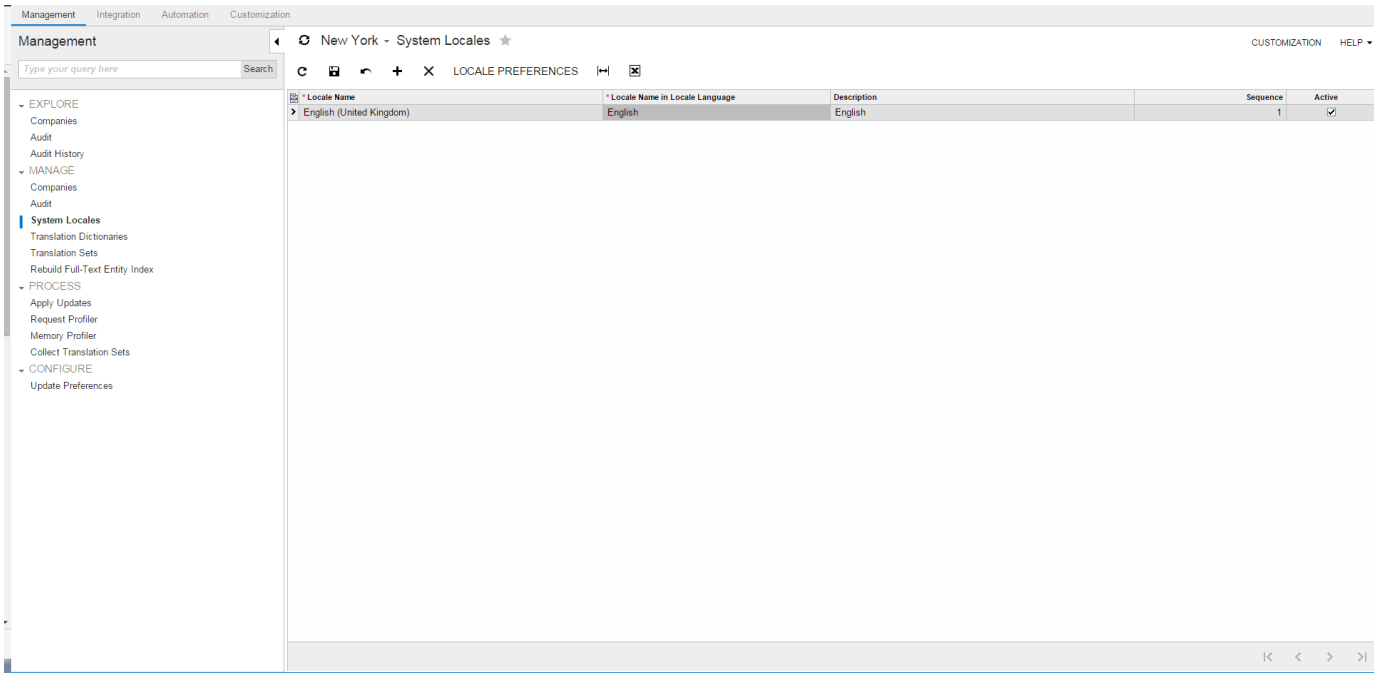
The code below shows how to reset the browser profile to use the default settings. After this code is executed, the browser restarts with the new profile.

```
Browser.ProfileBuilder.ResetProfile().ApplyProfile();
```

How to Change Culture Settings

Dates and other country-specific data that you enter into Acumatica ERP by using Test SDK are sensitive to the culture settings. For dates and other country-specific data to be recognized correctly by Acumatica ERP, the Test SDK culture must be the same as the culture that is specified in Acumatica ERP.

By default, Acumatica Test SDK uses the English (United States) culture. You can find out which culture is used in Acumatica ERP on the System Locales form (SM200550; System > Management > Manage), as shown in the following screenshot.



To change the culture that is used in Test SDK, you should specify the value in the `<default_culture_info>` tag in the Config.xml file. By default, this tag is not specified. Consider the following examples of culture settings in the configuration file.

The following configuration makes Test SDK use default (English (United States)) culture.

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns="http://tempuri.org/XMLSchema2.xsd">
   <general>
       <browser>*firefox</browser>
       <!-- Sets the culture to English (United States) -->
       <!-- default_culture_info is not specified-->
       <site_dst>
           <rmhost/>
           <url>http://localhost/testsite</url>
           <login>admin</login>
           <pswd>123</pswd>
           <lang>English</lang>
           <cmpid/>
       </site_dst>
       <logging>
   <logStorage type ="txtfile" level="INFO" outputFolder ="" screenshotActive="true" screenshotOutputFolder ="" />
       </logging>
   </general>
   <testing>
       <Check Name="Test"/>
   </testing>
</config>
```

The following configuration makes Test SDK use English (United Kingdom) culture.

```
<?xml version="1.0" encoding="utf-8"?>
<config xmlns="http://tempuri.org/XMLSchema2.xsd">
   <general>
       <browser>*firefox</browser>
       <!-- Sets the culture to English (United Kingdom) -->
       <default_culture_info>enGB</default_culture_info>
       <site_dst>
           <rmhost/>
           <url>http://localhost/testsite</url>
           <login>admin</login>
           <pswd>123</pswd>
           <lang>English</lang>
           <cmpid/>
       </site_dst>
      <logging>
   <logStorage type ="txtfile" level="INFO" outputFolder ="" screenshotActive="true" screenshotOutputFolder ="" />
      </logging>
   </general>
   <testing>
       <Check Name="Test"/>
   </testing>
</config>
```

How to Change Predefined Timeouts

In Acumatica Test SDK, there are five predefined timeouts. The primary timeout, which is widely used in tests and lasts two minutes by default, is `LongTimeOut`. The lengths of all other timeouts are expressed in relation to the length of `LongTimeOut`. (See the following code fragment for details.)

```
/// <summary>
/// Long timeout: 6 minutes.
/// </summary>
public static int LongTimeOut = 360000;

/// <summary>
/// Medium timeout: 10 seconds.
/// </summary>
public static int MediumTimeOut = LongTimeOut / 36;

/// <summary>
/// Short timeout: 4 seconds.
/// </summary>
public static int ShortTimeOut = LongTimeOut / 90;

/// <summary>
/// Default timeout: 3 seconds.
/// </summary>
public static int DefaultTimeOut = LongTimeOut / 120;
```

The predefined timeouts are used for long-running operations. For example, the code fragment below opens the Journal Transactions form (GL301000), releases a batch, and waits for the long-running release operation to complete within two minutes.

```
var batch = new JournalEntry();
batch.OpenScreen();
batch.ToolBar.Release.WaitAction = Wait.WaitForLongOperationToComplete;
batch.Release();
```

If you need to change the time the test waits for the long-running operation to complete, you can change the default timeout. For example, the code fragment below shows how to make the *`LongTimeOut`* twice as long.

```
var batch = new JournalEntry();

using (new Wait(Wait.LongTimeOut * 2))
{
 batch.OpenScreen();
 batch.ToolBar.Release.WaitAction = Wait.WaitForLongOperationToComplete;
 batch.Release();
}
```

You can set default timeout in the Config.xml file as well:

```
<?xml version="1.0" encoding="utf-8"?>
<config>
    <general>
        ...
        <long_timeout_ms>global timeout in milliseconds</long_timeout_ms>
        ...
    </testing>
    ...
</config>
```

How to Manage Log Providers

Acumatica Test SDK provides logs saved in an HTML file: Screenshots are displayed on the HTML page if screenshot saving is turned on.

Configuring Log Settings

Before running your test, you can specify the settings for the log by using the <logging> tag of the Config.xml file, as shown below. You cannot change the parameters that apply to the saving of logs during test execution.

```
<?xml version="1.0" encoding="utf-8"?>
<config>
    <general>
        <browserbin>C:\TestSDK\Chrome\chrome.exe</browserbin>
  <browser_downloads_folder>c:\share\download\</browser_downloads_folder>
  <default_culture_info>en-US</default_culture_info>
        <site_dst>
            <url>http://localhost/Instance_Name/</url>
            <login>username</login>
            <pswd>password</pswd>
        </site_dst>
  <logging>
      <logStorage type ="htmlfile" level="INFO" outputFolder ="" screenshotActive="true" />
  </logging>
    </general>
    <testing>
        <Check Name="Test"/>
    </testing>
</config>
```

When configuring log settings, you have to specify the following mandatory attributes of each <logStorage> tag:

- `type`: Specifies the type of log storage. Set the attribute to htmlfile to save the log in an HTML file.
- `level`: Specifies which messages should be included in the log. Set the attribute to one of the predefined levels, which are described below in this topic. You can also turn off logging of the current type by setting this attribute to OFF.
- `screenshotActive`: Turns on or off the saving of screenshots for the current type of the log. Set the attribute to true to turn on the saving of screenshots, or to false to turn off the saving of screenshots for the current type of the log. If screenshots are turned off for all types of logs, screenshots won't even be captured. If screenshots are turned on for at least one type of log, then screenshots will be captured and will be saved according to the settings of each log type.

Log Levels

You can specify which messages should be included in the log by specifying one of the log levels, which are described in the table below.

Level	Description	Included Messages	Screenshots
OFF	Turns off logging for the current log type.	No messages are written.	No screenshots are taken.
ERROR	Logs error messages and messages that help you navigate in the log file.	• Check • Testcase • Step • Operation • Action • Error	Only nested screenshots from the Error messages and Controls screenshots are created if the saving of screenshots is turned on.
WARN	Logs error and warning messages, as well as messages that help you navigate in the log file.	• Check • Testcase • Step • Operation • Action • Warning • Error	Only nested screenshots from the Error and Warning messages and Controls screenshots are created if the saving of screenshots is turned on.
INFO	Logs all types of messages (except for the Debug messages). This level is suitable for daily regression tests.	• Check • Testcase • Step • Operation • Action • Screenshot • Information • Warning • Error	All screenshots (except for the nested screenshots of the Debug messages) are created if the saving of screenshots is turned on.

DEBUG	Logs all types of log messages.	• Check • Testcase • Step • Operation • Action • Debug • Screenshot • Information • Warning • Error	All screenshots are created (if the saving of screenshots is turned on).

Log in an HTML File

The HTML log file contains a tree representation of the log. Log messages in the file have colors that depend on their type: Errors are red, warnings are yellow, and all other messages are black. Color highlighting is applied to the parent nodes of the highlighted message. For example, if an error occurs in one step of a test case, then the nodes for the operation that contains the error, the test step, the test case, and the root `Test` node will be colored with red.

If you have configured the log to be saved in HTML format, the test runner saves the following files and the folder for each test in the sequence of tests:

- {test_name}_{start_execution_time}_Log_{level}_{status}.html: Displays the log with the specified level for the test with the specified status of processing. See the description of the test statuses below in this topic.
- {test_name}_{start_execution_time}_Log_{level}.json: Contains the log data.
- {test_name}_{start_execution_time}_{level} folder: Contains screenshots (JPG files) for the test. The folder is created if configuring screenshots is turned on for the log.

Test Statuses

The table below describes the test statuses that can be used in the names of HTML log files.

Status	Description	Example
InProcess	The test is in progress. This status can be combined with the WARN or ERROR status.	• The test is in progress. No error or warning has occurred yet. Test_2016_05_19_03_12_18_Log_INFO_InProcess.html • The test is in progress. At least one warning has occurred. Test_2016_05_19_03_12_18_Log_INFO_InProcess_WARN.html • The test is in progress. At least one error has occurred. Also, there may be at least one warning. Test_2016_05_19_03_12_18_Log_INFO_InProcess_Error.html
PASS	The test has completed. Errors and warnings haven't occurred.	• The test has been passed. Test_2016_05_19_03_15_24_Log_INFO_PASS.html
WARN	A warning has occurred during the test execution. WAR N is added to the name of the log file as soon as the warning log message is produced by the test.	• The test is in progress. At least one warning has occurred. Test_2016_05_19_03_12_18_Log_INFO_InProcess_WARN.html • The test is completed. At least one warning has occurred. Test_2016_05_19_03_12_18_Log_INFO_WARN.html
ERROR	An error has occured during the test execution. ERROR i s added to the name of the log file as soon as the error log message is produced by the test.	• The test is in progress. At least one error has occurred. Test_2016_05_19_03_12_18_Log_INFO_InProcess_ERROR.html • The test is completed. At least one warning has occurred. Also, there may be at least one warning. Test_2016_05_19_03_12_18_Log_INFO_ERROR.html

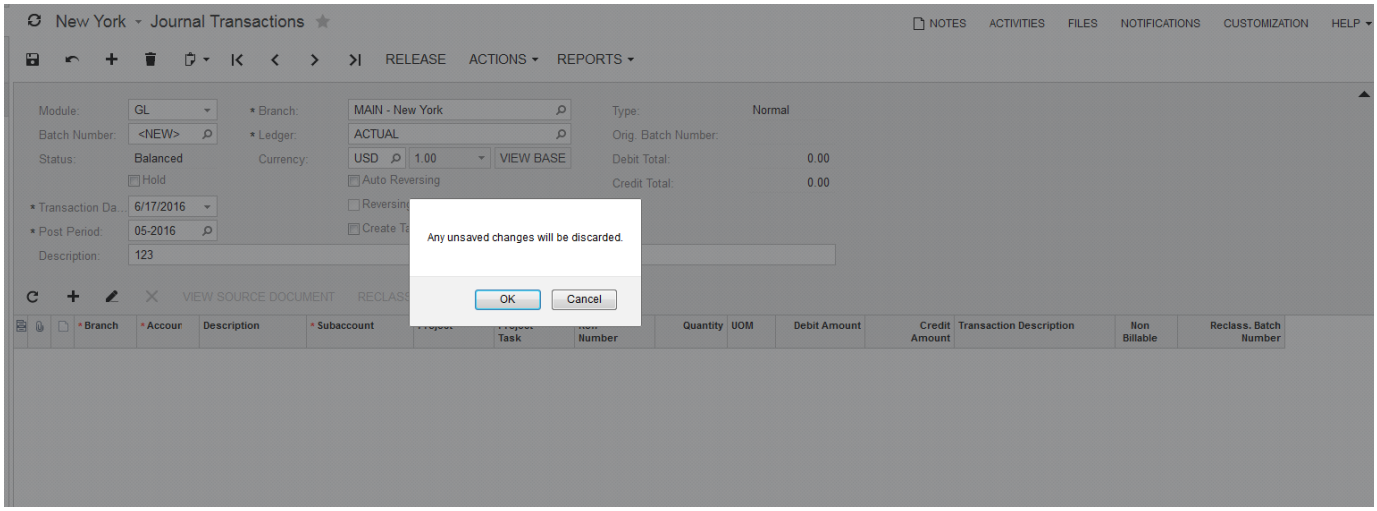
How to Work with Alerts

In this topic, you can find examples of how to work with alerts by using Test SDK. You may need to work with alerts, prompts, and confirmations in the following scenarios, which are described in detail below:

- Navigating Away from The Form Where Unsaved Changes Have Been Made
- Verifying that an Alert Has Been Thrown
- Verifying that the Alert with Exact Message Has Been Thrown

Navigating Away from The Form Where Unsaved Changes Have Been Made

If you navigate away from the form where unsaved changes have been made, the Acumatica application throws the standard confirmation, which is shown in the following screenshot. You need to discard all unsaved changes before navigation.



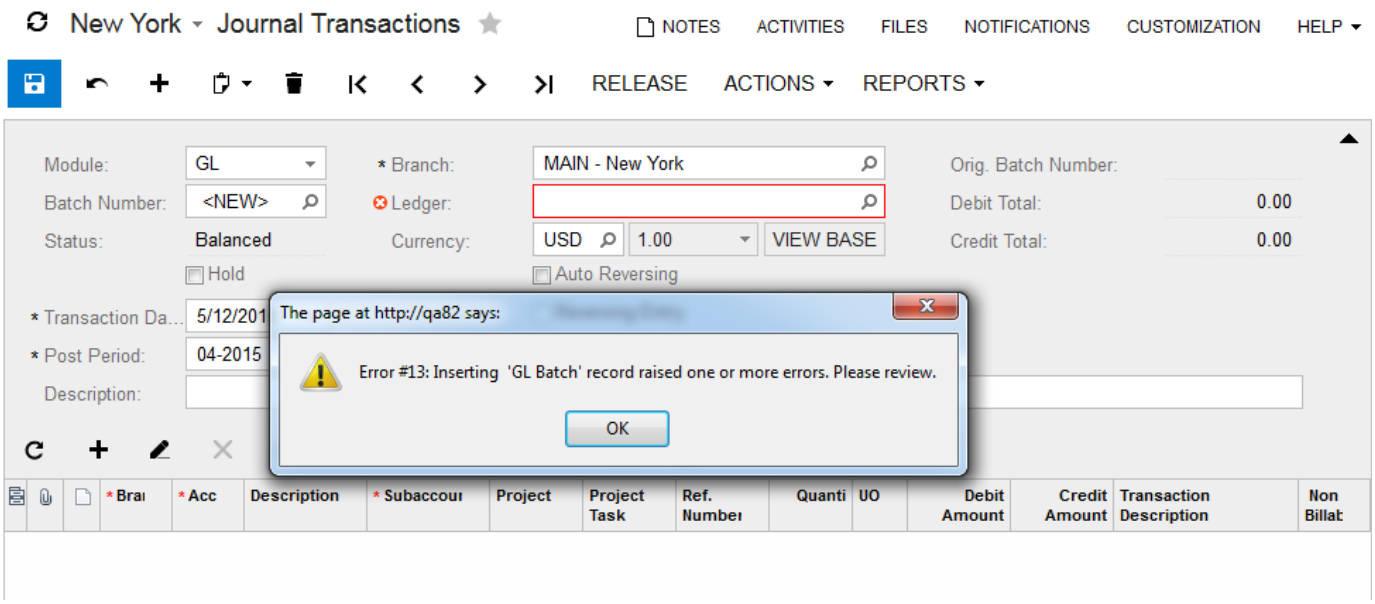
The code fragment below clicks OK in the alert and dismisses it without throwing an exception.

```
JournalEntry batch = new JournalEntry();
batch.OpenScreen();
batch.Insert();
batch.Summary.BranchID.Select("MAIN");
batch.Summary.LedgerID.Reset();
batch.Cancel();
JournalVouchers journalVouchers = new JournalVouchers();
journalVouchers.OpenScreen();
```

Verifying that an Alert Has Been Thrown

If an error message is displayed in your Acumatica application after the test application clicks a button and you need to verify that an alert has been thrown, you can handle the error as described in this scenario.

An example of an error message, which appears if you click Save on the Journal Transactions form (GL301000; Finance > General Ledger > Enter) before filling in all required elements, is shown in the following screenshot.



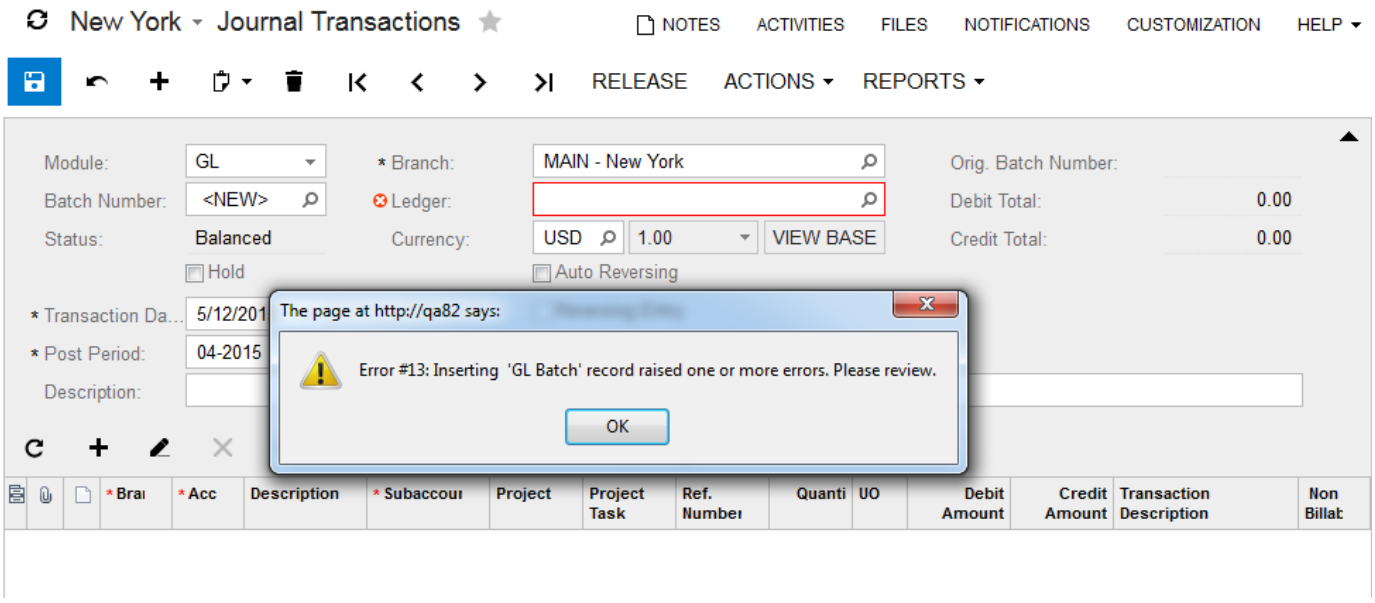
The code fragment below clicks Save on the Journal Transactions form (GL301000), validates that an alert exists, and clicks OK to dismiss it without validation of the text of the alert.

```
JournalEntry batch = new JournalEntry();
batch.OpenScreen();
batch.Insert();
batch.Summary.BranchID.Select("MAIN");
batch.Summary.LedgerID.Reset();
batch.VerifyAlert(batch.Save);
```

Verifying that the Alert with Exact Message Has Been Thrown

If an error message is displayed in your Acumatica application after the test application clicks a button and you need to verify the text of the alert, you can handle the error as described in this scenario.

An example of an error message, which appears if you click Save on the Journal Transactions form (GL301000; Finance > General Ledger > Enter) before filling in all required elements, is shown in the following screenshot.

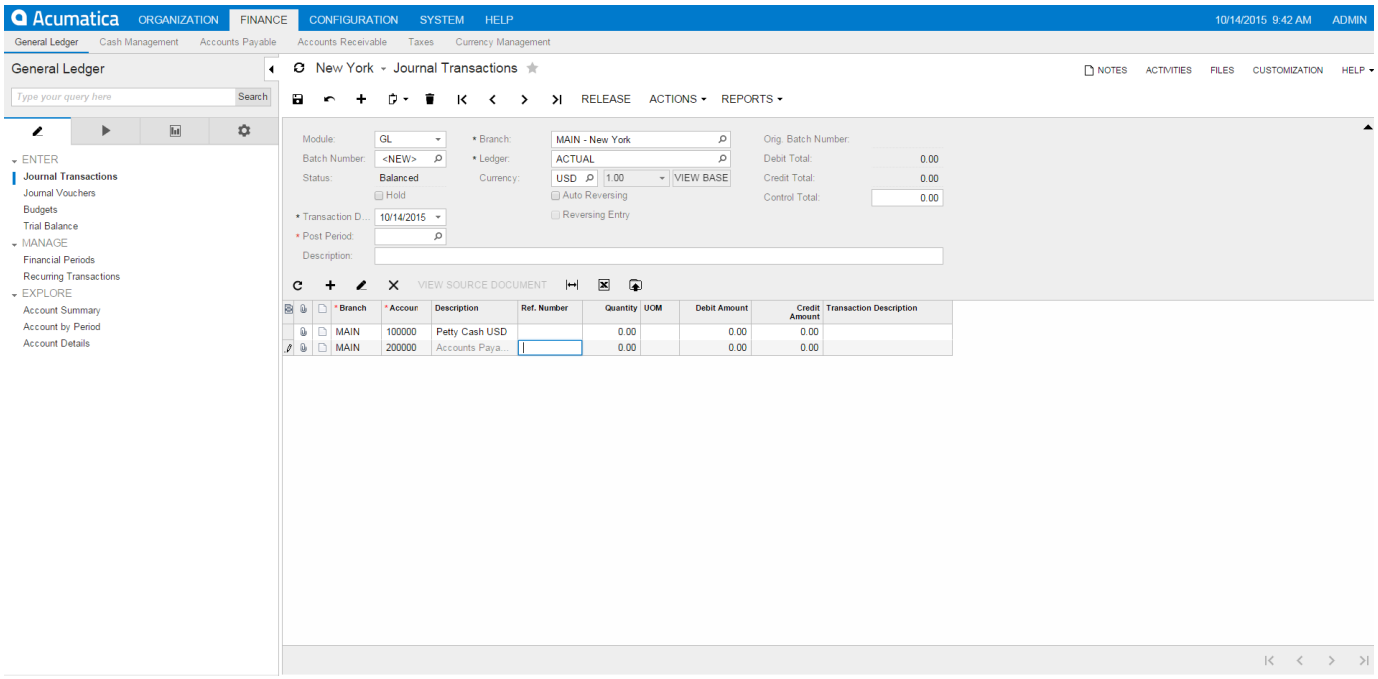


The code fragment below clicks Save on the Journal Transactions form (GL301000), validates that the alert exists, validates its text message, and clicks OK to dismiss the alert.

```
JournalEntry batch = new JournalEntry();
batch.OpenScreen();
batch.Insert();
batch.Summary.BranchID.Select("MAIN");
batch.Summary.LedgerID.Reset();
batch.VerifyAlert(batch.Save, "Error #13: Inserting 'GL Batch' record raised one or more errors. Please review.");
```

How to Commit Changes in Rows of a Detail Table

To commit changes in the rows of a detail table, use the `CommitRows()` method. This method commits all changes made to the rows of the detail table and exits the table editing mode. The following screenshot shows two rows that are committed on the Journal Transactions form (GL301000; Finance > General Ledger > Enter).



The following code shows how to commit all changes made to the rows of the detail table on the Journal Transactions form (GL301000).

```
JournalEntry.OpenScreen();
JournalEntry.Insert();
JournalEntry.Details.New();
JournalEntry.Details.Row.AccountID.Select("100000");
JournalEntry.Details.New();
JournalEntry.Details.Row.AccountID.Select("200000");
JournalEntry.Details.CommitRows();
```

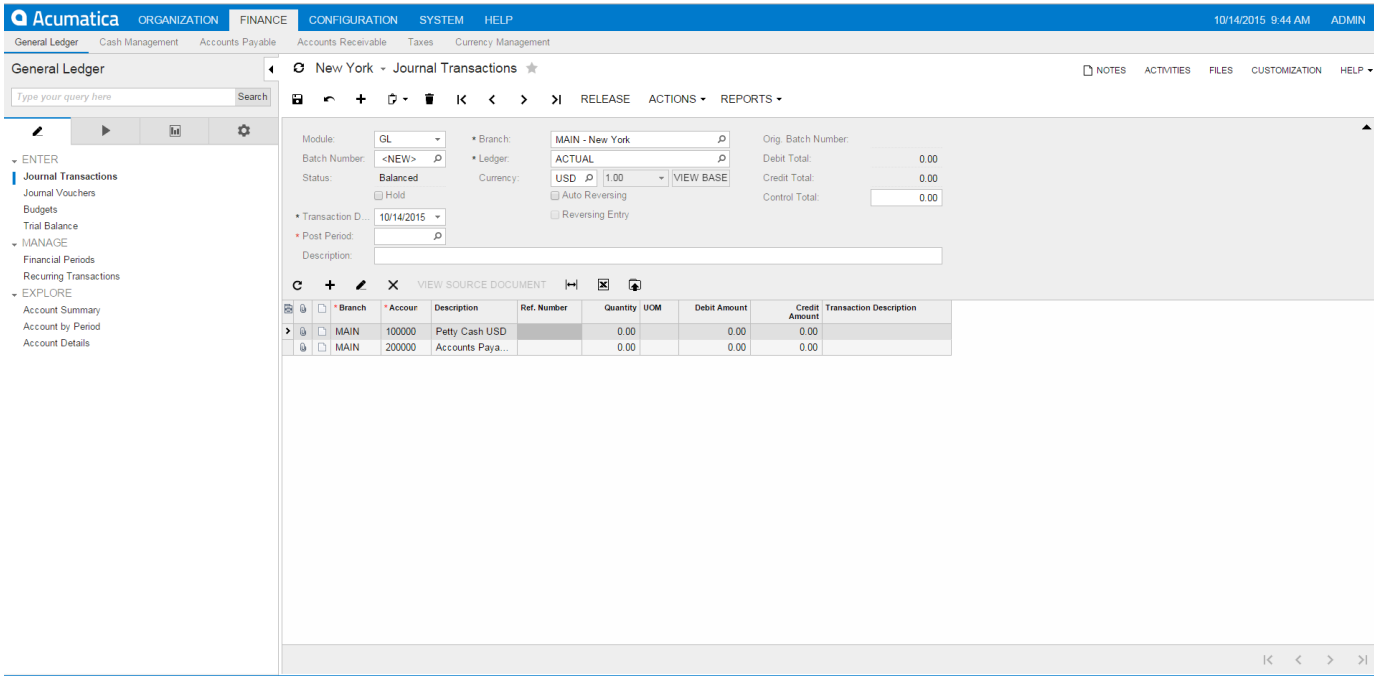
How to Navigate through the Rows of a Detail Table

To navigate through the rows of a detail table, use the `SelectRow()` method. This method can select a record in the detail table by its row number or by a <Column Name>:<Cell Value> pair.

If the specified record cannot be activated (for example, a wrong index or nonexistent <Column Name>:<Cell Value> pair is specified as a parameter of the method), the `SelectRow()` method prints an error message in the log.

> IMPORTANT NOTE
> To select the very first row in a grid use index 1, e.g.: JournalEntry.Details.SelectRow(1);

The following screenshot illustrates the selection of the first row of a detail table on the Journal Transactions form (GL301000; Finance > General Ledger > Enter).



The code below shows how to select the first row of the detail table on the Journal Transactions form (GL301000) by row number:

```
JournalEntry.OpenScreen();
JournalEntry.Insert();
JournalEntry.Details.New();
JournalEntry.Details.Row.AccountID.Select("100000");
JournalEntry.Details.New();
JournalEntry.Details.Row.AccountID.Select("200000");
JournalEntry.Details.SelectRow(1);
```

The code below shows how to select the first row of the detail table on the Journal Transactions form (GL301000) by <Column Name>:<Cell Value> pair:

```
JournalEntry.OpenScreen();
JournalEntry.Insert();
JournalEntry.Details.New();
JournalEntry.Details.Row.AccountID.Select("100000");
JournalEntry.Details.New();
JournalEntry.Details.Row.AccountID.Select("200000");
JournalEntry.Details.SelectRow(JournalEntry.Details.Columns.Description, "Petty Cash USD");
```

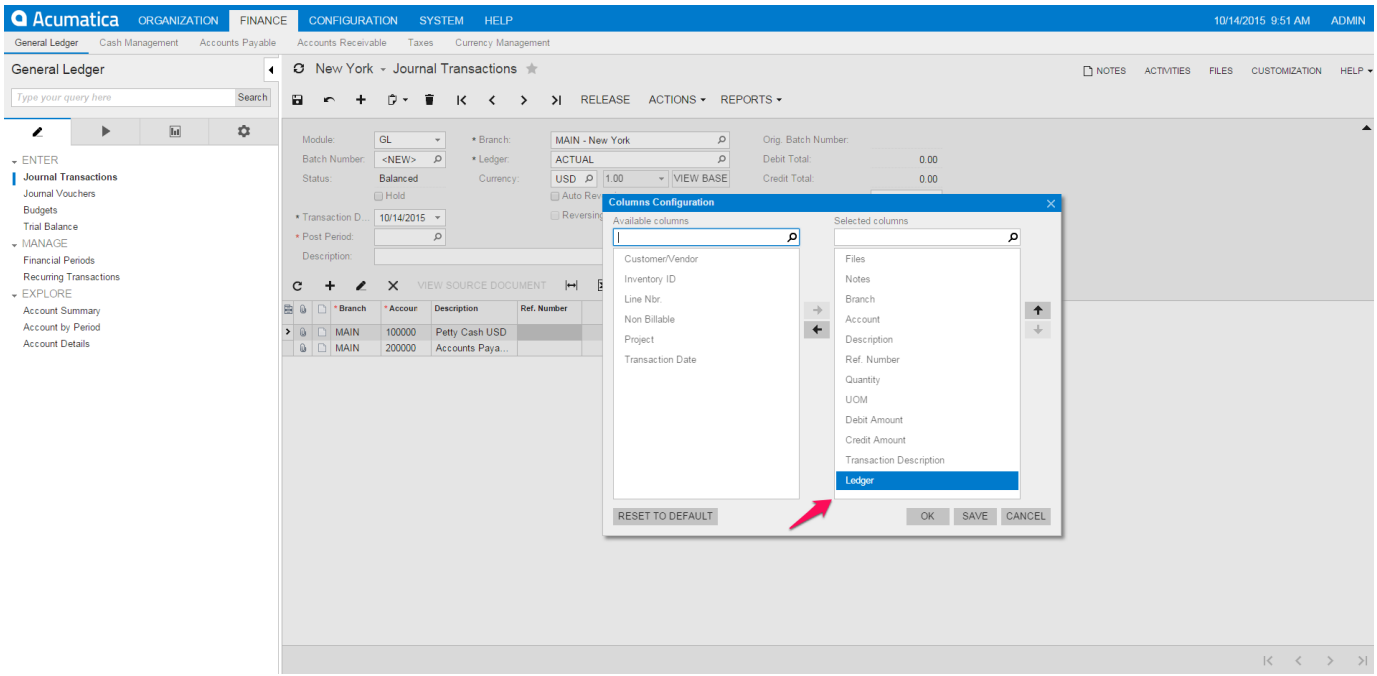
How to Show or Hide Columns in a Detail Table

In this topic, you can find the descriptions of the following tasks:

- Showing a Hidden Column
- Hiding a Displayed Column
- Restoring the Default Layout of the Columns

Showing a Hidden Column

To display a hidden column, use the `ShowColumn()` method. The `ShowColumn()` method moves the specified column to the Selected columns list of the Columns Configuration dialog box, as shown in the following screenshot, and sets the column position in the table to its position in the corresponding wrapper class.

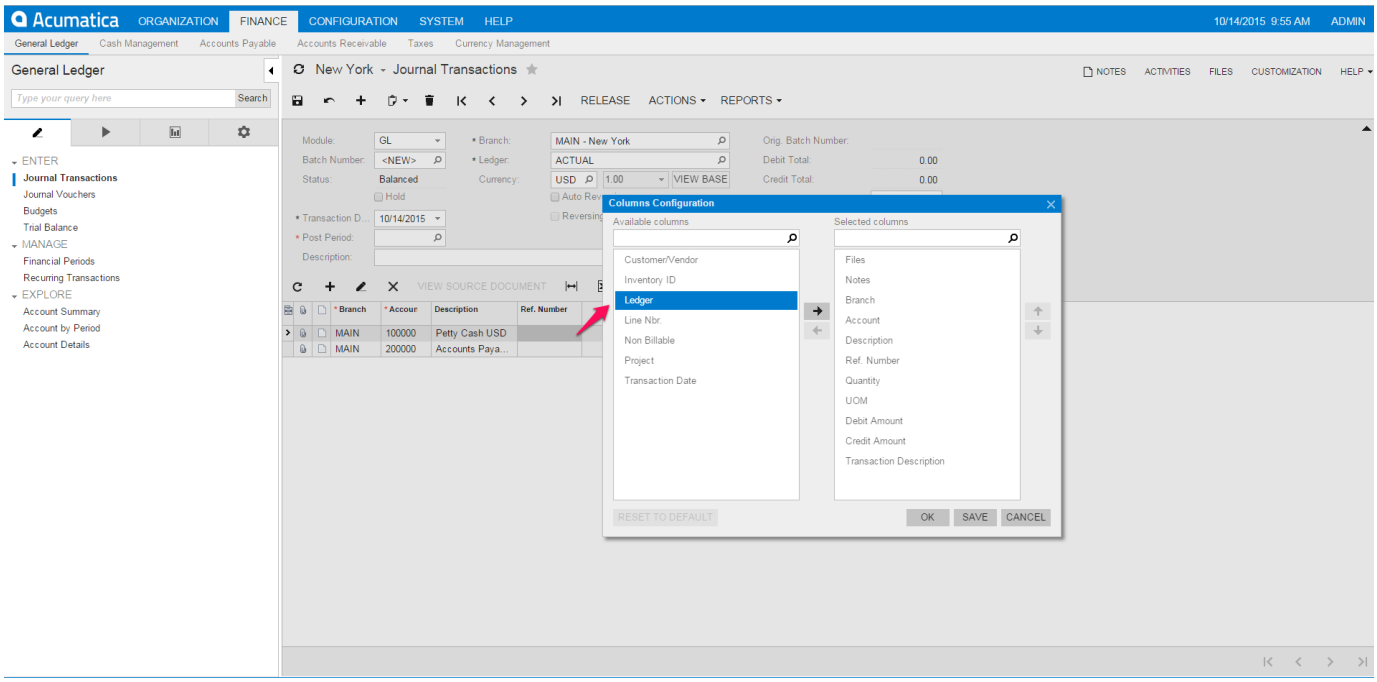


The following code shows how to show the hidden Ledger column in the details table of the Journal Transactions form (GL301000; Finance > General Ledger > Enter).

```
JournalEntry.OpenScreen();
JournalEntry.Details.Columns.LedgerID.ShowColumn();
```

Hiding a Displayed Column

To hide a displayed column of a detail table, use the `HideColumn()` method. The `HideColumn()` method moves the specified column to the Available columns list of the Columns Configuration dialog box, as shown in the following screenshot.

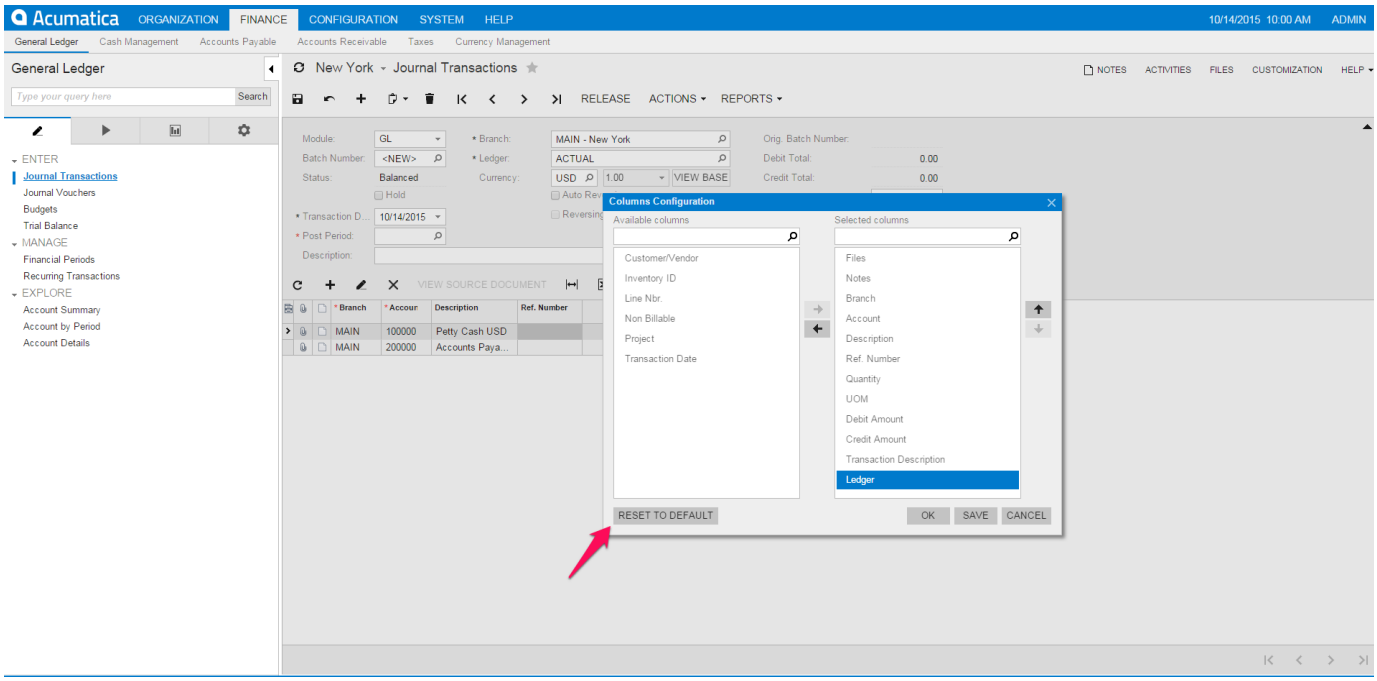


The following code shows how to hide the Ledger column of the detail table on the Journal Transactions form (GL301000).

```
JournalEntry.OpenScreen();
JournalEntry.Details.Columns.LedgerID.HideColumn();
```

Restoring the Default Layout of the Columns

To restore the default layout of the columns of a detail table, use the `ResetLayout()` method. The `ResetLayout()` method clicks the Reset to Default button in the Columns Configuration dialog box, which is shown in the following screenshot.



The following code shows how to restore the default layout of the columns on the Journal Transactions form (GL301000).

```
    JournalEntry.OpenScreen();
    JournalEntry.Details.ResetLayout();
```

How to Use Column Filters of a Detail Table

You can specify a filtering condition in every column of a detail table. The list of conditions and filters is available in the user interface of your Acumatica application and depends on the control type of the column (such as text, number, or date).

In this topic, you can find the descriptions of the following tasks:

- Setting a Column Filter
- Clearing a Filter

Setting a Column Filter

To specify a column filter, use the `Columns` property. See below for examples of setting different column filters.

The following code shows an example of setting the Starts With filter, which is applicable for any text or selector field.

```
SalesPrice.Details.Columns.InventoryID.StartsWith("REQ");
```

The following code examples illustrate the setting of the Equals filter:

- For a date picker

```
SalesPrice.Details.Columns.EffectiveDate.Equals(new System.DateTime(2011, 5, 16));
```

- For a box with text

```
SalesPrice.Details.Columns.InventoryID_InventoryItem_Descr.Equals("RQ13");
```

- For a box with a number

```
SalesPrice.Details.Columns.SalesPrice.Equals(60);
```

The following code illustrates the setting of the Is Between and Is Greater than or Equal to filters, which are available for date pickers and boxes with numbers.

```
SalesPrice.Details.Columns.SalesPrice.IsBetween(50, 60);
SalesPrice.Details.RowsCount().VerifyEquals(2);
SalesPrice.Details.Columns.SalesPrice.IsGreaterThanOrEqualTo(60);
SalesPrice.Details.RowsCount().VerifyEquals(20);
```

The code below shows how to use the True and False filters, which are available for check box columns.

```
SalesPrice.Details.Columns.IsPromotionalPrice.IsTrue();
SalesPrice.Details.Columns.IsPromotionalPrice.IsFalse();
```

The following code shows how to select the filtering values for a drop-down column. To specify the values to be selected, you can use either the values visible in the UI or the internal IDs of the values.

```
//Using values visible in the UI
EntryType.Details.Columns.Module.SelectValues("CA", "AP");
//Using internal IDs of the values
FixedAsset.TransactionHistory.Columns.TranType.SelectValues("A+", "A-", "P+", "P-");
```

Clearing a Filter

To clear the filter of a column, use the `ClearFilter()` method. This method is not available for drop-down columns.

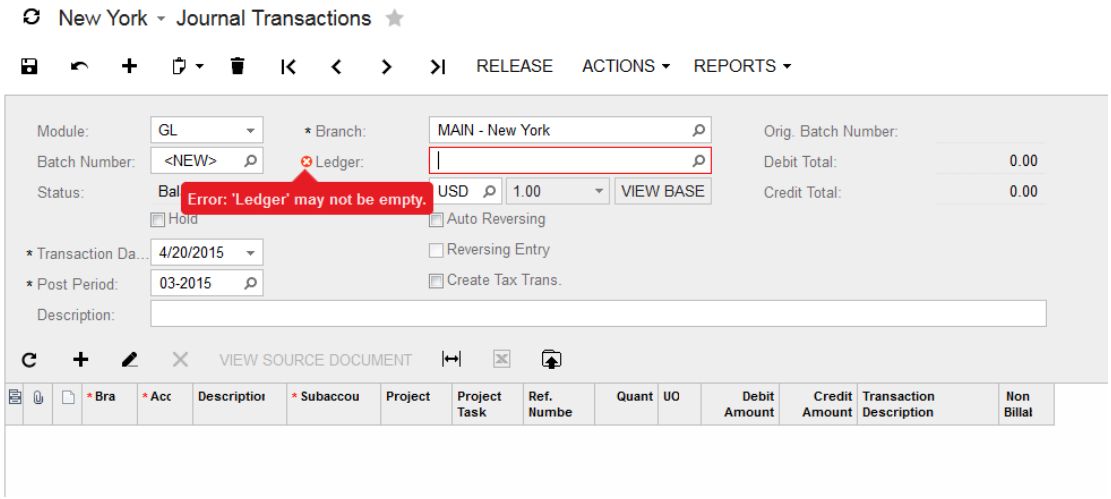
```
EntryType.Details.Columns.EntryTypeId.ClearFilter();
```

To clear the filter for a drop-down column, call the `SelectAll()` method.

```
EntryType.Details.Columns.Module.SelectAll();
```

How to Work with Errors

Test SDK provides the `GetError()` method for working with errors that occur on the form or on a control, such as the error in the following screenshot. By using the `GetError()` method, you can verify whether an error appears on the form or on the control, and check the text of the error message.



In your test application, you can verify an error on the form or control by performing one of the following tasks:

- Verifying Whether an Error Appears on the Form
- Verifying that No Error Appears on the Form
- Verifying the Text of the Error Message on the Form
- Verifying Whether an Error Appears on the Control
- Verifying the Text of the Error Message on a Control
- Verifying the Text of the Error Message on the Cell of a Detail Table
- Verifying the Text of the Error Message on the Row of a Detail Table

Verifying Whether an Error Appears on the Form

To check whether an error appears on the form, use the `GetError()` method of the form class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.LedgerID.Reset();
ReceiptPo.HasError().VerifyEquals(true);
```

Verifying that No Error Appears on the Form

To make sure that no error appears on the form, use the `GetError()` method of the form class with the parameter set to `False`, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.HasError().VerifyEquals(false);
```

Verifying the Text of the Error Message on the Form

To check that the correct text of the error message is displayed on the form, use the `GetError()` method of the form class with the text of the error as the parameter, as shown in the following example. You can specify a part of the error message in the parameter of the method; in this case, the method checks whether the error message on the form contains the specified text. You can also use wildcards in the parameter of the method (for details, see How to Use Wildcards When Verifying Errors and Warnings).

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.LedgerID.Reset();
ReceiptPo.GetErrors().VerifyContains("'Ledger' may not be empty.");
```

Verifying Whether an Error Appears on the Control

To check whether an error appears on the particular control, use the `GetError()` method of the control class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.LedgerID.Reset();
ReceiptPo.Summary.LedgerID.HasError().VerifyEquals(true);
```

Verifying the Text of the Error Message on a Control

To check that the correct text of the error message is displayed on the control, use the `GetError()` method of the control class with the text of the error as the parameter, as shown in the following example. You can specify a part of the error message in the parameter of the method; in this case, the method checks whether the error message on the form contains the specified text. You can also use wildcards in the parameter of the method (for details, see How to Use Wildcards When Verifying Errors and Warnings).

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.LedgerID.Reset();
ReceiptPo.Summary.LedgerID.GetError().VerifyContains("'Ledger' may not be empty.");
```

Verifying the Text of the Error Message on the Cell of a Detail Table

To check whether an error appears on the particular cell of a detail table, activate the needed row and use the `GetError()` method of the cell class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Details.New();
ReceiptPo.Details.Row.BranchID.Reset();
ReceiptPo.Details.SelectRow(1);
ReceiptPo.Details.Row.BranchID.GetError().VerifyContains("'Branch' may not be empty.");
```

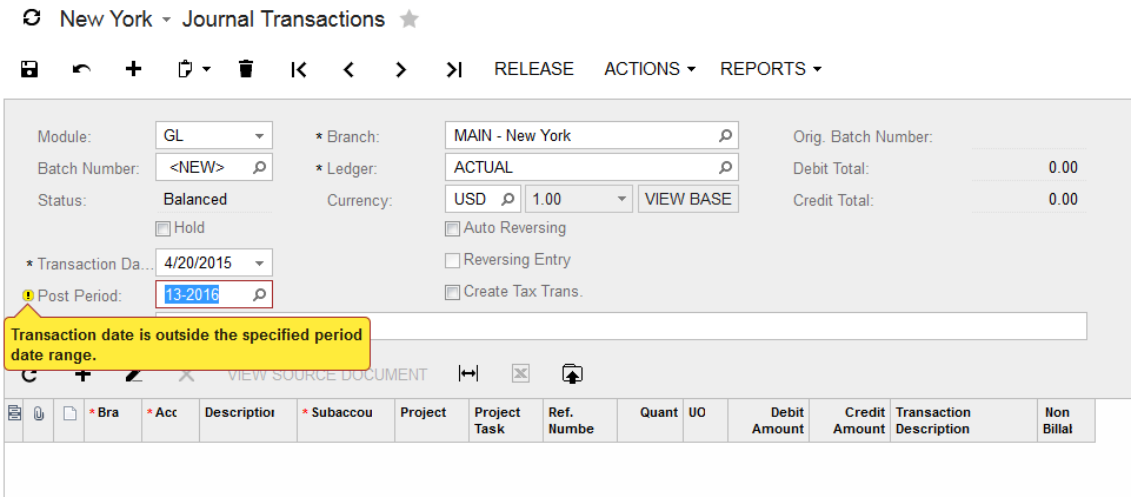
Verifying the Text of the Error Message on the Row of a Detail Table

To check the text of the error message that appears on the row of a detail table, activate the needed row and use the `GetError()` method of the detail table class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Details.New();
ReceiptPo.Details.Row.BranchID.Reset();
ReceiptPo.Details.SelectRow(1);
ReceiptPo.Details.Row.GetError().VerifyContains("'Branch' may not be empty.");
```

How to Work with Warnings

Test SDK provides the `GetWarning()` method for working with the warnings that occur on the form or on a control, such as the warning shown in the following screenshot. By using the `GetWarning()` method, you can verify whether a warning appears on the form or control, and check the text of the warning message.



In your test application, you can verify a warning on the form or control by performing one of the following tasks:

- Verifying the Number of Warnings That Appear on the Form
- Verifying the Text of the Warning Message on the Form
- Verifying Whether a Warning Appears on the Control
- Verifying the Text of the Warning Message on the Cell of a Detail Table
- Verifying the Text of the Warning Message on the Row of a Detail Table

Verifying the Number of Warnings That Appear on the Form

To check the number of warnings that appear on the form, use the `GetWarning()` method of the form class. The following example verifies that one error appears on the form.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.FinPeriodID.Type("13-2016");
ReceiptPo.GetWarnings().VerifyCount(1);
```

Verifying the Text of the Warning Message on the Form

To check that the correct text of the warning message is displayed on the form, use the `GetWarning()` method of the form class with the text of the warning as the parameter, as shown in the following example. You can specify a part of the warning message in the parameter of the method; in this case, the method checks whether the warning message on the form contains the specified text. You can also use wildcards in the parameter of the method (for details, see Supported Wildcards).

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.FinPeriodID.Type("13-2016");
ReceiptPo.GetWarnings().VerifyContains("Transaction date is outside the specified period date range");
```

Verifying Whether a Warning Appears on the Control

To check whether a warning appears on the control, use the `GetWarning()` method of the control class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Summary.FinPeriodID.Type("13-2016");
ReceiptPo.Summary.FinPeriodID.GetWarning().VerifyContains("Transaction date is outside the specified period date range");
```

Verifying the Text of the Warning Message on the Cell of a Detail Table

To check whether a warning appears on the cell of a detail table, activate the needed row and use the `GetWarning()` method of the cell class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Details.New();
ReceiptPo.Details.Row.BranchID.Reset();
ReceiptPo.Details.SelectRow(1);
ReceiptPo.Details.Row.BranchID.GetWarning().VerifyContains("'Branch' may not be empty.");
```

Verifying the Text of the Warning Message on the Row of a Detail Table

To check the text of the warning message that appears on the row of a detail table, activate the needed row and use the `GetWarning()` method of the detail table class, as shown in the following example.

```
ReceiptPo.OpenScreen();
ReceiptPo.Insert();
ReceiptPo.Summary.BranchID.Select("MAIN");
ReceiptPo.Details.New();
ReceiptPo.Details.Row.BranchID.Reset();
ReceiptPo.Details.SelectRow(1);
ReceiptPo.Details.Row.GetWarning().VerifyContains("'Branch' may not be empty.");
```

How to Work with Windows

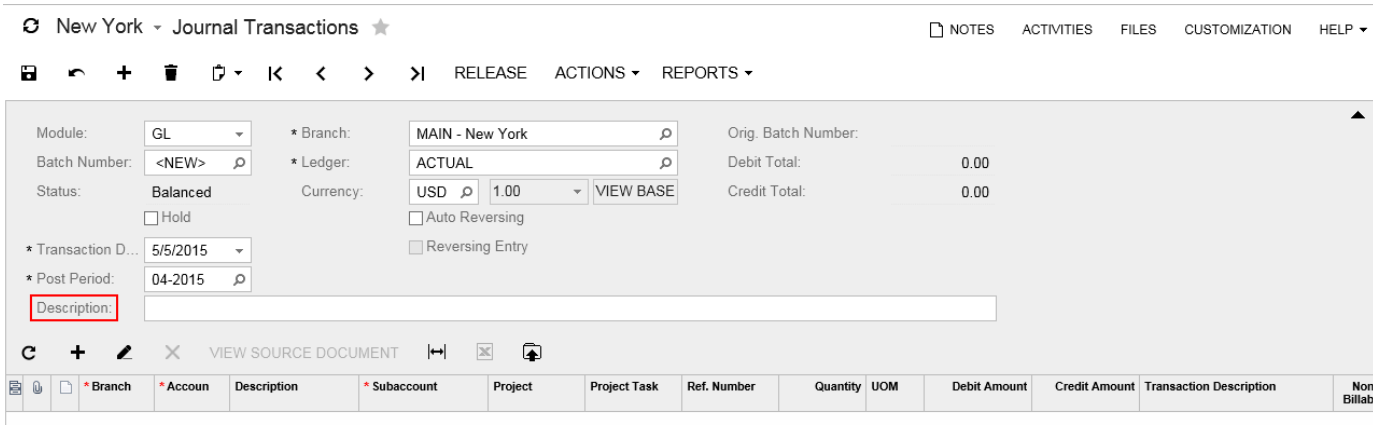
On some Acumatica forms, pop-up forms can appear as you work with the form.

The following example shows how to select an active window when multiple Acumatica forms appear on the screen of your computer.

```
...

public JournalEntry()
{
    Details.ToolBar.ViewDocument.WaitAction = Wait.WaitForNewWindowToOpen;
}

...

JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen(); //The Journal Transactions form (GL301000) is the first form that appears on the screen.
JournalEntry.Summary.Module.Select("AP");
JournalEntry.Last();
JournalEntry.Details.ViewDocument(); //The Bills and Adjustments form (AP301000) is the second form that appears on the screen.

Bill Bill = new Bill();
Bill.Cancel();
Bill.CloseWindow(); //Closes currently selected window

JournalEntry.Summary.Module.VerifyEquals("AP");
```

How to Work with the Names of UI Elements

In this topic, you can find examples that illustrate how to work with the names of UI elements by using Test SDK. The name of the Description element on the Journal Transactions form (GL301000; Finance > General Ledger > Work Area > Enter) is highlighted in the following screenshot.



You may need to perform the following tasks with the names of UI elements:

- Obtaining the Name of a UI Element
- Verifying the Name of a UI Element

Obtaining the Name of a UI Element

The following code shows how to obtain the name of a UI element.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Summary.DescriptionLabel.GetValue();
```

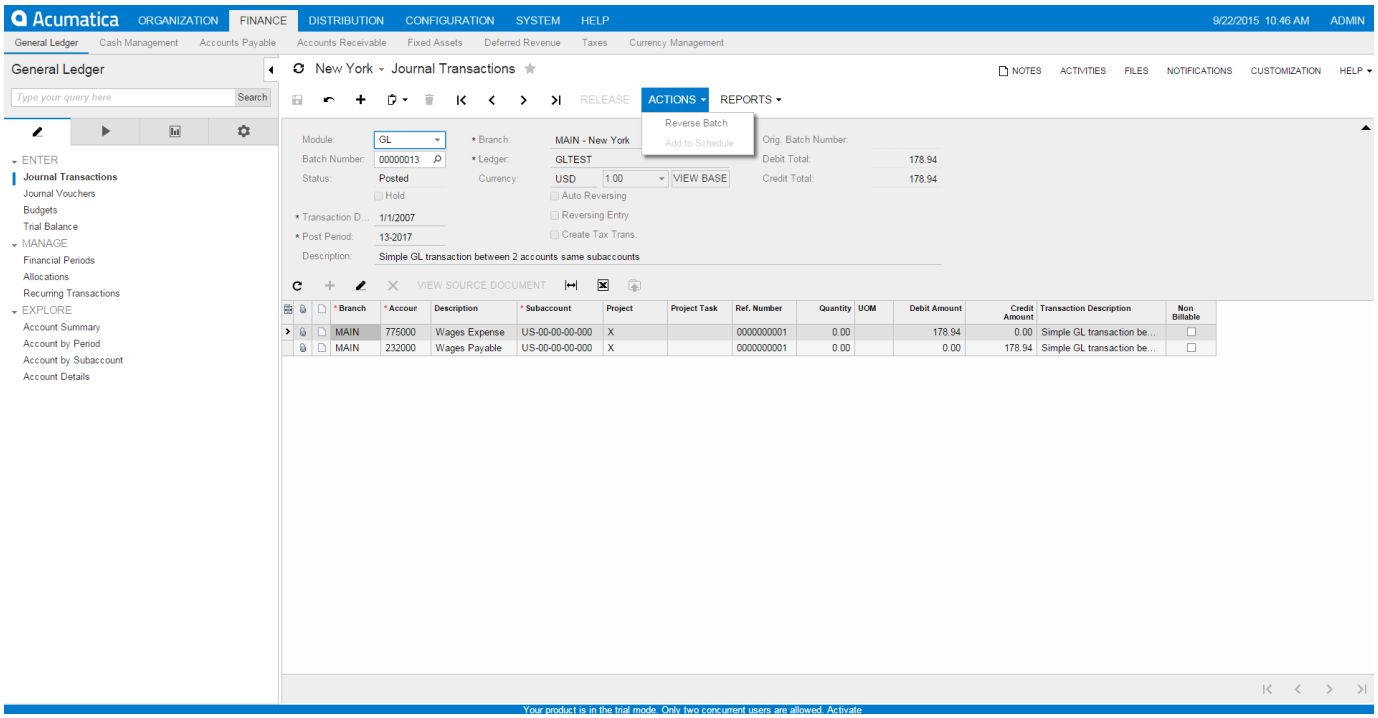
Verifying the Name of a UI Element

The following code shows how to verify the name of a UI element.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Summary.DescriptionLabel.GetValue().VerifyEquals("Description:");
```

How to Work with Drop-Down Menus on Toolbars

In this topic, you can find examples that illustrate how to work with a drop-down menu on a toolbar by using Test SDK. The following screenshot shows the toolbar buttons that are located on the drop-down menu of the Journal Transactions form (GL301000; Finance > General Ledger > Work Area > Enter).



You may need to perform the following tasks with the drop-down menu of a toolbar:

- Clicking a Toolbar Button That Is Located on a Drop-Down Menu
- Verifying That a Toolbar Button That Is Located on a Drop-Down Menu is Enabled

Clicking a Toolbar Button That Is Located on a Drop-Down Menu

To click a toolbar button that is located on a drop-down menu, you need to invoke the `Click()` method of the button class, as shown in the following code. You do not need to open the menu first.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Next();
JournalEntry.ToolBar.ReverseBatch.Click();
```

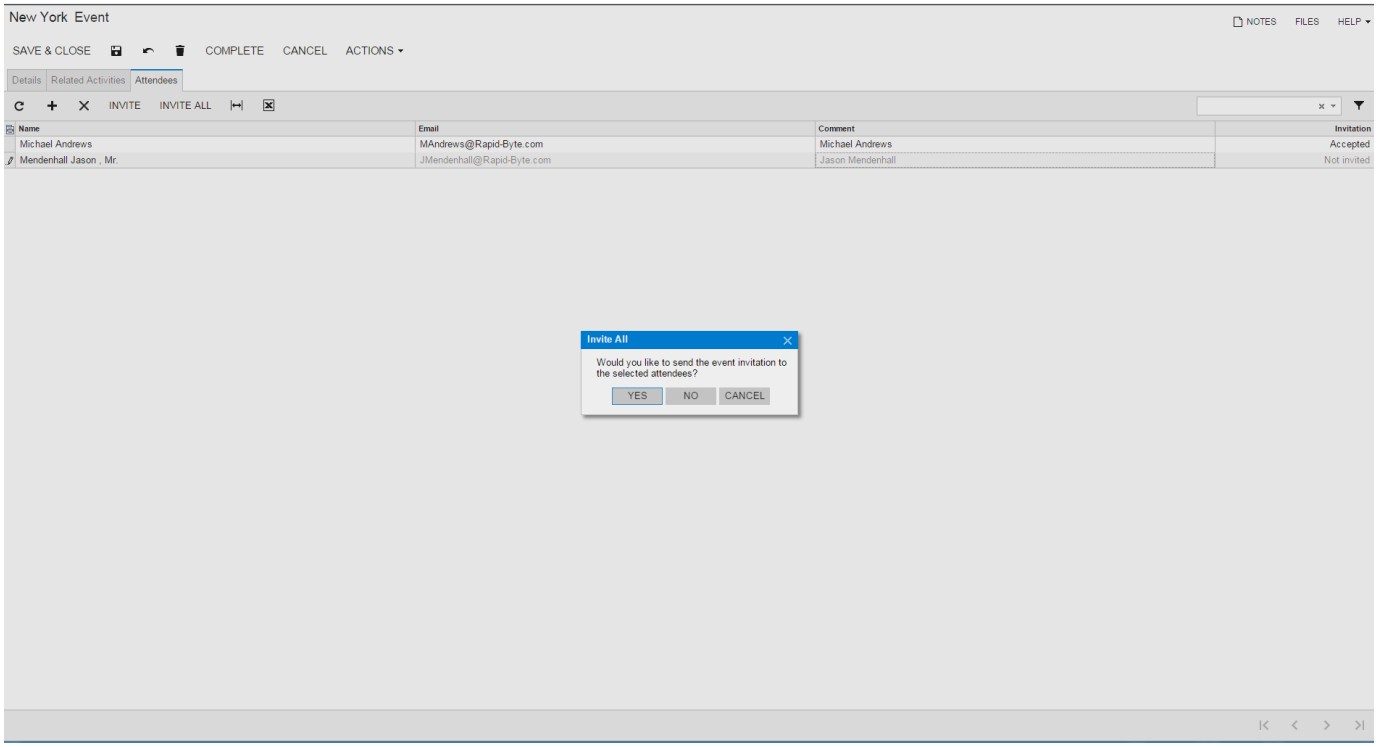
Verifying That a Toolbar Button That Is Located on a Drop-Down Menu is Enabled

To verify that a toolbar button that is located on a drop-down menu is enabled, you need to invoke the IsEnabled() method of the button class. You do not need to open the menu first.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Next();
JournalEntry.ToolBar.ReverseBatch.IsEnabled().VerifyEquals(true);
```

How to Work with Pop-Up Dialog Boxes

In this topic, you can find examples that show how to work with pop-up dialog boxes. An example of pop-up dialog box is shown in the following screenshot.



You may need to perform the following tasks with pop-up dialog boxes:

- Clicking a Button in a Pop-Up Dialog Box
- Closing a Pop-Up Dialog Box
- Obtaining the Message Text from a Pop-Up Dialog Box

Clicking a Button in a Pop-Up Dialog Box

The following code shows how to click a button in a pop-up dialog box through the example of the Invite All dialog box, which can be invoked on the Event form (CR306030).

```
EventList EventList = new EventList();
EventList.OpenScreen();
EventList.New();
Event Event = new Event();
Event.Summary.Subject.Type("test");
Event.Attendees.New();
Event.Attendees.Row.Name.Select("Jason Mendenhall");
Event.Save();
//Click Yes
Event.MessageBox.Yes();
//or click No
//Event.MessageBox.No()
//or click Cancel
//Event.MessageBox.Cancel()
```

Closing a Pop-Up Dialog Box

The following code shows how to close a pop-up dialog box through the example of the Invite All dialog box, which can be invoked on the Event form (CR306030).

```
EventList EventList = new EventList();
EventList.OpenScreen();
EventList.New();
Event Event = new Event();
Event.Summary.Subject.Type("test");
Event.Attendees.New();
Event.Attendees.Row.Name.Select("Jason Mendenhall");
Event.Save();
Event.MessageBox.Close();
```

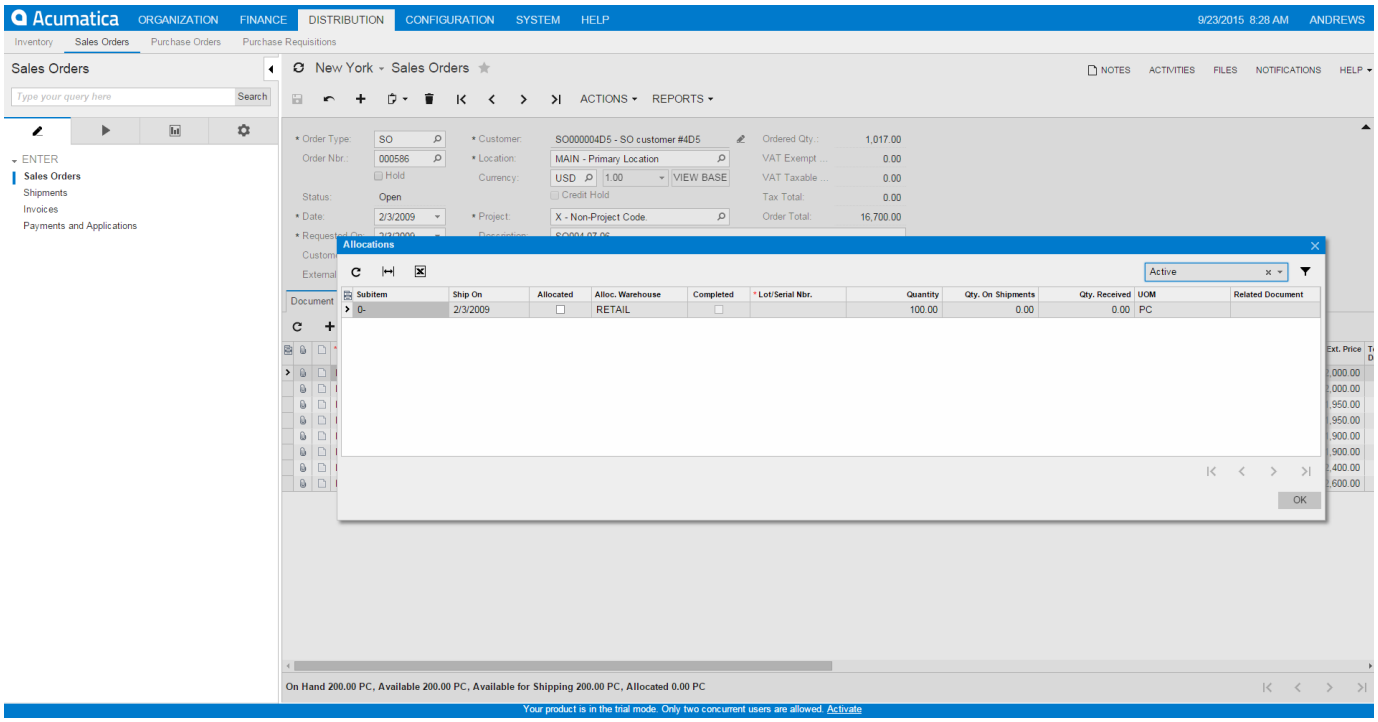
Obtaining the Message Text from a Pop-Up Dialog Box

The following code shows how to obtain the text from a pop-up dialog box through the example of the Invite All dialog box, which can be invoked on the Event form (CR306030).

```
EventList EventList = new EventList();
EventList.OpenScreen();
EventList.New();
Event Event = new Event();
Event.Summary.Subject.Type("test");
Event.Attendees.New();
Event.Attendees.Row.Name.Select("Jason Mendenhall");
Event.Save();
var messageText = Event.MessageBox.GetValue();
```

How to Work with Pop-Up Panels

In this topic, you can find examples that show how to work with pop-up panels that appear on Acumatica forms by using Test SDK. An example of a pop-up panel is shown in the following screenshot.



You may need to perform the following task with pop-up panels:

- Closing a Pop-up Panel

Closing a Pop-up Panel

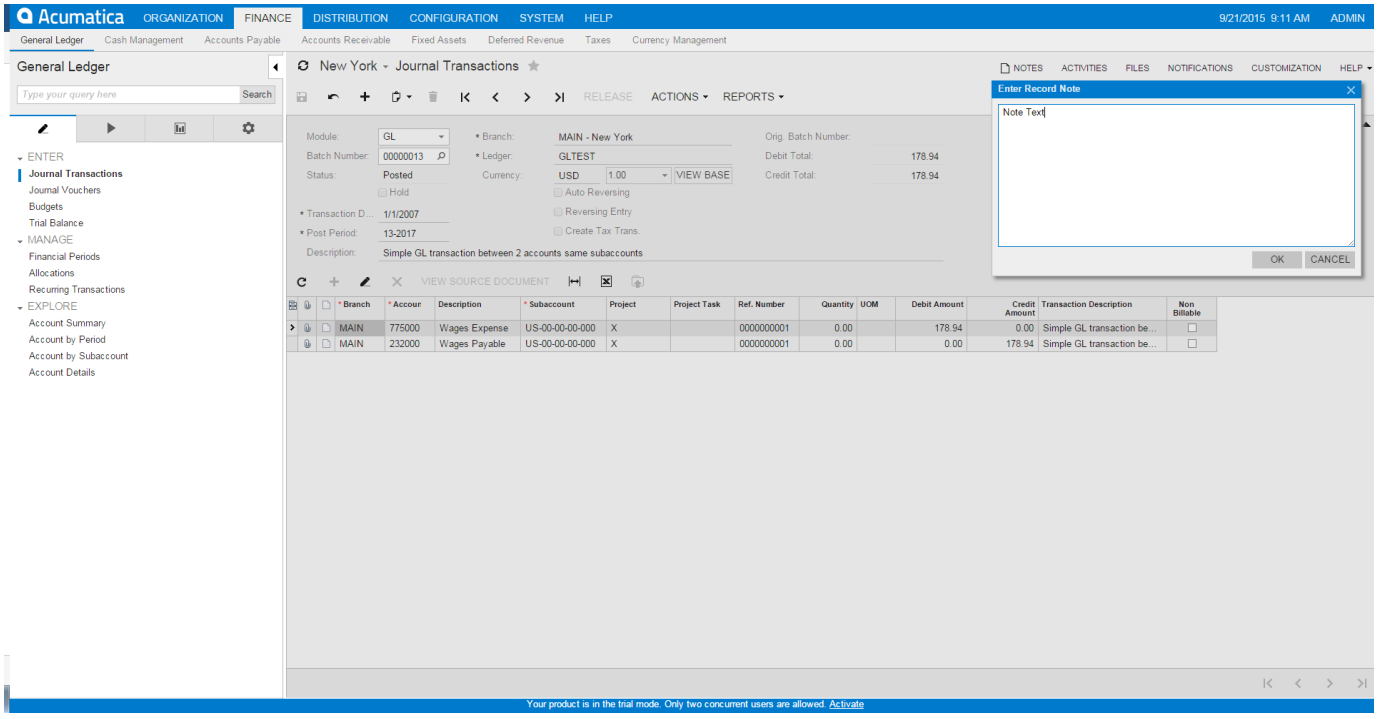
The following code shows how to close a pop-up panel through the example of the Allocations pop-up panel that can appear on the Sales Orders form (SO301000; Distribution > Sales Orders > Work Area > Enter).

```
OrderSo.OpenScreen();
OrderSo.Last();
OrderSo.DocumentDetails.Allocations();
OrderSo.AllocationsForm.Close();
```

How to Verify a Note on a Form

In this topic, you will see an example of the note text on a form being verified.

The following screenshot illustrates the verification of a note on the Journal Transactions form (GL301000).



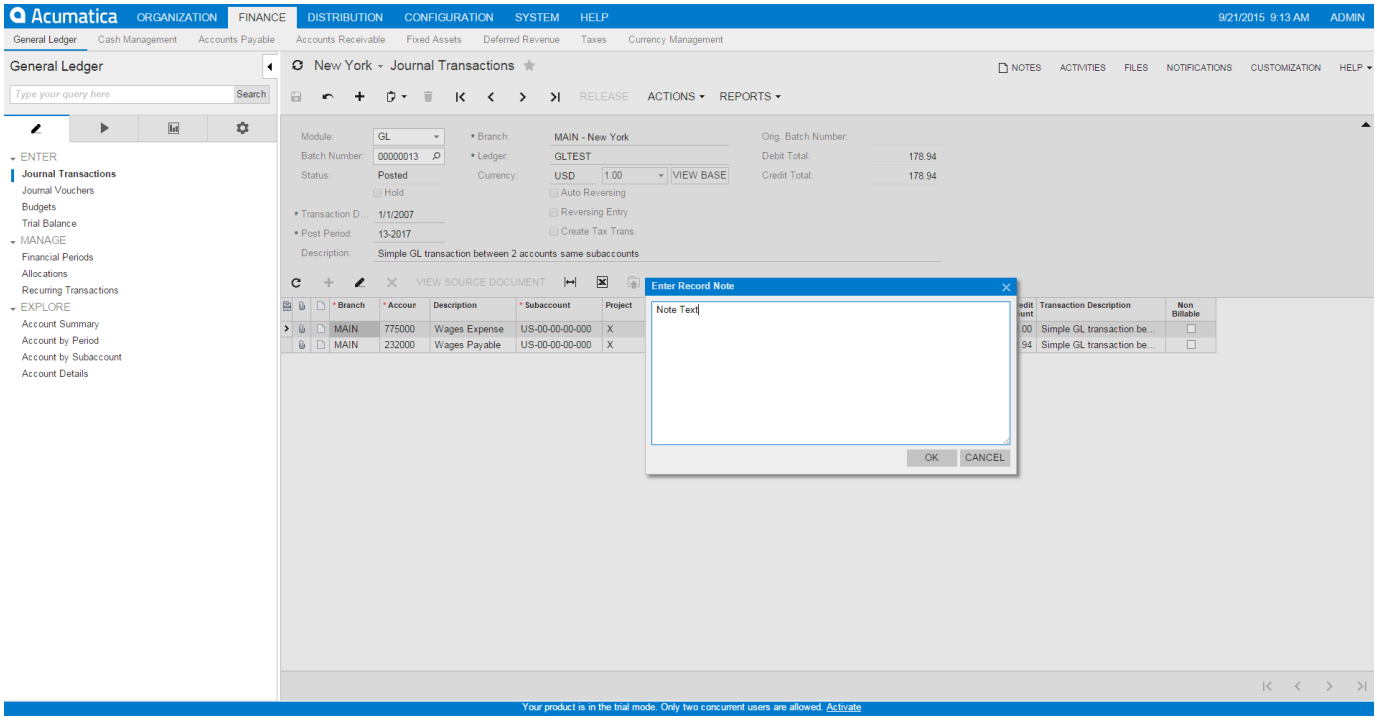
The code below shows how to verify the text of a note in your test.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Note();
JournalEntry.NotePanel.NoteEdit.VerifyEquals("Note Text")
JournalEntry.NotePanel.Cancel();
```

How to Add a Note to a Detail Line

In this topic, you will see an example of adding a note to a detail line of a document on an Acumatica ERP form.

The following screenshot illustrates a note being added to a detail line on the Journal Transactions form (GL301000).



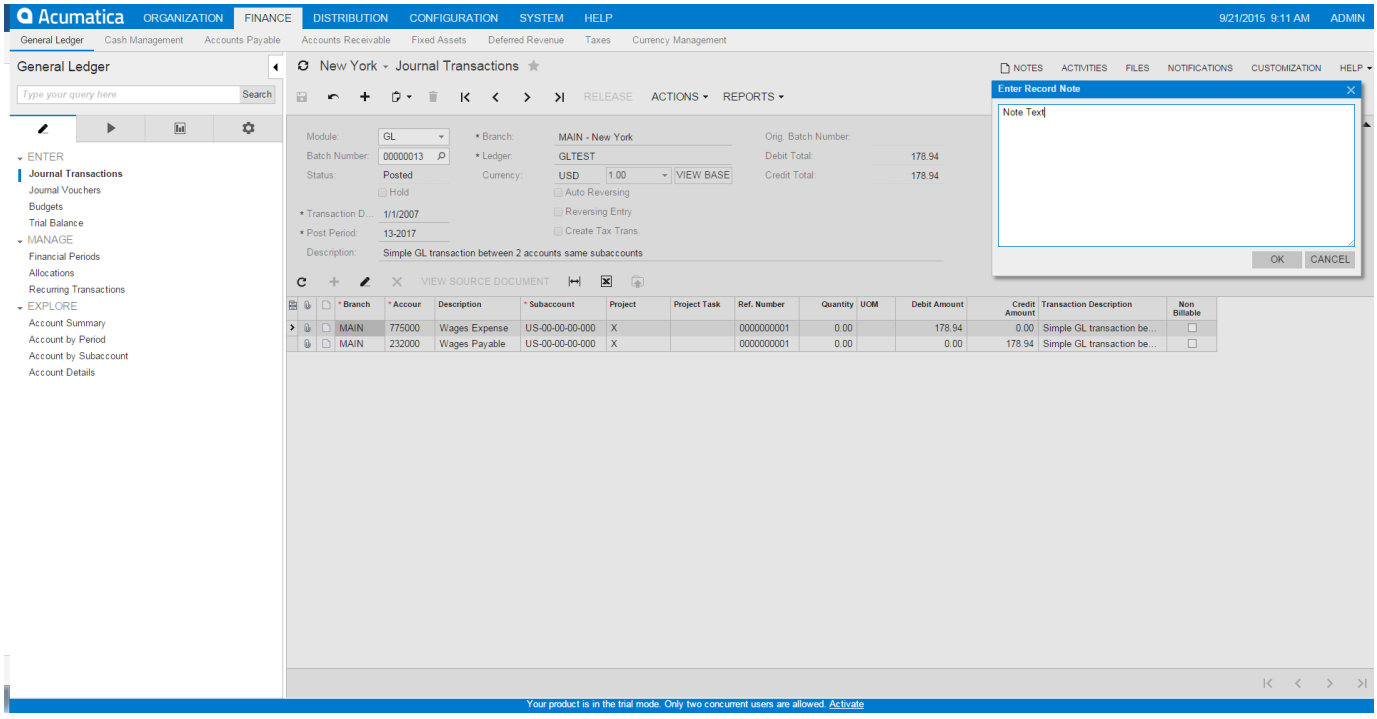
The code below shows how to implement adding a note to a detail line in your test.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Details.SelectRow(1);
JournalEntry.Details.Row.Notes.Click();
JournalEntry.Details.NotePanel.Type("Note Text");
JournalEntry.Details.NotePanel.Ok();
```

How to Add a Note to an Acumatica Form

In this topic, you will see an example of adding a note to a record on a form by using Test SDK.

The following screenshot shows a note being added in Acumatica ERP.

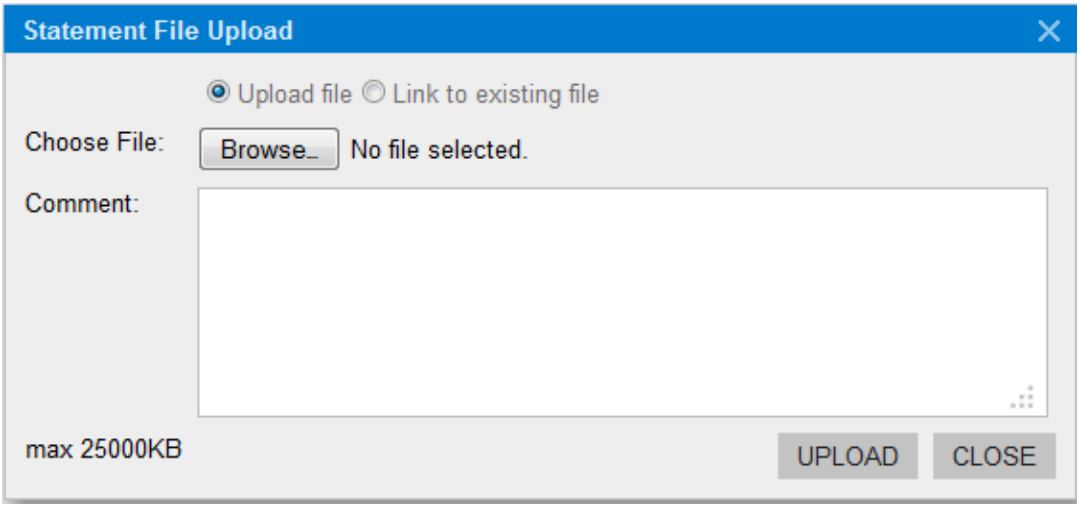


The code below shows how to implement adding a note in your test.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Note();
JournalEntry.NotePanel.Type("Note Text")
JournalEntry.NotePanel.Ok();
```

How to Upload a File

On some Acumatica ERP forms, you can upload files of different formats to the form by using form-specific buttons on the form toolbar. For example, on the Import Bank Transactions form (CA306500; Finance > Cash Management > Enter), you can upload a file by clicking the Upload File button on the form toolbar. This opens the Statement File Upload dialog box, which is shown in the following screenshot.



The example below shows how to upload a file on the Import Bank Transactions form (CA306500).

```
BankTransactionsImport BankTransactionsImport = new BankTransactionsImport();
BankTransactionsImport.OpenScreen();
BankTransactionsImport.UploadFile();
BankTransactionsImport.StatementFileUpload.SelectFile("C:\\share\\BankTransactions.dat");
BankTransactionsImport.StatementFileUpload.Upload();
```

How to Upload Data from an Excel File into a Detail Table

The following example shows how to upload data from an Excel file into a detail table on the Chart of Accounts form (GL202500; Finance > General Ledger > Configuration > Manage).

```
Account Account = new Account();
Account.OpenScreen();
Account.Upload();
Account.Details.UploadForm.SelectFile("C:\\share\\Accounts.xlsx");
Account.Details.UploadForm.Upload();

Account.ExcelImportSettings.NullValue.Type("NULL");
Account.ExcelImportSettings.Culture.Select("Norwegian");
Account.ExcelImportSettings.Mode.Select("Bypass Existing");
Account.ExcelImportSettings.Ok();

Account.Columns.Ok();
```

How to Upload Data from a CSV File into a Detail Table

The following code shows an example of data being uploaded from a CSV file into a detail table on the Chart of Accounts form (GL202500; Finance > General Ledger > Configuration > Manage).

```
Account Account = new Account();
Account.OpenScreen();
Account.Upload();
Account.Details.UploadForm.SelectFile("C:\\share\\Accounts.csv");
Account.Details.UploadForm.Upload();

Account.CsvImportSettings.Separator.Type("|");
Account.CsvImportSettings.NullValue.Type("NULL");
Account.CsvImportSettings.CodePage.Select("Norwegian (IA5)");
Account.CsvImportSettings.Culture.Select("Norwegian");
Account.CsvImportSettings.Mode.Select("Bypass Existing");
Account.CsvImportSettings.Ok();

Account.Columns.Ok();
```

How to Attach a File to an Acumatica Form

The following code shows an example of a file being attached to a record on the Journal Transactions form (GL301000).

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.FilesMenuShow();
JournalEntry.FilesUploadDialog.FilesAttached.VerifyRowsCount(0);
JournalEntry.FilesUploadDialog.FileUploader.SelectFile("C:\\share\\Test.txt");
JournalEntry.FilesUploadDialog.FileUploader.Upload();
JournalEntry.FilesUploadDialog.FilesAttached.VerifyRowsCount(1);
string fileName = JournalEntry.FilesUploadDialog.FilesAttached.Row.FileName.GetValue();
JournalEntry.FilesUploadDialog.FilesAttached.Row.Comment.Type(
   string.Format("This file has been attached to Acumatica page: {0}", fileName));
JournalEntry.FilesUploadDialog.Close();
```

How to Attach a File to a Detail Line

The following code shows how to attach a file to a detail line of a record on the Journal Transactions form (GL301000).

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.Details.VerifyRowsCount(1);
JournalEntry.Details.SelectRow(1);
JournalEntry.Details.Row.Files.Click();
JournalEntry.Details.FilesUploadDialog.FilesAttached.VerifyRowsCount(0);
JournalEntry.Details.FilesUploadDialog.FileUploader.SelectFile("C:\\share\\Test.xlsx");
JournalEntry.Details.FilesUploadDialog.FileUploader.Upload();
JournalEntry.Details.FilesUploadDialog.FilesAttached.VerifyRowsCount(1);
string fileName = JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.FileName.GetValue();
JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.Comment.Type(
   string.Format("This file has been attached to a detail line: {0}", fileName));
JournalEntry.Details.FilesUploadDialog.Close();
```

How to Remove an Attached File from an Acumatica Form

You can use one of the following approaches when removing an attached file from an Acumatica form:

- You can remove the file by using the File Maintenance form (SM202510).
- You can remove the file by using the Search in Files form (SM202520).

Removing an Attached File by Using the File Maintenance Form (SM202510)

The following code shows how to remove the attached file from the Journal Transactions form (GL301000) by using the File Maintenance form (SM202510).

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.FilesMenuShow();
JournalEntry.FilesUploadDialog.FilesAttached.VerifyRowsCount(1);
JournalEntry.FilesUploadDialog.FilesAttached.SelectRow(1);
string fileName = JournalEntry.FilesUploadDialog.FilesAttached.Row.FileName.GetValue();
JournalEntry.FilesUploadDialog.FilesAttached.Row.Comment.Type(
  string.Format("This file will be removed from the Acumatica ERP form: {0}", fileName));
string linkText = JournalEntry.FilesUploadDialog.FilesAttached.Row.Edit.GetValue();
JournalEntry.FilesUploadDialog.FilesAttached.Row.Edit.ClickLink(linkText);

FileMaintenance FileMaintenance = new FileMaintenance();
FileMaintenance.Delete();
FileMaintenance.CloseWindow();

JournalEntry.FilesUploadDialog.FilesAttached.Refresh();
JournalEntry.FilesUploadDialog.FilesAttached.VerifyRowsCount(0);
JournalEntry.FilesUploadDialog.Close();
```

Removing an Attached File by Using the Search in Files Form (SM202520)

The following code shows how to remove the attached file from the Journal Transactions form (GL301000) by using the Search in Files form (SM202520).

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.FilesMenuShow();
JournalEntry.FilesUploadDialog.FilesAttached.VerifyRowsCount(1);
JournalEntry.FilesUploadDialog.FilesAttached.SelectRow(1);
string fileName = JournalEntry.FilesUploadDialog.FilesAttached.Row.FileName.GetValue();
JournalEntry.FilesUploadDialog.FilesAttached.Row.Comment.Type(
            string.Format("This file will be removed from the Acumatica ERP form: {0}", fileName));
JournalEntry.FilesUploadDialog.Close();

SearchInFiles SearchInFiles = new SearchInFiles();
SearchInFiles.OpenScreen();
SearchInFiles.Summary.DocName.Type(fileName);
SearchInFiles.Details.VerifyRowsCount(1);
SearchInFiles.Details.DeleteFile();
SearchInFiles.Details.VerifyRowsCount(0);

JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.FilesMenuShow();
JournalEntry.FilesUploadDialog.FilesAttached.VerifyRowsCount(0);
JournalEntry.FilesUploadDialog.Close();
```

How to Remove an Attached File from a Detail Line

You can use one of the two approaches to remove the file attached to a detail line:

- Remove the file by using the File Maintenance form (SM202510)
- Remove the file by using the Search in Files form (SM202520)

Removing the File Attached to a Detail Line by Using the File Maintenance form (SM202510)

The following code shows how to remove the file attached to a detail line on the Journal Transactions form (GL301000). The approach illustrated in this code uses the File Maintenance form (SM202510) to remove the attached file.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.Details.VerifyRowsCount(1);
JournalEntry.Details.SelectRow(1);
JournalEntry.Details.Row.Files.Click();
JournalEntry.Details.FilesUploadDialog.FilesAttached.VerifyRowsCount(1);
JournalEntry.Details.FilesUploadDialog.FilesAttached.SelectRow(1);
string fileName = JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.FileName.GetValue();
JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.Comment.Type(
  string.Format("This file will be removed from a detail line of the Acumatica ERP form: {0}", fileName));
string linkText = JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.Edit.GetValue();
JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.Edit.ClickLink(linkText);

FileMaintenance FileMaintenance = new FileMaintenance();
FileMaintenance.Delete();
FileMaintenance.CloseWindow();

JournalEntry.Details.FilesUploadDialog.FilesAttached.Refresh();
JournalEntry.Details.FilesUploadDialog.FilesAttached.VerifyRowsCount(0);
JournalEntry.Details.FilesUploadDialog.Close();
```

Removing the File Attached to a Detail Line by Using the Search in Files Form (SM202520)

The following code shows how to remove the file attached to a detail line on the Journal Transactions form (GL301000). The approach illustrated in this code uses the Search in Files form (SM202520) to remove the attached file.

```
JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Last();
JournalEntry.Details.VerifyRowsCount(1);
JournalEntry.Details.SelectRow(1);
JournalEntry.Details.Row.Files.Click();
JournalEntry.Details.FilesUploadDialog.FilesAttached.VerifyRowsCount(1);
JournalEntry.Details.FilesUploadDialog.FilesAttached.SelectRow(1);
string fileName = JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.FileName.GetValue();
JournalEntry.Details.FilesUploadDialog.FilesAttached.Row.Comment.Type(
            string.Format("This file will be removed from a detail line of the Acumatica ERP form: {0}", fileName));
JournalEntry.Details.FilesUploadDialog.Close();

SearchInFiles SearchInFiles = new SearchInFiles();
SearchInFiles.OpenScreen();
SearchInFiles.Summary.DocName.Type(fileName);
SearchInFiles.Details.VerifyRowsCount(1);
SearchInFiles.Details.DeleteFile();
SearchInFiles.Details.VerifyRowsCount(0);
JournalEntry.OpenScreen();

JournalEntry.Last();
JournalEntry.Details.VerifyRowsCount(1);
JournalEntry.Details.SelectRow(1);
JournalEntry.Details.Row.Files.Click();
JournalEntry.Details.FilesUploadDialog.FilesAttached.VerifyRowsCount(0);
JournalEntry.Details.FilesUploadDialog.Close();
```

How to Publish Customization Projects

In your test application, you can upload a customization project and publish it by using any of the following approaches.

Example 1

1. Define the following extension class for the page wrapper of the Customization Projects form (SM204505; System > Customization > Manage).

```
using Controls.CompilationPanel;

namespace Core
{
    public class CustomizationProjects : SM204505_ProjectList
    {
        public c_projects_grid Details
        {
            get { return base.Projects_grid; }
        }

        public CompilationPanel CompilationPanel
        {
            get { return base.CompilationPanel; }
        }
    }
}
```

2. Upload and publish a customization project in your test as follows.

```
CustomizationProjects.OpenScreen();
CustomizationProjects.ActionImport();
CustomizationProjects.Details.UploadForm.SelectFile("C:\share\CustomizationProject.zip");
CustomizationProjects.Details.UploadForm.Upload();
CustomizationProjects.Details.Row.IsWorking.SetTrue();
CustomizationProjects.ActionPublish();
CustomizationProjects.CompilationPanel.Validate(true, "Validation finished successfully.");
CustomizationProjects.CompilationPanel.Publish(true, "Website has been updated.");
CustomizationProjects.CompilationPanel.Close();
```

How to Work with Smart Delays

In this topic, you can find examples of how to work with smart delays. You may need to use smart delays when performing the following tasks:

- Waiting For a New Window to Open
- Waiting for a Condition

Waiting for a Long-Running Operation to Complete

To wait for a long-running operation to complete, use the `WaitForLongOperationToComplete()` method of the `Wait` class.

By default, for all buttons that start long-running operations, waiting for the long-running operation to complete is already implemented in Test SDK. However, you may need to change the default wait action. The code example below shows how to change the default wait action.

```
...
public JournalEntry()
{
 ToolBar.Release.WaitAction = Wait.WaitForLongOperationToComplete;
}

...
JournalEntry journalEntry = new JournalEntry();
journalEntry.OpenScreen();
journalEntry.Insert();
journalEntry.Summary.BranchID.Select("MAIN");
journalEntry.Summary.LedgerID.Select("ACTUAL");
journalEntry.Summary.Description.Type("Test journal entry 1");
journalEntry.Details.New();
journalEntry.Details.Row.AccountID.Type("100000");
journalEntry.Details.Row.ProjectID.Select("X");
journalEntry.Details.Row.CuryDebitAmt.Type(100);
journalEntry.Details.New();
journalEntry.Details.Row.AccountID.Select("101000");
journalEntry.Details.Row.CuryCreditAmt.Type(100);
journalEntry.Release();
```

You can also make the test application wait for the specified result of the long-running operation (success or failure) and check the message on completion of the operation, as shown in the following code.

```
...
public JournalEntry()
{
 ToolBar.Release.WaitAction = () => Wait.WaitForLongOperationToComplete(false, "Operation failed.");
}

...
JournalEntry journalEntry = new JournalEntry();
journalEntry.OpenScreen();
journalEntry.Insert();
journalEntry.Summary.BranchID.Select("MAIN");
journalEntry.Summary.LedgerID.Select("ACTUAL");
journalEntry.Summary.Description.Type("Test journal entry 1");
journalEntry.Details.New();
journalEntry.Details.Row.AccountID.Type("100000");
journalEntry.Details.Row.ProjectID.Select("X");
journalEntry.Details.Row.CuryDebitAmt.Type(100);
journalEntry.Details.New();
journalEntry.Details.Row.AccountID.Select("101000");
journalEntry.Details.Row.CuryCreditAmt.Type(100);
journalEntry.Release();
```

Waiting For a New Window to Open

The code example below shows how to wait for a form to load in another window.

```
...

public JournalEntry()
{
 Details.ToolBar.ViewDocument.WaitAction = Wait.WaitForNewWindowToOpen;
}

...

JournalEntry JournalEntry = new JournalEntry();
JournalEntry.OpenScreen();
JournalEntry.Summary.Module.Select("AP");
JournalEntry.Last();
JournalEntry.Details.ViewDocument();

Bill Bill = new Bill();
Bill.Cancel();
Bill.CloseWindow();
```

Waiting for a Condition

To wait for a condition in your test application, use the `WaitForCondition()` method of the `Wait` class, as shown in the following code.

You do not need to check simple conditions, such as whether a box or a button is visible or available on the form, because these conditions are checked by Test SDK automatically when the test application clicks a button or changes data in a box. You may need to use waiting for a condition when you compose a complex reaction on some action.

```
Wait.WaitForCondition(JournalEntry.ToolBar.Save.IsEnabled, Wait.LongTimeOut);
```

How to Capture Screenshots

The following code shows how to capture a screenshot during the test.

```
Log.Screenshot();
```

How to Get Datetime in the Current User's Timezone

You can get a datetime in the current user's timezone using the following property:

```
Browser.InstanceTime
```

How to use Comparator to compare .xml, .csv, .pdf, Excel files

Methods of a static class Core.Comparator can be used to verify data in a bulk

1. XML data (Comparator.Xml.Compare(...), is used inside Report.Compare(...))
2. PDF files (Comparator.Pdf.Compare(...))
3. Image files (Comparator.Image.Compare(...))
4. Table Data in different formats
    a. .xlsx
    b. .csv
    c. string[][]
    d. Grids in Acumatica UI

Usage:

1. Set up in tests: (for example you need to verify that exported .xlsx is correct)

```
#region Step 2. Export to Excel

                using (TestExecution.CreateTestStepGroup("Export to Excel"))
                {
                    AccountDetails.Export(); //Note: Wait action should be set ToolBar.Export.WaitAction = Wait.Wait.WaitForFileDownloadComplete;
                    Comparator.Table.Compare("AD100_TC2_S2", Core.Core.Browser.Browser.Downloads.GetLastFile());
                }

                #endregion
```

2. Get the baseline (from test results or by manually exporting the file)

3. Add the file to your Tests project (.csv format is recommended), set Copy To Output Directory file property to Copy if Newer (for old-style csproj)

```
<None Include="ERP\FINANCE\General Ledger\Account Details\Data\AD100_TC2_S2.csv">
      <CopyToOutputDirectory>PreserveNewest</CopyToOutputDirectory>
    </None>
```

4. Run the test.

How to Verify string/long/int/DateTime returned by a method

In this topic, you will see an example of verification of sting/long/int/DateTime returned by a custom method.

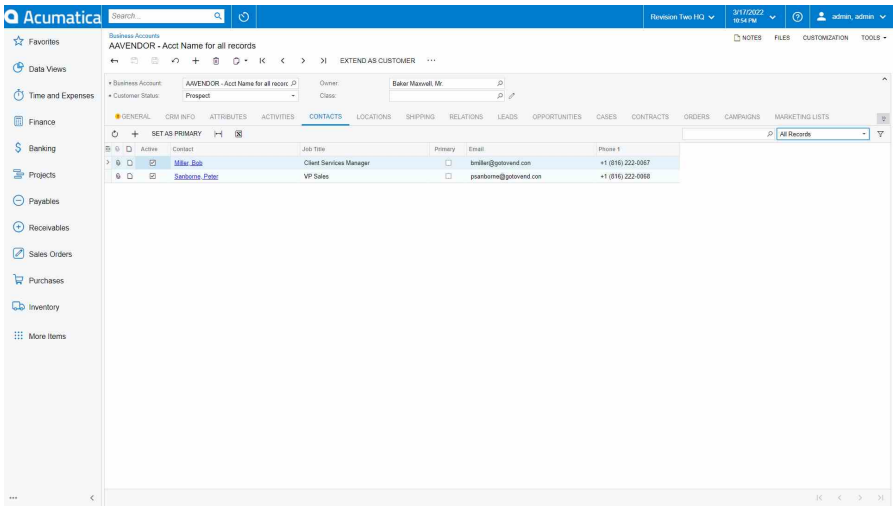
```
using PX.QA.Tools;
...
SomeMethod().VerifyEquals("test", "Method result");

long value = SomeMethod();
value.VerifyEquals(123, "Method result");

"test".VerifyEquals("test", "Some string");
```

Pay attention: these Verify* methods have second argument (verifiableName), which allows you to specify what exactly you are verifying. Don't forget to set it in test.

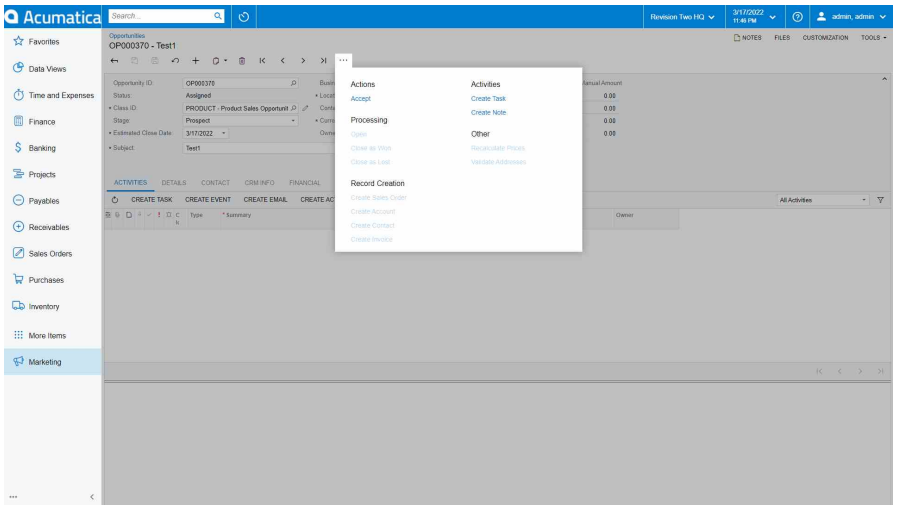
How to work with dynamic controls

Test SDK provides ability to find a control (ToolBarButton, Button,
CheckBox, Input, Selector, FormulaCombo, DropDown, DateSelector, Container, SmartPanel, GroupBox, QuickSearch, Grid) not generated in WG, e.g. created via customization project. There are two methods: DynamicControl and DynamicGrid (last one has been created just for convenience and it invokes DynamicControl inside with specific type parameter).

How to Use DynamicControl/DynamicGrid

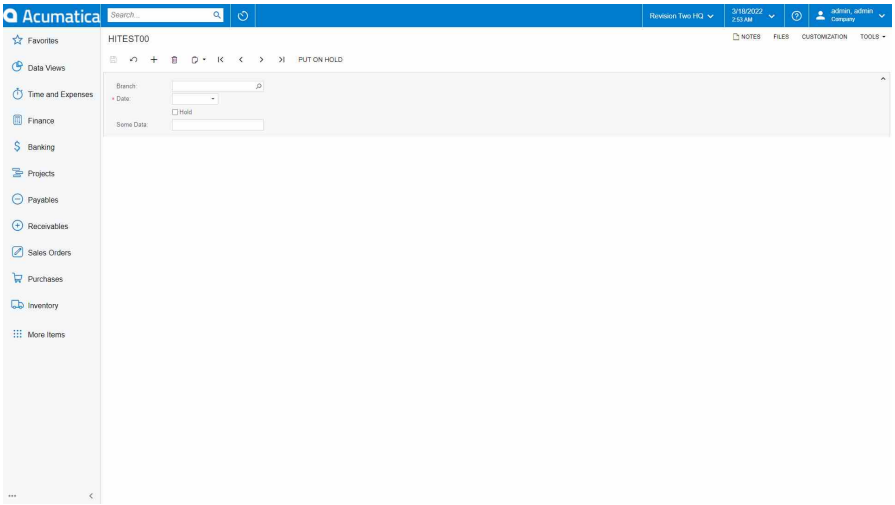



```
BAccountMaint.DynamicGrid("Contacts").RowsCount().VerifyEquals(2);
```




```
Opportunity.DynamicControl<ToolBarButton>("Accept").Click();
```

And few rows for totally new screen




```
OrderPo HiTestScreen = new OrderPo() { ScreenId = "HITEST00" };
HiTestScreen.OpenScreen();
HiTestScreen.DynamicControl<Selector>("Branch").Select("HQ");
var date = HiTestScreen.DynamicControl<DateSelector>("Date");
date.Type(DateTime.Now);
HiTestScreen.DynamicControl<Input>("Some Data").Type("Test");
HiTestScreen.DynamicControl<ToolBarButton>("Put On Hold").Click();
HiTestScreen.DynamicControl<CheckBox>("Hold").GetValue().VerifyEquals(true);
date.IsEnabled().VerifyEquals(true);
```

Known Issues & Helpful Tips

In case you have multiple invocations save the control into variable:

```
var customButton = Wrapper.DynamicControl<ToolBarButton>("Button Name");
customButton.IsVisible().VerifyEquals(false);
customButton.Click();
```