

Dynamic Control Usages in Acumatica Test SDK

This document contains prepared examples using **Dynamic Control** to select and interact with fields using the field's label text. This allows you to select and access fields not accessible from the wrapper/extension files. It is a great way to do 1 or 2 quick actions on a screen without the need to generate wrappers and extensions, or to get around certain fields not working through the extension file.

Accessible Elements:

ToolBarButton, Button, CheckBox, Input, Selector,

FormulaCombo, DropDown, DateSelector, Container,

SmartPanel, GroupBox, QuickSearch, DynamicGrid

ToolBarButton, Dropdown, Dynamic Grid, Selector:

```
OrderSo.OpenScreen();
OrderSo.Insert();

OrderSo.Summary.CustomerID.Select("REARDEN");
OrderSo.Summary.OrderDesc.Type("FILMFEST Billing");
OrderSo.Summary.CustomerOrderNbr.Type("FF_01");
OrderSo.Summary.OrderDate.Type(DateTime.Now);
OrderSo.DocumentDetails.New();
OrderSo.DocumentDetails.Row.InventoryID.Select("FLYERS");
OrderSo.DocumentDetails.Row.SiteID.Type("MAIN");
OrderSo.DocumentDetails.Row.OrderQty.Type(1);
OrderSo.DocumentDetails.Row.CuryUnitPrice.Type(350);
OrderSo.DocumentDetails.New();
OrderSo.DocumentDetails.Row.InventoryID.Select("POSTCARD");
OrderSo.DocumentDetails.Row.SiteID.Type("MAIN");
OrderSo.DocumentDetails.Row.OrderQty.Type(1000);
OrderSo.DocumentDetails.Row.CuryUnitPrice.Type(10);

dynamicGrid = OrderSo.DynamicGrid("Relations");
dynamicGrid.DynamicControl<ToolBarButton>("Add Row").Click();
dynamicGrid.Row.DynamicControl<DropDown>("Role").Select("Source");
dynamicGrid.Row.DynamicControl<DropDown>("Type").Select("Campaign");
dynamicGrid.Row.DynamicControl<Selector>("Document").Select("FILMFEST");
OrderSo.Summary.CuryControlTotal.Type(11385);
OrderSo.Save();
```

Button

```
DialogBoxes.Save();
DialogBoxes.ActionTest();
DialogBoxes.DynamicControl<Selector>("Workgroup").GetValue().VerifyEquals(string.Empty);
DialogBoxes.DynamicControl<Selector>("Owner").GetValue().VerifyEquals(string.Empty);
DialogBoxes.DynamicControl<Button>("Ok").Click();
```

Form Input:

```
ImportByScenario.PreparedData.Row.DynamicControl<Input>("Postal Code").Type("CB3 0JY");
```

--Or--

```
AssignmentMap.Rules.New();
AssignmentMap.Rules.Row.Entity.DynamicControl<DropDown>().Select("Billing Contact");
AssignmentMap.Rules.Row.Entity.PressTab(); AssignmentMap.Rules.Row.FieldName.DynamicControl<DropDown>().Select("Email");
AssignmentMap.Rules.Row.Condition.Select("0"); AssignmentMap.Rules.Row.FieldValue.DynamicControl<Input>().Type("approve@apr.con");
```

--Or--

```
OrderSo.OpenScreen();
OrderSo.Summary.OrderType.Select("SO");
OrderSo.Summary.OrderNbr.Select(LocalOrderId.Split(',')[1]);
```

```
OrderSo.DocumentDetails.Columns.DynamicControl<InputColumnFilter>("Product Modifier").ShowColumn();
```

```
OrderSo.DocumentDetails.Columns.DynamicControl<InputColumnFilter>("Product Modifier").SortDescending();
OrderSo.DocumentDetails.SelectRow(1);
OrderSo.DocumentDetails.Row.DynamicControl<Input>("Product Modifier").GetValue().VerifyEquals("Yes");
OrderSo.DocumentDetails.SelectRow(2);
OrderSo.DocumentDetails.Row.DynamicControl<Input>("Product Modifier").GetValue().VerifyEquals("No");
```

DateSelector:

```
NonStockItem.OpenScreen();
NonStockItem.Summary.InventoryCDLabel.GetValue().VerifyEquals("Inventory ID:");
NonStockItem.Summary.InventoryCD.Select("ONLCOURSE");
NonStockItem.Attributes.SelectRow(NonStockItem.Attributes.Columns.AttributeID, "Pre-Order Release Date");
NonStockItem.Attributes.Row.DynamicControl<DateSelector>("Value").Type(new DateTime(2021, 9, 1));
```

CheckBox:

```
ApprovalMap.Conditions.New();
ApprovalMap.Conditions.Row.Entity.Select("Quote Products");
ApprovalMap.Conditions.Row.Entity.PressTab();
ApprovalMap.Conditions.Row.FieldName.Select("Manual Price");
ApprovalMap.Conditions.Row.Condition.Select("Equals");
ApprovalMap.Conditions.Row.Value.DynamicControl<CheckBox>().SetTrue();
```

--Or--

```
PivotMaint.Properties.Row.Value.DynamicControl<CheckBox>().GetValue().VerifyEquals(true);
```

Selector:

```
StockItem.Conversions.ClickAndAddNewRow();

StockItem.Conversions.Row.FromUnit.DynamicControl<Selector>().Type("PALETT");
StockItem.Conversions.Row.UnitMultDiv.Select("Multiply");
StockItem.Conversions.Row.UnitRate.Type(10000);
```

DynamicGrid:

```
BAccountMaint.DynamicGrid("Contacts").RowCount().VerifyEquals(2);
-OR-
Project.Tasks.Row.Description.Type("2018 Q1");
Project.Tasks.Row.Status.Select("Active");

dynamicGrid = Project.DynamicGrid("Defaults");
dynamicGrid.DynamicControl<ToolBarButton>("Add Row").Click();
dynamicGrid.Row.DynamicControl<Selector>("Account").Select("400000");
dynamicGrid.Row.DynamicControl<Selector>("Default Task").Select("2018Q1");
Project.Save();
```

DropDown Select:

```
AssignmentMap.Conditions.New();
AssignmentMap.Conditions.Row.Entity.DynamicControl<DropDown>().Select("Opportunity");
AssignmentMap.Conditions.Row.FieldName.DynamicControl<DropDown>().Select("Class ID");
AssignmentMap.Conditions.Row.Condition.Select("Equals");
AssignmentMap.Conditions.Row.Value.DynamicControl<Selector>().Select("PROJECT");
AssignmentMap.RuleDetails.RuleType.Select("Employee");
AssignmentMap.RuleDetails.WorkgroupID.Select("Sales/Project Sales");
AssignmentMap.Save();
```

Container

```
StockItem.InspectElementCtrlAltClick();
StockItem.DynamicControl<Container>("Subitems").IsEnabled();
CustomizationMenu.ElementProperties.AspxControl.GetValue().VerifyEquals("Tab");
CustomizationMenu.ElementProperties.CacheType.GetValue().VerifyEquals("InventoryItem");
CustomizationMenu.ElementProperties.GraphName.GetValue().VerifyEquals("InventoryItemMaint");
```

SmartPanel:

```
StockItem_CU400.OpenScreen(true);
var openSmartPanel = StockItem_CU400.Summary.DynamicControl<Button>("Open Smart Panel");
openSmartPanel.WaitAction = Wait.WaitForCallbackToComplete;
openSmartPanel.Click();
StockItem_CU400.DynamicControl<SmartPanel>("Smart Panel").GetValue().VerifyEquals("Smart Panel\r\nCU400_SmartPanel");
StockItem_CU400.DynamicControl<SmartPanel>("Smart Panel").Close();
```

GroupBox:

```
StockItem_CU400.OpenScreen(true);
StockItem_CU400.DynamicControl<Container>("Group Box").Click();
StockItem_CU400.DynamicControl<Container>("Group Box").Element.Text.VerifyContains("GroupBox control presented below:");
```

```

StockItem_CU400.DynamicControl<GroupBox>("Options").Set("First Option");
StockItem_CU400.DynamicControl<GroupBox>("Options").Set("Second Option");
StockItem_CU400.DynamicControl<GroupBox>("Options").Set("Third Option");
StockItem_CU400.Cancel();

```

FormulaCombo:

```

ImportScenario.OpenScreen();
ImportScenario.Summary.Name.Select("Import Opportunity from Salesforce");
ImportScenario.Mapping.New();
ImportScenario.Mapping.Row.ObjectName.DynamicControl<DropDown>().Type("Contact Info ->
Address");
ImportScenario.Mapping.Row.FieldName.DynamicControl<DropDown>().Type("Country");
ImportScenario.Mapping.Row.Value.DynamicControl<FormulaCombo>().Type("='US'");
ImportScenario.Mapping.CommitRows();
ImportScenario.Mapping.RowUp();

```

QuickSearch:

```

GenericInquiry.PageHeader.ExpandScreen.Click();
VA02.DynamicControl<Input>("TESTTEXT1").GetValue().VerifyEquals("(0
) - -");
var sidePanelQuickSearch = VA02.DynamicControl<QuickSearch>();
sidePanelQuickSearch.Type("0");
VA02.Details.RowsCount().VerifyEquals(1);
sidePanelQuickSearch.Reset();
GenericInquiry.PageHeader.CollapseScreen.Click();
GenericInquiry.ViewInquiry();

```

GI Input:

```

GenericInquiry.PageHeader.ExpandScreen.Click();
OrderSoGI.Details.SelectRow("Tracking Number", "398453614");
OrderSoGI.Details.Row.DynamicControl<Input>("Tracking Number").ClickLink();
Browser.Url.ToString().VerifyContains("398453614", "URL contains tracking number");
OrderSoGI.CloseWindow();

```