



# COMMUNITY SUMMIT

## BUSINESS CENTRAL

San Antonio, Texas  
October 13-17, 2024



2024



Microsoft Dynamics

**Business Central**

Education  
Speaker

@ 2024 Dynamic Communities

The largest independent innovation, education,  
and training event for Microsoft Biz Apps



# Introduction to Test Automation in Business Central

*"Testing can show the presence of errors, but not their absence."*

- Edsger W. Dijkstra

The largest independent innovation, education, and training event for Microsoft Biz Apps



## About Me

- 20+ years in Microsoft Dynamics, specializing in Business Central
- Unique dual perspective as a partner & end user
- Member of Summit Programming Committee & Board of Advisors
- Co-host of the popular Dynamics Corner podcast
- Passionate about bringing people together in the Dynamics space



Brad Prendergast



A Dynamics-First, 'For User, By User' Event.

@ 2024 Dynamic Communities



# Agenda

- Objective
- What is automated testing?
- Why use automated testing?
- Setting up an Environment
- Writing Tests
- Running Tests



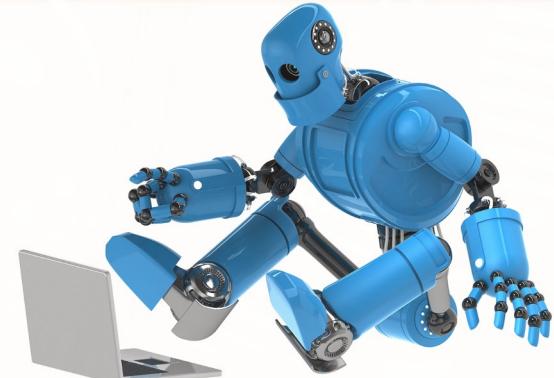


# Session Objective

- Basic Understanding of Automated Testing in Business Central
- Environment Setup
- Writing Tests
- Running Tests

# What is Automated Testing

- Automated testing in Business Central involves using software to run tests automatically on the system to ensure that it functions as expected. This approach reduces manual testing, which can be time-consuming, error-prone, and costly.
- Automated testing in Business Central involves creating a set of test cases that simulate different scenarios and use cases. These test cases are then executed automatically and checks the output against the expected results. The results of each test are analyzed to identify any defects or issues.



# Why use Automated Testing

- Saves Time
- Improves Quality\*
- Repeatable and Reliable
- Reduces Risk and Cost
- Less Disruption / Maximize Error Detection
- Latest Releases





# Automated Testing in Business Central

9.0.51972.NA.DVD\TestToolkit

New | Sort | +

Name

- Outlook
- Prerequisite Components
- RoleTailoredClient
- ServiceTier
- Setup
- SQLDemoDatabase
- Test Assemblies
- TestToolkit
- UpgradeToolKit
- WebClient
- WindowsPowerShellScripts



AL Test Tool

Suite Name: DEFAULT

Test Runner Codeunit: 130450 - Test Runner - Isol...

Manage | Run Tests | Run Selected Tests | Get Test Codeunits | Get Test Codeunits by Range | ...

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration	
Function	134385	ERM Sales Document	<input checked="" type="checkbox"/>	-			
	134385	SalesReturnOrderCreation	<input checked="" type="checkbox"/>	-			
	134385	VATAmountOnReturnOrder	<input checked="" type="checkbox"/>	-			
	134385	ReceiveandInvoiceReturnOrder	<input checked="" type="checkbox"/>	-			
	134385	InvoiceDiscountReturnOrder	<input checked="" type="checkbox"/>	-			
	134385	LineDiscountReturnOrder	<input checked="" type="checkbox"/>	-			
	134385	ApplyDiscountFromReturnOrder	<input checked="" type="checkbox"/>	-			
	134385	LineDiscountOnCreditMemo	<input checked="" type="checkbox"/>	-			
	134385	InvoiceDiscountOnGLEntry	<input checked="" type="checkbox"/>	-			
	134385	ApplyInvoiceFromCreditMemo	<input checked="" type="checkbox"/>	-			
	Working on it...						
	Executing Tests...						
	Test Suite DEFAULT						
	Test Codeunit 134385						
	Test Function VerifyDefaultBinOnSalesLine						
No. of Results with:							
Success 42							
Failure 0							
Skip 0							



# Setting up an Environment

- Environment testing support and limitations

Business solution	Environment	Testing allowed	More details
Online Central solution	Production		Running tests isn't allowed because it might have an adverse effect on your business. Testing can incidentally invoke external systems, like CDS, PayPal, and web hook subscriptions. Invoking these systems may slow down the solution for other users or cause data corruption.
	Sandbox	✓	You can use a sandbox environment to run tests manually to verify functionality on an environment. Running a large number of tests or tests that take a long time (more than 15 minutes per test method) isn't allowed. It's recommended that you don't run tests more than one or two hours a day.
On-premises	Production	✓	For Business Central on-premises, running automated tests is only possible with a Partner license or a license that includes the Application Builder module.  You can disable the ability to run tests by turning off <b>Enable Test Automation</b> ( <code>TestAutomationEnabled</code> ) on the Business Central Server instance. For more information, see <a href="#">Configuring Business Central Server - General Settings</a> .
Container-based development environment		✓	This setup should be the default environment for running large number of tests or setting up CI/CD gates. For more information, see <a href="#">Running a Container-Based Development Environment</a> or <a href="#">Running Tests In Containers</a> .

Source: <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/developer/devenv-testing-application>



# Setting up an Environment

- Available on Microsoft Dynamics 365 Business Central installation “DVD”

Applications > testframework >				
Name	Date modified	Type	Size	
performance toolkit	3/3/2023 3:47 PM	File folder		
TestLibraries	2/3/2023 8:13 AM	File folder		
TestRunner	3/3/2023 3:47 PM	File folder		
teststabilitytools	2/3/2023 8:13 AM	File folder		

# Setting up an Environment

- BCContainerHelper: NewBCContainer

```
$startTime = [DateTime]::Now
Write-Host -ForegroundColor Green "$startTime : New-BCContainer start"

$artifactUrl = Get-BCArtifactUrl -version 22 -country "us" -select Latest -type Sandbox
Write-Host -ForegroundColor Yellow ($artifactUrl)

New-BCContainer `-
    -accept_eula `-
    -accept_outdated `-
    -alwaysPull `-
    -artifactUrl $artifactUrl `-
    -auth NavUserPassword `-
    -containerName $containerName `-
    -Credential $credential `-
    -includeTestToolkit `-
    -isolation hyperv `-
    -updateHosts `-

$timeSpend = [Math]::Round(([DateTime]::Now.Subtract($startTime)).TotalSeconds)
$stopTime = [DateTime]::Now

Write-Host -ForegroundColor Green "$stopTime : New-BCContainer took $timespend seconds"
```

```
-includeTestToolkit
-includeTestLibrariesOnly
-includeTestFrameworkOnly
```



# Setting up an Environment

- BCContainerHelper: ImportTestToolkitToBCContainer

# Writing Tests

- Where do you start?
  - Testing Plan
  - Testing Scenarios
  - Code Coverage
  - Test Driven Development



# Writing Tests

- What do we test?
  - Test Pages
  - Performance Toolkit Extension
  - Test Codeunits
  - UI Handlers



# Writing Tests

- TestApp Dependencies

```
...  
  "dependencies": [  
    {  
      "id": "a3c9c3aa-6c42-4cba-aaea-33d9a155f1b1",  
      "name": "MainApp",  
      "publisher": "Default publisher",  
      "version": "24.0.0.0"  
    },  
    {  
      "id": "dd0be2ea-f733-4d65-bb34-a28f4624fb14",  
      "publisher": "Microsoft",  
      "name": "Library Assert",  
      "version": "24.0.0.0"  
    },  
    {  
      "id": "e7320ebb-08b3-4406-b1ec-b4927d3e280b",  
      "publisher": "Microsoft",  
      "name": "Any",  
      "version": "24.0.0.0"  
    },  
    {  
      "id": "5d86850b-0d76-4eca-bd7b-951ad998e997",  
      "publisher": "Microsoft",  
      "name": "Tests-TestLibraries",  
      "version": "24.0.0.0"  
    },  
    {  
      "id": "9856ae4f-d1a7-46ef-89bb-6ef056398228",  
      "publisher": "Microsoft",  
      "name": "System Application Test Library",  
      "version": "24.0.0.0"  
    }  
  ]
```



# Writing Tests

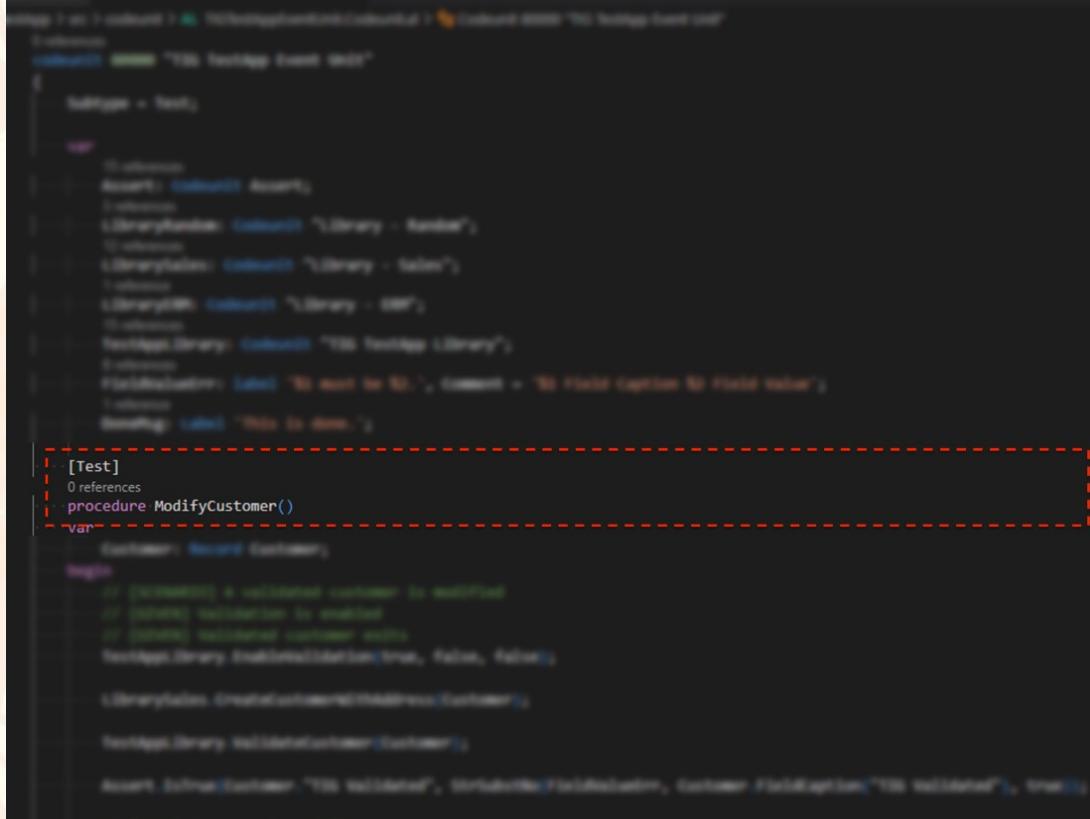
- Test Codeunits: Subtype Property



```
codeunit 80000 "TI6 TestApp Event Unit"
{
    Subtype = Test;
    var
        [Test]
        [Event]
        procedure Multiplier();
        var
            Customer: Record Customer;
        begin
            // [3020000] A validated customer is modified
            // [300000] Validation is enabled
            // [304000] Validation customer exists
            TestAppLibrary.ValidateCustomer(true, false, false);
            LibraryTables.CreateCustomer("Customer");
            TestAppLibrary.ValidateCustomer("Customer");
            Assert.IsTrue(Customer."Customer No." = 'Customer', 'Customer Field is not set');
        end;
    }
}
```

# Writing Tests

- Test Codeunits: Test Attribute on Methods



```
codeunit 1000 TestCustomerCodeunit {  
    [Test]  
    procedure ModifyCustomer()  
    var  
        Customer: Record Customer;  
    begin  
        // (codeunit) A validated customer is modified  
        // (codeunit) Validation is enabled  
        // (codeunit) Validated customer exists  
        TestModifyCustomer(true, false, false);  
  
        LibrarySales.CreateCustomerWithAddress(Customer);  
  
        TestModifyCustomer(false, true, false);  
  
        Assert.Error(Customer."Test Is Failed", TestModifyCustomer, CustomerValidation."Test Is Failed", true);  
    end;  
}  
library  
    TestLibrary;  
    LibraryCustomer;  
    LibrarySales;  
    LibraryXunit;  
    LibraryXunitTest;  
    LibraryXunitTest;
```



# Writing Tests

- Test Codeunits: Handler Methods

Method type	Purpose	Signature
<b>MessageHandler</b>	Handles Message statements.	<Function name>(<Message>: Text[1024])Error! Filename not specified.
<b>ConfirmHandler</b>	Handles Confirm statements.	<Function name>(<Question>: Text[1024]; var <Reply>: Boolean)
<b>StrMenuHandler</b>	Handles StrMenu statements.	<Function name>(<Options: Text[1024]; var <Choice>: Integer; <Instruction>: Text[1024])
<b>ReportHandler</b>	Handles specific reports. If you create a ReportHandler method, then that method replaces all code for running the report, including the request page, and a RequestPageHandler isn't called. Only create a RequestPageHandler if you aren't using a ReportHandler.	<Function name>(var <Report>: Report <report id>)



# Writing Tests

- Gherkin Scenario

```
2  
3  
4 ... // [SCENARIO] A description of the behavior you're going to test  
5 ... // [GIVEN] Beginning state of the scenario; what is needed to run the test  
6 ... // [WHEN] A specific action the user takes; Something Happens  
7 ... // [THEN] A testable outcome; usually caused by the [WHEN] Action; Do your test  
8
```



# Writing Tests

- Assert Library

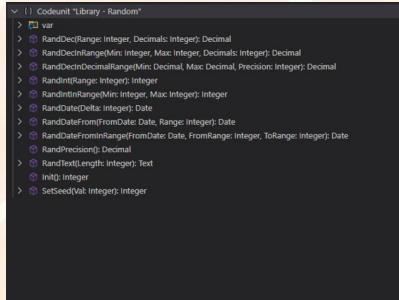
```
/// <summary>
/// This module provides functions for easy verification of expected values and error handling in test code.
/// </summary>
codeunit 130002 "Library Assert"
{
```

```
< () Codeunit "Library Assert"
  OnRun()
> ⚡ var
> ⓘ IsTrue(Condition: Boolean; Msg: Text)
> ⓘ IsFalse(Condition: Boolean; Msg: Text)
> ⓘ AreEqual(ExpectedVariant: Variant; ActualVariant: Variant; Msg: Text)
> ⓘ AreEqual(Expected: Dictionary of [Text, Text]; Actual: Dictionary of [Text, Text])
> ⓘ AreEqualDateTime(Expected: DateTime; Actual: DateTime; Msg: Text)
> ⓘ AreNotEqual(ExpectedVariant: Variant; ActualVariant: Variant; Msg: Text)
> ⓘ AreNearlyEqual(Expected: Decimal; Actual: Decimal; Delta: Decimal; Msg: Text)
> ⓘ AreNotNearlyEqual(Expected: Decimal; Actual: Decimal; Delta: Decimal; Msg: Text)
> ⓘ Fail(Msg: Text)
> ⓘ RecordsIsEmpty(RecVariant: Variant)
> ⓘ RecordsNotEmpty(RecVariant: Variant)
> ⓘ TabletsEmpty(TableNo: Integer)
> ⓘ TabletsNotEmpty(TableNo: Integer)
> ⓘ RecRefIsEmpty(var RecordRef: RecordRef)
> ⓘ RecRefIsNotEmpty(var RecordRef: RecordRef)
> ⓘ RecordCount(RecVariant: Variant; ExpectedCount: Integer)
> ⓘ KnownFailure(Expected: Text; WorkItemNo: Integer)
> ⓘ ExpectedError(Expected: Text)
> ⓘ ExpectedErrorCode(Expected: Text)
> ⓘ ExpectedMessage(Expected: Text; Actual: Text)
> ⓘ AssertRecordNotFound()
> ⓘ AssertRecordAlreadyExists()
> ⓘ AssertNothingInsideFilter()
> ⓘ AssertNoFilter()
> ⓘ AssertPrimRecordNotFound()
> ⓘ TypeOf(ValueVariant: Variant) : Integer
> ⓘ TypeNameOf(ValueVariant: Variant) : Text
> ⓘ UnsupportedTypeName(ValueVariant: Variant) : Text
> ⓘ Equal(LeftVariant: Variant; RightVariant: Variant) : Boolean
> ⓘ IsNumber(ValueVariant: Variant) : Boolean
> ⓘ EqualNumbers(Left: Decimal; Right: Decimal) : Boolean
> ⓘ VerifyFailure(ErrorCodeExpected: Text; FailureText: Text)
```

# Writing Tests

- Microsoft\_Test

<b>CU</b>	130440	Library - Random
<b>CU</b>	130509	Library – Sales
<b>CU</b>	130512	Library - Purchase
<b>CU</b>	131000	Library – Utility

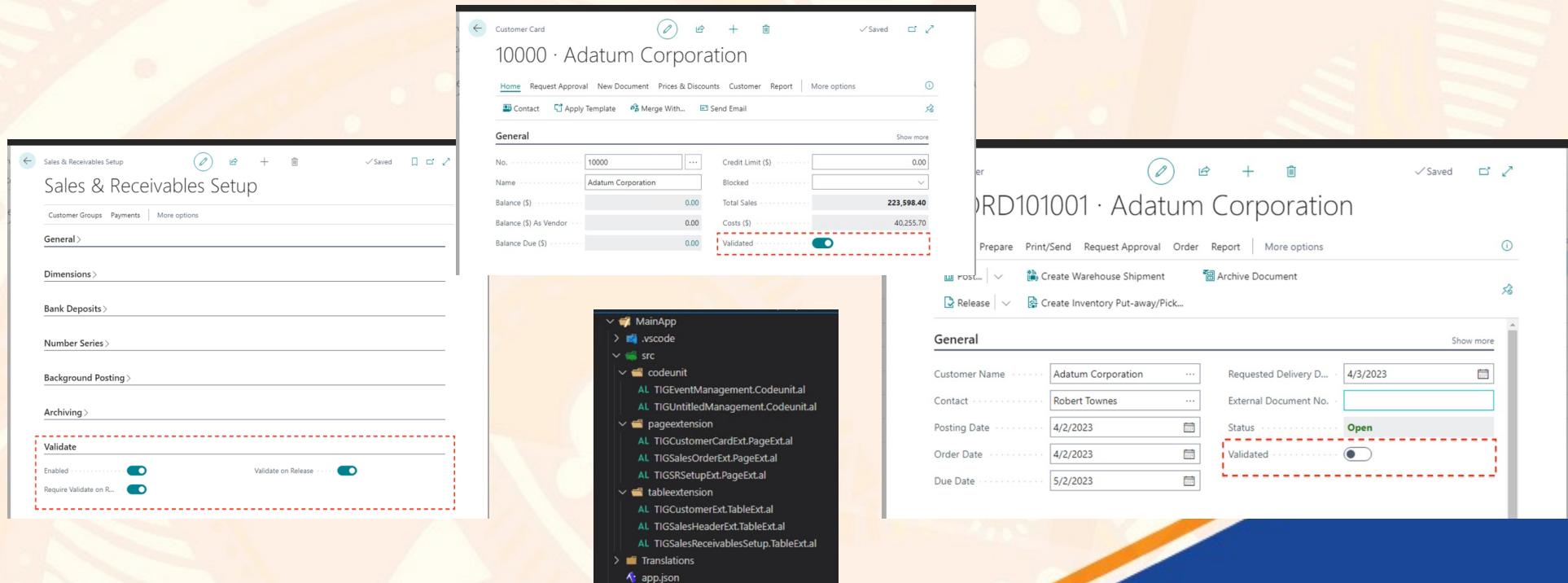


```

    () Codeunit "Library - Utility"
    > var
    > CreateNoSeries(var NoSeries: Record "No. Series", Default: Boolean, Manual: Boolean, DateOrder: Boolean)
    > CreateNoSeriesLine(var NoSeriesLine: Record "No. Series Line", SeriesCode: Code[20], StartingNo: Code[20], ...)
    > CreateNoSeriesRelationship(Code: Code[20], SeriesCode: Code[20])
    > CreateRecordLink(RecVar: Variant): Integer
    > CreateRecordLink(RecVar: Variant, LinkType: Option): Integer
    > CheckFieldExistenceInTable(TableNo: Integer, FieldName: Text[30]): Boolean
    > CheckFileNotEmpty(FileName: Text): Boolean
    > CompareTwoRecords(RecRef1: RecordRef, RecRef2: RecordRef, FieldCountsToBeIgnored: Integer, DiscardDate...)
    > ConvertMilliSecToHours(TimePeriod: Decimal): Decimal
    > ConvertHoursToMilliSec(TimePeriod: Decimal): Decimal
    > ConvertNumericToText(NumericCode: Text): Text
    > ConvertCRLFToBackSlash(TextIn: Text): Text
    > FindControl(ObjectNo: Integer, FieldNo: Integer): Boolean
    > FindEditable(ObjectNo: Integer, FieldNo: Integer): Boolean
    > FindFieldNoInTable(TableNo: Integer, FieldName: Text[30]): Integer
    > FindMaxValueForField(ObjectNo: Integer, FieldNo: Integer): Integer
    > FindMinValueForField(ObjectNo: Integer, FieldNo: Integer): Integer
    > FindVisible(ObjectNo: Integer, FieldNo: Integer): Boolean
    > GetFieldLength(TableNo: Integer, FieldNo: Integer): Integer
    > GetLastTransactionNo(): Integer
    > GetNewRecNo(RecVariant: Variant, FieldNo: Integer): Integer
    > GetNewLineNo(RecRef: RecordRef, FieldNo: Integer): Integer
    > GetGlobalNoSeriesCode(): Code[20]
    > GetNextNoSeriesSalesDate(NoSeriesCode: Code[20]): Date
    > GetNextNoSeriesPurchaseDate(NoSeriesCode: Code[20]): Date
    > GetPropertyValueForControl(ObjectNo: Integer, FieldNo: Integer, PropertyName: Text[30], SuppressError: Bo...
    > FindReturnShipmentHeader(var ReturnShipmentHeader: Record "Return Shipment Header", Refur...)
    > GetDropShipment(var PurchaseHeader: Record "Purchase Header")
    > GetInvRoundingAccountID(VendPostGroup: VendorPostingGroupCode: Code[20]; Code[20])
    > GetMaxFieldOptionIndex(TableNo: Integer, FieldNo: Integer): Integer
    > GetFieldMaxText(RecVar: Variant, FieldNo: Integer)
    > GetNetRoot(): Text
  
```

# Writing Tests

- Example App: MainApp



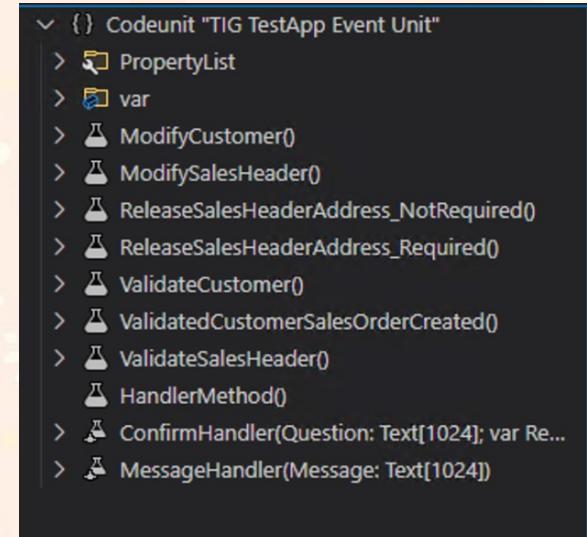
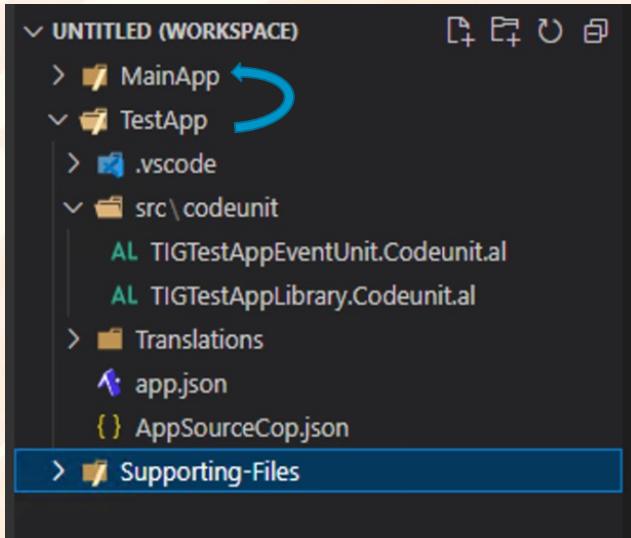
The image displays three screenshots of Microsoft Dynamics 365 application interfaces, illustrating validation logic across different entities and stages.

- Sales & Receivables Setup:** Shows the "Validate" section under "General". It includes two toggle switches: "Enabled" and "Require Validate on Release". Both are highlighted with red dashed boxes.
- Customer Card:** Shows the "General" tab for customer "10000 · Adatum Corporation". A "Validated" toggle switch is highlighted with a red dashed box.
- Sales Order:** Shows the "General" tab for sales order "ORD101001 · Adatum Corporation". A "Validated" toggle switch is highlighted with a red dashed box.

A central screenshot shows the file structure of the "MainApp" solution, specifically the "SRC" folder containing "codeunit" and "pageextension" files, which likely implement the validation logic seen in the UI.

# Writing Tests

- Example App: TestApp



The screenshot shows the generated AL code for the 'Codeunit "TIG TestApp Event Unit"' in the 'src\codeunit' folder. The code includes several methods:

- > PropertyList
- > var
- > ModifyCustomer()
- > ModifySalesHeader()
- > ReleaseSalesHeaderAddress\_NotRequired()
- > ReleaseSalesHeaderAddress\_Required()
- > ValidateCustomer()
- > ValidatedCustomerSalesOrderCreated()
- > ValidateSalesHeader()
- > HandlerMethod()
- > ConfirmHandler(Question: Text[1024]; var Re...)
- > MessageHandler(Message: Text[1024])

# Writing Tests

- Example App: TestApp Methods

```
codeunit 80000 "TIG TestApp Event Unit"
{
    [Test]
    O references
    procedure ModifySalesHeader()
    var
        Customer: Record Customer;
        SalesHeader: Record "Sales Header";
        CustomerNo: Code[28];
    begin
        // [SCENARIO] A validated sales header is modified
        // [GIVEN] Validation is enabled
        // [GIVEN] Validated sales header exists
        TestApplibLibrary.EnableValidation(true, false, false);

        LibrarySales.CreateCustomerWithAddress(Customer);
        CustomerNo := Customer."No.";

        LibrarySales.CreateSalesHeader(SalesHeader, "Sales Document Type":>Order, CustomerNo);

        TestApplibLibrary.ValidateSalesHeader(SalesHeader);

        Assert.IsTrue(SalesHeader."TIG Validated", StrSubstNo(FieldValueErr, SalesHeader.FieldCaption("TIG Validated"), true));

        // [WHEN] Sales header is modified
        SalesHeader.Validate("Sell-to Address", LibraryRandom.RandText(MaxStrLen(SalesHeader."Sell-to Address")));

        // [THEN] Sales header is not validated
        Assert.Equal(SalesHeader."TIG Validated", false);
    end;

    [Test]
    O references
    procedure ModifyCustomer()
    var
        Customer: Record Customer;
    begin
        // [SCENARIO] A validated customer is modified
        // [GIVEN] Validation is enabled
        // [GIVEN] Validated customer exists
        TestApplibLibrary.EnableValidation(true, false, false);

        LibrarySales.CreateCustomerWithAddress(Customer);

        TestApplibLibrary.ValidateCustomer(Customer);

        Assert.IsTrue(Customer."TIG Validated", StrSubstNo(FieldValueErr, Customer.FieldCaption("TIG Validated"), true));

        // [WHEN] Customer is modified
        Customer.Validate(Address, LibraryRandom.RandText(MaxStrLen(Customer.Address)));

        // [THEN] Customer is not validated
        Assert.Equal(Customer."TIG Validated", false);
    end;
}
```

```
codeunit 80000 "TIG TestApp Event Unit"
{
    [Test]
    O references
    procedure ModifySalesHeader()
    var
        Customer: Record Customer;
        SalesHeader: Record "Sales Header";
        CustomerNo: Code[28];
    begin
        // [SCENARIO] A validated sales header is modified
        // [GIVEN] Validation is enabled
        // [GIVEN] Validated sales header exists
        TestApplibLibrary.EnableValidation(true, false, false);

        LibrarySales.CreateCustomerWithAddress(Customer);
        CustomerNo := Customer."No.";

        LibrarySales.CreateSalesHeader(SalesHeader, "Sales Document Type":>Order, CustomerNo);

        TestApplibLibrary.ValidateSalesHeader(SalesHeader);

        Assert.IsTrue(SalesHeader."TIG Validated", StrSubstNo(FieldValueErr, SalesHeader.FieldCaption("TIG Validated"), true));

        // [WHEN] Sales header is modified
        SalesHeader.Validate("Sell-to Address", LibraryRandom.RandText(MaxStrLen(SalesHeader."Sell-to Address")));

        // [THEN] Sales header is not validated
        Assert.Equal(SalesHeader."TIG Validated", false);
    end;

    [Test]
    O references
    procedure ReleaseSalesHeaderAddress_NotRequired()
    var
        SalesHeader: Record "Sales Header";
    begin
        // [SCENARIO] User releases sales order that is not validated and validation on release is enabled
        // [GIVEN] Validation is enabled and required on release
        // [GIVEN] Sales order exists and is not validated
        TestApplibLibrary.EnableValidation(true, false, false);

        LibrarySales.CreateSalesOrder(SalesHeader);
        SalesHeader.Modify();

        Assert.IsFalse(SalesHeader."TIG Validated", StrSubstNo(FieldValueErr, SalesHeader.FieldCaption("TIG Validated"), false));

        // [WHEN] Sales Header released
        TestApplibLibrary.ReleaseSalesDocumentCheckSkip(SalesHeader, false, true);

        // [THEN] Sales Header is released
        Assert.Equal(SalesHeader.Status, "Sales Document Status":>Released);
    end;
}
```

# Writing Tests

- Example App: TestApp Methods

```
[Test]
0 references
procedure ReleaseSalesHeaderAddress_Required()
var
    Customer: Record Customer;
    SalesHeader: Record "Sales Header";
    SalesLine: Record "Sales Line";
    CustomerNo: Code[20];
    GLAccNo: Code[20];

begin
    // [SCENARIO] User releases sales order that is not validated and validation on release is enabled
    // [GIVEN] Validation is enabled and required on release
    // [GIVEN] Sales order exists and is not validated
    TestAppLibrary.EnableValidation(true, true, true);

    LibrarySales.CreateCustomerWithAddress(Customer); <-----|
    CustomerNo := Customer."No.";

    GLAccNo := LibraryERM.CreateGLAccountWithSalesSetup(); <-----|
    LibrarySales.CreateSalesHeader(SalesHeader, "Sales Document Type":>Order, CustomerNo); <-----|
    SalesHeader.Modify();

    LibrarySales.CreateSalesLine(SalesLine, SalesHeader, "Sales Line Type":>"G/L Account", GLAccNo, LibraryRandom.RandInt(5));
    SalesLine.Modify();

    Assert.IsFalse(SalesHeader."TIG Validated", StrSubstNo(FieldValueErr, SalesHeader.FieldCaption("TIG Validated"), false));

    // [WHEN] Sales Header released
    TestAppLibrary.ReleaseSalesDocumentCheckSkip(SalesHeader, false, true);

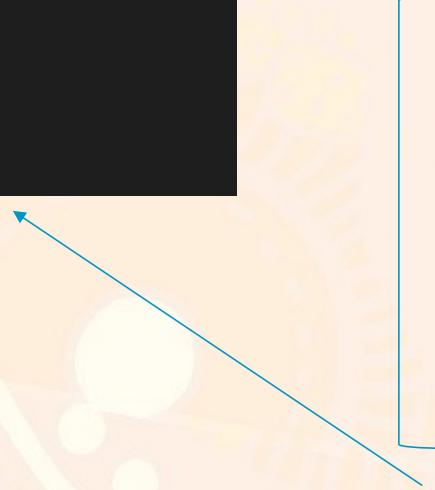
    // [THEN] Sales Header is not released
    Assert.Equal(SalesHeader.Status, "Sales Document Status":>Open);
end;
```

# Writing Tests

- Example App: TestApp Handlers

```
codeunit 50100 "TIG Untitled Management"
 1 reference
procedure ConfirmMessage()
var
  ConfirmMsg: Label 'Are you sure that you want to do this?';
  DoneMsg: Label 'This is done.';
  NotDoneMsg: Label 'This is not done.';

begin
  if Confirm(ConfirmMsg, false) then
    Message(DoneMsg)
  else
    Message(NotDoneMsg);
end;
```



```
[Test]
[HandlerFunctions('MessageHandler,ConfirmHandler')]
0 references
procedure HandlerMethod()
begin
  // [SCENARIO] Demonstrate handler methods
  // [GIVEN] Handlers assigned
  // [WHEN] Test performance with confirm and message
  TestAppLibrary.ConfirmHandler();
  // [THEN] No Assert
end;

[ConfirmHandler]
0 references
procedure ConfirmHandler(Question: Text[1024]; var Reply: Boolean);
begin
  Reply := true;
end;

[MessageHandler]
0 references
procedure MessageHandler(Message: Text[1024]);
begin
  Assert.IsTrue(StrPos(Message, DoneMsg) > 0, Message);
end;
```

```
codeunit 80001 "TIG TestApp Library"
 1 reference
procedure ConfirmHandler()
var
  UntitledManagement: Codeunit "TIG Untitled Management";
begin
  UntitledManagement.ConfirmMessage();
end;
```



# Writing Tests

- Example App:  
TestApp Library

```
1 reference
codeunit 80001 "TIG TestApp Library"
{
    3 references
    procedure ValidateCustomer(var Customer: Record Customer)
    var
        UntitledManagement: Codeunit "TIG Untitled Management";
    begin
        UntitledManagement.ValidateCustomer(Customer, true);
    end;

    2 references
    procedure ValidateSalesHeader(var SalesHeader: Record "Sales Header")
    var
        UntitledManagement: Codeunit "TIG Untitled Management";
    begin
        UntitledManagement.ValidateSalesHeader(SalesHeader, true);
    end;

    7 references
    procedure EnableValidation(Enabled: Boolean; ReqValidateonRelease: Boolean; ValidateonRelease: Boolean)
    var
        SalesReceivablesSetup: Record "Sales & Receivables Setup";
    begin

        if not SalesReceivablesSetup.FindFirst() then
            SalesReceivablesSetup.Init();

        SalesReceivablesSetup."TIG Enabled" := Enabled;
        SalesReceivablesSetup."TIG Req Validate on Release" := ReqValidateonRelease;
        SalesReceivablesSetup."TIG Validate on Release" := ValidateonRelease;
        if not SalesReceivablesSetup.Insert() then
            SalesReceivablesSetup.Modify();
    end;

    2 references
    procedure ReleaseSalesDocumentCheckSkip(var SalesHeader: Record "Sales Header"; SkipCheckReleaseRestrictions: Boolean; Commit: Boolean): Boolean
    var
        ReleaseSalesDocument: Codeunit "Release Sales Document";
    begin
        if SkipCheckReleaseRestrictions then
            ReleaseSalesDocument.SetSkipCheckReleaseRestrictions();

        if Commit then
            Commit();

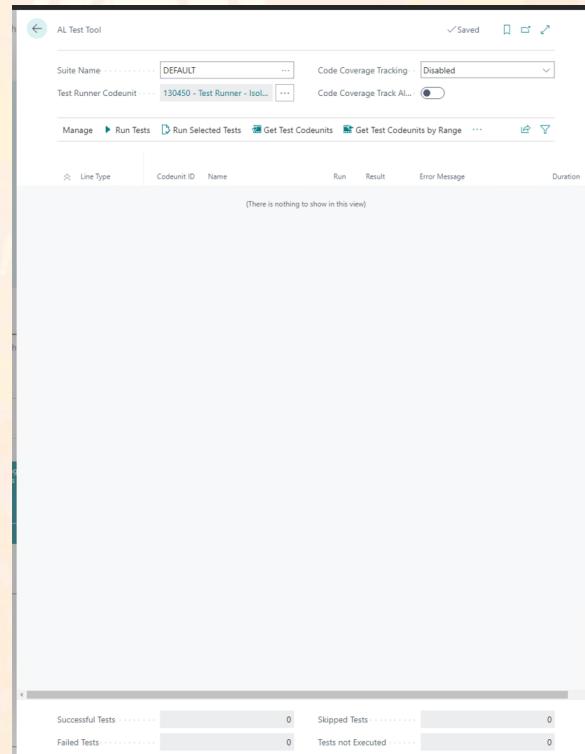
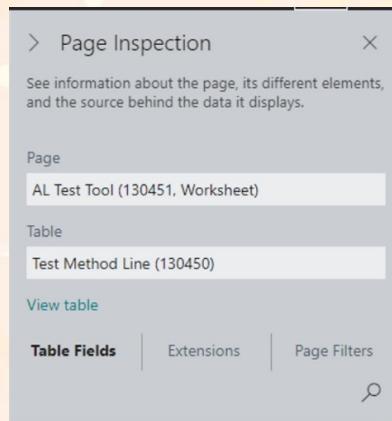
        exit(ReleaseSalesDocument.Run(SalesHeader));
    end;

    1 reference
    procedure ConfirmHandler()
    var
        UntitledManagement: Codeunit "TIG Untitled Management";
    begin
        UntitledManagement.ConfirmMessage();
    end;
}
```

Dynamic Communities

# Running Tests

- AL Test Tool

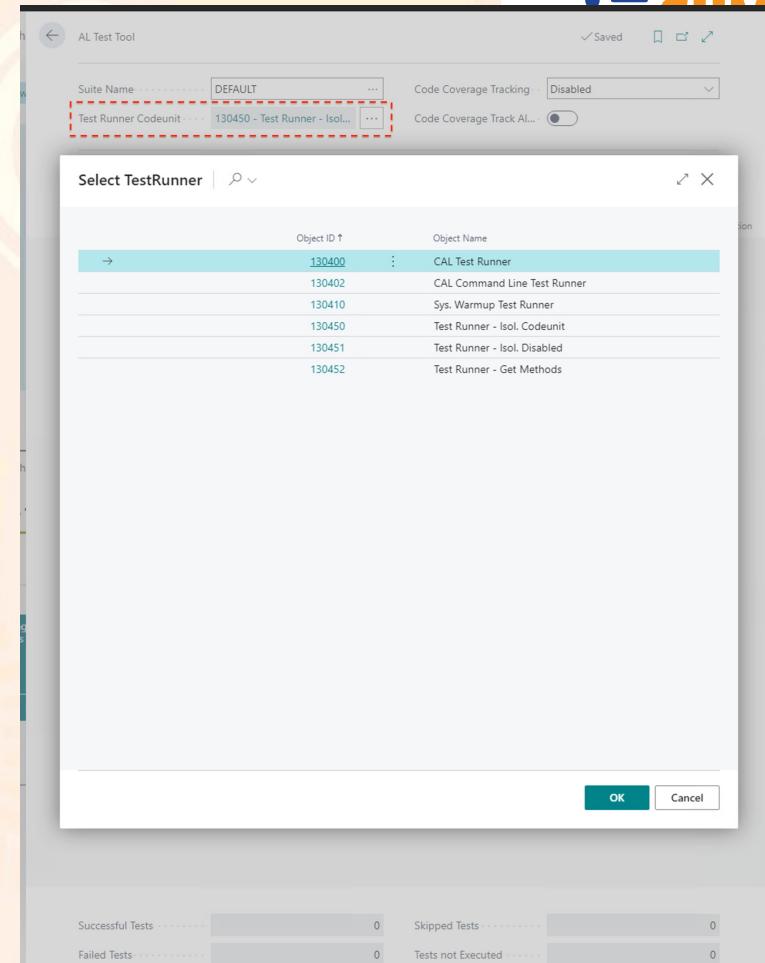


# Running Tests

- Test Runner Codeunit

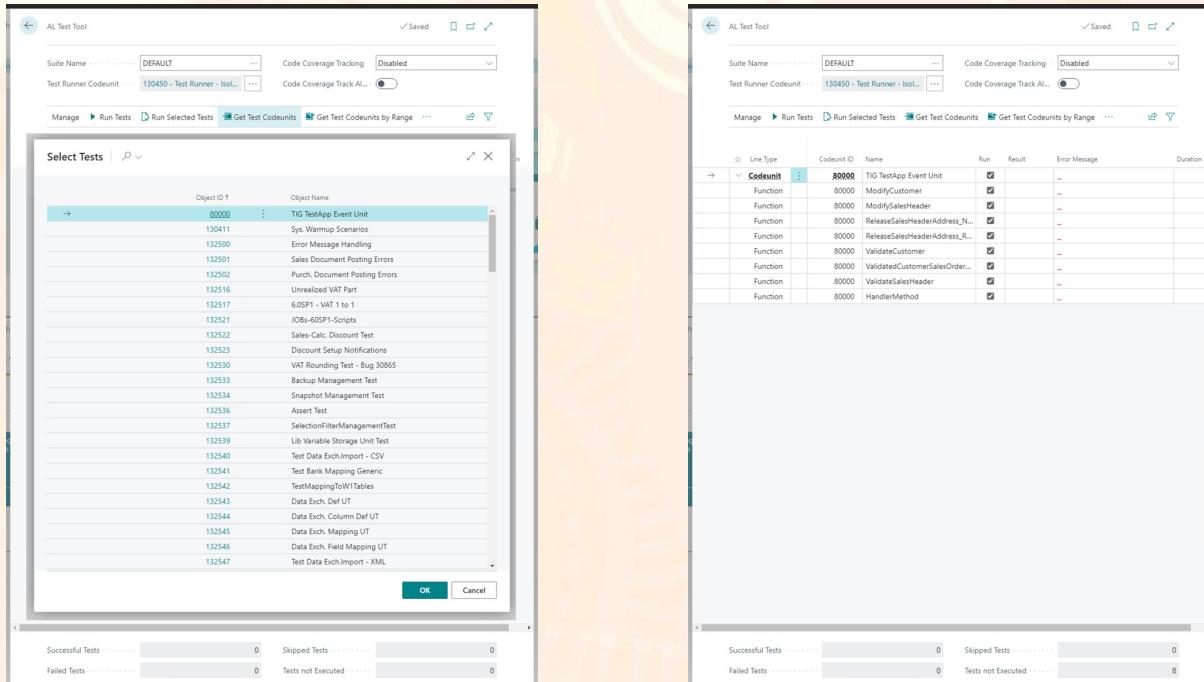
## TestIsolation Property

Value	Description
<b>Disabled</b>	Do not roll back any changes to the database. Tests are not isolated from each other. This is the default value.
<b>Codeunit</b>	Roll back all changes to the database after each test codeunit executes.
<b>Function</b>	Roll back all changes to the database after each test method executes.



# Running Tests

- Test Codeunits



The image shows two side-by-side screenshots of the AL Test Tool interface.

**Left Screenshot:** A modal dialog titled "Select Tests" is open. It lists various test codeunits under "Object Name". One item is selected: "TIG TestApp Event Unit". At the bottom of the dialog, there are four status bars: "Successful Tests" (0), "Skipped Tests" (0), "Failed Tests" (0), and "Tests not Executed" (0).

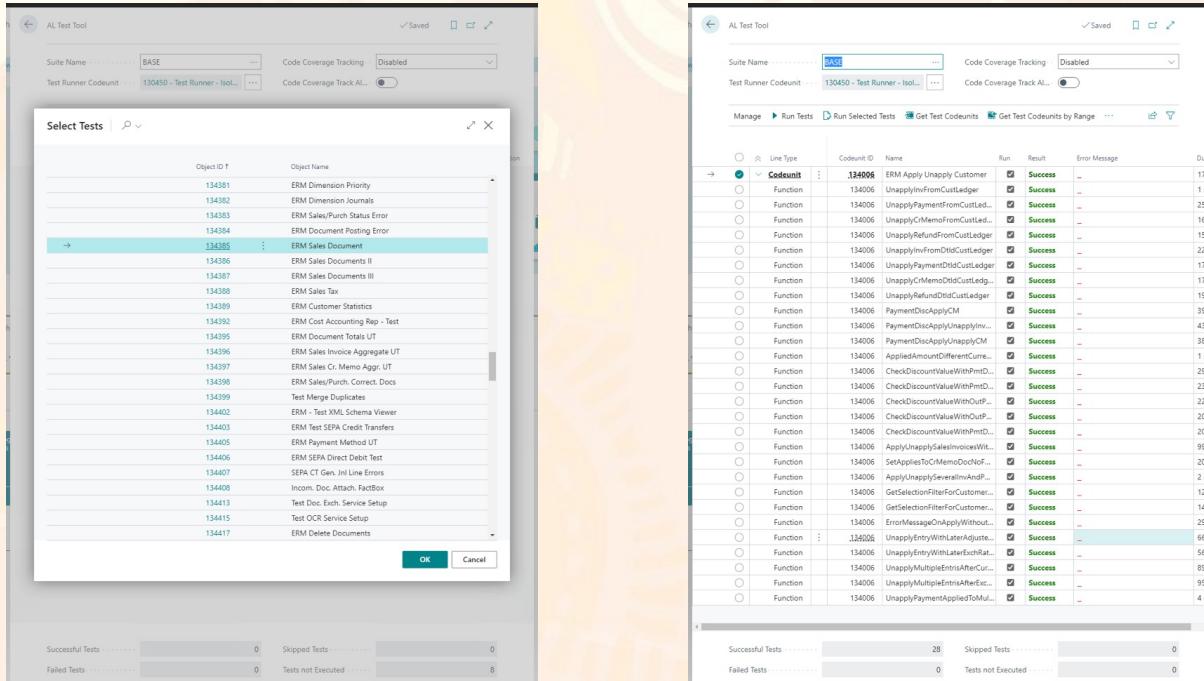
Object Name
130411 Sys_Warmup Scenarios
132500 Error Message Handling
132501 Sales Document Posting Errors
132502 Purch. Document Posting Errors
132516 Unrealized VAT Part
132517 60SP1 - VAT 1 to 1
132521 JOBs-60SP1-Scripts
132522 Sales-Calc_Discount Test
132523 Discount Setup Notifications
132530 VAT Rounding Test - Bug 30865
132533 Backup Management Test
132534 Snapshot Management Test
132536 Assert Test
132537 SelectionFilterManagementTest
132539 Lub Variable Storage Unit Test
132540 Test Data Exch.Import - CSV
132541 Test Bank Mapping Generic
132542 TestMappingToV1Tables
132543 Data Exch. Def UT
132544 Data Exch. Column Def UT
132545 Data Exch. Mapping UT
132546 Data Exch. Field Mapping UT
132547 Test Data Exch.Import - XML

**Right Screenshot:** The main AL Test Tool window displays a table of test results for the selected codeunit. The table has columns: Line Type, Codeunit ID, Name, Run, Result, Error Message, and Duration. Most tests have passed (Run checked, Result green). One test, "ReleaseSalesHeaderAddress\_N...", has failed (Run checked, Result red). The bottom of the window shows summary statistics: Successful Tests (0), Skipped Tests (0), Failed Tests (0), and Tests not Executed (8).

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Function	80000	TIG TestApp Event Unit	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	
Function	80000	ModifyCustomer	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	
Function	80000	ModifySalesHeader	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	
Function	80000	ReleaseSalesHeaderAddress_N...	<input checked="" type="checkbox"/>	<span style="color: red;">✗</span>	Test failed	
Function	80000	ValidateCustomer	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	
Function	80000	ValidateCustomerSalesOrder...	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	
Function	80000	ValidateSalesHeader	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	
Function	80000	HandleMethod	<input checked="" type="checkbox"/>	<span style="color: green;">✓</span>	-	

# Running Tests

- Base Tests



The image shows two screenshots of the AL Test Tool interface.

**Select Tests Dialog:** This window shows a list of tests categorized by Object ID. One test, "134495: ERM Sales Document", is highlighted with a teal border. At the bottom, there are buttons for "OK" and "Cancel".

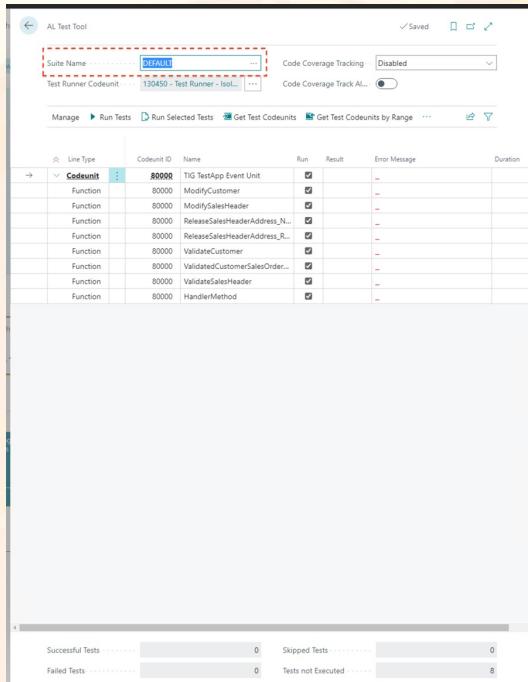
Object ID	Object Name
134381	ERM Dimension Priority
134382	ERM Dimension Journals
134383	ERM Sales/Purch Status Error
134384	ERM Document Posting Error
134385	<b>ERM Sales Document</b>
134386	ERM Sales Documents II
134387	ERM Sales Documents III
134388	ERM Sales Tax
134389	ERM Customer Statistics
134392	ERM Cost Accounting Rep - Test
134395	ERM Document Totals UT
134396	ERM Sales Invoice Aggregate UT
134397	ERM Sales Cr. Memo Aggr. UT
134398	ERM Sales/Purch. Correct. Docs
134399	Test Merge Duplicates
134402	ERM - Test XML Schema Viewer
134403	ERM Test SEPA Credit Transfers
134405	ERM Payment Method UT
134406	ERM SEPA Direct Debit Test
134407	SEPA CT Gen. Jnl Line Errors
134408	Incom. Doc. Attach. FactBox
134413	Test Doc. Exch. Service Setup
134415	Test OCR Service Setup
134417	ERM Delete Documents

**Test Results Grid:** This window shows a detailed list of test results for the selected suite. The table includes columns for Line Type, Codunit ID, Name, Run, Result, Error Message, and Duration.

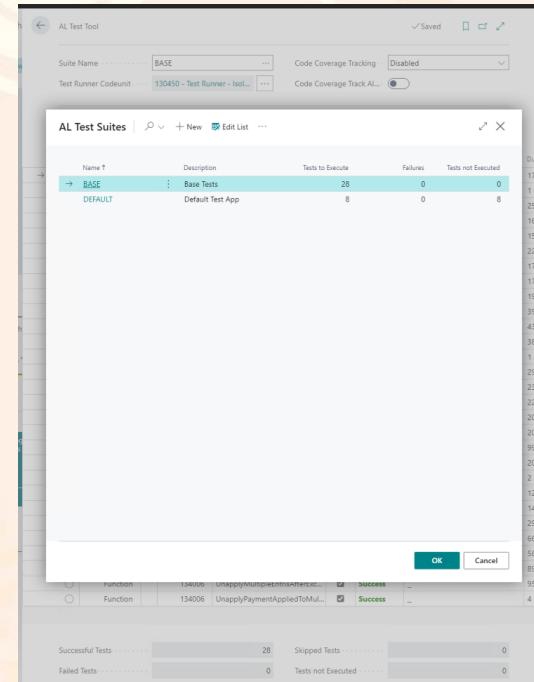
Line Type	Codunit ID	Name	Run	Result	Error Message	Duration
Codeunit	134495	ERM Apply Unapply Customer	✓	Success	—	17 sec
Function	134006	UnapplyInFromCustLedger	✓	Success	—	250 m
Function	134006	UnapplyPaymentFromCustLed...	✓	Success	—	160 m
Function	134006	UnapplyCrMemFromCustLedger	✓	Success	—	153 m
Function	134006	UnapplyRefundFromCustLedger	✓	Success	—	227 m
Function	134006	UnapplyPaymentDtlCustLedger	✓	Success	—	176 m
Function	134006	UnapplyCrMemDtlCustLedger	✓	Success	—	170 m
Function	134006	UnapplyRefundDtlCustLedger	✓	Success	—	196 m
Function	134006	PaymentDtlApplyCM	✓	Success	—	393 m
Function	134006	PaymentDtlApplyInapplyCM	✓	Success	—	410 m
Function	134006	PaymentDtlApplyInapplyCM	✓	Success	—	380 m
Function	134006	AppliedAmountDifferentCure...	✓	Success	—	1 sec
Function	134006	CheckDiscountValueWithPmtD...	✓	Success	—	290 m
Function	134006	CheckDiscountValueWithPmtD...	✓	Success	—	237 m
Function	134006	CheckDiscountValueWithOutD...	✓	Success	—	227 m
Function	134006	CheckDiscountValueWithOutD...	✓	Success	—	206 m
Function	134006	CheckDiscountValueWithPmtD...	✓	Success	—	203 m
Function	134006	ApplyInapplySalesInvoicesWit...	✓	Success	—	990 m
Function	134006	SetAppliesToCrMemDtlCustL...	✓	Success	—	206 m
Function	134006	ApplyInapplyOvervalAndP...	✓	Success	—	2 sec
Function	134006	GetSelectionFilterForCustomer...	✓	Success	—	127 m
Function	134006	GetSelectionFilterForCustomer...	✓	Success	—	147 m
Function	134006	ErrorMessageInapplyWithout...	✓	Success	—	254 m
Function	134006	UnapplyInapplyInLaterAdjust...	✓	Success	—	666 m
Function	134006	UnapplyMultipleInAfterCur...	✓	Success	—	566 m
Function	134006	UnapplyMultipleInAfterCur...	✓	Success	—	897 m
Function	134006	UnapplyPaymentAppliedToMul...	✓	Success	—	950 m

# Running Tests

- Test Suite



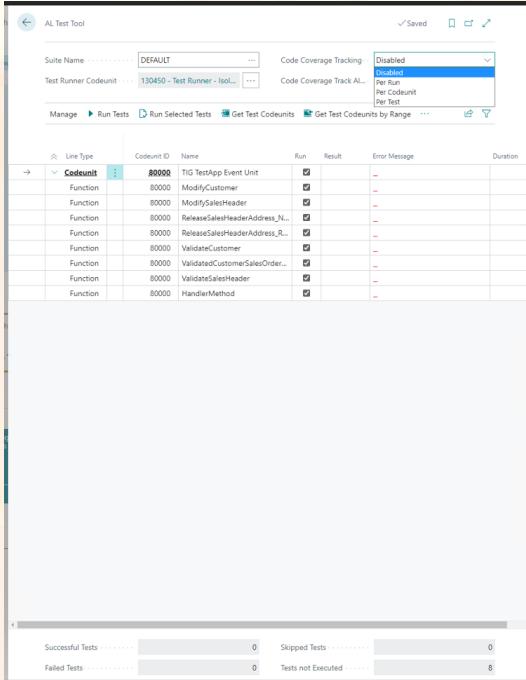
The screenshot shows the AL Test Tool interface. At the top, there's a toolbar with buttons for 'Manage', 'Run Tests', 'Run Selected Tests', 'Get Test Codeunits', and 'Get Test Codeunits by Range'. Below the toolbar is a table with columns: Line Type, Codeunit ID, Name, Run, Result, Error Message, and Duration. The table lists 12 function tests under 'Codeunit' and 1 function test under 'HandlerMethod'. The 'Run' column contains checkboxes, all of which are checked. The 'Result' column shows mostly '-' (not run) or 'Success'. The 'Duration' column shows various times from 1 sec to 4 sec. At the bottom, there are summary statistics: Successful Tests (0), Skipped Tests (0), Failed Tests (0), and Tests not Executed (8).



The screenshot shows the AL Test Tool interface. At the top, there's a toolbar with buttons for 'Manage', 'Run Tests', 'Run Selected Tests', 'Get Test Codeunits', and 'Get Test Codeunits by Range'. Below the toolbar is a table with columns: Name, Description, Tests to Execute, Failures, and Tests not Executed. There are two entries: 'BASE' (Description: 'Base Tests', Tests to Execute: 28, Failures: 0, Tests not Executed: 0) and 'DEFAULT' (Description: 'Default Test App', Tests to Execute: 8, Failures: 0, Tests not Executed: 8). At the bottom, there's a modal dialog showing a table with columns: Function, Codeunit ID, Name, Result, and Duration. It lists 2 functions under 'Function' with 'Result' showing 'Success'. At the bottom of the dialog, there are 'OK' and 'Cancel' buttons.

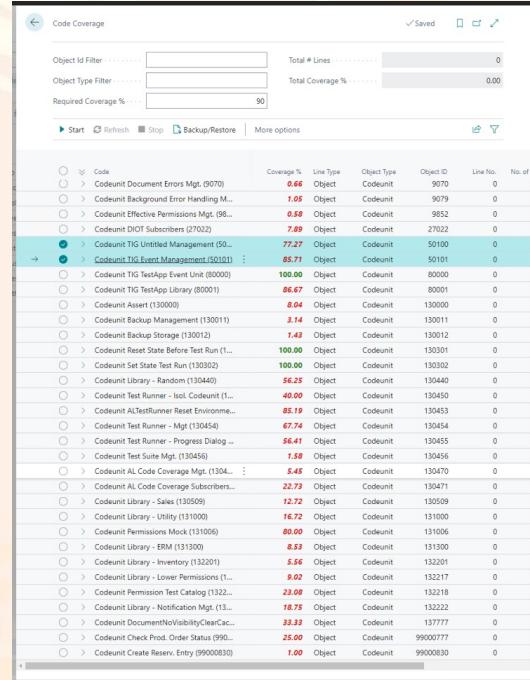
# Running Tests

- Code Coverage



The screenshot shows the AL Test Tool interface with the following details:

- Suite Name:** DEFAULT
- Test Runner Codeunit:** 130450 - Test Runner - Isol.
- Code Coverage Tracking:** Disabled (selected)
- Codeunit ID Filter:** 80000
- Table Headers:** Line Type, Codeunit ID, Name, Run, Result, Error Message, Duration
- Table Data:** A list of test cases for the 'TIG TestApp Event Unit' codeunit, including 'ModifyCustomer', 'ModifySalesHeader', 'ReleaseSalesHeaderAddress\_N...', 'ReleaseSalesHeaderAddress\_R...', 'ValidateCustomer', 'ValidatedCustomerSalesOrder...', and 'ValidateSalesHeader'.
- Metrics at the bottom:**
  - Successful Tests: 0
  - Skipped Tests: 0
  - Failed Tests: 0
  - Tests not Executed: 8

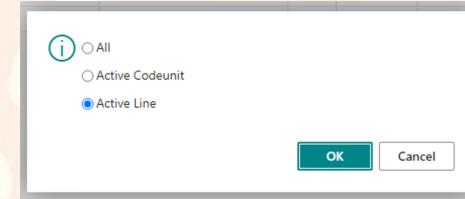
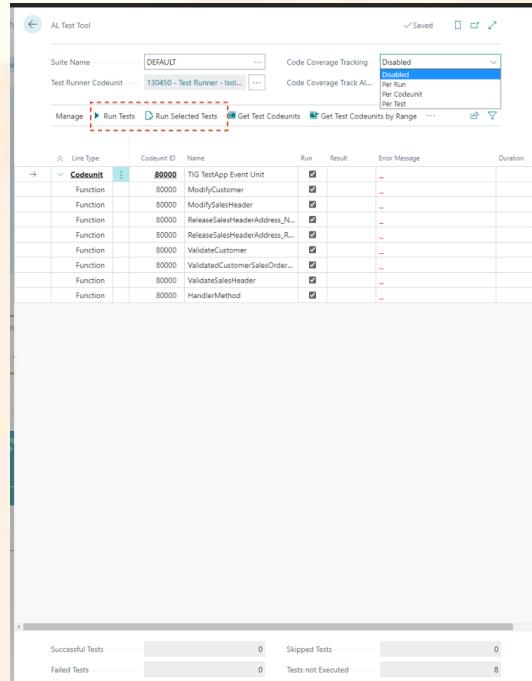
  


The screenshot shows the Code Coverage tool interface with the following details:

- Object Id Filter:** 80000
- Object Type Filter:** Object
- Required Coverage %:** 90
- Table Headers:** Coverage %, Line Type, Object Type, Object ID, Line No., No. of Lines
- Table Data:** A list of objects with their coverage percentages, including 'Codeunit Document Errors Mgt.' (9070), 'Codeunit Background Error Handling M...' (9079), 'Codeunit Effective Permissions Mgt.' (9852), 'Codeunit DIOT Subscribers' (27022), 'Codeunit TIG Untitled Management' (50100), 'Codeunit TIG TestManagement' (50101), 'Codeunit TIG TextApp Event Unit' (80000), 'Codeunit TIG TextApp Library' (80001), 'Codeunit Asset' (130000), 'Codeunit Backup Management' (130011), 'Codeunit Backup Storage' (130012), 'Codeunit Reset State Before Test Run' (130301), 'Codeunit Set State Test Run' (130302), 'Codeunit Library - Random' (130440), 'Codeunit Test Runner - Isol. Codeunit' (130450), 'Codeunit ALTestRunner Reset Environment' (130453), 'Codeunit Test Runner - Mgt' (130454), 'Codeunit Test Runner - Progress Dialog' (130455), 'Codeunit Test Suite Mgt.' (130456), 'Codeunit AI Code Coverage Mgt.' (130457), 'Codeunit AI Code Coverage Subscribers...' (130471), 'Codeunit Library - Sales' (130509), 'Codeunit Library - Utility' (131000), 'Codeunit Permissions Mock' (131006), 'Codeunit Library - ERM' (131300), 'Codeunit Library - Inventory' (132201), 'Codeunit Library - Lower Permissions' (132217), 'Codeunit Permission Test Catalog' (132218), 'Codeunit Library - Notification Mgt.' (132222), 'Codeunit DocumentVisibilityClearCac...' (137777), 'Codeunit Check Prod. Order Status' (9900077), and 'Codeunit Create Reserv. Entry' (99000830).

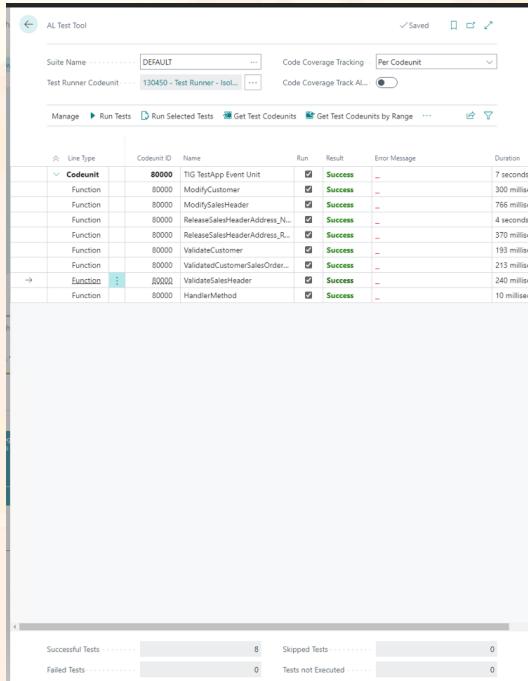
# Running Tests

- Running Tests



# Running Tests

- Completed Tests



The screenshot shows the AL Test Tool interface with the following details:

**Suite Name:** DEFAULT

**Test Runner Codeunit:** 130450 - Test Runner - Isol...

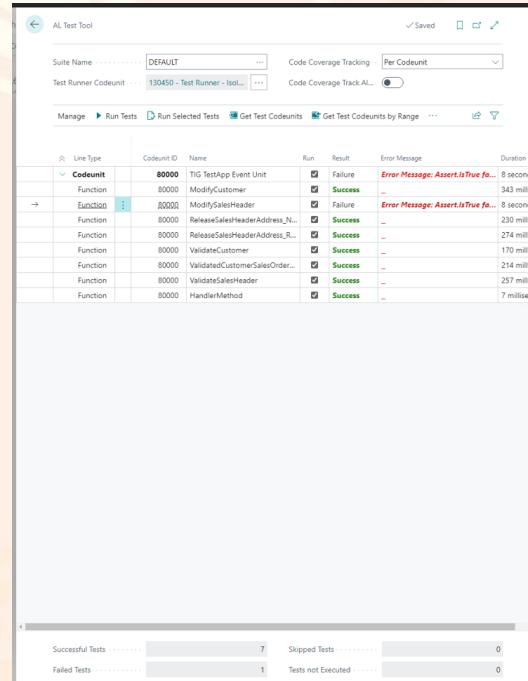
**Code Coverage Tracking:** Per Codeunit

**Test Results:**

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	80000	TIG TestApp Event Unit	<input checked="" type="checkbox"/>	Success	-	7 seconds
	80000	ModifyCustomer	<input checked="" type="checkbox"/>	Success	-	300 millisec
	80000	ModifySalesHeader	<input checked="" type="checkbox"/>	Success	-	766 millisec
	80000	ReleaseSalesHeaderAddress_N...	<input checked="" type="checkbox"/>	Success	-	4 seconds 8
	80000	ReleaseSalesHeaderAddress_R...	<input checked="" type="checkbox"/>	Success	-	370 millisec
	80000	ValidateCustomer	<input checked="" type="checkbox"/>	Success	-	193 millisec
	80000	ValidateCustomerSaleOrder...	<input checked="" type="checkbox"/>	Success	-	213 millisec
	80000	ValidateSalesHeader	<input checked="" type="checkbox"/>	Success	-	240 millisec
Function	80000	HandleMethod	<input checked="" type="checkbox"/>	Success	-	10 millisec

**Summary:**

- Successful Tests: 8
- Skipped Tests: 0
- Failed Tests: 0
- Tests not Executed: 0



The screenshot shows the AL Test Tool interface with the following details:

**Suite Name:** DEFAULT

**Test Runner Codeunit:** 130450 - Test Runner - Isol...

**Code Coverage Tracking:** Per Codeunit

**Test Results:**

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	80000	TIG TestApp Event Unit	<input checked="" type="checkbox"/>	Failure	Error Message: Assert.IsTrue fo...	8 seconds 2
	80000	ModifyCustomer	<input checked="" type="checkbox"/>	Success	-	343 millisec
	80000	ModifySalesHeader	<input checked="" type="checkbox"/>	Failure	Error Message: Assert.IsTrue fo...	8 seconds 2
	80000	ReleaseSalesHeaderAddress_N...	<input checked="" type="checkbox"/>	Success	-	230 millisec
	80000	ReleaseSalesHeaderAddress_R...	<input checked="" type="checkbox"/>	Success	-	274 millisec
	80000	ValidateCustomer	<input checked="" type="checkbox"/>	Success	-	170 millisec
	80000	ValidateCustomerSaleOrder...	<input checked="" type="checkbox"/>	Success	-	214 millisec
	80000	ValidateSalesHeader	<input checked="" type="checkbox"/>	Success	-	257 millisec
	80000	HandleMethod	<input checked="" type="checkbox"/>	Success	-	7 millisec

**Summary:**

- Successful Tests: 7
- Skipped Tests: 0
- Failed Tests: 1
- Tests not Executed: 0

# COMMUNITY SUMMIT

# Running Tests

- `BCCContainerHelper`: `Run-TestsInBCCContainer`

```
Run-TestsInBCCContainer.ps1 > Run-TestsInBCCContainer.ps1 > ...  
Supporting-Files > Powershell > Run-TestsInBCCContainer.ps1 > ...  
1 #####  
2 #####  
3 $containerName = 'BC22-Test'  
4 $userName = 'admin'  
5 $password = 'password'  
6 $testSuite = 'DEFAULT'  
7 $xunitResultsFile = "C:\ProgramData\BcContainerHelper\Extensions\$($containerName)\results.xml"  
8 #####  
9 #####  
10 #####  
11 $scriptTitle = 'Run-TestsInBCCContainer'  
12 #####  
13 $startTime = [DateTime]::Now  
14 Write-Host -ForegroundColor Green "$($startTime) : $($scriptTitle) start"  
15 #####  
16 $securePassword = ConvertTo-SecureString $password -AsPlainText -Force  
17 $Credential = New-Object System.Management.Automation.PSCredential ($userName,$securePassword)  
18 Run-TestsInBCCContainer `<br>...`  
19     -containerName $containerName `<br>...`  
20     -credential $credential `<br>...`  
21     -testSuite $testSuite `<br>...`  
22     -detailed `<br>...`  
23     -AppendToXUnitResultFile:($true) `<br>...`  
24     -XUnitResultFileName $xunitResultsFile  
25 #####  
26 #####  
27 $stopTime = [DateTime]::Now  
28 $timeSpend = ($stopTime - $startTime).TotalSeconds  
29 #####  
30 Write-Host -ForegroundColor Green "$($stopTime) : $($scriptTitle) took $timeSpend" > Run-TestsInBCCContainer.log  
04/10/2023 16:35:02 : Run-TestsInBCCContainer start  
Connecting to http://localhost:80/BC/cs?tenant=default  
Codeunit 80000 TIG TestApp Event Unit Success (7.79 seconds)  
    Testfunction ModifyCustomer Success (1.94 seconds)  
    Testfunction ModifySalesHeader Success (1.397 seconds)  
    Testfunction ReleaseSalesHeaderAddress_NotRequired Success (1.543 seconds)  
    Testfunction ReleaseSalesHeaderAddress_Required Success (0.333 seconds)  
    Testfunction ValidateCustomer Success (0.22 seconds)  
    Testfunction ValidatedCustomerSalesOrderCreated Success (0.263 seconds)  
    Testfunction ValidateSalesHeader Success (0.196 seconds)  
    Testfunction HandlerMethod Success (0.007 seconds)  
04/10/2023 16:35:22 : Run-TestsInBCCContainer took 19.1651415 seconds
```

# Running Tests

- BCContainerHelper: Run-TestsInBCContainer

```

Get-TestsFromBCContainer-pipeline.ps1
Supporting-Files > PowerShell > Get-TestsFromBCContainer-pipeline.ps1 > ...
1 #####
2 $containerName = 'BC22-Test'
3 $userName = 'admin'
4 $password = 'password'
5 $testSuite = 'DEFAULT'
6 $xunitResultsFile = "C:\ProgramData\BcContainerHelper\Extensions\$($containerName)\results.xml"
7 #####
8 #####
9 #####
10 $scriptTitle = 'Get-TestsFromBCContainer Pipeline'
11 #####
12 $startTime = [DateTime]::Now
13 Write-Host -ForegroundColor Green "$($startTime) : $($scriptTitle) start"
14 #####
15 $securePassword = ConvertTo-SecureString $password -AsPlainText -Force
16 $credential = New-Object System.Management.Automation.PSCredential ($userName,$securePassword)
17 #####
18 #####
19 $first = $true
20 $tests = Get-TestsFromBCContainer `-
21     -containerName $containerName `-
22     -Credential $credential `-
23     -testSuite $testSuite `-
24     -ignoreGroups
25 #####
26 $tests | ForEach-Object {
27     Run-TestsInBCContainer `-
28         -containerName $containerName `-
29         -credential $credential `-
30         -xUnitResultFileName $xunitResultsFile `-
31         -AppendToXUnitResultFile:(!$first) `-
32         -testCodeunit $_.Id
33     $first = $false
34 }
35 #####
36 $stopTime = [DateTime]::Now
37 $timeSpend = ($stopTime - $startTime).TotalSeconds
38 #####
39 Write-Host -ForegroundColor Green "$($stopTime) : $($scriptTitle) took $($timeSpend) seconds"

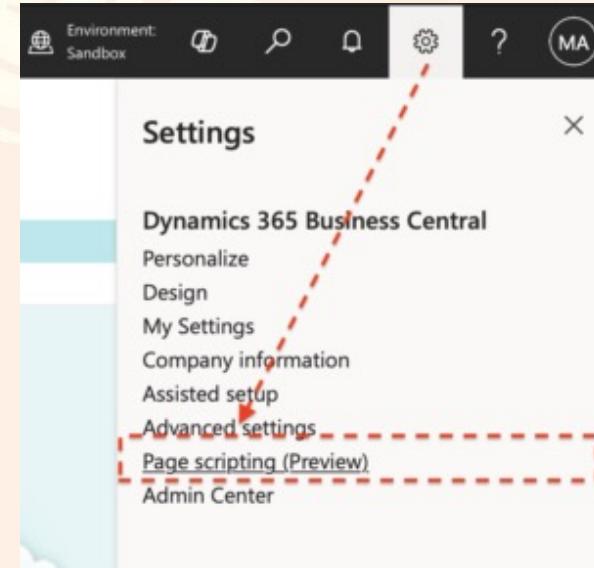
```

04/10/2023 16:41:53 : Get-TestsFromBCContainer Pipeline start  
 Connecting to http://localhost:80/BC/cs?tenant=default  
 Codeunit 80000 TIG TestApp Event Unit Success (9.447 seconds)  
 04/10/2023 16:42:25 : Get-TestsFromBCContainer Pipeline took 32.6839124 seconds  
 PS C:\Windows\system32>

<?xml version="1.0" encoding="UTF-8"?>
<assemblies>
 <assembly name="80000 TIG TestApp Event Unit" test-framework="PS Test Runner" run-date="2023-04-10" run-time="17:49:08" total="8" passed="8" failed="0" skipped="0" time="7.777">
 <collection name="TIG TestApp Event Unit" total="8" passed="8" failed="0" skipped="0" time="7.777" Skipped="0">
 <test name="TIG TestApp Event Unit:ModifyCustomer" method="ModifyCustomer" time="2.627" result="Pass" />
 <test name="TIG TestApp Event Unit:ModifySalesHeader" method="ModifySalesHeader" time="1.983" result="Pass" />
 <test name="TIG TestApp Event Unit:ReleaseSalesHeaderAddress\_NotRequired" method="ReleaseSalesHeaderAddress\_NotRequired" time="2.01" result="Pass" />
 <test name="TIG TestApp Event Unit:ReleaseSalesHeaderAddress\_Required" method="ReleaseSalesHeaderAddress\_Required" time="0.4" result="Pass" />
 <test name="TIG TestApp Event Unit:ValidateCustomer" method="ValidateCustomer" time="0.256" result="Pass" />
 <test name="TIG TestApp Event Unit:ValidatedCustomerSalesOrderCreated" method="ValidatedCustomerSalesOrderCreated" time="0.24" result="Pass" />
 <test name="TIG TestApp Event Unit:ValidateSalesHeader" method="ValidateSalesHeader" time="0.247" result="Pass" />
 <test name="TIG TestApp Event Unit:HandlerMethod" method="HandlerMethod" time="0.014" result="Pass" />
 </collection>
 </assembly>
</assemblies>

# In-Client Page Scripting

- Record and Reply tests directly within Business Central
- Internal Clipboard
- Expressions in Steps
- Conditional Steps
- Validate Outcome
- Handle Optional Pages





# Summary

- Basic Understanding of Automated Testing in Business Central
- Setup
- Writing Tests
- Running Tests

# Questions



# What's Next

- Explore Resources
- Review and Practice
- Setup your Environment
- Start Small
- Iterate and Improve



# References



@ 2024 Dynamic Communities

The largest independent innovation, education,  
and training event for Microsoft Biz Apps



# Thank you for Attending!



@ 2024 Dynamic Communities

@ 2024 Dynamic Communities



# COMMUNITY SUMMIT

## BUSINESS CENTRAL



© 2024 Dynamic Communities. All rights reserved.