

The background of the slide features a night-time photograph of a city skyline, likely San Diego, with its lights reflecting on the water. Overlaid on the right side of the image is a large, stylized logo consisting of concentric, multi-colored arcs (green, yellow, blue, red) and dots, resembling a network or orbital diagram.

Introduction to Test Automation in Business Central

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

- Edsger W. Dijkstra

About Me

Brad Prendergast
Tigunia, LLC

✉ brad@tigunia.com
✕ @dvlprlife
in [linkedin.com/in/brad-prendergast/](https://www.linkedin.com/in/brad-prendergast/)



April 14-17, 2024 Sheraton San Diego Hotel & Marina, San Diego, CA

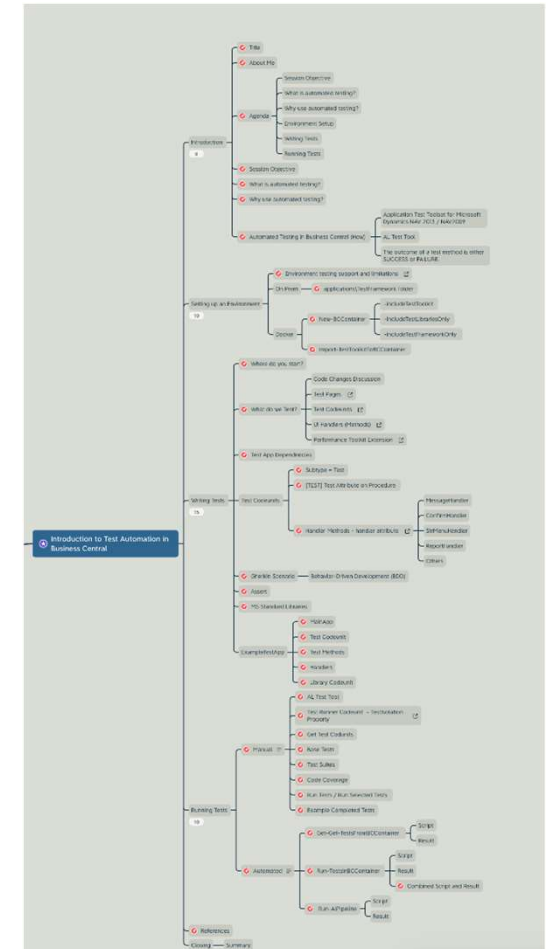
Agenda

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



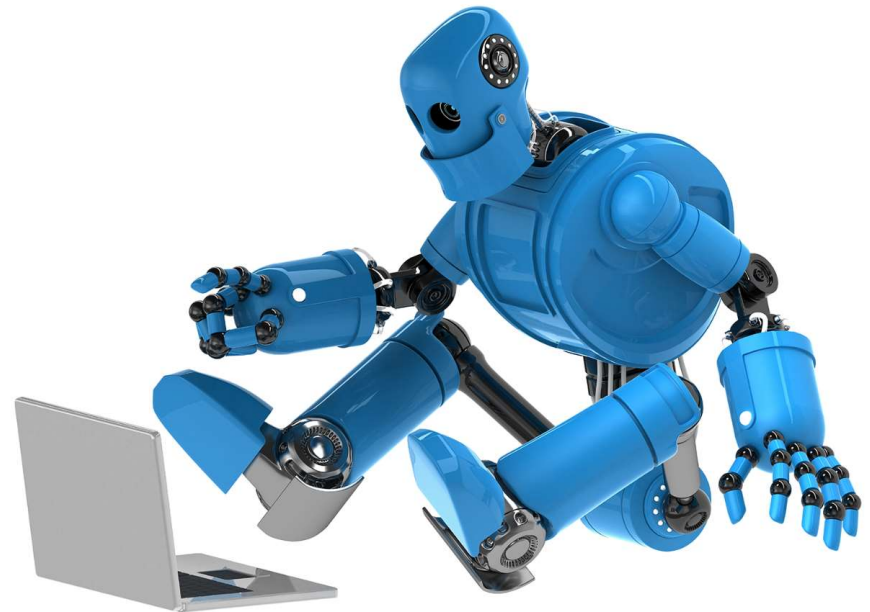
Session Objective

- 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.

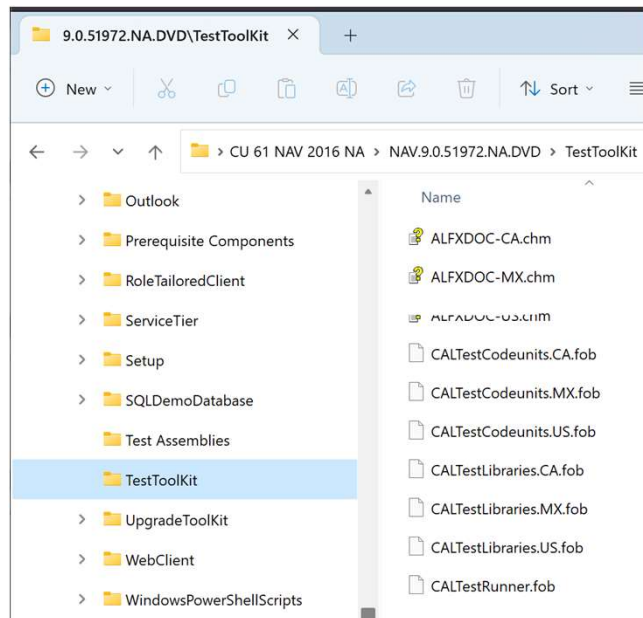


Why use Automated Testing

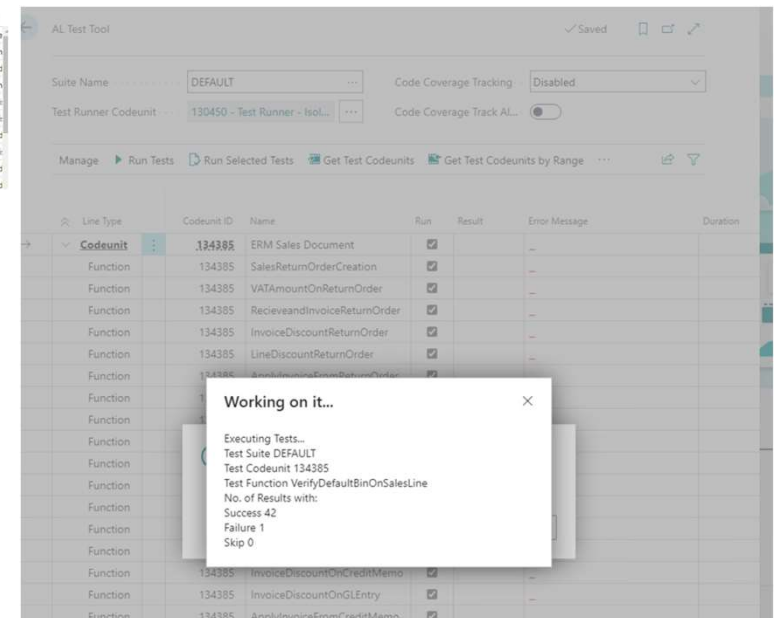
- Saves Time
- Improves Quality*
- Repeatable and Reliable
- Reduces Risk and Cost
- Less Disruption / Maximize Error Detection
- Latest Releases



Automated Testing in Business Central



Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	134385	ERM Sales Document	✓	Failure	Error Message: The No. Series ...	3 minute
Function	134385	SalesReturnOrderCreation	✓	Success	---	14 second
Function	134385	VATAmountOnReturnOrder	✓	Success	---	1 second
Function	134385	RecieveandInvoiceReturnOrder	✓	Success	---	10 second
Function	134385	InvoiceDiscountReturnOrder	✓	Success	---	930 milli
Function	134385	LineDiscountReturnOrder	✓	Success	---	207 milli
Function	134385	ApplyInvoiceFromReturnOrder	✓	Success	---	3 second
Function	134385	LocationforReturnOrder	✓	Failure	Error Message: The Inventory ...	774 milli
Function	134385	CurrencyOnReturnOrder	✓	Success	---	1 second
Function	134385	CopyDocumentFromReturnOr...	✓	Success	---	1 second



Setting up an Environment





- Environment testing support and limitations

Business Central solution	Environment	Testing allowed	More details
Online	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 Enable Test Automation (TestAutomationEnabled) on the Business Central Server instance. For more information, see Configuring Business Central Server - General Settings .
	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 Running a Container-Based Development Environment or Running Tests In Containers .

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	
 performancetoolkit	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) : $($scriptTitle) start"

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

New-BCContainer `
  -accept_eula `
  -accept_outdated `
  -alwaysPull `
  -artifactUrl $artifactUrl `
  -auth NavUserPassword `
  -containerName $containername `
  -Credential $credential `
  -includeTestToolkit `
  -isolation hyperv `
  -updateHosts `
  -memoryLimit 8G

$stopTime = [DateTime]::Now
$timeSpend = ($stopTime - $startTime).TotalSeconds

Write-Host -ForegroundColor Green "$($stopTime) : $($scriptTitle) took $($timespend) seconds"
```

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

Setting up an Environment

- BCContainerHelper: ImportTestToolkitToBCContainer

```
$scriptTitle = 'Import-TestToolkitToBCContainer'

$startTime = [DateTime]::Now
Write-Host -ForegroundColor Green "$($startTime) : $($scriptTitle) start"

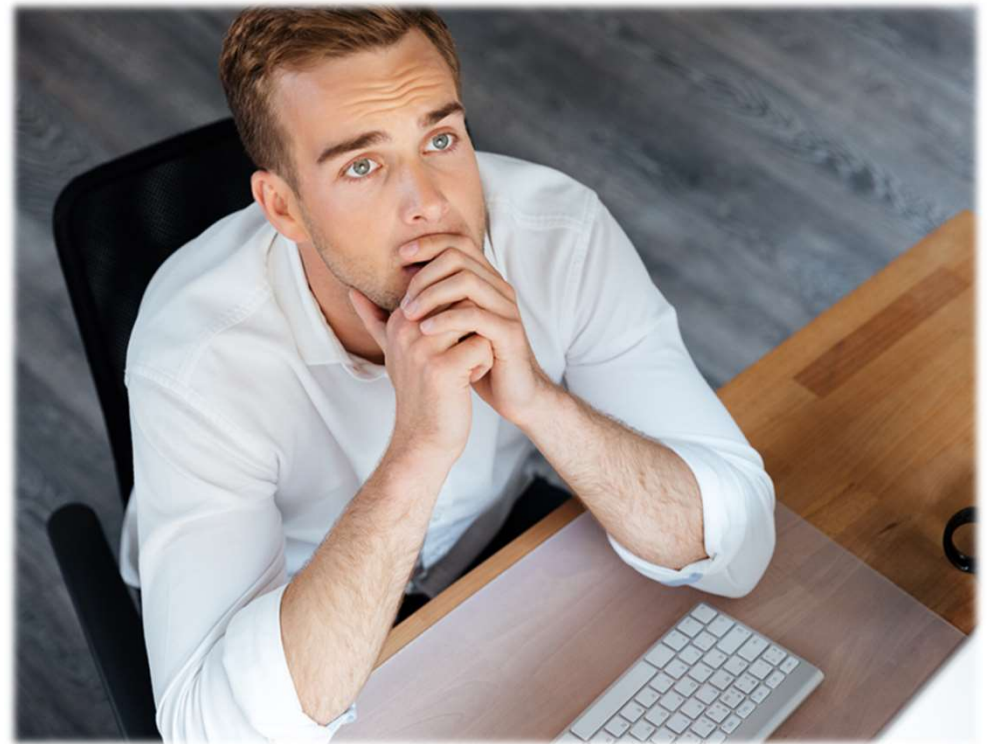
Import-TestToolkitToBCContainer -containerName $containerName ← -----

$stopTime = [DateTime]::Now
$timeSpend = ($stopTime - $startTime).TotalSeconds

Write-Host -ForegroundColor Green "$($stopTime) : $($scriptTitle) took $($timeSpend) seconds"
```

Writing Tests

- Where do you start?
 - Familiarize
 - 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

```
0 references
codeunit 80000 "TIG TestApp Event Unit"
{
    Subtype = Test;

    var
        Assert: Codeunit Assert;
        LibraryERM: Codeunit "Library - ERM";
        LibraryRandom: Codeunit "Library - Random";
        LibrarySales: Codeunit "Library - Sales";
        TestAppLibrary: Codeunit "TIG TestApp Library";
        FieldValueErr: label '%1 must be %2.', Comment = '%1 Field Caption %2 Field Value';

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

        LibrarySales.CreateCustomerWithAddress(Customer);

        TestAppLibrary.ValidateCustomer(Customer);

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

        // [THEN] Customer is not validated
        Assert.IsFalse(Customer."TIG Address Validated", StrSubstNo(FieldValueErr, Customer.FieldCaption("
    end;
```

Writing Tests

- Test Codeunits: Test Attribute

```
0 references
codeunit 80000 "TIG TestApp Event Unit"
{
    Subtype = Test;

    var
        Assert: Codeunit Assert;
        LibraryERM: Codeunit "Library - ERM";
        LibraryRandom: Codeunit "Library - Random";
        LibrarySales: Codeunit "Library - Sales";
        TestAppLibrary: Codeunit "TIG TestApp Library";
        FieldValueErr: label '%1 must be %2.', Comment = '%1 Field Caption %2 Field Value';

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

        LibrarySales.CreateCustomerWithAddress(Customer);

        TestAppLibrary.ValidateCustomer(Customer);

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

        // [THEN] Customer is not validated
        Assert.IsFalse(Customer."TIG Address Validated", StrSubstNo(FieldValueErr, Customer.FieldCaption("
    end;
```

Writing Tests

- Test Codeunits: Handler Methods

Method type	Purpose	Signature
MessageHandler	Handles Message statements.	<Function name>(<Message>: Text[1024])Error! Filename not specified.
ConfirmHandler	Handles Confirm statements.	<Function name>(<Question>: Text[1024]; var <Reply>: Boolean)
StrMenuHandler	Handles StrMenu statements.	<Function name>(<Options>: Text[1024]; var <Choice>: Integer; <Instruction>: Text[1024])
ReportHandler	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>)

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

Writing Tests

- Gherkin Scenario

```
1  
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

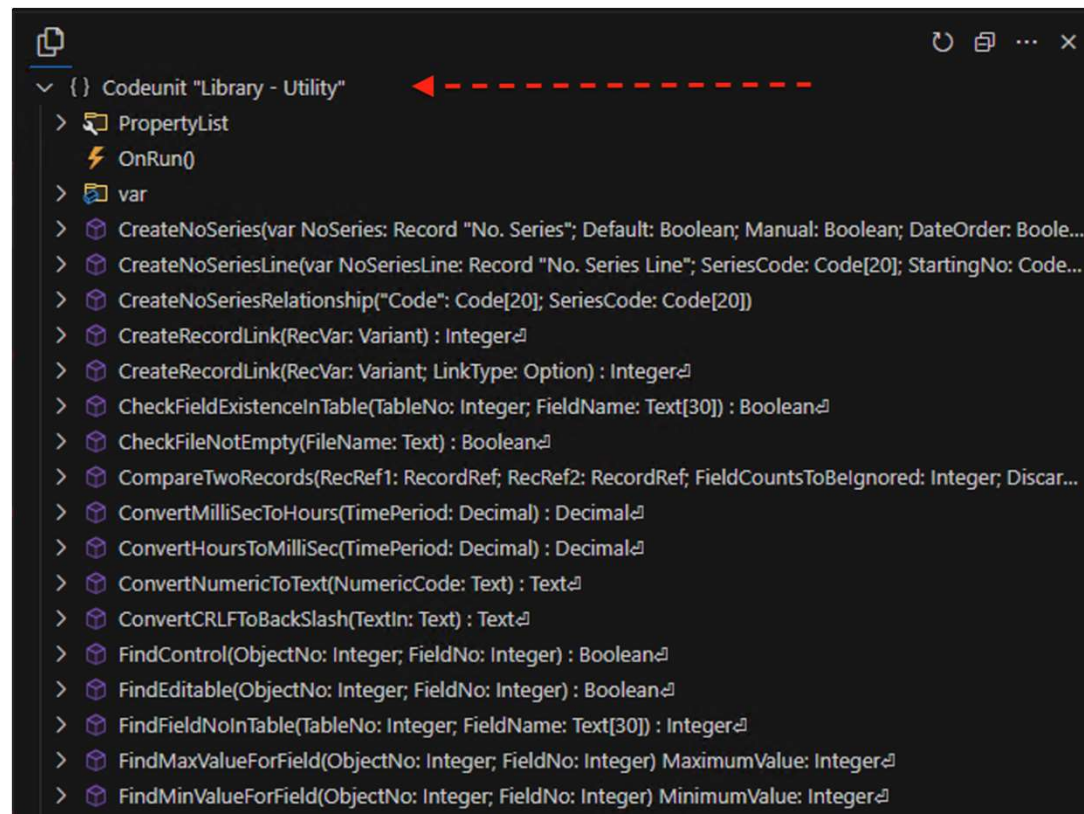
- Assert Library

```
/// <summary>
/// This module provides functions for easy verification of expected values and error handling in test code.
/// </summary>
codeunit 130002 "Library Assert"
{
    namespace System.TestLibraries.Utilities
    using System.Reflection
    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)
    > AreEqualDateTime(Expected: DateTime; Actual: DateTime; Delta: Duration; 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)
    > RecordIsEmpty(RecVariant: Variant)
    > RecordIsNotEmpty(RecVariant: Variant)
    > TableIsEmpty(TableNo: Integer)
    > TableIsNotEmpty(TableNo: Integer)
    > RecRefIsEmpty(var RecordRef: RecordRef)
    > RecRefIsNotEmpty(var RecordRef: RecordRef)
```

Writing Tests

Microsoft_Tests-TestLibraries

CU	130440	Library - Random
CU	130509	Library - Sales
CU	130512	Library - Purchase
CU	131000	Library - Utility



The screenshot shows a code editor window with a dark theme. The title bar indicates the file is 'Codeunit "Library - Utility"'. The code is written in a language that appears to be AL (Application Language) for Microsoft Dynamics 365. The code starts with a 'PropertyList' section, followed by an 'OnRun()' event trigger, and then a 'var' section. Below this, there is a long list of methods, each preceded by a cube icon. The methods include 'CreateNoSeries', 'CreateNoSeriesLine', 'CreateNoSeriesRelationship', 'CreateRecordLink', 'CheckFieldExistenceInTable', 'CheckFileNotEmpty', 'CompareTwoRecords', 'ConvertMilliSecToHours', 'ConvertHoursToMilliSec', 'ConvertNumericToText', 'ConvertCRLFToBackSlash', 'FindControl', 'FindEditable', 'FindFieldNoInTable', 'FindMaxValueForField', and 'FindMinValueForField'. Each method signature includes parameters and return types, some of which are truncated in the screenshot.

Writing Tests

- Example App: MainApp

Sales Order

S-ORD101001 · Adatum Corporation

Start validating data in documents and journals while you work. Messages are shown here. Enable this feature | Don't show this message again

Home Prepare Print/Send Request Approval Order Report More options

Post... Create Warehouse Shipment Archive Document

Release Create Inventory Put-away/Pickup

General > 4/2/2024 4/2/2024 4/2/2024 5/2/2024 Open

Lines >

Invoice Details > 1M(8D) Yes ATLANTA, GA

Shipping and Billing Show more

Ship-to Default (Sell-to Address) Location Code

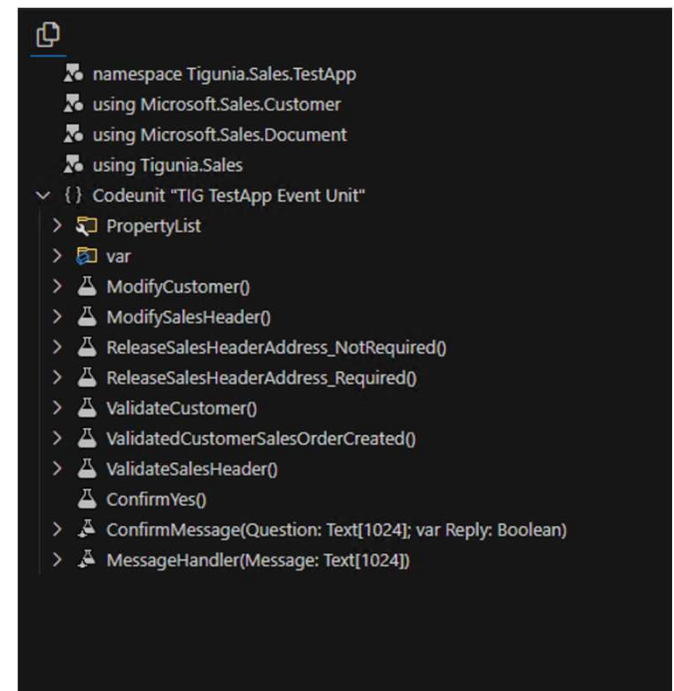
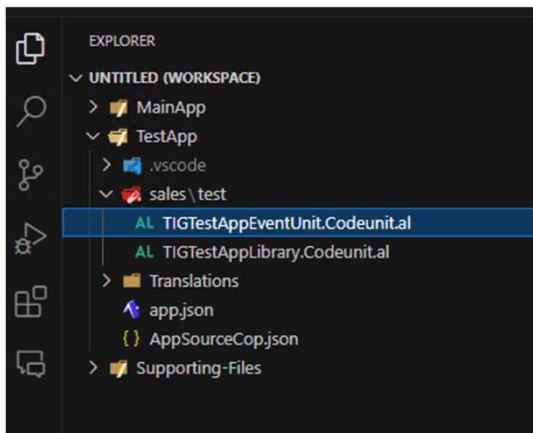
Contact Robert Townes Shipment Date 4/12/2024

Bill-to Default (Customer) Address Validated

Prepayment > 0 5/2/2024

Writing Tests

- Example App: TestApp



Writing Tests

- Example App: TestApp Methods

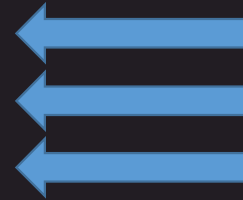
```
-----[Test]
0 references
-----procedure ModifyCustomer()
-----var
-----Customer: Record Customer;
-----begin
-----// [SCENARIO] A validated customer is modified
-----// [GIVEN] Validation is enabled
-----// [GIVEN] Validated customer exists
-----TestAppLibrary.EnableValidation(true, false, false);

-----LibrarySales.CreateCustomerWithAddress(Customer);

-----TestAppLibrary.ValidateCustomer(Customer);

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

-----// [THEN] Customer is not validated
-----Assert.IsFalse(Customer."TIG Address Validated", StrSubstNo(FieldValueErr, Customer.FieldCaption("TIG Address Validated"), false));
-----end;
```



Writing Tests

- Example App: TestApp Methods

```
....[Test]
  0 references
  ....procedure ModifySalesHeader()
  ....var
  ....Customer: Record Customer;
  ....SalesHeader: Record "Sales Header";
  ....CustomerNo: Code[20];
  ....begin
  ....// [SCENARIO] A validated sales header is modified
  ....// [GIVEN] Validation is enabled
  ....// [GIVEN] Validated sales header exists
  ....TestAppLibrary.EnableValidation(true, false, false);

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

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

  ....TestAppLibrary.ValidateSalesHeader(SalesHeader);

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

  ....// [THEN] Sales header is not validated
  ....Assert.IsFalse(SalesHeader."TIG Address Validated", StrSubstNo(FieldValueErr, SalesHeader.FieldCaption("TIG Address Validated"), false));
  ....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, false);

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

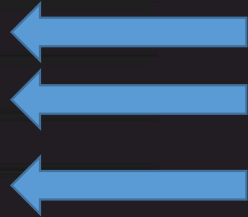
-- GLAccNo := LibraryERM.CreateGLAccountWithSalesSetup();

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

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

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

-- // [THEN] Sales Header is not released
-- Assert.IsFalse(SalesHeader.Status = "Sales Document Status"::Released, StrSubstNo(FieldValueErr, SalesHeader.FieldCaption(Status), "Sales Document Status"::Open));
-- end;
```



Writing Tests

- Example App: TestApp Handlers

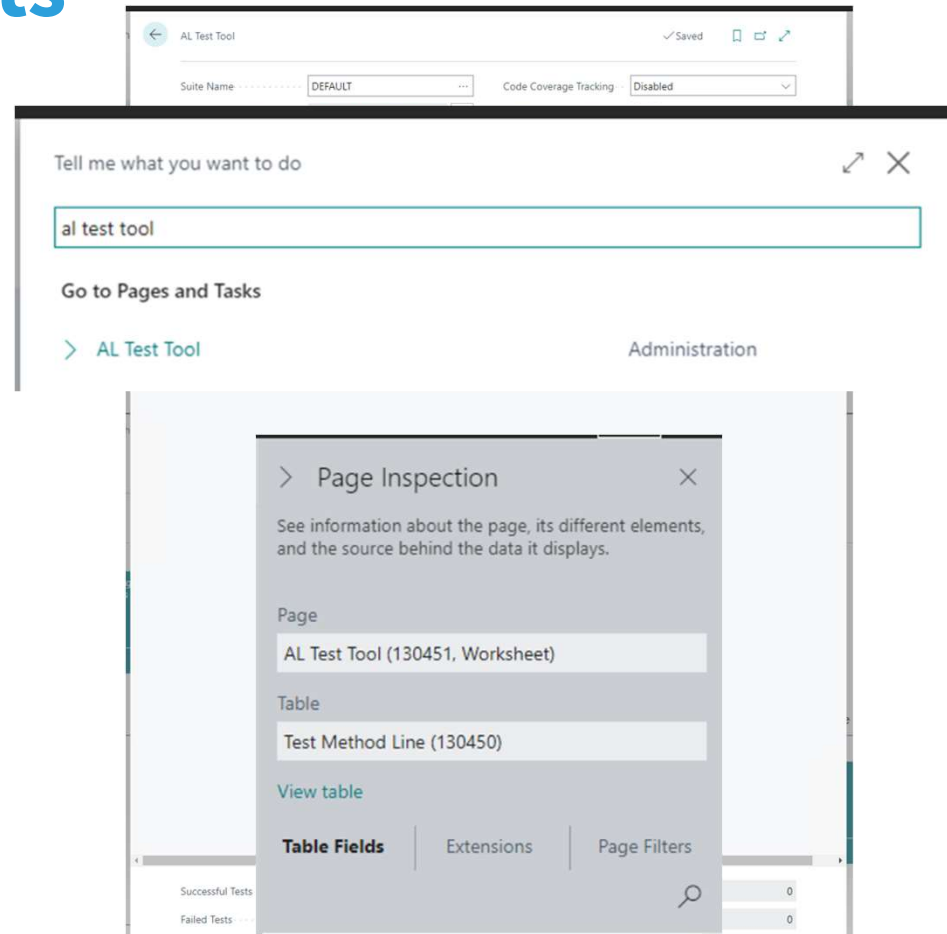
```
...[Test]
...[HandlerFunctions('MessageHandler,ConfirmMessage')]
...0 references
...procedure ConfirmYes()
...begin
...    // [SCENARIO] Demonstrate handler methods
...    // [GIVEN] Handlers assigned
...    // [WHEN] Test confirm and message
...    TestAppLibrary.AskConfirmMessage();
...    // [THEN]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

...[ConfirmHandler]
...0 references
...procedure ConfirmMessage(Question: Text[1024]; var Reply: Boolean);
...var
...    SalesManagement: Codeunit "TIG Sales Management";
...begin
...    Reply := (Question = SalesManagement.GetConfirmMsg());
...end;

...[MessageHandler]
...0 references
...procedure MessageHandler(Message: Text[1024])
...var
...    SalesManagement: Codeunit "TIG Sales Management";
...begin
...    Assert.IsTrue(StrPos(Message, SalesManagement.GetYesMsg()) > 0, Message);
...end;
```


Running Tests

- AL Test Tool



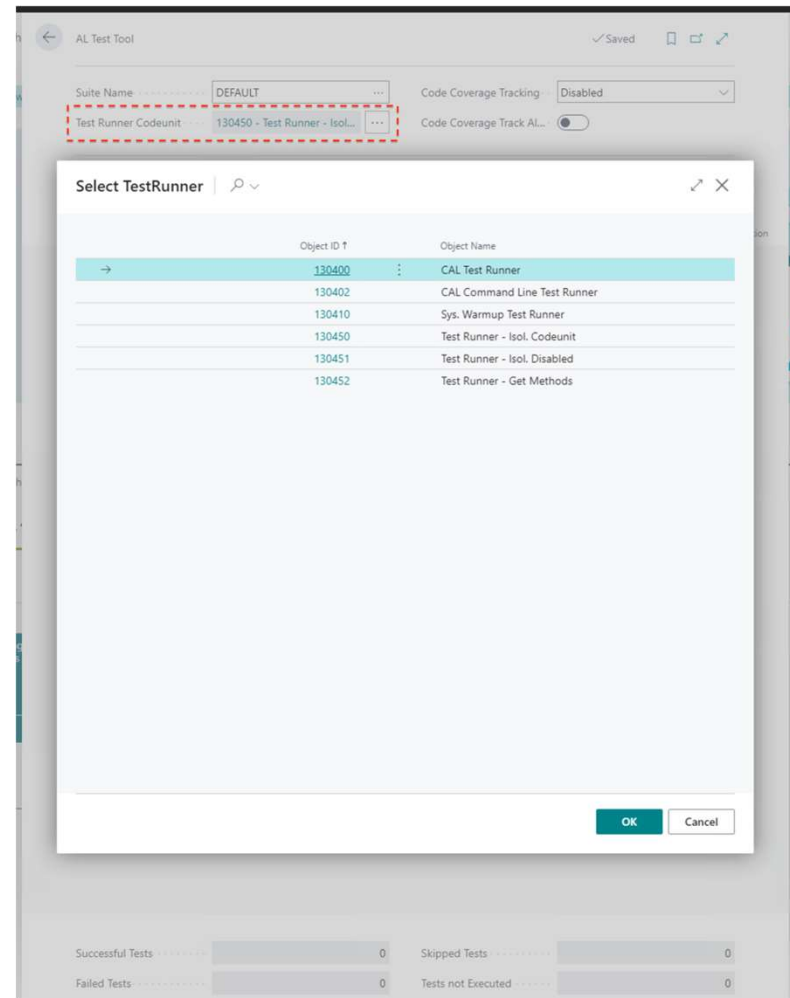
Running Tests

- Test Runner Codeunit

TestIsolation Property

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

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

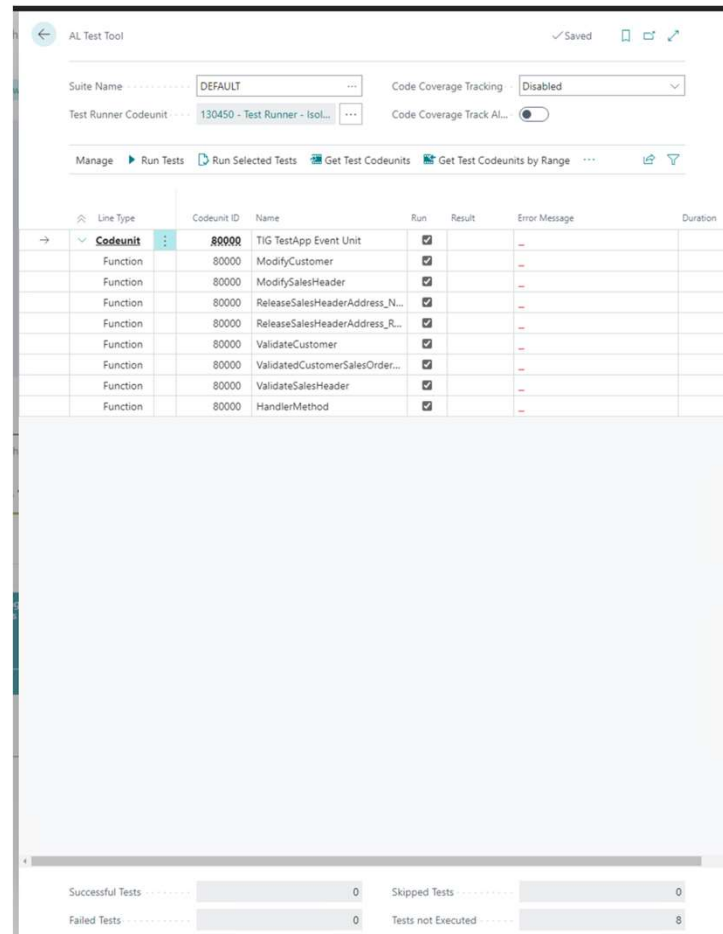


April 14-17, 2024

Sheraton San Diego Hotel & Marina, San Diego, CA

Running Tests

- Test Codeunits



The screenshot shows the AL Test Tool interface. At the top, there are fields for Suite Name (DEFAULT) and Test Runner Codeunit (130450 - Test Runner - Isol...). Below these are buttons for Manage, Run Tests, Run Selected Tests, Get Test Codeunits, and Get Test Codeunits by Range. The main table lists test codeunits with columns for Line Type, Codeunit ID, Name, Run, Result, Error Message, and Duration. The table shows a Codeunit named 'TIG TestApp Event Unit' with ID 80000, and several Functions with ID 80000. The Run column shows checkboxes, and the Result column shows status icons. At the bottom, there is a summary section with counts for Successful Tests, Failed Tests, Skipped Tests, and Tests not Executed.

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	80000	TIG TestApp Event Unit	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ModifyCustomer	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ModifySalesHeader	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ReleaseSalesHeaderAddress_N...	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ReleaseSalesHeaderAddress_R...	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ValidateCustomer	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ValidatedCustomerSalesOrder...	<input checked="" type="checkbox"/>	✓	---	
Function	80000	ValidateSalesHeader	<input checked="" type="checkbox"/>	✓	---	
Function	80000	HandlerMethod	<input checked="" type="checkbox"/>	✓	---	

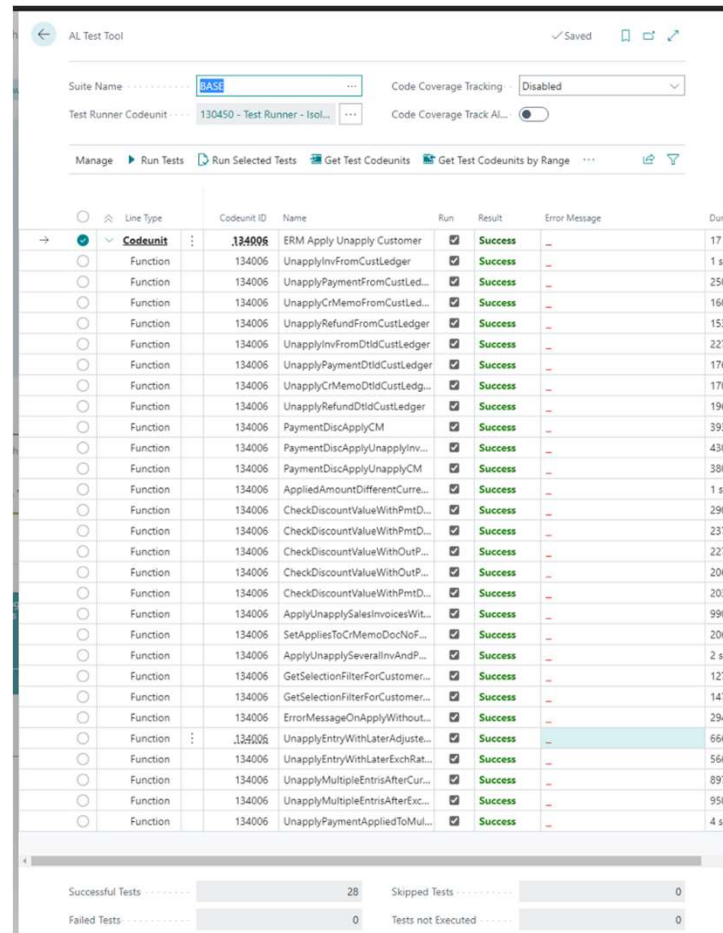
Successful Tests: 0 Skipped Tests: 0
Failed Tests: 0 Tests not Executed: 8

April 14-17, 2024

Sheraton San Diego Hotel & Marina, San Diego, CA

Running Tests

- Base Tests



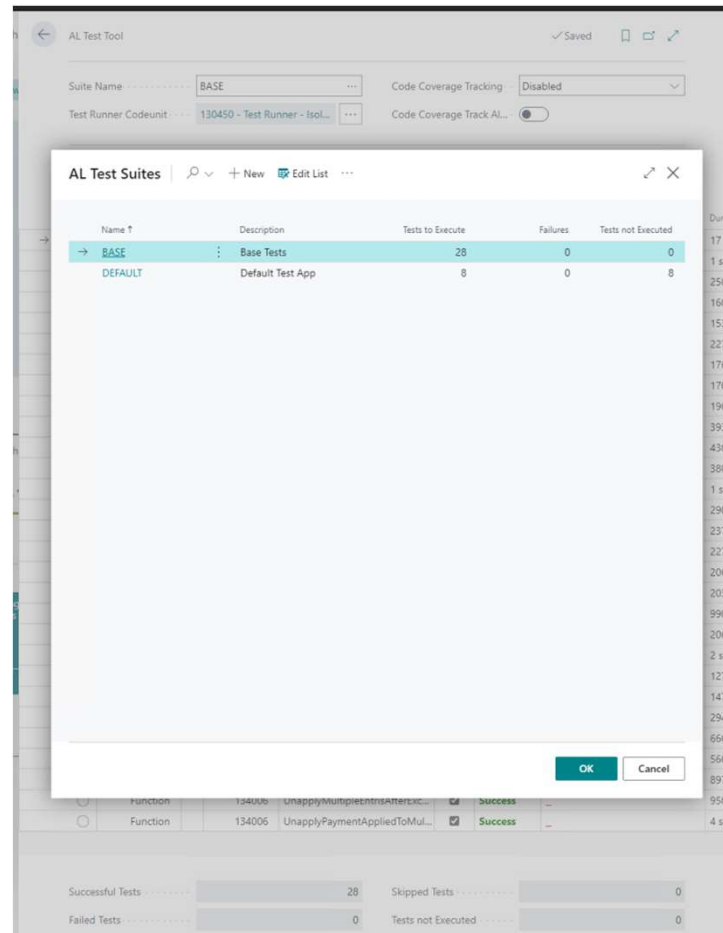
The screenshot shows the AL Test Tool interface. At the top, there are fields for Suite Name (BASE) and Test Runner Codeunit (130450 - Test Runner - Isol...). Below these are buttons for Manage, Run Tests, Run Selected Tests, Get Test Codeunits, and Get Test Codeunits by Range. The main table lists test codeunits with columns for Line Type, Codeunit ID, Name, Run, Result, Error Message, and Duration. All tests shown are successful. At the bottom, there is a summary section showing Successful Tests (28), Failed Tests (0), Skipped Tests (0), and Tests not Executed (0).

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	134006	ERM Apply Unapply Customer	✓	Success		17 sec
Function	134006	UnapplyInvFromCustLedger	✓	Success		1 sec
Function	134006	UnapplyPaymentFromCustLed...	✓	Success		250 m
Function	134006	UnapplyCrMemoFromCustLed...	✓	Success		160 m
Function	134006	UnapplyRefundFromCustLedger	✓	Success		153 m
Function	134006	UnapplyInvFromDtldCustLedger	✓	Success		227 m
Function	134006	UnapplyPaymentDtldCustLedger	✓	Success		176 m
Function	134006	UnapplyCrMemoDtldCustLedg...	✓	Success		170 m
Function	134006	UnapplyRefundDtldCustLedger	✓	Success		196 m
Function	134006	PaymentDiscApplyCM	✓	Success		393 m
Function	134006	PaymentDiscApplyUnapplyInv...	✓	Success		430 m
Function	134006	PaymentDiscApplyUnapplyCM	✓	Success		380 m
Function	134006	AppliedAmountDifferentCurre...	✓	Success		1 sec
Function	134006	CheckDiscountValueWithPmtD...	✓	Success		290 m
Function	134006	CheckDiscountValueWithPmtD...	✓	Success		237 m
Function	134006	CheckDiscountValueWithOutP...	✓	Success		227 m
Function	134006	CheckDiscountValueWithOutP...	✓	Success		206 m
Function	134006	CheckDiscountValueWithPmtD...	✓	Success		203 m
Function	134006	ApplyUnapplySalesInvoicesWit...	✓	Success		990 m
Function	134006	SetAppliesToCrMemoDocNoF...	✓	Success		206 m
Function	134006	ApplyUnapplySeveralInvAndP...	✓	Success		2 sec
Function	134006	GetSelectionFilterForCustomer...	✓	Success		127 m
Function	134006	GetSelectionFilterForCustomer...	✓	Success		147 m
Function	134006	ErrorMessageOnApplyWithout...	✓	Success		294 m
Function	134006	UnapplyEntryWithLaterAdjuste...	✓	Success		666 m
Function	134006	UnapplyEntryWithLaterExchRat...	✓	Success		566 m
Function	134006	UnapplyMultipleEntsAfterCur...	✓	Success		897 m
Function	134006	UnapplyMultipleEntsAfterExc...	✓	Success		950 m
Function	134006	UnapplyPaymentAppliedToMul...	✓	Success		4 sec

Successful Tests: 28
Failed Tests: 0
Skipped Tests: 0
Tests not Executed: 0

Running Tests

- Test Suite

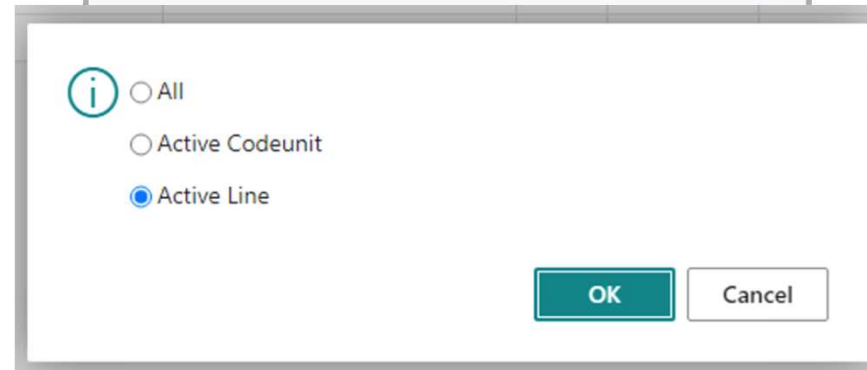
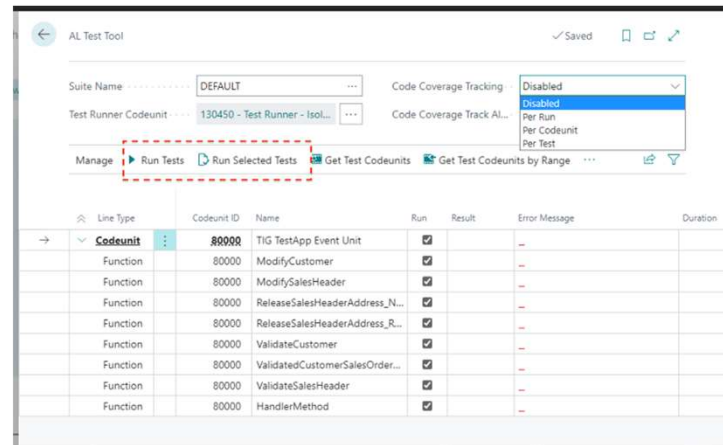


April 14-17, 2024

Sheraton San Diego Hotel & Marina, San Diego, CA

Running Tests

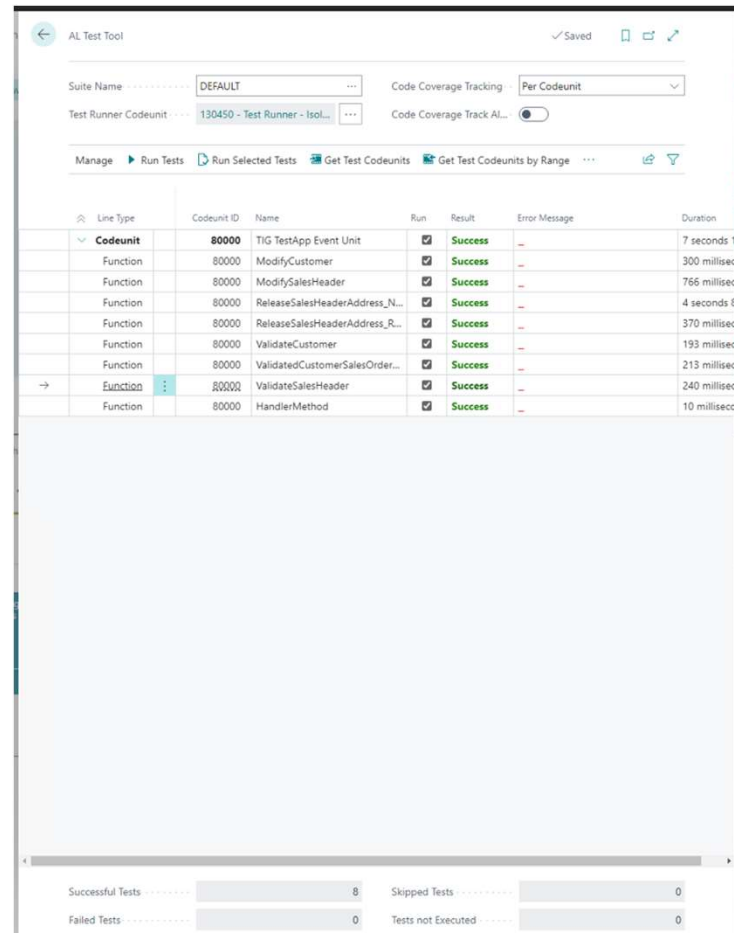
- Running Tests



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

Running Tests

- Completed Tests



The screenshot shows the AL Test Tool interface. At the top, there are fields for Suite Name (DEFAULT) and Test Runner Codeunit (130450 - Test Runner - Isol...). Below these are buttons for Manage, Run Tests, Run Selected Tests, Get Test Codeunits, and Get Test Codeunits by Range. The main table displays test results with columns for Line Type, Codeunit ID, Name, Run, Result, Error Message, and Duration. All tests shown are successful. At the bottom, a summary bar shows 8 Successful Tests, 0 Failed Tests, 0 Skipped Tests, and 0 Tests not Executed.

Line Type	Codeunit ID	Name	Run	Result	Error Message	Duration
Codeunit	80000	TIG TestApp Event Unit	✓	Success	---	7 seconds
Function	80000	ModifyCustomer	✓	Success	---	300 millise
Function	80000	ModifySalesHeader	✓	Success	---	766 millise
Function	80000	ReleaseSalesHeaderAddress_N...	✓	Success	---	4 seconds
Function	80000	ReleaseSalesHeaderAddress_R...	✓	Success	---	370 millise
Function	80000	ValidateCustomer	✓	Success	---	193 millise
Function	80000	ValidatedCustomerSalesOrder...	✓	Success	---	213 millise
Function	80000	ValidateSalesHeader	✓	Success	---	240 millise
Function	80000	HandlerMethod	✓	Success	---	10 millise

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

Running Tests

- Code Coverage

Code Coverage

Object Id Filter: Total # Lines: 0

Object Type Filter: Total Coverage %: 0.00

Required Coverage %: 90

Start Refresh Stop Backup/Restore More options

	Coverage %	Line Type	Object Type	Object ID	Line No.	No. of Lines
<input type="radio"/> Code						
<input type="radio"/> > Codeunit Document Errors Mgt. (9070)	0.66	Object	Codeunit	9070	0	
<input type="radio"/> > Codeunit Background Error Handling M...	1.05	Object	Codeunit	9079	0	
<input type="radio"/> > Codeunit Effective Permissions Mgt. (98...	0.58	Object	Codeunit	9852	0	
<input type="radio"/> > Codeunit DIOT Subscribers (27022)	7.89	Object	Codeunit	27022	0	
<input checked="" type="radio"/> > Codeunit TIG Untitled Management (50...	77.27	Object	Codeunit	50100	0	
→ <input checked="" type="radio"/> > Codeunit TIG Event Management (50101)	85.71	Object	Codeunit	50101	0	
<input type="radio"/> > Codeunit TIG TestApp Event Unit (80000)	100.00	Object	Codeunit	80000	0	
<input type="radio"/> > Codeunit TIG TestApp Library (80001)	86.67	Object	Codeunit	80001	0	
<input type="radio"/> > Codeunit Assert (130000)	8.04	Object	Codeunit	130000	0	
<input type="radio"/> > Codeunit Backup Management (130011)	3.14	Object	Codeunit	130011	0	
<input type="radio"/> > Codeunit Backup Storage (130012)	1.43	Object	Codeunit	130012	0	
<input type="radio"/> > Codeunit Reset State Before Test Run (1...	100.00	Object	Codeunit	130301	0	
<input type="radio"/> > Codeunit Set State Test Run (130302)	100.00	Object	Codeunit	130302	0	
<input type="radio"/> > Codeunit Library - Random (130440)	56.25	Object	Codeunit	130440	0	
<input type="radio"/> > Codeunit Test Runner - Isol. Codeunit (1...	40.00	Object	Codeunit	130450	0	
<input type="radio"/> > Codeunit ALTestRunner Reset Environme...	85.19	Object	Codeunit	130453	0	
<input type="radio"/> > Codeunit Test Runner - Mgt (130454)	67.74	Object	Codeunit	130454	0	
<input type="radio"/> > Codeunit Test Runner - Progress Dialog ...	56.41	Object	Codeunit	130455	0	
<input type="radio"/> > Codeunit Test Suite Mgt. (130456)	1.58	Object	Codeunit	130456	0	
<input type="radio"/> > Codeunit AL Code Coverage Mgt. (1304...	5.45	Object	Codeunit	130470	0	
<input type="radio"/> > Codeunit AL Code Coverage Subscribers...	22.73	Object	Codeunit	130471	0	
<input type="radio"/> > Codeunit Library - Sales (130509)	12.72	Object	Codeunit	130509	0	
<input type="radio"/> > Codeunit Library - Utility (131000)	16.72	Object	Codeunit	131000	0	
<input type="radio"/> > Codeunit Permissions Mock (131006)	80.00	Object	Codeunit	131006	0	
<input type="radio"/> > Codeunit Library - ERM (131300)	8.53	Object	Codeunit	131300	0	
<input type="radio"/> > Codeunit Library - Inventory (132201)	5.56	Object	Codeunit	132201	0	
<input type="radio"/> > Codeunit Library - Lower Permissions (1...	9.02	Object	Codeunit	132217	0	
<input type="radio"/> > Codeunit Permission Test Catalog (1322...	23.08	Object	Codeunit	132218	0	
<input type="radio"/> > Codeunit Library - Notification Mgt. (13...	18.75	Object	Codeunit	132222	0	
<input type="radio"/> > Codeunit DocumentNoVisibilityClearCac...	33.33	Object	Codeunit	137777	0	
<input type="radio"/> > Codeunit Check Prod. Order Status (990...	25.00	Object	Codeunit	99000777	0	
<input type="radio"/> > Codeunit Create Reserv. Entry (99000830)	1.00	Object	Codeunit	99000830	0	

Running Tests

- BCCContainerHelper: Get-TestsFromBCCContainer

```
Get-TestsFromBCCContainer `
  -containerName $containerName `
  -Credential $credential `
  -testSuite $testSuite `
  -ignoreGroups
```

```
Tests
-----
Name                                     Id
-----
{ModifyCustomer, ModifySalesHeader, ReleaseSalesHeaderAddress_NotRequired, ReleaseSalesHeaderAddress_Required...} TIG TestApp Event Unit 80000
```

Running Tests

- BCCContainerHelper: Get-TestsFromBCCContainer

```
$tests = Get-TestsFromBCCContainer `
    -containerName $containerName `
    -Credential $credential `
    -testSuite $testSuite `
    -ignoreGroups

foreach($test in $tests) {
    Write-Host -ForegroundColor Yellow $tests.Id $tests.Name
    Write-Output $test.Tests
}
```

```
80000 TIG TestApp Event Unit
ModifyCustomer
ModifySalesHeader
ReleaseSalesHeaderAddress_NotRequired
ReleaseSalesHeaderAddress_Required
ValidateCustomer
ValidatedCustomerSalesOrderCreated
ValidateSalesHeader
ConfirmYes
```

Running Tests

- BCContainerHelper: Run-TestsInBCContainer

```
Run-TestsInBCContainer`  
-containerName $containerName`  
-credential $credential`  
-testSuite $testSuite`  
-detailed`  
-AppendToXUnitResultFile:($true)`  
-XUnitResultFileName $xunitResultsFile
```

```
Codeunit 80000 TIG TestApp Event Unit Success (3.875 seconds)  
Testfunction ModifyCustomer Success (0.66 seconds)  
Testfunction ModifySalesHeader Success (0.233 seconds)  
Testfunction ReleaseSalesHeaderAddress_NotRequired Success (0.4 seconds)  
Testfunction ReleaseSalesHeaderAddress_Required Success (0.244 seconds)  
Testfunction ValidateCustomer Success (0.18 seconds)  
Testfunction ValidatedCustomerSalesOrderCreated Success (0.163 seconds)  
Testfunction ValidateSalesHeader Success (0.213 seconds)  
Testfunction ConfirmYes Success (0.007 seconds)
```

Running Tests

- BCCContainerHelper: Run-TestsInBCCContainer

```
$first = $true
$tests = Get-TestsFromBCCContainer `
    -containerName $containerName `
    -Credential $Credential `
    -testSuite $testSuite `
    -ignoreGroups

$tests | ForEach-Object {
```

```
<?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>
```


Running Tests

- BCContainerHelper: Run-ALPipeline

```
#####

$scriptTitle = 'Run-ALPipeline'

$startTime = [DateTime]::Now
Write-Host -ForegroundColor Green "$($startTime) : $($scriptTitle) start"

Run-ALPipeline `
  --pipelineName $pipelineName `
  --baseFolder $baseFolder `
  --appFolders $appFolders `
  --testFolders $testFolders `
  --installTestFramework `
  --installTestLibraries `
  --enablePerTenantExtensionCop `
  --enableUICop `
  --enableCodeCop `
  --enableAppSourceCop `
  --appVersion $appVersion `
  --Credential $credential `
  --CreateRuntimePackages `
  --AppSourceCopMandatoryAffixes $affix `
  --AppSourceCopSupportedCountries "US" `
  --rulesetFile $rulesetFile `
  --outputFolder $outputFolder `
  --testResultsFile $testResultsFile `
  --packagesFolder $packagesFolder

$stopTime = [DateTime]::Now
$timeSpend = ($stopTime - $startTime).TotalSeconds

Write-Host -ForegroundColor Green "$($stopTime) : $($scriptTitle) took $($timeSpend) seconds"
```


Running Tests

- BCContainerHelper: Run-ALPipeline

```
duction-to-Test-Automation-in-Business-Central\Supporting-Files\Powershell\Run-ALPipeline.ps1'
04/10/2023 18:32:28 : Run-ALPipeline start
Determining artifacts to use

Parameters

Pipeline name           DirectionsNA
Container name          directionsna-bld
Image name              my
ArtifactUrl             https://bcartifacts.azureedge.net/sandbox/22.0.54157.55112/us
SasToken                Not Specified
BcAuthContext           Not Specified
Environment
ReUseContainer          False
KeepContainer           False
Auth                   UserPassword
Credential              Specified
CompanyName
MemoryLimit            8G
FailOn                 none
TreatTestFailuresAsWarnings False
Enable Task Scheduler  False
Assign Premium Plan    False
Install Test Runner    False
Install Test Framework True
Install Test Libraries True
Install Perf. Toolkit  False
InstallOnlyReferencedApps False
generateDependencyArtifact False
CopySymbolsFromContainer False
enableCodeCop          True
enableAppSourceCop     True
enableUICop            True
enablePerTenantExtensionCop True
doNotPerformUpgrade    False
doNotPublishApps       False
uninstallRemovedApps   False
escapeFromCops         False
doNotBuildTests        False
doNotRunTests          False
doNotRunBcptTests      False
useDefaultAppSourceRuleSet False
rulesetFile            C:\Temp\Introduction-to-Test-Automation-in-Business-Central\Supporting-Files\untitled.ruleset.json
azureDevOps            False
gitlab                 False
githubActions          False
license file           Not specified
```

Questions?

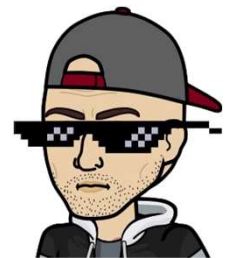
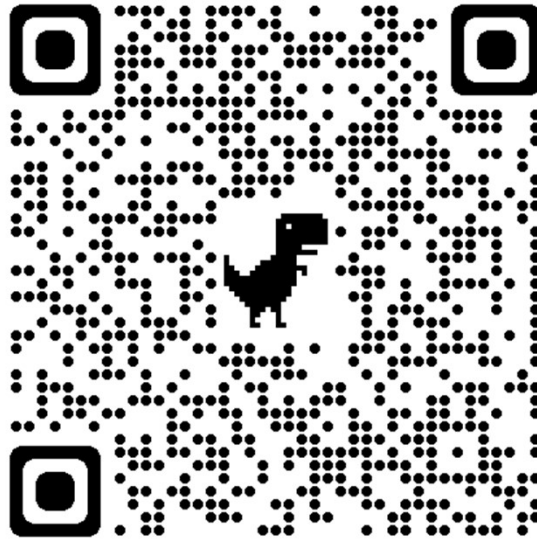


What's Next

- Establish Test Automation Policy
- Educate
- Explore



References





Brad Prendergast



Thank You!

Please remember to fill out your
session survey using the
Directions 2024 Mobile App!