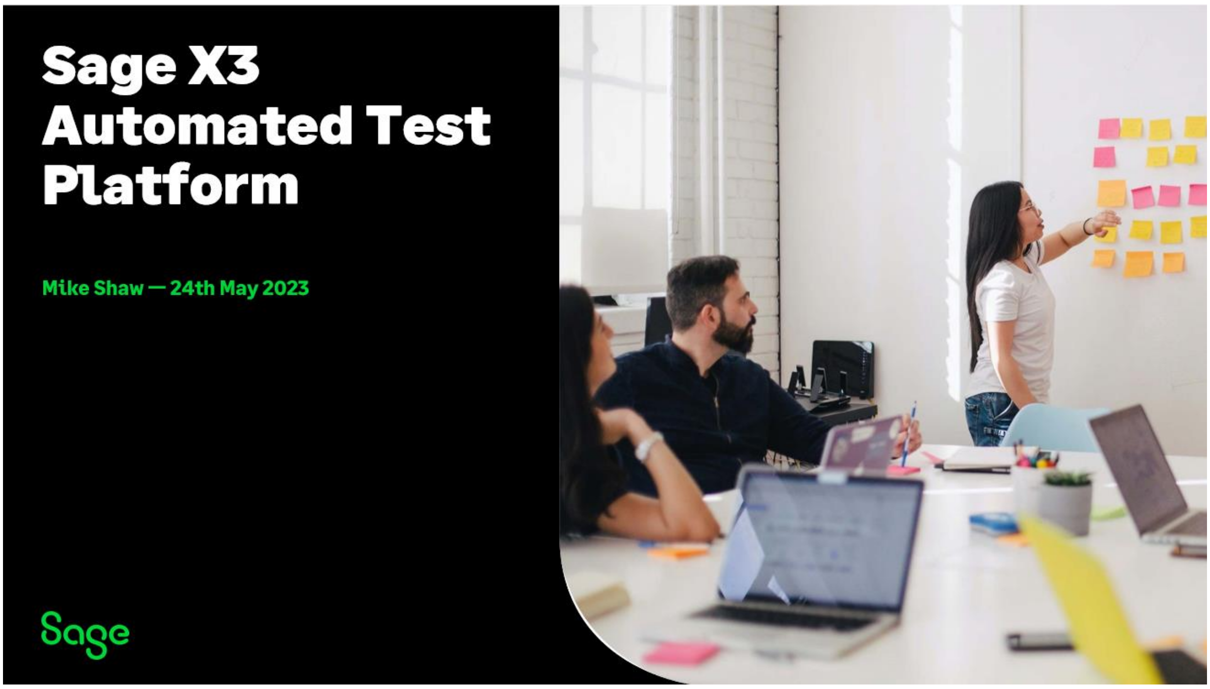
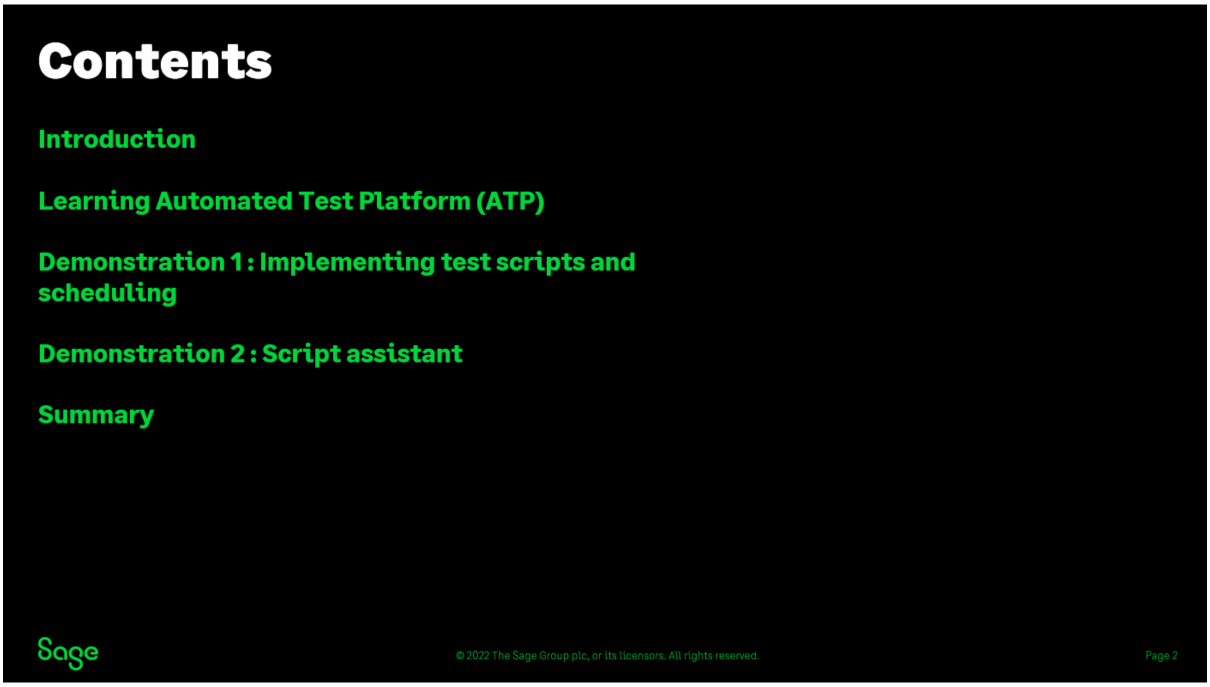
Updated: 23 April 2023



Sage X3 Automated Test Platform

Mike Shaw — 24th May 2023

Sage

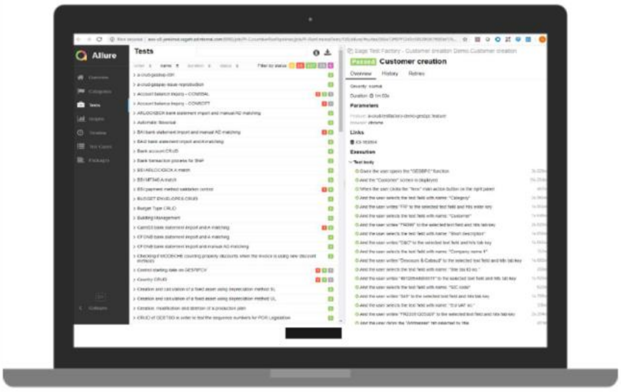


Contents

Sage

Introduction

Simplify Sage X3 upgrades and configurations and minimize manual testing.
Scalable and extensible to support ever changing real-world scenarios



Quickly upgrade with confidence and complete accuracy

Intelligent automated testing platform

Maximise efficiency and effectiveness of QA processes

Save time by automating user testing

Confidence that all customizations and modifications have gone through full, end to end regression testing

Introduction

Benefits

Improve quality, reduce costs and time

- Extend and expand testing capacity
- Cover more scenarios
- Provide continuous monitoring of ERP systems, which helps to find and resolve issues
- Meet and exceed predefined business requirements
- Supports regular continuous updates

Increase efficiency

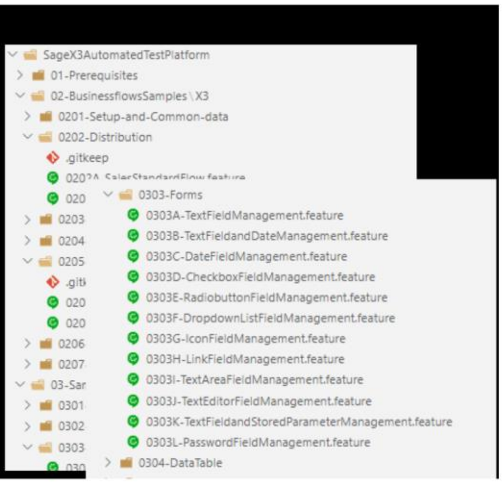
- Release testing resources from manual, repetitive processes
- Improved tester motivation and effectiveness
- User testing focused to business-critical tasks
- Automating regular and mundane testing tasks
- Documented evidence of the tests carried

Introduction

Fast start with library of sample scripts

2 different types of samples delivered:

- **Business flow samples**:
  - Sage X3 business flow examples for the different modules (Finance, distribution etc..)
- **Samples**: aimed at demoing the use of the different steps definitions delivered with Sage X3 Automated Test Platform Library and their interactions with Sage X3


SageX3AutomatedTestPlatform
01-Prerequisites
02-BusinessflowsSamples\X3
0201-Setup-and-Common-data
0202-Distribution
.gitkeep
0203
0204
0205
0206
0207
03-Sar
0301
0302
0303
0303-Forms
0303A-TextFieldManagement.feature
0303B-TextFieldandDateManagement.feature
0303C-DateFieldManagement.feature
0303D-CheckboxFieldManagement.feature
0303E-RadiobuttonFieldManagement.feature
0303F-DropdownListFieldManagement.feature
0303G-IconFieldManagement.feature
0303H-LinkFieldManagement.feature
0303I-TextAreaFieldManagement.feature
0303J-TextEditorFieldManagement.feature
0303K-TextFieldandStoredParameterManagement.feature
0303L-PasswordFieldManagement.feature
0304-DataTable

Introduction

- Install client program to create and run scripts. Implemented as Sage add-in to Visual Studio.
  - Create scripts using provided code sample templates, code snippets and/or from scratch.
  - Run scripts interactively using Google engine. i.e. runs in the same way as if user is typing into a Chrome browser.
  - (Optional) Separately provided and installed “Script assistant” speeds up script development. Delivered as Sage X3 patch.
- (Optional) Jenkins (https://www.jenkins.io/) schedules test script runs. Independent of Sage X3.

Test scripts should ideally reflect real user activity and emulate a representative workload covering all essential business functionality. You should therefore carefully design your scripts and prepare data set(s) to test against which will achieve your testing objectives.

Learning Automated Test Platform (ATP)

Sage University classes

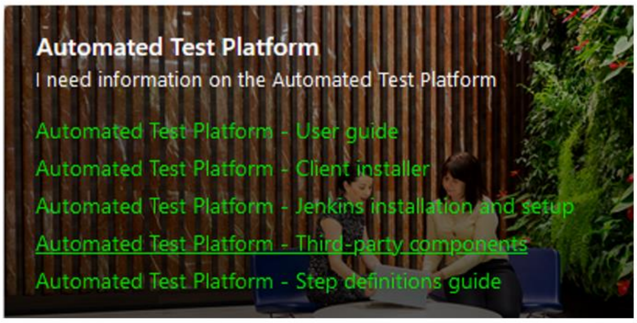
- X3 - Sage X3 - Introduction to Sage X3 Automated Test Platform
  - 15-minute overview
- X3 - Sage X3 - Knowledge Bites: ATP (Automated Test Platform)
  - 90-minute introduction
- X3 - Sage X3 - Installing and Configuring Sage X3 Automated Test Platform
  - 30-minute steps to install
- X3 - Sage X3 - Automated Test Platform: Writing Test Scripts for Sage X3
  - 12 hours. Understand writing ATP scripts
- X3 - Sage X3 - Expert Workshop: Meeting the challenges of a Sage X3 project with ATP
  - 9 hours. How to use

Learning Automated Test Platform (ATP)

- **Online Help**

Prereqs https://online-help.sageerpx3.com/erp/12/public/prerequisites_atp.html

Sage X3 Technical Help https://online-help.sageerpx3.com/erp/12/public/index.html



Automated Test Platform

I need information on the Automated Test Platform

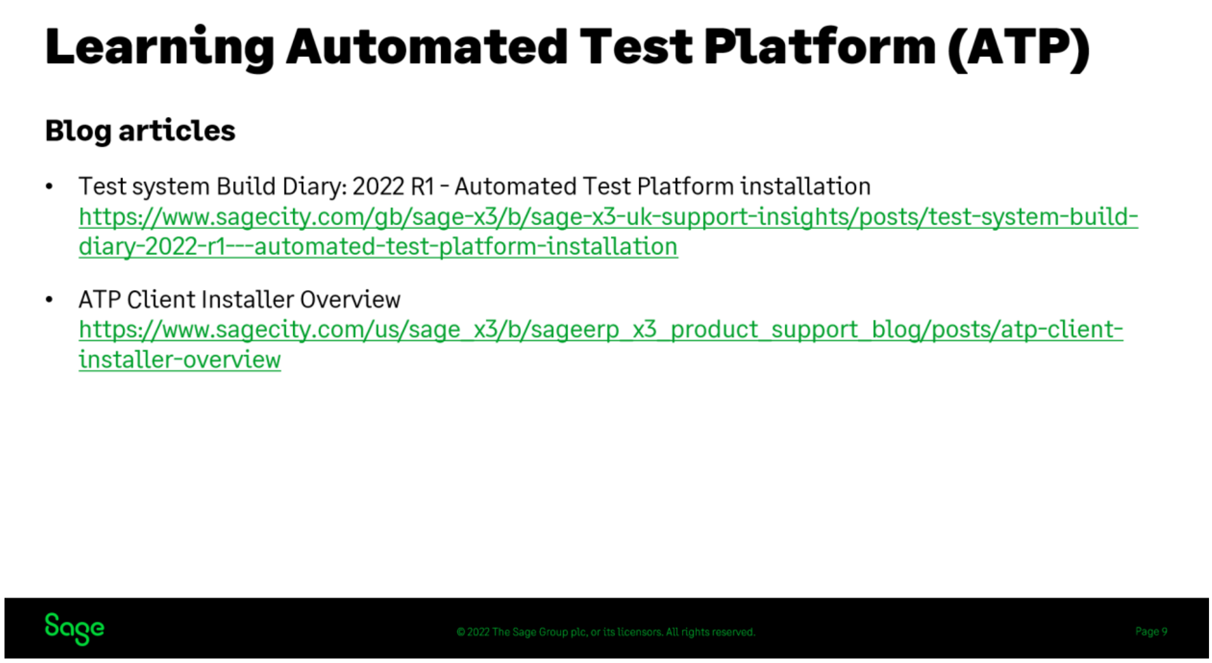
Automated Test Platform - User guide

Automated Test Platform - Client installer

Automated Test Platform - Jenkins installation and setup

Automated Test Platform - Third-party components

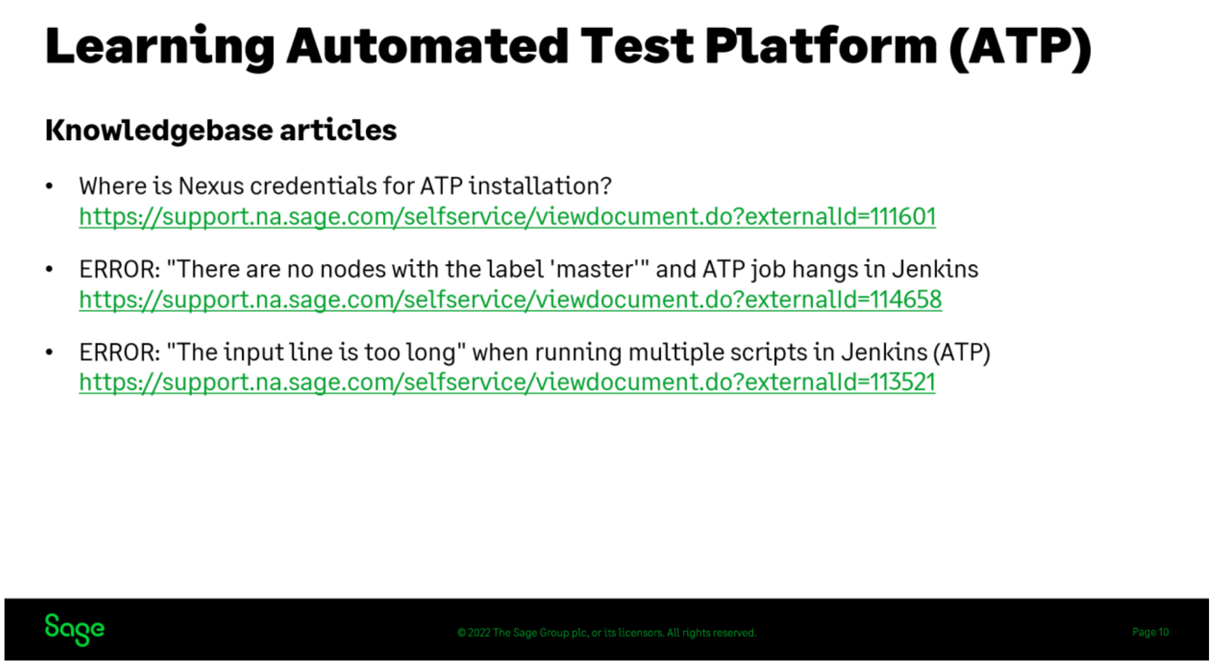
Automated Test Platform - Step definitions guide



Learning Automated Test Platform (ATP)

Blog articles

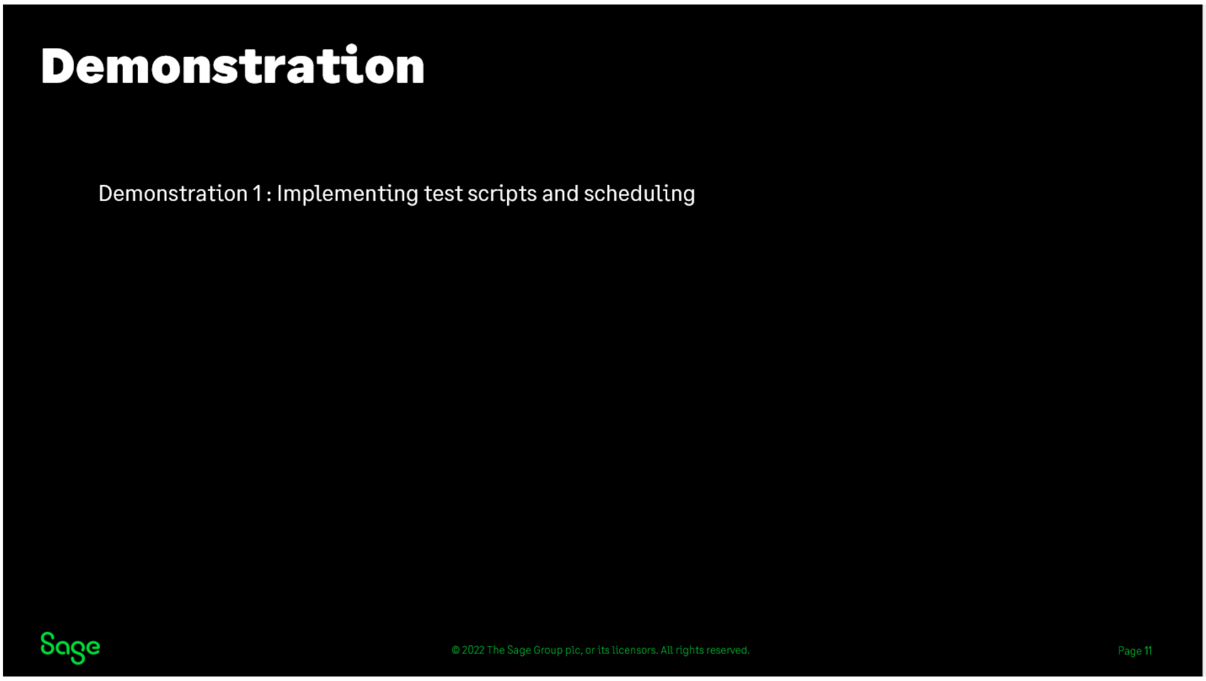
- Test system Build Diary: 2022 R1 - Automated Test Platform installation
  https://www.sagecity.com/gb/sage-x3/b/sage-x3-uk-support-insights/posts/test-system-build-diary-2022-r1---automated-test-platform-installation
- ATP Client Installer Overview
  https://www.sagecity.com/us/sage_x3/b/sageerp_x3_product_support_blog/posts/atp-client-installer-overview



Learning Automated Test Platform (ATP)

Knowledgebase articles

- Where is Nexus credentials for ATP installation?
  https://support.na.sage.com/selfservice/viewdocument.do?externalId=111601
- ERROR: "There are no nodes with the label 'master'" and ATP job hangs in Jenkins
  https://support.na.sage.com/selfservice/viewdocument.do?externalId=114658
- ERROR: "The input line is too long" when running multiple scripts in Jenkins (ATP)
  https://support.na.sage.com/selfservice/viewdocument.do?externalId=113521



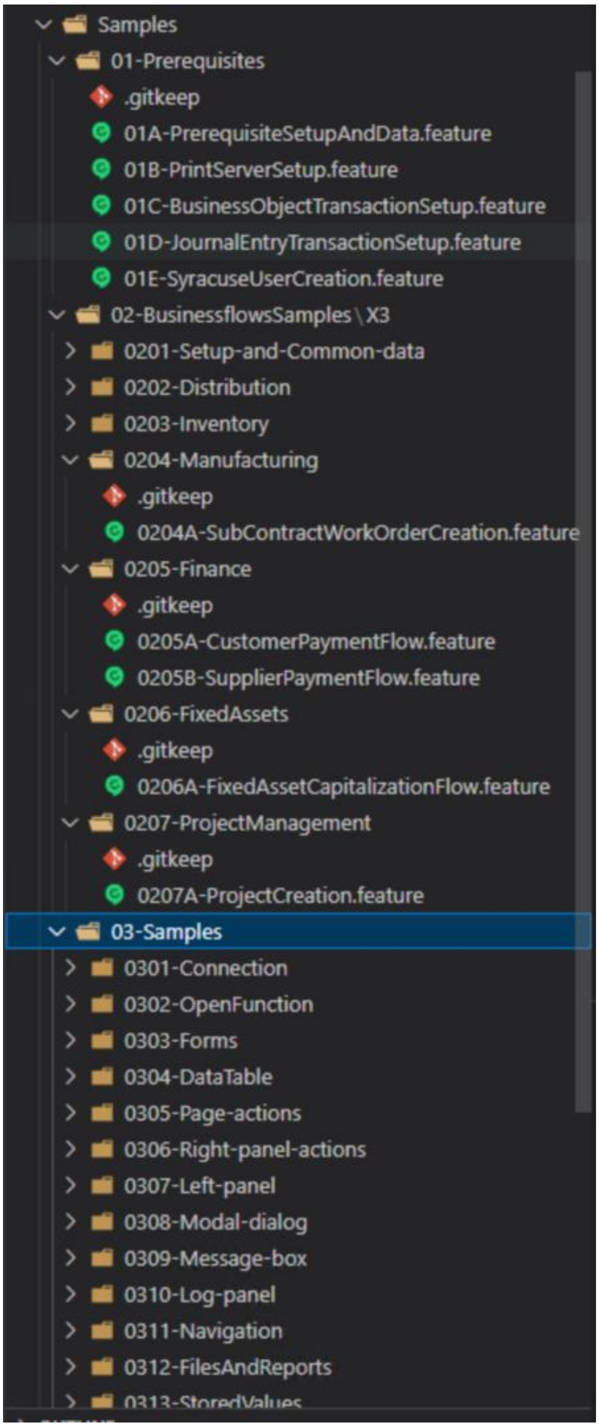
Demonstration

Demonstration 1 : Implementing test scripts and scheduling

Launch Visual Studio

Have a quick look at the provided samples

- Cucumber-config
  - Parameters
- Cucumber-user-config
  - User, user-stored-values



```
Samples
  01-Prerequisites
    .gitkeep
    01A-PrerequisiteSetupAndData.feature
    01B-PrintServerSetup.feature
    01C-BusinessObjectTransactionSetup.feature
    01D-JournalEntryTransactionSetup.feature
    01E-SyracuseUserCreation.feature
  02-BusinessflowsSamples \ X3
    0201-Setup-and-Common-data
    0202-Distribution
    0203-Inventory
    0204-Manufacturing
      .gitkeep
      0204A-SubContractWorkOrderCreation.feature
    0205-Finance
      .gitkeep
      0205A-CustomerPaymentFlow.feature
      0205B-SupplierPaymentFlow.feature
    0206-FixedAssets
      .gitkeep
      0206A-FixedAssetCapitalizationFlow.feature
    0207-ProjectManagement
      .gitkeep
      0207A-ProjectCreation.feature
  03-Samples
    0301-Connection
    0302-OpenFunction
    0303-Forms
    0304-DataTable
    0305-Page-actions
    0306-Right-panel-actions
    0307-Left-panel
    0308-Modal-dialog
    0309-Message-box
    0310-Log-panel
    0311-Navigation
    0312-FilesAndReports
    0313-StoredValues
```

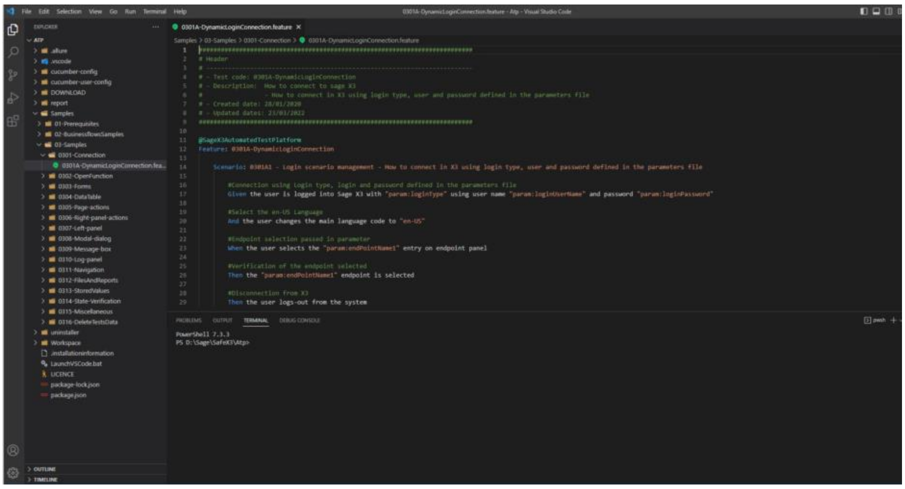
Samples, 02-Businessflowssamples

Complete example business flow to do something reasonably useful.

Samples, 03-Samples

Complete, but more contrived in order to demonstrate specific functionality.

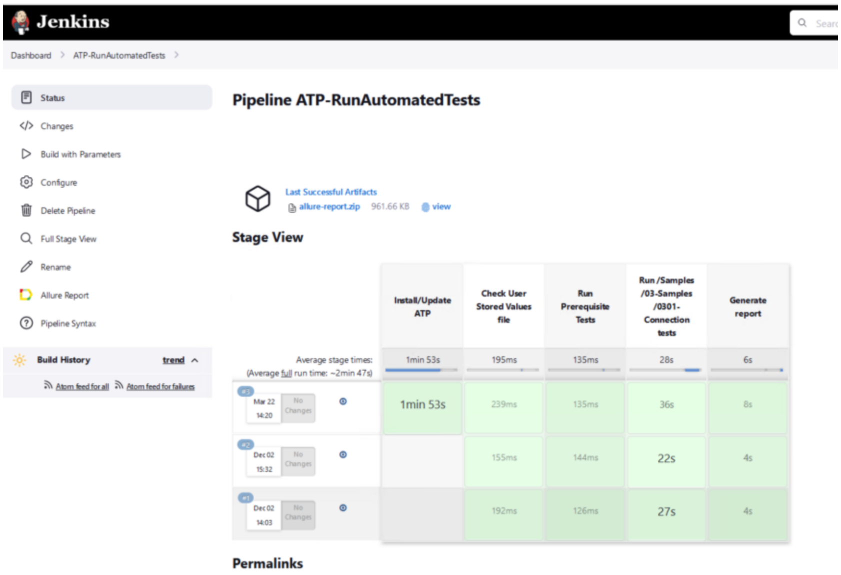
Look at the script “0301-Connection”


0301A-DynamicLoginConnection.feature - Atp - Visual Studio Code
0301A-DynamicLoginConnection.feature
Samples > 03-Samples > 0301-Connection > 0301A-DynamicLoginConnection.feature
PROBLEMS
OUTPUT
TERMINAL
DEBUG CONSOLE

```
##############################################################################
# Header
# ----------------------------------------------------------------------------
# - Test code: 0301A-DynamicLoginConnection
# - Description:  How to connect to sage X3
#                 - How to connect in X3 using login type, user and password defined in the parameters file
# - Created date: 28/01/2020
# - Updated date1: 23/03/2022
##############################################################################

@SageX3AutomatedTestPlatform
Feature: 0301A-DynamicLoginConnection

    Scenario: 0301A1 - Login scenario management - How to connect in X3 using login type, user and password defined in the parameters file

        #Connection using login type, login and password defined in the parameters file
        Given the user is logged into Sage X3 with "param:loginType" using user name "param:loginUserName" and password "param:loginPassword"

        #Select the en-US language
        And the user changes the main language code to "en-US"

        #Endpoint selection passed in parameter
        When the user selects the "param:endPointName1" entry on endpoint panel

        #Verification of the endpoint selected
        Then the "param:endPointName1" endpoint is selected

        #Disconnection from X3
        Then the user logs-out from the system
```

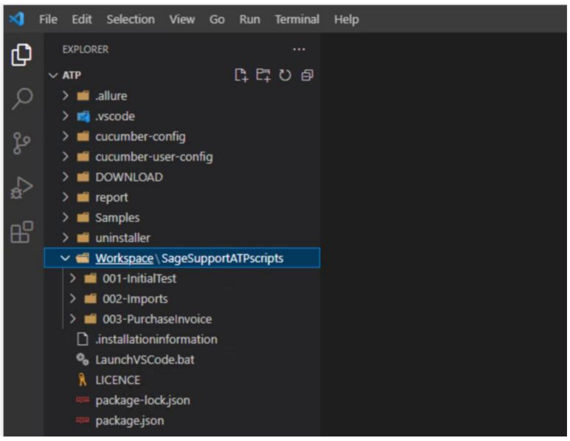
```
PowerShell 7.3.3
PS D:\Sage\SafeX3\Atp>
```

Launch Jenkins and run "ATP-RunAutomatedTests" (Build with parameters) to confirm all works OK



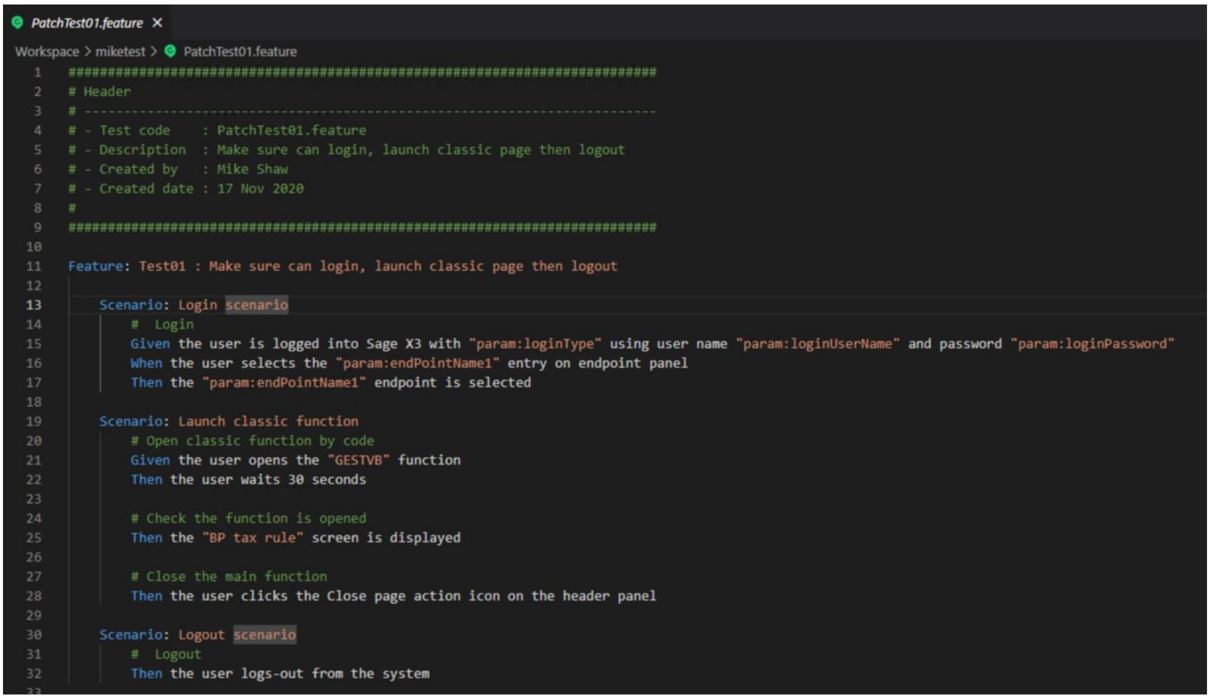
	Install/Update ATP	Check User Stored Values file	Run Prerequisite Tests	Run /Samples /03-Samples /0301-Connection tests	Generate report
Average stage times: (Average full run time: ~2min 47s)	1min 53s	195ms	135ms	28s	6s
#3 Mar 22 14:20 No Changes	1min 53s	239ms	135ms	36s	8s
#2 Dec 02 15:32 No Changes		155ms	144ms	22s	4s
#1 Dec 02 14:03 No Changes		192ms	126ms	27s	4s

Copy local copy of ATP test scripts "SageSupportATPscripts" directory to D:\Sage\SafeX3\Atp\Workspace directory



Take a look at (some of) the provided scripts and execute individually.

PatchTest01



```
###########################################################################
# Header
# -------------------------------------------------------------------------
# - Test code    : PatchTest01.feature
# - Description  : Make sure can login, launch classic page then logout
# - Created by   : Mike Shaw
# - Created date : 17 Nov 2020
#
###########################################################################

Feature: Test01 : Make sure can login, launch classic page then logout

    Scenario: Login scenario
        #  Login
        Given the user is logged into Sage X3 with "param:loginType" using user name "param:loginUserName" and password "param:loginPassword"
        When the user selects the "param:endPointName1" entry on endpoint panel
        Then the "param:endPointName1" endpoint is selected

    Scenario: Launch classic function
        # Open classic function by code
        Given the user opens the "GESTVB" function
        Then the user waits 30 seconds

        # Check the function is opened
        Then the "BP tax rule" screen is displayed

        # Close the main function
        Then the user clicks the Close page action icon on the header panel

    Scenario: Logout scenario
        #  Logout
        Then the user logs-out from the system
```

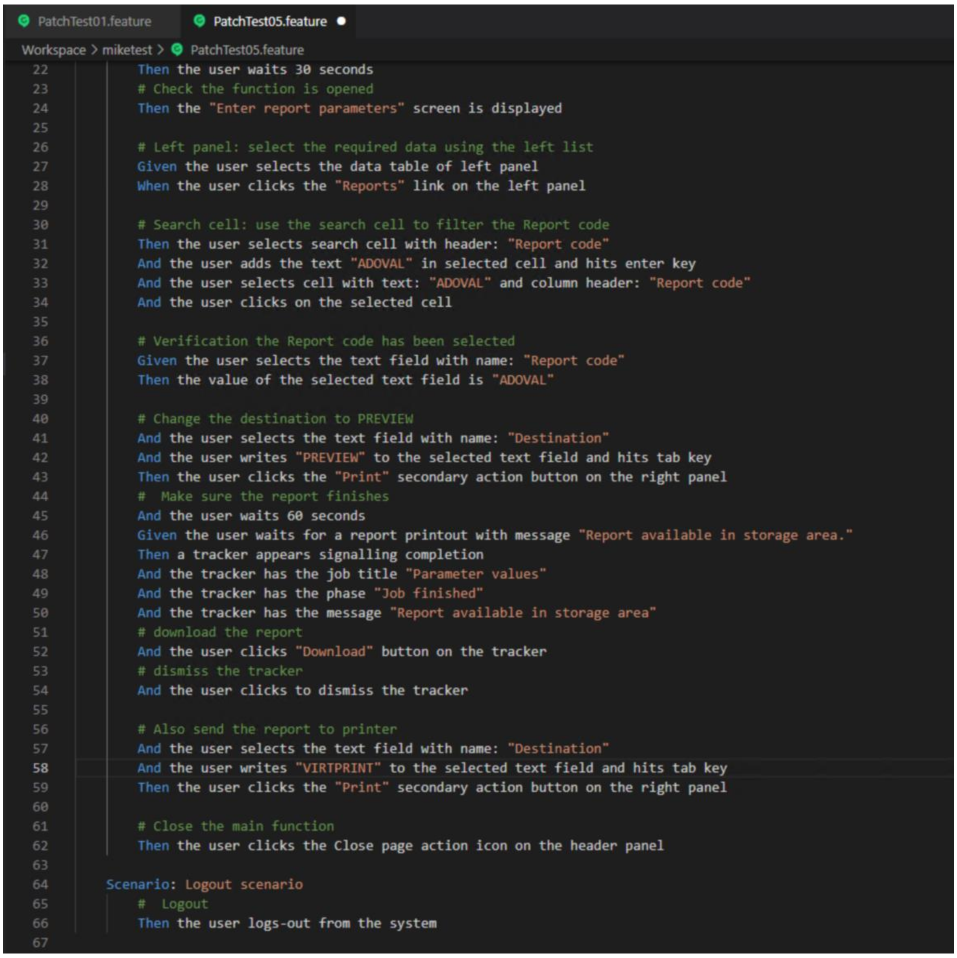
Note that my script does not bother to select the en-US language. This relies on the user I'm using to not have the language changed manually. If it is then I may see failures generated when I run the script because of the language being UK rather than US English (for example)

Can either press F5 key, click "Run and Debug" icon on left side, or select Run, Start Debugging from drop down menu

NOTE: the username/password is needed for each script.

PatchTest05

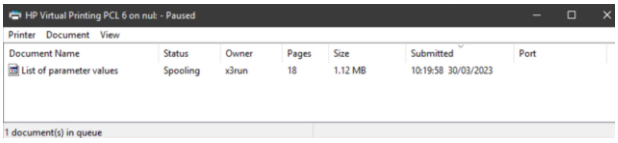
Check the file and see we assume preview destination "PREVIEW" and print destination "PRINT" so need to confirm these exist. PREVIEW is there and active, but need to change the printer to "VIRTPRINT". Also check in Control Panel, Printers that the printer is "Paused"



```
        Then the user waits 30 seconds
        # Check the function is opened
        Then the "Enter report parameters" screen is displayed

        # Left panel: select the required data using the left list
        Given the user selects the data table of left panel
        When the user clicks the "Reports" link on the left panel

        # Search cell: use the search cell to filter the Report code
        Then the user selects search cell with header: "Report code"
        And the user adds the text "ADOVAL" in selected cell and hits enter key
        And the user selects cell with text: "ADOVAL" and column header: "Report code"
        And the user clicks on the selected cell

        # Verification the Report code has been selected
        Given the user selects the text field with name: "Report code"
        Then the value of the selected text field is "ADOVAL"

        # Change the destination to PREVIEW
        And the user selects the text field with name: "Destination"
        And the user writes "PREVIEW" to the selected text field and hits tab key
        Then the user clicks the "Print" secondary action button on the right panel
        #  Make sure the report finishes
        And the user waits 60 seconds
        Given the user waits for a report printout with message "Report available in storage area."
        Then a tracker appears signalling completion
        And the tracker has the job title "Parameter values"
        And the tracker has the phase "Job finished"
        And the tracker has the message "Report available in storage area"
        # download the report
        And the user clicks "Download" button on the tracker
        # dismiss the tracker
        And the user clicks to dismiss the tracker

        # Also send the report to printer
        And the user selects the text field with name: "Destination"
        And the user writes "VIRTPRINT" to the selected text field and hits tab key
        Then the user clicks the "Print" secondary action button on the right panel

        # Close the main function
        Then the user clicks the Close page action icon on the header panel

    Scenario: Logout scenario
        #  Logout
        Then the user logs-out from the system

```

Save the file then run it (F5)

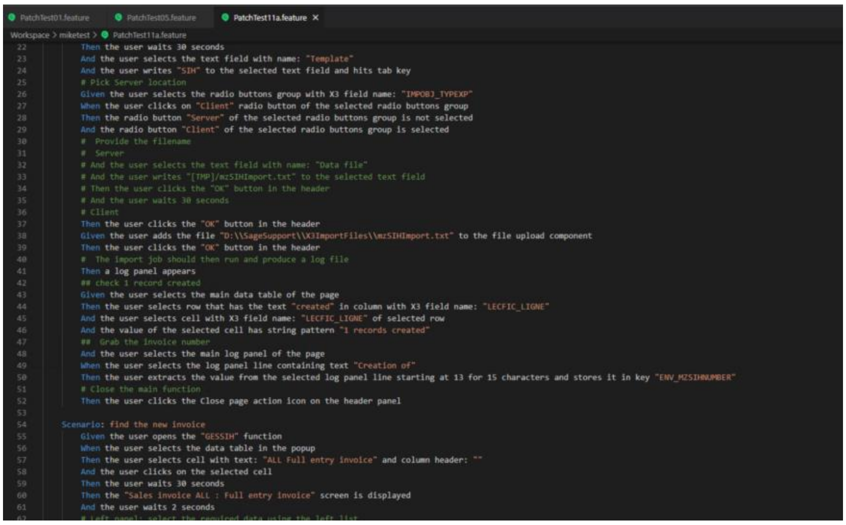
Check windows report queue and remove the printed file.



Document Name	Status	Owner	Pages	Size	Submitted	Port
List of parameter values	Spooling	x3run	18	1.12 MB	10:19:58 30/03/2023	

1 document(s) in queue

PatchTest11a



```
        Then the user waits 30 seconds
        And the user selects the text field with name: "Template"
        And the user writes "SIH" to the selected text field and hits tab key
        # Pick Server location
        Given the user selects the radio buttons group with X3 field name: "IMPOBJ_TYPEXP"
        When the user clicks on "Client" radio button of the selected radio buttons group
        Then the radio button "Server" of the selected radio buttons group is not selected
        And the radio button "Client" of the selected radio buttons group is selected
        #  Provide the filename
        #  Server
        # And the user selects the text field with name: "Data file"
        # And the user writes "[TMP]/mzSIHImport.txt" to the selected text field
        # Then the user clicks the "OK" button in the header
        # And the user waits 30 seconds
        # Client
        Then the user clicks the "OK" button in the header
        Given the user adds the file "D:\\SageSupport\\X3ImportFiles\\mzSIHImport.txt" to the file upload component
        Then the user clicks the "OK" button in the header
        #  The import job should then run and produce a log file
        Then a log panel appears
        ## check 1 record created
        Given the user selects the main data table of the page
        Then the user selects row that has the text "created" in column with X3 field name: "LECFIC_LIGNE"
        And the user selects cell with X3 field name: "LECFIC_LIGNE" of selected row
        And the value of the selected cell has string pattern "1 records created"
        ##  Grab the invoice number
        And the user selects the main log panel of the page
        When the user selects the log panel line containing text "Creation of"
        Then the user extracts the value from the selected log panel line starting at 13 for 15 characters and stores it in key "ENV_MZSIHNUMBER"
        # Close the main function
        Then the user clicks the Close page action icon on the header panel

    Scenario: find the new invoice
        Given the user opens the "GESSIH" function
        When the user selects the data table in the popup
        Then the user selects cell with text: "ALL Full entry invoice" and column header: ""
        And the user clicks on the selected cell
        Then the user waits 30 seconds
        Then the "Sales invoice ALL : Full entry invoice" screen is displayed
        And the user waits 2 seconds
```

Notice here we are importing a client file so need to make sure the file exists there.   Also notice we are storing a SIH number to then retrieve as a query later in the script.

We can see the popup file selection screen is left behind as this is out of our control, which is a bit annoying, but can manually close it if you want to see what's going on.

Do a quick review of the other scripts to see any other pre-reqs/assumptions:

Copy “mzPAYimport.txt” and “mzITFimport.txt” files to “D:\Sage\X3ERPV12\Folders\SEED\tmp” for test 11b and 11c

Make sure the Accounting Task is “active” for test 16.   Also confirm ADMIN user language is set to “US”

Jenkins setup

Now we’ll setup a new Jenkins Pipeline to be able to run these tests in background.

Login to Jenkins

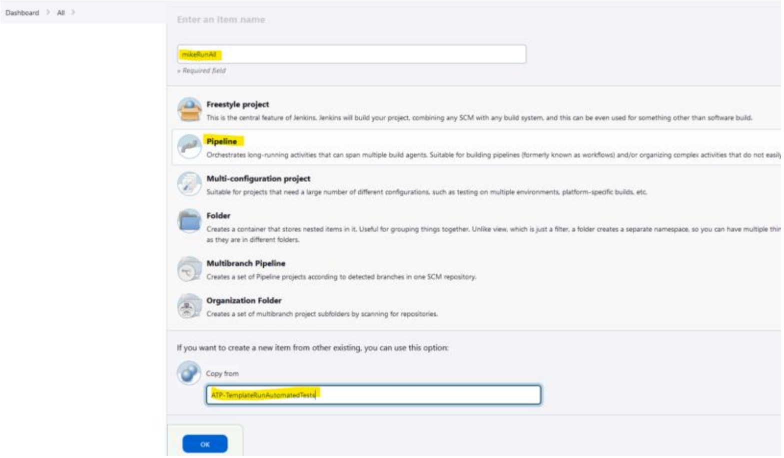
Click "New item"

Enter item name for the new pipeline, for example "mikeRunAll"

Select "Pipeline"

In "Copy from" enter ATP then pick the "ATP-TemplateRunAutomatedTests"

Click OK



Enter an item name

mikeRunAll

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easil

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple thin as they are in different folders.

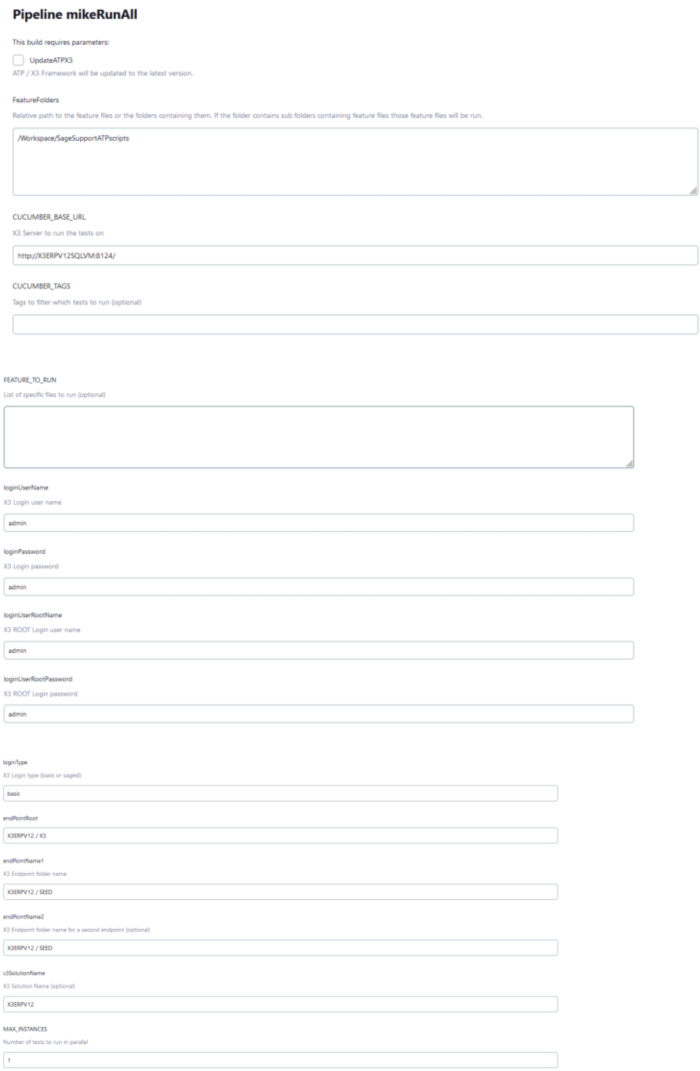
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

Organization Folder
Creates a set of multibranch project subfolders by scanning for repositories.

If you want to create a new item from other existing, you can use this option:

Copy from
ATP-TemplateRunAutomatedTests

Fill in the parameters as needed.



Pipeline mikeRunAll

This build requires parameters:

☐ UpdateATPX3
ATP / X3 Framework will be updated to the latest version.

FeatureFolders
Relative path to the feature files or the folders containing them. If the folder contains sub folders containing feature files those feature files will be run.
/Workspace/SageSupportATPscripts

CUCUMBER_BASE_URL
X3 Server to run the tests on
http://X3ERPV12SQLVM:8124/

CUCUMBER_TAGS
Tags to filter which tests to run (optional)

FEATURE_TO_RUN
List of specific files to run (optional)

loginUserName
X3 Login user name
admin

loginPassword
X3 Login password
admin

loginUserRootName
X3 ROOT Login user name
admin

loginUserRootPassword
X3 ROOT Login password
admin

loginType
X3 Login type (basic or sageid)
basic

endPointRoot
X3ERPV12 / X3

endPointName1
X3 Endpoint folder name
X3ERPV12 / SEED

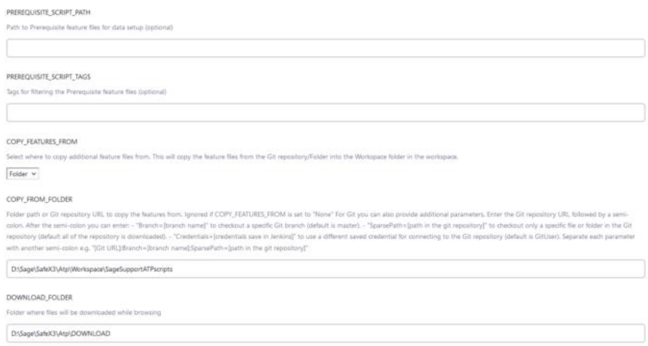
endPointName2
X3 Endpoint folder name for a second endpoint (optional)
X3ERPV12 / SEED

x3SolutionName
X3 Solution Name (optional)
X3ERPV12

MAX_INSTANCES
Number of tests to run in parallel
1

NOTES:

- The endPointName needs to be “Description”, not “Name”.
- Regardless of the online help, setting MAX_INSTANCES does run scripts in parallel.



PREREQUISITE_SCRIPT_PATH

Path to Prerequisite feature files for data setup (optional)

PREREQUISITE_SCRIPT_TAGS

Tags for filtering the Prerequisite feature files (optional)

COPY_FEATURES_FROM

Select where to copy additional feature files from. This will copy the feature files from the Git repository/Folder into the Workspace folder in the workspace.

Folder

COPY_FROM_FOLDER

Folder path or Git repository URL to copy the features from. Ignored if COPY_FEATURES_FROM is set to "None" For Git you can also provide additional parameters. Enter the Git repository URL followed by a semi-colon. After the semi-colon you can enter: - "Branch=[branch name]" to checkout a specific Git branch (default is master). - "SparsePath=[path in the git repository]" to checkout only a specific file or folder in the Git repository (default all of the repository is downloaded). - "Credentials=[credentials save in Jenkins]" to use a different saved credential for connecting to the Git repository (default is GitUser). Separate each parameter with another semi-colon e.g. "[Git URL];Branch=[branch name];SparsePath=[path in the git repository]"

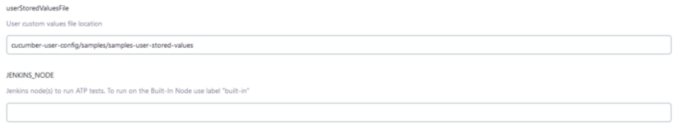
D:\Sage\SafeX3\Atp\Workspace\SageSupportATPscripts

DOWNLOAD_FOLDER

Folder where files will be downloaded while browsing

D:\Sage\SafeX3\Atp\DOWNLOAD

In my case I am running my feature files on the same machine as creating them, so can just use the "COPY_FEATURES_FROM" and "Folder" option to copy the scripts automatically. You can use GitHub to store the scripts or copy them into the Jenkins runtime area manually if you want to. (The scripts are stored in C:\Users\x3admin\AppData\Local\Jenkins\.jenkins\workspace\mikeRunAll\Workspace directory in my case)



userStoredValuesFile

User custom values file location

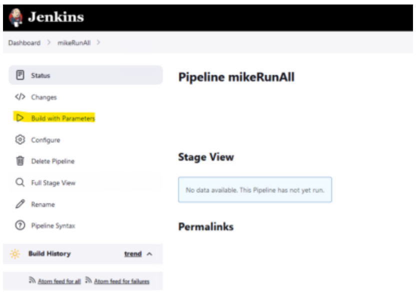
cucumber-user-config/samples/samples-user-stored-values

JENKINS_NODE

Jenkins node(s) to run ATP tests. To run on the Built-In Node use label "built-in"

Then "Save"

Run the pipeline using "Build with parameters" You can change the parameters if you want, but in my case, I can run with the defaults.



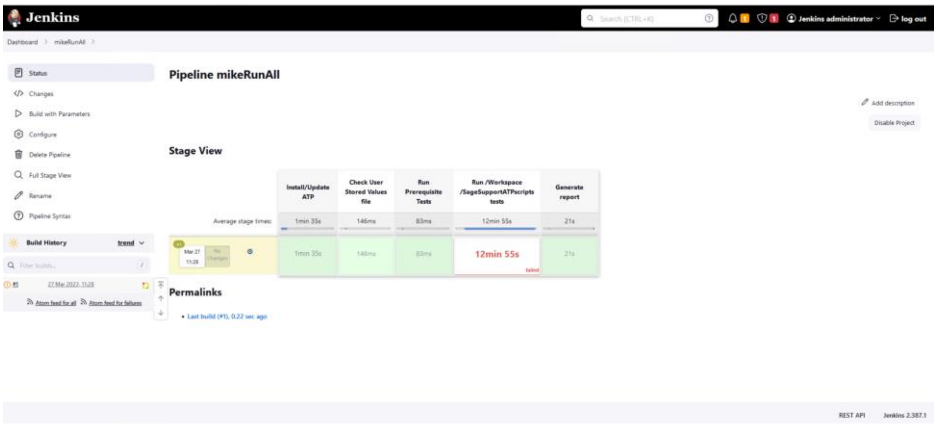
Pipeline mikeRunAll

Stage View

No data available. This Pipeline has not yet run.

Permalinks

Check the results look OK



Pipeline mikeRunAll

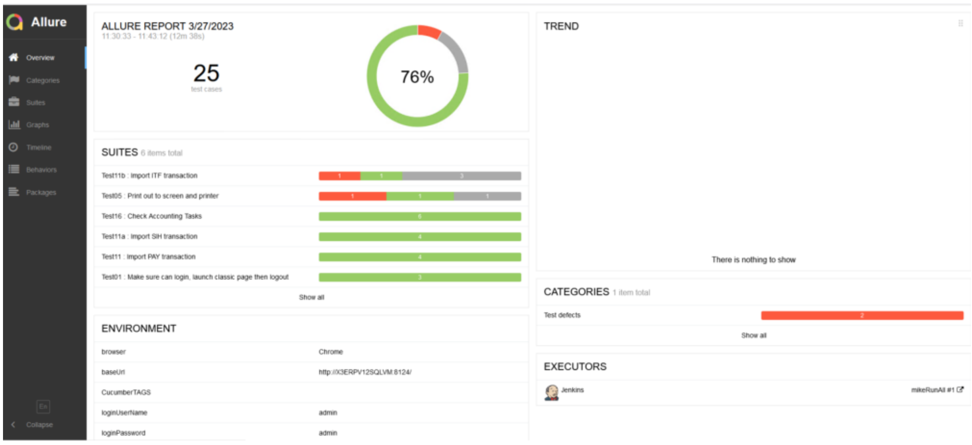
Stage View

	Install/Update ATP	Check User Stored Values file	Run Prerequisite Tests	Run /Workspace /SageSupportATPscripts tests	Generate report
Average stage times:	1min 35s	146ms	83ms	12min 55s	21s
#1 Mar 27 11:28 No Changes	1min 35s	146ms	83ms	12min 55s failed	21s

Permalinks

- Last build (#1), 0.22 sec ago

If there are errors, look at the Allure report to start with where you can drill into any tests that failed.



ALLURE REPORT 3/27/2023
11:30:33 - 11:43:12 (12m 38s)
25 test cases

SUITES 6 items total

Test11b : Import ITF transaction	1 1 3
Test05 : Print out to screen and printer	1 1 1
Test16 : Check Accounting Tasks	6
Test11a : Import SIH transaction	4
Test11 : Import PAY transaction	4
Test01 : Make sure can login, launch classic page then logout	3

ENVIRONMENT

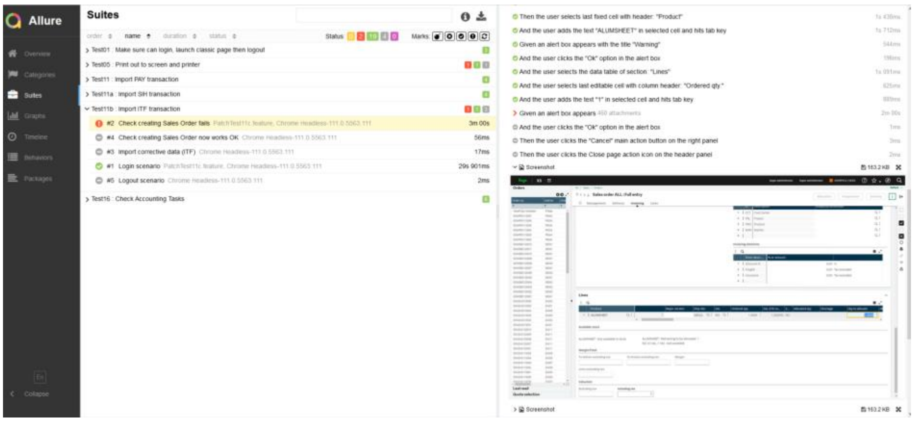
browser	Chrome
baseUrl	http://X3ERPV12SQLVM:8124/
CucumberTAGS	
loginUserName	admin
loginPassword	admin

There is nothing to show

CATEGORIES 1 item total

Test defects	2

EXECUTORS
Jenkins mikeRunAll #1



Suites

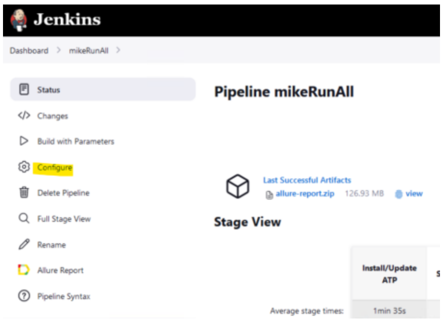
- Test01 : Make sure can login, launch classic page then logout
- Test05 : Print out to screen and printer
- Test11 : Import PAY transaction
- Test11a : Import SIH transaction
- Test11b : Import ITF transaction
  - #2 Check creating Sales Order fails PatchTest11c.feature, Chrome Headless-111.0.5563.111 3m 00s
  - #4 Check creating Sales Order now works OK Chrome Headless-111.0.5563.111 56ms
  - #3 Import corrective data (ITF) Chrome Headless-111.0.5563.111 17ms
  - #1 Login scenario PatchTest11c.feature, Chrome Headless-111.0.5563.111 29s 901ms
  - #5 Logout scenario Chrome Headless-111.0.5563.111 2ms
- Test16 : Check Accounting Tasks

Then the user selects last fixed cell with header: "Product"
And the user adds the text "ALUMSHEET" in selected cell and hits tab key 1s 712ms
Given an alert box appears with the title "Warning"
And the user clicks the "Ok" option in the alert box
And the user selects the data table of section: "Lines"
And the user selects last editable cell with column header: "Ordered qty."
And the user adds the text "1" in selected cell and hits tab key
Given an alert box appears
And the user clicks the "Ok" option in the alert box 1ms
Then the user clicks the "Cancel" main action button on the right panel 3ms
Then the user clicks the Close page action icon on the header panel 2ms

NOTE: the Sales Order import failure (Test 11b) is due to the way the test script is written does not cope with multiple runs.

Once the test is working satisfactorily from Jenkins, you can optionally schedule the test to run periodically.

Select your pipeline and click "Configure"



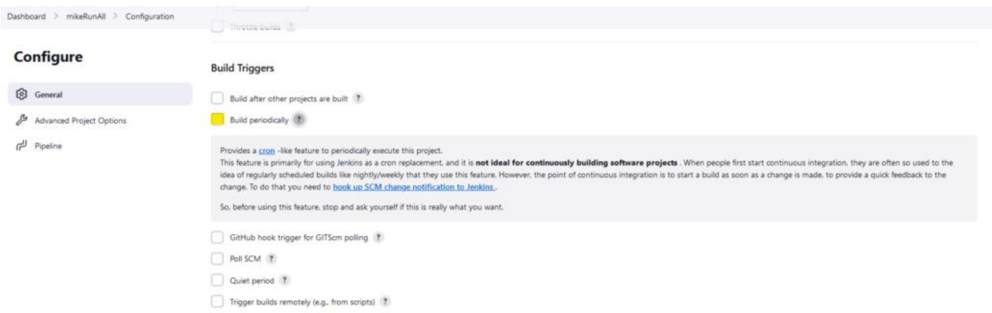
Pipeline mikeRunAll

Last Successful Artifacts
allure-report.zip 126.93 MB

Stage View

Install/Update ATP
Average stage times: 1min 35s

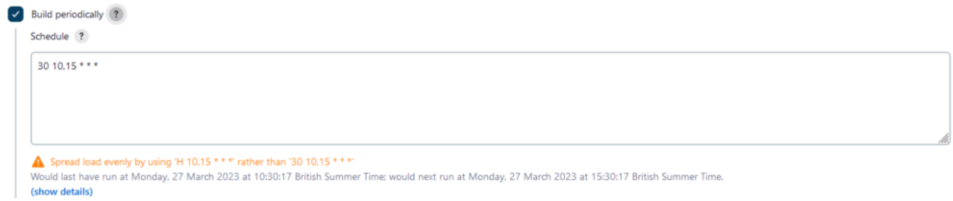
Scroll to the "Build triggers" towards the bottom of the page, where you can click the "?" to see the help.


Dashboard > mikeRunAll > Configuration
Throttle builds
Configure
General
Advanced Project Options
Pipeline
Build Triggers
☐ Build after other projects are built ?
Build periodically ?
☐ GitHub hook trigger for GITScm polling ?
☐ Poll SCM ?
☐ Quiet period ?
☐ Trigger builds remotely (e.g., from scripts) ?

Provides a cron -like feature to periodically execute this project.
This feature is primarily for using Jenkins as a cron replacement, and it is **not ideal for continuously building software projects** . When people first start continuous integration, they are often so used to the idea of regularly scheduled builds like nightly/weekly that they use this feature. However, the point of continuous integration is to start a build as soon as a change is made, to provide a quick feedback to the change. To do that you need to hook up SCM change notification to Jenkins .

So, before using this feature, stop and ask yourself if this is really what you want.

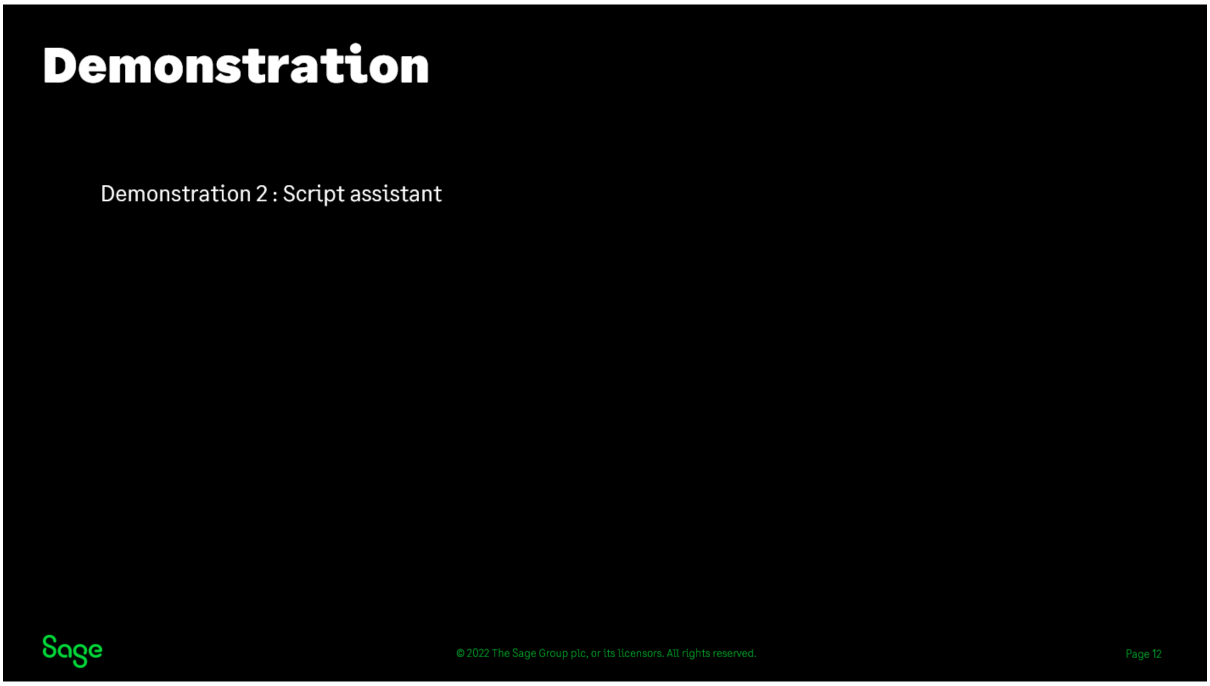
Check the “Build periodically” and select the “?” to read the help, to put together the schedule you wish to use. For example, if I want to run every day at 10:30am and 3:30pm then would use the following syntax:



```
30 10,15 * * *
```

Spread load evenly by using 'H 10,15 * * *' rather than '30 10,15 * * *'
Would last have run at Monday, 27 March 2023 at 10:30:17 British Summer Time; would next run at Monday, 27 March 2023 at 15:30:17 British Summer Time.

Once you have saved the change the job will run automatically as per the schedule you created.



Demonstration

Demonstration 2 : Script assistant

Page 12

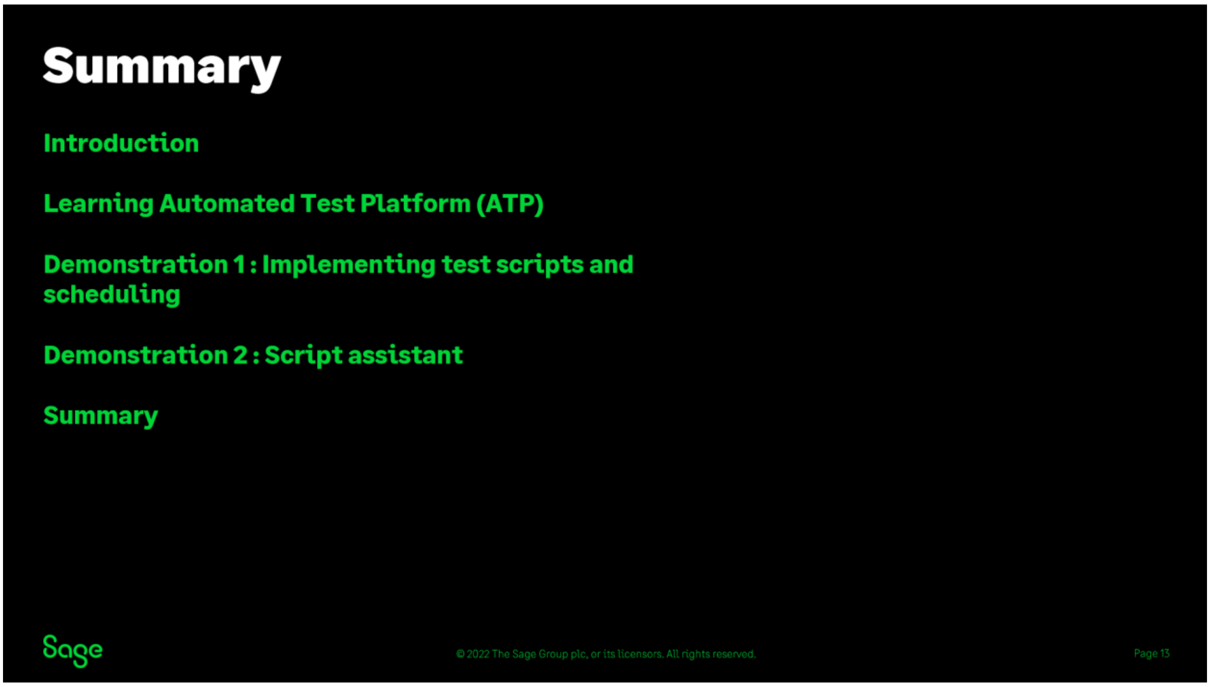
The script assistant files can be requested from Sage if you wish to try them out. I have already installed them for the purposes of this demonstration.

Follow “Documentation_XATP_script assistant_english.docx” document provided with the scripts to install and use.

- Apply Classic patch.
- Import Syracuse menu.
- Login as ADMIN for quickness to test.

To test:

- Launch a classic page, for example Sales Orders.
- Enter values as needed (I will query back SOIGB0130001 and duplicate)
- In the actions, select Utilities then “Currency converter”.
- Click “OK” to take defaults.
- Review the generated script.



Summary

Introduction

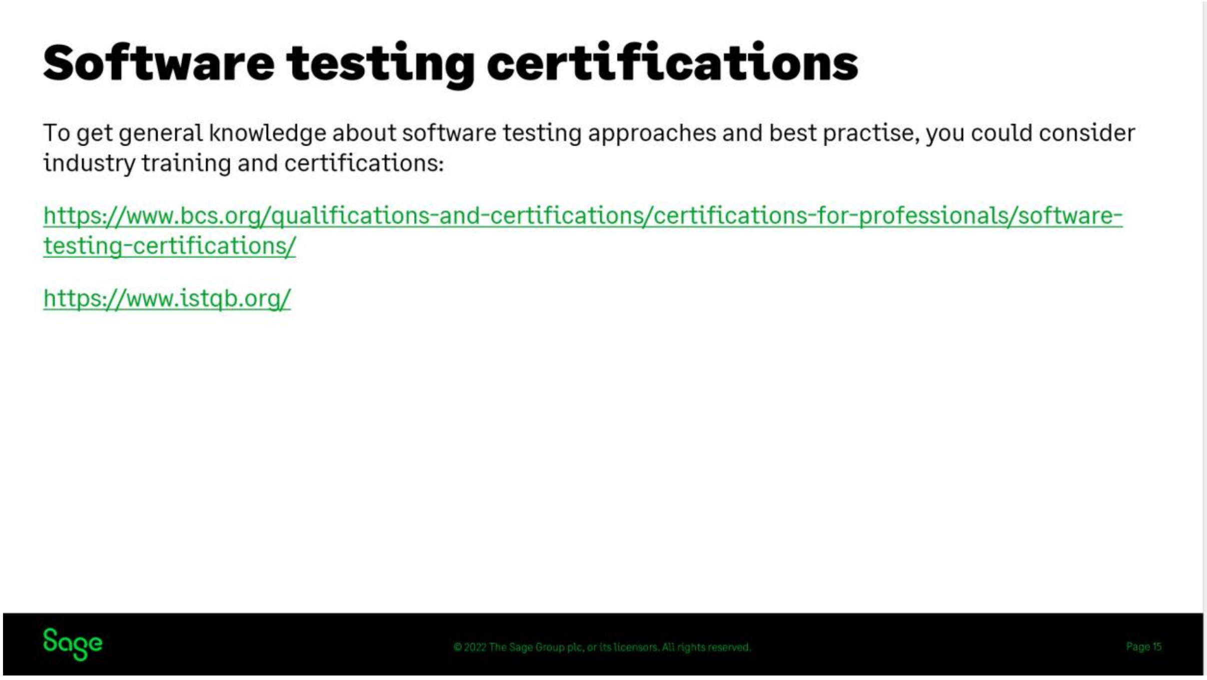
Learning Automated Test Platform (ATP)

Demonstration 1 : Implementing test scripts and scheduling

Demonstration 2 : Script assistant

Summary

Appendices



Software testing certifications

To get general knowledge about software testing approaches and best practise, you could consider industry training and certifications:

https://www.bcs.org/qualifications-and-certifications/certifications-for-professionals/software-testing-certifications/

https://www.istqb.org/

Thank you!