

Participant Training Guide

Version 10.0.0

Certify® Fundamentals



Worksoft, Inc. • 15851 Dallas Parkway, Suite 855 • Addison, TX 75001
www.worksoft.com • 866-836-1773

Worksoft Certify Fundamentals

Participant Training Guide
July 2017

Certify 10.0.0 Participant Training Guide

V10.2 last updated 7/3/17

© Copyright 1999-2017 by Worksoft, Inc.

All rights reserved. All text and figures included in this publication are the exclusive property of Worksoft and may not be copied, reproduced, or used in any way without the express written permission of Worksoft. Information in this document is subject to change without notice and does not represent a commitment on the part of Worksoft.

Worksoft may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents except as expressly provided in any written license agreement from Worksoft, Inc.

All other copyrights are the property of their respective owners.

Trademarks

Worksoft and Worksoft Certify are registered trademark of Worksoft Inc. All other trademarks and trade names mentioned in this guide are the property of their respective owners.

Contents

Overview	15
About This Guide	15
Training at a Glance	15
 Lesson 1	 18
Introduction to Worksoft Platform	18
Overview	18
Objectives	18
Business Process Certification Methodology	20
Certify Module	21
Certify Architecture	23
Lesson Summary	27
 Lesson 2	 29
Defining Processes Overview	29
Overview	29
Objectives	29
Processes Overview	30
Worksoft Web Sample Application – The sample application for this course	30
[Optional] Configuring Windows Internet Explorer	31
[Optional] Configuring Google Chrome	34
Business Process Discovery	41
Using Process Capture to Capture Business Knowledge	41
Worksoft Analyze	49
Creating Business Process Documentation	57
Lesson Summary	61
 Lesson 3	 63
Developing Automation Using Worksoft Certify	63
Overview	63
Objectives	63

Table of Contents

Accessing Worksoft Certify	64
Exploring the Certify Interface	68
General Navigation Flow	69
Worksoft Naming Convention – Best Practices	70
Transaction Code/Component or Unit Test Process	70
SAP Human Resource Design	70
Integrated or End-to-End Process	71
Naming Conventions for Variables	71
Data Layouts.....	72
Recordsets.....	72
Naming Processes.....	77
Creating Process Folders.....	78
Your Sandbox folder	78
Parts of a Process Step	84
Variables Overview	89
Preparing to Use Variables.....	89
Defining Variables in Certify	89
Using Variables in Certify	91
Creating and Modifying Variables	92
Adding Variables to Existing Process Steps	92
Execute Steps with Dialog	97
Layouts Overview	100
Recordsets Overview	101
Recordset Modes.....	102
Defining Layouts in Certify.....	102
Creating a Layout Using Add to Layout	105
Add to Layout Window.....	105
Assets Area	106
Certify LiveTouch Overview	115
Key Functions of Certify LiveTouch.....	116
Certify LiveTouch Menu Options	116
Preparing to Use Certify LiveTouch.....	117
Lesson Summary	143
Lesson 4.....	145

Creating and Executing Integrated Processes.....	145
Overview	145
Objectives	145
Configuring Process Execution	146
Settings Tab.....	147
<i>Run Setting Area</i>	147
<i>Logging Setting Area</i>	147
<i>Data Area</i>	148
<i>Application Version Startup Data Area</i>	148
Attributes Tab	149
Configuration Dialog Box Buttons	149
Execution.....	150
<i>Execution Buttons</i>	150
<i>Execution Dialog Box Tabs</i>	151
Creating an Integrated Process.....	152
Exporting Data.....	162
Importing Data.....	165
Lesson Summary	167
Lesson 5	169
Advanced Executing Processes, Troubleshooting, and Viewing Results	169
Overview	169
Objectives	169
Managing Execution Results	170
Viewing Execution Results	171
Creating a Results Folder	174
Configuring a Results Folder and Advanced Execution.....	176
Editing Process Steps from the Result Viewer	187
Generating Execution Reports	188
Promoting Results to “Result of Record”	192
Resetting Result of Record.....	192
Exporting the Test Process to a .BAT File	194
Lesson Summary	200
Lesson 6	201

Advanced Layouts and Recordsets	201
Overview	201
Objectives	201
Recordset Filters	202
Recordset Filter for the Worksoft Web Sample Application	203
Lesson Summary	218
Lesson 7.....	219
Developing Advanced Processes	219
Overview	219
Objectives	219
Advanced Processes	220
Lesson Summary	241
Lesson 8.....	243
Certify System Objects and Actions	243
Overview	243
Certify System Classes and Actions.....	243
Section 2	255
Lesson 1.....	256
Introduction to Worksoft Certify for SAP	256
Overview	256
Objectives	256
Naming Process Folders	257
Exercise 1A: Configuring Your SAP Environment	258
Certify Capture Overview	260
When do I use Certify Capture versus LiveTouch? What are the differences?	260
Using Certify Capture	261
General steps for using Certify Capture	261
Best Practices for using Certify Capture	262
<i>EXERCISE 1.1 — Configuring Certify Capture and Creating a Folder for Training</i>	<i>263</i>
Lesson Summary	268
Lesson 2.....	269
Defining and Developing Processes for Order to Cash	269
Overview	269

Objectives	269
Defining the SAP OTC Processes	270
Data Needed for the OTC_StandardOrder Process.....	271
Creating the Order to Cash Process Folder.....	272
<i>EXERCISE 2.1 — Creating a Folder for the SAP Order to Cash Processes</i>	<i>272</i>
<i>EXERCISE 2.2 — Creating the OTC_StandardOrder Integrated Process.....</i>	<i>273</i>
Creating Processes Using Certify Capture	274
<i>EXERCISE 2.3 — Use Certify Capture to Create a Process in SAP GUI.....</i>	<i>274</i>
<i>EXERCISE 2.4 — Enhance the VA01_CreateStandardOrder Process</i>	<i>279</i>
<i>EXERCISE 2.5 — Create a Layout and Recordset for VA01_CreateStandardOrder</i>	<i>281</i>
<i>EXERCISE 2.6 — Create the VA01_CreateStandardOrder_C_EnterMaterials Process.....</i>	<i>283</i>
<i>EXERCISE 2.7 — Create the VL01N_CreateOutboundDelivery Process</i>	<i>287</i>
<i>EXERCISE 2.8 — Create the VL02N_PostGoodsIssue Process.....</i>	<i>289</i>
<i>EXERCISE 2.9 — Create the VF01_CreateBillingDocument Process.....</i>	<i>291</i>
<i>EXERCISE 2.10 — Assemble the OTC_StandardOrder Process.....</i>	<i>293</i>
Using Layouts with the OTC_StandardOrder process.....	294
VA01_CreateStandardOrder Layout/Recordset.....	294
VA01_CreateStandardOrder_C_EnterMaterials Layout/Recordset.....	294
Sample completed OTC_StandardOrder process	295
VA01_CreateStandardOrder Sample	295
VA01_CreateStandardOrder_C_EnterMaterials Sample Process.....	297
VL01N_CreateOutboundDelivery Sample Process	298
VL02N_PostGoodsIssue Sample Process.....	302
VF01_CreateBillingDocument Sample Process.....	303
OTC_StandardOrder Sample Process	304
Lesson Summary	305
Lesson 3	307
Defining and Developing Processes for SAP GUI using Certify Data	307
<hr/>	
Overview	307
Objectives	307
Certify Data Overview	308
Certify Data Sections	309
Using Certify Data with the VA01 Transaction.....	311

Table of Contents

<i>EXERCISE 3.1 — Defining the VA01_CreateStandardOrder Process using Certify Data ...</i>	311
Lesson Summary	320
Lesson 4.....	321
Advanced Processes	321
Overview	321
Add Logic to a Process	322
<i>EXERCISE 4.1 — Adding Logic to a Process</i>	322
Handling Dynamic Windows.....	334
<i>EXERCISE 4.2 — Handling Dynamic Windows</i>	337
Lesson 5.....	345
SAP Classes and Actions	345
Overview	345
General Classes and Actions	345
SAP Classes and Actions.....	346
Common Scenarios/Examples	354
Section 3	363
Lesson 1.....	364
Introduction to Worksoft Certify for HTML.....	364
Overview	364
Objectives	364
Configuring Your HTML Environment	365
Windows Internet Explorer Configuration	365
Firefox and Chrome Configuration	365
Naming Process Folders	366
Naming Process Folders (con't).....	367
Lesson Summary	368
Lesson 2.....	369
Learning Screens and Importing Maps.....	369
Overview	369
Objectives	369
Application Maps Overview	370
Creating a Project and Application for Training	371
<i>EXERCISE 2.1 — Create an Application for Training</i>	372

<i>EXERCISE 2.2 — Create a Project for Training</i>	374
<i>EXERCISE 2.3 — Launching the Certify Web Learn Utility and Sample Application</i>	380
<i>EXERCISE 2.4 — Mapping the Login Page and Saving the Map File</i>	384
Creating Search String Anchors	387
<i>EXERCISE 2.5 — Assigning Anchors to Learned HTML Pages</i>	388
<i>EXERCISE 2.6 — Import Map into Certify Application</i>	390
Touch Learn Overview	393
<i>EXERCISE 2.7 — Using the Touch Learn Feature in Certify Web Learn</i>	393
Lesson Summary	396
Lesson 3	397
Advanced Topics with HTML Applications	397
Overview	397
Objectives	397
Best Practices for using the HTML Interface	398
Guidelines	398
Recommended Configuration Steps to Consider	399
<i>Set Page Context Timeout</i>	399
<i>Set Object Context Timeout</i>	399
<i>Set Busy Check</i>	399
<i>Set Cursor Sync</i>	399
<i>Set Scroll Into View</i>	400
<i>Set Step Delay</i>	400
Defining the CW_FieldCheck Processes	401
Creating the CW_FieldCheck Process	402
<i>EXERCISE 3.1 — Creating a folder for the CW_FieldCheck Processes</i>	402
<i>EXERCISE 3.2 — Creating the UTL_CW_Login Process</i>	403
<i>EXERCISE 3.3 — Creating the CW.Buttons Process</i>	408
<i>EXERCISE 3.4 — Continuing with the CW.Buttons Process using Set Attributes action</i>	412
<i>EXERCISE 3.5 — Create the CW.Buttons_C.VerifyRadioButtons Process</i>	417
<i>EXERCISE 3.6 — Create the CW.Editable Process</i>	423
<i>EXERCISE 3.7 — Create the CW_FieldCheck Parent Process</i>	429
Lesson Summary	431
Lesson 4	433
HTML Classes and Actions	433

Table of Contents

Overview	433
General Classes and Actions	433
HTML Classes and Actions	434
Lesson 5.....	443
Resources.....	443
Overview	443
Worksoft Certify Resources.....	444
Access Worksoft Certify Help	444
Certify Log File.....	445
Certify Diagnostic tool.....	445
Worksoft Community and Customer Solutions Site	446
Worksoft Help Portal.....	446
Job Aid: Techniques Used when Defining Processes	447
To create a process:	447
To use LiveTouch to replace current step:	447
To use LiveTouch to insert one or many steps:	448
To execute the entire process:	448
Job Aid: Starting a New Integrated Process	449
Job Aid: Creating Layouts Manually.....	450
Job Aid: Creating Layouts Using Processes	453
Job Aid: Adding Layouts and Recordsets to an Existing Process	457
Course Evaluation	459

Table of Contents

Overview

About This Guide

Welcome to the Worksoft Certify Training. This training guide is yours to keep and personalize for your own needs. The guide contains instructions, exercises, examples, and definitions that you will need in this class. Furthermore, your instructor may provide additional training material throughout the class.

In addition to the content included in this guide, you may also review a series of eLearning tutorials on various topics in Certify. You are encouraged to take notes. Feel free to take notes directly in this guide. By personalizing this guide to your specific needs, you will have your own customized training and reference guide to use after you complete the class. You will find this helpful back on the job whenever you need to refresh your memory on what you have learned during this class.

Training at a Glance

This manual is divided into the following sections:

Section 1

- ▶ **Lesson 1** — **Introduction to Worksoft Platform** explains the Certify Life Cycle approach and methodology, Worksoft Certify Process Capture, and Worksoft Analyze
- ▶ **Lesson 2** — **Defining Processes Overview** covers how to configure Internet Explorer and Google Chrome, outlines Business Process Discovery, and explains how to generate documentation and automation using Worksoft Analyze
- ▶ **Lesson 3** — **Developing Automation using Worksoft Certify** covers processes and how to build them in Certify, using LiveTouch to create steps and introduces the concept of using varying data in data driven automation.
- ▶ **Lesson 4** — **Creating and Executing Integrated Processes** explains how to use basic processes to create and execute an integrated (end-to-end) critical business process.
- ▶ **Lesson 5** — **Advanced Executing Processes, Troubleshooting, and Viewing Results** explores the different methods and options you can use to achieve successful testing results.
- ▶ **Lesson 6** — **Advanced Layouts and Recordsets** explains how to use recordset filters and variables to develop data-driven testing.
- ▶ **Lesson 7** — **Developing Advanced Processes** covers creating advanced processes that can interpret and react to different testing situations. It discusses simple tests versus robust tests, conditional logic, and Certify's tools and methods for adding conditional logic to automated tests.
- ▶ **Lesson 8** — **System Actions and Objects**

Section 2

- ▶ **Lesson 1 — Introduction to Worksoft Certify for SAP GUI** discusses the basics of using Certify to automate business processes in SAP GUI.
- ▶ **Lesson 2 — Defining and Developing Processes for Order to Cash** provides an example set of processes for an Order to Cash (OTC) integrated process.
- ▶ **Lesson 3 — Defining and Developing Processes for SAP GUI with Certify Data** discusses how the Certify Data feature can be used to enhance SAP processes.
- ▶ **Lesson 4 — Advanced Processes** discusses creating SAP processes with advanced functionality.
- ▶ **Lesson 5 — SAP Classes and Actions** lists the Classes and Actions used by Certify with SAP GUI application.

Section 3

- ▶ **Lesson 1 — Introduction to Worksoft Certify for HTML** discusses the basics of using Certify to automate business processes in HTML.
- ▶ **Lesson 2 — Learning Screens and Importing Maps** discusses the process for learning HTML screens and how they are imported into Certify.
- ▶ **Lesson 3 — Advanced Topics with HTML applications** discusses how to work with applications that require login, logout and/or use special features.
- ▶ **Lesson 4 — HTML Classes and Actions** lists the Classes and Actions used by Certify with HTML applications.
- ▶ **Lesson 5 — Resources**

Section 1

Lesson 1

Introduction to Worksoft Platform

Overview

This lesson discusses the Certify Life Cycle approach and methodology, Worksoft Certify Process Capture, and Worksoft Analyze.

Objectives

After completing this lesson, you will be able to:

- Explain the Certify Life Cycle approach.
- Understand Worksoft Certify Process Capture.
- Understand Worksoft Analyze.
- Understand Worksoft's Three-Step Automation.

The Worksoft Platform

1	Discover	2	Automate	3	Run
	Business Analysts/Super Users		Quality Assurance/Automation Professionals		IT Operations/Project Management
	Capture, visualize, document & analyze end-to-end business processes		Build and maintain a reusable portfolio of business process automation		Run automation at industrial scale for projects & ongoing operations
	Worksoft Analyze®		Worksoft Analyze®, Worksoft Certify®		Worksoft Certify® Impact, Execution Manager, Business Process Procedure

Engaging a business' subject matter experts, in a collaborative methodology that requires as little of their time as possible, is critical. Worksoft's collaborative methodology and enabling technology is the only solution on the market that meets this critical requirement.

Business Process Certification Methodology

Business Process Certification is a testing methodology from Worksoft that parallels the software development and deployment cycle but focuses attention on ensuring critical business processes work.

Business Process Certification prioritizes the user tasks that carry the highest business risk. Business risks arise from frequency of use or from the consequences of failure. For example, a certain type of transaction may account for 80% of all daily tasks; therefore, should be classified as high risk. Other types of transactions may be performed less frequently, but if they fail they have dramatic consequences, such as incurred costs.

In terms of the system life cycle, Business Process Certification is:

- The gateway to business operations.
- The point at which the business confirms that it can continue to carry on uninterrupted when the software is introduced into production.
- An event that happens last in the system life cycle. Why is that so important? Because someone in a deadline crunch might skip unit or regression testing but would take the necessary steps to certify his/her critical business processes.

To effectively implement Business Process Certification, remember the following:

- **When determining coverage, prioritize by business risk.** It is not necessary to test every possible error condition, every bug ever detected, or each and every combination, pathway, data type, etc. This distinction is crucial because it implies risk management.
- **Accept the fact that you cannot test every scenario.** The average commercial application would require billions of tests to cover every possible circumstance. Simple economics and physics prevent you from achieving complete coverage.
- **Define your success by what you do accomplish instead of what you don't.** Business Process Certification shifts the focus from what is possible to what is essential. The underlying logic is that if you can't test everything, then you must test the business tasks or processes that are most critical to the business.
- **Rate the identified defects by operational impact.** When defects are found, you should evaluate the impact of the defect on your business operations.
- As additional needs or potential risks are uncovered, remember to keep your priorities straight. If you try to test everything, you will end up testing nothing very well.

Important: The result of the Business Process Certification methodology is a model or profile of production that provides a known level of assurance that your critical business processes continue to operate as expected even after software changes.

Certify Module

Certify is an integrated test repository and automated test execution solution that supports the Business Process Certification methodology. The combination of the Business Process Certification and Certify allows you to parallel the entire product life cycle and capture it in an organized, measurable, and maintainable way.

Certify provides focus on test coverage through a user-friendly, point and click interface that can be used to document and automate test cases. Documenting your test coverage includes capturing and maintaining information about your applications, business processes, and operating cycles in a way that is standard across applications. This means you can perform end-to-end certification of business processes that cross systems. Automating test cases means using what you have documented to certify your applications.

Certify's integrated components not only allow you to document and automate your test cases, but they empower you to track the progress of your certification and measure results throughout the complete product life cycle.

Certify's complete life cycle approach prepares you to execute the certification process as soon as the software is ready for test, instead of waiting to define or develop it. This, in turn, gives you more time and minimizes the impact of the inevitable schedule slip.

Table 1 shows how Certify addresses the required Business Process Certification tasks for each of the phases within the product life cycle.

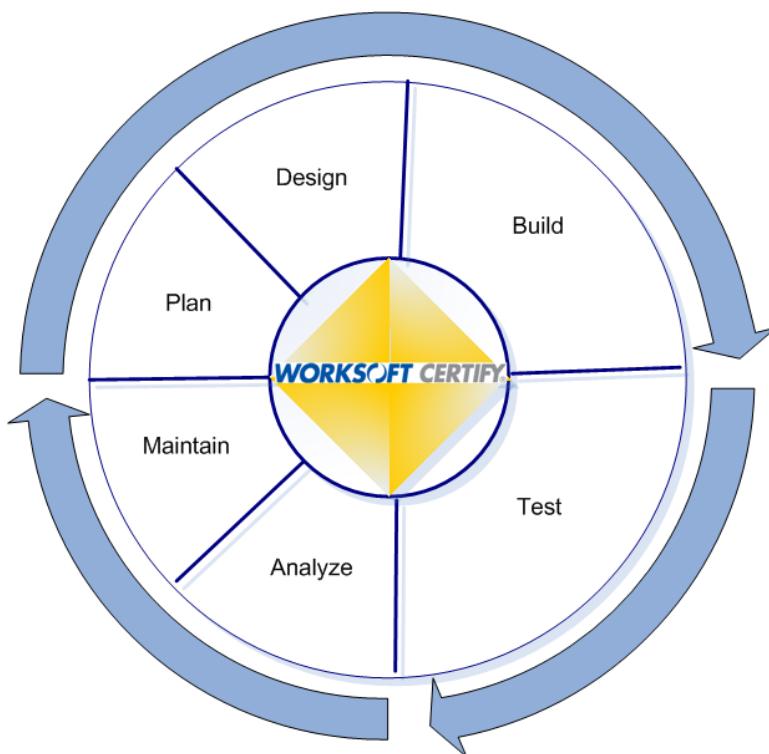
Table 1 — The Life Cycle Approach Using Certify

Life Cycle Phase	Business Process Certification	Certify
Plan	Identify application(s) for test Identify end users Capture business functions	Application(s) and Version(s) Project Requirements
Design	Adopt naming standards Define critical business tasks Define operating cycles	Processes Capture
Build	Define data elements Develop test cases Define data states	Maps Interfaces Variables Processes Recordsets
Test	Execute certification Capture results	Automated or Manual Execution Result Viewer

Life Cycle Phase	Business Process Certification	Certify
Analyze	Analyze results Identify incidents Status analysis	Result Viewer Queries and Reports
Maintain	Receive new application version or feedback from previous application version Identify and implement changes Reconcile application maps	Import Resolve and Processes Interfaces

Figure 1 illustrates Certify's approach to the product life cycle. Certify's approach starts with planning your Business Process Certification, designing your certification process, and building the actual test cases. Certify allows you to document all of the business process components from your plan, design, and build phases and stores them in a test repository. Once you have completed these three phases, you then execute your business processes, analyze the results, and begin maintenance of your certification process.

Figure 1 — Certify Life Cycle Approach



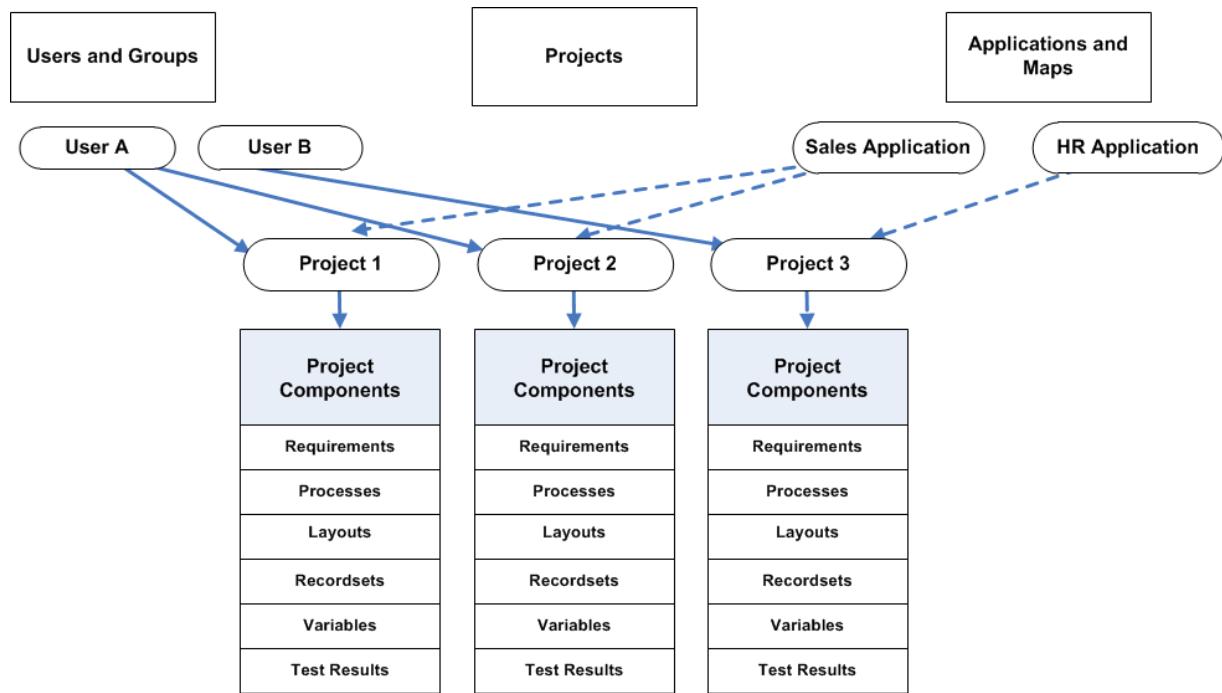
Certify Architecture

Certify maintains an inventory of all of your applications and components. This structure ties all of the components into a consistent hierarchy.

Figure 2 shows how Certify organizes your application and components in a structured hierarchy. A **Project** is a collection of requirements, processes, layouts, recordsets, variables, and test results. Each project is independent, and information is not shared between them.

An **Application** is a set of maps for **windows** and **objects**. Applications can be available in one or more projects. **Users** are given access to Projects. A user may be authorized to read and write everything in one project but can only read aspects of another project.

Figure 2 — Business Process Certification Components



Tip: By following this structure, you can create a life cycle that mirrors the sequence and flow of events and data throughout all of your enterprise systems. Figure 3 shows how the Certify Life Cycle approach interrelates with the functionality of Certify.

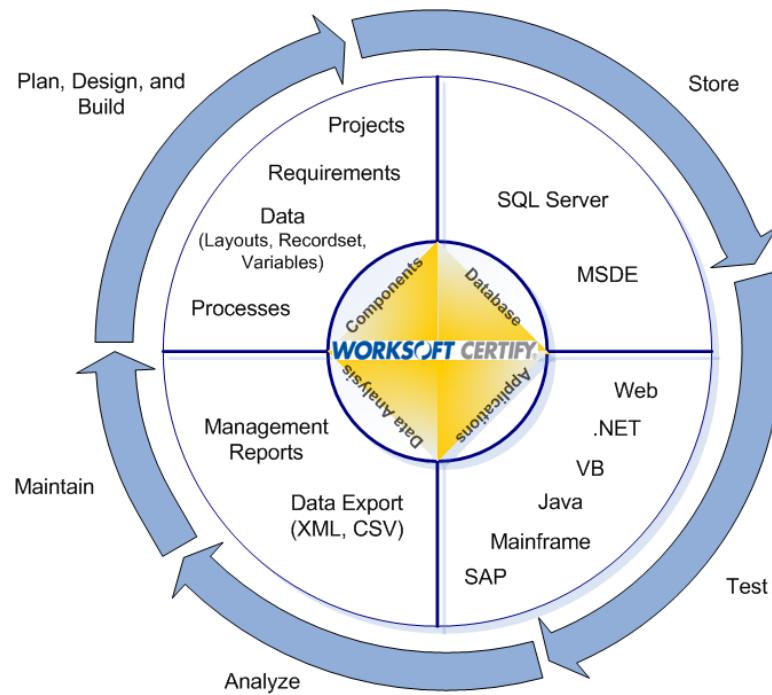
Figure 3 — Certify Life Cycle Approach with Components

Figure 3 illustrates how the Certify components are used in a typical product life cycle.

There are four primary areas in this life cycle:

- **Components** represent what you are planning, designing, and building in Certify.
- **Database** represents the repository where documented components are stored.
- **Applications** represent the type of application being tested.
- **Data Analysis** represents the data for your certification.

Worksoft Certify Process Capture

Worksoft Certify Process Capture™ (Figure 4) is a standalone application that records your actions as you work through your business process. The captured information can be used later to **Analyze** business processes and generate **Automation** and **Visualizations**.

Once Certify Process Capture is started, all interactions with your SAP GUI, Web, and Mobile applications are captured. It is important to note that Capture only creates steps for actions you perform in your application. For example, fields that are modified or check boxes that are checked. As all interactions are retained while Process Capture is in capture mode, be careful when entering confidential information while using Process Capture.

Figure 4: Worksoft Certify Process Capture Icon



Note: Worksoft Process Capture plays an important role in Business Process Discovery. Process Capture not only documents the process but does most of the automation work.

Worksoft Analyze

Worksoft Analyze is a cloud based, easy-to-use automated business process discovery solution to collaborate with business users to more easily discover, visualize, and analyze their critical business processes. By illustrating business process flows as performed by actual business users, Worksoft Analyze creates true visualization of end-to-end business processes.

Worksoft Analyze creates true visualization of your business processes as streamlined activities. These activities are the As-Is business requirements that can be further used for developing automation assets like test scenarios and automated processes. Once installed, Worksoft Process Capture, Analyze, and Worksoft Certify integrate seamlessly to produce the business process documentation and automation assets. While the Certify tool is not a system requirement for using Worksoft Analyze, if you want to generate a Certify process from a particular Process Capture output, you will need to have Worksoft Certify v9.0.2 or a later version installed.

Worksoft's Three-Step Automation

In this Worksoft Basics course, you will follow a simple, three-step process to Discover, Visualize, and Automate end-to-end business processes.

- **Step 1**
 - Using Worksoft Certify Process Capture, capture business users' work **without their intervention** and upload the results to Worksoft Analyze.
- **Step 2**
 - The captured business process activity is **automatically transformed** into a business process flow diagram in Analyze.
- **Step 3**
 - Shows business process variations.
 - Uses Heat Map overlays to highlight error frequency, bottlenecks in the flow, etc.
 - Automatically produces Business Process Procedure documentation on demand, including screenshots.
 - Generates automation of the captured business processes for Certify.

Lesson Summary

You've completed the **Introduction to Worksoft Platform** lesson.

Key points to remember:

- Business Process Certification is a methodology from Worksoft that parallels the software development and deployment cycle but focuses attention on ensuring critical business processes work.
- Worksoft Certify Process Capture™ is a standalone application that records your actions while you work through your business process.
- Process Capture only creates steps for actions you perform in your application.
- Worksoft Analyze is a cloud based, easy-to-use automated business process discovery solution to collaborate with business users to more easily discover, visualize, and analyze critical business processes.

Lesson 2

Defining Processes Overview

Overview

In this lesson, you will configure Internet Explorer and Google Chrome, learn about Business Process Discovery, and learn how to generate documentation and automation using Worksoft Analyze.

Objectives

After completing this lesson, you will be able to:

- Explain what processes are and how they are used.
- Easily navigate and use the sample application for this course.
- Use Process Capture to record your actions while you work with the sample application.
- Use Worksoft Analyze to create Business Process documentation.
- Use Worksoft Analyze to generate automation.

Processes Overview

In Certify, processes are used to document and validate the end-to-end execution of your critical business processes. Processes typically map your existing test cases to the business functions associated with the application under test (AUT).

Processes consist of a series of individual steps where a step performs a specific action against an object. Actions can include entering, pressing keys or buttons, or verifying results like field values, object states, or messages.

Each process performs a discrete function, such as adding an order or finding an existing order. When defining processes, you select the objects and actions to show the sequence of how your application operates.

Worksoft Web Sample Application – The sample application for this course

The sample application for this course is an example of a standard Purchase Order (PO) application that lets a business owner create a PO with a unique PO Number.

The application used to support this functionality is called Worksoft Web Sample Application (WSA). With WSA you can create a unique PO with materials, quantity, and price. You can add up to three materials for one PO. The application also calculates the total amount and generates a document number for each unique PO created.

The techniques learned in this course will be applicable to any interface including SAP, .NET, or Mobile.

Important: We will use either Internet Explorer or Google Chrome to create our integrated process for the Worksoft Web Sample Application. Using Internet Explorer requires the settings modifications outlined in the *Optional Internet Explorer Configuration Exercise*. Using Google Chrome requires the extension addition outlined in the *Optional Chrome Configuration Exercise*.

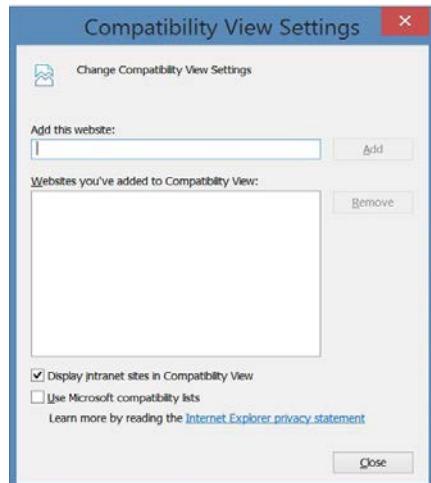
[Optional] Configuring Windows Internet Explorer

If you are using Internet Explorer (IE) to develop and execute processes for the Worksoft Web Sample Application (WSA), there are certain setting adjustments needed in IE for the sample application to display correctly and so that Certify can interact correctly with the application. Your window may look slightly different depending on your version of IE and the security settings enforced by your company. If your windows are significantly different, ask your Instructor or Certify support person to verify the settings are correct.

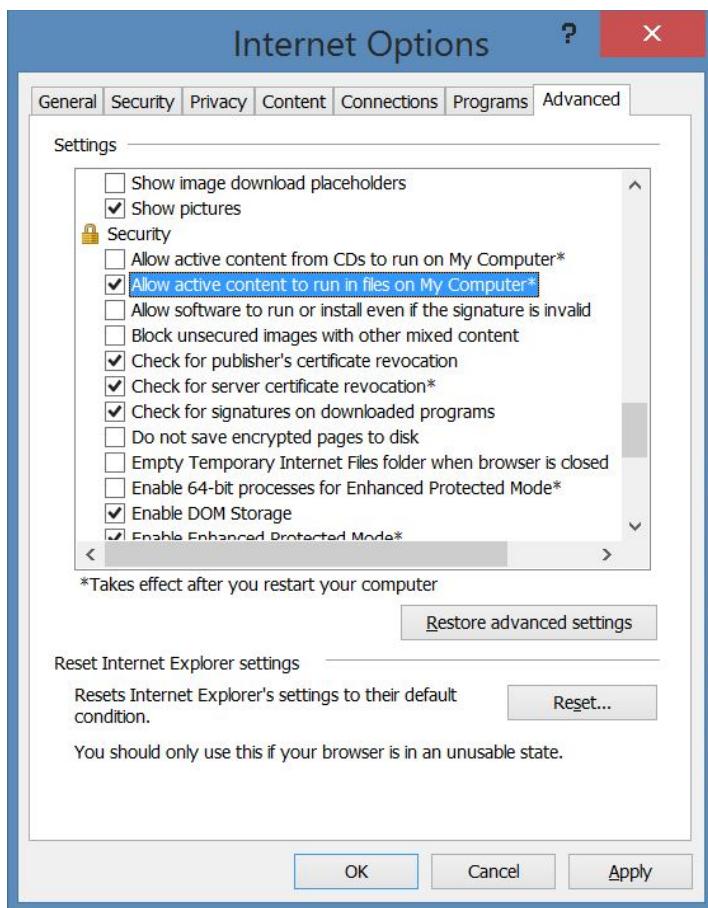
Note: For any additional requirements, refer to the Worksoft Certify Installation Guide.

Step	Action
------	--------

1. Open IE.
2. Go to the **Tools** menu.
3. Select **Compatibility View Settings**.
4. Check the **Display intranet sites in Compatibility View** checkbox.



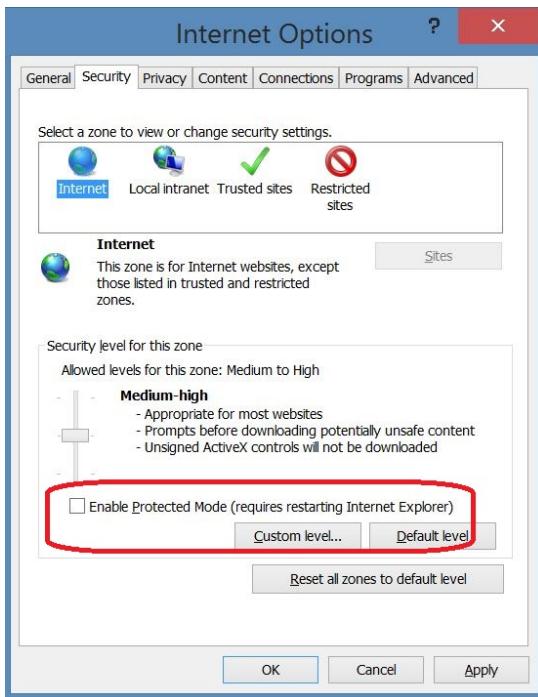
5. Go to the **Tools** menu, and select **Internet Options**.
6. Click the **Advanced** tab.
7. Scroll down to **Security**.
8. Select **Allow active content to run in files on My Computer***.



Enable protected mode should be unchecked for the appropriate zone.

9. Click on the **Security** tab.
10. Select zone (e.g. Internet).

11. Uncheck **Enable protected mode**.



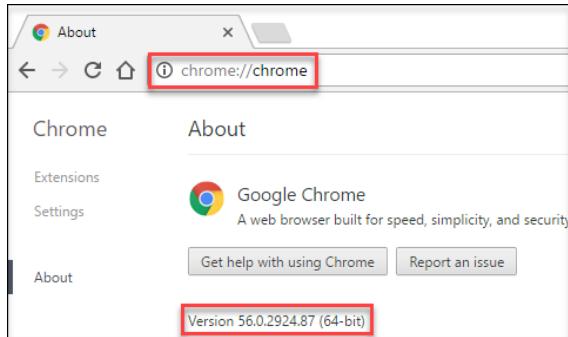
12. Click Trusted sites→Sites.
13. Uncheck **Require server verification for all sites** in this zone.
14. Add a trusted site. Enter **file:///127.0.0.1** in the first field, and then click Add.
15. Click Apply.
16. Click OK.
17. Verify the Zoom percentage is set to 100%. This must be set when creating steps but will be handled automatically during execution.

[Optional] Configuring Google Chrome

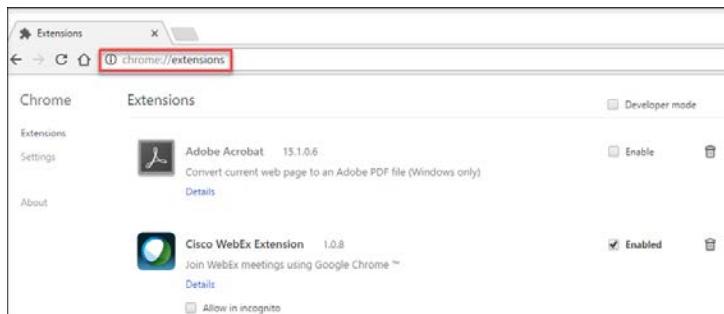
If you are using Google Chrome to develop and execute processes for the Worksoft Web Sample Application (WSA), a Worksoft extension file must be added to Chrome. This extension allows Certify to interact correctly with an application opened in Chrome. Your window may look slightly different depending on your version of Chrome and the security settings enforced by your company. If your windows are significantly different, ask your Instructor or Certify support person to verify the settings are correct.

Step	Action
------	--------

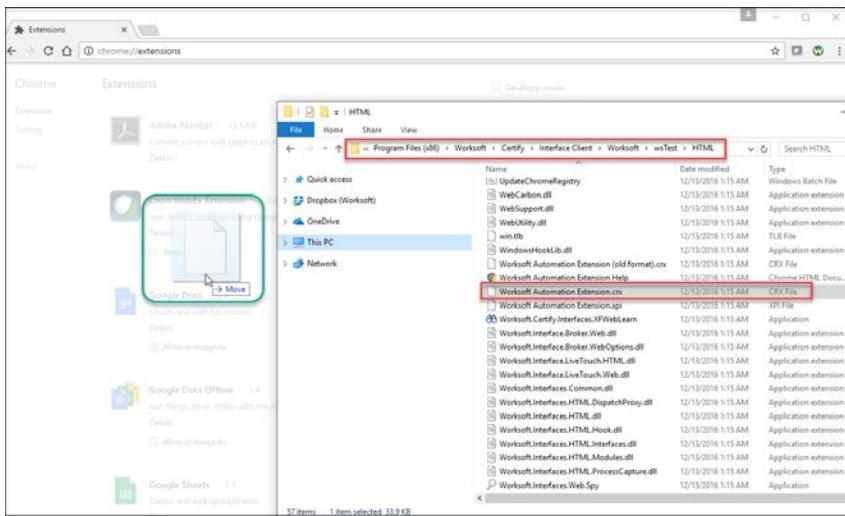
1. Open Google Chrome.
2. Your version of Chrome determines the extension file needed. Your Chrome version can be found by entering **chrome://chrome** into the address bar and pressing enter.



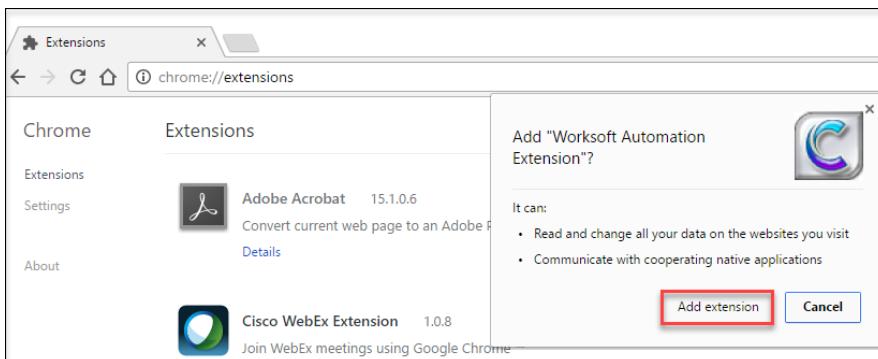
3. For Chrome versions 28+, you will need the Worksoft Automation Extension.crx file. This file is located in C:\Program Files (x86)\Worksoft\Certify\Interface Client\Worksoft\wsTest\HTML.
4. For Chrome versions earlier than 28, you will need the Worksoft Automation Extension (old format).crx file. This file is located in C:\Program Files (x86)\Worksoft\Certify\Interface Client\Worksoft\wsTest\HTML.
5. Once you've located the file, in the Chrome address bar, type **chrome://extensions**, and press Enter.



6. Drag and drop the extension file into the Chrome Extensions page.



7. Chrome will request your permission to add the extension. Click to **Add extension**.



8. After adding the extension, verify it is **Enabled**. Select the **Allow access to file URLs** checkbox.

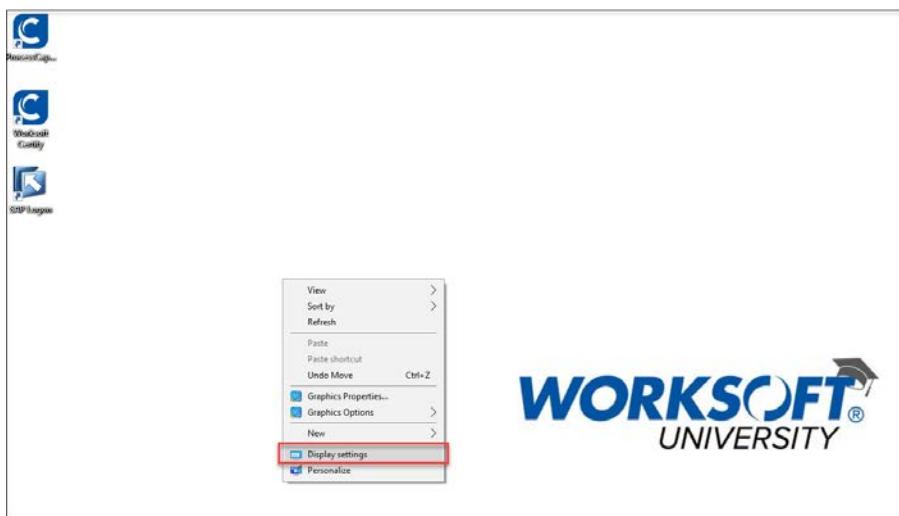


EXERCISE 2.A — Configuring Desktop Resolution

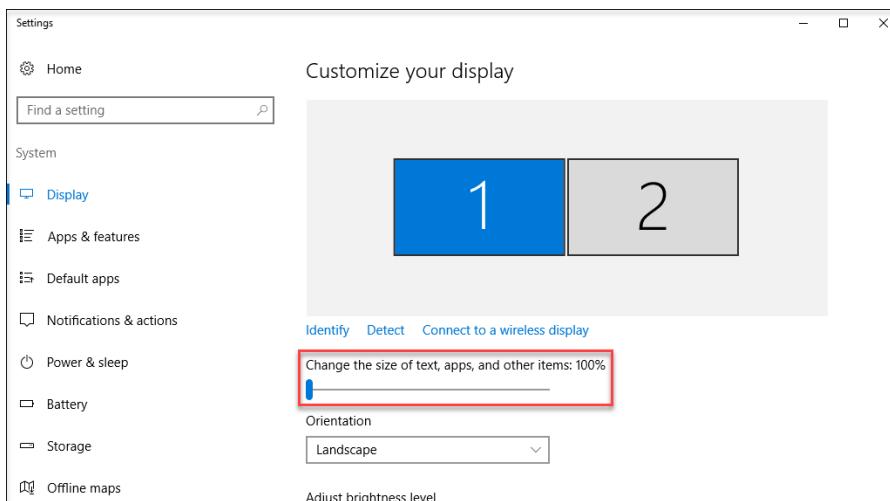
In this exercise, you will set your desktop resolution before using Process Capture. In order to capture fields and screens correctly, your desktop resolution must be set to 100%.

Step	Action
------	--------

1. Right-click on your desktop, and select **Display settings**.



2. Verify the size of text, apps, and other items is set to: 100%. If it isn't, change to 100%.



In this series of exercises, you will become familiar with the Worksoft Web Sample Application by creating, editing, and deleting a purchase order.

EXERCISE 2.1 — Creating a Purchase Order

In this exercise, you will become familiar with the sample application by accessing the Worksoft Web Sample Application and creating a purchase order.

Step	Action
------	--------

1. Click **Start** and select All Programs → Worksoft → Sample Applications.
2. Select **Worksoft Web Sample Application**.

The Worksoft Web Sample Application appears.

The screenshot shows a web browser window titled "WebAccountManager" with the sub-page "WORKSOFT Web Sample Application". The page contains fields for "PO Number", "Name", "Ship to:", and "Bill to:". Below these is a table for entering purchase order items, with columns for "Material", "Quantity", "Price", and "Amount". The table has three rows, with the third row highlighted in green. At the bottom of the form are "Save" and "Cancel" buttons, and a summary row with columns for "Action", "DocNumber", "PONumber", "Name", "ShipTo", and "Amount".

Important: ActiveX controls must be enabled to run the sample application. If the following information bar appears, right-click and select "Allow Blocked Content" to turn the feature on.

 To help protect your security, Internet Explorer has restricted this webpage from running scripts or ActiveX controls that could access your computer. Click here.

3. To create a purchase order, type in the field values as follows:

Field	Value
PO Number	1234
Name	Mary Wilson
Ship to	100 West Fifth Street
Bill to	1444 North Ave
Material	Smart TV
Quantity	10
Price	780

4. Click **Save**.

EXERCISE 2.2 — Explore the Sample Application and Identify Supporting Processes

The sample application has several options and business processes. This exercise will be done with the instructor. Before automating, we need to explore the sample application. We can't automate the application if we don't know how it works manually. Then we will identify the supporting processes needed.

To launch the sample application:

1. Click **Start** and select All Programs → Worksoft → Sample Applications.
2. Select the **Worksoft Web Sample Application**.

Criteria for this business process:

- The PO Number must be unique.
- A valid Name, Shipping Address, and Billing Address should be provided.
- Each PO can have up to three Material Items.
- The Amount is calculated as the product of the Price and Quantity ordered.
- The PO can be edited.
- The PO can be deleted.
- Your instructor will walk you through a whiteboard exercise to plan the processes and data used in this end-to-end business process.

Table 1 shows the critical business process and sub-processes needed to satisfy the test requirements for the Worksoft Web Sample Application. Depending on your test design, verification steps do not need to be in a separate process.

Table 1 — Identified Processes for Worksoft Web Sample Application

Critical Business Process	Sub-Processes	Description
WSA_CreateandVerify	WSA_Input	Creates a PO.
	WSA_Input_C_Materials	A reusable child process that enters multiple materials for a PO.
	WSA_SelectandVerify	A reusable process that selects and verifies the PO and stores the document number for future use.
	UTL_WSA_Launch	Launches the Worksoft Web Sample Application.
	UTL_WSA_Close	Closes the Worksoft Web Sample Application.
WSA_EditandDelete	WSA_EditQuantityandPrice	Selects a purchase order and edits a specific material's quantity and price.
	UTL_WSA_ExportRecordset	Exports the WSA_Input recordset to a .txt file.
	WSA_UpdateRecordset	Adds each purchase order's amount to each record imported to the WSA_UpdateRecordset process and exports the data to a .txt file.
	WSA_DeletePO_Amount	Identifies and deletes any Purchase Order with an amount less than \$5000. This will limit purchase orders in the WSA PO table by the amount criterion.
	WSA_DeletePO_Cleanup	Identifies and deletes all purchase orders in the PO table.

Once you have defined all the processes for your application, you should go back to your list of requirements and create links between the requirements and the defined processes. Once the processes are executed, you can run a Requirement Coverage report to validate whether the requirement was met. These tasks are covered in later lessons.

Business Process Discovery

Worksoft Process Capture™ plays an important role in the Business Process Discovery. Business users leverage Process Capture to discover end-to-end business processes across all departments. It provides the flexibility to capture all the business process information in a non-interventional mode, making it easier for business users to collaborate with IT without the need for interview cycles or detailed knowledge transfer meetings.

Using Process Capture to Capture Business Knowledge

Process Capture is a standalone application that records your actions while you work through your business process. Processes captured will be automatically sent to Worksoft Analyze. It is also possible to save them as a file in a network directory to be imported into Certify without being sent to Analyze. The captured information can be later used to **Analyze** business processes and generate **Automation** and **Visualizations**.

Once Process Capture is started, all interactions with your SAP GUI, Web, and Mobile applications are captured. It is important to note that Process Capture only creates steps for actions you perform in your processes. For example, fields that are modified or check boxes that are checked. As all interactions are retained while Process Capture is in Capture mode, be careful when entering confidential information while using Worksoft Capture.

Best Practices for Process Capture

Know the business process flow and data before you start Process Capture.

Capture as many business processes as you deem “critical to quality”.

Capture the business process as real users perform the activity.

Use the Comment, Screenshot, and Verify features of Process Capture to capture detailed and comprehensive process steps.

Small units are best – single transactions that can be combined with other processes. Don’t capture for an hour!

Capture of SAP activities are named automatically based on the transaction code.

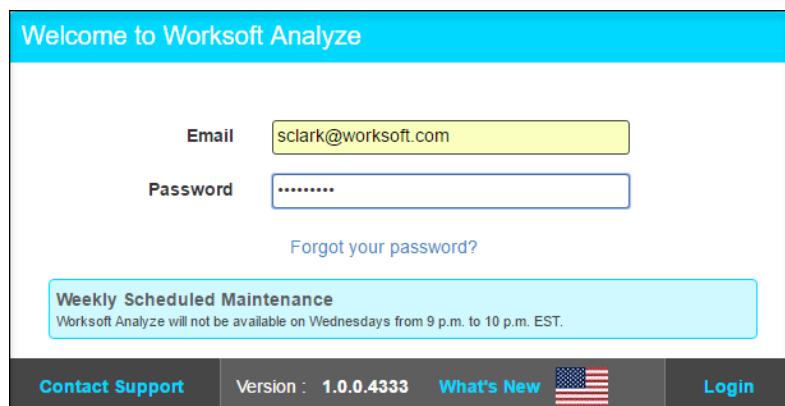
While capturing non-SAP / Web related activities, create a Name that describes the Action being completed against the Objects. For example, if the screen of the Web application is for creating an order, use “Create Order” as the Name. Describe the process outcome and not just the action itself.

EXERCISE 2.3 — Download Process Capture

In this exercise, you will download Certify Process Capture from Worksoft Analyze. If you have already downloaded Certify Process Capture, skip to the next exercise.

Step	Action
------	--------

1. Login to Worksoft Analyze by using URL <https://analyze.worksoft.com/analyze/>
2. Enter your Worksoft Analyze credentials, and click **Login**.



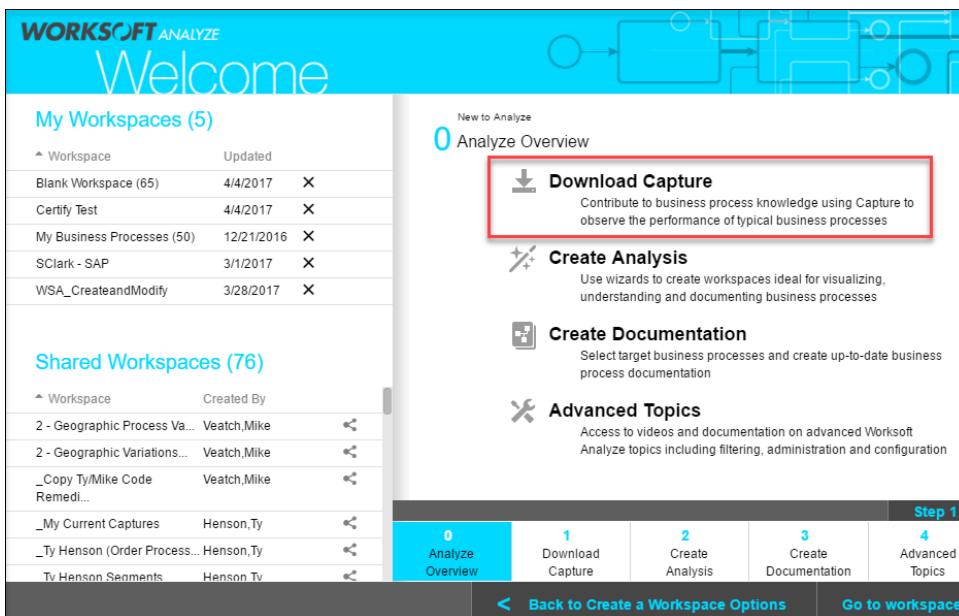
3. Click **New to Analyze? Start Here!**.

New to Analyze? Start Here!
Shortcuts to success with Worksoft Analyze

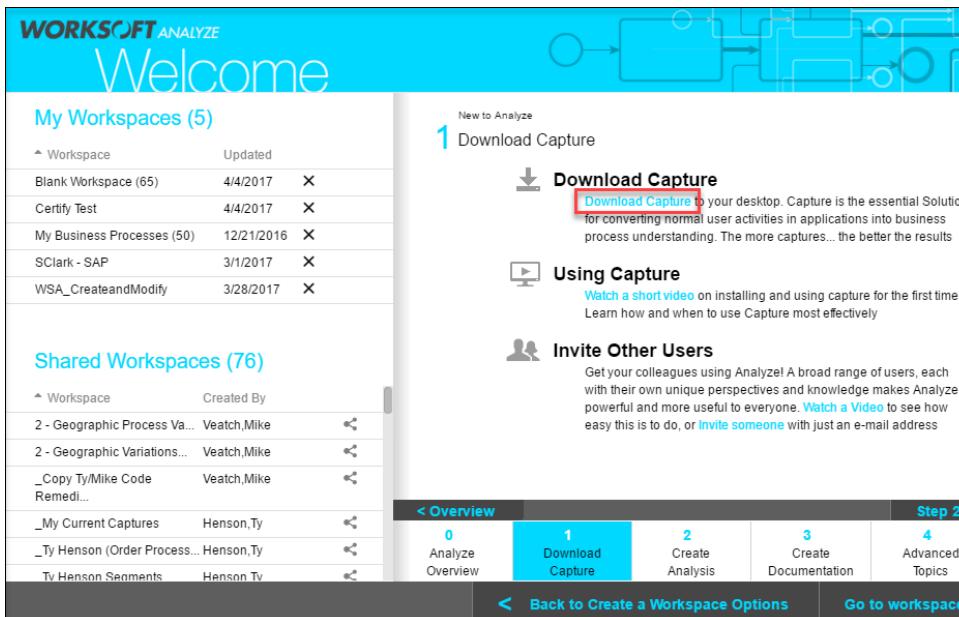
Blank Workspace
Create a new workspace without using a template

Create a New Workspace
Use a helpful wizard to identify business processes of interest

4. Select **Download Capture**.



- Click the **Download Capture** link.



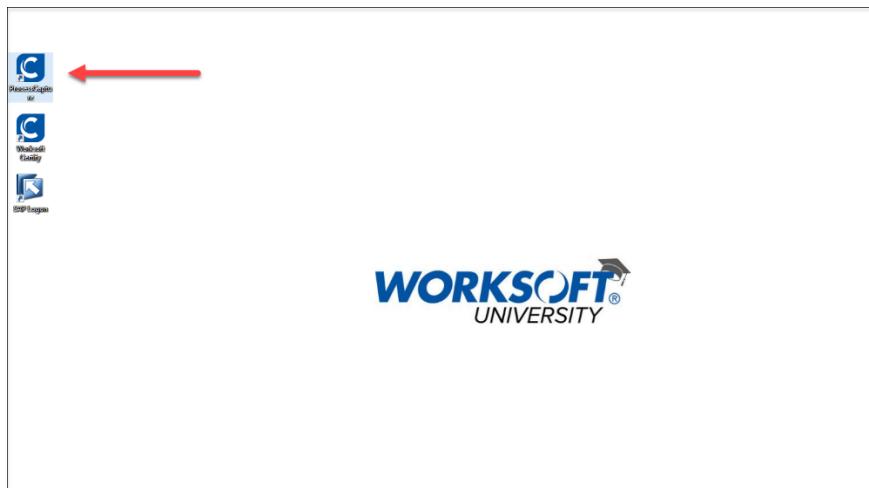
- Select and run the **WorksoftProcessCapture.exe** file to install Process Capture.
- You should see a **Process Capture shortcut** on your Desktop.
- Exit **Worksoft Analyze**.

EXERCISE 2.4 — Use Process Capture to Record Purchase Order Creation

In this exercise, you will configure Certify Process Capture and capture the process to create a Purchase Order in the Worksoft Web Sample Application.

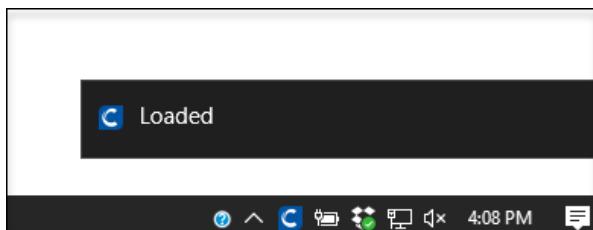
Step	Action
------	--------

1. Launch Process Capture by double-clicking the Process Capture Icon on your Desktop.

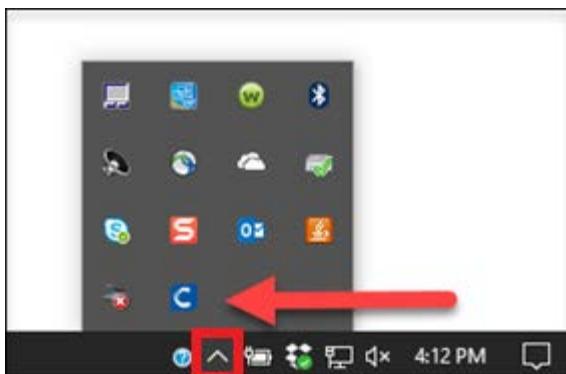


Important: The Worksoft Web Sample Application must be open before using Process Capture. For best results, make sure no other web applications are running.

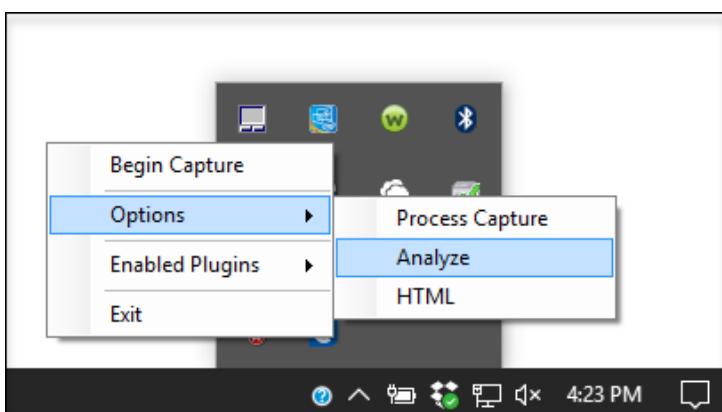
2. Once Process Capture has started, you will see a "Loaded" message, and then Capture will minimize to the taskbar. You do not need to load Process Capture again if it is in the taskbar.



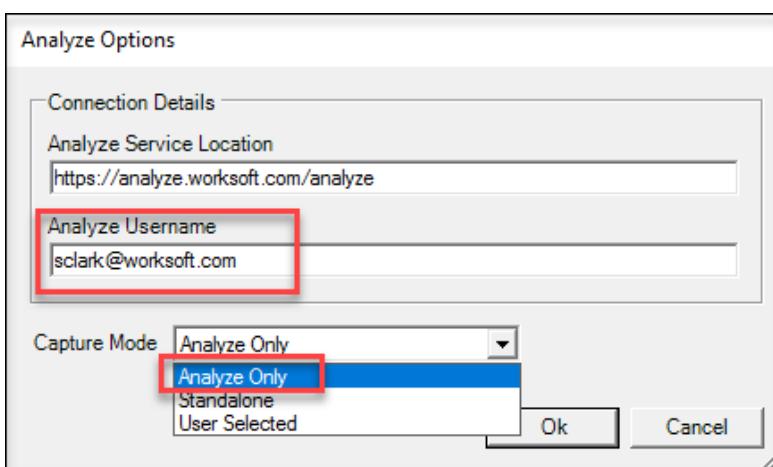
3. Click the Hidden Icons arrow in the taskbar to see the Process Capture icon.



4. Right-click the Process Capture icon in the taskbar. From **Options**, select **Analyze**.



5. An Analyze Options dialogue box will appear. In the **Analyze Username** field, enter your Analyze user name.
6. Open the **Capture Mode** drop-down, and select **Analyze Only**.



Note: This will save all the Process Capture files to Analyze in a virtual workspace.

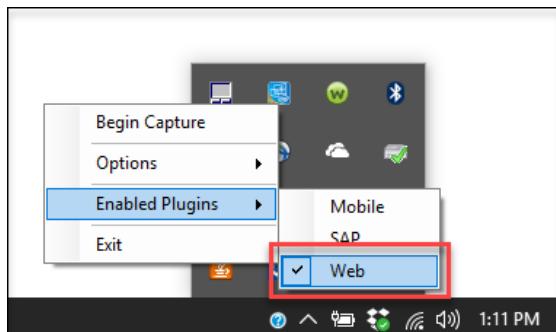
7. Click **OK**.
8. Once again, right-click the Process Capture icon in the taskbar, and select **Enabled Plugins**. From the dropdown, select the **Web** option as shown below.

Note: Capture Mode Options:

Standalone if you always want to save to a folder.

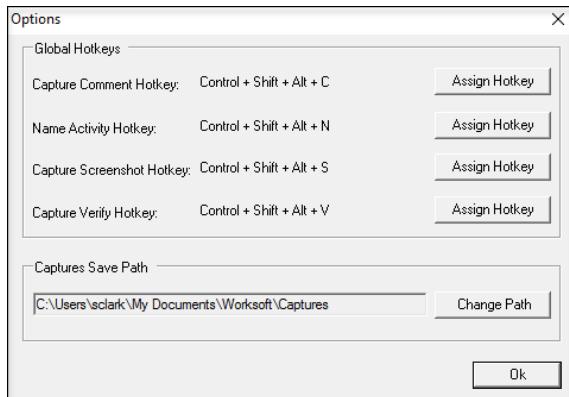
Analyze Only if you always want to save to Analyze.

User Selected if you want to select either Standalone or Analyze each time you save a capture.



Important: Process Capture can recognize Mobile, SAP and Web applications. It is best to only enable the plugins you are sure you want to use.

9. Process Capture uses Global Hotkeys to use special features while capturing a process. To view the hotkeys and storage location of your captures, right-click the Worksoft Process Capture icon in the taskbar. Select **Options** and **Process Capture**.



Important: These hotkeys must be unique. Don't reuse hotkeys like Ctrl+C or you won't be able to copy anymore!

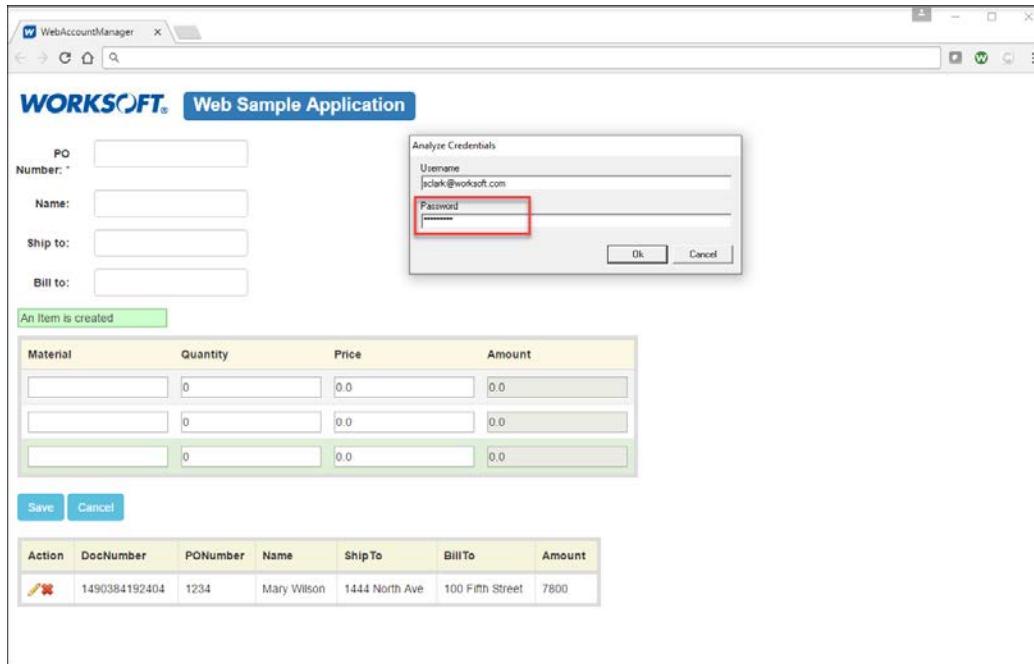
10. To begin capturing your Worksoft WSA purchase order creation process, right-click the Process Capture icon in the taskbar, and select **Begin Capture**.
11. First, we need to name our activity. To name the activity, press **Ctrl+Shift+Alt+N**.
12. Enter **WSA_Input** in the Name Activity dialogue box.

13. Click **OK**.
14. Enter **1234** in the **PO Number** field, and then click the **Name** field.
15. Enter the following values in the fields listed below:

Field	Value
#2 – Name	<i>Mary Wilson</i>
#3 – Ship to	1444 North Ave
#4 – Bill to	100 Fifth Street
#5 – Material	Smart TV
#6 – Quantity	10
#7 – Price	780

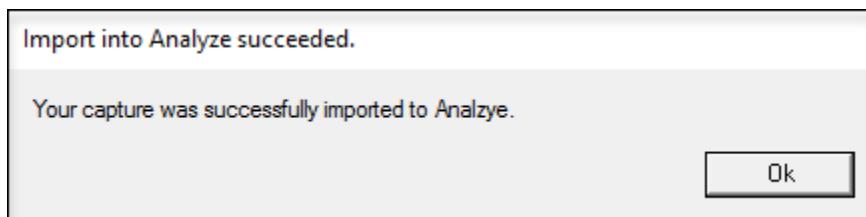
16. Press  in the Web Sample Application.

17. Right-click the Process Capture icon in the taskbar, and select **End Capture**.



18. Enter your Worksoft Analyze credentials provided by your instructor/Worksoft administrator.

19. Click **OK**.



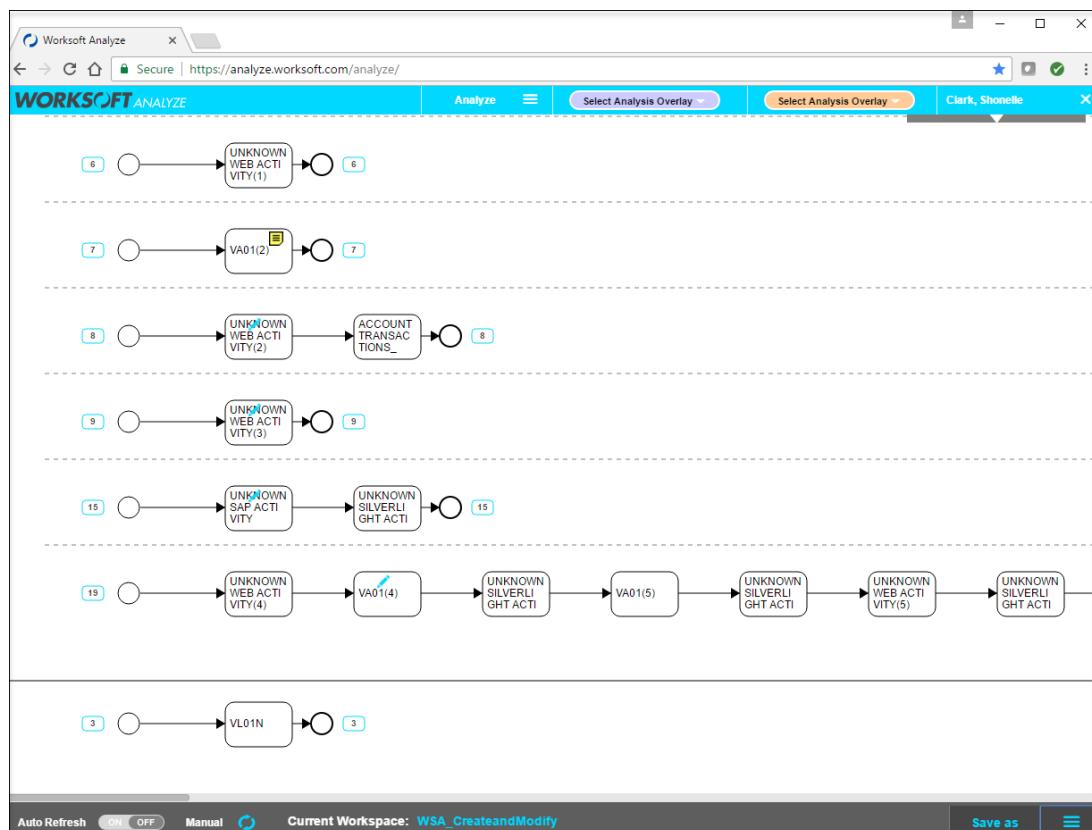
20. Click **OK**.

Worksoft Analyze

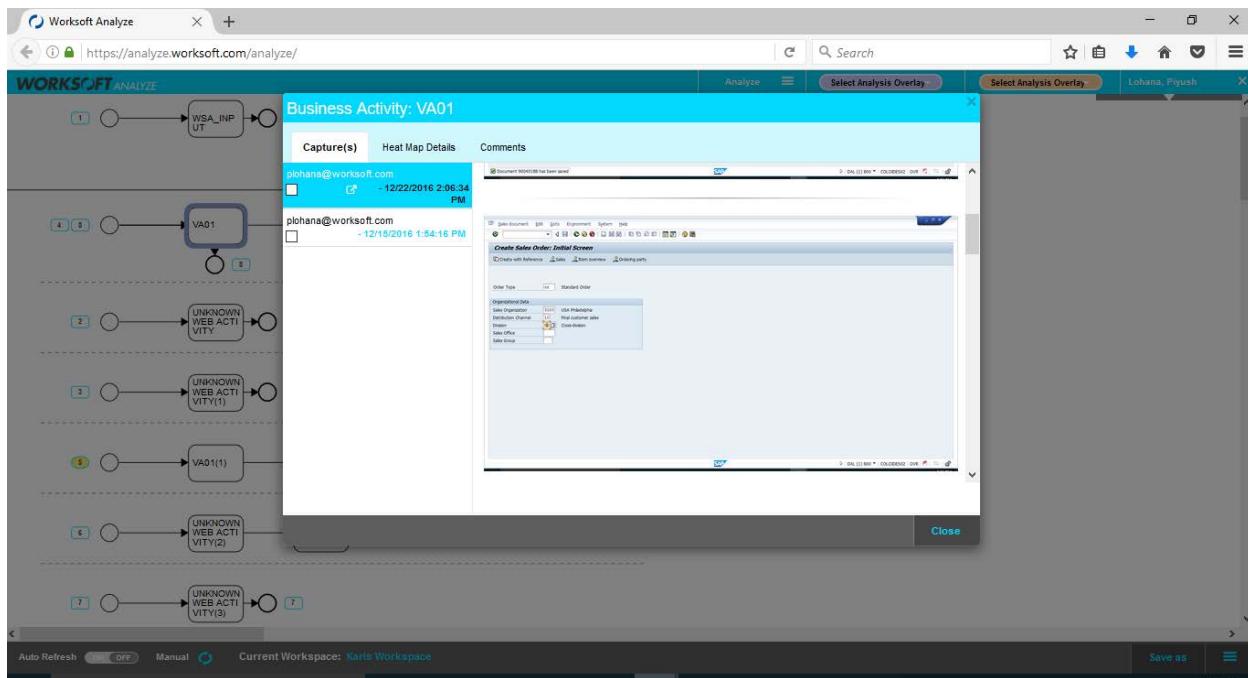
Worksoft Analyze is a cloud based, easy-to-use automated business process discovery solution to collaborate with business users to more easily discover, visualize, and analyze their critical business processes. By illustrating business process flows as performed by actual business users, Worksoft Analyze creates true visualization of end-to-end business processes.

Worksoft Analyze creates true visualization of your business processes as streamlined activities. These activities are the As-Is business requirements that can be further used for developing assets like test scenarios and automated tests. Once installed, Process Capture, Analyze, and Certify integrate seamlessly to produce the test documentation and test automation assets. While the Certify tool is not a system requirement for using Analyze, if you want to generate a Certify process from a particular Process Capture output, you will need to have a version of Worksoft Certify v9.0.2 or later installed.

When using Analyze, all the captured business processes can be visualized as independent workflow activities. The logged in user of Analyze will be able to visualize the captured activities of all the users. As shown in the picture below, there can be as many activities as required. Analyze has the intelligence to filter and consolidate the duplicates and uniquely identify the variations of a similar activity across silos and divisions.



Right-clicking on any of the activities will present you with an option to visualize the screenshot captures associated with that activity. This information will help you understand the data entered as part of the Process Capture activity.



After the confirmation of all the captured activities are Visualized, the responsible QA Manager/Lead will coordinate with the respective stake holders (Business Owners, Subject Matter Experts) of the captured activities to meet and perform the next activity – the Analysis.

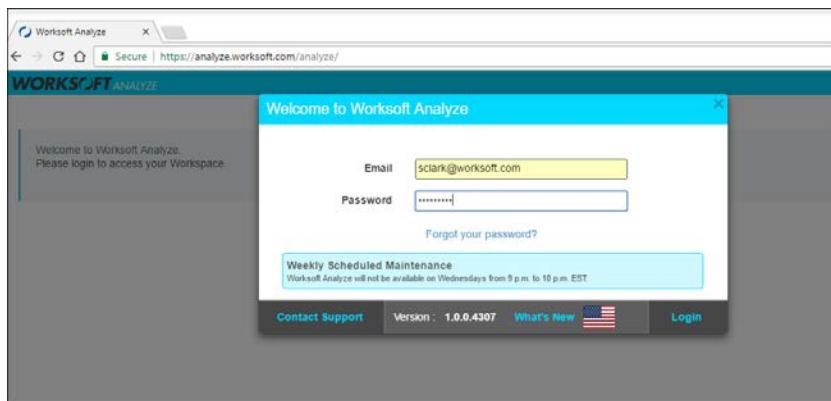
This is a collaborative discussion where the subject matter experts discover the sequence of segmented activities as cross-functional, end-to-end business processes. The positive output of this activity plays a pivotal role in the efficient use of Analyze.

EXERCISE 2.5 – Analyzing WSA_Input

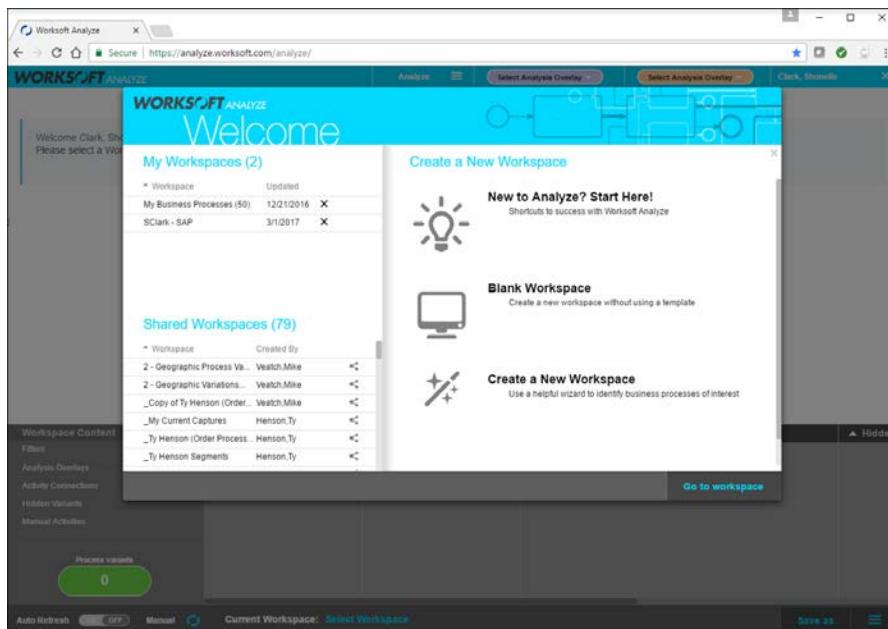
In this exercise, you will become familiar with Worksoft Analyze by reviewing the WSA_Input capture generated in the previous exercise.

Step	Action
------	--------

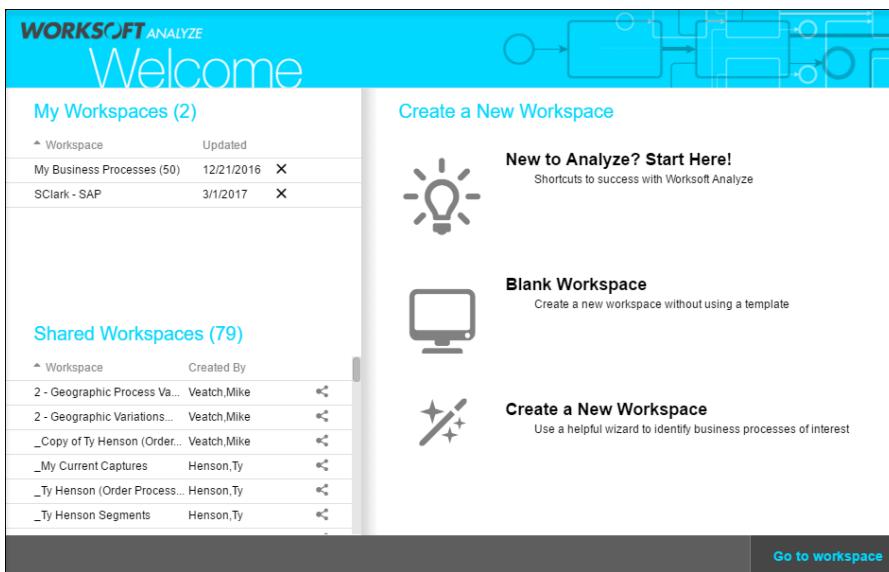
1. Login to Worksoft Analyze by using URL <https://analyze.worksoft.com/analyze/>
2. Enter your Worksoft Analyze credentials, and click **Login**.



3. The Analyze Welcome screen is displayed, as shown below.



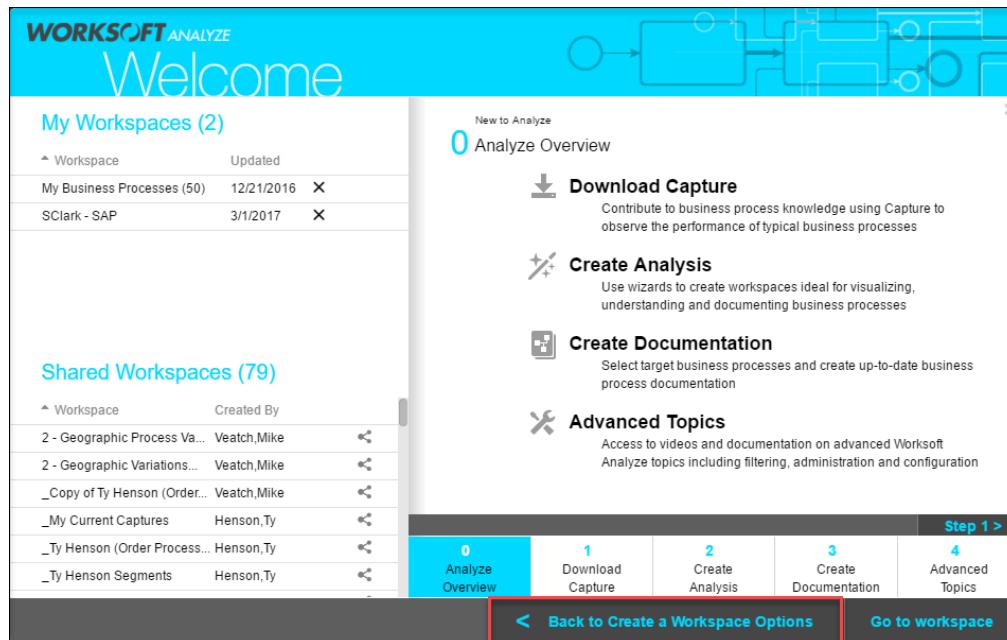
4. From the Analyze Welcome Screen, click the **New to Analyze? Start Here!** option.



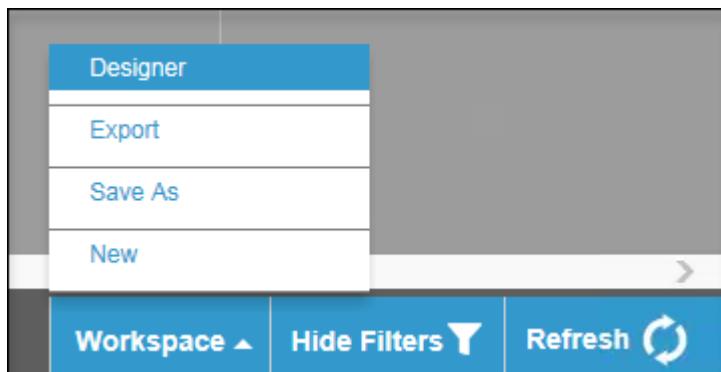
The “New to Analyze” option allows you to

- Download the Latest Version of Worksoft Process Capture
- Create Analysis
- Create Documentation
- Access Videos and Documents on Advanced Worksoft Analyze Topics

5. Click on **Back to Create a Workspace Options**.



6. From the Welcome Screen, click on the **Blank Workspace** option.
7. A Blank Workspace is created. Click the **Workspace** menu in the lower, right-hand corner of the screen, and select the **Designer** option.



The workspace name is *typically* based on the business process and uses the following format:

Format: ApplicationArea_Description1_Description2_Description3

Example: OTC_ShipBill_ConstantQuote

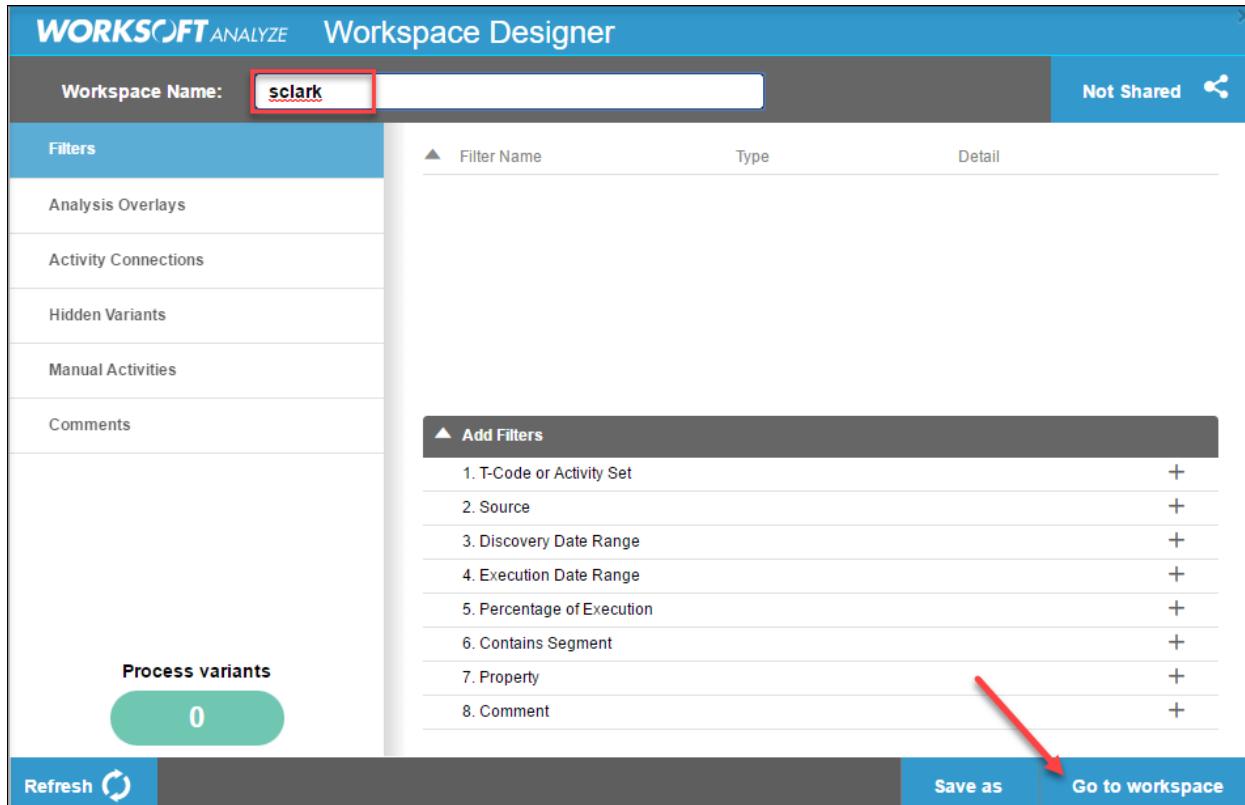
CRM_CustomOrder_ProcessThroughSAP

WebOrder_FromBrowser_ToSAP_ToMainframe

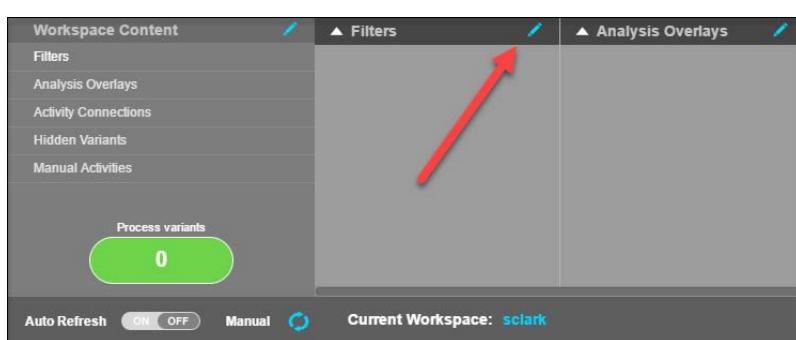
Note: Each workspace view is unique because you can add filters to narrow the focus of the view. These filters help you focus on business processes or analysis of interest. You should create a different workspace for each unique way of looking at business process knowledge. For example, you may have workspaces named: Sales Processes, Inventory, or Quarter Close.

8. Because we will be completing the same exercises in one tenant space, we will use the combination of first initial and last name. For example, **sclark**.

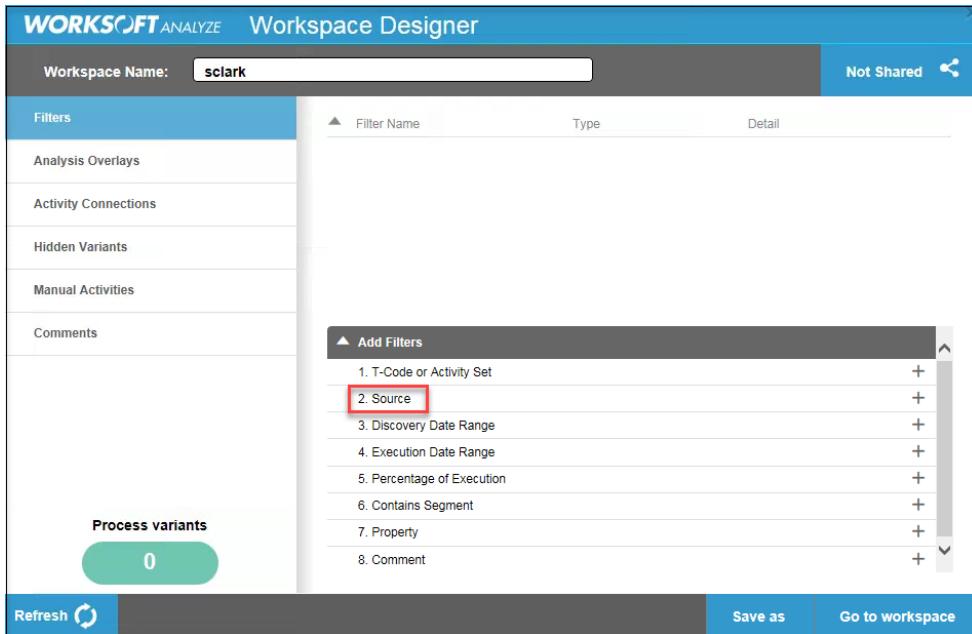
9. Click the **Go to Workspace** button.



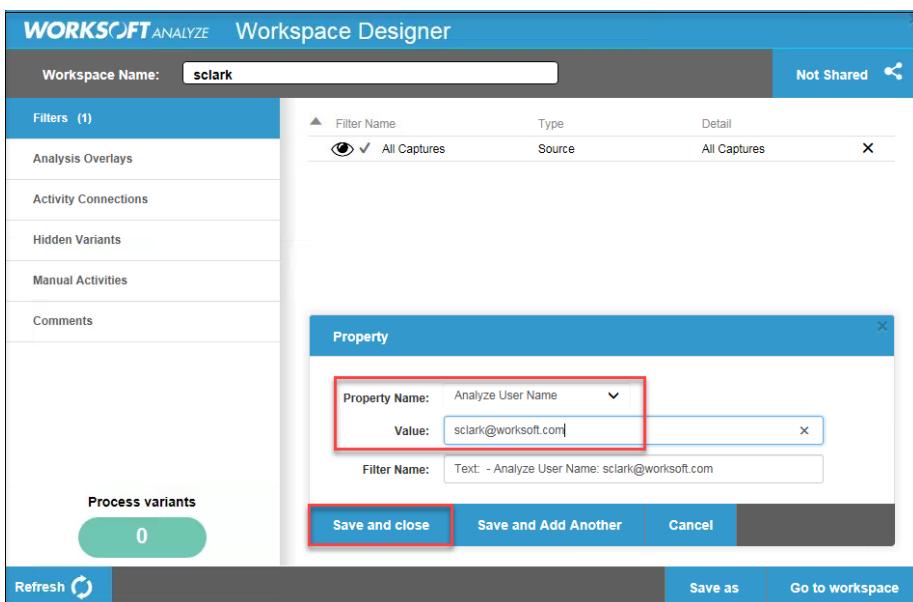
10. In the newly created workspace, we will define and use filters to bring in the required Process Capture activities.
11. Click on the Filters edit option (the pencil button).



12. Select **Source** from the **Add Filters** menu.



13. Select Data Source as **All Captures** from the drop-down menu, and click **Save and Close**.
14. Select **Property** from the **Add Filters** menu.
15. Select **Analyze Username** from the property dropdown list.
16. Enter your Analyze username, and click **Save and Close**.

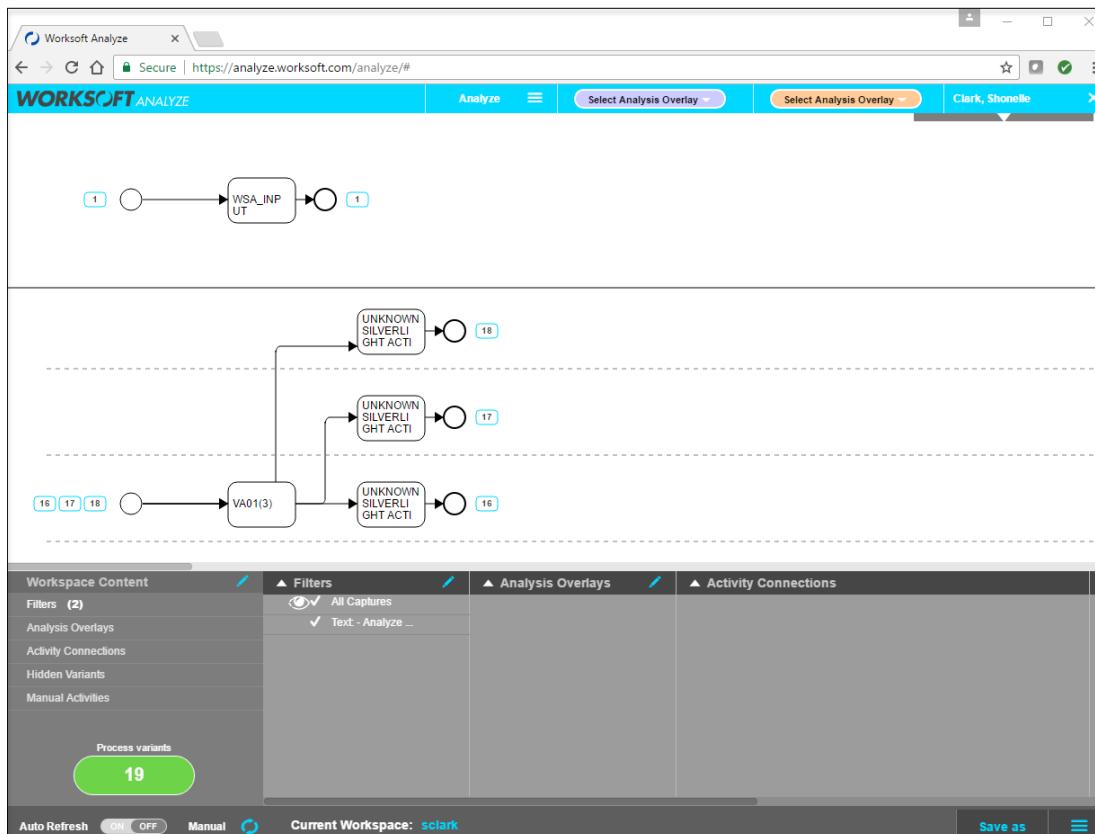


17. Click the **Go to Workspace** button.



18. Press the refresh button.

19. This should load all the Process Capture activities associated with your user name.



Important: In this screenshot, there are 18 variants associated with this Analyze User ID. What is shown will vary depending on the user's previous activities with Analyze. You can select different filters, depending on your requirements. Your instructor will show you how to select the Discover Date filter to filter variants based on dates.

Creating Business Process Documentation

One of the valuable byproducts of using Analyze is the ability to generate business process documentation, which can be used as training documents and as test scenario wireframes. These documents will be valuable resources to rely on to follow and use for manual testing efforts when required.

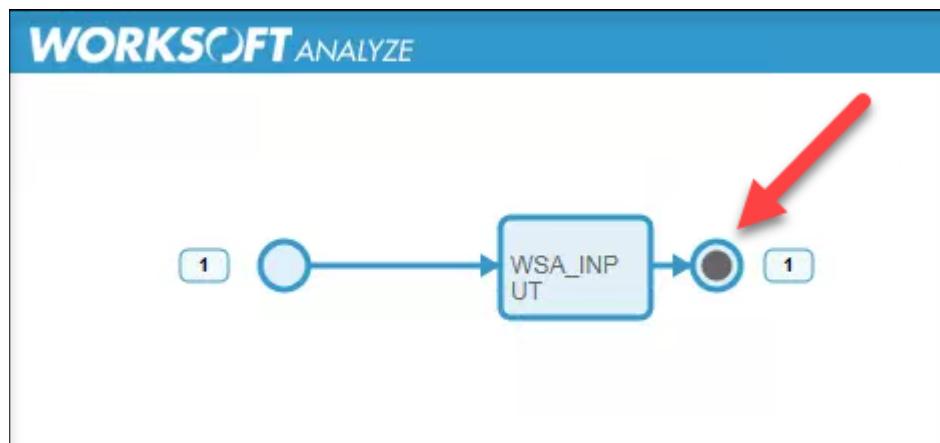
In the next exercise, we will generate business process documentation for the WSA_Input process.

EXERCISE 2.6 — Creating Business Process Documentation in Analyze

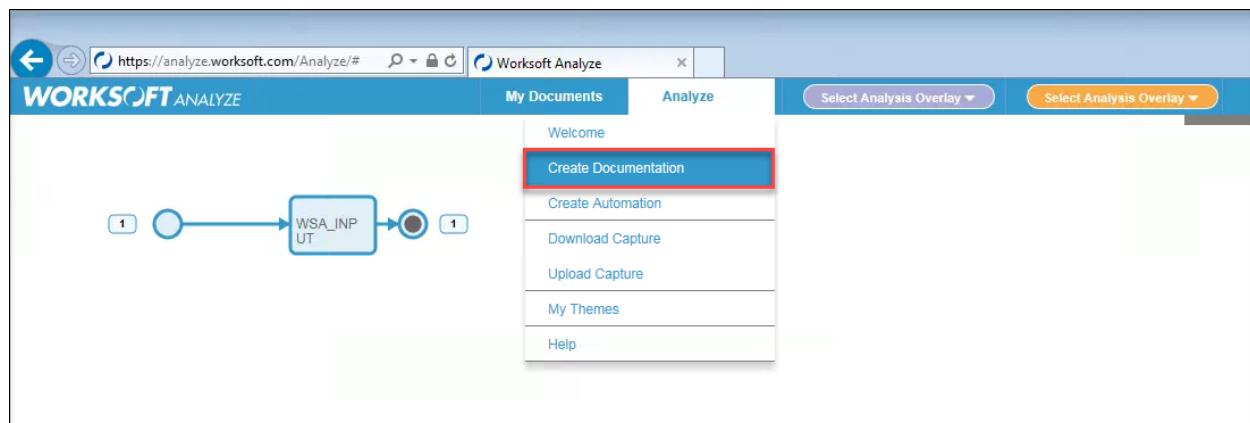
In this exercise, you will create detailed business process documentation for the WSA_Input business process.

Step	Action
------	--------

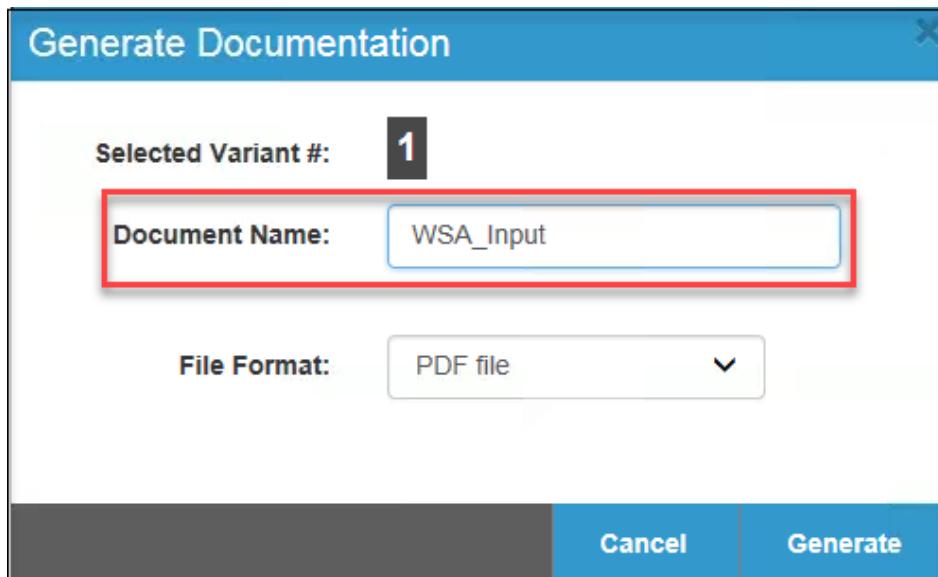
1. Highlight the required business process flow by clicking on the variant.



2. Click on the **Analyze** menu, and select **Create Documentation** from the list.



3. Enter the document name as **WSA_Input**.



4. Click **Generate**. *The user will receive an email with a link to the document generated. The document can also be found in the **My Documents** menu. The document is comprehensive with descriptions of each step; including screenshots of the objects as well as the data used.*
5. Select the **My Documents** menu. This will show a list of documents generated by you. Select the desired document. *The next page is a screenshot of the sample documentation produced.*

WORKSOFT®

Worksoft Analyze Business Process Documentation

Document Name: WSA_Input **Capture Segments:** 1
Date Generated: Friday, June 16, 2017 (UTC) **Certify Segments:** 0
User: Shonelle Clark **Manual Activities:** 0
Workspace: sclark **Dates included:** 5/11/2017 (UTC) through 5/11/2017 (UTC)

Documented Business Process

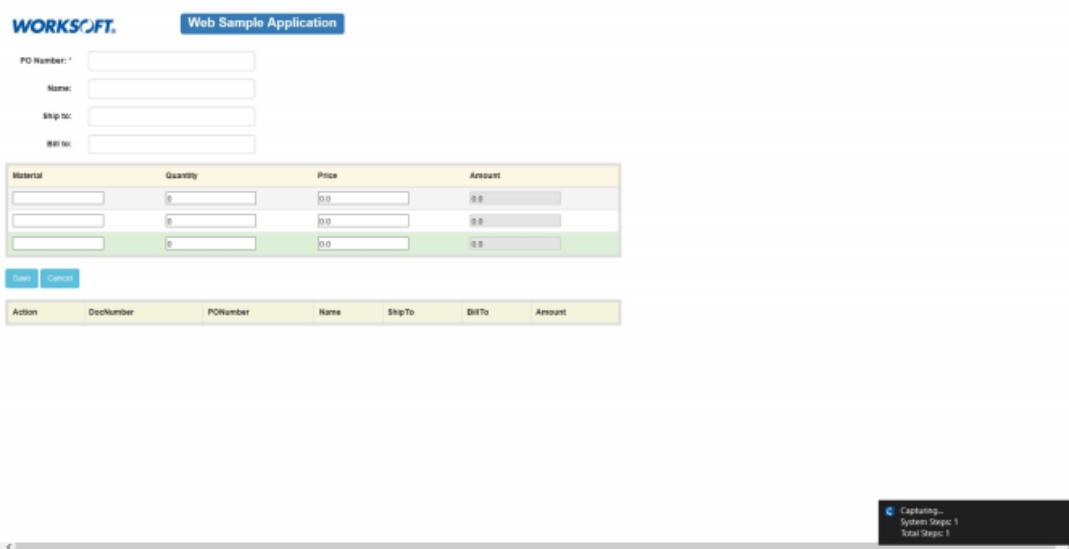
Activity Flow: WSA_INPUT

Business Process Execution

Activity: WSA_INPUT

Narrative: WSA_Input

Window Name: WebAccountManager

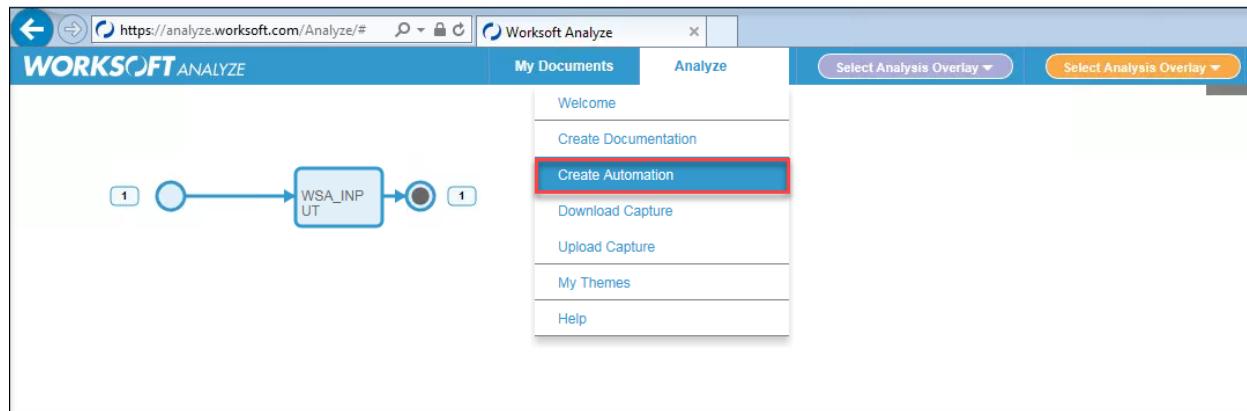


EXERCISE 2.7 – Generating Automation in Analyze

Analyze provides the ability to generate automation of a selected variant in the Analyze workspace. In this exercise, we will generate automation of the **WSA_Input** business process.

Step	Action
------	--------

1. Highlight the required business process flow by clicking the variant.
2. Click on the **Analyze** menu, and select **Create Automation** from the list.



3. Provide a name for the automation as per best practice, in this case **WSA_Input**.
4. Click **Generate**. A message indicating the successful generation of automation is displayed. **The automation will be saved and made available as an import file within Certify. We will complete the import in a later exercise.

Lesson Summary

You've completed the [Defining Processes Overview](#) lesson.

Key points to remember:

- Capture can be used to document and validate the end-to-end execution of your critical business processes.
- The key to successfully certifying your application is to first identify all the critical business processes. Each of the identified processes has a set of steps that describes how the process should be executed.
- As a best practice, you should break down the critical business processes into supporting processes.
- Process Capture™ plays an important role in the Business Process Discovery. Business users leverage Worksoft Process Capture to discover end-to-end business processes across all departments.
- The captured process with all the detailed comments and validation points will help reduce the efforts of the automation team by 40%.
- Process Capture can recognize Mobile, SAP, and Web applications.
- When choosing a name for your process, it is worthwhile to invest the time to develop and follow a process naming convention.
- Worksoft Analyze is a cloud based, automated business process discovery solution to collaborate with business users to more easily discover, visualize, and analyze their critical business processes.
- Using Worksoft Analyze you can create detailed business process documentation.
- Using Worksoft Analyze you can generate business process automation.

Lesson 3

Developing Automation Using Worksoft Certify

Overview

In this lesson, you will learn about:

- Processes and how to build them in Worksoft Certify
- Variables
- Layouts
- Recordsets

Objectives

After completing this lesson, you will be able to:

- Explain what processes are and how they are used.
- Define processes in Certify and add steps to processes.
- Explain LiveTouch and add steps to processes using LiveTouch.
- Explain how variables work and how they are used.
- Describe what layouts are and how they are used in Certify.
- Describe what recordsets are and how they work with layouts.
- Create a process from steps using the Create Process from Steps option.
- Create a layout, recordset, and variables using the Add to Layout feature.

Accessing Worksoft Certify

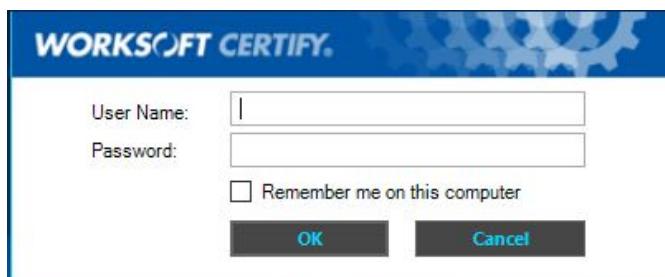
Before you can begin to use Certify, you must first launch it and log in successfully. Once logged in, you can explore the interface and become familiar with its navigation.

EXERCISE 3A — Logging in to Worksoft Certify

In this exercise, you will log in to Certify for the first time using the username and password provided by your instructor or Worksoft Administrator.

Step	Action
1.	Click Start , and select All Programs .
2.	Select the Worksoft folder.
3.	Select the Certify folder.
4.	Select Worksoft Certify .

The Worksoft Certify Login dialog box appears.



5. Type in field values as follows:

Field	Value
User Name	admin
Password	password

6. Click **OK**.

Note: You can also log in to Certify using the desktop shortcut icon.

EXERCISE 3B — Adding Yourself as a User

Important: These functions may have already been performed by a designated Administrator. If so, this training class will advance to the next set of exercises.

In this exercise, you will define your user profile and add it to the ReadWriteExecute group.

Step	Action
------	--------

1. In the Navigation Taskbar, click **Users & Groups**.
2. In the Navigation Tree, right-click **Users** and select **New User**.

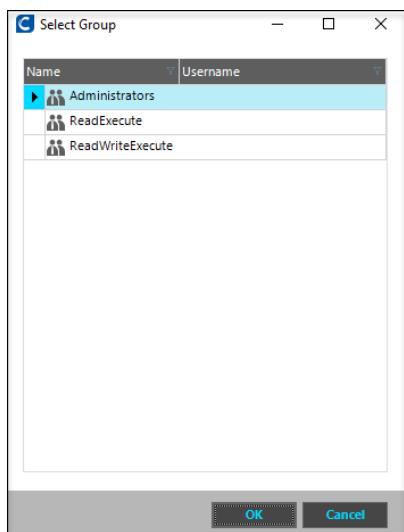
The New User dialog box appears.

The screenshot shows the 'New User' dialog box. The 'User Details' tab is active, displaying fields for First Name, Last Name, Username, Email ID, and Role. The 'Password Details' tab is also visible, showing fields for New Password and Verify Password, and options for Force password change on login, Password Never Expires, and Password Expires (selected). A 'Current Groups Containing User' section is present but empty. The dialog has standard 'OK' and 'Cancel' buttons at the bottom.

3. Type in field values or select options as follows:

Field	Value/Action
First Name	<your first name>
Last Name	<your last name>
Username	<your chosen username>
Email ID	<your e-mail>
Role	Leave blank Note: The Role field is an optional field. Users can use this field to state their role on the project.
New Password	<your password>
Verify Password	<your password> (repeated)
Force password change on logon	Leave blank
Password Never Expires	Select

4. Still within the New User dialog box, right-click in the **Current Groups Containing User** pane.
5. Select Add Group. *The Select Group dialog box appears.*



6. Select **ReadWriteExecute**.

7. Click **OK**.

The ReadWriteExecute Group appears in the Current Groups Containing User pane.

8. In the New User dialog box, click **OK**.

You are now added as a user of Certify and a member of the default ReadWriteExecute group.

9. Close Certify, and log in with **your new User ID and password**.

Exploring the Certify Interface

Figure 1 — Certify Interface

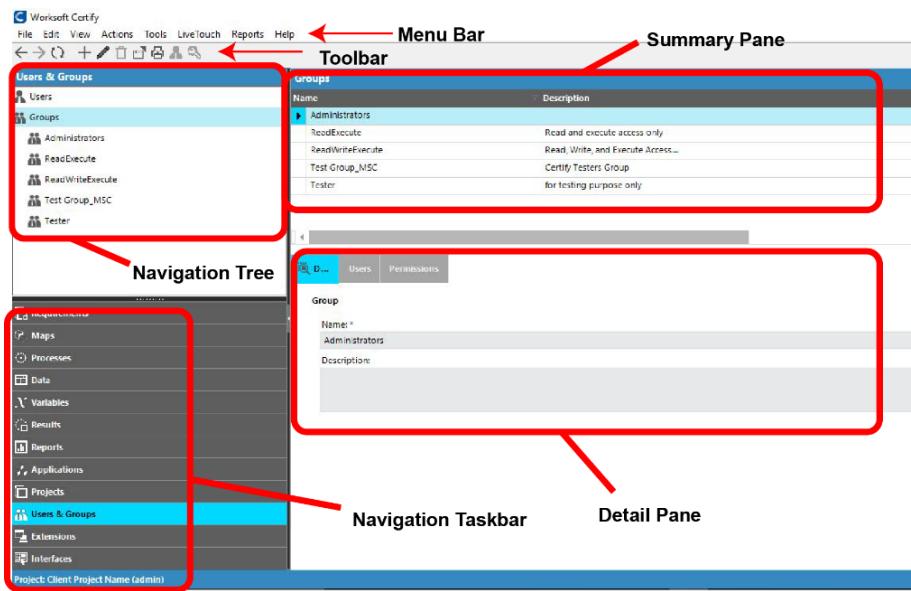


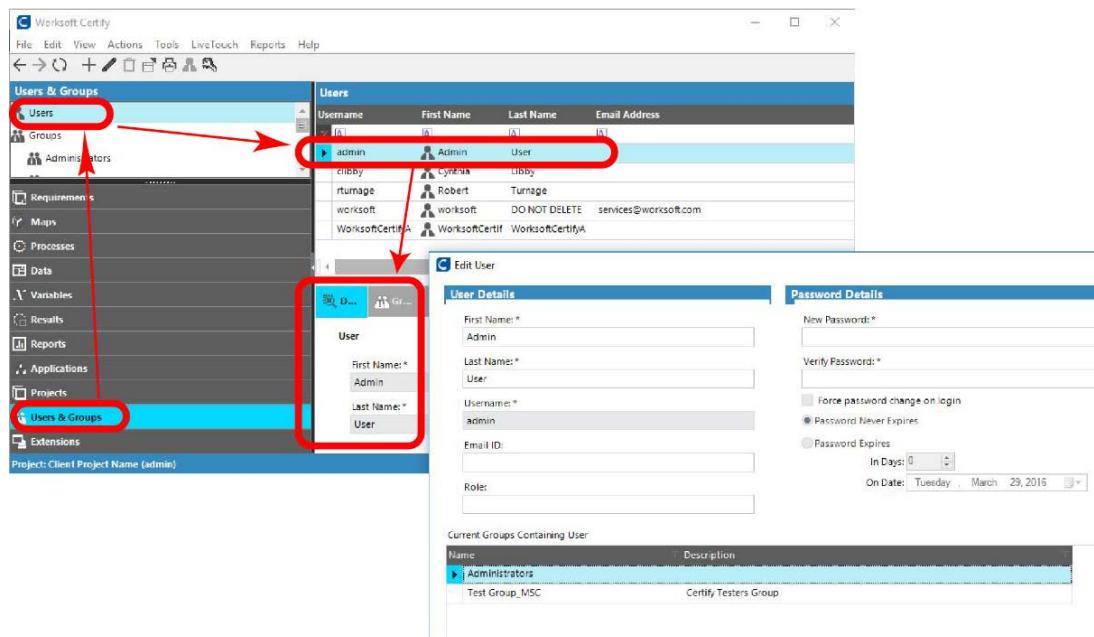
Figure 1 illustrates the seven sections of the main Certify Interface.

- **Menu Bar** — contains menus of functions and commands needed to carry out tasks in Certify. Some menu options become inactive depending on the task you select in the Navigation Taskbar.
- **Toolbar** — gives quick access to commonly performed actions. The buttons are specific to the selected tasks in the Navigation Taskbar.
- **Navigation Tree** — contains a hierarchy or tree view of the selected project component or administrative module.
- **Navigation Taskbar** — contains the buttons you click to access key Certify components. The button or task chosen affects what you see in the Summary and Detail Panes. Specific user access to these buttons is controlled by a Certify administrator.
- **Summary Pane** — located on the top right-side of the primary window. This pane shows dynamic, display only information that changes depending on the task and nodes selected in the Navigation Pane.
- **Detail Pane** — located on the bottom right-side of the primary interface (beneath the Summary Pane). It contains tabs of information related to the items in the Summary Pane. This information is also dynamic and display-only.
- **Status Bar** — located on the bottom left-side of the screen (beneath the Navigation Taskbar). It displays the name of the open project.

General Navigation Flow

There is a consistent pattern to navigating the interface elements previously discussed. Depending on the task you want to perform, you first select the appropriate button in the Navigation Taskbar. The other elements change to reflect the data contained in the chosen area. Next, you select a node in the Navigation Tree. Related items then appear in the Summary Pane. You then select an item in the Summary Pane to carry out tasks (as shown in Figure 3) or click on the tabs in the Detail Pane. You can also carry out tasks in those tabs (other than the Details tab). You will discover that Certify actions are accessed by a right-click.

Figure 2 — Relationship of Certify Interface Elements



Note: You will follow this clockwise pattern repeatedly as you learn about and carry out tasks within the various Certify components.

Worksoft Naming Convention – Best Practices

We at Worksoft highly recommend investing the time to develop and follow a process naming convention. As test volumes expand and assets are transferred among users, naming conventions help to promote reuse and reduce confusion. Ideally, names are descriptive enough that their purpose and use is immediately clear and external documentation is not needed.

This section describes naming conventions for Certify concepts that you have not yet learned (e.g. Variables, Data Layouts, and Recordsets). As you look through this section, you will see how the naming conventions work together. Later, when you begin using these concepts, you will use these conventions.

Transaction Code/Component or Unit Test Process

- SAP transaction codes (T-codes) or components are capitalized.
- Separate descriptions by an underscore (_).
- Remove spaces from descriptions and capitalize the first letter of each word.
 - **Format:** TCode_Description1_Description2_Description3
Component_ComponentDescription
 - **SAP Example:** VA01_CreateStandardOrder_SingleItem
VA01_CreateStandardOrder_MultipleItems
 - **HTML Example:** WebOrder_CustomerCreate
MyPortal_RegisterStudent
- If a process uses a child process to work with a multi-element object such as a table or grid, it should be extended with the parent process name +, _C, + additional description. This will ensure the parent and child processes are sorted next to each other.
 - **Example:** VA01_CreateContractOrder_MultipleItems
VA01_CreateContractOrder_MultipleItems_C_TableInput

SAP Human Resource Design

- For transactions that encapsulate functionality (for example HR transactions such as PA30 and PA40), it is best to subdivide processes further than the transaction level.

- The use of Info Types is recommended for naming and defining processes.
- HR processes may also be grouped by Country or State (e.g. CA for Canada or CA for California).
 - **Format:** Tcode_IT<ITypeNumber>_<InfoTypetext>_<Country or State>
 - **Examples:** PA40_IT0068_GarnishmentCompensation_CA
PA40_IT0068_NewInitialBalance
PA40_IT0071_PensionFunds_GB

Integrated or End-to-End Process

- Separate test names and descriptions by an underscore (_).
- Remove spaces from descriptions and capitalize the first letter of each word.
- Integrated Processes contain only calls to other processes and Comments.
 - **Format:** ApplicationArea_Description1_Description2_Description3
 - **Example:** OTC_ShipBill_ConstantQuote
CRM_CustomOrder_ProcessThroughSAP
WebOrder_FromBrowser_ToSAP_ToMainframe

Naming Conventions for Variables

- Named the same as the field it represents.
 - **Format:** Variable names will closely resemble field names
 - **Example:** Order Type
Ship-to-party
Exp.Date (ex.: if field is "Expiration Date" then make use of Exp. Date rather than create a new variable)
- If a field needs more than one variable (e.g. Requested delivery date as a Text type variable and as a Date type variable), then delimit the variable with the difference – such as a _VariantName.
 - **Format:** Req. deliv.date_Format Type
 - **Example:** Req. deliv.date_Date

- A field may appear on the screen multiple times such that you need different values for each occurrence. The same variable name should be used with a '#' to indicate that it indeed is a necessary variable and not a duplicate of an existing variable.

Example: Customer#1 is not equal to Customer#2

Data Layouts

- Named exactly as the process name.
- Variant Layouts will have the same name as the Process name plus identifier.
 - **Format:** Layout Name = Processes Name (not manual test case name)
 - **Example:** Process Name = VA01_StandardOrder
Layout Name = VA01_StandardOrder

Process Name = MM01_CreateMasterData
Layout Name = MM01_CreatemasterData

Integrated Test Name = O2C_ShipBill_ContractQuote
Layout Name = O2C_ShipBill_ContractQuote
- Most Certify processes have only one layout. However, there may be a need to have more than one. In this case, use the same name as the original layout and add a variant identifier.
 - **Example:** Process Name = MM01_CreateMasterData
Primary Layout Name = MM01_CreateMasterData
Secondary Layout Name = MM01_CreateMasterData_3rdPartyData

Recordsets

- When implementing the Recordset naming convention, use the Process Name as the recordset name for Integrated Processes. Be descriptive as possible when naming recordsets.
- Recordset names will, in most cases, match the Process name and Layout name being created. Any number of recordsets are possible for any given test.
 - **Format:** Process Name_Purpose
 - **Example:**
Process Name = VA01_StandardOrder
Layout Name = VA01_StandardOrder
Recordset Name = VA01_StandardOrder
Recordset Name = VA01_StandardOrder_InternationalCustomers

Recordset Name = VA01_StandardOrder_NationalCustomers

o **Example:**

Integrated Test Name = O2C_ShipBill_ContractQuote

Layout Name = O2C_ShipBill_ContractQuote

Recordset Name = O2C_ShipBill_ContractQuote_Europe

Recordset Name = O2C_ShipBill_ContractQuote_China

Processes Overview

In Certify, processes are used to document and validate the end-to-end execution of your critical business processes. Processes typically map your existing test cases to the business functions associated with the application under test (AUT).

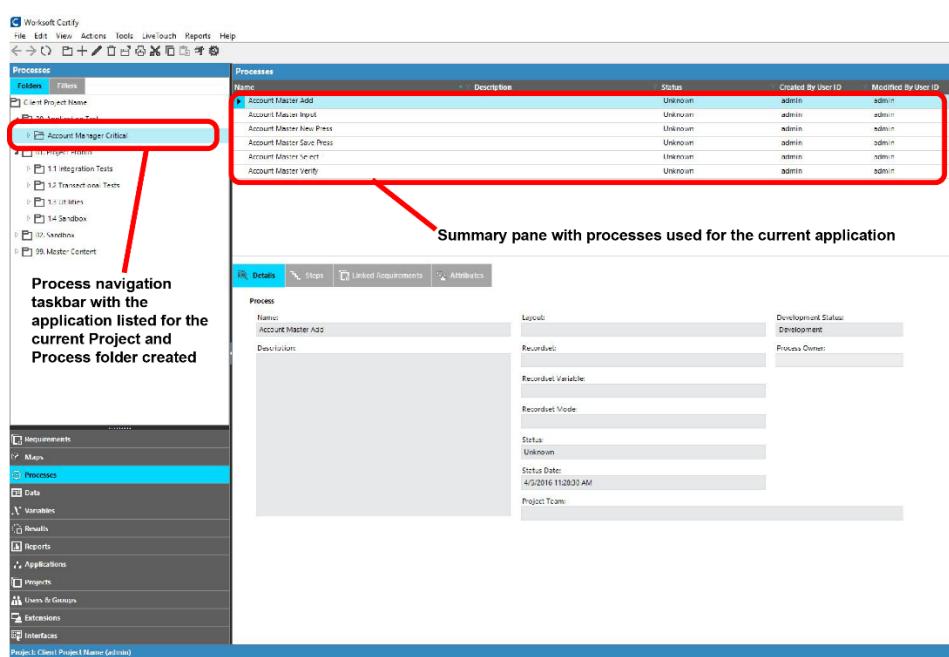
Processes consist of a series of individual steps where a step performs an action against an object. Actions can include entering or verifying a field value, pressing keys or buttons, or verifying results like field values, object states, or messages.

Each process performs a discrete function, such as adding an order or finding an existing order. When defining processes, you select the objects and actions to show the sequence of how your application operates.

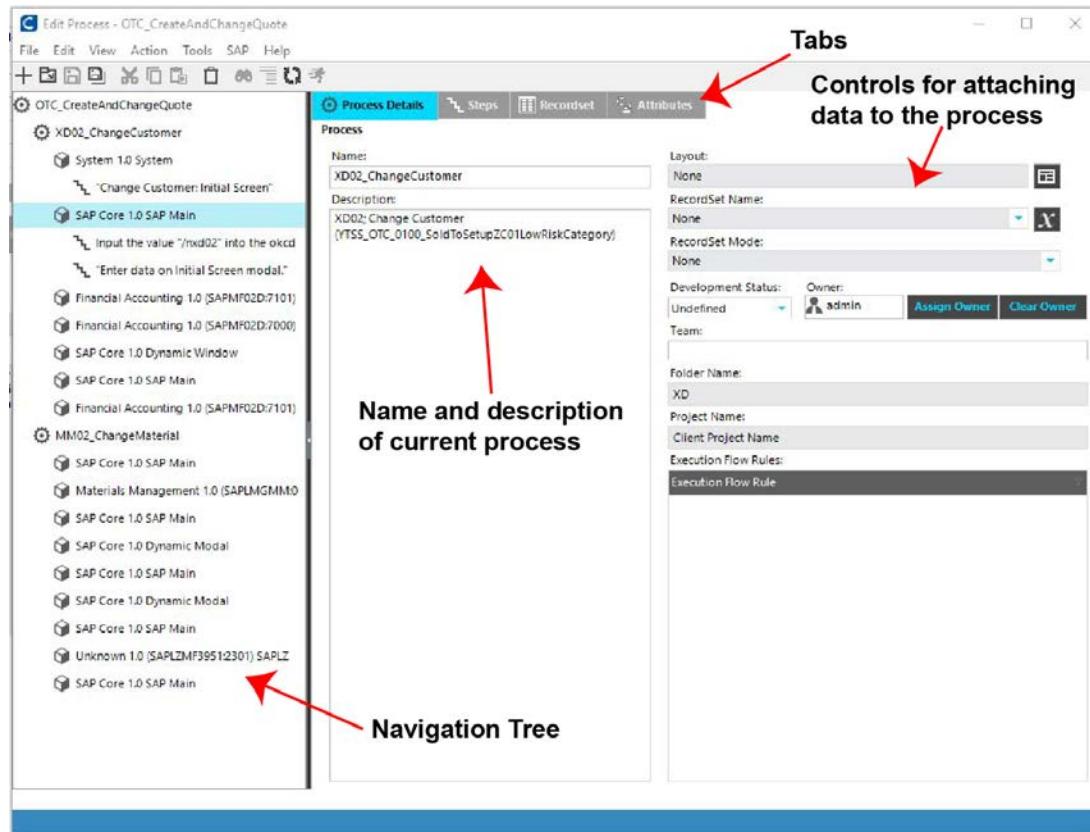
Process and Data Editor Overview

Defining and using processes is carried out in the Processes window and Process Editor. Figure 3 shows the Process window with important sections highlighted. The Summary Pane of this window shows the list of processes that have been created for the current application.

Figure 3 — Process Window Interface with Callouts



Selecting New Process from the shortcut menu takes you to the Process Editor, shown in Figure 4, where each process is built.

Figure 4 — Process Editor Interface with Callouts

The **Process Details** tab is the default tab. The screen is divided into sections and tabs. The left side of the screen is the **Navigation Tree**. The tree view displays the processes created to verify your business workflows. The middle section of the Process Details tab displays the **Process Name** field and the **Description** field. Filling out this information will improve your test documentation. The right side of the screen provides controls for attaching data to your processes with the **layout**, **recordset name**, and **recordset mode** fields. Additional information about the process location and project are provided with the read-only fields. The **Execution Flow Rules** associated with the processes are displayed in the lower right corner of the screen. You can add and modify the rules. Execution Flow Rules provide you with a way to control the process flow by evaluating conditions during execution.

The **Steps** tab is where you will create the steps that will run when the process is executed. You can view recordsets under the **Recordsets** tab, as well as attach attributes by clicking the **Attributes** tab.

Once all steps are entered and the process is saved, it appears in the Summary Pane (see Figure 3).

Identifying Processes

The key to successfully certifying your application is to first identify all the critical business processes. Each of the identified processes has a set of steps that describes how the process should be executed. In many cases, a critical business process could be made up of several smaller processes that focus on functional areas or navigation.

As a best practice, you should break-down the critical business processes into supporting processes. For instance, navigation to a window and pressing a button would make up a navigation process that can be reused. This will provide wider coverage and reuse of processes. For example, a CF01_CreateBillingDocument process can be reused, meaning it can be called from one or more other end-to-end process.

The actual number of processes you need to create will vary based on the size and complexity of your application.

To assist in identifying the processes that you need for your application, try these methods:

- Analyze the requirements to determine how many processes are needed to thoroughly cover each type of feature or function.
- Analyze the functional areas of the application to determine if any shared processes would make the processes more efficient.
- Verify that each process maps to a requirement that covers a business function or feature.
- Organize your identified processes by business function and adopt a naming convention that will ensure your processes are easy to locate. (See “Naming Processes” for naming guidelines.)

Designing Processes

Process design is just as important to the success of certifying your application. A solid test should be designed where manual intervention is omitted.

When planning a process, keep these best practices in mind:

- All transactional or component processes should be created with the purpose of being used in an end-to-end test.
- Good data management helps ensure that a process will run without manual participation. Adding a process for adjusting inventory to meet the expectations of the test.
- Audit (validation) points should be included in the test. All validation/audit points should include a screenshot for audit purposes.
- Tests should be linear. **Do not write logic to test logic.**

Naming Processes

When choosing a name for your process, it's wise to invest the time to develop and follow a process naming convention. As test volumes expand and assets are transferred among users, naming conventions help to promote reuse and reduces confusion. Ideally, names are descriptive enough that their purpose and use is immediately clear and external documentation is not needed.

It is important to note that Certify lists are often sorted alphabetically and can be searched using filters based on name, description, keyword, and user defined fields. For example, processes called AddCustomer and UpdateCustomer would not appear together in an alphabetic sorting, while CustomerAdd and CustomerUpdate would. On the other hand, the former structure could yield a useful list if a filter were created on the word Customer within the name field. These types of naming standards are essential to designing a reusable, maintainable test suite.

As a recommended best practice, start your process names with the window name, followed by an underscore, and finally the action to perform. For example, WSA_Input and WSA_SelectandVerify would describe two processes that use the same page Web Sample Application (WSA) but the actions are different.

As another best practice, you should plan to organize your processes into folders that describe the processes contained within the folder. For example, you could have a folder called **WSA_CreateandModify** containing the integrated critical business processes, as well as the sub-processes within that integrated process.

Creating Process Folders

In Certify, creating and using folders is a way to organize project components. You can create as many folders and subfolders as necessary to contain the processes for your projects. The folder names you use should be descriptive of the processes contained within them. **As a best practice, use a similar naming convention to the one used for naming processes discussed previously.**

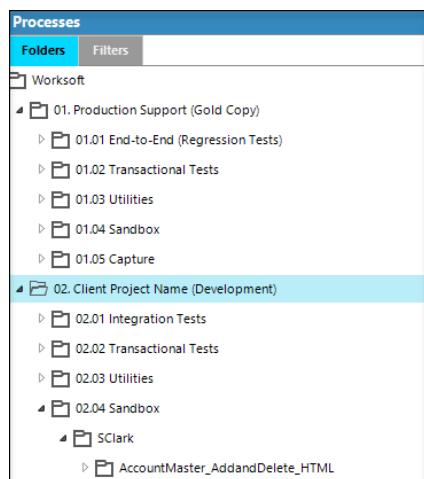
You can create folders by right-clicking the project folder or an existing folder in the Navigation Tree and selecting New Folder or by copying and pasting folders to create new folders.

For example, you can create a folder called WSA_CreateandModify and place all the processes created for the Worksoft Web Sample Application in that folder. Later, as you create more and more processes, you may want to create subfolders under WSA_CreateandModify to hold specific process types or window-related processes so they are easier to locate.

Your Sandbox folder

Throughout this course you will be using a folder to contain your processes. In the following exercises, you will define and add the steps for all the required processes. However, before you begin defining the processes, you'll need to create your Sandbox folder and a folder called WSA_CreateandModify to contain the processes you will build.

The folder path may be as indicated in one of the examples below, or your administrator may suggest another location. When the exercises reference ***Your Sandbox Folder*** simply select the appropriate folder from the tree.



Note: In the following exercise the 02.04 Sandbox folder is used as the container for your Sandbox folder. If this is not correct, use the path suggested by your administrator.

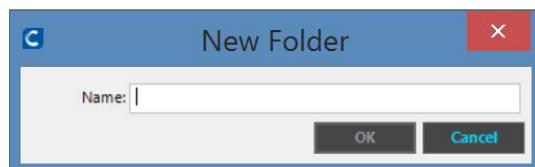
EXERCISE 3.1 — Creating a Folder to Hold Processes

In this exercise, you will create a folder structure to hold your processes.

Step	Action
------	--------

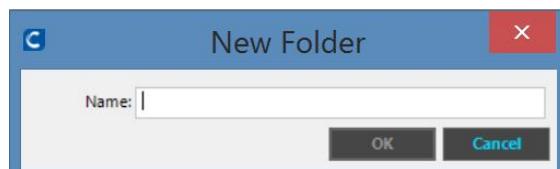
1. In the Navigation Taskbar, click **Processes**.
2. In the Navigation Tree, click the arrow next to the **New Project** folder.
3. Click the arrow next to the **02.04 Sandbox** folder.
4. Right-click **02.04 Sandbox**, and select **New Folder**.

The New Folder dialog box appears.



5. In the **Name** field, type <your name>.
6. Click **OK**.
7. The folder appears under Sandbox in the Navigation Tree.
8. To create the process subfolder, right-click your name, and select **New Folder**.

The New Folder dialog box appears.



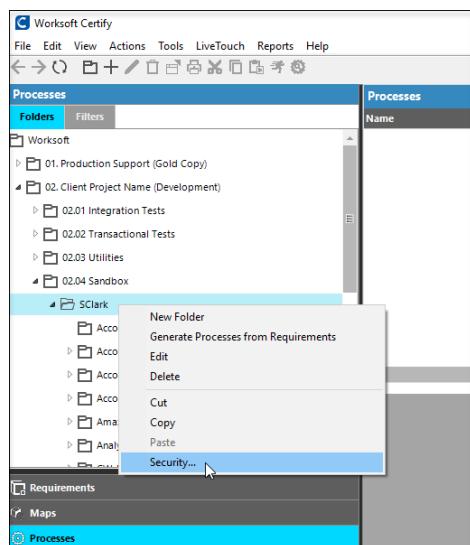
9. In the Name field, type WSA_CreateandModify.
10. Click **OK**.
11. The folder appears under your Sandbox folder in the Navigation Tree.

EXERCISE 3.2 — Adding Folder Security

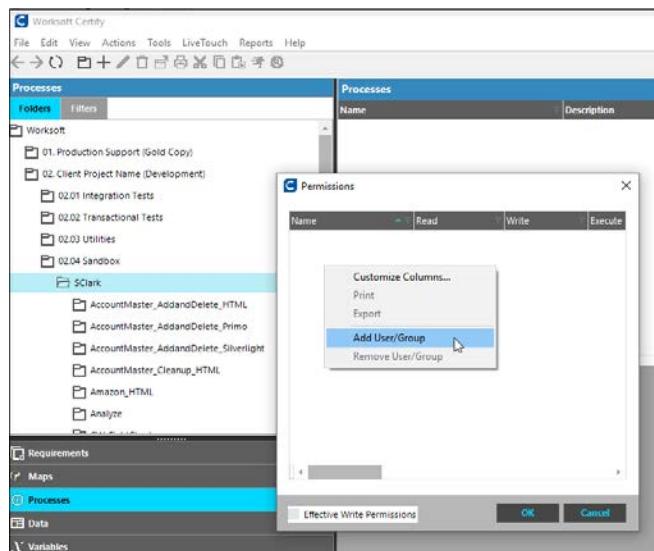
In this exercise, you will add security to your Sandbox folder; this will allow only you or an admin to edit your processes.

Step	Action
------	--------

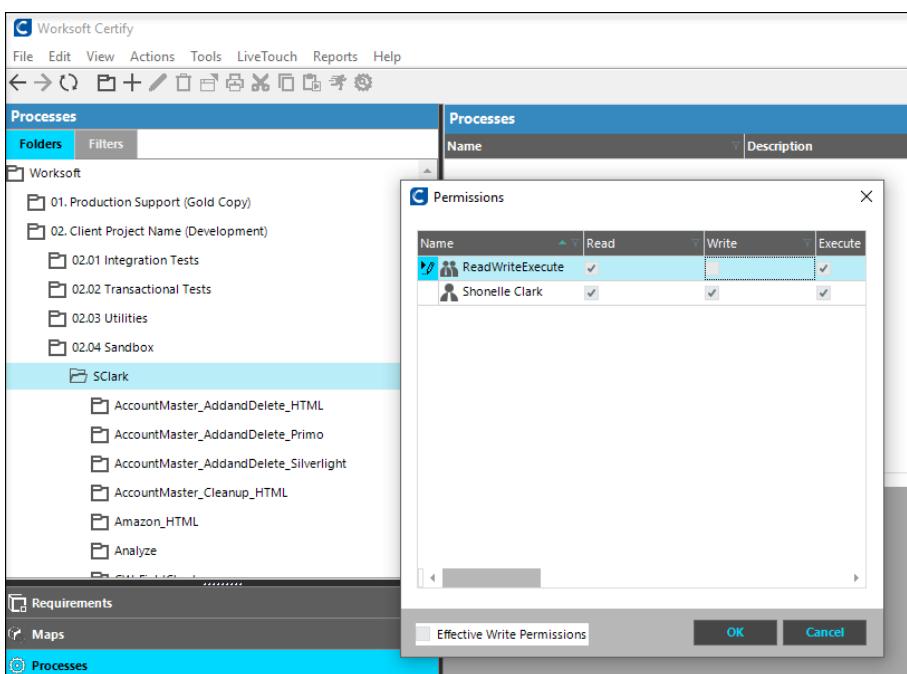
1. In the Navigation Tree, right-click your **named Sandbox** folder, and select **Security...**.



2. In the Permissions dialog box, right-click and select **Add User/Group**.



3. In the Select User/Group dialog box, select **your user ID** and the **ReadWriteExecute group** and click **OK**.
4. Modify the Permissions of the ReadWriteExecute group by unchecking the **Write** permission. This will allow others to view, execute, and copy your processes.



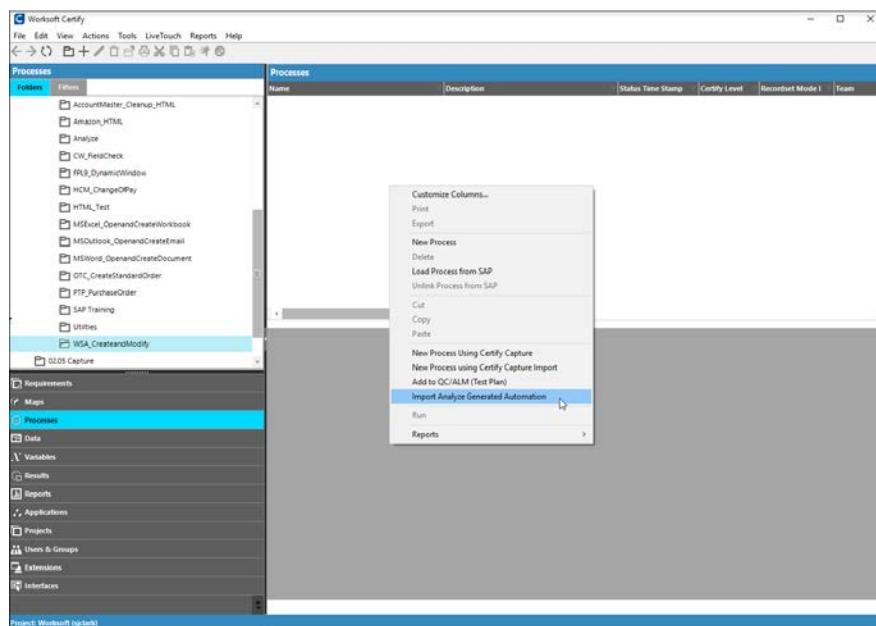
5. Click **OK** to close the Permissions window.

EXERCISE 3.3 — Import Analyze Generated Automation into Certify

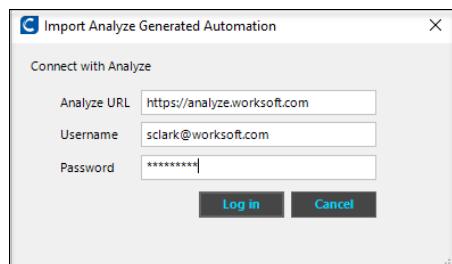
In this exercise, you will import the Analyze generated automation of your WSA_Input process.

Step	Action
------	--------

1. In the Navigation Tree, select the **New Project** folder.
2. Select the **02.04 Sandbox** folder.
3. Navigate to your Sandbox folder, and select your **WSA_CreateandModify** folder.
4. Right-click in the Summary Pane, and select **Import Analyze Generated Automation**.

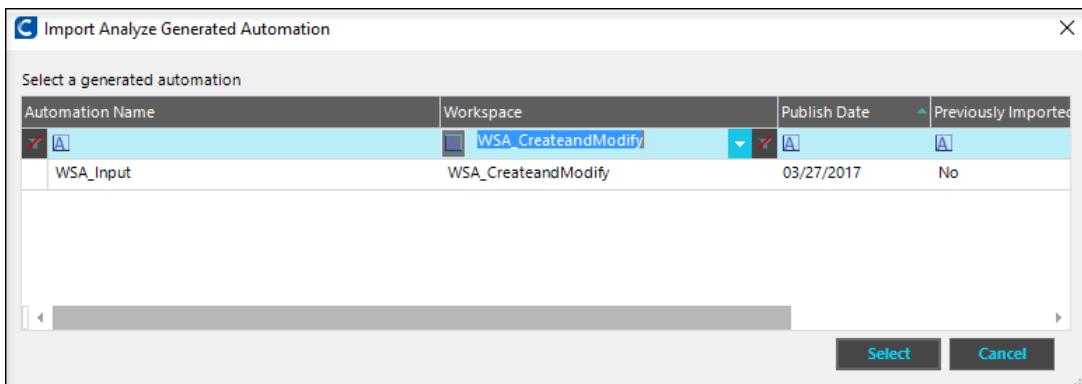


5. Enter your Worksoft Analyze credentials and click **Log in**.

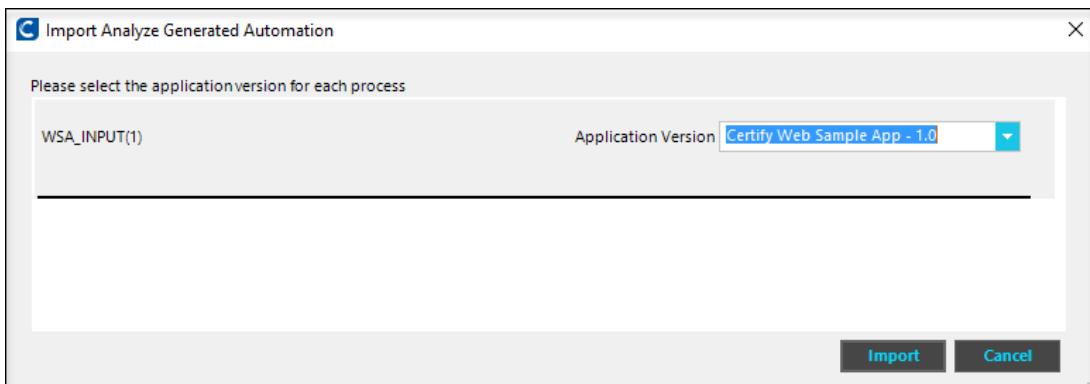


6. An **Import Analyze Generated Automation** dialog box is displayed. Select

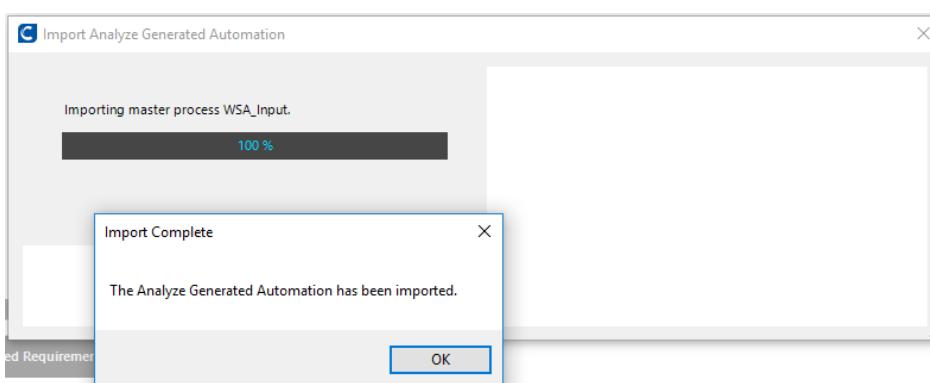
WSA_Input and click **Select**.



7. Select **Certify Web Sample App – 1.0** from the Application Version drop-down list.



8. Click **Import**.



A message is displayed indicating the Analyze Generated Automation has been imported.

9. Press **OK**.

10. Press Refresh in Certify .

11. You will notice a Master Process **WSA_Input** and a sub-process **WSA_INPUT(1)**.

Parts of a Process Step

Each step in a process contains information needed by Certify.

- **Application Version** — generally this is the name of the application under test. For very large applications, such as the SAP GUI, the Application Version may be a component such as "Sales and Distribution". The **Application Version** drop-down contains a list of available application versions that are associated with the current project.
- **Window** — screen, window, or page in the application. The **Window** drop-down contains a list of learned and system windows available for the selected application version.
- **Object** — field or control to be acted upon. The **Object** drop-down contains a list of available objects for the selected window.
- **Action** — input, verify, store, etc. The **Action** drop-down contains a list of valid actions for the selected object.
- **Narrative** — this is a phrase describing the step. The narrative is read-only and automatically updates as the step is updated.

At the bottom of the steps area:

- **Parameters** — value(s) to modify the selected action.

As an item in a column is selected, the columns to the right change to reflect the items available. If a different Window is selected, the object column updates to reflect the objects in that window.

As an Action is selected, the Parameters area changes to reflect the information needed for that Action. If the Action is Input, the Parameters will include the value that will be input.

Note: The Application Version, Window, and Object may have slightly different names in your database. Generally, this is completely fine. If your steps are significantly different, ask your Instructor or Certify support person to verify that the step was created correctly.

Let's explore Step 2 of the WSA_INPUT(1) process. It contains:

- **Application Version** — CertifyWebSampleApp 1.0
- **Window** — Web Account Manager
- **Object** — txtPONumber
- **Action** — Input

- **Narrative** — Input “1234” into the txtPONumber EditBox.
- **Parameters** — Value = 1234 and Follow-up Key = None or {TAB}

You can also manually create a step by selecting from each drop-down list.

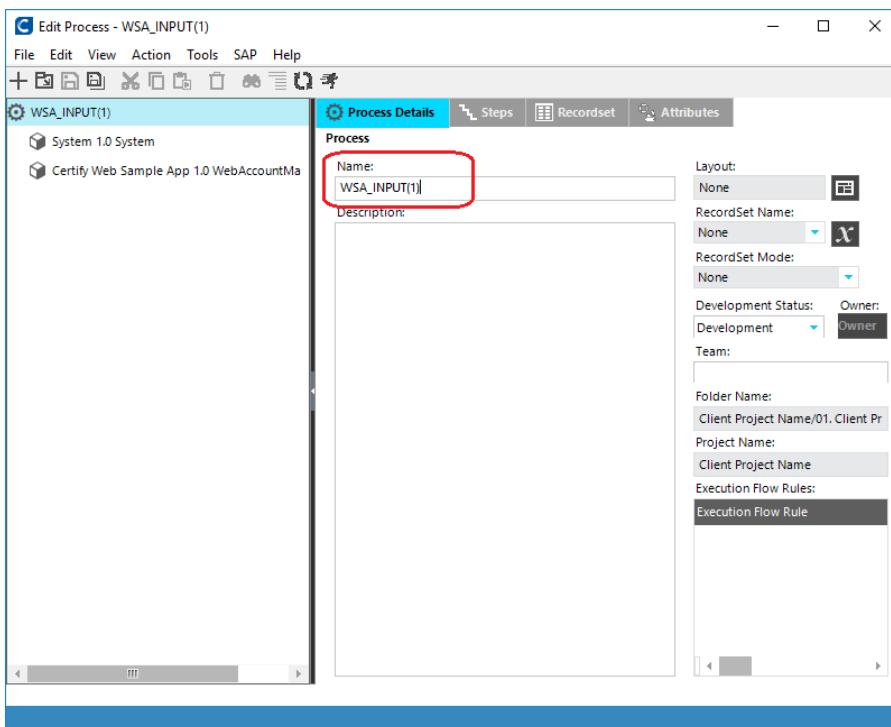
Next, we will execute the step so that data will appear in the fields for the next series of exercises. There are several ways to execute a step. Once you highlight the step(s), you can: 1) click **Action** from the menu and select **Execute Step**, 2) press the **F6** function key, or 3) right-click on the step and select **Execute Step**.

EXERCISE 3.4 — Explore WSA_Input, Rename the Process, and Execute Steps

In this exercise, you will modify the WSA_Input process and execute the process steps.

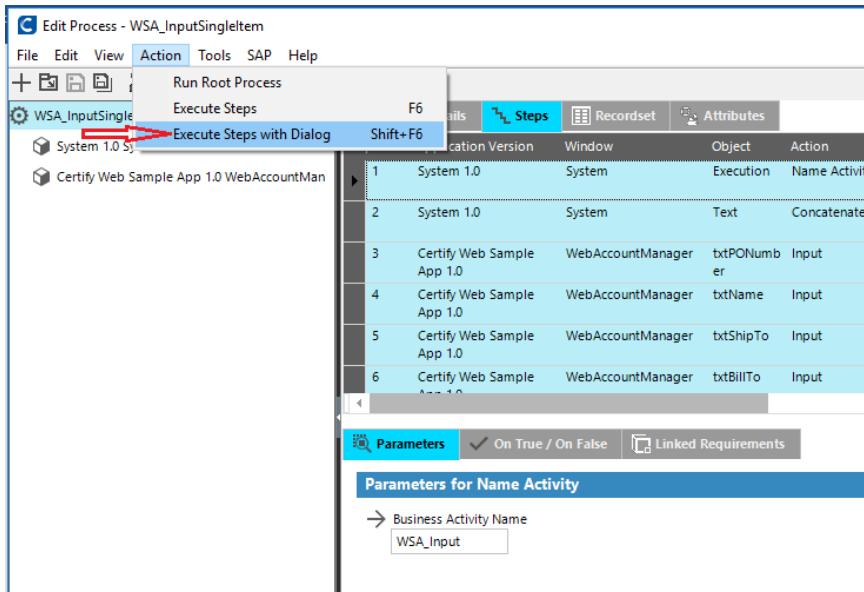
Step	Action
------	--------

1. Double-click the **WSA_INPUT(1)** process to open it, click the **Steps** tab, verify there are steps inside the process.
2. *If you see an additional process named WSA_Input in the Summary Pane, this is the integrated process for WSA_INPUT(1). Right-click and delete the WSA_Input process. We do not need it now.*
3. Open the **WSA_INPUT(1)** process.

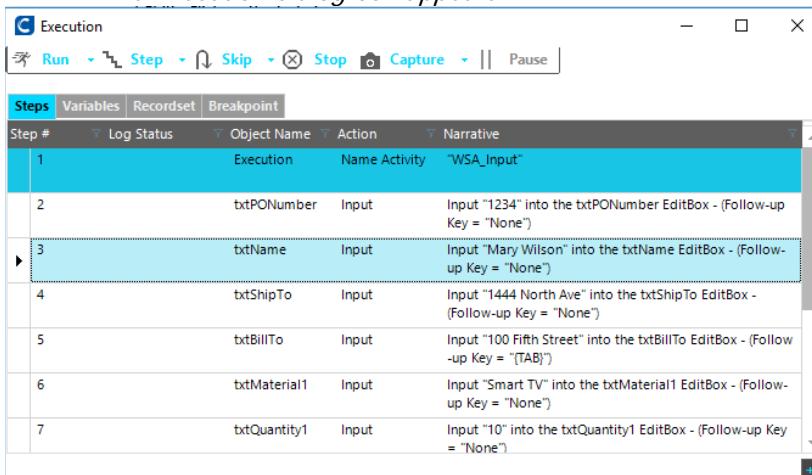


4. Rename the process to **WSA_Input**, and add the following Description: **Creates a purchase order**.
5. Click **Save** .
6. To execute the steps:
 - a. In the Steps area, press CTRL+A to select all the steps or click on the first step and Shift+Click on the last step.

- b. In the Menus, click **Action**, and select **Execute Steps with Dialog**.



The Execution dialog box appears



- c. In the Execution dialog box, click the **Step** button
- d. Click again.

The first two steps execute. In the Execution dialog box, notice that the Log Status field displays a check mark and the word "passed".

Execution				
Run Step Skip Stop Capture Pause				
Steps	Variables	Recordset	Breakpoint	
Step #	Log Status	Object Name	Action	Narrative
1	passed	Execution	Name Activity	"WSA_Input"
2	passed	txtPONumber	Input	Input "1234" into the txtPONumber EditBox - (Follow-up Key = "None")
3	passed	txtName	Input	Input "Mary Wilson" into the txtName EditBox - (Follow-up Key = "None")
4		txtShipTo	Input	Input "1444 North Ave" into the txtShipTo EditBox - (Follow-up Key = "None")
5		txtBillTo	Input	Input "100 Fifth Street" into the txtBillTo EditBox - (Follow-up Key = "{TAB}")
6		txtMaterial1	Input	Input "Smart TV" into the txtMaterial1 EditBox - (Follow-up Key = "None")
7		txtQuantity1	Input	Input "10" into the txtQuantity1 EditBox - (Follow-up Key = "None")

Process Selected steps of WSA_INPUTSingleItem Step Count 2 0 0

- e. Click **Step** again to execute the next step.

The next step executes, and the Log Status field is populated.

- f. Click **Run** to execute the final steps.

The steps execute. Notice the status bar displays the number of steps run and the results.

Selected steps count = 9. Overall status = passed. Last executed step status = passed

- g. If one step fails, the Overall status will be "failed". Correct any failures before moving on to the next section.

Variables Overview

A **variable** is a symbol or name that represents a value. In Certify, variables are an important component of the Business Process Certification because they provide a means of representing the data that you want to use in your processes. Variables contain a specific type of value that can be inserted into a field, acted on, or stored for future comparison against another value.

The most common use of variables is for data-driven testing, where process execution loops through a series of data values. For example, if the steps for your process are the same but the data is expected to vary from one execution to another, then the process needs to be defined only once using variables in place of static values. Certify processes use variables to create several executions in one single test.

Variables can also be used when you need to store or verify system data, such as the System Date or Machine Name, or when specific data for a user is required, such as a user ID or password to log into an external system or application.

Preparing to Use Variables

When preparing to use variables, the first thing you should do is go through your existing processes and identify all the places where variables can be used in place of static data. **As a best practice, create a table or spreadsheet showing the processes and all the variables you will use for each process.** This method will help you later when you create the recordsets to be used with your processes.

For example, Table 1 below shows the WSA_Input process and all the variables required for that process.

Table 1 — Variables Needed for the WSA_Input Process

Process	Variables Needed	Data Type
WSA_Input	PO Number	Text
	Name	Text
	Ship to	Text
	Bill to	Text
	Material	Text
	Quantity_Numeric	Number
	Price_Numeric	Number

Defining Variables in Certify

In Certify, there are several types of variables:

Project Variables — variables that are specific to the project you are working in and the most commonly used processes. Project variables can be created and managed in the Variables window or

while creating processes in the Process Editor. You can use Project variables on both the Data window to define the layouts of your recordset and in the Process Editor to hold the values of objects within your process steps.

System Variables — pre-defined and used as read-only variables for data values during test execution. The system variables that you can use in your process steps are:

- Date
- Process Name
- Recordset Name
- Recordset Row Number
- Last Step Status
- User Name
- Layout Name
- Machine Name

User Variables — user-defined variables that can be used during process execution, such as other system user IDs and passwords, test dates, test numbers, tester ID number, etc. User variables are created and managed from the Extensions window by a Certify Administrator or manager, and they are shared across all projects. The values for user variables are specified by each user by selecting Set User Variables from the Tools menu.

Each type of variable contains text, number, or date values.

- **Text** — consists of alphanumeric and special characters. Text variables have a system maximum length of 65,535 characters (64K).
- **Number** — consists of integers, floating point values, and exponential notations. You can also apply a specific data type format to a number variable to display the value in a certain format (i.e. 12345.00 or \$12,435.00). If no data type format is specified, the default system number format is used.
- **Date** — consists of any combination of month, day, and year in any Certify-supported format. Date variables default to today's date. You can also apply a specific data type format to a date variable to display the date in a certain format (i.e., MM/DD/YYYY). If no data type format is specified, the default system date format is used.

Variables are shared among all processes in Certify. **As a best practice, you should search through the list of existing variables to see if there is one that satisfies your need.** If a new variable is needed, the naming convention should be followed as previously discussed. **Variables should be named the same as the field it represents.**

You can also input an initial value for the variable and/or select to mask the value of the variable. If an execution step contains a variable with an initial value, the object associated with the step is populated with the initial value. When the Mask checkbox is selected, the value is replaced with asterisks (e.g.,

****) to hide the real value from view in the Certify Result Viewer. This feature is useful when your variable value is a user ID or password.

Using Variables in Certify

Listed below are some additional scenarios when you might want to use variables in your processes:

Field value is unknown before execution. The value of a variable is changed when an application returns a value that is randomly generated or generated on the fly. For example, creating an order generates an order number. The order number can be stored in a variable for verification or other uses.

A value must be derived from a combination of other variable values and/or static values.

The variable value is based on calculations or concatenations of other variables or static values. These operations are performed by process steps and the resulting value is stored in a variable. For example, an application requires the month and year to be added to a filename to locate the file within a list. The month and year can be concatenated from two variables and added to another variable containing the file name. The variable can then be used to locate a file within the list.

Process execution order varies. The names of processes can be named by a variable so that each iteration through the recordset calls a different process. For example, you may have a set of processes for your application to add an account, add a transaction, and verify a portfolio. Instead of creating three separate Execute Process steps, you can associate a recordset with your master process and have one Execute Process step that calls an Account Process variable. Since the variable is associated with a recordset containing the names of all the processes you want to execute, each of the processes are executed in the order they appear in the recordset.

You need to work with system information. Pre-defined system variables can be used in your process steps to capture information such as the computer name, user name, and process name. For example, you may want to capture the system date for comparison against another date variable, or get the status of the last step executed to verify the step passed.

You need to log into other systems or applications during execution. User-defined variables can be used in your process steps to log into other systems or applications referenced in your process execution. For example, you may have a legacy system containing data that you need to compare with data from the application you are testing. You can add a user variable in your process to input the user ID and password for the legacy system or application.

Creating and Modifying Variables

Variables are created and modified from the Variables window. The Variables window shows all the variables that have been created in the project.

Variables can also be created from the Process Editor. While creating a process step that requires an input, verify, or store value, after clicking the Select Variable button, variables can then be added or modified in the Select Variable window.

Adding Variables to Existing Process Steps

Variables are applied to process steps in the Process Editor. When you create a new step or select an existing step, you use the Parameters option for the selected step to select the variable you want to use. If you are using static data, you will replace the static data with the variable you have created.

You created the **WSA_Input** process with static data. You entered a value for the PO Number, Name, Ship to, Bill to etc. fields. You did this primarily to learn how to define processes without the added complication of including variable data. However, to automate your testing, you will want to use variables within your processes so that you can perform data-driven testing.

The PO Number should be unique for each PO created. We will concatenate a static value and the System Variable **Date** to create a unique PO number.

For example:

PO Number = 8403115637

(Requirement is the PO number should be unique and a numeric value.)

EXERCISE 3.5 —Adding Variables to Process Steps

In this exercise, you will add variables to the steps in the WSA_Input process to generate a unique PO number using the concatenate action in a Certify step.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, click the WSA_CreateandModify folder in Your Sandbox Folder .
3.	Locate and right-click the WSA_Input process, and select Edit .
	<i>The Process Editor appears.</i>
4.	Click the Steps tab.
5.	Locate and select the txtPONumber/Input step (step 2) .
6.	Right-click Step #2 , and select Insert Step Above .
	Tip: When inserting a step above or below, the application version, window, and object of the original step are copied to the newly inserted step.
7.	Click the Application Version drop-down arrow, and select System 1.0 .
	Tip: System 1.0 is a special Application Version and will always appear at the top of the list of applications, just after Select... and Select Using LiveTouch...
8.	Click the Window drop-down arrow, and select System .
9.	Click the Object drop-down arrow, and select Text .
10.	Click the Action drop-down arrow, and select Concatenate .
	Note: The Parameters tab may have multiple parameters for an action. The direction of the parameter can be Input (indicated by →) or Output (indicated by ←). An input parameter provides values to an action (e.g. text for a field). An output parameter passes a value from an action (e.g. a variable that can be used in subsequent steps). All output parameters are required to have a variable selected to save the process.
11.	In the Parameters tab, in the Variable field, click the Select Variable icon.
	<i>The Select Variable dialog box appears.</i>
12.	In the left Navigation Tree, click the Client Name folder .

Tip: You can scroll down the list to select a variable. You can also type part of the name in the quick entry box at the top of the list. This narrows the list of variables to match the characters you have typed.

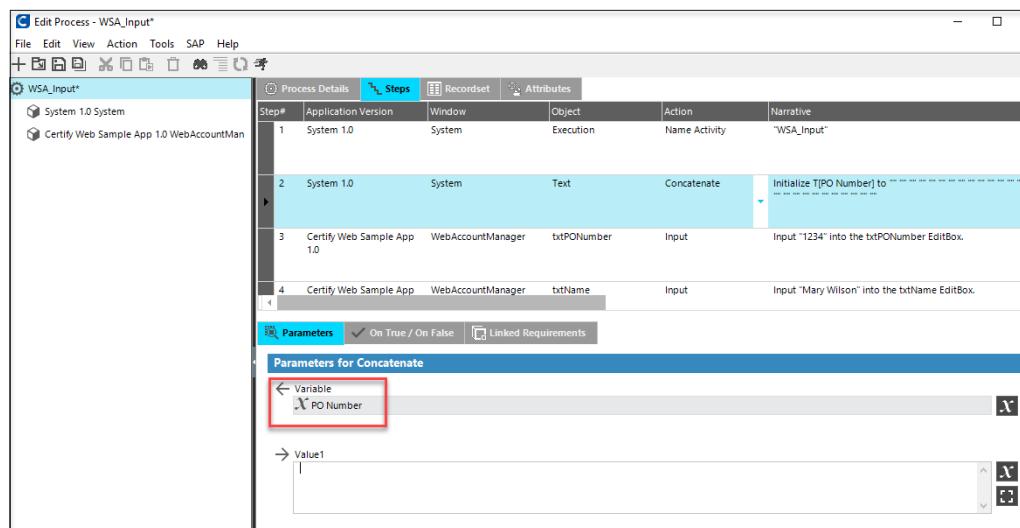
13. In the Variables pane, select **PO Number**.

14. Click **OK**.

The Process Editor reappears and the PO Number variable is added to the parameters for the WSA_Input process.

15. Click **Save** .

The Narrative for Step 2 has been updated with the PO Number variable.



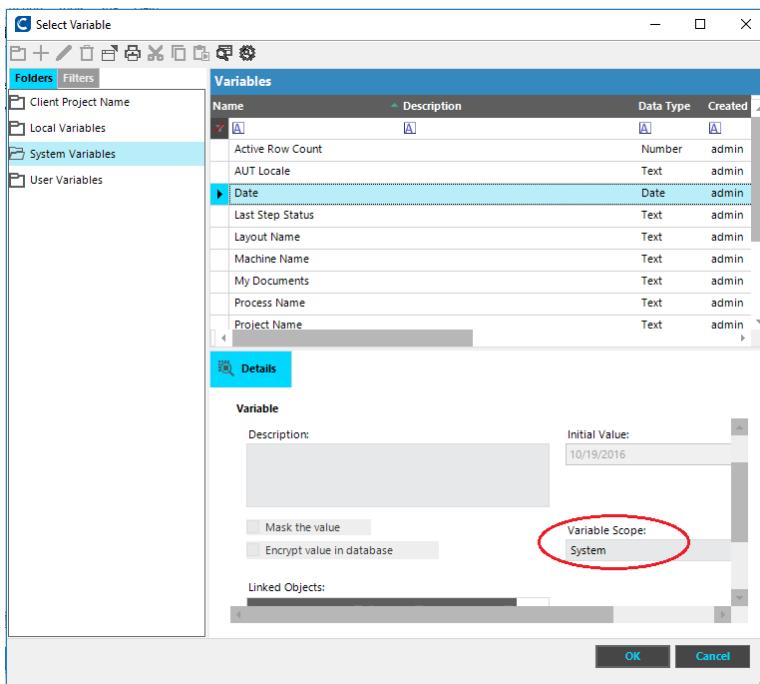
16. In the **Value1** field, enter a **4-digit number** as static data. In this example, the number 8403 has been used; this number will be added at the beginning of each unique PO number generated.

17. In the **Value2** field, click the **Select Variable**  icon.

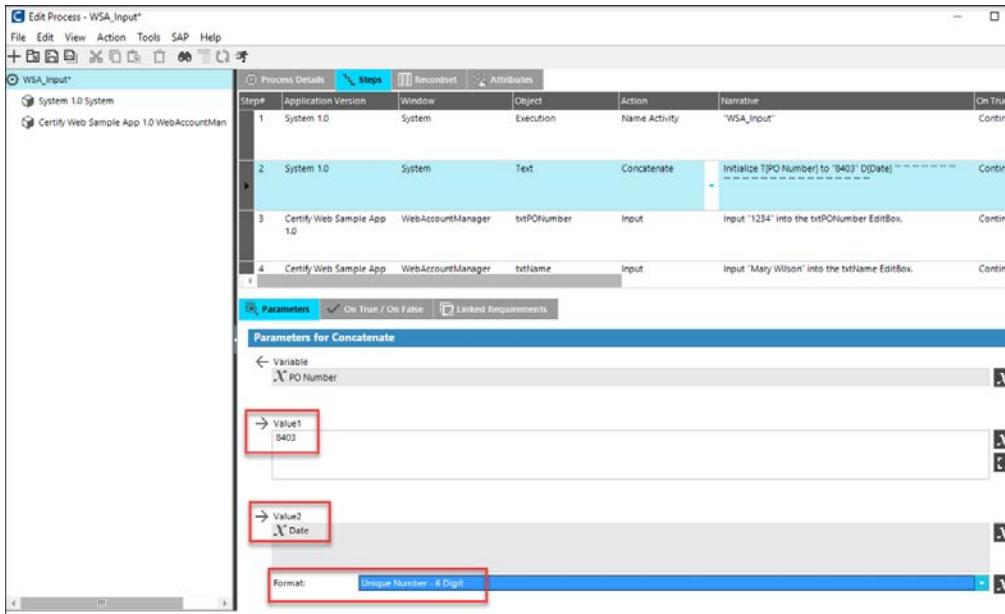
The Select Variable dialog box appears.

18. In the left Navigation Tree, click **System Variables**.

19. In the Variables pane, select **Date**.



20. Click **OK**.
21. Below the (V) Date field, click the **Format** drop-down and select **Unique Number – 6 Digit**.



22. Locate and select the **txtPONumber/Input step (step 3)**.
23. In the Parameters tab, in the **Value** field, click the **Select Variable** icon.

The Select Variable dialog box appears.

24. In the left Navigation Tree, click the **Client Name folder**.
25. In the Variables pane, select **PO Number**.
26. Click **OK**.
27. Click **Save** 

Execute Steps with Dialog

For these steps, we will use the Execute Steps with Dialog option. With this option, the Execution dialog box appears and allows us to step through the execution displaying variables, return status, and errors.

The dialog box is like the Process Execution dialog box which is explained in detail in the next lesson.

EXERCISE 3.6 —Executing Steps with Dialog to Show Variables

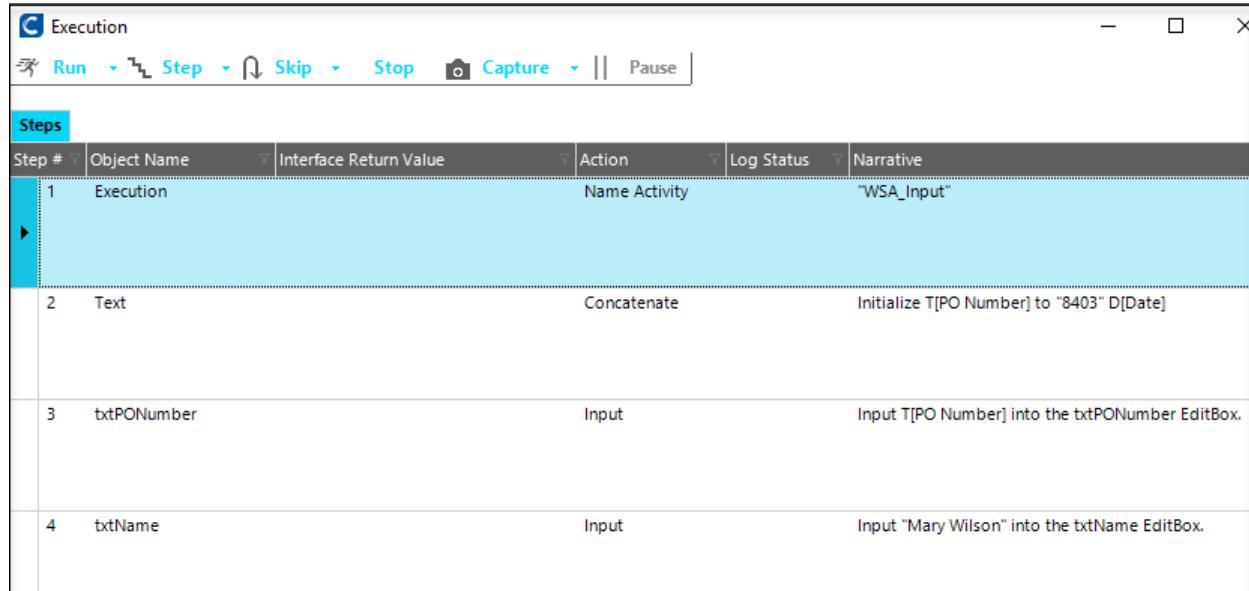
In this exercise, you will execute steps that set the PO variable in the WSA_Input process.

Step	Action
------	--------

1. To execute the steps:

- a. Locate and right-click the **WSA_Input** process, and select **Edit**.
- b. In the Steps area, select to highlight **Steps 1 through 4**. Step 4 is an extra step that will allow you to see the results for Step 3.
- c. In the Menus, click **Action**, and select **Execute Steps with Dialog** or **Shift+F6**.

The Execution dialog box appears.



2. Right-click anywhere in the table, and select Customize Columns.

Tip: Most Certify windows and dialog boxes that display a table of information can be customized to show/hide columns. The order of columns is not mandatory, so users can customize for their personal preference.

To show columns, select a column in the Hidden Columns field and click the Show button. The field moves to the Visible Column Order field.

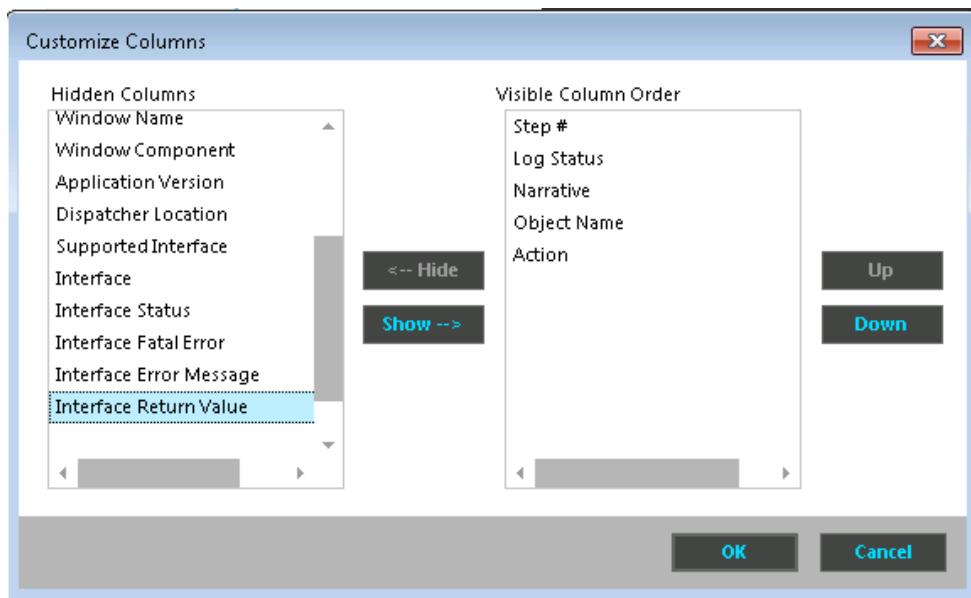
To hide columns, select a column in the Visible Column Order field and click the Hide button. The field moves to the Hidden Columns field.

To rearrange the order of the columns, select a column in the Visible Column Order field and use the Up and Down buttons to rearrange the order.

After you have completed your customization, click OK.

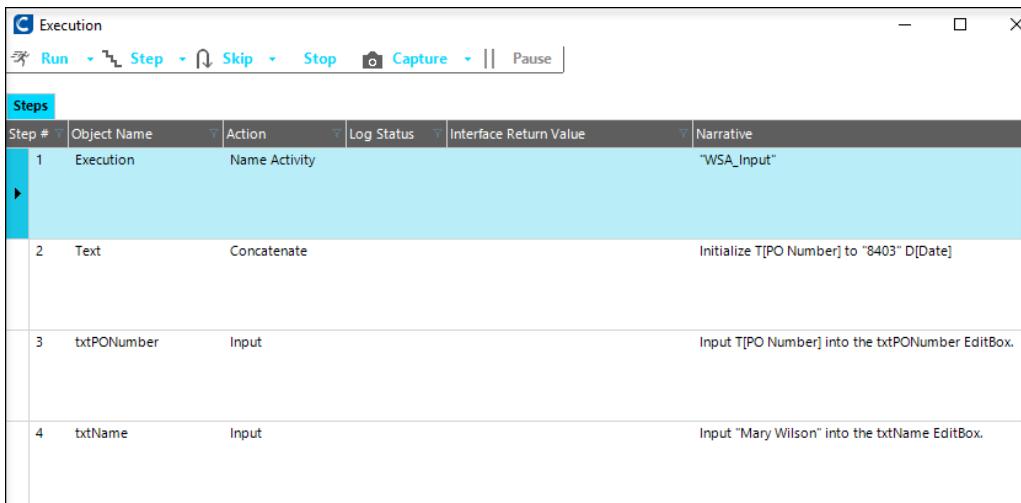
The columns in the pane appear per your selection(s).

3. Select the **Interface Return Value** column in the Hidden Columns field, and click the **Show** button.



4. Select the **Interface Return Value** column in the Visible Column Order field, and click the **Up** button to position the column just before the **Narrative** column.
5. Click **OK**.

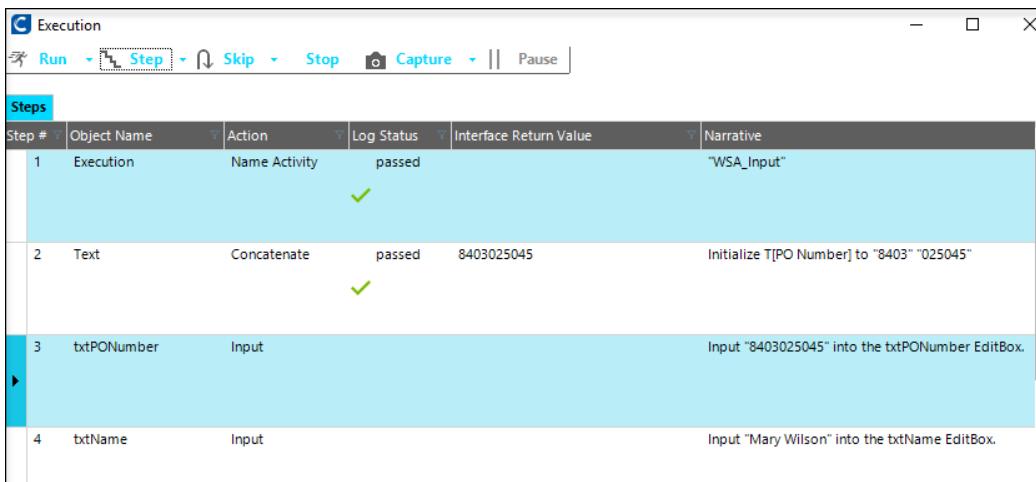
The columns in the pane appear per your selection.



Step #	Object Name	Action	Log Status	Interface Return Value	Narrative
1	Execution	Name Activity			"WSA_Input"
2	Text	Concatenate			Initialize T[PO Number] to "8403" D[Date]
3	txtPONumber	Input			Input T[PO Number] into the txtPONumber EditBox.
4	txtName	Input			Input "Mary Wilson" into the txtName EditBox.

6. In the Execution dialog box, click **Step**. Notice the first step is executed with the Log Status as **Passed**.
7. In the Execution dialog box, click **Step** again.

The second step executes. In the Execution dialog box, notice that the Log Status field displays a check mark and the word 'passed'. The Interface Return Value field displays the value that will be stored in the variable PO Number. The narrative for Step 3 also shows the value.



Step #	Object Name	Action	Log Status	Interface Return Value	Narrative
1	Execution	Name Activity	passed		"WSA_Input"
2	Text	Concatenate	passed	8403025045	Initialize T[PO Number] to "8403" "025045"
3	txtPONumber	Input			Input "8403025045" into the txtPONumber EditBox.
4	txtName	Input			Input "Mary Wilson" into the txtName EditBox.

8. Click **Step** to execute the next step.

The next step executes, and the Log Status and Interface Value Return fields are populated.

9. Click **Stop** to stop the execution.

Notice the status bar displays the number of steps executed and the results.

Layouts Overview

A **layout** is a collection of variables that define the data used by a process.

A layout can be visualized as the header row of a table. The layout for the table shown below would include the variables PO Number, Name, Ship to, Bill to, Material, Quantity_Numeric, and Price_Numeric.

PO Number	Name	Ship to	Bill to	Material	Quantity_Numeric	Price_Numeric

Recordsets Overview

A **recordset** contains data values for the variables defined in a layout. For example, a recordset for creating a new Purchase Order might contain multiple data values for PO Number, Name, Ship to, Bill to and so forth.

A recordset can be visualized as the detail rows of a table. The recordset for the table shown below would include two rows of data – one for Purchase Order 1234 and one for 1574.

1234	Mary Wilson	100 Fifth Street	1444 North Fifth Ave	Smart TV	10	780
1574	John Smith	123 Sing Lane	1455 North Main	Phone	7	450

When a recordset is used with a process, the process will execute one time for each row of data in the recordset. After completing a test with the first row of data, the process will execute again with the second row of data.

A layout can have one or more recordsets associated with it. Separate recordsets can be created for each test scenario needed. For example, a layout that contains Purchase Order information might have 3 recordsets: one that creates the initial set of data, another that adds data for regression testing, and one that adds data specifically to test new features.

The recordset for the table shown below would include three rows of data – and the process would execute three times.

14524	Gerardo Jones	10 Elm Street	11 Fifth Street	Keyboard	70	45
96586	Dustin Brown	5 Wind Lane	754 N Central Road	Mouse	52	30
854796	Mark Ebbert	1 Zephyr Road	143 George Bush Tpk	Speakers	^	12

The caret (^) character is used to designate the skip character for any value in a recordset. Any step that refers to a recordset whose value contains this character is skipped. For example, if certain fields or objects are enabled or disabled based on data values, the skip character can be used in a recordset to designate when disabled fields should be skipped.

Recordset Modes

Recordsets can be used in different modes depending on how the data should be handled during execution. By default, a recordset is used in Read Only mode so that at the end of execution the data in the recordset is the same as when the execution started.

Mode	When Executed	How Executed
Read Only	Reads recordsets at the beginning of execution.	Loops process once for each row until the end of the file.
Append	Writes recordsets at the end of execution.	Appends to existing recordset and loops process until Abort or Exit.
Clear and Append	Writes recordsets at the end of execution.	Creates new recordset for each execution session and loops process until Exit.
Read and Update	Reads recordsets at the beginning of execution.	Updates the recordset at the end of the process.

Defining Layouts in Certify

Layouts can be created in the following ways:

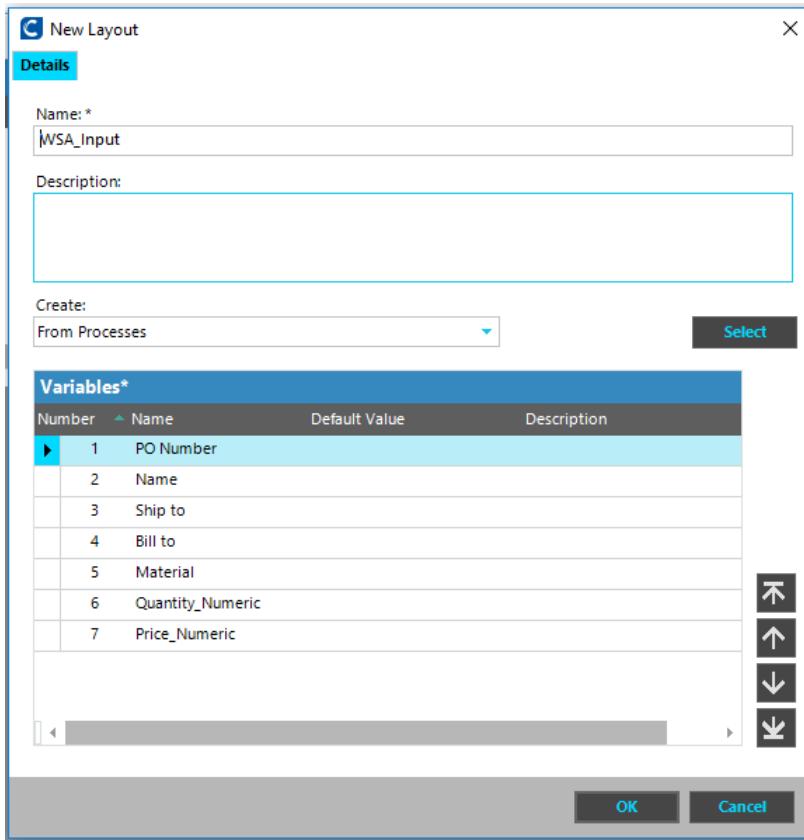
When you create a layout manually, you insert or add existing variables into the layout. Manually creating a layout requires you to select variables in a project and add or remove the variables in the layout. ****Note: A Job Aid example is provided in Lesson 5: Resources.**

When you create a layout from a process, all variables used by the chosen process are added to the layout. If the process has sub-processes, their variables can be included or excluded. ****Note: A Job Aid example is provided in Lesson 5: Resources.**

When you create a layout from steps (Add to Layout), steps are selected and a variable is created or selected for the parameters in the step. These variables are added to the layout. A recordset is created with any data that you had in the step. This method will be discussed in another lesson.

The first two types of Layouts are created in the Data window. All three types can be managed in the Data window or indirectly through a process. When you click Data in the Navigation Taskbar, you see the existing layouts for the project within which you are working. From here, you can create new or modify existing layouts. Layout creation and modification takes place in the New/Edit Layout dialog box as shown below.

New/Edit Layout Dialog Box

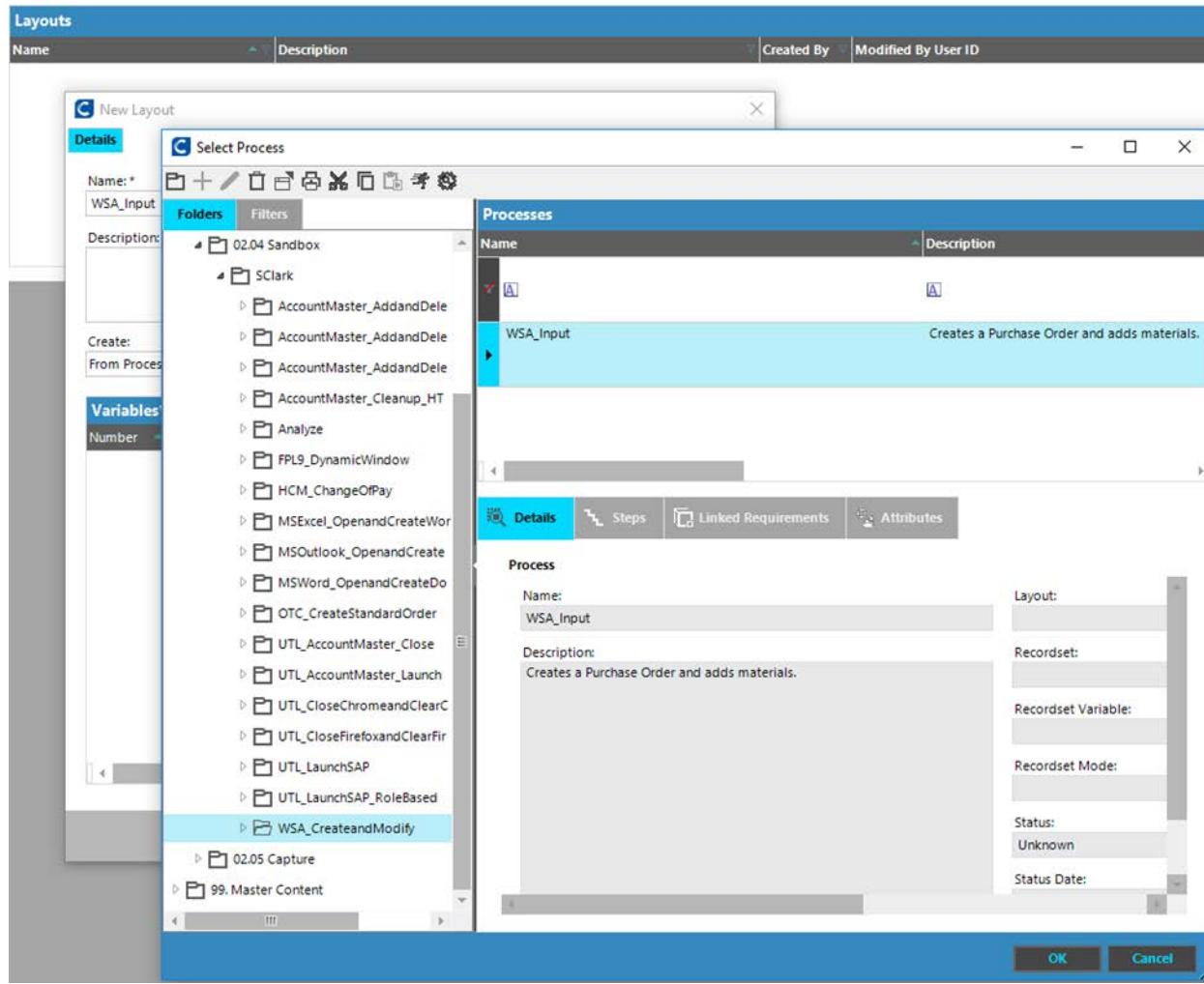


From the New/Edit Layout dialog box you specify a unique name and description for the layout. **As a best practice, layouts should be named the same as the process to which they are attached.** If a layout is created for the WSA_Input process, the layout should also be named WSA_Input.

After a name and description for the layout has been entered, variables can be added manually in the **Variables** pane or an option to add the variables from processes or objects can be selected from the **Create** list.

If one of the Process options is chosen, the Select Processes dialog box will open and after choosing a process, all the variables in that process will be added to the layout.

Select Processes Dialog Box



Creating a Layout Using Add to Layout

The Certify Add to Layout feature allows you to convert static data to variable data. If you have a process that has steps containing literal values or variables and no associated layout, you can create a layout for the process.

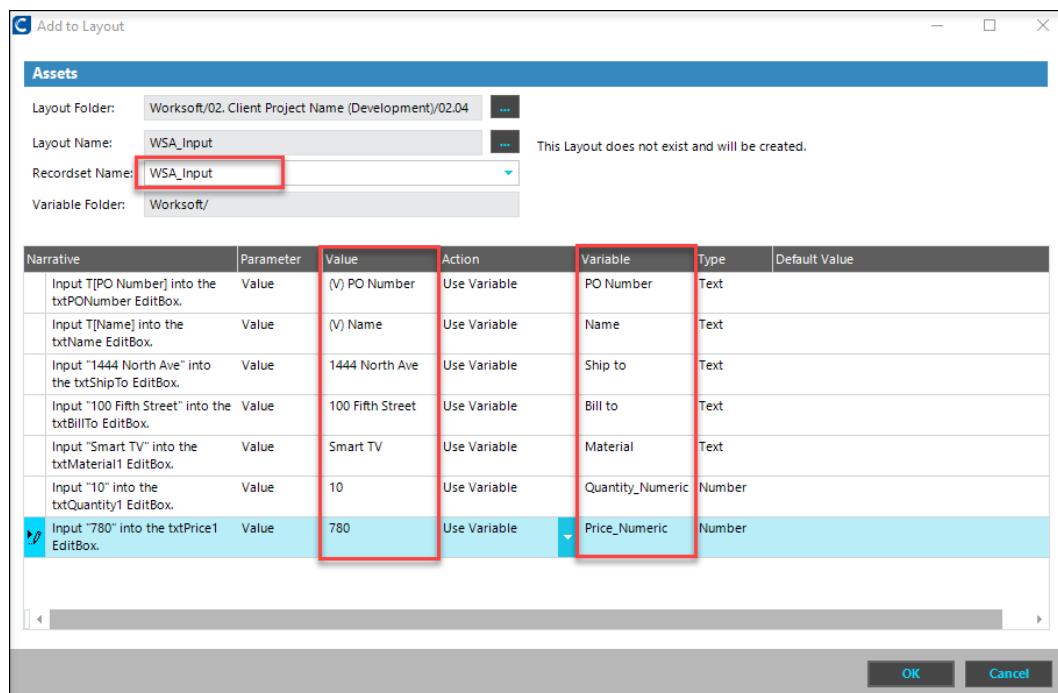
This is the simplest way to add variables/layouts/recordset and data to your process in a few simple steps. The feature is being introduced after you are familiar with this terminology.

You will be able to:

- Select steps of interest within the Steps tab.
- Replace literal values in those steps with variables.
- Create a layout for the process.
- Create a recordset for the layout.
- Collect variables from selected steps and add those variables to the layout.
- Create one row of recordset data with the values replaced by variables.

Add to Layout Window

The Add to Layout dialog box contains two sections: The Assets Pane and the Step Columns Area.



Assets Area

Layout Folder — Displays the folder path of the existing or new layout for the current process. If a new layout is being created, it will be created in the same folder path as the process.

Layout Name — Displays the name of the existing or new layout for the current process. If there is no layout associated to the process, the new layout will have the same name as the process.

Recordset Name — Allows you to input a name for the recordset. If it is an existing layout with recordsets, you can select a recordset from the drop-down list.

Variable Folder — Displays the folder where the new variable will be placed. The value in the field is the root variable folder of the project. A different location is rarely used.

Columns Area

The following table outlines the columns on the Add to Layout window.

Table 2 — Add to Layout Column Options

Column	Description
Narrative	Displays the narrative of the selected step. This column is read-only. If the step is not a typical Input or Verify step, the narrative will be prefixed by "????". If you see this prefix, this step may not be an appropriate step to drive the data.
Parameter	If the parameter can accept a variable, you can select a variable from the drop-down list. If the parameter cannot accept a variable, the drop-down list is not displayed, nor is it displayed if a parameter contains a user, local, or system variable.
Value	If the current value of the parameter contains a variable, the value appears as (V) plus the variable name. This column is read-only.
Action	Allows you to select one of the following values: Create Variable — This value will appear in the list if a variable needs to be created for the current parameter. Use Variable — This value will appear in the list if the current parameter value is a variable. Select Variable — This value will appear in the list if the current parameter value is a literal value. If you select this option, the Select Variable dialog box appears where you can select the variable to use in this parameter.

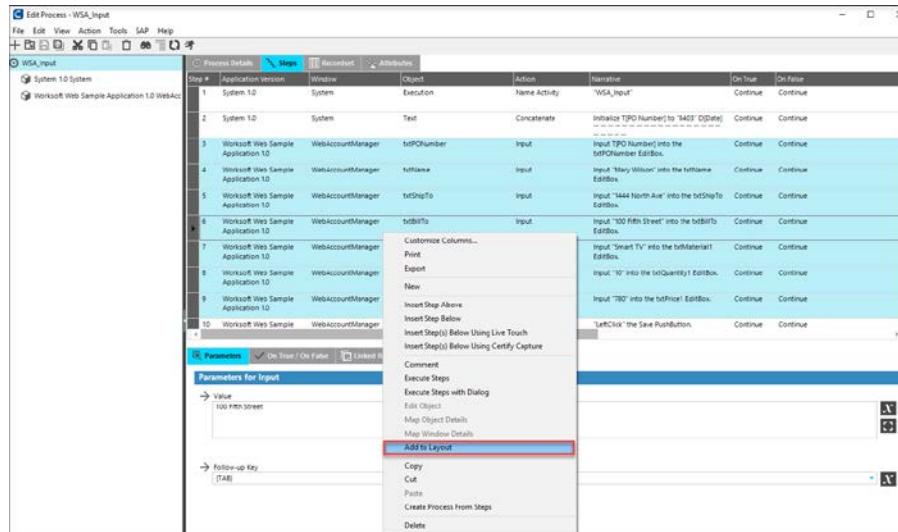
Column	Description
Variable	Shows the name of the variable to be used in this parameter. If the parameter already contains a variable, the column is not editable. If the parameter contains a literal value, you can edit the variable name. If you change the variable name, Certify will check to see if a variable by that name already exists in the specified variable folder. If the variable does exist, the Action column value is updated to Use Variable . If a variable does not exist, the Action column value is updated to Create Variable .
Type	Allows you to select the data type of the parameter. The available values in the list depend on the data type of the parameter. If the value for the Action column is Use Variable , this column displays the data type of the selected variable. If the Action column value is Create Variable , you can select the data type to use when creating the variable.

EXERCISE 3.7 — Creating a Layout Using Add to Layout

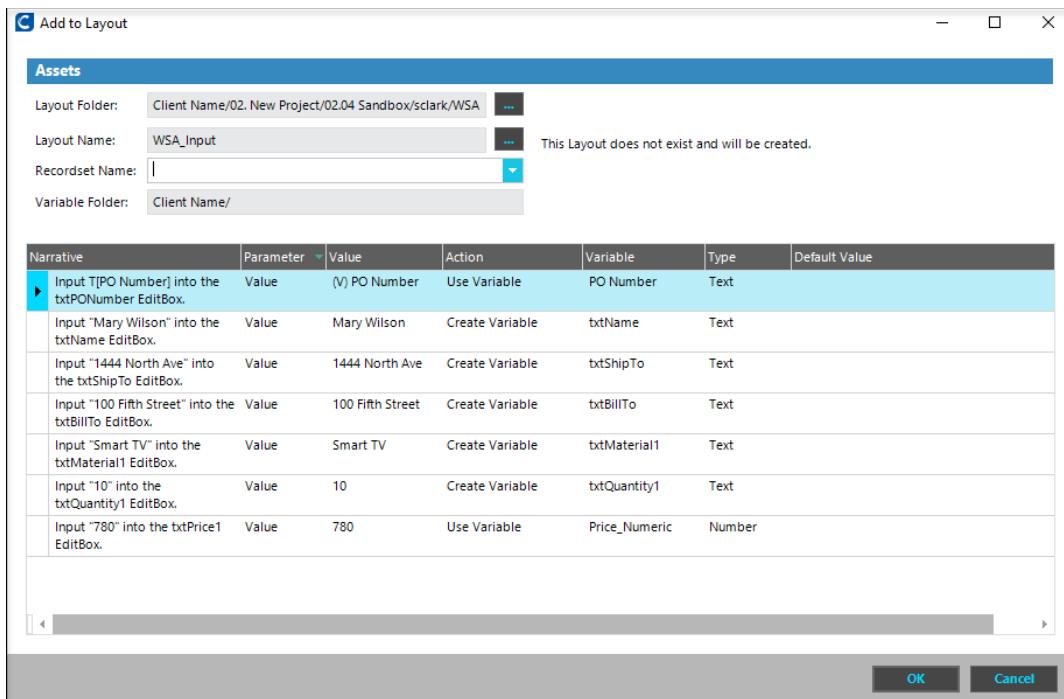
This exercise will show you how to create the **WSA_Input** Layout and Recordset using the Add to Layout feature to variabilize the data.

Step	Action
------	--------

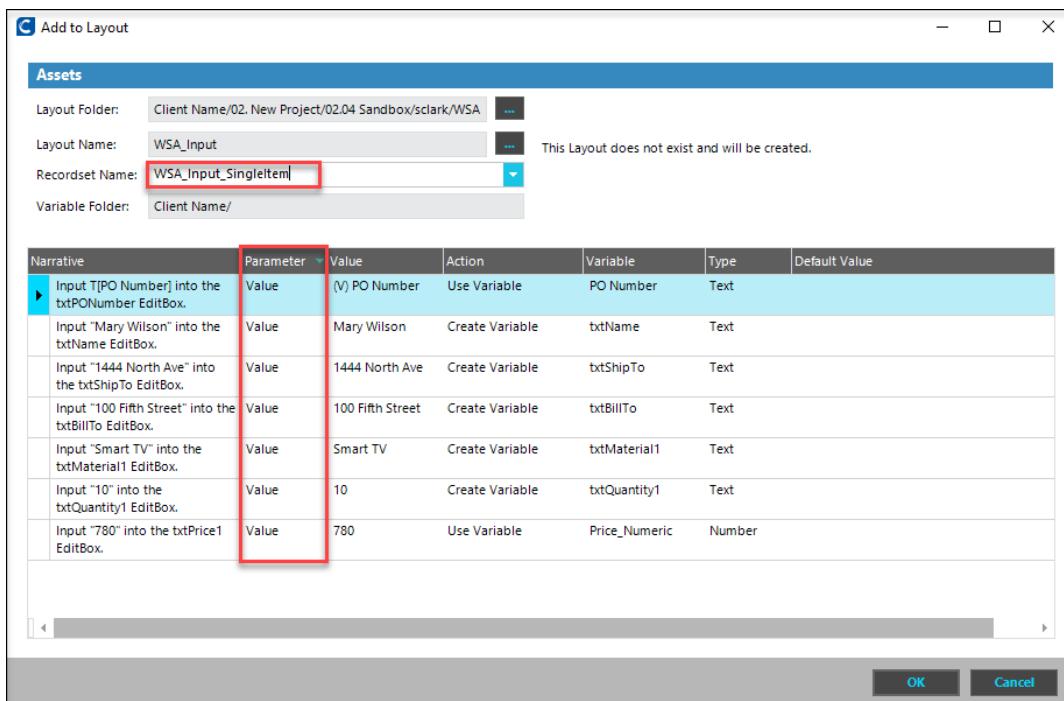
- In the **WSA_Input** process, select Steps 3-9. Right click and select **Add to Layout**.



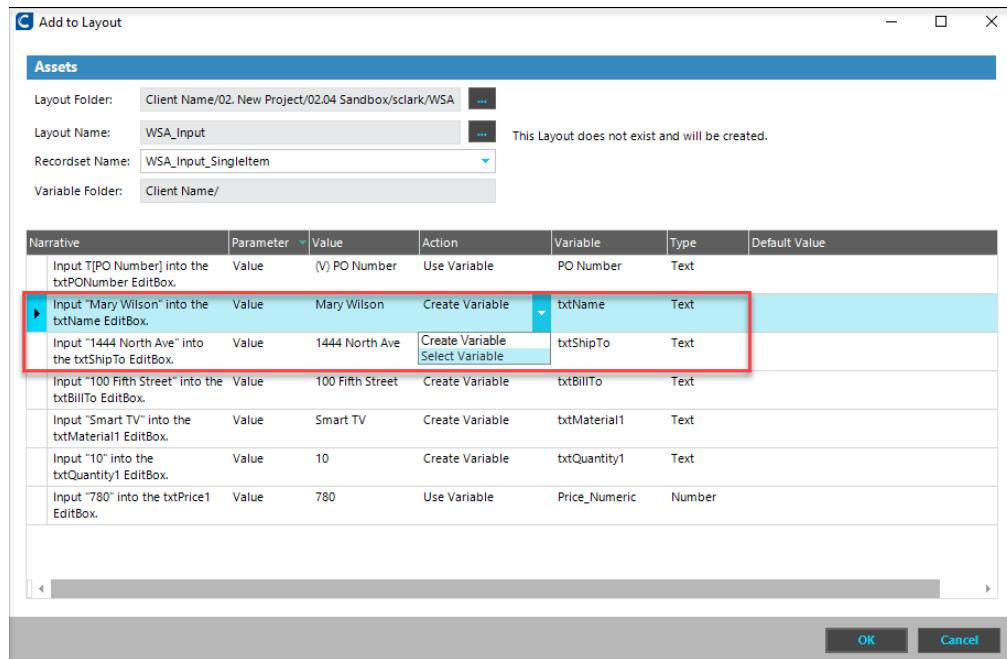
- The **Add to Layout** dialog box appears.



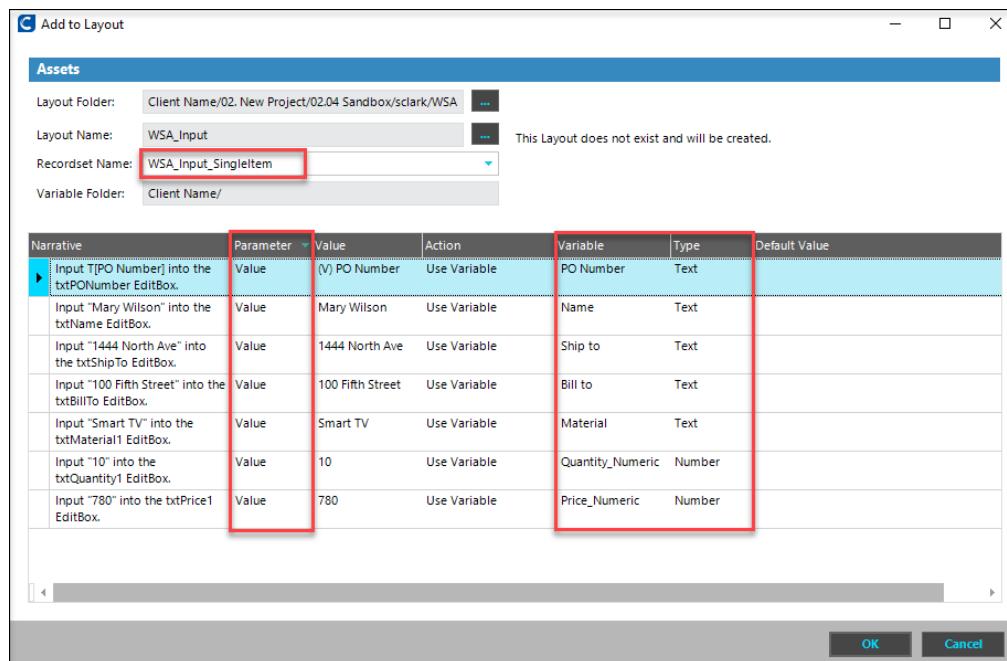
3. Modify the following in the **Add to Layout** dialog box.
4. Enter **WSA_Input_SingleItem** as the Recordset Name.
5. In the Parameter column for **each step**, verify **Value** is displayed.



6. For the second step: Open the **Action** drop-down and choose **Select Variable**. Select the **Name** variable.

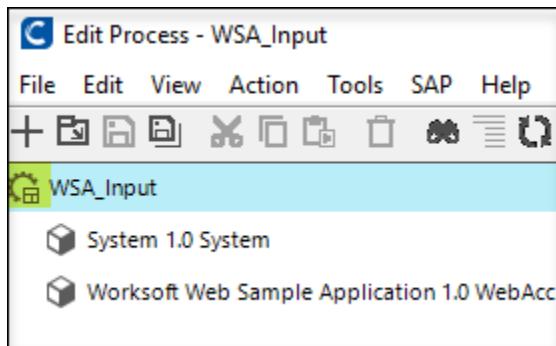


7. Use the **Action Select Variable** to choose the appropriate variable for each of the remaining steps.
Note: Even if the default action is "Use Variable", verify the variable name. If needed, use the Select Variable action to choose the correct variable name. Your **Add to Layout** dialog box should match exactly the screenshot shown below.



8. Click **OK**.
9. **Save** the process.
10. After saving, you will notice a recordset is attached to your **WSA_Input** process. This is

shown with a small paper sign  on the process gear in the Navigation Tree. ****Note:** If you do not see the small paper sign, refresh your process using the Refresh button in the toolbar.



11. Click on the **Recordset** tab. You should see a recordset with one row of data.

PO Number	Name	Ship to	Bill to	Material	Quantity_Numeric	Price_Numeric
1	Mary Wilson	1444 North Ave	100 Fifth Street	Smart TV	10	780
*						

12. Add one more record to the **WSA_Input_SingleItem** recordset as shown below.

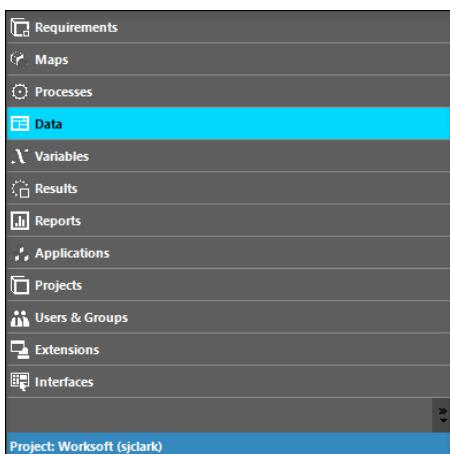
PO Number	Name	Ship to	Bill to	Material	Quantity_Numeric	Price_Numeric
1	Mary Wilson	1444 North Ave	100 Fifth Street	Smart TV	10	780
2	John Smith	123 Renner Lane	1455 North Main	Phone	7	480
*						

Note: While entering data into a recordset, use TAB to move from one column to another; this eliminates the chances of adding additional spaces or carriage returns.

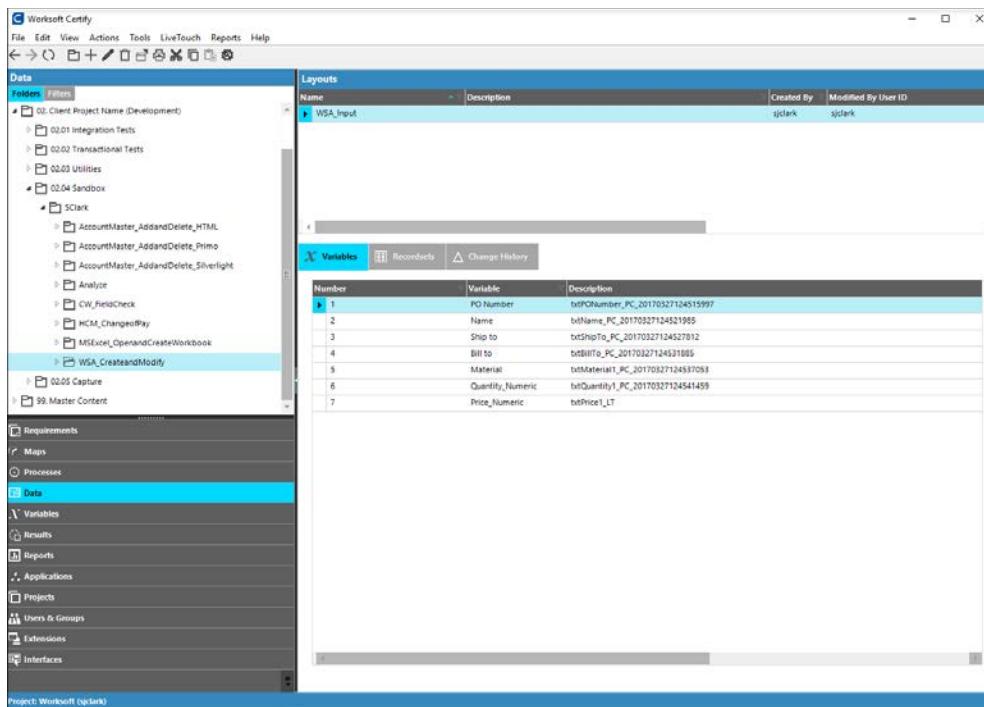
Important: If you don't see an empty recordset row, close and reopen the WSA_Input process.

13. Save the process.
14. To view or modify the layout and recordset, in the Navigation Taskbar, click **Data**.

The Data window appears.

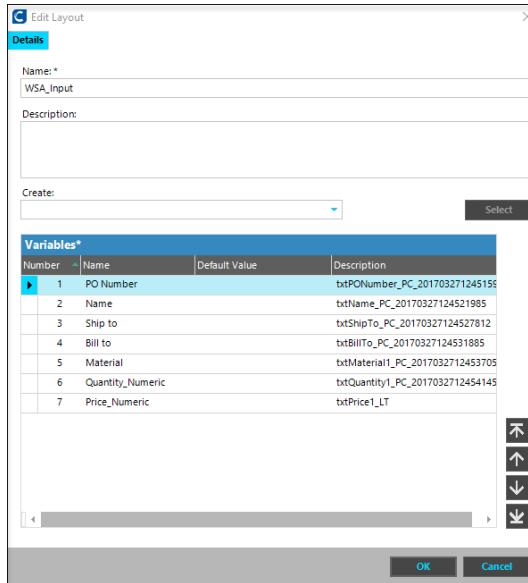


15. Select the **New Project** folder.
16. Select the **Sandbox** folder.
17. You will notice your Sandbox folder and WSA_CreateandModify subfolder were created automatically. If the folders do not exist, Certify creates them automatically when you use the **Add to Layout** option.
18. Select your **WSA_CreateandModify** folder.



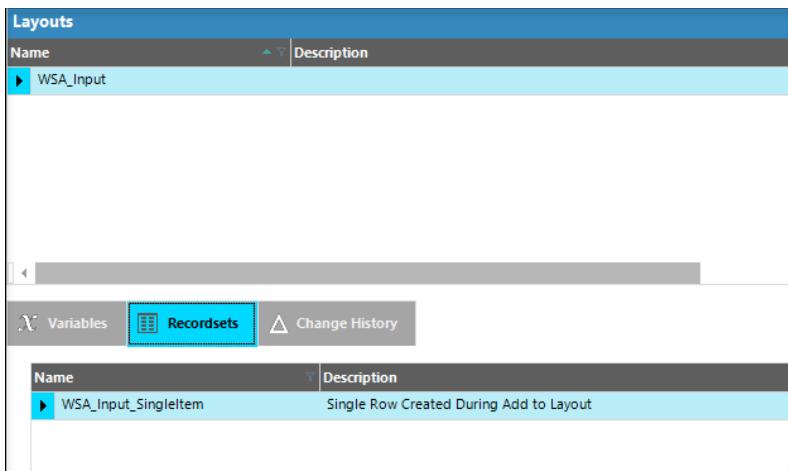
19. Right-click the **WSA_Input** layout, and select **Edit**.

The *WSA_Input* layout is displayed. Here you can add, modify, and delete variables.

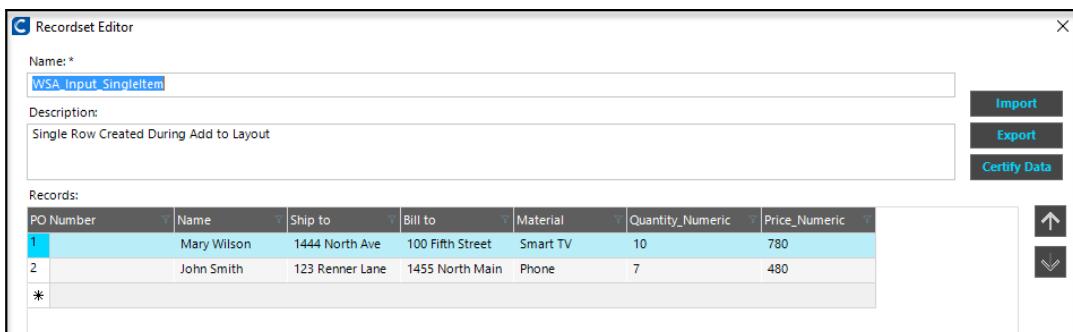


20. Click **OK** to close the layout.

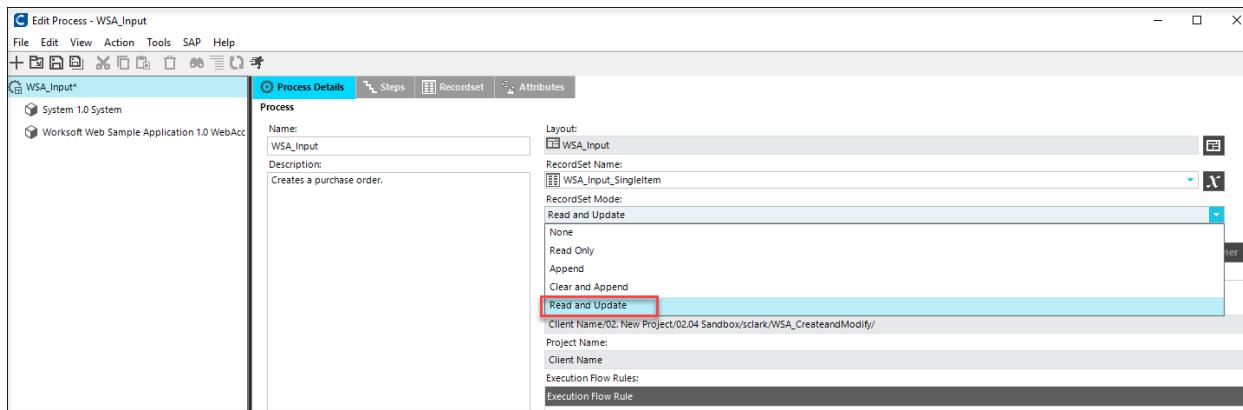
21. To view and edit the recordset, click the Recordsets tab and select the **WSA_Input_SingleItem** recordset.



22. Right-click the **WSA_Input_SingleItem** recordset and select **Edit**.
23. Your recordset should show two rows of data.



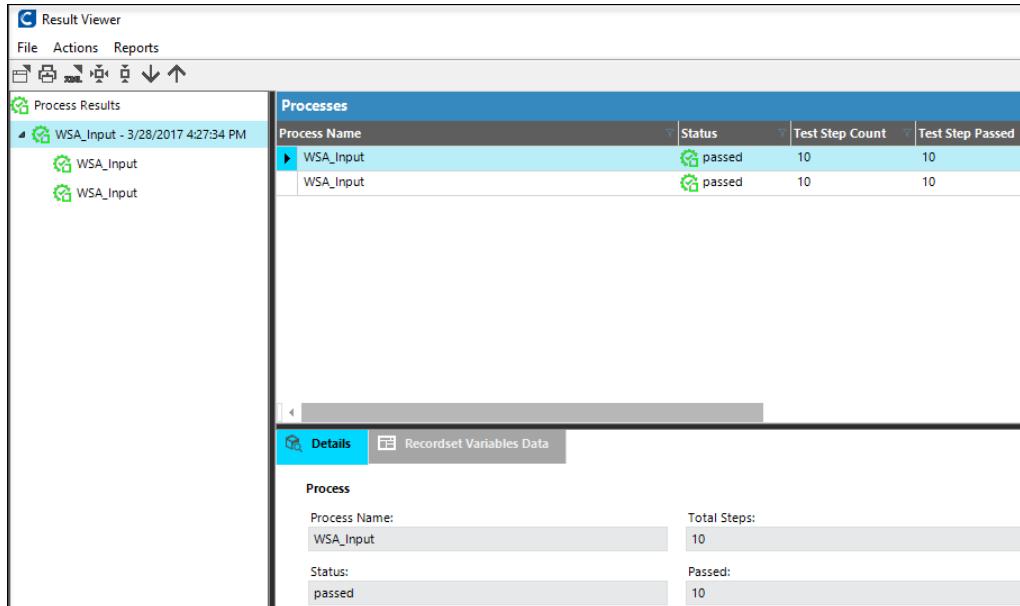
24. Click **OK** to close the Recordset Editor.
25. In the Navigation Taskbar, select **Processes** and navigate to your Sandbox folder. Select your **WSA_CreateandModify** folder.
26. Select and double-click your **WSA_Input** process to open it.
27. Click the **Process Details** tab.
28. Notice the **WSA_Input Layout** and **WSA_Input_SingleItem Recordset** are attached to your process.
29. Change the Recordset Mode to **Read and Update** as shown below.



The recordset mode Read and Update will read the data in the WSA_Input_SingleItem recordset to create two purchase orders. At the end of the execution, the recordset will be updated with the generated purchase order numbers.

30. Click .

The process executes and creates two purchase orders; each purchase order has a single material item.



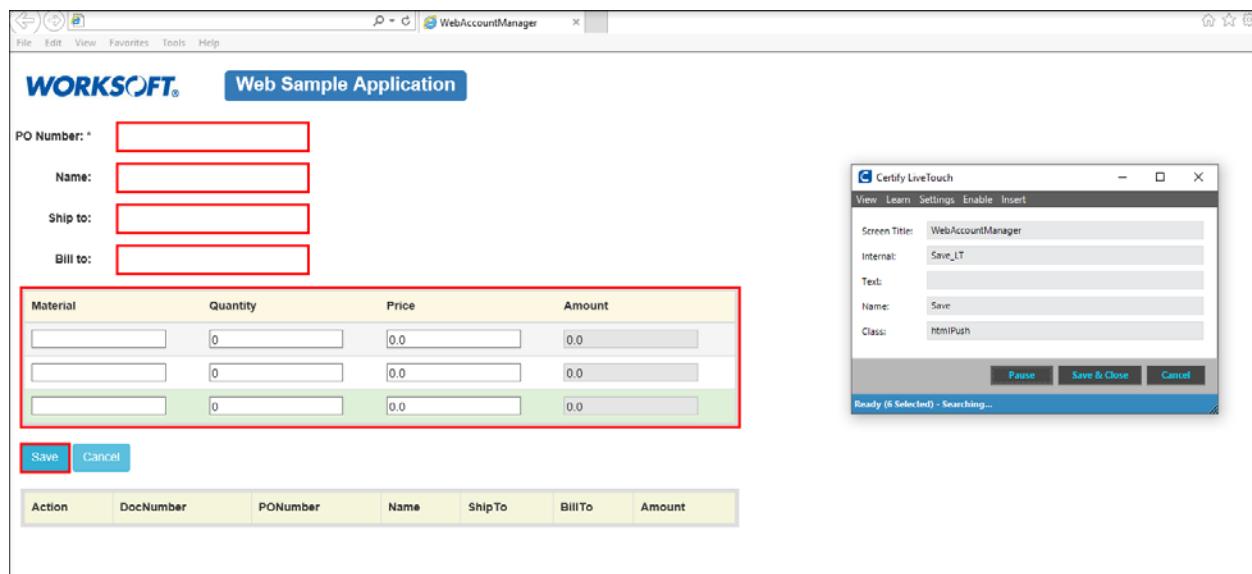
Certify LiveTouch Overview

The Certify LiveTouch feature allows you to select multiple objects in the application under test. You can create steps in a new process or insert steps into existing processes by clicking on one or more object(s) in the application under test.

LiveTouch inserts corresponding process steps for each of the selected objects. LiveTouch also pre-populates the appropriate step parameters with values entered or produced by the application under test.

As shown in Figure 5, from the Process and Data Editor, you can add steps using LiveTouch by right-clicking a step and selecting **Insert Step Below Using LiveTouch**. Certify will minimize and the application under test will appear, along with the Certify LiveTouch dialog box.

Figure 5 — Certify LiveTouch



LiveTouch allows you to learn multiple windows in one session. First, access a window and learn the objects on that window. Clicking the **Pause** button allows you to move to another window in your application. Lastly, click the **Start** button and learn the objects in the new window, all in the same session.

Notes:

- When LiveTouch is open and Ready, any click you make with the mouse is captured. If you need to move windows or start a new application, press Pause before using your mouse.
- Do not try to click any objects until the LiveTouch window displays Ready.
- After you have clicked objects, wait for the message "Ready (x Selected)" to equal the number you clicked before pressing "Save & Close".
- Clicking an object that you have previously selected will deselect it.

Key Functions of Certify LiveTouch

When Certify LiveTouch adds steps to your process, it performs many functions almost instantaneously. However, there are several other key functions of Certify LiveTouch:

Adds an object(s) and step(s) to your Certify process.

Attempts to pull the object and corresponding window from the Certify database. If the object doesn't exist in the database, Certify LiveTouch creates the object and adds it to the database and corresponding window.

If the window doesn't exist, Certify LiveTouch creates the window and corresponding object.

Certify LiveTouch Menu Options

Table 3 — Certify LiveTouch Menu Options

Menu	Option	Description										
View	Fields	<p>Displays the following options:</p> <table> <tr> <td>Application</td><td>Text</td></tr> <tr> <td>Program</td><td>Name</td></tr> <tr> <td>Transaction</td><td>Physical</td></tr> <tr> <td>Screen Title</td><td>Location</td></tr> <tr> <td>Class</td><td></td></tr> </table> <p>This selection is continued throughout your session.</p>	Application	Text	Program	Name	Transaction	Physical	Screen Title	Location	Class	
Application	Text											
Program	Name											
Transaction	Physical											
Screen Title	Location											
Class												
Fields (Show when hovering mouse)	As you hover over objects in the application, the field information is displayed.											
Learn	Learn Objects as Needed	Learn new objects while adding steps to your process.										
	Save Objects to Application Version	Save the newly learned objects to a specific application version.										
Settings	Target Folders	Select a target folder for the variables. Note: This feature is only available to users who purchased Certify Data.										
	Use Variables for New Steps	Create new variables for the application's values. Note: This feature is only available to users who purchased Certify Data.										

Menu	Option	Description
Enable	List of interfaces supported by Certify LiveTouch	Only interfaces licensed can be enabled.

Preparing to Use Certify LiveTouch

To take advantage of the LiveTouch feature, you must do the following:

- Create a new process or open an existing process that you wish to modify.
- Open the application under test and navigate to the screen that you will use for your Certify process.

EXERCISE 3.8 — Modifying the WSA_Input Process to Insert Multiple Materials

In this exercise, you will modify WSA_Input process to add multiple materials using the table object.

Step	Action
1.	From the Processes window, in the Navigation Tree, click your WSA_CreateandModify folder.
2.	Right-click your WSA_Input process in the Summary Pane, and select Edit .

The Process Editor appears.

3.	Click the  Steps tab.
4.	Select steps 7-9, right-click, and select Delete .

The screenshot shows the 'Steps' tab of the Worksoft Certify interface. The process consists of ten steps:

- Step 3: Certify Web Sample App, WebAccountManager, txtPONumber, Input, Narrative: Input T[PO Number] into the txtPONumber EditBox. On True: Continue, On False: Continue.
- Step 4: Certify Web Sample App, WebAccountManager, txtName, Input, Narrative: Input T[Name] into the txtName EditBox. On True: Continue, On False: Continue.
- Step 5: Certify Web Sample App, WebAccountManager, txtShipTo, Input, Narrative: Input T[Ship to] into the txtShipTo EditBox. On True: Continue, On False: Continue.
- Step 6: Certify Web Sample App, WebAccountManager, txtBillTo, Input, Narrative: Input T[Bill to] into the txtBillTo EditBox. On True: Continue, On False: Continue.
- Step 7: Certify Web Sample App, WebAccountManager, txtMaterial1, Input, Narrative: Input T[Material] into the txtMaterial1 EditBox. On True: Continue, On False: Continue.
- Step 8: Certify Web Sample App, WebAccountManager, txtQuantity1, Input, Narrative: Input N[Quantity_Numeric] into the txtQuantity1 EditBox. On True: Continue, On False: Continue.
- Step 9: New_CertifyWebSampleApp 1.0, WebAccountManager, txtPrice1, Input, Narrative: (empty). On True: Continue, On False: Continue.
- Step 10: Certify Web Sample App, WebAccountManager, Save, Send Click, Narrative: (empty). On True: Continue, On False: Continue.

A context menu is open over Step 6, listing options such as 'Customize Columns...', 'Print', 'Export', 'New', 'Insert Step Above', etc., with 'Delete' highlighted with a red box at the bottom.

Important: We delete the steps for entering Material, Quantity, and Price to use the table object; this allows us to enter multiple values into these fields.

5. Click the **Save**  button to save the process.
6. Right-click Step 6 (step **txtBillTo/Input**), and select Insert Step(s) Below Using LiveTouch.

The screenshot shows the 'Steps' tab of the Worksoft Certify interface. A process flow is displayed with seven steps. Step 6 is currently selected. A context menu is open over this step, with the 'Insert Step(s) Below Using Live Touch' option highlighted.

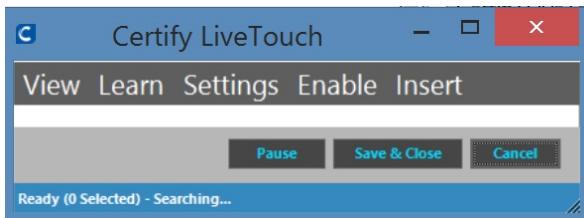
Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Execution	Name Activity	"WSA_Input"	Continue	Continue
2	System 1.0	System	Text	Concatenate	Initialize T[PO Number] to '8403' D[date] -----	Continue	Continue
3	Certify Web Sample App 1.0	WebAccountManager	txtPONumber	Input	Input T[PO Number] into the txtPONumber EditBox.	Continue	Continue
4	Certify Web Sample App 1.0	WebAccountManager	txtName	Input	Input T[Name] into the txtName EditBox.	Continue	Continue
5	Certify Web Sample App 1.0	WebAccountManager	txtShipTo	Input	Input T[Ship to] into the txtShipTo EditBox.	Continue	Continue
6	Certify Web Sample App 1.0	WebAccountManager	txtBillTo	Input	Input T[Bill to] into the txtBillTo EditBox.	Continue	Continue
7	Certify Web Sample App 1.0	WebAccountManager	Save		Save PushButton.	Continue	Continue

Parameters On True / On False Linked Requirements

Parameters for Input

- Value:
- Follow-up Key:

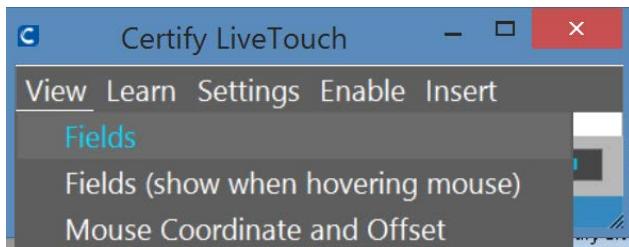
- Certify minimizes and the HTML window, along with the Certify LiveTouch dialog box appears. Before using LiveTouch to choose fields on the page, wait until LiveTouch loads and “Ready” appears at the bottom of the LiveTouch dialog box.



Note: With the Insert Step(s) Below Using LiveTouch option, you can select multiple objects from the window.

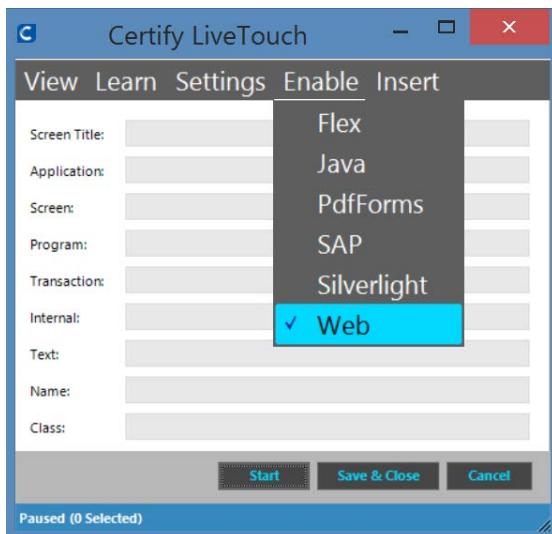
- Click the **Pause** button to pause LiveTouch.
- Pause will allow you to position the AUT screen and, change LiveTouch settings without LiveTouch trying to learn objects and create steps.
- Select the **Fields and Fields (show when hovering mouse)** items from the View menu.

This enables the Fields LiveTouch setting. The fields displayed vary depending on the type of objects being learned.

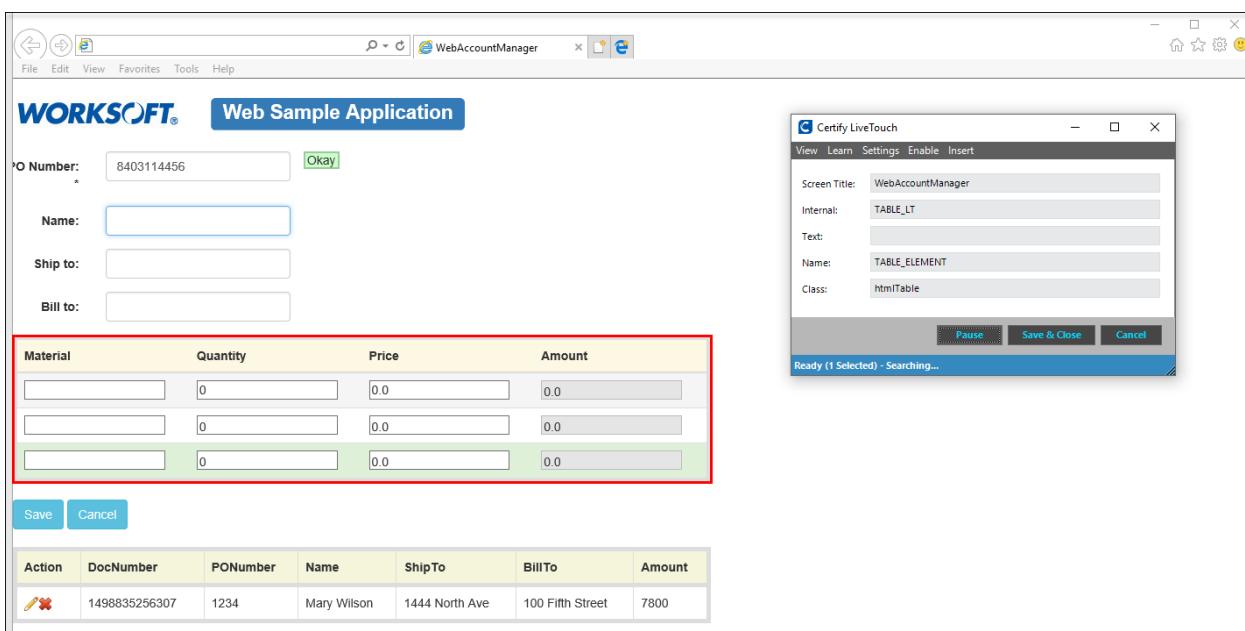


11. Select the **Enable** menu item. Click to disable any checked interfaces except for the Web interface.
12. Each interface that is licensed will display on this list. As a best practice, disable all interfaces that will not be used during the current LiveTouch session.

The *Enable* screen with only the *Web* interface enabled.



13. Click the **Start** button to start LiveTouch.
14. Wait for the status of LiveTouch to change to *Ready*.
15. LiveTouch the **TABLE** for Entering Material, Quantity, and Price as shown below.



16. Verify the LiveTouch status bar shows "Ready (1 Selected)", indicating the TABLE object was selected with LiveTouch.
17. Click **Save & Close**.
18. A new Step 7 is inserted as shown below.

Step #	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Execution	Name Activity	"WSA_Input"	Continue	Continue
2	System 1.0	System	Text	Concatenate	Initialize T[PO Number] to "8403" D[Date]	Continue	Continue
3	Worksoft Web Sample Application 1.0	WebAccountManager	txtPONumber	Input	Input T[PO Number] into the txtPONumber EditBox.	Continue	Continue
4	Worksoft Web Sample Application 1.0	WebAccountManager	txtName	Input	Input T[Name] into the txtName EditBox.	Continue	Continue
5	Worksoft Web Sample Application 1.0	WebAccountManager	txtShipTo	Input	Input T[Ship to] into the txtShipTo EditBox.	Continue	Continue
6	Worksoft Web Sample Application 1.0	WebAccountManager	txtBillTo	Input	Input T[Bill to] into the txtBillTo EditBox.	Continue	Continue
7	Worksoft Web Sample Application 1.0	WebAccountManager	TABLE_ELEMENT	Find Row	Find a row in the TABLE_ELEMENT Table that contains	Continue	Continue
8	Worksoft Web Sample Application 1.0	WebAccountManager	Save	Send Click	"LeftClick" the Save PushButton.	Continue	Continue

Modify the step as follows:

19. Change the Action of Step 7 from Find Row to Find Row (Advanced).

7	Worksoft Web Sample Application 1.0	WebAccountManager	TABLE_ELEMENT	Find Row (Advanced)	Find a row in the TABLE_ELEMENT Table column "" with "innerText" that Is Equal To "	Continue	Continue
8	Worksoft Web Sample Application 1.0	WebAccountManager	Save	Send Click	"LeftClick" the Save PushButton.	Continue	Continue

20. In the parameters section, for **Store Found Row Number In, click . Select the _Row.**

Parameters for Find Row (Advanced)

← Store Found Row Number In _Row

Format:

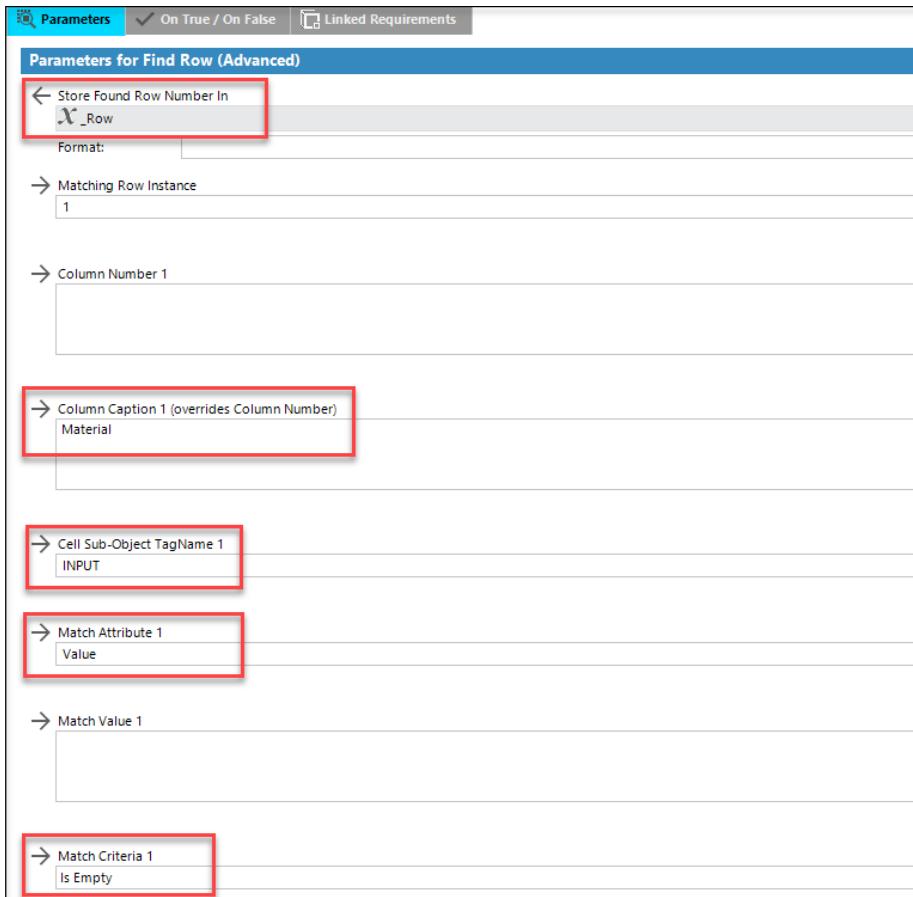
→ Matching Row Instance 1

21. Enter **Material in **Column Caption 1 (overrides Column Number)**. This will identify the Row number with the Material Column name.**

22. Select **INPUT from dropdown for **Cell Sub-Object TagName 1**.**

23. Select **Value** from **Match Attribute 1** dropdown.
24. Select **Is Empty** from drop down for **Match Criteria 1** parameter. This will identify the empty row and store that Row number in **_Row** variable.

The parameters for your Step 7 should match as follows:



Now, we will create steps to insert values into each column: Material, Quantity, and Price.

25. Right-click on Step 7, and select **Insert Step Below**. This will create a copy of Step 7 and insert it as Step 8.
26. For Step 8, change the Action to **Input Into Cell**.
27. Modify the parameter for **Row Number** by clicking on **X** to select the **_Row** variable. This will take the empty row identified in Step 7 and use it to insert values into the Material Column.

28. Type *Material* into the **Column Caption (overrides Column Number)** parameter. This will identify the column with the Material caption.
29. Enter the value *Smart TV* in the **Value** parameter.

Your Step 8 parameters should match the following:

The screenshot shows the 'Parameters' dialog box for an 'Input Into Cell' step. The 'Row Number' parameter is set to '_Row'. The 'Column Caption (overrides Column Number)' parameter is set to 'Material'. The 'Value' parameter is set to 'Smart TV'.

Parameter	Value
Row Number	_Row
Column Caption (overrides Column Number)	Material
Value	Smart TV

We will add two more steps to insert values into the Quantity and Price columns.

30. Right-click on Step 8, and select **Insert Step Below**.
31. For Step 9, change the Action to **Input Into Cell**.
32. Modify the parameter for **Row Number** by clicking on **X** to select the **_Row** variable. This will take the empty row identified in Step 7 and use it to insert values into the Quantity Column.
33. Type *Quantity* into the **Column Caption (overrides Column Number)** parameter. This will identify the column with the Quantity caption.
34. Enter the value *10* in the **Value** parameter.

Your Step 9 parameters should match the following:

The screenshot shows the 'Parameters' dialog box for the 'Input Into Cell' action. The 'Row Number' parameter is set to '_Row', 'Column Caption (overrides Column Number)' is set to 'Quantity', and 'Value' is set to '10'. Both 'Row Number' and 'Value' are highlighted with red boxes.

Parameter	Value
Row Number	<input type="text" value="X_Row"/>
Column Caption (overrides Column Number)	<input type="text" value="Quantity"/>
Value	<input type="text" value="10"/>

35. Right-click on Step 9, and select **Insert Step Below**.
36. For Step 10, change the Action to **Input Into Cell**.
37. Modify the parameter for **Row Number** by clicking on **X** to select the **_Row** variable. This will take the empty row identified in Step 7 and use it to insert values into the Quantity Column.
38. Type *Price* into the **Column Caption (overrides Column Number)** parameter. This will identify the column with the Quantity caption.
39. Enter the value *785* in the **Value** parameter.

Your Step 10 parameters should match the following:

Searched Parameters On True / On False Linked Requirements

Parameters for Input Into Cell

→ Row Number
X _Row

Format:

→ Column Number

→ Column Caption (overrides Column Number)
Price

→ Input Type
Input Text

→ Value
785

→ Value Criteria (only for Select From DropDown)
Is Equal To

→ Follow-up Keystroke
None

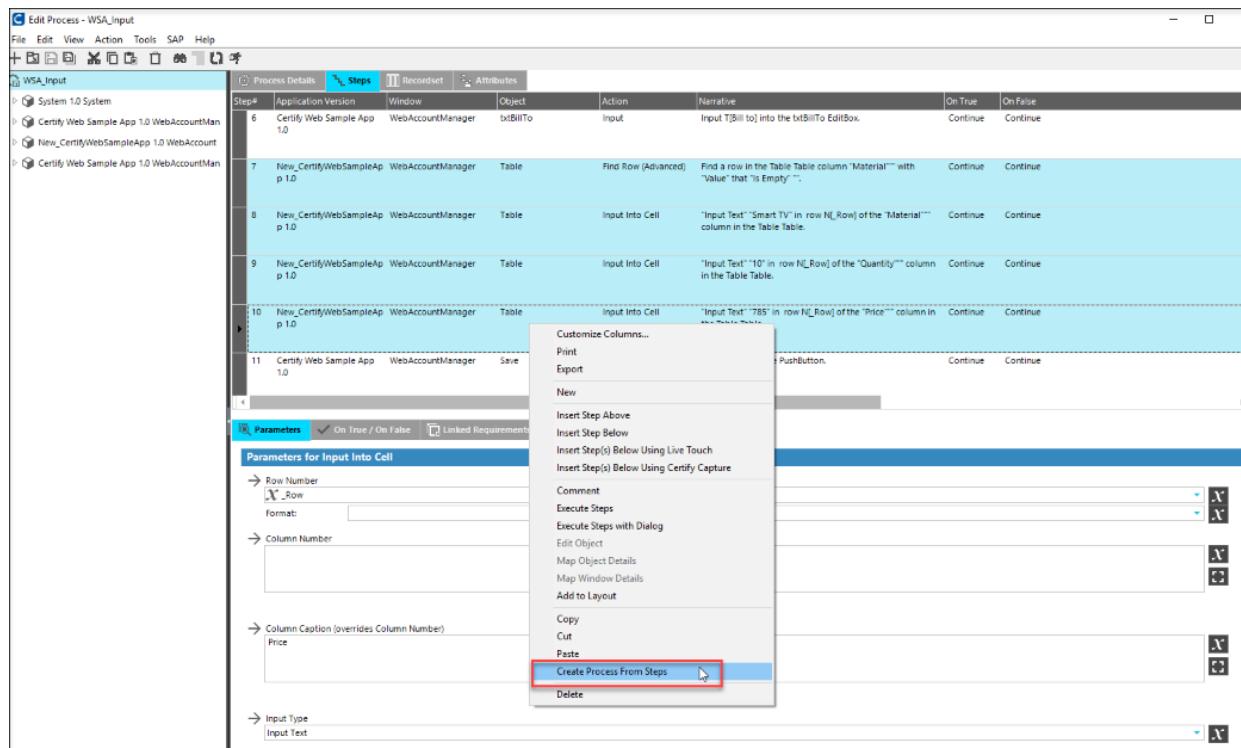
40. Click **Save** .

EXERCISE 3.9 — Creating the WSA_Input_C_Materials Process from Steps

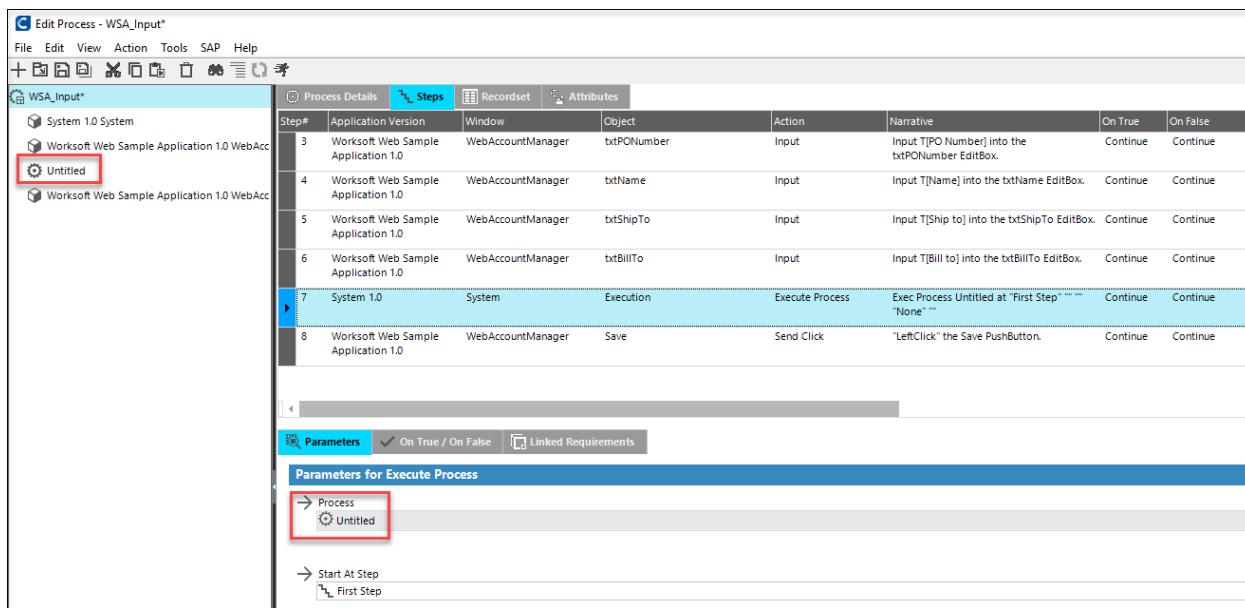
This exercise will show you how to create a sub-process from existing steps. We will use steps 7-10 of the WSA_Input process and create a child process to enter materials.

Step	Action
------	--------

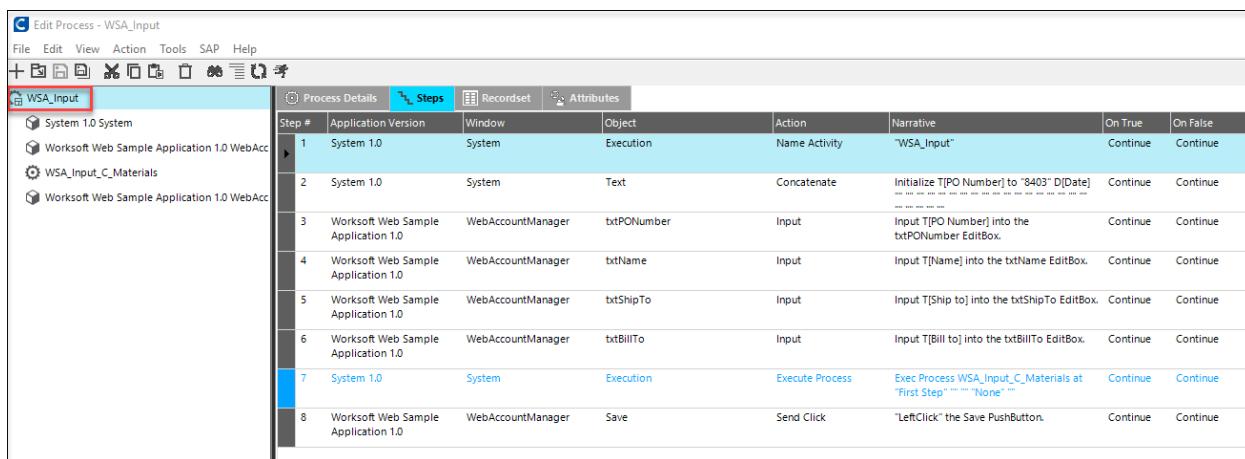
- In the **WSA_Input** process select **Step 7-10**. Right-click and select **Create Process From Steps**.



- When asked if you wish to continue creating the sub-process, click **OK**.
- A new process is created with the name **Untitled**, as shown below.



4. Click on the **Untitled** process in the Navigation Tree.
 5. When prompted, click **Yes** to save the process.
 6. In the **Process Details** tab, name the new process, **WSA_Input_C_Materials**.
- Enter the following description: *Child process used to add materials to a purchase order.*
7. In the Navigation Tree, select the **WSA_Input** parent process.

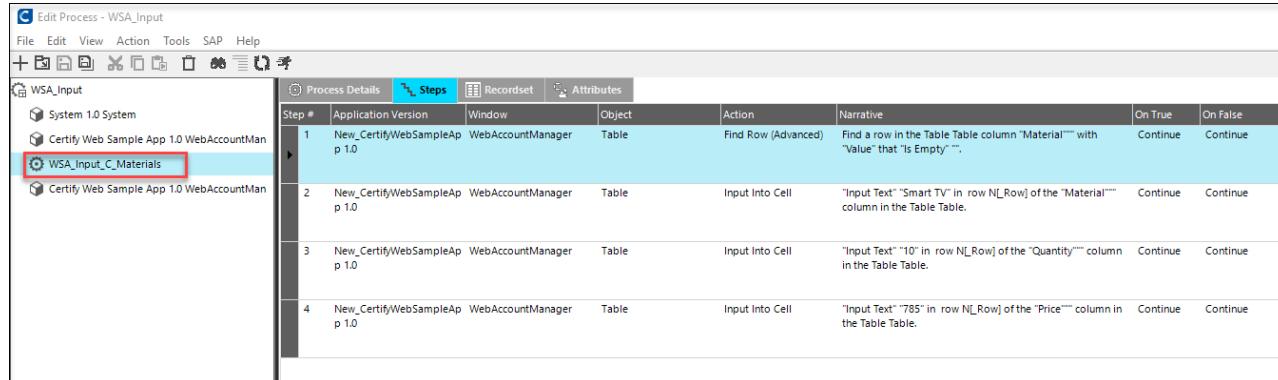


8. Save both processes.

EXERCISE 3.10 — Creating a Layout and Recordset for the WSA_Input_C_Materials Process

This exercise will show you how to create the WSA_Input_C_Materials Layout and Recordset using the Add to Layout feature to variabilize the data.

Step	Action
1. In the WSA_Input process, select the WSA_Input_C_Materials process in the Navigation Tree.	



2. In WSA_Input_C_Materials process, select Steps 2-4. Right-click and select **Add to Layout**.
3. The **Add to Layout** dialogue box appears.
4. Modify the **Add to Layout** dialogue box as follows:
 - a. Enter **WSA_Input_C_Materials_MultipleItems** in the Recordset Name field.
 - b. In the Parameter column for Steps 1-3, select **Value** from the drop-down.
 - c. Use the **Action Select Variable** to select the appropriate variables for each step: Material, Quantity_Numeric, and Price_Numeric.

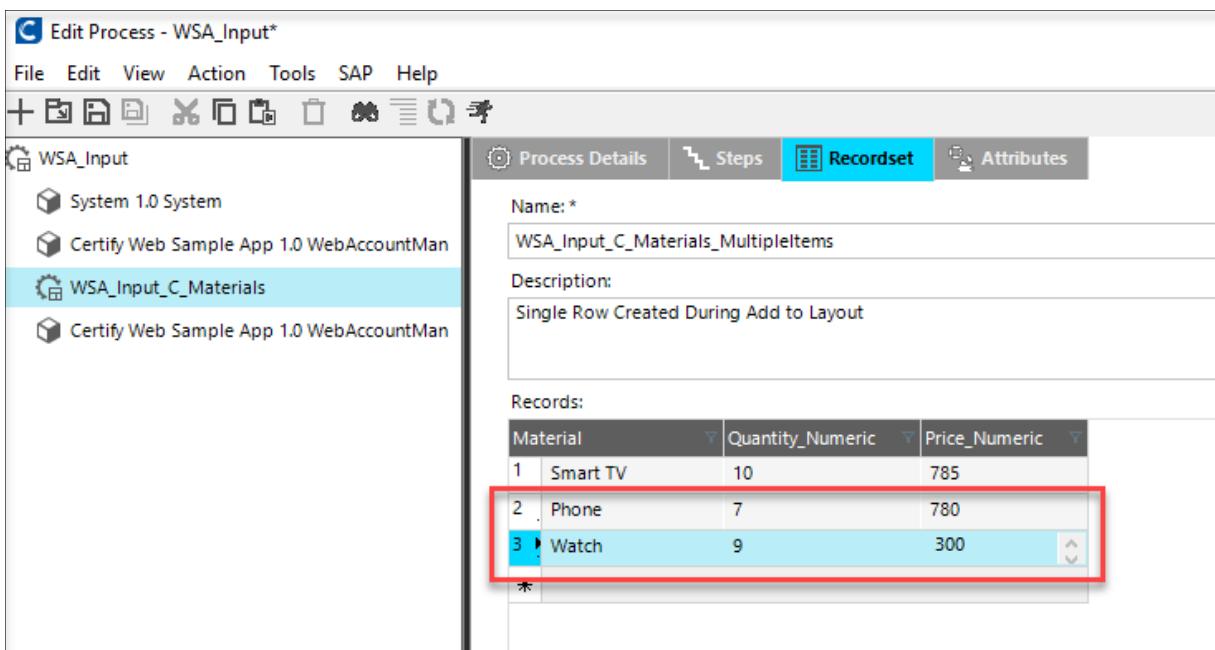
Your Add to Layout Screen should exactly match the screen shown below.

Narrative	Parameter	Value	Action	Variable	Type	Default Value
[???"Input Text" "Smart TV" in row N_Row] of the "Material"	Value	Smart TV	Use Variable	Material	Text	
[???"Input Text" "10" in row N_Row of the "Quantity"" colu	Value	10	Use Variable	Quantity_Numeric	Number	
[???"Input Text" "785" in row N_Row of the "Price"" colum	Value	785	Use Variable	Price_Numeric	Number	

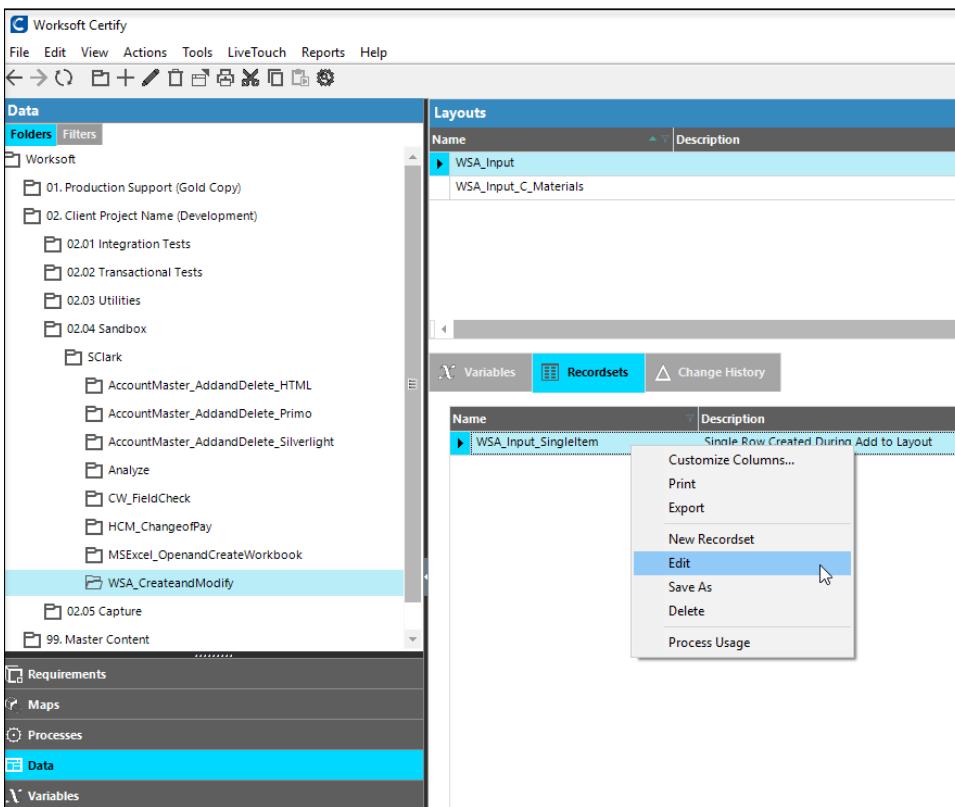
5. Click OK.
6. Save the process.
7. After saving, you will notice a recordset is attached to your **WSA_Input_C_Materials** process.
8. In the Navigation Tree, select the WSA_Input_C_Materials process. Click the Recordset tab in the Process Editor. You should see a recordset with one row of data.

Material	Quantity_Numeric	Price_Numeric
1 Smart TV	10	785
*		

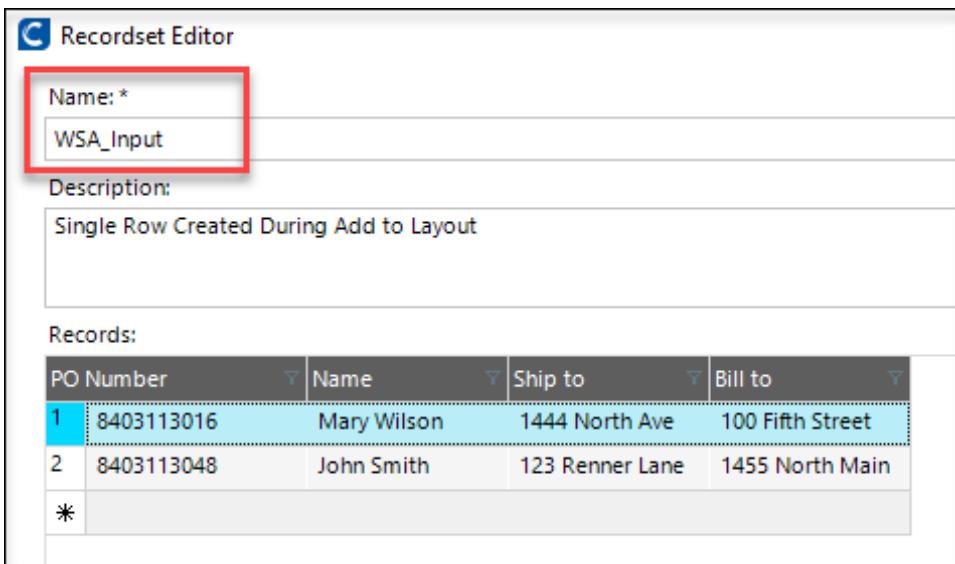
9. Add two more records to the **WSA_Input_C_Materials_MultipleItems** recordset.



10. Save the process.
11. Click on the **WSA_Input** process from the Navigation Tree, and click Save.
12. There is one last step we need to complete before we can execute our **WSA_Input** process. We need to delete the columns Material, Quantity_Numeric and Price_Numeric from the WSA_Input Layout since we have created a separate layout and recordset for those fields.
13. Click on **Data** in the Navigation Taskbar. Select your sandbox folder and then select your WSA_CreateandModify folder.
14. Right-click on the **WSA_Input** Layout and select **Edit**.
15. Select the **Material** variable, right-click, and **Delete**.
16. Select the **Quantity_Numeric** variable, right-click, and **Delete**.
17. Select the **Price_Numeric** variable, right-click, and **Delete**.
18. Click **OK** to close the Edit Layout window.
19. Select the Recordsets tab. Right-click your **WSA_Input_SingleItem** recordset, and select **Edit**.



20. Rename the recordset WSA_Input.



21. Click OK to close the Recordset Editor window.

Executing Processes that Use Data

You are ready to execute the WSA_Input process with data. The process will execute the WSA_Input process, execute the WSA_Input_C_Materials child process, and use their layouts and recordsets for data.

EXERCISE 3.11 — Executing Processes with Data

In this exercise, you will execute a process that has layouts and recordsets. This lesson uses  to run the entire process rather than executing a step at a time.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, click your WSA_CreateandModify folder.
3.	In the Summary Pane, right-click your WSA_Input process and select Run or press the "Running Man" button. <i>The Configuration dialog box appears.</i>
4.	Click Start . <i>The Execution dialog box appears.</i>
<p>Important: Make sure the Worksoft Web Sample Application is open before you start executing your process.</p>	
5.	Click Run . <i>The process executes, and the Result Viewer appears upon completion. Notice the Layout, Recordset, and Recordset Mode fields are populated in the Details tab of the Result Viewer.</i>
6.	When you are finished reviewing the results, click the  in the top right corner to close the Result Viewer.
7.	Open the Process Editor and make any needed corrections to failed steps.

EXERCISE 3.12 — Creating the WSA_SelectandVerify Process

In this exercise, you will create the WSA_SelectandVerify process in Certify which completes the following tasks:

- Selects the PO recently created.
- Stores the Document Number into a variable.
- Verifies “*An Item is created*” message from WSA.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, click the WSA_CreateandModify folder in Your Sandbox Folder .
3.	Right-click in the Summary Pane , and select New Process .
4.	Give the process a name and description: <ol style="list-style-type: none">In the Process section, in the Name field, type <code>WSA_SelectandVerify</code>.In the Description field, type Selects and verifies the PO recently created and stores the document number in a variable.Click the Steps tab. <p><i>You are now ready to add steps to the WSA_SelectandVerify process.</i></p>
5.	Use LiveTouch to create a step: <ol style="list-style-type: none">First, verify the Worksoft Web Sample Application is open.In Certify, right-click in the Steps area, and select Insert Step(s) Below Using LiveTouch.

Certify minimizes and the sample application, along with the Certify LiveTouch dialog box, appears.

The screenshot displays the WORKSOFT Web Sample Application interface. On the left, there is a form with fields for 'PO Number', 'Name', 'Ship to', and 'Bill to'. Below this is a table with columns 'Material', 'Quantity', 'Price', and 'Amount'. On the right, the Certify LiveTouch dialog box is open, showing settings for an internal table named 'tblList_LT'. The dialog box has tabs for 'View', 'Learn', 'Settings', 'Enable', and 'Insert'. It also has buttons for 'Start', 'Save & Close', and 'Cancel'. The status bar at the bottom of the dialog box indicates 'Paused (0 Selected)'.

c. Use LiveTouch to select the table with the saved Purchase Orders (**tblList**).

A red highlight appears around the object and the information appears in the Certify LiveTouch dialog box.

The screenshot shows the WORKSOFT Web Sample Application. The application form has fields for 'PO Number', 'Name', 'Ship to', and 'Bill to'. Below is a table with columns 'Material', 'Quantity', 'Price', and 'Amount'. On the right, the Certify LiveTouch dialog box is open with the 'Internal' field set to 'tblList_LT'. A red box highlights this entry. The dialog box also shows 'Screen Title: WebAccountManager', 'Text:', 'Name: tblList', and 'Class: htmlTable'. The status bar at the bottom of the dialog box says 'Ready (1 Selected) - Searching-'.

6. In the Certify LiveTouch dialog box, click **Save & Close**.

Next, we will modify Step 1 so Certify can identify the row number where the most recent PO is stored.

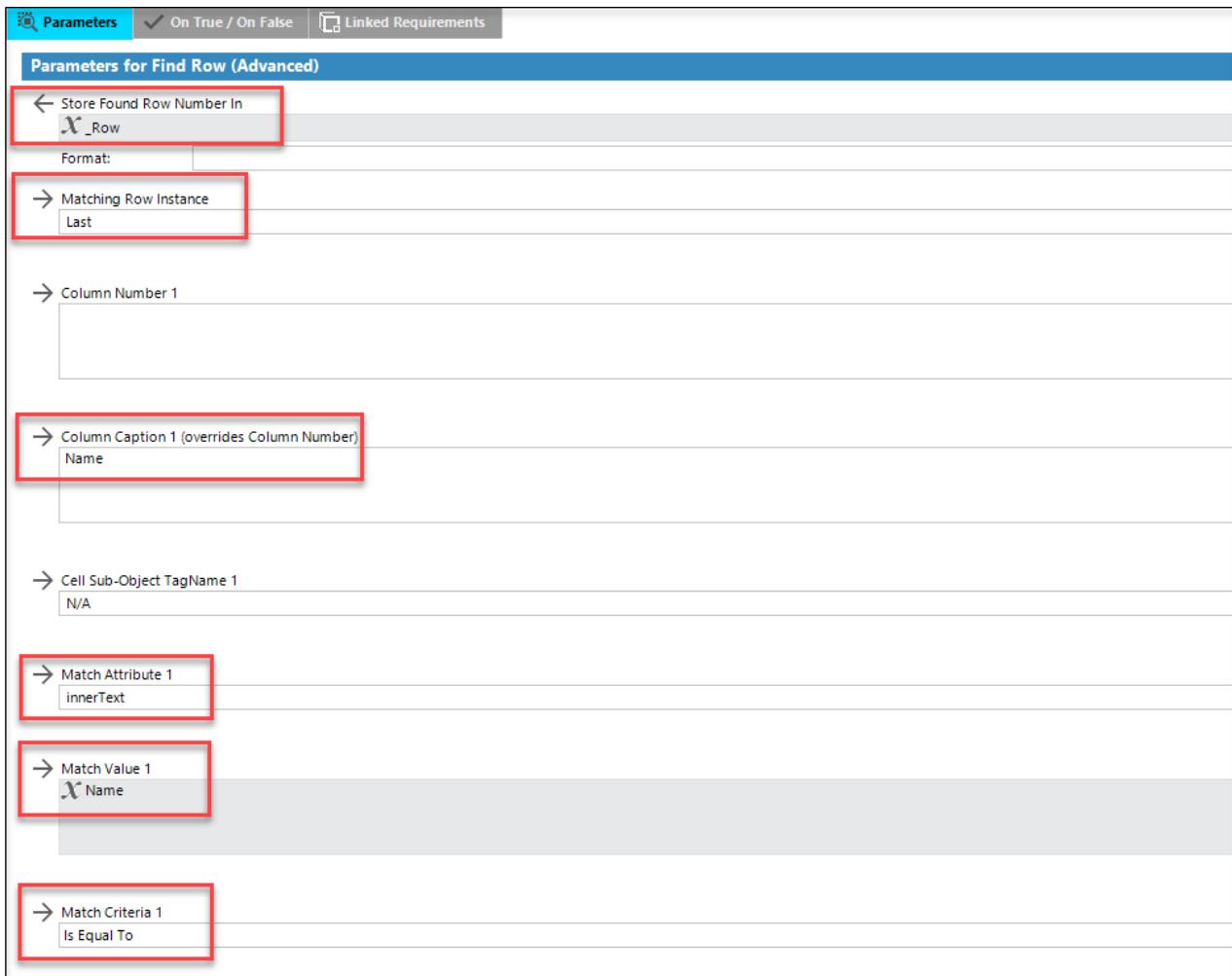
7. Select **Step 1**.
8. From the **Action** drop-down, select the **Find Row (Advanced)** option.
9. Modify the **Step 1** parameters as follows.

10. In the **Store Found Row Number In** field, click the **Select Variable**  icon.

The *Select Variable* dialog box appears.

11. In the Variables pane, select the **_Row** variable.
12. In the **Matching Row Instance** field, select **Last** from the drop-down. *The last row will always be where the most recent PO is saved.*
13. In the **Column Caption 1 (overrides Column Number)** field, type **Name**.
14. In the **Match Attribute 1** drop-down, select **innerText**.
15. In the **Match Value 1** field, select the **Name** variable.
16. In the **Match Criteria 1** field, select **Is Equal To** from the drop-down.

Your Step 1 parameters should match the following:



Next, we will insert a step to verify that the row identified in Step 1 contains the most recent PO.

17. Right-click on **Step 1**, and select **Insert Step Below**.
 18. From the **Action** drop-down, select the **Verify Cell** option.
 19. Modify the **Step 2** parameters as follows.
20. In the **Row Number** field, click the **Select Variable**  icon.
- The *Select Variable* dialog box appears.
21. In the Variables pane, select the **_Row** variable.
 22. In the **Column Caption 1 (overrides Column Number)** field, type **PONumber**.

23. In the **Verify Type** drop-down, select **Cell Text**.

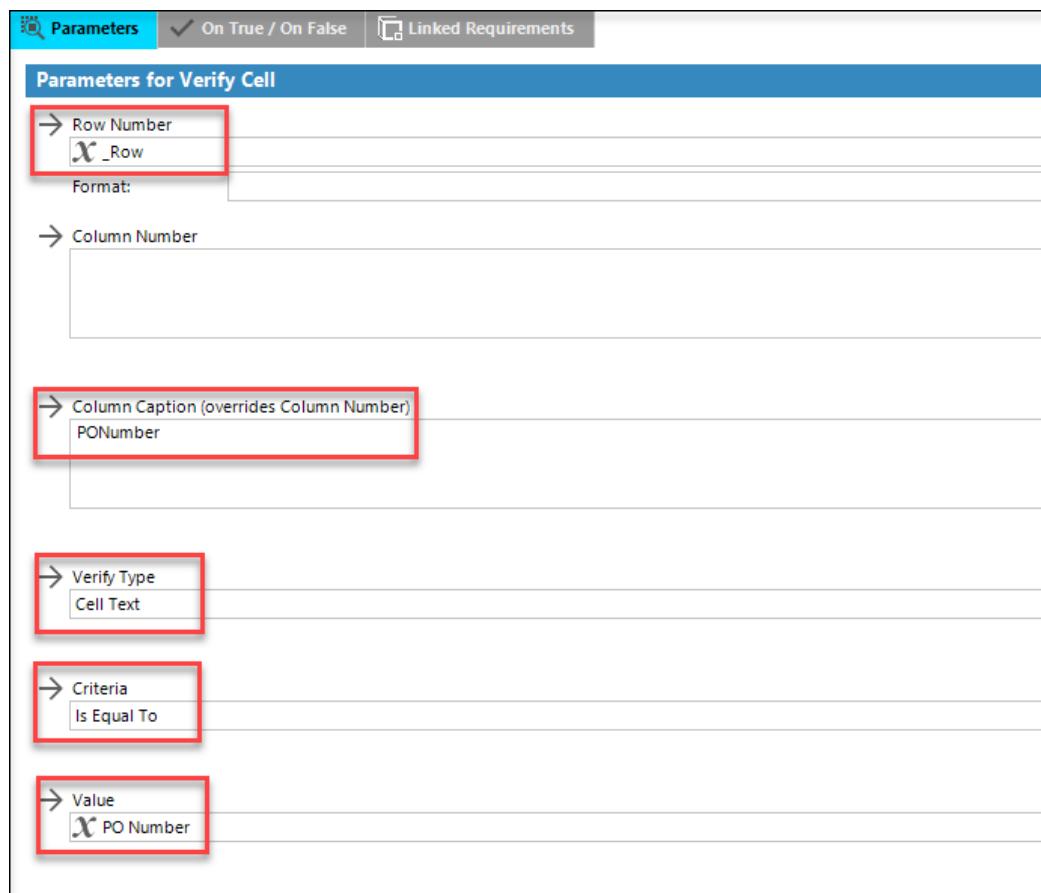
24. In the **Criteria** drop-down, select **Is Equal To**.

25. In the **Value** field, click the **Select Variable**  icon.

The Select Variable dialog box appears.

26. In the Variables pane, select the **PO Number** variable.

Your Step 2 parameters should match the following:



Next, we will create a step to store the Doc Number generated into a variable so it can be used in future.

27. Right-click on Step 2, and select **Insert Step Below**.

28. Change the **Action** of Step 3 to **Store Cell**.

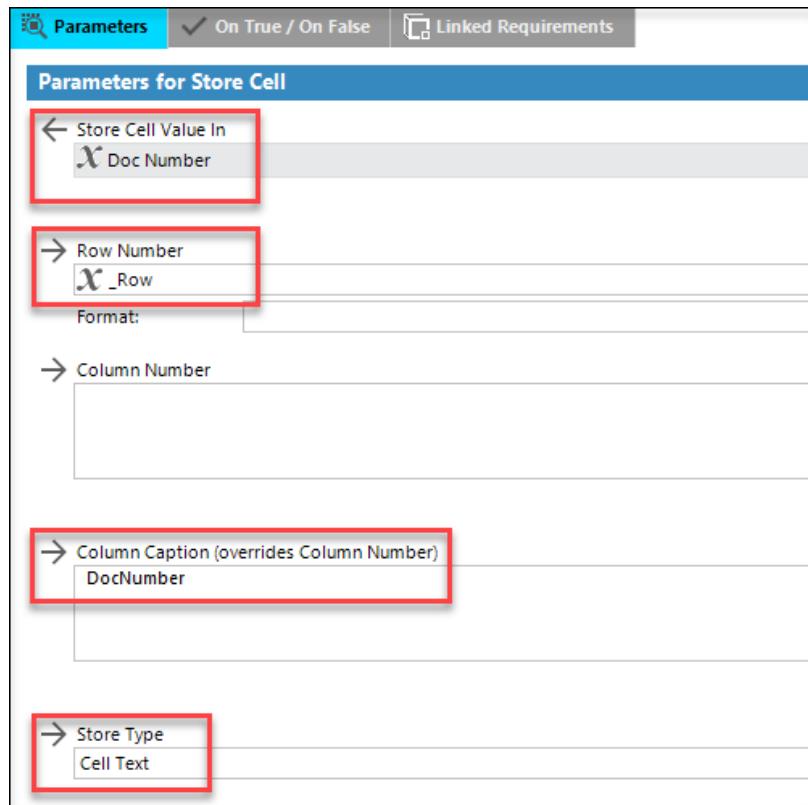
29. Modify the Step 3 parameters as follows.

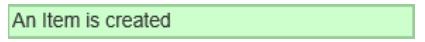
30. In the **Store Cell Value in** field, click the **Select Variable**  icon.

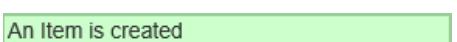
The Select Variable dialog box appears.

31. In the Variables pane, select **Doc Number**.
32. In the **Row Number** field, select the **_Row** variable.
33. In the **Column Caption 1 (overrides Column Number)** field type **DocNumber**.
34. In the **Store Type** field select the **Cell Text** from the dropdown.

Your Step 3 parameters should match the following:



Next, we will insert a step to verify the application's  message.

35. Right click on Step 3, and select **Insert Step Below Using LiveTouch**.
36. Use LiveTouch to select the  message.
37. Click **Save & Close**.

38. Change the Action in Step 4 from Press to **Verify**.

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Find Row (Advanced)	Find a row in the tblList Table column "Name"" with "innerText" that "Is Equal To": TIName	Continue	Continue
2	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Verify Cell	Verify that the "Cell Text" at Row N[_Row], Column ""PONumber" in the tblList Table "Is Equal To": TPO Number	Continue	Continue
3	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Store Cell	Store the "Cell Text" at Row N[_Row], Column ""DocNumber" of the tblList Table into TDoc Number	Continue	Continue
4	Worksoft Web Sample Application 1.0	WebAccountManager	status	Press	Press the status Link.	Continue	Continue

Parameters On True / On False Linked Requirements

Parameters for Press

Verify

39. In the **Value** field of the Parameters tab, type **An Item is created**.

40. Set the **Criteria** field to **Is Equal To**.

41. Save  the process.

42. Close the Process Editor.

EXERCISE 3.13 — Modifying the WSA_Input Layout

In the previous exercise, the generated Document Number will be stored in the Doc Number variable. We are going to add that variable to the WSA_Input Layout so the Document Number will be written to the WSA_Input Recordset after each execution.

Step	Action
1.	In the Navigation Taskbar, click Data .
2.	Select your WSA_CreateandModify folder, and open your WSA_Input Layout for editing.
3.	In the Variables pane, right-click and select Add .

The screenshot shows the 'Edit Layout' dialog box for the 'WSA_Input' layout. The 'Variables*' section contains the following data:

Number	Name	Default Value	Description
1	PO Number	txtPONumber_PC_201703271245159	
2	Name	txtName_PC_20170327124521985	
3	Ship to	txtShipTo_PC_20170327124527812	
4	Bill to	txtBillTo_PC_20170327124531885	

A context menu is open over the grid, with the 'Add' option highlighted. Other options in the menu include 'Customize Columns...', 'Print', 'Export', 'Insert', 'Edit', 'Delete', 'Move to Top', 'Move Up', 'Move Down', and 'Move to Bottom'. The 'OK' and 'Cancel' buttons are at the bottom right of the dialog.

4. Select the **Doc Number variable** and click **OK**.

Details

Name: *
WSA_Input

Description:

Create:

Variables*

Number	Name	Default Value	Description
1	PO Number	txtPONumber_PC_201703271245155	
2	Name	txtName_PC_20170327124521985	
3	Ship to	txtShipTo_PC_20170327124527812	
4	Bill to	txtBillTo_PC_20170327124531885	
5	Doc Number		

OK Cancel

- Click **OK** to save the layout.
- Now, each time the **WSA_CreateandModify** process is executed, each Document Number will be written to the **WSA_Input** recordset. ***Note: You will see this in a later lesson.**

Edit Process - WSA_CreateandModify

File Edit View Action Tools SAP Help

WSA_CreateandModify

Recordset

Name: *
WSA_Input

Description:
Single Row Created During Add to Layout

Records:

PO Number	Name	Ship to	Bill to	Doc Number
1 8403050134	Mary Wilson	1444 North Drive	100 Fifth Street	1491861720572
2 8403050203	John Smith	123 Renner Lane	1455 North Main	1491861741692
*				

Lesson Summary

You've completed the [Developing Automation Using Worksoft Certify lesson](#).

Key points to remember:

- A variable is a symbol or name that represents a value. The most common use of variables is for data-driven testing, where process execution loops through a series of data values.
- Variables can also be used when you need to store or verify system data (such as the System Date or Machine Name) or when specific data for a user is required (such as a user ID or password to log into an external system or application).
- When preparing to use variables, the first thing you should do is go through your existing processes and identify all the places where variables can be used in place of data. Once you have all the variables for the process identified, you then need to determine the type of variables you are using: System, Project, Process, or User.

As a best practice, you should do the following:

- Create a table or spreadsheet showing the processes and all the variables you will use for each process.
- Use variable names that describe what type of data the variable contains and how you will use the variable.
- A layout is a collection of variables that define the data in a recordset.
- You can create a layout at any time after creating a project although a layout is not meaningful until it is connected to a process.
- If you have a process that has steps containing literal values or variables and no associated layout, you can create a layout for the process using the Add to Layout feature.
- Recordsets contain data values used in data-driven testing. A recordset is defined by a layout. Once you have defined at least one layout, you can create recordsets and add data values for the variables defined in that layout.
- You can attach recordsets to processes or process steps and either read data values from the recordset to process steps or write values provided by the application you are testing to the recordset.
- Using recordsets provides a way to loop through your application while a test is being executed. The looping allows each row of data in the recordset to be used in the process.
- When a recordset is used with a process, the process will execute one time for each row of data in the recordset. After completing a test with the first row of data, the process will execute again with the second row of data.

- Recordsets can also be read, overwritten, or appended at the step level within any process using the system Read Recordset or Write Recordset action. If either of these actions is used within a step, only a single recordset row is read or written to.

Lesson 4

Creating and Executing Integrated Processes

Overview

This lesson explores the different methods and options you can use to execute processes and how to combine processes into an integrated process.

Objectives

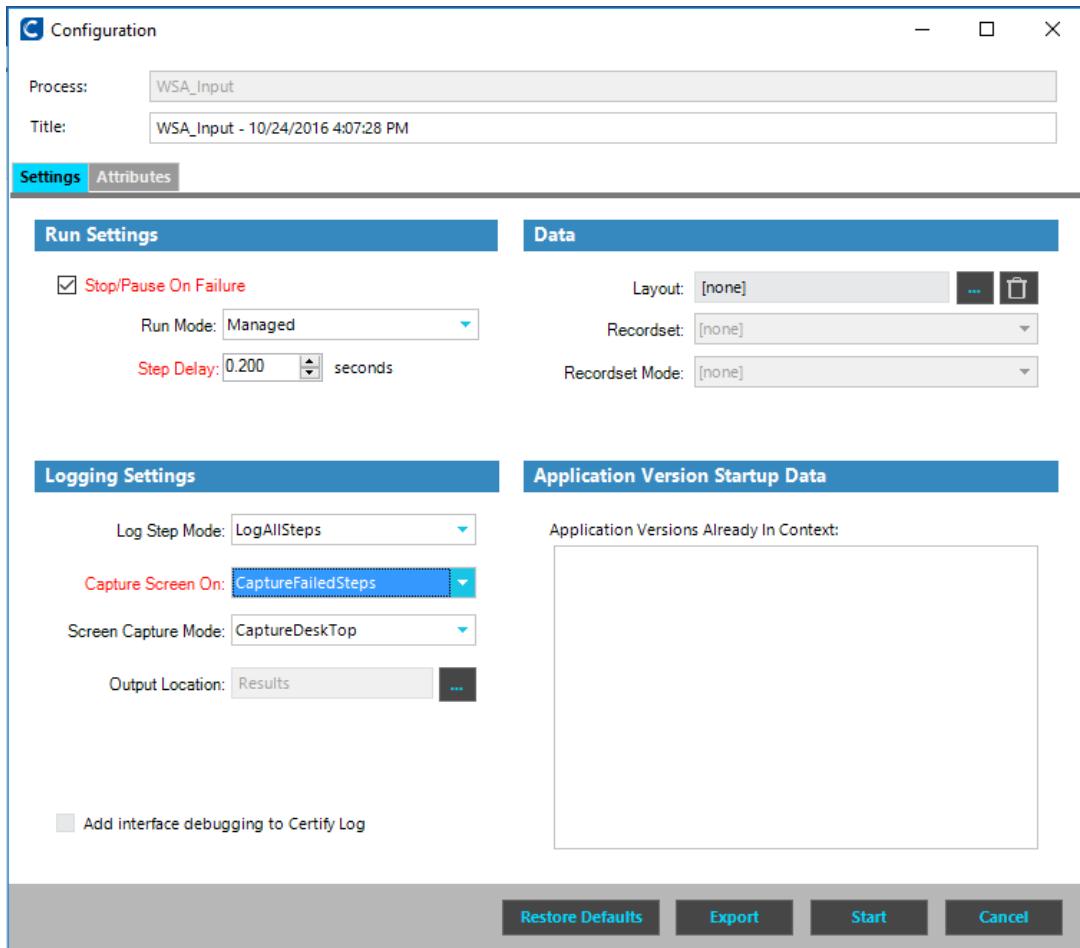
After completing this lesson, you will be able to:

- Execute processes using data.
- Determine where a Layout/Recordset should be placed for optimum execution.
- Explain the various methods and options for executing processes.
- View execution results in the Results Viewer window.
- Import data.
- Export data.

Configuring Process Execution

Prior to the actual execution of a process, you will need to configure the process execution to meet your requirements. Configuration consists of setting the run and log settings and providing where to find input data and startup data. As shown in Figure 1, the Configuration dialog box provides options for choosing the type of execution, how and when execution is performed, and how the results are handled.

Figure 1 — Configuration Dialog Box



Settings Tab

Run Setting Area

Stop/Pause On Failure — By default, execution does not stop on failure. If you want your execution to stop when a failure is encountered in your process, select this checkbox. Failure includes failed and aborted steps.

Tip: Select this checkbox when you are running your process. This allows you to debug the errors and prevents your process from continuing to execute after a failure.

Run Mode — Select how you want to manage the execution or accept the default of managed run mode. Modes include managed, manual, and unmanaged.

Managed mode allows user intervention to step through processes, set breakpoints, and capture screens. Managed mode assists you in debugging your processes.

Manual mode requires complete user intervention. Manual mode is required when an application is not being run (or not developed), and you have no automation or the automation is not working. If your mode is Manual, the Execution dialog box will have an additional tab called Manual Execution.

Unmanaged mode prevents you from stepping through execution, setting breakpoints, and capturing screens.

Step Delay — Select the number of seconds you want to delay execution between each step or accept the default of 0 seconds. Step delay helps resolve timing issues when testing your application.

Logging Setting Area

Log Step Mode — Select how you want the step to be logged or accept the default of LogAllSteps. The modes include: LogAllSteps, LogAbortedStepsOnly, LogFailedSteps, LogProcessOnly, LogStatusOnly, and LogDisabled. When you determine your process is complete, you can log aborted or failed steps only.

Capture Screen On — Select if you want to capture screens and when you want to capture the screens during process execution or accept the default of CaptureNone. Modes include: CaptureNone, CaptureAbortedSteps, CaptureAllSteps, CaptureFailedSteps, CaptureAbortedAndFailedSteps, or WindowChange. The WindowChange option is helpful when you want to see the window or object that is failing in your application. It is also good for a final result, giving you a screenshot after every window change. All screen captures are in *.jpg format.

Tip: We suggest selecting CaptureAbortedAndFailedSteps so that any steps with errors will include a screen capture for debugging purposes.

Screen Capture Mode — Select how much of the area of the application screen you want to capture or accept the default of CaptureDesktop. Modes include CaptureDeskTop and CaptureActiveWindow.

Output Location — Click the Output Location browse  button to change the location of where you want the results stored for this process execution or accept the default displayed. The Select Results Folder dialog box allows you to select any project for which you have permissions. Make your folder selection and click OK in the Select Results Folder dialog box. The default is the top-level Results folder for the opened project.

If you want to add interface debugging to your interface log, select the **Add interface debugging to Certify log** checkbox.

Data Area

Layout (optional) — Click the Layout browse  button to browse for a layout. The default is the current layout associated with the process. If blank, no layout is associated with this process. If you choose a layout and do not choose a recordset, the layout is ignored during execution. To delete a selected layout, click the  button.

Recordset (optional) — If a layout is associated with this process and a recordset exists for that layout, this field is active. Select the down arrow  to choose the recordset you want to use.

Recordset Mode — If a recordset is associated with the layout, this field is active. Select the down arrow to choose the mode for the selected recordset. Options include Read Only, Append, Clear and Append, and Read and Update. Table 1 below explains when and how each mode is executed.

Table 1 — Identified Recordset Modes for Processes

Mode	When Executed	How Executed
Read Only	Reads recordsets at the beginning of execution.	Loops process once for each row until End of File.
Append	Writes recordsets at the end of execution.	Appends to existing recordset and loops process until Abort or Exit.
Clear and Append	Writes recordsets at the end of execution.	Creates new recordset for each execution session and loops process until Exit.
Read and Update	Reads recordsets at the beginning of execution.	Updates the recordset at the end of the process.

Application Version Startup Data Area

Application Version Already in Context — In context implies that you have already loaded your application and positioned the application where the selected process begins. Insert checkmarks for any or all the application versions to be checked for context. By default, there is no checkmark, meaning that the process will run. If the box is checked then the process will not run.

Checkboxes will appear under 'Application Version Already In Context' for the application versions which are assigned Start process in the current project. The application version will have the name of the Start process if one has been created and selected in the project.

If you do not check any application versions and have a defined start process, the start process loads the application, performs logging into the system, and navigates to the main window or starting state before starting execution of the process.

Attributes Tab

Results Attributes — Shows all attributes created for Web links. Set the fields with values before executing so the values are stored with the results.

Configuration Dialog Box Buttons

Restore Defaults — Resets the fields back to the system defaults.

Export — Displays the Save As dialog box to save the current execution configuration parameters as a program command line in a batch file.

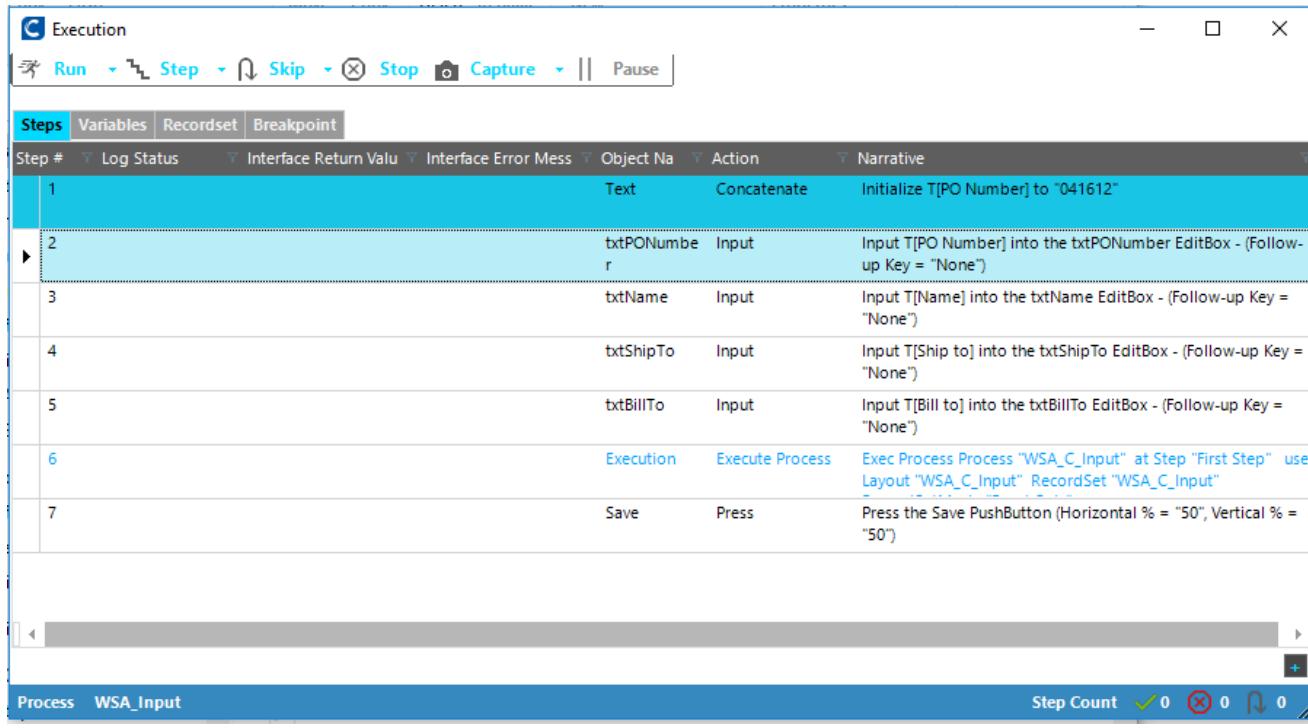
Start — Displays the Execution dialog box so you can run your process.

Cancel — Ignores all information and returns you to the previous screen. No changes are saved.

Execution

Once you click Start in the Configuration dialog box, the Execution dialog box appears, as shown in Figure 2 below. After execution, the Result Viewer dialog box will appear if the process was executed using the Run button. Once configuration is complete, you can set additional execution functionality in the Execution dialog box, such as skipping process steps, capturing application windows, and setting breakpoints.

Figure 2 — Execution Dialog Box



Note: You can restart after a failed step. If you selected the **Stop on Failure** option in a Managed execution mode, you can restart the execution on a different step. Right-click on the step, and select **Set Execution Pointer** to restart execution.

Execution Buttons

The following Execution buttons offer you different options when running your processes.

Run — Executes the process to completion without using intervention.

Steps — Executes the current step of a process. The Step drop-down menu offers three options:

- **Step** — Executes each step of the process. If the step is an Execute Process step, then the execution moves to the first step of the called process.

- **Step Over** — Executes the current step. If the current step is an Execute Process step or is a result of True/False logic, then all sub-processes are executed, and the execution moves to the next step of the current process.
- **Step Out** — Executes to the end of the current process and stops at the next step of the parent process or the first breakpoint set in the current process.

Skip — Skips a step during execution of the process. The Skip drop-down menu offers two options:

- **Skip** — Allows you to skip the selected step and continue execution at the next step. If the current step is an Execute Process, then the process will not be executed.
- **Skip Out** — Ignores the remaining steps of the current process and execution starts again at the next step of the parent process or the first breakpoint set in the current process.

Stops — Stops the execution at the next step, and the Results Viewer appears. Test will be marked **aborted** by the user.

Capture — Adds a step to capture a screen image of the active window or desktop during execution. This option is helpful when you want to see the window or object that is failing in your application. The Capture drop-down menu offers two options:

- **Capture Active Window** — Captures the application screen.
- **Capture Desktop** — Captures the entire desktop.

Pause — Pauses the managed process execution.

Execution Dialog Box Tabs

The following Execution tabs offer you different views when debugging your process execution.

Steps — Displays the steps of the current process to be executed. You can expand the Step view by clicking the Expand/Collapse button on the right of the dialog box. The Expand view allows you to view the steps while using the other tabs.

Variables — Displays the current values of the variables in the current process (Process tab), in the current step (Step tab), or added variables (Watch tab). Watched variables are persistent throughout execution. The Process tab shows local variables.

Recordset — Displays the recordset data that is used in the execution of the current process.

Breakpoint — Allows you to view, set, remove, or clear all breakpoints in any called process during execution.

Creating an Integrated Process

In the previous lessons, you created individual processes. These processes will be combined into an Integrated Process so they will execute together.

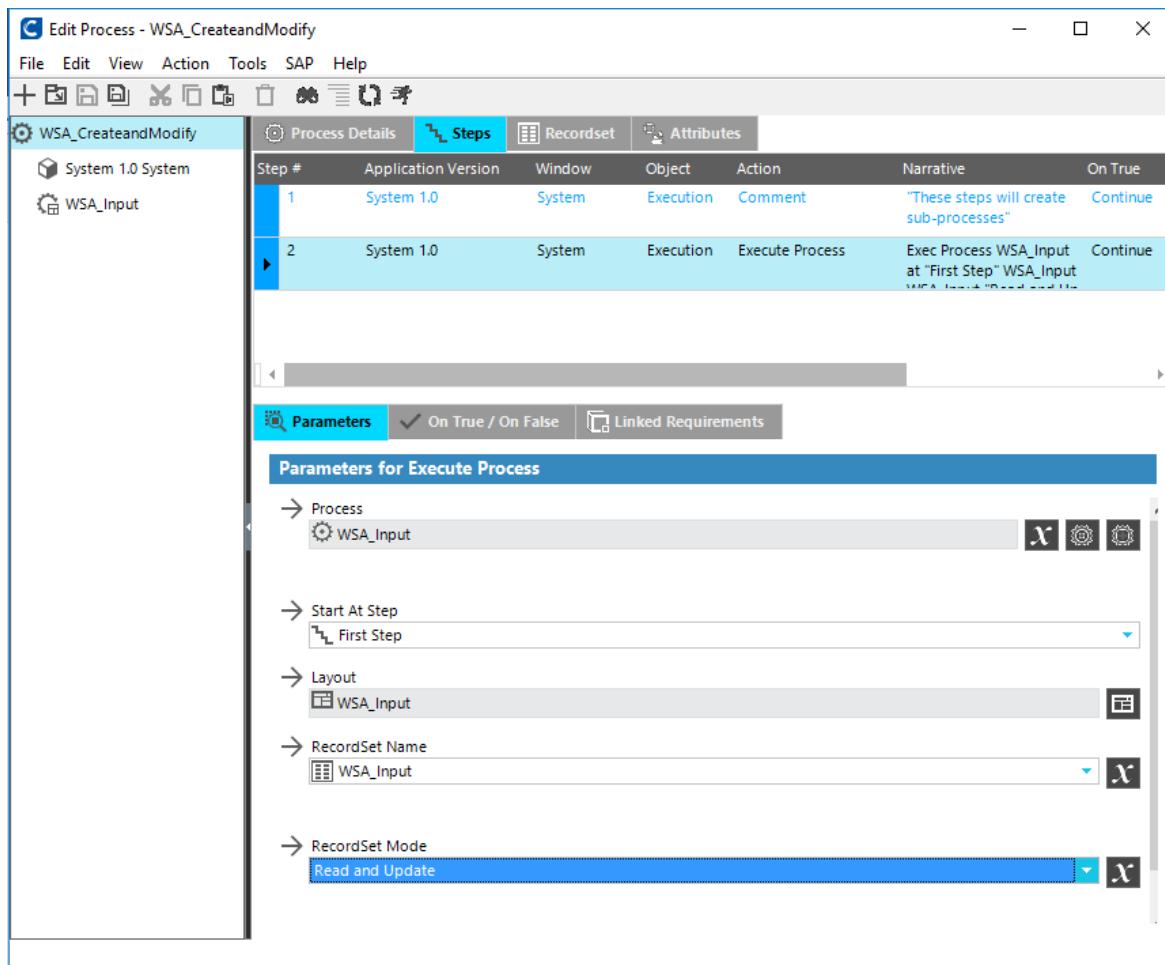
Critical Business Process	Sub-Processes
WSA_CreateandModify	WSA_Input
	WSA_SelectandVerify

EXERCISE 4.1 — Creating and Executing the WSA_CreateandModify Process

In this exercise, you will create an integrated process.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, click your WSA_CreateandModify folder.
3.	In the Summary Pane, right-click and select New Process .
4.	In the Name field, type WSA_CreateandModify . Enter the following description: <i>This is the end-to-end, integrated process.</i>
5.	Click the Steps tab.
6.	Right-click in the Steps tab, and select New.
	<i>This should create a Comment step. We will use this step to describe the following steps.</i>
7.	In the Parameters, for Comment field type <i>These steps will execute sub-processes.</i>
8.	Right-click on Step #1 and select Insert Step Below .
9.	Verify the Application Version is System 1.0 .
10.	Verify the Window is System .
11.	Verify the Object is Execution .
12.	Click the Action drop-down arrow, and select Execute Process .

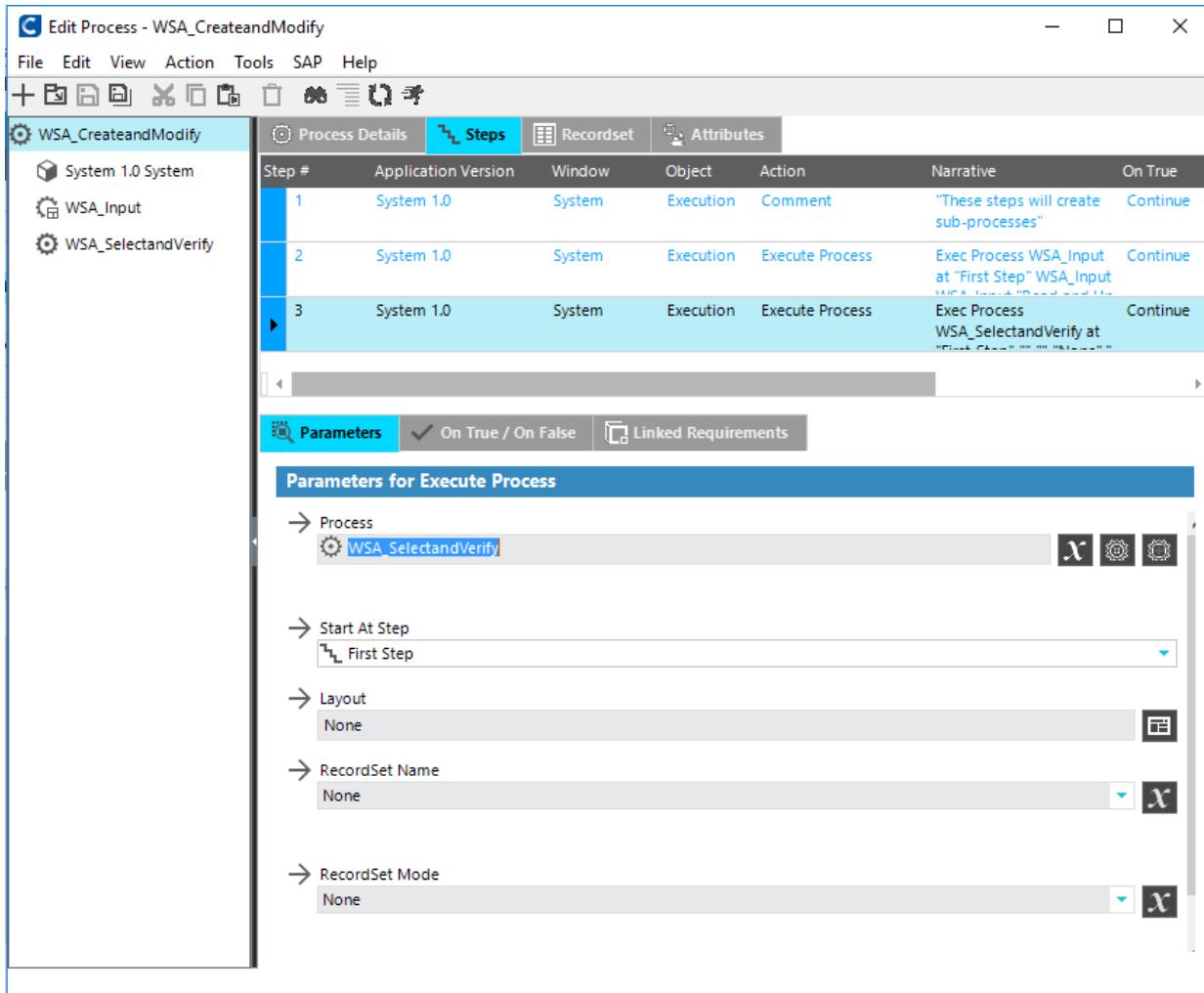
13. In the Parameters tab and in the Process field, click the **Select Existing Process**  icon.
14. Navigate to your **Sandbox folder**, then to your **WSA_CreateandModify** folder.
15. Select the **WSA_Input** process and click **OK**.



Notice that the Layout, RecordSet Name, and RecordSet Mode parameters have been populated automatically.

16. Click **Save** .
17. Right-click on **Step #2** and select **Insert Step Below**.
18. In the Parameters tab and in the Process field, click the **Select Existing Process**  icon.
19. Navigate to your **Sandbox folder**, then to your **WSA_CreateandModify** folder.

20. Select the **WSA_SelectandVerify** process and click **OK**.



21. Click the **Save** button. Your process should contain 3 steps.

22. In the toolbar, press the “Running Man” button.

The Configuration dialog box appears.

23. Click **Start**.

The Execution dialog box appears.

24. Click **Run**.

The process executes, and the Result Viewer appears upon completion. Notice, the Layout, Recordset, and Recordset Mode fields are populated in the Details tab of the Result Viewer.

- In the Result Viewer click the signs to expand the tree on the left.

Notice that the **WSA_Input** process executed twice – once for each row of data in the **WSA_Input** recordset. The other process, **WSA_SelectandVerify**, executed once using the last values set by **WSA_Input**.

The screenshot shows the Result Viewer interface. On the left, a tree view displays processes: WSA_CreateandModify (expanded), WSA_CreateandModify (under it), WSA_Input (under the first WSA_CreateandModify), WSA_Input (under the second WSA_CreateandModify), and WSA_SelectandVerify (under the second WSA_CreateandModify). The WSA_SelectandVerify node is highlighted with a blue selection bar. To the right is a table titled 'Steps' showing five test steps. Below the table is a 'Details' tab panel containing execution parameters and results.

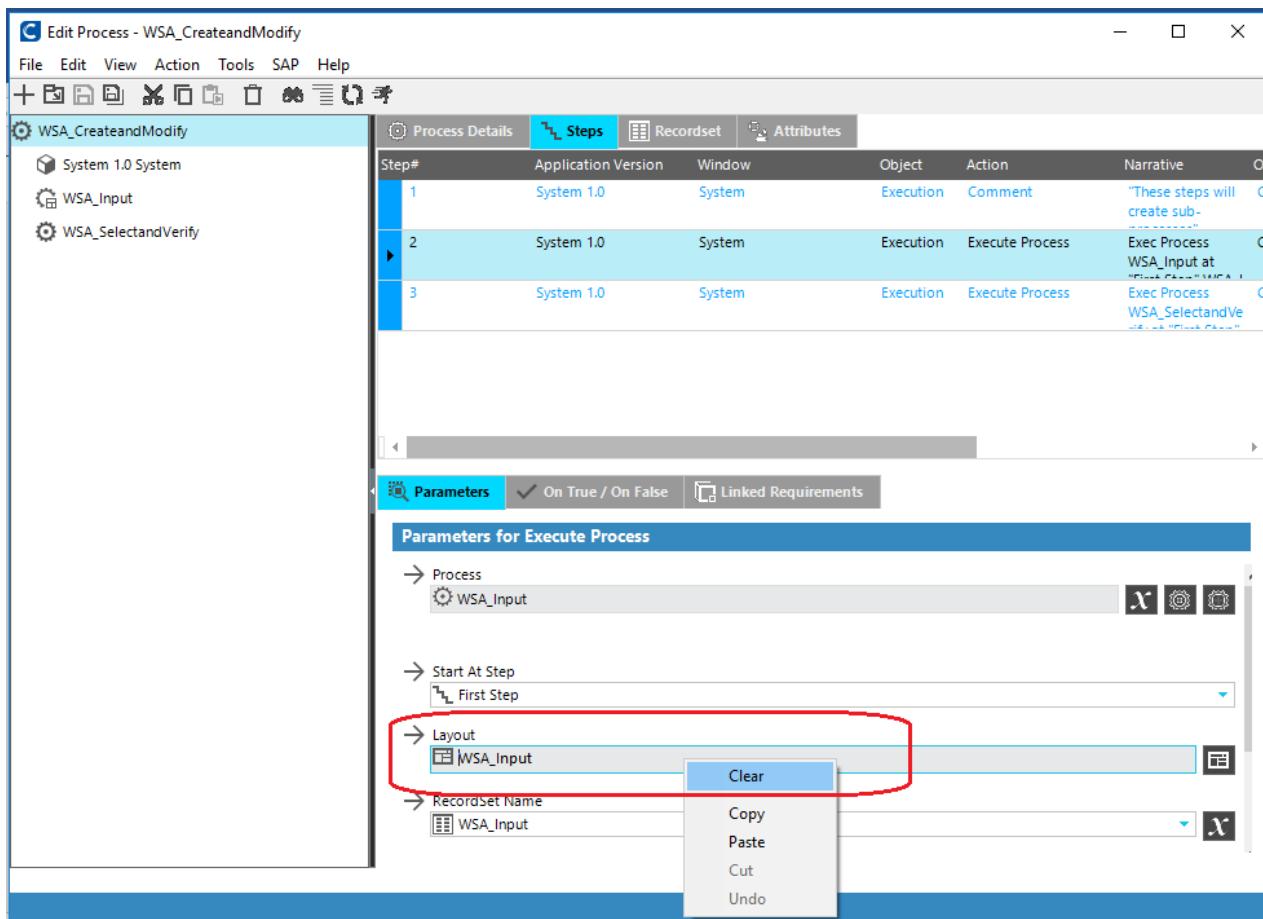
Test Step ID	Application Version Name	Window	Object Name	Action Name	Narrative
65422	New_CertifyWebSampleApp -	WebAccountManage	Search Result	Find Row	Find a row in the Search Result Table_1 Table column "Name"
65423	New_CertifyWebSampleApp -	WebAccountManage	Search Result	Verify Cell	Verify that the value in the Search Result Table_1 Table column "Name" is equal to "John"
65424	New_CertifyWebSampleApp -	WebAccountManage	Search Result	Store Cell	Store the value from the previous step into the variable "Name"
65425	New_CertifyWebSampleApp -	WebAccountManage	status	Verify	Verify that the value in the WebAccountManage status is equal to "Success"

Details Tab Content:

- Object Name: Search Result Table_1, Result: True
- Action Name: Find Row (Advanced), Result Action: None
- Narrative: Find a row in the Search Result Table_1 Table column "Name", Log Status: passed
- Start Time: 4/3/2017 2:14:09 PM, Execution Status: ContinueExecution

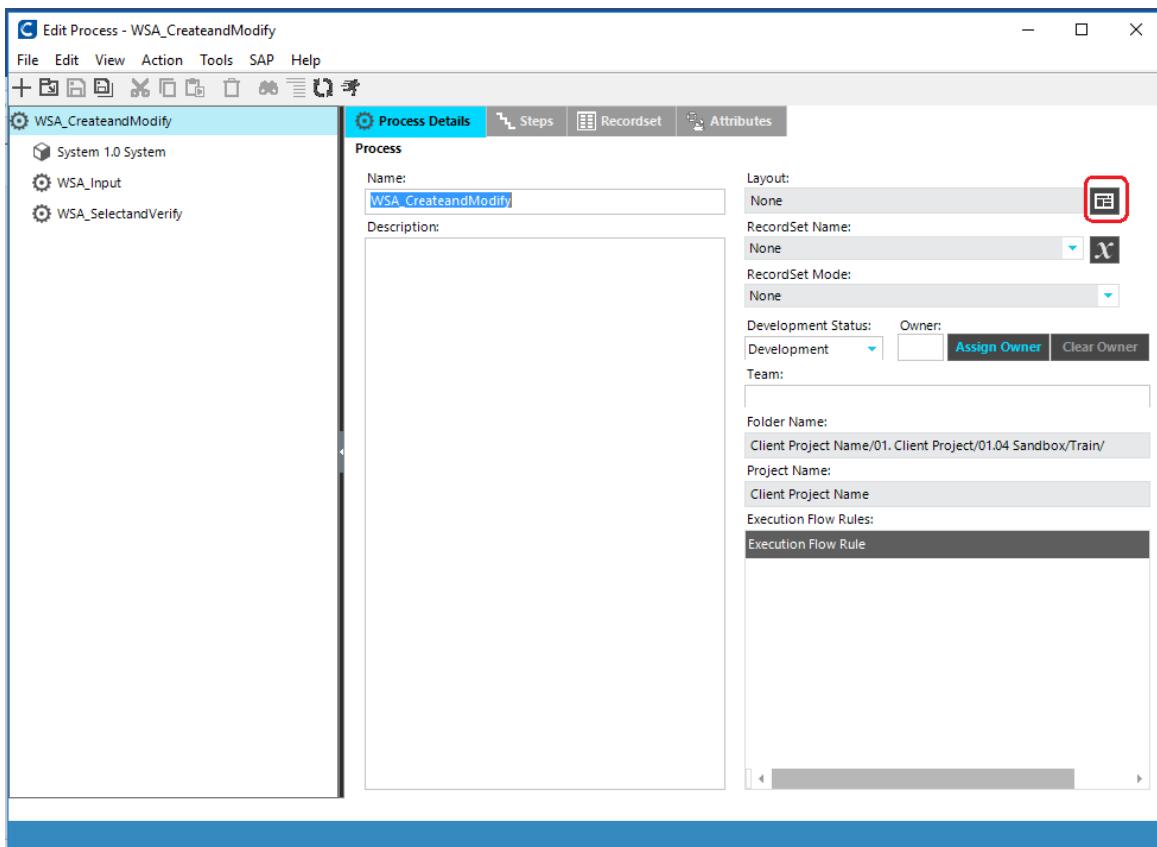
To execute all the processes for each record in the **WSA_Input** recordset, we will move the location of that recordset to the top-level, integrated process – **WSA_CreateandModify**.

26. Close the Result Viewer window, and return to the **WSA_CreateandModify** process editor.
27. Single click on **Step 2** to make it active.
28. In the Parameters area, right-click on the **Layout** field and select **Clear**.



29. Click on the **WSA_CreateandModify** process in the Navigation Tree, and then click on the **Process Details** tab.

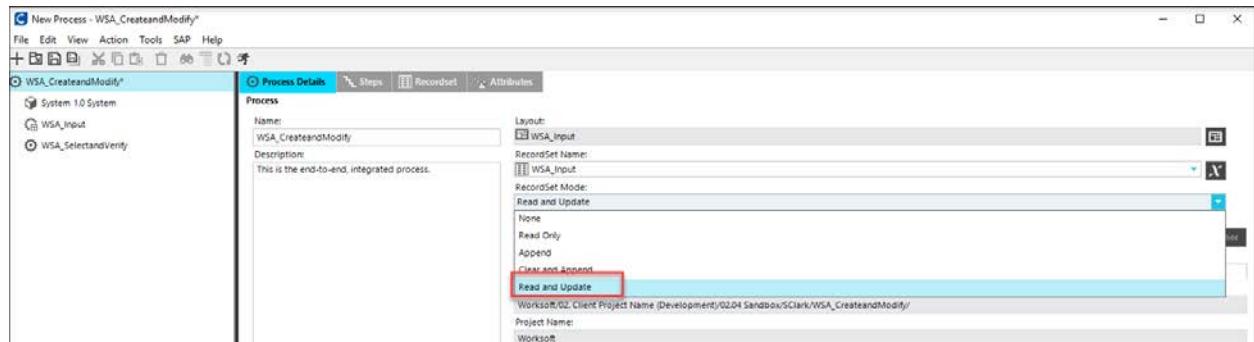
30. In the Layout area, click the Select Layout  button.



31. Navigate to your **Sandbox folder**, and select your **WSA_Input** layout.

32. Click **OK**.

33. Select **Read and Update** from the Recordset Mode drop-down list.



34. Run your **WSA_CreateandModify** process. In the **Result Viewer**, expand the results in the Navigation Tree. You should see each process executed twice.

The screenshot shows the 'Result Viewer' interface. The left pane displays a navigation tree with the following structure:

- Process Results
- WSA_CreateandModify - 4/3/2017 2:3
 - WSA_CreateandModify
 - WSA_Input
 - WSA_SelectandVerify
 - WSA_CreateandModify
 - WSA_Input
 - WSA_SelectandVerify

The right pane is titled 'Steps' and contains a table with the following data:

Test Step ID	App
65426	Syst
65427	Syst
65429	Syst

Note: If your processes execute without data, check that the layout and recordset are attached at the top-level, integrated process:

- Check the Process Details of the WSA_CreateandModify
- Check the Step 2 parameters of WSA_CreateandModify process. There should not be a layout/recordset showing in the parameters.

Examine the Recordset usage.

35. Select the top **WSA_CreateandModify** execution in the Navigation Tree.
36. Select one of the **WSA_CreateandModify** executions in the Processes list.
37. Click the **Recordset Variables Data** tab.

The screenshot shows the Result Viewer application window. On the left, the 'Process Results' pane displays a tree view of process executions. A red arrow points to the first item, 'WSA_CreateandModify - 10/24/2016 7:05:53 PM'. The main area, titled 'Processes', lists two entries: 'WSA_CreateandModify' and 'WSA_CreateandModify'. A red arrow points to the first entry. Below this is a table titled 'Recordset Variables Data'. A red arrow points to the tab header. The table has columns: Variable ID, Variable Name, Variable Value, Variable Type, and LogRecordSet. It contains four rows of data:

Variable ID	Variable Name	Variable Value	Variable Type	LogRecordSet
1064	PO Number	070602	Text	WSA_Input
697	Name	Mary Wilson	Text	WSA_Input
4002	Ship to	100 Fifth Street	Text	WSA_Input
4003	Bill to	1444 North Fifth	Text	WSA_Input

Verify the PO verification step to see the PO number was compared.

38. Select one of the **WSA_SelectandVerify** executions in the Navigation Tree.
39. Select the **Step 2 Verify Cell** step.
40. Click the **Parameters** tab.

The screenshot shows the Result Viewer application window. On the left, the Process Results pane displays a tree view of test executions, with the 'WSA_SelectandVerify' node highlighted and selected. The main area shows the 'Steps' table and the 'Parameters' tab of the selected step's details.

Steps Table Data:

Test Step ID	Application Version Name	Window	Object Name	Action Name	Narrative	Log Status
65422	New_CertifyWebSampleApp -	WebAccountManage	Search Result	Find Row	Find a row in the Search Result Table_1	passed
65423	New_CertifyWebSampleApp -	WebAccountManage	Search Result	Verify Cell	Verify that the "Cell Text" at Row "6", Column	passed
65424	New_CertifyWebSampleApp -	WebAccountManage	Search Result	Store Cell	Store the "Cell Text" at Row "6", Column	passed
65425	New_CertifyWebSampleApp -	WebAccountManage	status	Verify	Verify that the status Link "Is Equal To"	passed

Parameters Tab Data:

Parameter Name	Parameter Value	Variable Name
Row Number	6	_Row
Column Number		
Column Caption (overrides Column)	PONumber	
Verify Type	Cell Text	
Criteria	Is Equal To	
Value	8403023034	PO Number
Returned Value	8403023034	

Select the **Parameters** tab for one of the **Verify** steps to see the Variable in use.

41. Select one of the **WSA_SelectandVerify** executions in the Navigation Tree.
42. Select Step 3 which has the action **Store Cell**.
43. Click the **Parameters** tab.
44. You should be able to see the Document Number stored in the Variable **Doc Number**.

The screenshot shows the Result Viewer application interface. On the left, the navigation tree displays a process named "WSA_CreateandModify - 4/3/2017 2:3" with several execution nodes, one of which is "WSA_SelectandVerify" highlighted with a red box. The main area is a table titled "Steps" with columns: Test Step ID, Application Version Name, Window, Object Name, Action Name, Narrative, and Log Status. The row for "WSA_SelectandVerify" (Step ID 65424) is also highlighted with a red box. Below the table, there are tabs for Details, Parameters, Recordset Variables Data, and Test Step Image. The "Parameters" tab is selected and shows a table with columns: Parameter Name, Parameter Value, and Variable Name. The first row, "Store Cell Value In", has its value "1491247867224" and variable name "Doc Number" highlighted with a red box. Other parameters listed include Row Number (6), Column Number, Column Caption (overrides Column), Store Type (Cell Text), and Returned Value (1491247867224).

Parameter Name	Parameter Value	Variable Name
Store Cell Value In	1491247867224	Doc Number
Row Number	6	_Row
Column Number		
Column Caption (overrides Column)	DocNumber	
Store Type	Cell Text	
Returned Value	1491247867224	

Exporting Data

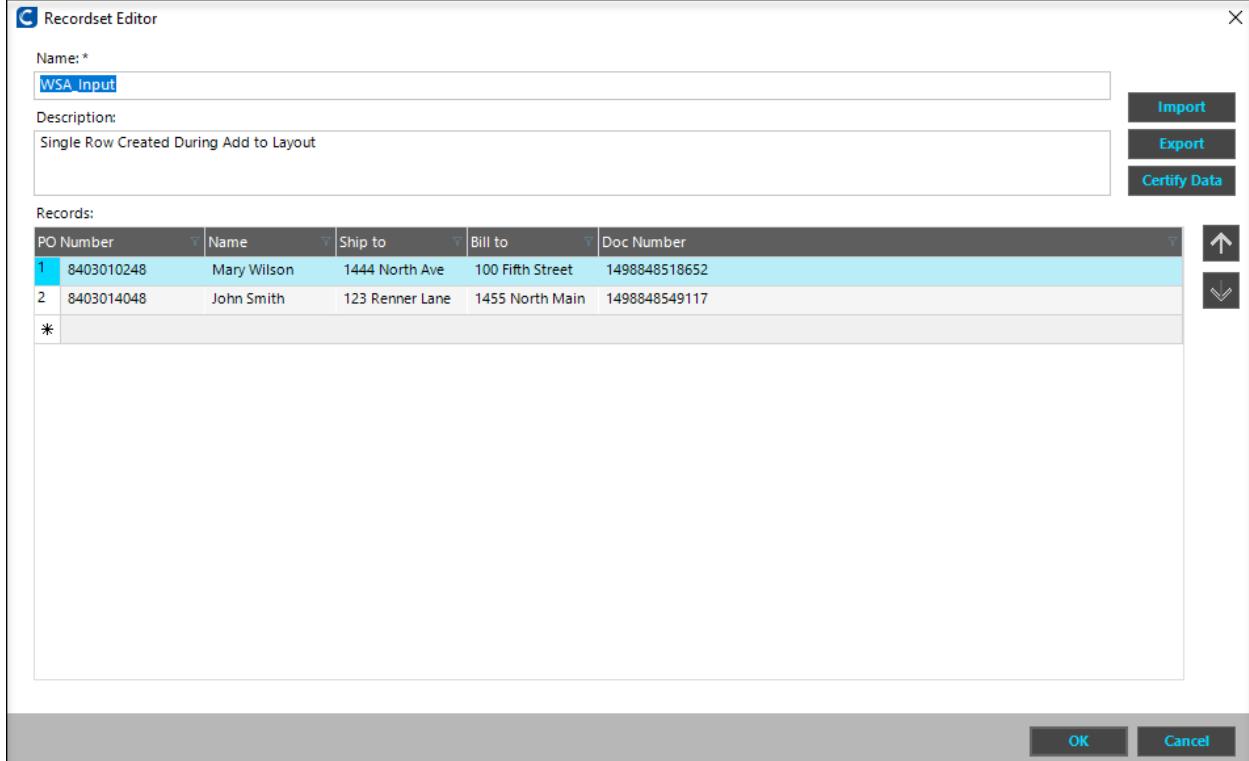
Data can be imported to and exported from recordsets while working in the Recordset Editor. Exporting data is saved to a text file using a delimiter specified by the user. Once exported, the data can be managed in another data management tool and when ready, the modified data can be imported back into the original or into a different recordset.

EXERCISE 4.2 — Exporting Data

In this exercise, you will export data to a file from a recordset you created in a previous exercise.

Step	Action
1.	In the Navigation Taskbar, click Data .
2.	Navigate to your Sandbox folder and then your WSA_CreateandModify folder.
3.	In the Summary Pane, select your WSA_Input layout.
4.	In the Details Pane, click the Recordsets tab.
5.	Right-click the WSA_Input recordset, and select Edit .

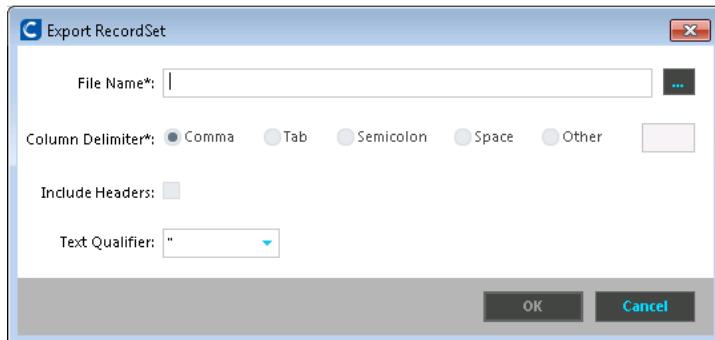
The Recordset Editor dialog box appears.



The screenshot shows the 'Recordset Editor' dialog box. At the top, there are fields for 'Name:' (containing 'WSA Input') and 'Description:' (containing 'Single Row Created During Add to Layout'). To the right of these fields are three buttons: 'Import' (disabled), 'Export' (disabled), and 'Certify Data'. Below these fields is a section titled 'Records:' containing a table with two rows of data. The table has columns: PO Number, Name, Ship to, Bill to, and Doc Number. The first row (highlighted in light blue) contains values: 8403010248, Mary Wilson, 1444 North Ave, 100 Fifth Street, 1498848518652. The second row contains values: 8403014048, John Smith, 123 Renner Lane, 1455 North Main, 1498848549117. At the bottom of the dialog box are 'OK' and 'Cancel' buttons.

6. Click **Export**.

The Export Recordset dialog box appears.



7. Click the **File Name** browse button.

The Select Export File dialog box appears.

8. In the **File Name** field, enter **WSA_Input.txt**.

9. Browse to the **My Documents** folder.

10. Click **Save**.

The Select Export File dialog box closes, and the file name appears in the File Name field of the Export Recordset dialog box.

11. Select the **Comma** option for the Column Delimiter.

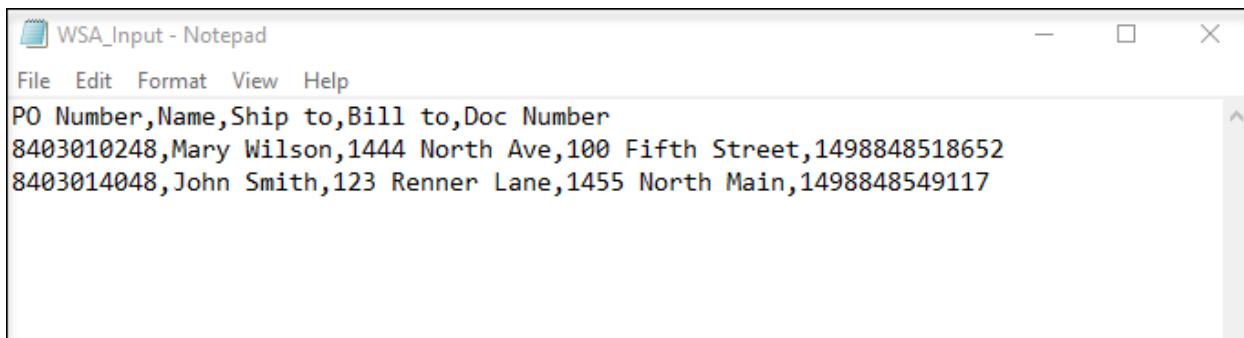
12. Select the **Include Headers** option.

13. In the Text Qualifier field, click the drop-down arrow and select <none>.

14. Click **OK**. *The file is exported as a .txt file*

15. Open Microsoft File Explorer, and navigate to the saved file.

16. Open and view the file named **WSA_Input.txt** in Windows Notepad or another editor.



WSA_Input - Notepad

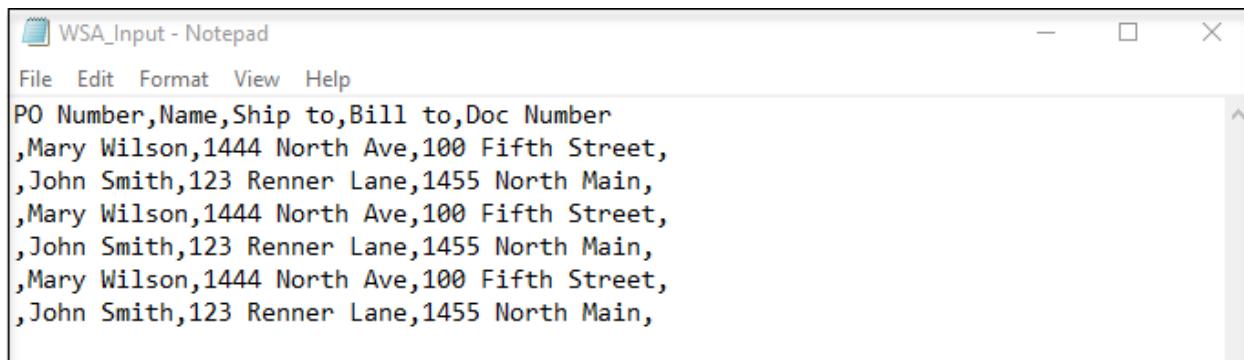
File Edit Format View Help

PO Number,Name,Ship to,Bill to,Doc Number

8403010248,Mary Wilson,1444 North Ave,100 Fifth Street,1498848518652

8403014048,John Smith,123 Renner Lane,1455 North Main,1498848549117

17. Add lines to the file by copying and pasting. Delete the PO Number and Doc Number for each record (**remember each line MUST start and end with a comma**).



WSA_Input - Notepad

File Edit Format View Help

PO Number,Name,Ship to,Bill to,Doc Number

,Mary Wilson,1444 North Ave,100 Fifth Street,

,John Smith,123 Renner Lane,1455 North Main,

,Mary Wilson,1444 North Ave,100 Fifth Street,

,John Smith,123 Renner Lane,1455 North Main,

,Mary Wilson,1444 North Ave,100 Fifth Street,

,John Smith,123 Renner Lane,1455 North Main,

18. Save and close the file. *This is the file we'll use to import.*

19. Close the **Edit Recordset** dialog box.

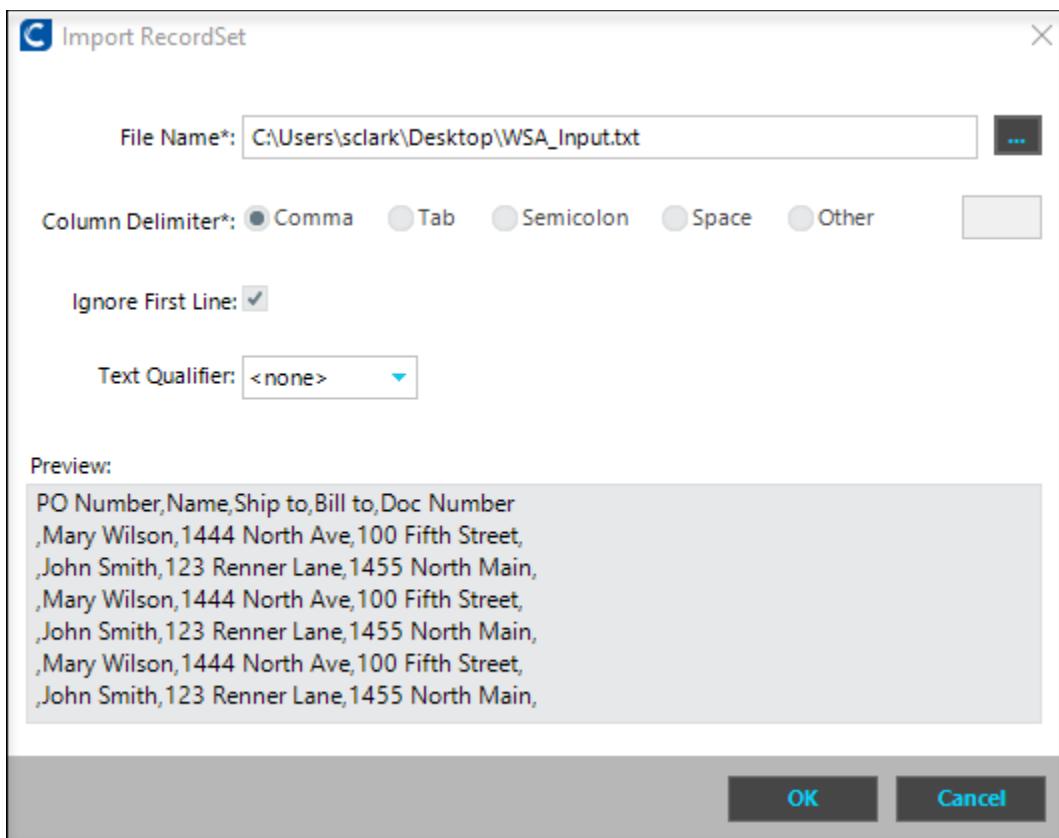
Importing Data

Importing a recordset is like exporting a recordset.

EXERCISE 4.3 — Importing Data

In this exercise, you will import data from the file you created in the previous exercise. It is important to select the same options you specified when you exported the file.

Step	Action
1.	In the Navigation Taskbar, click Data .
2.	Navigate to your Sandbox folder and then your WSA_CreateandModify folder.
3.	In the Summary Pane, select your WSA_Input layout.
4.	In the Details Pane, click the Recordsets tab.
5.	Right-click and select New Recordset .
	<i>The Recordset Editor dialog box appears.</i>
6.	In the Recordset name field, type Import .
7.	Click Import .
	<i>The Import Recordset dialog box appears.</i>
8.	Click the File Name browse  button.
	<i>The Select Import File dialog box appears.</i>
9.	Browse to the file you created earlier.
10.	Click Open .
11.	Click the Comma option for the Column Delimiter.
12.	Check the Ignore First Line option. This will ignore the header line as an input record.
13.	In the Text Qualifier field, click the drop-down arrow and select <none> .



14. Click **OK**.

The file is imported to a recordset.

15. Click **OK** to save, and close the **Edit Recordset** dialog box.

Note: There are Certify steps which allow a process to Import Recordsets and Export Recordsets during execution.

Lesson Summary

You've completed the **Creating and Executing Integrated Processes** lesson.

Key points to remember:

- The Configuration dialog box provides options for choosing the type of execution, how and when execution is performed, and how the results are handled.
- Once configuration is complete, you can set additional execution functionality in the Execution dialog box, such as skipping process steps, capturing application windows, and setting breakpoints.
- After process execution, the Result Viewer shows a log of the execution results.
- The Result Viewer allows you to expand the execution results to show each process and step executed. You can expand each level of the hierarchy and view information about the executed processes and steps in the Summary Pane.
- The placement of a Layout/Recordset determines execution looping.
- You can export data from Certify to an external file and make changes.
- You can import data from an external file into Certify.

Lesson 5

Advanced Executing Processes, Troubleshooting, and Viewing Results

Overview

Although you have executed processes in previous lessons, this lesson explores the different methods and options you can use to achieve successful testing results.

Objectives

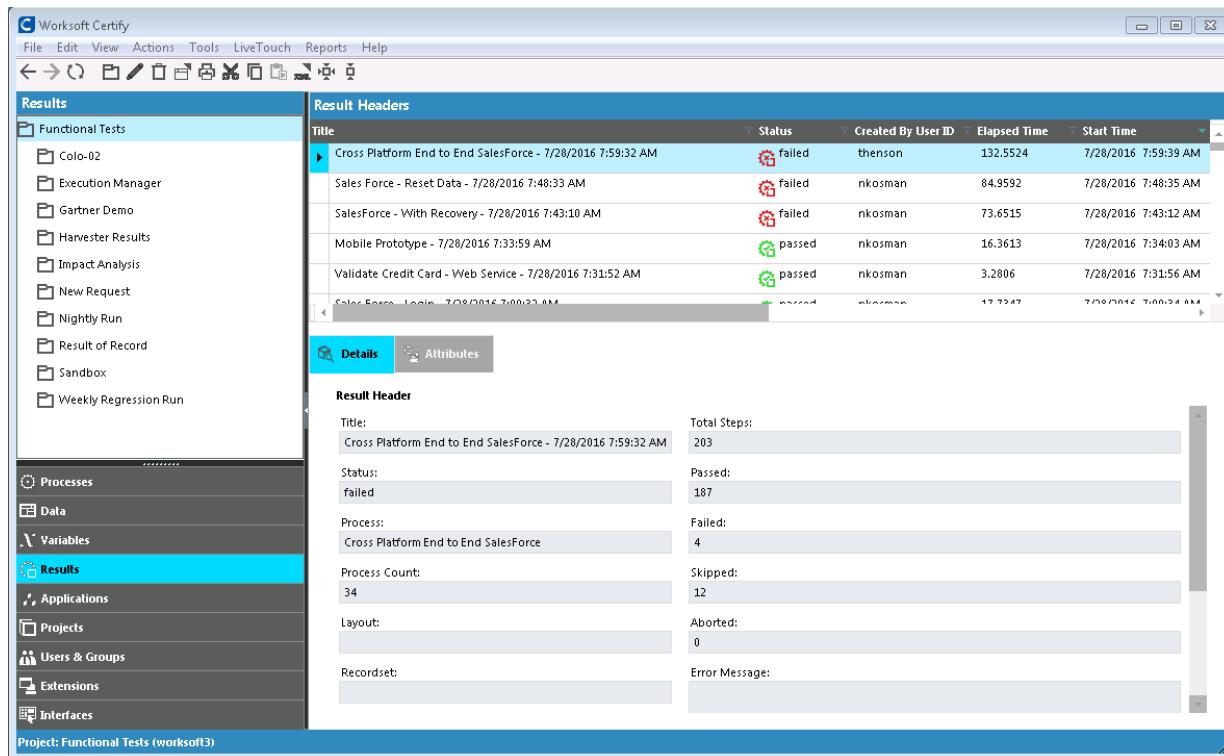
After completing this lesson, you will be able to:

- Create Results folders.
- View execution results in the Results Viewer window.
- Edit Process Steps from the Result Viewer.
- Generate Detail and Summary Reports of the execution.

Managing Execution Results

After process execution, the Result Viewer shows a log of the execution results. By now you have seen the Result Viewer a few times. Once you close the Result Viewer, it is stored and managed in the Results window and is accessible from the Certify Navigation Taskbar.

Figure 1 — Result Window



By right-clicking a result in the Summary Pane, you can perform tasks such as:

View execution results — All your execution results that you saved are stored in the Results window so that you can open them in the Result Viewer or perform routine maintenance.

Generate a results report — A report for a selected execution result can be generated showing either a summary of the result or all the result details. The Report Viewer allows you to navigate the report, as well as print and/or export the report.

Compress the results — Helps reduce the size of your database and increase performance by removing all process step execution results. You can keep only process summary information or both process summary and failed step information.

Promote to a Result of Record — You can identify a specific execution result as a Result of Record, which means that result is the official execution result used to determine the status of processes linked to test requirements, and therefore, the overall satisfaction of the requirement.

Export Result Summary — The summary of a selected result can be saved to an .xml file for external use.

In the Details Pane, you can view the details for a selected result and view the values of any user-defined attributes associated with the results.

Viewing Execution Results

The Result Viewer allows you to expand the execution results to show each process and step executed. The Navigation Tree displays a log header with details about the process executed. Under the log header is a hierarchy of processes executed. You can expand each level of the hierarchy and view information about the executed processes and steps in the Summary Pane.

Processes in the Summary Pane display information about test status, elapsed time, start time, end time, and who created and modified the process. The Details Pane displays additional information about the process and any associated recordset values that were used by the process.

Figure 2 — Result Viewer

The screenshot shows the 'Result Viewer' application window. The top menu bar includes 'File', 'Actions', and 'Reports'. Below the menu is a toolbar with icons for search, refresh, and navigation. A 'Process Results' tree view on the left shows a hierarchy of processes: 'WSA_CreateandModify - 4/6/2017 3:35:56' which contains 'WSA_CreateandModify', 'WSA_Input' (which further branches into 'WSA_Input_C_Materials'), 'WSA_SelectandVerify', and 'WSA_CreateandModify' again. The 'WSA_Input' node is currently selected. The main pane is titled 'Steps' and displays a table of test steps:

Test Step ID	Application Version Name	Window	Object Name	Action Name	Narrative	Log Status
65402	System - 1.0	System	Execution	Name Activity	"WSA_Input"	passed
65411	System - 1.0	System	Text	Concatenate	Initialize T[PO Number] to "8403"	passed
65403	Certify Web Sample App - 1.0	WebAccountManage	txtPONumber	Input	Input "840303439" into the	passed
65404	Certify Web Sample App - 1.0	WebAccountManage	txtName	Input	Input "John Smith" into the txtName	passed
65405	Certify Web Sample App - 1.0	WebAccountManage	txtShipTo	Input	Input "123 Renner Lane" into the	passed
65406	Certify Web Sample App - 1.0	WebAccountManage	txtBillTo	Input	Input "1455 North Main" into the	passed
65417	System - 1.0	System	Execution	Execute Process	Exec Process Process	passed
65409	Certify Web Sample App - 1.0	WebAccountManage	Save	Send Click	"LeftClick" the Save PushButton.	passed

Below the table, a 'Details' tab is selected, showing detailed information for the selected step (Object Name: Execution, Action Name: Name Activity, etc.). Other tabs include 'Parameters', 'Recordset Variables Data', and 'Test Step Image'.

Steps in the Summary Pane display information about test status, elapsed time, start time, end time, as well as the components of the step. The Details Pane provides additional information about the executed step, including step execution details, parameters used in the step, recordset variable data used by the step, and any captured images.

Table 2 lists the statuses shown for the executed processes.

Table 2 — Executed Process Status

Status	Description
	Passed step or process
	Skipped step or process
	Failed step or process
	Aborted step or process

Table 3 lists the log statuses shown for the executed steps.

Table 3 — Executed Step Status

Status	Description
	Passed Step
	Skipped step
	Failed step
	Aborted step

Each executed process or step is given one of these statuses. In some cases, a process inherits a status based on the status of another process or step. For example:

- If a process passes, then all the steps in the process passed.
- If a process fails, then at least one step in the process failed.
- If a process is skipped, then all steps in the process are skipped.
- If a step or process is aborted, then the execution is stopped.
- If a step failed execution in a process, the process also fails unless the process was aborted.
- If a step is aborted, the process shows aborted even though some steps may have passed, skipped, or failed execution.
- If a process has more than one sub-process and the sub-process fails execution, the parent process is shown as a failed process.
- If a process has more than one sub-process and the sub-process is aborted, the parent process is shown as an aborted process even though some sub-processes may have passed, skipped, or failed.

Creating a Results Folder

All Results are kept in the Certify Database until they are deleted. You can create folders in the Results area to organize results. For example:

In a Sandbox folder for your use only. This structure makes it simple to delete results you no longer need.

In a folder, specific to a testing cycle. This structure allows team members to quickly determine the status of testing based on the results in a folder.

In a structure that matches the Process Folder structure.

EXERCISE 5.1 — Creating a Results Folder

In this exercise, you will create a Results Folder to store your execution results.

Step	Action
1.	In the Navigation Taskbar, click Results .
2.	In the Navigation Tree, select the New Project folder, and right-click the Sandbox folder.
3.	Select New Folder .
<i>The New Folder dialog box appears.</i>	



4. In the **Name** field, type <your name>.

5. Click **OK**.

Move the results you have created so far to your results folder.

6. In the Navigation Tree, select the top-level folder to see the results you have created so far.

7. Click on the **Created by User** column to sort by User.

Tip: If the **Created by User** column does not appear, use the **Customize Columns** option described in a previous lesson.

8. Scroll to where your results are listed.
9. Select your results and drag them to the folder you just created.

Results

Process	Title	Created By User	Result Of Record	Status	Start Time	End Time
WSA_EditandDelete	WSA_EditandDelete - 5/12/2017 9:06:23 AM	sclark		passed	5/12/2017 9:06:26 AM	5/12/2
WSA_CreateandModify	WSA_CreateandModify - 5/12/2017 9:11:24	sclark		passed	5/12/2017 9:11:37 AM	5/12/2
WSA_EditandDelete	WSA_EditandDelete - 5/12/2017 9:14:44 AM	sclark		passed	5/12/2017 9:14:46 AM	5/12/2
WSA_Input	WSA_Input - 6/5/2017 2:18:58 PM	sclark		passed	6/5/2017 2:19:02 PM	6/5/20
WSA_Input	WSA_Input - 6/5/2017 2:20:39 PM	sclark		passed	6/5/2017 2:20:41 PM	6/5/20
WSA_Input	WSA_Input - 6/5/2017 3:49:56 PM	sclark		passed	6/5/2017 3:50:06 PM	6/5/20
WSA_Input	WSA_Input - 6/6/2017 1:14:718 AM	sclark		passed	6/6/2017 1:14:724 AM	6/6/20
WSA_Input	WSA_Input - 6/6/2017 11:49:35 AM	sclark		aborted	6/6/2017 11:49:49 AM	6/6/20
WSA_Input	WSA_Input - 6/6/2017 11:56:45 AM	sclark		passed	6/6/2017 11:56:59 AM	6/6/20
WSA_Input	WSA_Input - 6/6/2017 2:17:37 PM	sclark		aborted	6/6/2017 2:29:12 PM	6/6/20
WSA_CreateandModify	WSA_CreateandModify - 6/6/2017 2:38:06 PM	sclark		passed	6/6/2017 2:38:14 PM	6/6/20
WSA_CreateandModify	WSA_CreateandModify - 6/6/2017 3:00:03 PM	sclark		passed	6/6/2017 3:00:06 PM	6/6/20

Result Header

Title:	WSA_CreateandModify - 6/6/2017 3:00:03 PM	Total Steps:	56
Status:	passed	Passed:	56
Process:	WSA_CreateandModify	Failed:	0
Process Count:	12	Skipped:	0
Layout:	WSA_Input	Aborted:	0
Recordset:	WSA_Input	Error Message:	
Recordset Mode:	RandomIndex		

Configuring a Results Folder and Advanced Execution

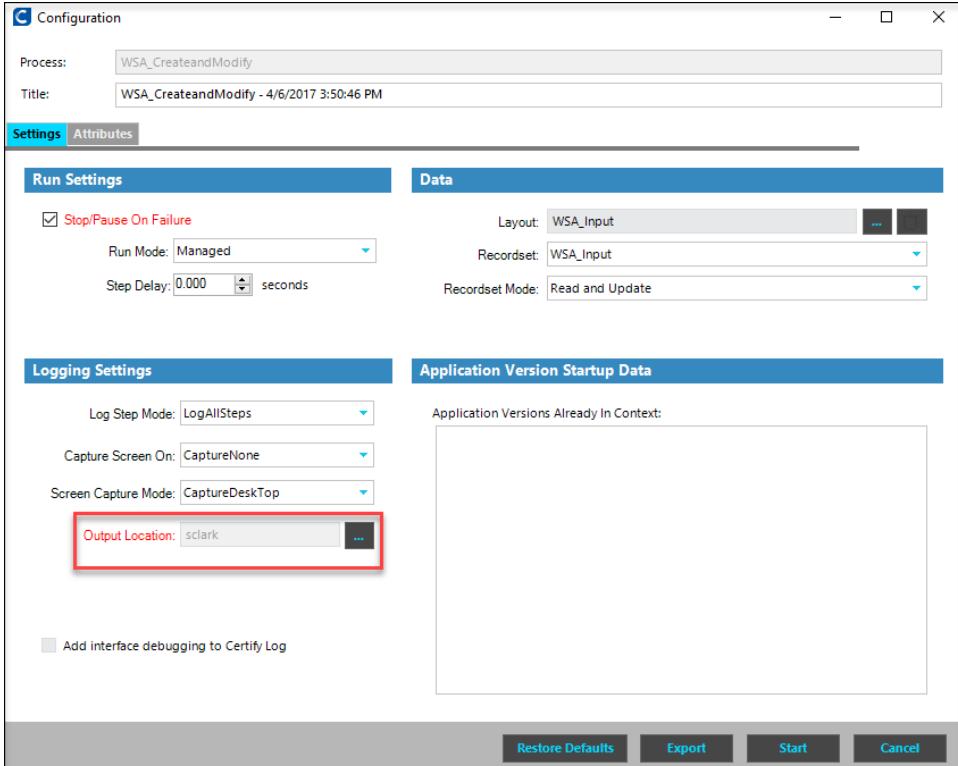
Now that you have created a Results folder, you need to select it during execution. In a previous lesson, the Execution window was described. In this exercise, you will use advanced execution techniques that will help with troubleshooting.

EXERCISE 5.2 — Configuring a Results Folder and Advanced Execution

In this exercise, you will configure a Results folder and use advanced execution techniques.

Step	Action
1.	In the Navigation Taskbar, select Processes .
2.	In the Navigation Tree, select your WSA_CreateandModify folder.
3.	In the Summary Pane, right-click the WSA_CreateandModify process and select Run or press the  “Running Man” button.

The Configuration dialog box appears.



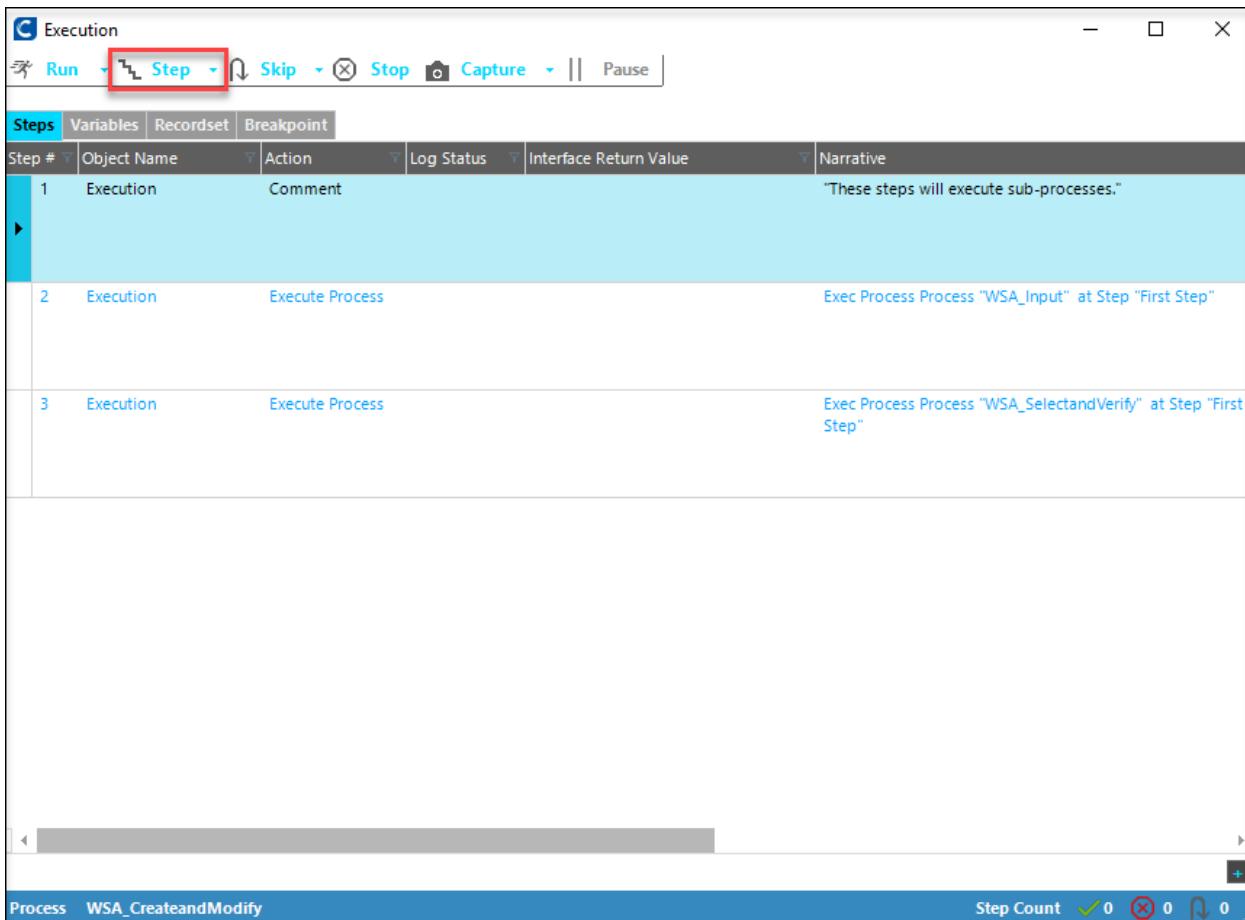
The Configuration dialog box appears.

The dialog box shows the following settings:

- Run Settings:** Stop/Pause On Failure checked, Run Mode: Managed, Step Delay: 0.000 seconds.
- Data:** Layout: WSA_Input, Recordset: WSA_Input, Recordset Mode: Read and Update.
- Logging Settings:** Log Step Mode: LogAllSteps, Capture Screen On: CaptureNone, Screen Capture Mode: CaptureDesktop, Output Location: sclick (highlighted with a red box).
- Application Version Startup Data:** Application Versions Already In Context: (empty list).

At the bottom are buttons for Restore Defaults, Export, Start, and Cancel.

4. Next to the Output Location, click the  button to select your Results folder.
5. Navigate to your Results folder, and click **OK**.
6. Click **Start**.
7. Press the **Step** button.



8. Press the **Step** button again.

Step #2 is a sub-process. The Execution window will now display the steps in that sub-process.

The screenshot shows the 'Execution' window with the 'Steps' tab selected. The window title is 'Execution'. The toolbar includes 'Run', 'Step' (which is highlighted), 'Skip', 'Stop', 'Capture', and 'Pause'. The main area displays a list of steps:

Step #	Object Name	Action	Narrative
1	Execution	Name Activity	"WSA_Input"
2	Text	Concatenate	Initialize T[PO Number] to "8403" D[Date]
3	txtPONumber	Input	Input T[PO Number] into the txtPONumber EditBox.
4	txtName	Input	Input T[Name] into the txtName EditBox.
5	txtShipTo	Input	Input T[Ship to] into the txtShipTo EditBox.
6	txtBillTo	Input	Input T[Bill to] into the txtBillTo EditBox.

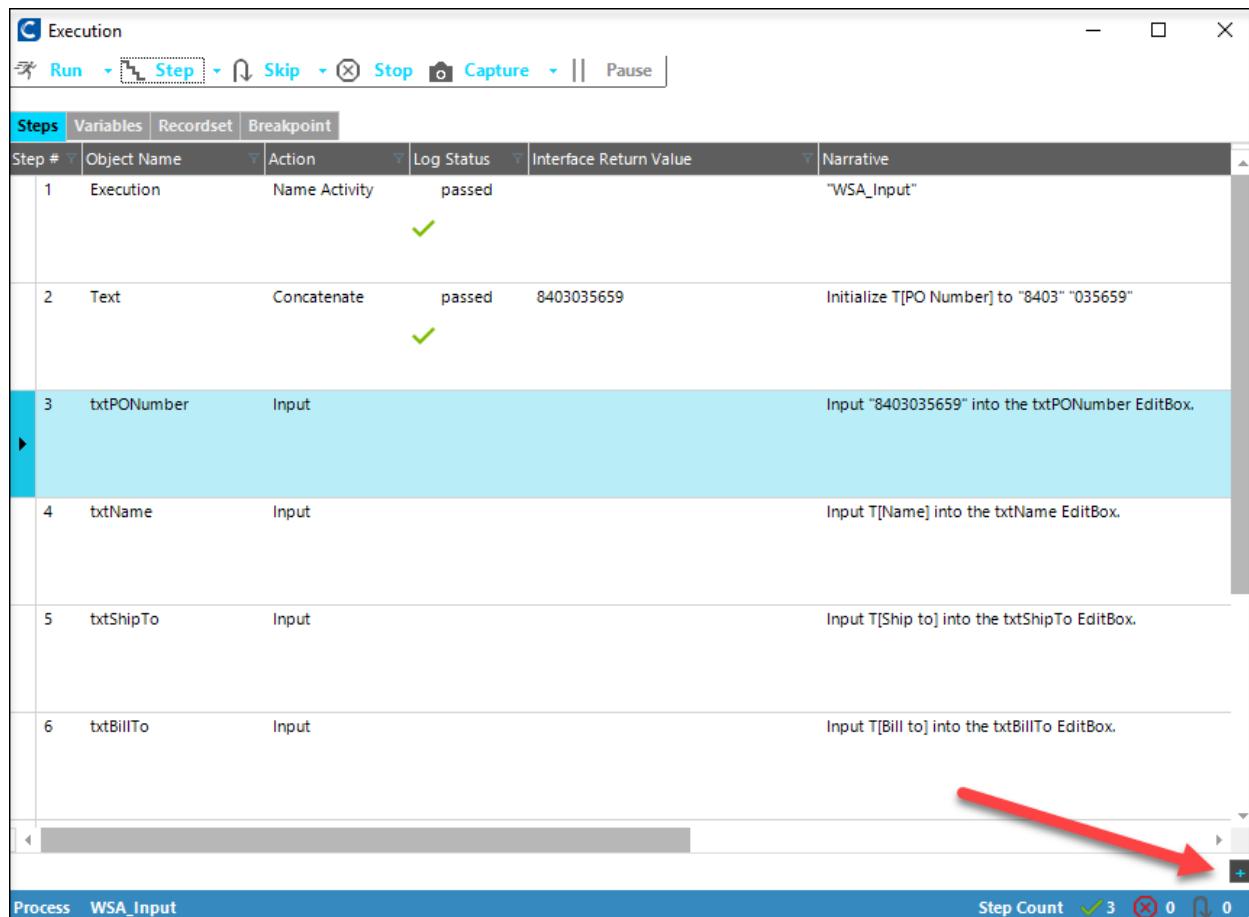
At the bottom, the process name is 'WSA_Input' and the step count is 'Step Count ✓ 1 ✘ 0 ⏪ 0 ⏩ 0'.

Notice that the Variable names are displayed. The values for the variables will display when a step containing a variable is about to be executed.

9. Press the **Step** button to execute the Name Activity step.

10. Press the **Step** button again.

11. Click the **Expand**  button in the lower, right-hand corner.



The screenshot shows the 'Execution' tool interface. At the top, there are buttons for Run, Step, Skip, Stop, Capture, and Pause. Below that is a navigation bar with tabs: Steps (selected), Variables, Recordset, and Breakpoint. The main area is a table with columns: Step #, Object Name, Action, Log Status, Interface Return Value, and Narrative. The table contains the following data:

Step #	Object Name	Action	Log Status	Interface Return Value	Narrative
1	Execution	Name Activity	passed		"WSA_Input"
2	Text	Concatenate	passed	8403035659	Initialize T[PO Number] to "8403" "035659"
3	txtPONumber	Input			Input "8403035659" into the txtPONumber EditBox.
4	txtName	Input			Input T[Name] into the txtName EditBox.
5	txtShipTo	Input			Input T[Ship to] into the txtShipTo EditBox.
6	txtBillTo	Input			Input T[Bill to] into the txtBillTo EditBox.

At the bottom, there is a status bar showing 'Process WSA_Input' and 'Step Count ✓ 3 ✘ 0 ⏪ 0'. A red arrow points to the '+' button in the bottom right corner of the status bar.

12. Click the **Variables** tab.
13. Click the **Step** tab. This allows you to view and edit the value at execution. **Only edit variable values if the Recordset Mode is set to **Read Only**, otherwise the edited value will be written to your recordset.

The screenshot shows the 'Execution' window with two tabs visible: 'Steps' (selected) and 'Variables'. The 'Steps' tab displays a list of execution steps:

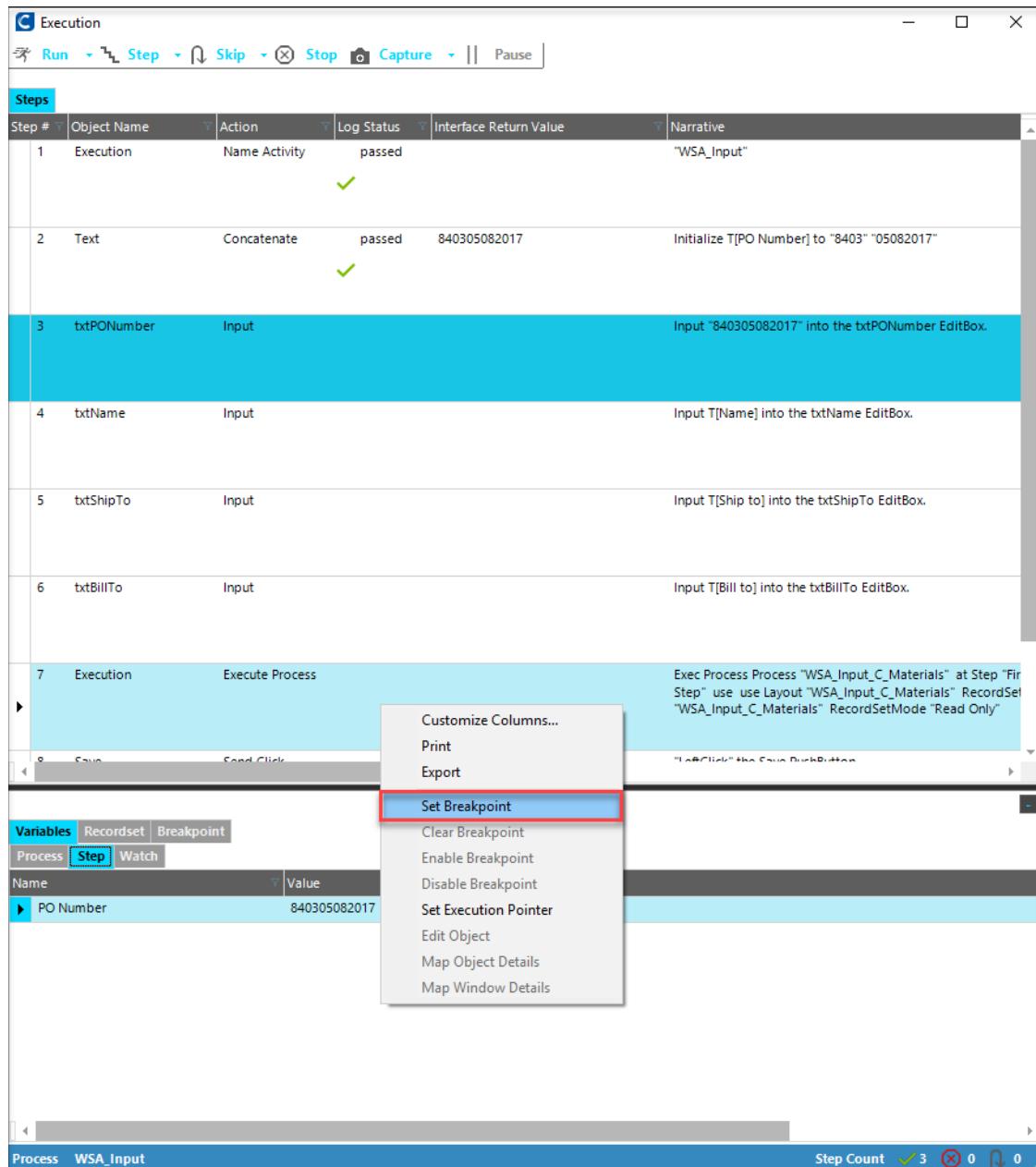
Step #	Object Name	Action	Log Status	Interface Return Value	Narrative
1	Execution	Name Activity	passed		"WSA_Input"
2	Text	Concatenate	passed	8403035659	Initialize T[PO Number] to "8403" "035659"
3	txtPONumber	Input			Input "8403035659" into the txtPONumber EditBox.
4	txtName	Input			Input T[Name] into the txtName EditBox.
5	txtShipTo	Input			Input T[Ship to] into the txtShipTo EditBox.
6	txtBillTo	Input			Input T[Bill to] into the txtBillTo EditBox.

The 'Variables' tab shows a single variable entry:

Name	Value
PO Number	8403035659

At the bottom of the window, the status bar shows 'Process WSA_Input' and 'Step Count ✓ 3 ✘ 0 ⏪ 0'.

14. Right-click on **Step 7**, and select **Set Breakpoint**.



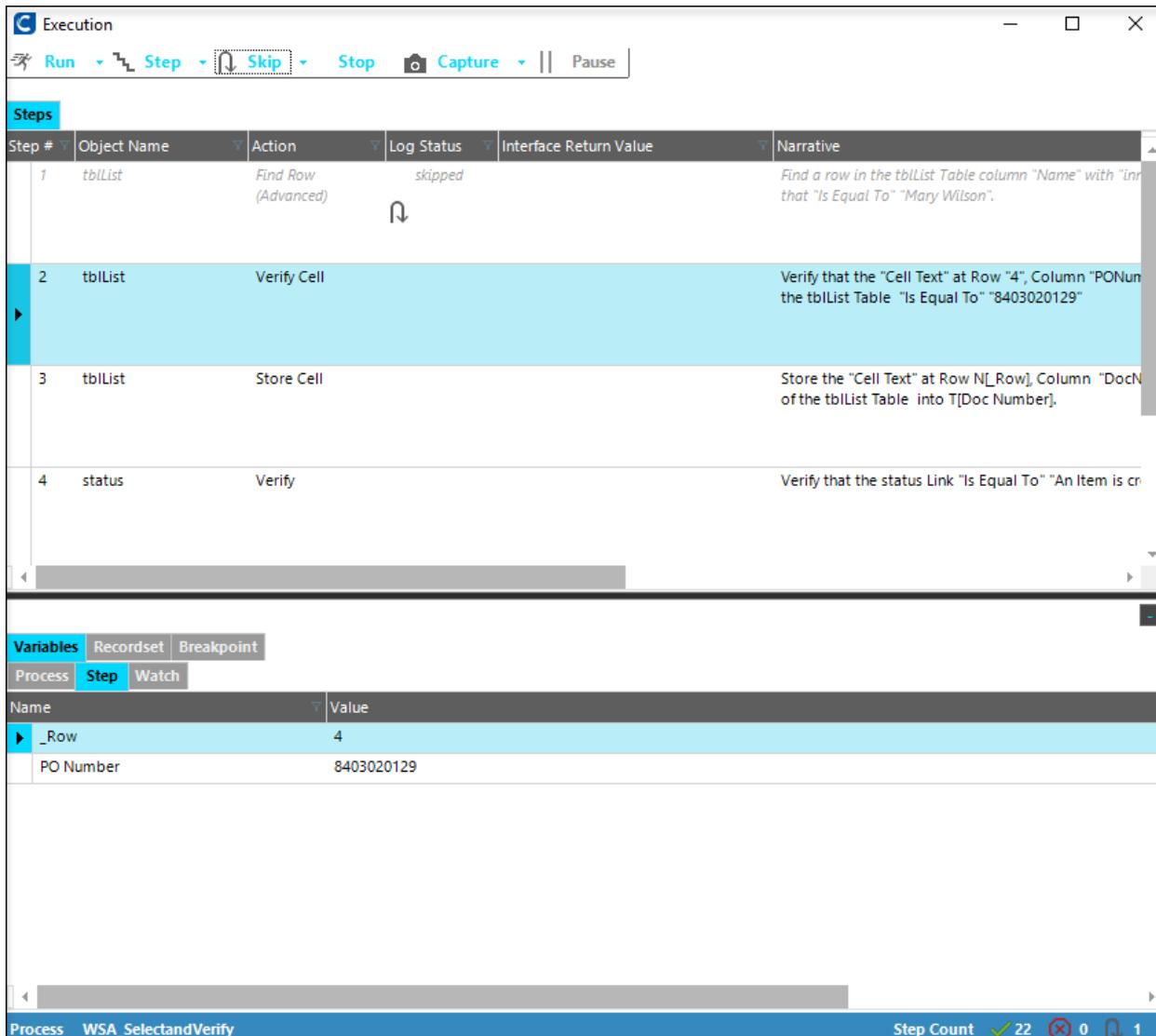
15. Press **Run**. Certify will execute until it reaches the breakpoint.
16. Click the **Step** drop-down list, and select **Step Out**. Certify will execute the rest of the sub-process steps and bring the execution back to the main process.

17. Press the **Step** button.

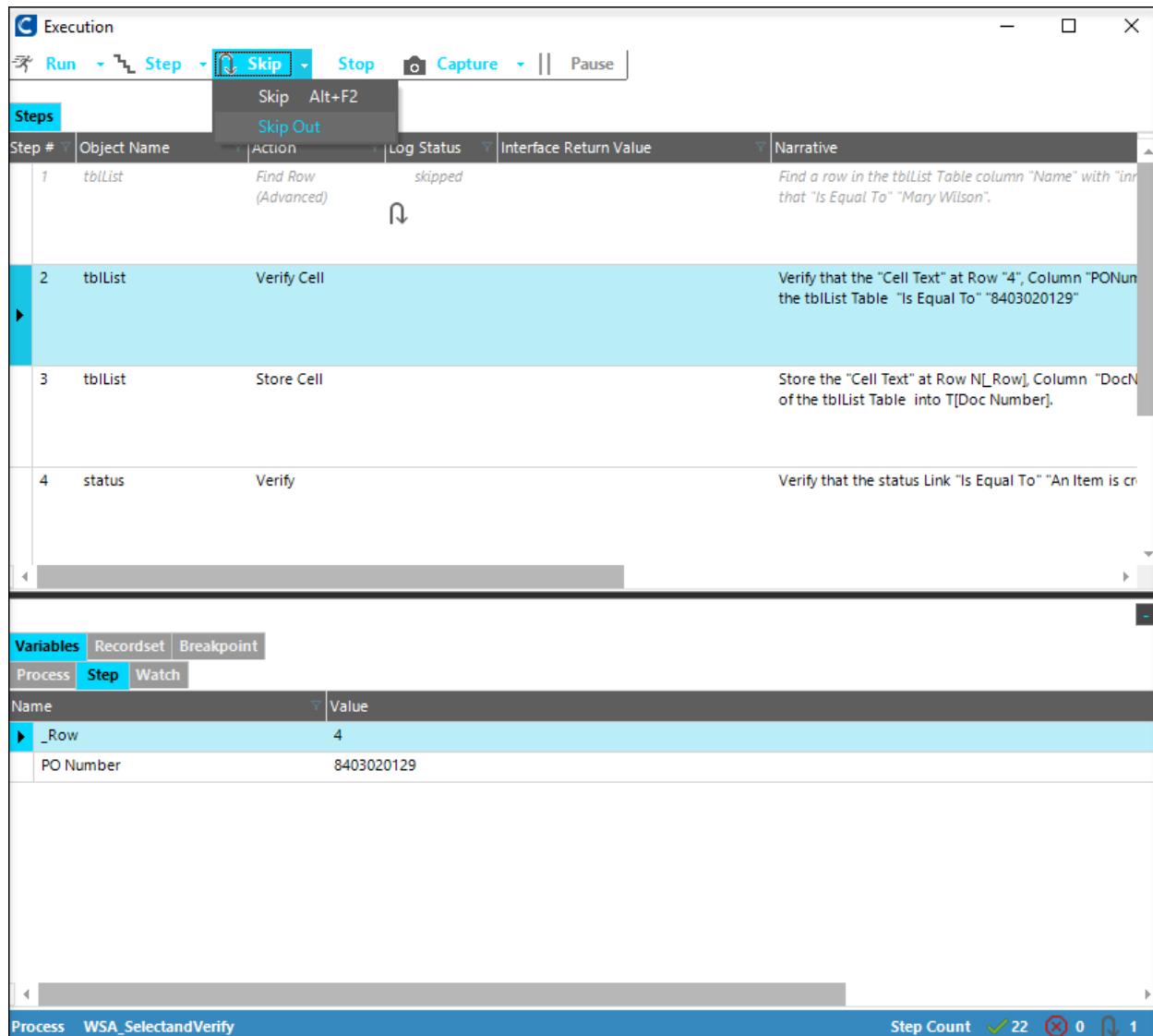
This will open the next sub-process.

18. Press the **Skip** button.

Certify will skip the first step in the sub-process. The Execution window will be positioned on Step 2 of the current sub process.



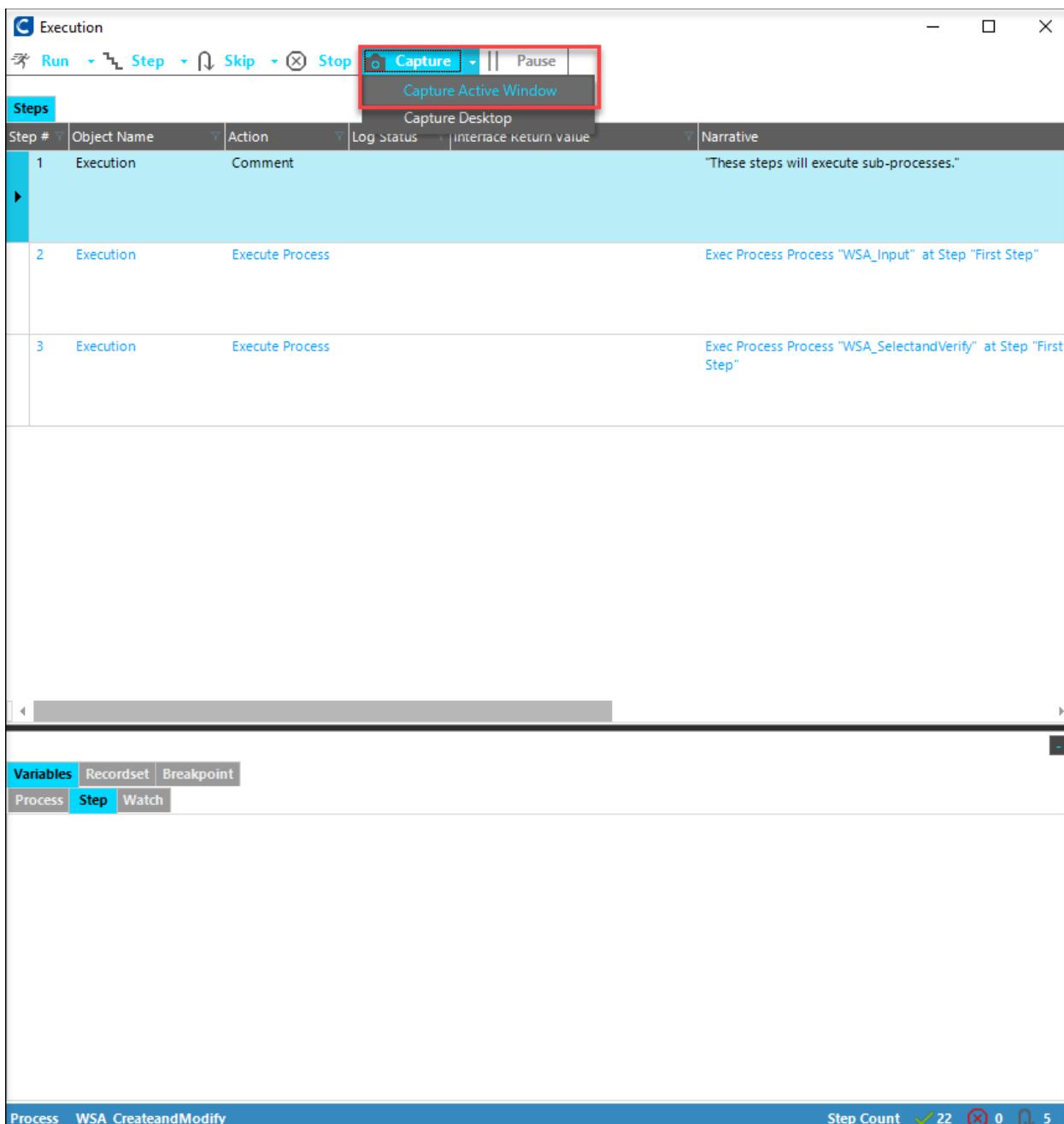
19. Click the **Skip** drop-down list, and select **Skip Out**.



Certify will skip the rest of the steps for this sub-process.

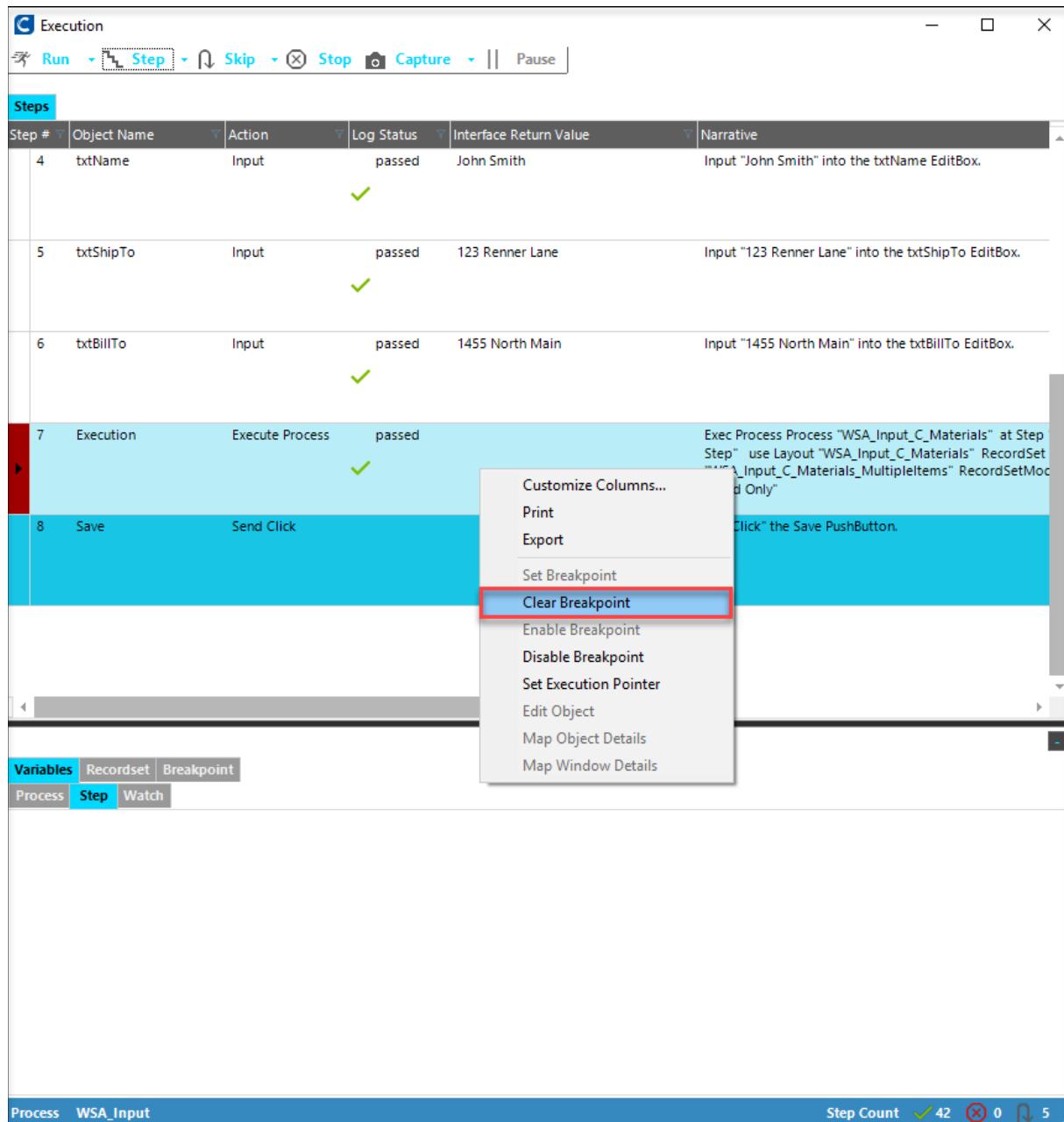
20. Click the **Capture** drop-down list, and select **Capture Active Window**.

A screen capture will be included in the Results Log. This can be used at any point you want to manually capture a screen during execution.



21. Press **Run**. Certify will execute steps until the Breakpoint at Step 7.
22. Click the **Step** drop-down list, and select **Step Over**. The **sub-process of Step 7** will execute and the execution will be brought back to the last step of the current sub-process.

23. Right-click **Step 7**, and select **Clear Breakpoint**. The breakpoint must be cleared or all future executions will stop at Step 7.



24. Press the **Step** button.
25. Press the **Stop** button to end the execution. This will give this execution the status of **Aborted**.

26. Navigate through the Results Viewer to see where steps skipped, failed, and passed.

The screenshot shows the 'Result Viewer' application window. The left sidebar displays a tree view of process results, with the 'WSA_Input' node expanded. The main area shows a table of 'Steps' with columns: Test Step ID, Application Version Name, Window, Object Name, Action Name, Narrative, Log Status, and Elapsed Time. A specific step, '65402 Execution', is selected and detailed in the bottom pane under the 'Details' tab. The step details include: Object Name: Execution, Result: True; Action Name: Name Activity, Result Action: None; Narrative: "WSA_Input", Log Status: passed; Start Time: 4/10/2017 1:37:19 PM, Execution Status: ContinueExecution; End Time: 4/10/2017 1:37:19 PM, Error Message:; Elapsed Time: 0.0338; Layout:; Recordset:.

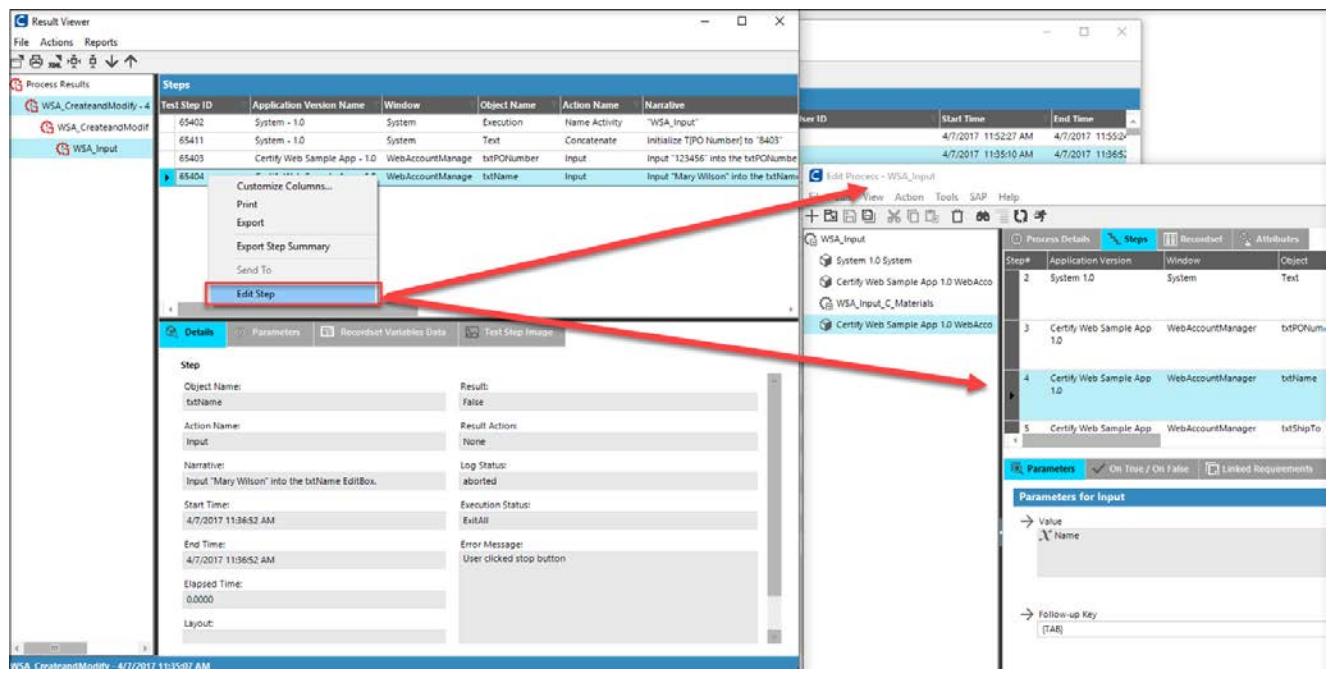
Test Step ID	Application Version Name	Window	Object Name	Action Name	Narrative	Log Status	Elapsed Time
65402	System - 1.0	System	Execution	Name Activity	"WSA_Input"	passed	0.0338
65411	System - 1.0	System	Text	Concatenate	Initialize T[PO Number] to "8403"	passed	0.0211
65403	Certify Web Sample App - 1.0	WebAccountManage	txtPONumber	Input	Input "8403013719" into the	passed	2.9045
65404	Certify Web Sample App - 1.0	WebAccountManage	txtName	Input	Input "John Smith" into the txtName	passed	2.8750
65405	Certify Web Sample App - 1.0	WebAccountManage	txtShipTo	Input	Input "123 Renner Lane" into the	passed	2.8381
65406	Certify Web Sample App - 1.0	WebAccountManage	txtBillTo	Input	Input "1455 North Main" into the	passed	2.8463
65417	System - 1.0	System	Execution	Execute Process	Exec Process Process	passed	18.3245
65409	Certify Web Sample App - 1.0	WebAccountManage	Save	Send Click	"LeftClick" the Save PushButton.	passed	0.5666

27. Close the Results Viewer.

Editing Process Steps from the Result Viewer

To quickly edit steps, you can navigate from a step in the Result Viewer to the corresponding step in the Process and Data Editor. As shown in Figure 3, once you select and right-click the step you wish to review and/or edit in the Result Viewer Summary Pane, you can select **Edit Step** from the list. The Process and Data Editor window opens to the step you selected. From here, you can modify the step and run the test process again.

Figure 3 — Editing Process Steps from the Result Viewer



Generating Execution Reports

Generating reports on execution results is performed from the Summary Pane of the Results window. From here you can run a report that shows a summary of the selected results from the process level or details of the results including step components, as well as, a recordsets report that displays only steps with values of the selected results.

Note: If a PDF viewer (such as Adobe Acrobat) is not installed on the workstation the execution report will still be created. If this is the case, copy the file to an area where a workstation that does have a PDF viewer can open the file.

The execution reports are displayed in .pdf format. As shown in Figure 4, the report shows the results in the same order they are displayed in the Results Viewer (order of execution).

Figure 4 — Sample Execution Report

Process Execution Report (Summary)

WORKSOFT CERTIFY.

Process: WSA_CreateandModify

Project: Client Project Name
Folder: Results
Status: passed
Layout: WSA_Input
Recordset: WSA_Input
Recordset Mode: ReadUpdate
Description:

Created By: admin
Date Created: 10/24/16
Executed By: admin
Date Executed: 10/24/16

Process: WSA_CreateandModify **Legend :**

Process Status: passed **Passed**: ✓
Layout: WSA_Input **Failed**: ✗
Recordset: WSA_Input **Skipped**: ⚡
Recordset Mode: ReadUpdate **Aborted**: !
Recordset Current Row: 1 **Pending**: ⏱

Application: System - 1.0
Window: System
✓ "These steps will create sub-processes"
✓ Exec Process Proces "WSA_Input" at Step "First Step"

Process: WSA_Input **Legend :**

Process Status: passed **Passed**: ✓
Layout: **Failed**: ✗
Recordset: **Skipped**: ⚡
Recordset Mode: **Aborted**: !
Recordset Current Row: **Pending**: ⏱

Application: System - 1.0
Window: System
✓ Initialize TPO Number to "070802"

Application: New_CertifyWebSampleApp - 1.0
Window: WebAccountManager
✓ Input "070802" Into the txtPONumber EditBox - (Follow-up Key = "None")
✓ Input "Mary Wilson" Into the txtName EditBox - (Follow-up Key = "None")
✓ Input "100 Fifth Street" Into the txtShipTo EditBox - (Follow-up Key = "None")

Page 1

10/25/2016 11:28:14 PM
© 2010 Worksoft Inc. All Rights Reserved.

worksoft.com

EXERCISE 5.3 — Generating an Execution Results Report from the Results Window

In this exercise, you will run a summary report of the execution results for reviewing or reporting needs.

Step	Action
1.	In the Navigation Taskbar, click Results .
2.	In the Summary Pane, right-click the desired process and select Reports .
3.	Select Results (Summary) .
	<i>The Choose a location... window appears.</i>
4.	Choose a location to save the report.
5.	In the File Name field, type the desired name for the report.
6.	Click Save .

The report opens in .pdf format.

The screenshot shows a PDF document titled "Process Execution Report (Summary)" from the WORKSOFT CERTIFY platform. The report details the execution of a process named "WSA_CreateandModify".

Process Information:

- Project: Client Project Name
- Folder: Results
- Status: passed
- Layout: WSA_Input
- Recordset: WSA_Input
- Recordset Mode: ReadUpdate
- Description:

Created By: admin
Date Created: 10/24/16
Executed By: admin
Date Executed: 10/24/16

Legend:

Status	Icon
Passed	✓
Failed	✗
Skipped	▢
Aborted	☒
Pending	○

Process Details:

WSA_CreateandModify

- Process Status: passed
- Layout: WSA_Input
- Recordset: WSA_Input
- Recordset Mode: ReadUpdate
- Recordset Current Row: 1

Application: System - 1.0

- Window: System
- ✓ These steps will create sub-processes
- ✓ Exec Process Process "WSA_Input" at Step "First Step"

WSA_Input

- Process Status: passed
- Layout:
- Recordset:
- Recordset Mode:
- Recordset Current Row:

Application: System - 1.0

- Window: System
- ✓ Initialize T[PO Number] to "070802"

Application: New_CertifyWebSampleApp - 1.0

- Window: WebAccountManager
- ✓ Input "070802" Into the txtPONumber EditBox - (Follow-up Key = "None")
- ✓ Input "Mary Wilson" Into the txtName EditBox - (Follow-up Key = "None")
- ✓ Input "100 Fifth Street" Into the txtShipTo EditBox - (Follow-up Key = "None")

Page 1

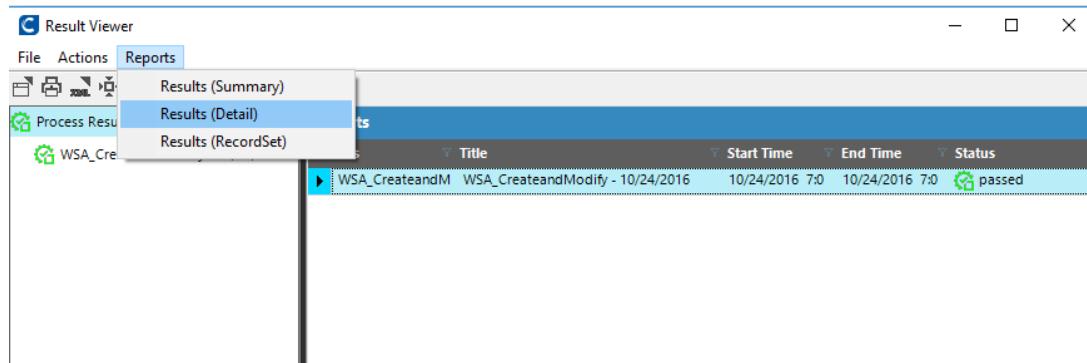
10/25/2016 11:28:14 PM
© 2010 Worksoft Inc. All Rights Reserved.
worksoft.com

EXERCISE 5.4 — Generating an Execution Results Report from the Result Viewer

In this exercise, you will run a detailed report of the execution results for reviewing or reporting needs. The same report could be run from the Results window.

Step	Action
------	--------

1. Select **Process Results** at the top of the Result Viewer Navigation Tree.
2. In the menu bar, select **Reports** and then **Results (Detail)**.



The *Choose a location...* window appears.

3. Choose a location to save the report.
4. In the **File Name** field, type the desired name for the report.
5. Click **Save**.

The report opens in .pdf format.

Promoting Results to “Result of Record”

You can identify a specific execution result as a **Result of Record**, which means that result is the official execution result used to determine the status of processes. For a particular iteration, you may have several executions, and this helps isolate the important ones.

This is not a required step unless you are using the Requirements feature in Certify.

If you are using the Requirements feature in Certify, the status and execution of each requirement linked to each process in the selected execution will be updated. This will help you determine which test requirements have completely passed.

When you promote a result to a Result of Record, the result will appear bolded and a **Result of Record** field is checked. If you promote a new result to a Result of Record, a message box appears alerting you that promoting the new result will overlay an existing Result of Record. Once you click OK to the message, the previously promoted results for the same process will be replaced and the result will no longer appear bolded.

Resetting Result of Record

You can reset results that have already been promoted to a Result of Record by right-clicking the result in the Summary Pane and selecting **Reset From Result of Record**. Once reset, the result will no longer appear bolded and the status for each requirement linked to each process in the selected execution will be updated with an Unknown status.

EXERCISE 5.5 — Promoting a Result to Result of Record

This exercise walks through the steps to promote a result to Result of Record.

Step	Action
1.	In the Navigation Taskbar, click Results .
2.	In the Summary Pane, right-click the desired result.
	<i>A short-cut menu appears.</i>
3.	Select Promote to Result of Record .

The result is bolded and a check mark appears in the Result of Record box.

Process	Title	Created By User	Result Of Record	Status	Start Time	End Time
WSA_Input	WSA_Input - 5/11/2017 1:59:16 PM	sclark	<input checked="" type="checkbox"/>	passed	5/11/2017 1:59:23 PM	5/11/2017 1:59:23 PM
WSA_Input	WSA_Input - 5/11/2017 2:01:11 PM	sclark	<input type="checkbox"/>	passed	5/11/2017 2:01:17 PM	5/11/2017 2:01:17 PM
WSA_Input	WSA_Input - 5/11/2017 2:19:05 PM	sclark	<input type="checkbox"/>	aborted	5/11/2017 2:19:09 PM	5/11/2017 2:19:09 PM
WSA_Input	WSA_Input - 5/11/2017 2:20:08 PM	sclark	<input type="checkbox"/>	aborted	5/11/2017 2:20:11 PM	5/11/2017 2:20:11 PM
WSA_Input	WSA_Input - 5/11/2017 2:21:40 PM	sclark	<input type="checkbox"/>	passed	5/11/2017 2:21:43 PM	5/11/2017 2:21:43 PM
WSA_CreateandModify	WSA_CreateandModify - 5/11/2017 3:06:42	sclark	<input type="checkbox"/>	passed	5/11/2017 3:06:45 PM	5/11/2017 3:06:45 PM
WSA_CreateandModify	WSA_CreateandModify - 5/11/2017 3:09:53	sclark	<input type="checkbox"/>	passed	5/11/2017 3:10:03 PM	5/11/2017 3:10:03 PM
WSA_CreateandModify	WSA_CreateandModify - 5/11/2017 3:14:43	sclark	<input type="checkbox"/>	passed	5/11/2017 3:14:46 PM	5/11/2017 3:14:46 PM
WSA_CreateandModify	WSA_CreateandModify - 5/11/2017 3:20:37	sclark	<input type="checkbox"/>	passed	5/11/2017 3:20:40 PM	5/11/2017 3:20:40 PM
WSA_EditQuantityandPrice	WSA_EditQuantityandPrice - 5/11/2017	sclark	<input type="checkbox"/>	aborted	5/11/2017 3:30:03 PM	5/11/2017 3:30:03 PM
WSA_EditQuantityandPrice	WSA_EditQuantityandPrice - 5/11/2017	sclark	<input type="checkbox"/>	aborted	5/11/2017 3:32:10 PM	5/11/2017 3:32:10 PM
WSA_EditQuantityandPrice	WSA_EditQuantityandPrice - 5/11/2017	sclark	<input type="checkbox"/>	aborted	5/11/2017 5:59:05 PM	5/11/2017 5:59:05 PM
WSA_EditQuantityandPrice	WSA_EditQuantityandPrice - 5/11/2017	sclark	<input type="checkbox"/>	failed	5/11/2017 6:02:25 PM	5/11/2017 6:02:25 PM

Once a test result has been promoted to Result of Record, the Promoted status on the linked process of the corresponding requirement is updated from Unknown to **passed**.

Exporting the Test Process to a .BAT File

Certify allows you to export any process to a Microsoft batch file (.BAT) that can then be executed by double-clicking on the file name from Windows Explorer or DOS prompt. The file can also be used by a scheduler to kick off test runs at predetermined times and days.

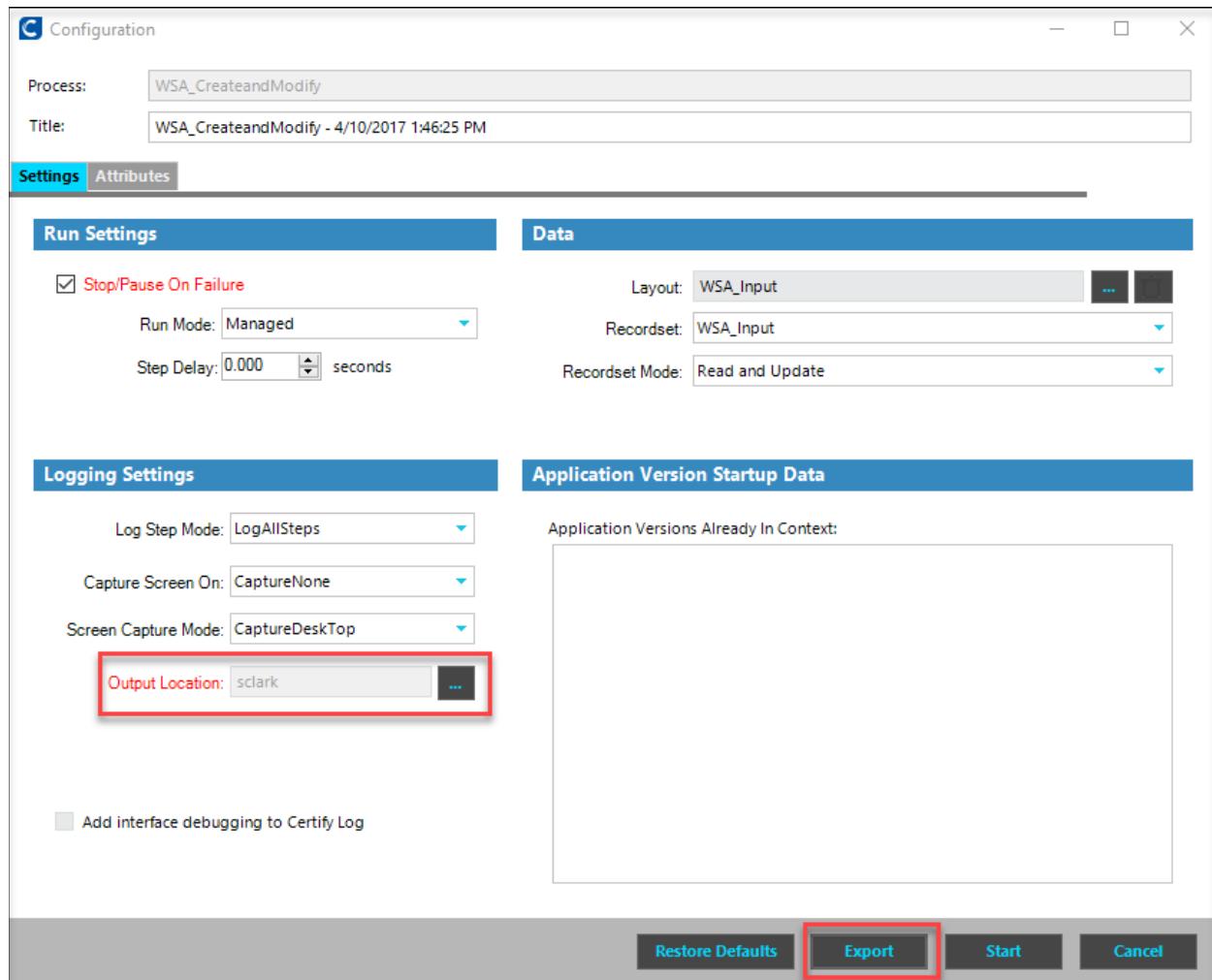
Note: Refer to the Worksoft Community Site <http://community.worksoft.com> for a complete description of Worksoft's Lights Out methodology for unattended test execution.

EXERCISE 5.6 — Creating a .BAT File for a Process

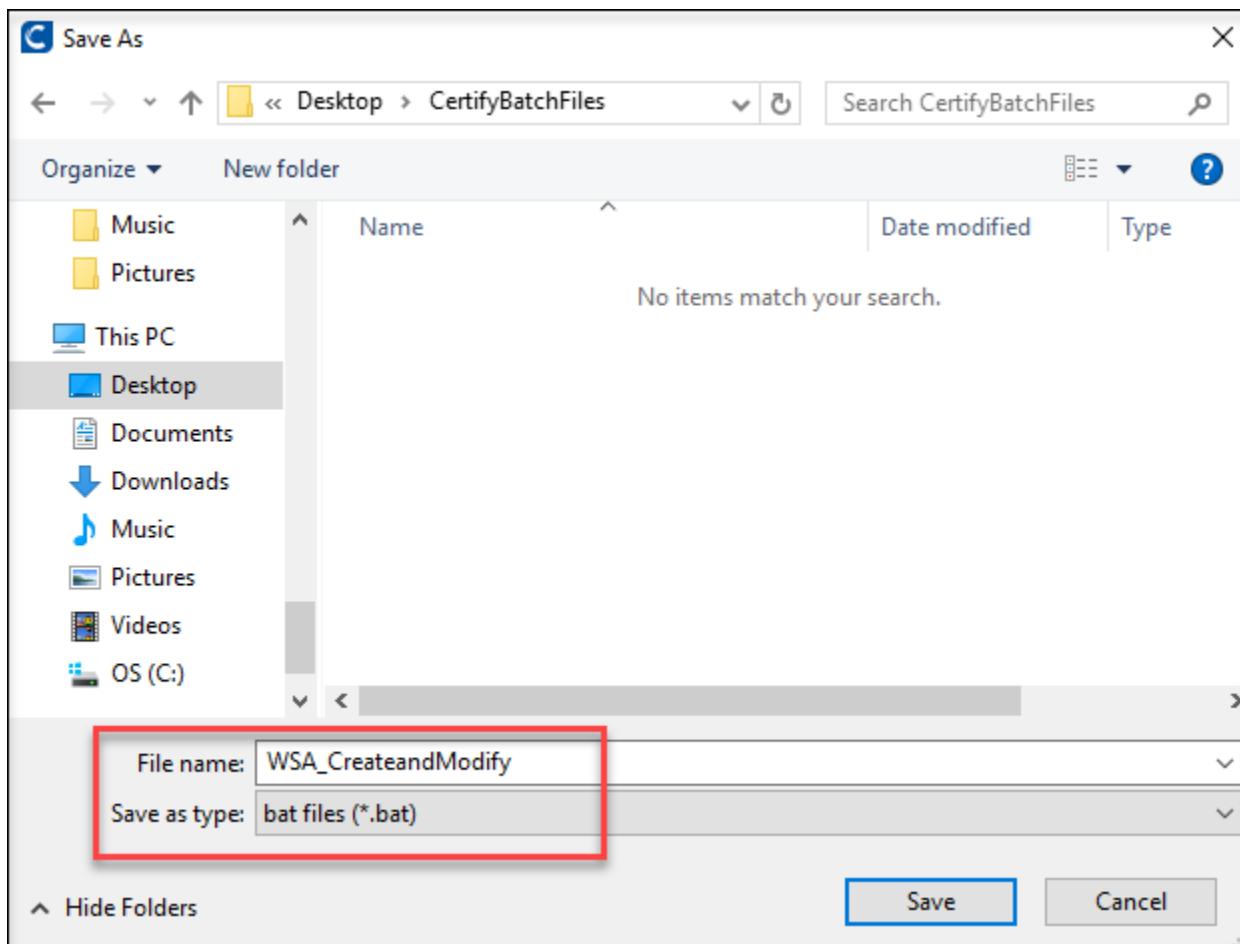
In this exercise, you will create a batch file that contains all the configuration information for the current process.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, click your WSA_CreateandModify folder.
3.	In the Summary Pane, right-click WSA_CreateandModify and select Run or press the 
	<i>The Configuration dialog box appears. For Batch File execution, you may wish to create and specify a specific folder to hold results.</i>
4.	Next to the Output Location, click the  button to select your Results folder (if you haven't already).
5.	Navigate to your Results folder, and click OK .

6. Click the **Export** button.



7. Navigate to a folder on your workstation or network. You may wish to create a specific folder that contains Certify batch files.

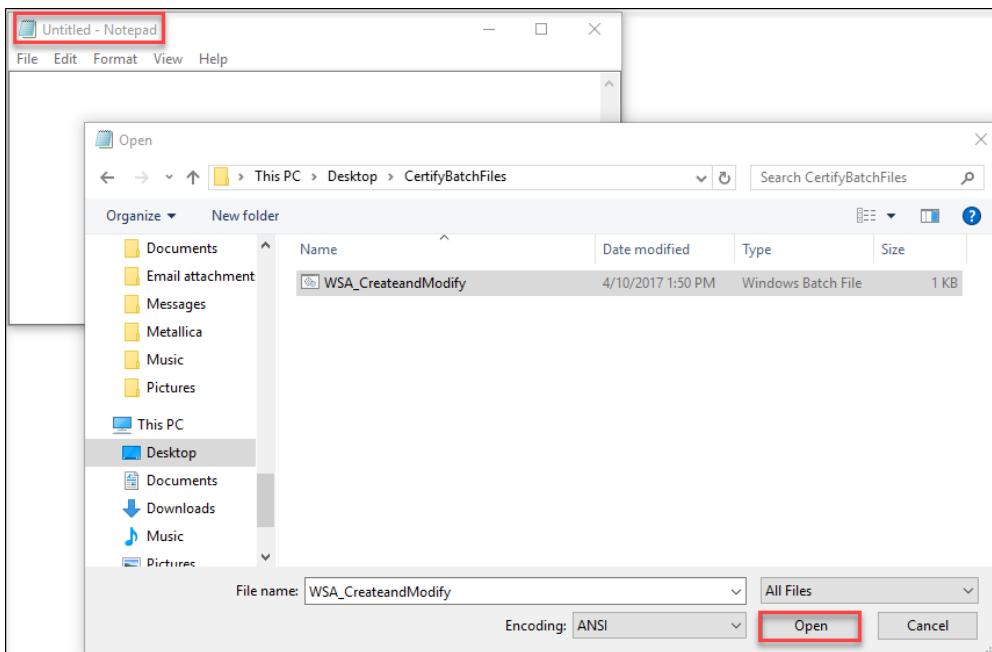


8. Type in a File name (ending in .BAT) and click **Save**.

9. Open the file with **Notepad** for editing.

Note: The example file created here is not completely ready for unattended execution.

- The batch file requires a Certify User and Password to be entered on the command line.
- The process expects that the Worksoft Web Sample application is open to the PO page. For true unattended execution, this process should be updated to include steps to launch and close the application.



10. Open the **Format menu, and select **Word Wrap**.**

A screenshot of a 'Notepad' window titled 'WSA_CreateandModify - Notepad'. The 'Format' menu is open, with the 'Word Wrap' option highlighted. The main text area contains a batch script:

```

@ECHO OFF
CLS
ECHO.
ECHO Worksoft Inc - Commandline Utility
ECHO.
if '%1' == '' goto usage
if '%2' == '' goto usage

Certify.exe /usecertifyconfig /stepdelay=0 /stoponfailure /Process="02. Client Project Name
(Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_CreateandModify" /Project="Worksoft"
/Recordset="WSA_Input" /RecordsetsMode="Read and Update" /Layout="02. Client Project Name
(Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_Input" /Attribute="QC TestName|"
/Attribute="QC TestRun|" /Attribute="QC TestSet|" /OutputLocation="sclark/"
/VerifyObjects=Disabled /user="%1" /password="%2"

GOTO finish

:Instructions On Parameters for Command Line
:usage
ECHO.
ECHO USAGE:
ECHO.
ECHO 2 Parameters are required. Please provide them
ECHO in the following order separated by a space:
ECHO.
ECHO    Certify User ID
ECHO    Certify Password

```

11. Delete the beginning text highlighted in blue in the screenshot below.

The screenshot shows a Notepad window titled "WSA_CreateandModify - Notepad". The content is a batch script. The first few lines are highlighted in blue:
@ECHO OFF
CLS
ECHO.
ECHO Worksoft Inc - Commandline Utility
ECHO.
if '%1' == '' goto usage
if '%2' == '' goto usage

Below this, the script continues with:
Certify.exe /usecertifyconfig+ /stepdelay=0 /stoponfailure /Process="02. Client Project Name (Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_CreateandModify" /Project="Worksoft" /Recordset="WSA_Input" /RecordsetsMode="Read and Update" /Layout="02. Client Project Name (Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_Input" /Attribute="QC TestName|" /Attribute="QC TestRun|" /Attribute="QC TestSet|" /OutputLocation="sclark/" /VerifyObjects=Disabled /user="%1" /password="%2"
GOTO finish

:Instructions On Parameters for Command Line
:usage
ECHO.
ECHO USAGE:
ECHO.
ECHO 2 Parameters are required. Please provide them
ECHO in the following order separated by a space:
ECHO.
ECHO Certify User ID
ECHO Certify Password

12. Delete the ending text highlighted in blue in the screenshot below.

The screenshot shows a Notepad window titled "WSA_CreateandModify - Notepad". The content is a batch script. The last few lines are highlighted in blue:
Certify.exe /usecertifyconfig+ /stepdelay=0 /stoponfailure /Process="02. Client Project Name (Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_CreateandModify" /Project="Worksoft" /Recordset="WSA_Input" /RecordsetsMode="Read and Update" /Layout="02. Client Project Name (Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_Input" /Attribute="QC TestName|" /Attribute="QC TestRun|" /Attribute="QC TestSet|" /OutputLocation="sclark/" /VerifyObjects=Disabled /user="%1" /password="%2"
GOTO finish

:Instructions On Parameters for Command Line
:usage
ECHO.
ECHO USAGE:
ECHO.
ECHO 2 Parameters are required. Please provide them
ECHO in the following order separated by a space:
ECHO.
ECHO Certify User ID
ECHO Certify Password
ECHO.
:finish

13. Modify the User ID and Password to provide values. These values need to be your Certify User ID and Password so that the batch file can access the Certify database.



```
WSA_CreateandModify - Notepad
File Edit Format View Help
Certify.exe /usecertifyconfig+ /stepdelay=0 /stoponfailure /Process="02. Client Project Name (Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_CreateandModify" /Project="Worksoft" /Recordset="WSA_Input" /RecordsetsMode="Read and Update" /Layout="02. Client Project Name (Development)\02.04 Sandbox\SClark\WSA_CreateandModify\WSA_Input" /Attribute="QC TestName|" /Attribute="QC TestRun|" /Attribute="QC TestSet|" /OutputLocation="sclark/" /VerifyObjects=Disabled /user=admin /password=password
```

14. Save the file.
15. Double-click the file to execute. **Make sure the Web Sample Application is open.**

Lesson Summary

You've completed the **Advanced Executing Processes, Troubleshooting and Viewing Results** lesson.

Key points to remember:

- The Configuration dialog box provides options for choosing the type of execution, how and when execution is performed, and how the results are handled.
- Once configuration is complete, you can set additional execution functionality in the Execution dialog box, such as skipping process steps, capturing application windows, and setting breakpoints.
- After process execution, the Result Viewer shows a log of the execution results.
- The Result Viewer allows you to expand the execution results to show each process and step executed. You can expand each level of the hierarchy and view information about the executed processes and steps in the Summary Pane.
- Generating reports for execution results can be performed from the Summary Pane of the Results window. From here you can run a report that shows a summary of the selected results from the process level or details of the results including step components, as well as a recordsets report that displays only steps with values of the selected results.
- Certify allows you to export any process to a Microsoft batch file (.BAT) that can then be executed by double-clicking on the file name from Windows Explorer or DOS prompt. The file can also be used by a scheduler to kick off test runs at predetermined times and days.

Lesson 6

Advanced Layouts and Recordsets

Overview

In the previous lessons, you learned about several ways to create and use layouts and recordsets. In this lesson, you will use recordsets with a filter and variables to develop data-driven testing.

Objectives

After completing this lesson, you will be able to:

- Create a recordset filter.
- Create a browser launch utility.
- Create a browser close utility.

Recordset Filters

Recordset filters are used to select rows of a recordset that will be used during execution of a child process.

The left pane of Figure 1 contains the variables from the WSA_Input_C_Materials layout.

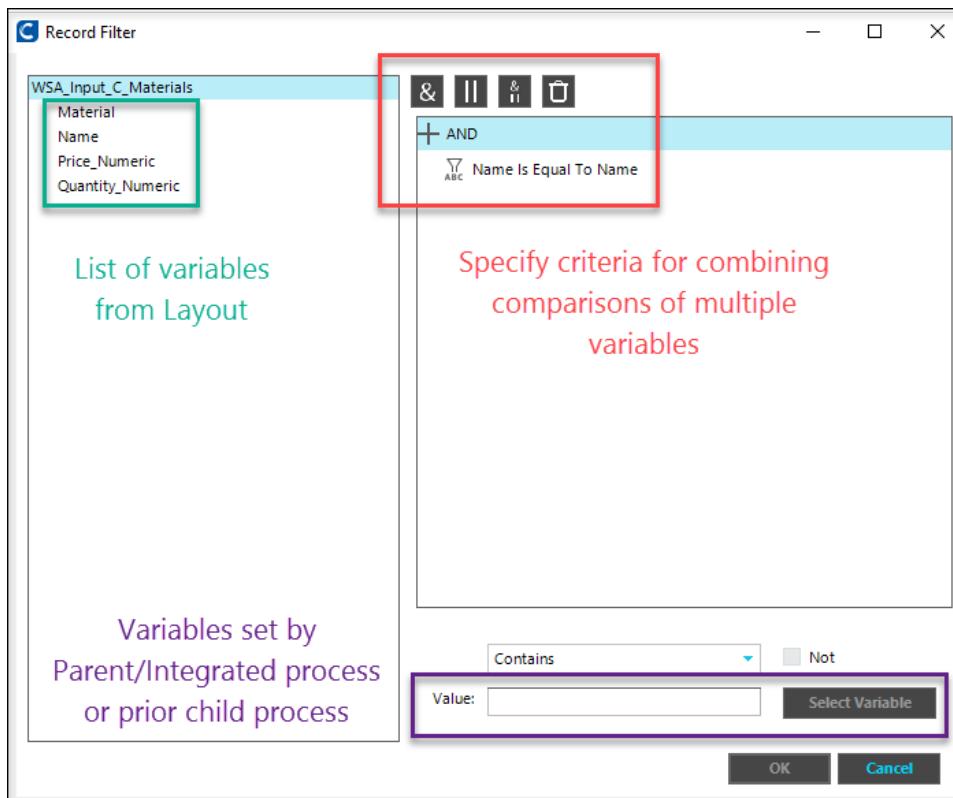
The bottom-right pane of Figure 1 is used to select a variable for comparison and the criteria for the comparison. The criteria list has many options such as **Contains**, **Is Equal To**, and **Is Empty**. These can be made “negative” by selecting the **Not** checkbox.

The top-right pane of Figure 1 contains a list of the selection criteria as set using the left pane and bottom-right panes. Multiple variables can be compared in a recordset filter.

For example, you may select the Variable **Name**, **Is Empty**, and the **Not** checkbox to include records where the **Name is Not Empty**.

Note: In many examples the Variables being compared are from two Layouts. The variable selected at the bottom for comparison does not need to be in a Layout and could be based on a value read from the application. The variable names do not need to match either.

Figure 1: Record Filter



Recordset Filter for the Worksoft Web Sample Application

The WSA_Input_C_Materials process will insert and create the same materials each time the process is executed. We want to modify this behavior so that when we create a purchase order (PO), only specific materials are added for that PO.

We need a common variable to use as the filtering criteria. Currently, there isn't a variable used in both the **WSA_Input Layout** and in the **WSA_Input_C_Materials Layout**. We need to find a good candidate from the WSA_Input layout that can be added to the WSA_Input_C_Materials_MultipleItems layout for specific materials to be added during execution.

The PO Number field will vary each time we run the process, so it would not be a good candidate. The Name does not vary so we can use that to tie the two Layouts/Recordsets together.

WSA_Input Recordset

	PO Number	Name	Ship to	Bill to	Doc Number
1	8403020129	Mary Wilson	1444 North Ave	100 Fifth Street	1498848518652
2	8403024536	John Smith	123 Renner Lane	1455 North Main	1498848549117
*					

WSA_Input_C_Materials_MultipleItems Recordset

	Material	Quantity_Numeric	Price_Numeric
1	Smart TV	10	785
2	Phone	7	780
3	Watch	9	300
*			

In the next exercise, we will add the Name variable to the WSA_Input_C_Materials Layout and assign a name from the WSA_Input recordset to each material.

WSA_Input_C_Materials_MultipleItems Recordset - after adding Name variable

Name	Material	Quantity_Numeric	Price_Numeric	
Mary Wilson	Smart TV	10	780	
John Smith	Phone	7	480	
Mary Wilson	Watch	9	300	

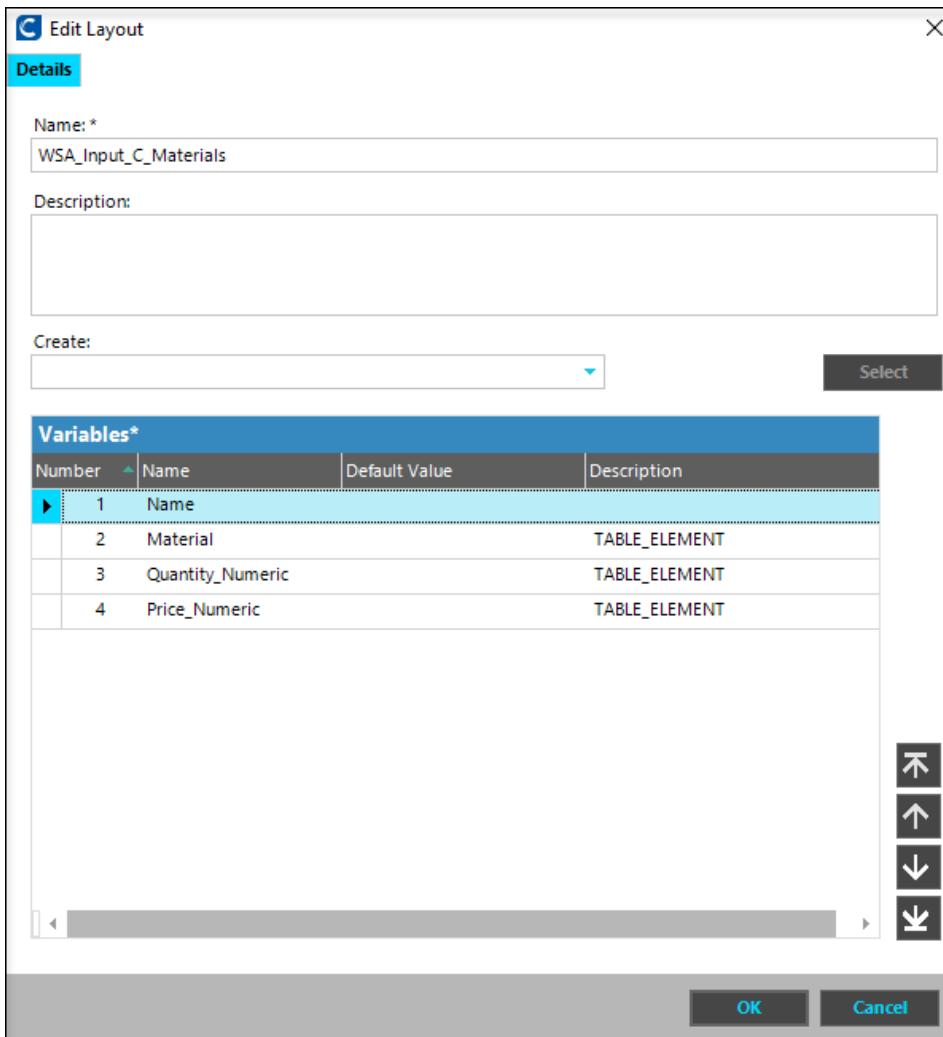
EXERCISE 6.1 — Using a Recordset Filter

This exercise will show you how to modify the WSA_Input_C_Materials layout to add an additional column that will be used as a recordset filter.

Step	Action
1.	From the Data window, in the Navigation Tree, find your WSA_CreateandModify folder.
2.	Right-click on the WSA_Input_C_Materials Layout and select Edit .
3.	Right-click on the first variable and select Insert .

The screenshot shows the 'Edit Layout' dialog for the 'WSA_Input_C_Materials' layout. In the 'Variables*' section, there are three variables listed: 'Material' (Number 1), 'Quantity_Numeric' (Number 2), and 'Price_Numeric' (Number 3). A context menu is open over the 'Material' variable, with the 'Insert' option highlighted in blue and a red box. Other options in the menu include 'Add', 'Edit', 'Delete', and movement keys ('Move to Top', 'Move Up', 'Move Down', 'Move to Bottom'). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

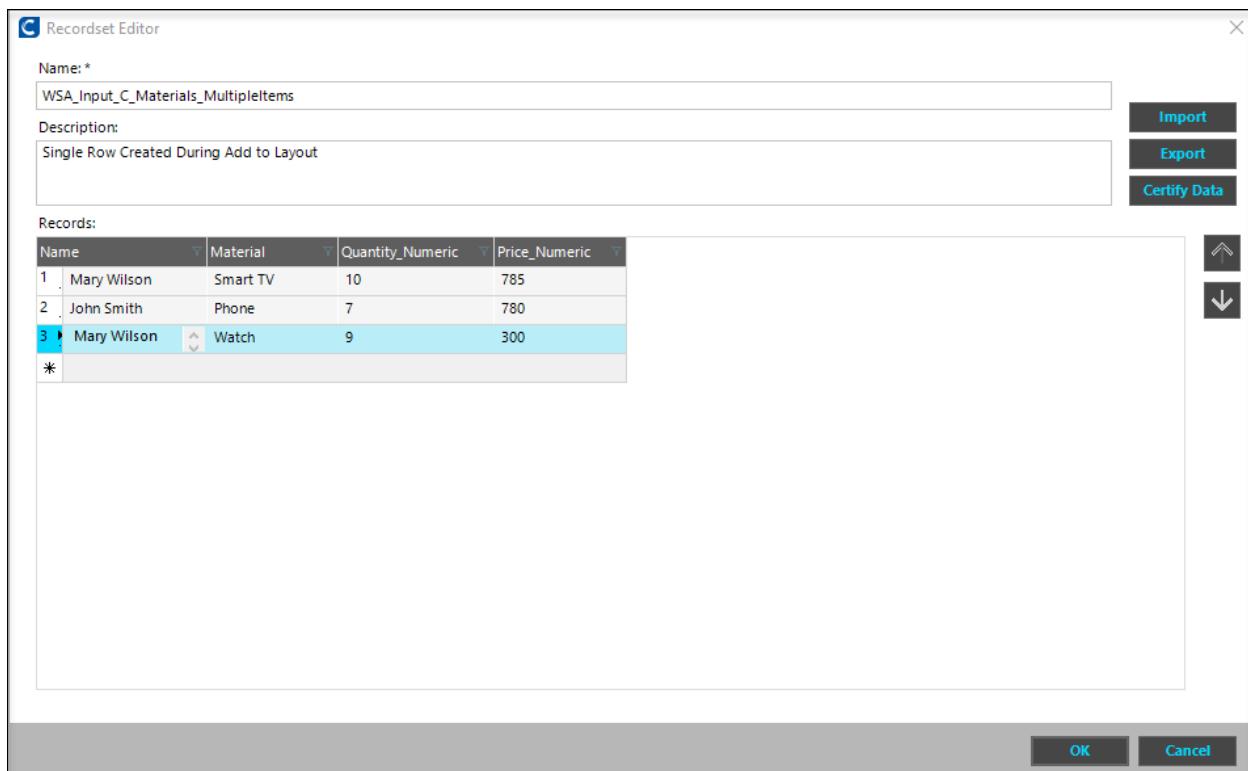
4. Select the **Name** variable from the list.
5. Press **OK** to close the Variable window.



6. Press **OK** to close the Edit Layout window.
7. Click the **Recordsets** Tab at the bottom of the screen.

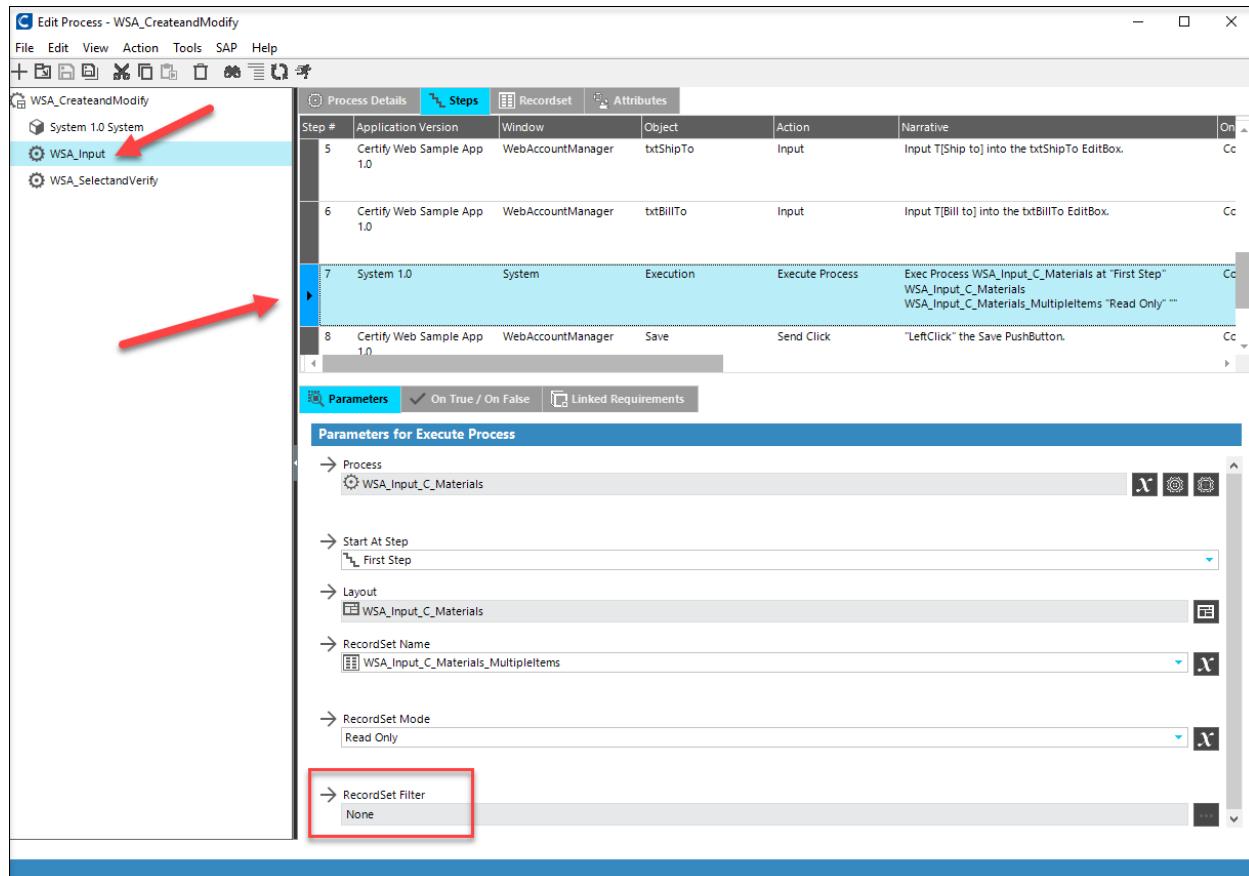
8. Double-click on the **WSA_Input_C_MultipleItems** recordset to edit the recordset. Include a Name for each transaction.

You may export/import the recordset if you wish to use another editor.



9. Press **OK** to close the Recordset Editor Window.
10. Select **Processes** in the Navigation Taskbar. In the Navigation Tree, click your **WSA_CreateandModify** folder.
11. If it is not open, double-click the **WSA_CreateandModify** process in the Summary Pane.

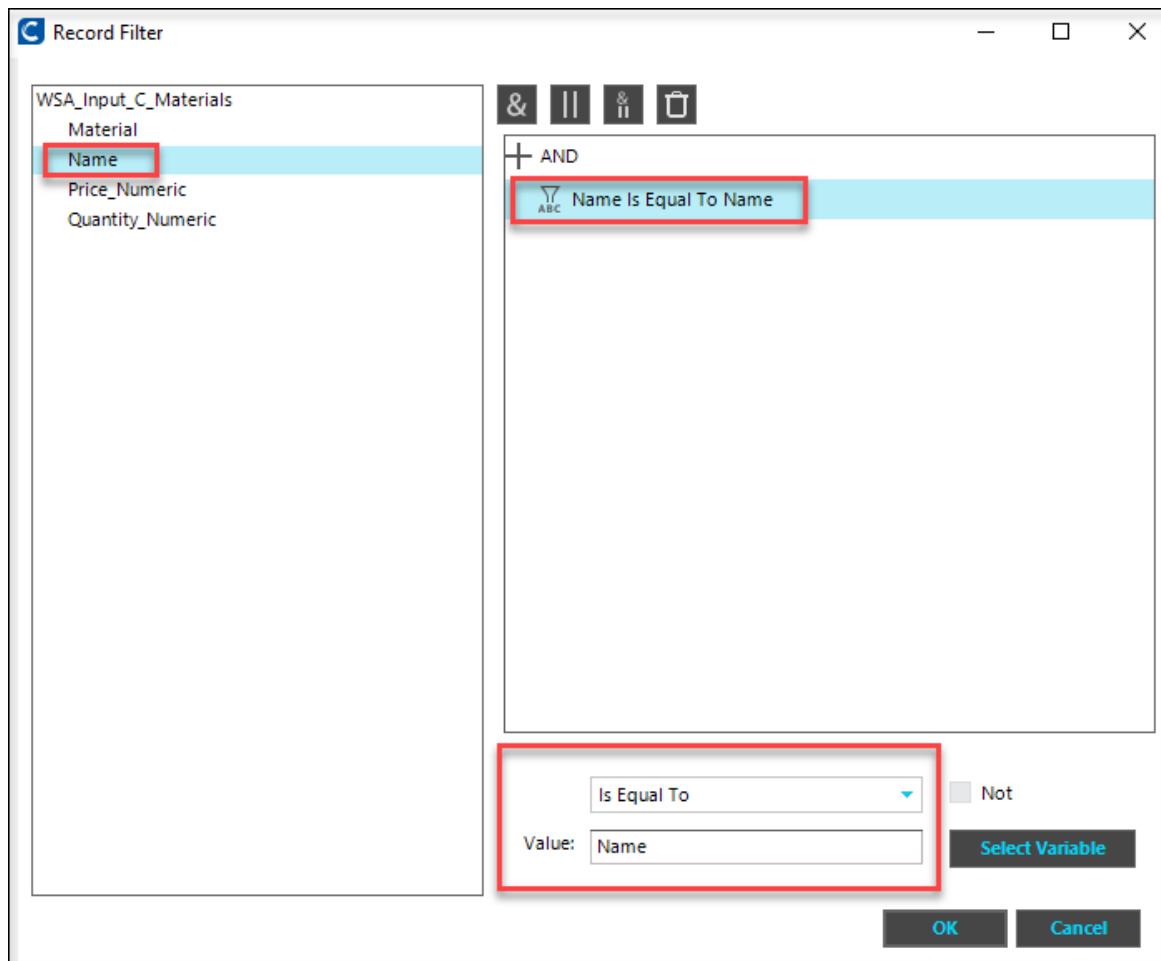
12. Select the **WSA_Input** process in the Navigation Tree.
13. Select **Step 7** – this step executes the WSA_Input_C_Materials sub-process.
14. In the Parameters of Step 7, the Recordset Filter option is at the bottom of the parameter list. You may need to scroll down to see this option.



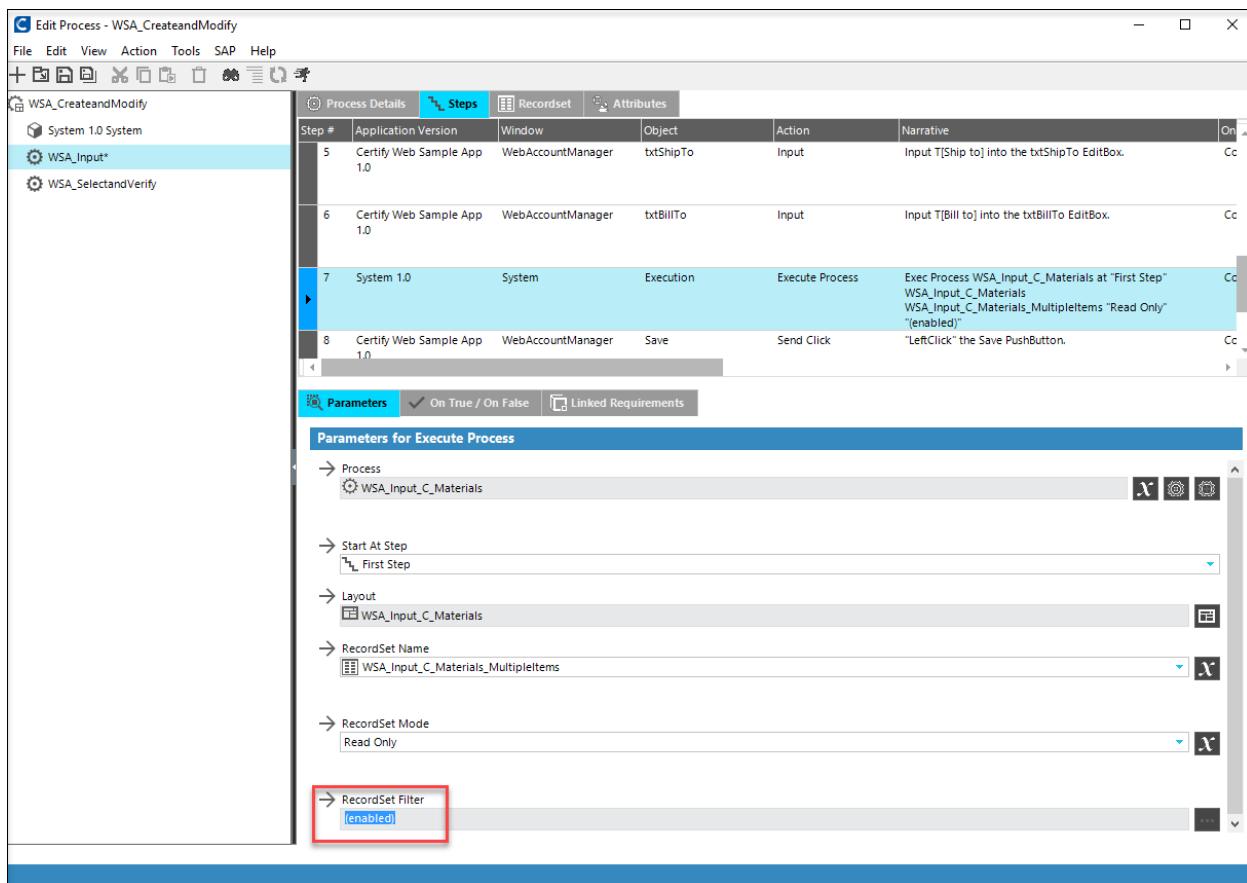
15. Press the Browse button next to the Recordset Filter parameter.



16. Double-click on the **Name** variable on the left side. This brings the Name variable in to the Criteria pane.
17. Verify the Name variable is in the value field next to the Select Variable button.
18. Select **Is Equal To** from the drop-down list.



19. Press **OK** to close the Record Filter window.
20. You will notice that Recordset filter option now shows as **enabled**.



21. Press **Save** to save the process.

The Recordset Filter is now set to (enabled).

22. In the Process Navigation Tree, select the **WSA_CreateandModify** process.
23. Click the **Recordset** tab.
24. Note the Names listed in the recordset.
25. In the Process Navigation Tree, select the **WSA_Input_C_Materials** process.

*The steps on the right side should now reflect the **WSA_Input_C_Materials** process.*

26. Click the **Recordset** tab.

27. Verify that the recordset includes a Name for each material and that those names are in the **WSA_Input** recordset.

The screenshot shows the SAP Fiori Launchpad with the process 'Edit Process - WSA_CreateandModify' selected. The 'Recordset' tab is active, showing a table with three rows of data:

	Name	Material	Quantity_Numeric	Price_Numeric
1	Mary Wilson	Smart TV	10	785
2	John Smith	Phone	7	780
3	Mary Wilson	Watch	9	300

28. Execute the WSA_CreateandModify process.

The screenshot shows the Result Viewer application interface. On the left, there is a tree view labeled "Process Results" showing the execution path: WSA_CreateandModify - 4/7/2017 4:12:2 → WSA_CreateandModify → WSA_Input → WSA_Input_C_Materials → WSA_SelectandVerify. The main area is titled "Steps" and contains a table with the following data:

Test Step ID	Application Version Name	Window	Object Name	Action Name	Narrative
65402	System - 1.0	System	Execution	Name Activity	"WSA_Input"
65411	System - 1.0	System	Text	Concatenate	Initialize T[PO Number] to "8403"
65403	Certify Web Sample App - 1.0	WebAccountManage	txtPONumber	Input	Input "8403041311" into the
65404	Certify Web Sample App - 1.0	WebAccountManage	txtName	Input	Input "John Smith" into the txtName
65405	Certify Web Sample App - 1.0	WebAccountManage	txtShipTo	Input	Input "123 Renner Lane" into the
65406	Certify Web Sample App - 1.0	WebAccountManage	txtBillTo	Input	Input "1455 North Main" into the
65417	System - 1.0	System	Execution	Execute Process	Exec Process Process
65409	Certify Web Sample App - 1.0	WebAccountManage	Save	Send Click	"LeftClick" the Save PushButton.

Below the table, there is a "Details" tab selected, showing the following step details:

- Object Name: Execution, Result: True
- Action Name: Name Activity, Result Action: None
- Narrative: "WSA_Input", Log Status: passed
- Start Time: 4/7/2017 4:13:11 PM, Execution Status: ContinueExecution
- End Time: 4/7/2017 4:13:11 PM, Error Message:
- Elapsed Time: 0.0019
- Layout:
- Recordset:

The bottom status bar shows "WSA CreateandModify - 4/7/2017 4:12:29 PM".

EXERCISE 6.2 — Modify the WSA_CreateandModify Process to Launch Application

This exercise will show how to modify the WSA_CreateandModify process to launch a browser and the Worksoft Web Sample Application.

Step	Action
------	--------

1. Select Processes in the Navigation Tree, and select your **WSA_CreateandModify** folder.
2. Open the **WSA_CreateandModify** process for editing.
3. Right-click **Step 1**, and select **Insert Step Below**.
4. Modify the new step as shown below:

Step	Application Version	Window	Object	Action
2	System 1.0	Browser	Browser	Load Browser
Browser				
Internet Explorer				
URL				
C:/Program Files (x86)/Worksoft/Samples/WebSampleApplication//index.html				

5. Right-click **Step 2**, and select **Insert Step Below**. Modify as shown below:

Step	Application Version	Window	Object	Action
3	Worksoft Web Sample Application 1.0	WebAccountManager	WebAccountManager	Set Window State
Window State				
Maximize				

6. If the **Window State** is already in **Maximize** mode, then we don't want our step to fail. Therefore, we will modify the Parameters for this step as follows
7. Click on Step 3.
8. Click the **On True/On False** tab.
9. Change the **On False Log Status As**: **Skipped**.

The screenshot shows the 'Steps' tab of a process configuration. The process consists of five steps:

- Step 1: System 1.0, Window: System, Object: Execution, Action: Comment, Narrative: "These steps will execute sub-processes.", On True: Continue, On False: Continue.
- Step 2: System 1.0, Window: Browser, Object: Browser, Action: Load Browser, Narrative: "Load \\127.0.0.1\c\$\Users\sclark\Dropbox (Worksoft)\ChrmMaxSampleApplication\WebSample", On True: Continue, On False: Continue.
- Step 3: Worksoft Web Sample Application 1.0, Window: WebAccountManager, Object: WebAccountManager, Action: Set Window State, Narrative: "Maximize the WebAccountManager Page.", On True: Continue, On False: Continue. This step is highlighted with a red box.
- Step 4: System 1.0, Window: System, Object: Execution, Action: Execute Process, Narrative: "Exec Process WSA_Input at 'First Step'", On True: Continue, On False: Continue.
- Step 5: System 1.0, Window: System, Object: Execution, Action: Execute Process, Narrative: "Exec Process WSA_SelectandVerify at 'First Step'", On True: Continue, On False: Continue.

Below the steps, the 'On True / On False' configuration is shown for Step 3. The 'On True' section has 'Passed' selected in the 'Log Status As:' dropdown. The 'On False' section has 'Skipped' selected in the 'Log Status As:' dropdown, which is highlighted with a red box.

This will set the status of Step 3 to Skip when the window is already set to Maximize

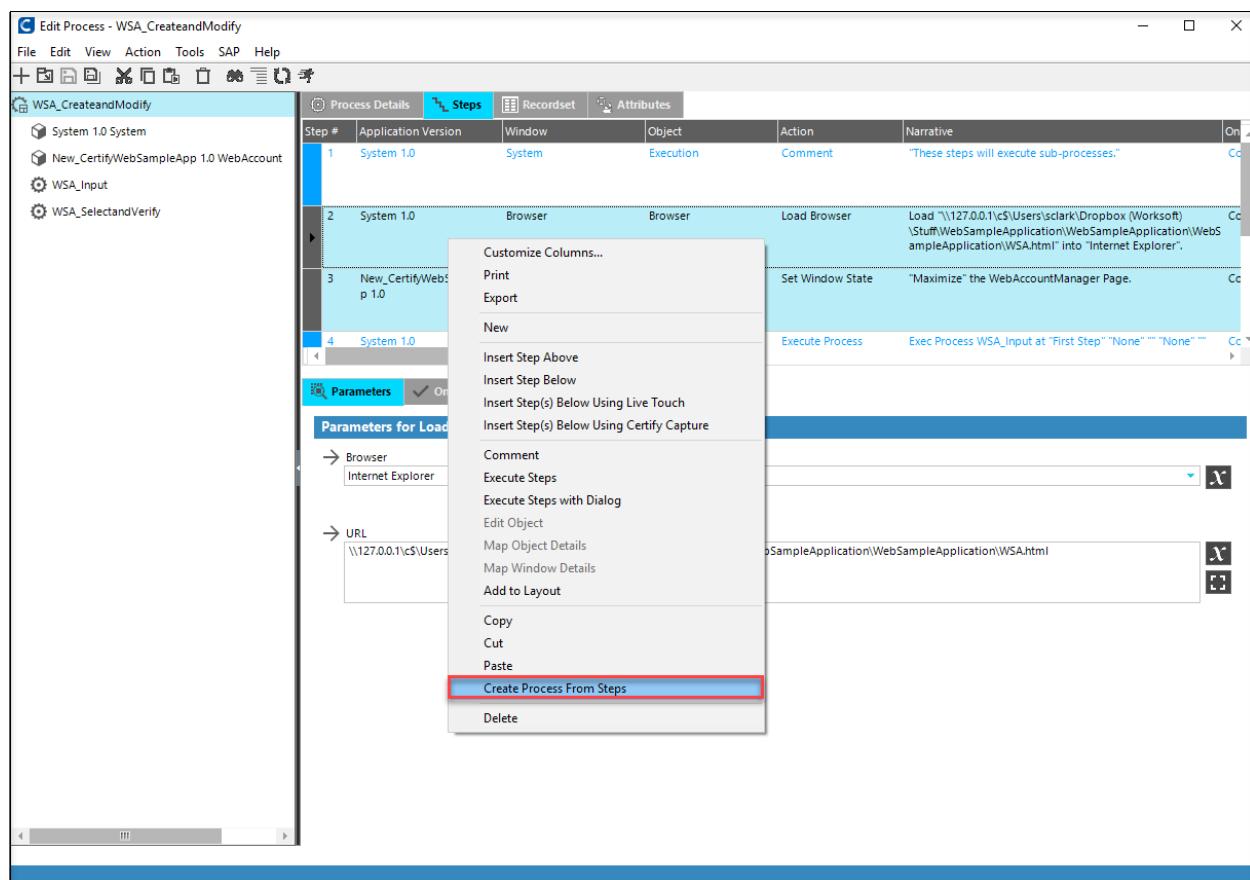
- 10.** Press the **Save**  button.

EXERCISE 6.3 — Creating the UTL_WSA_Launch Process from Steps

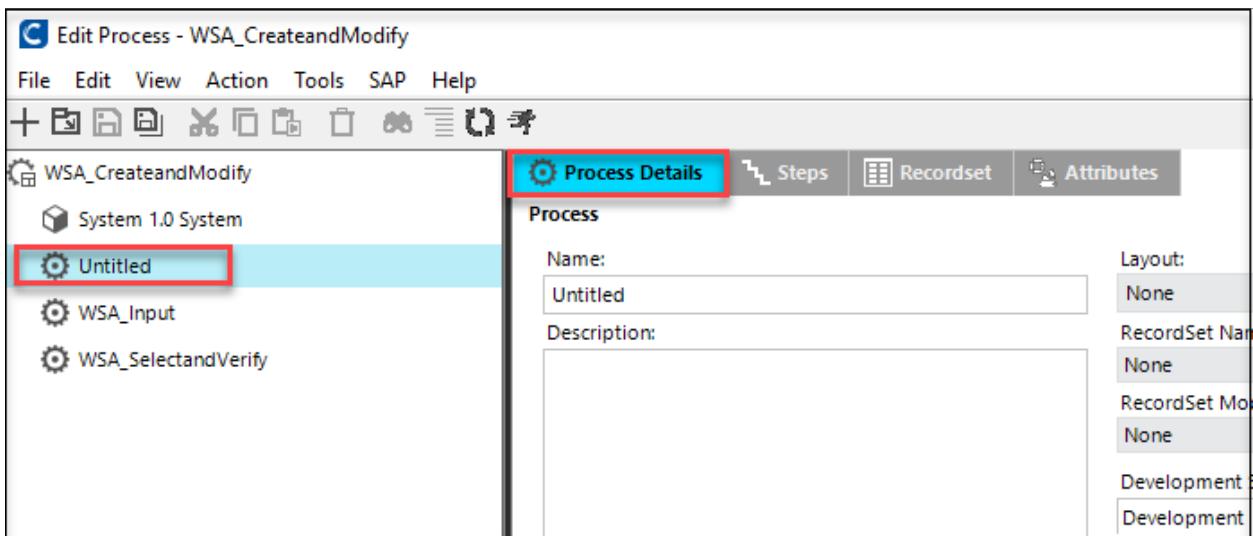
This exercise will show how to create a sub-process from existing steps.

Step	Action
------	--------

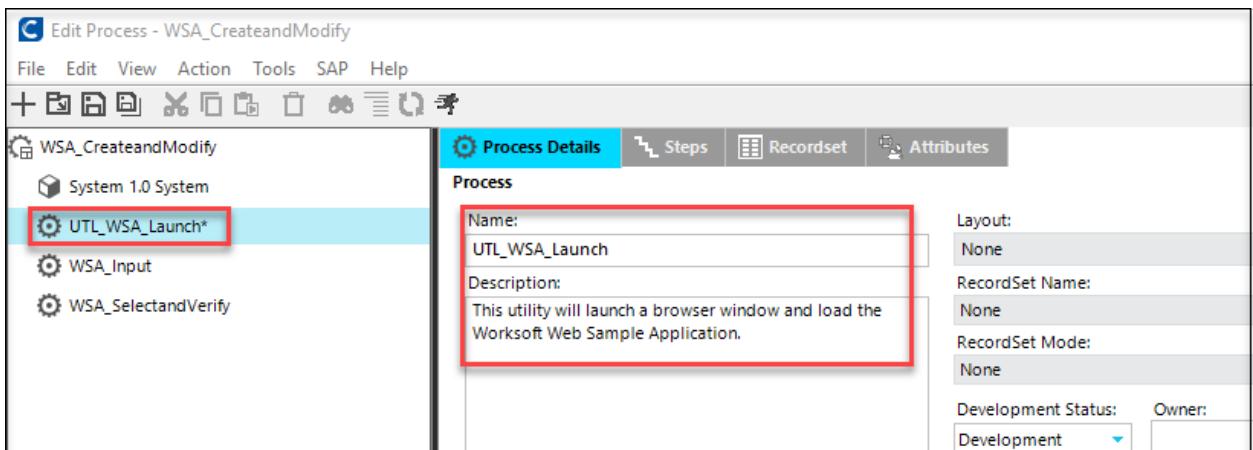
1. In the **WSA_CreateandModify** process editor, select **Steps 2-3** then right-click and click **Create Process From Steps**.



2. When asked if you wish to continue creating the sub-process, click **OK**.
3. Find and click on the **Untitled** process in the Navigation Tree.



4. Navigate to the Process Details tab. Name the new process: **UTL_WSA_Launch**.
5. Enter the description: **This utility will launch a browser window and load the Worksoft Web Sample Application.**



6. Save both processes.
7. Click Refresh .

EXERCISE 6.4 — Create the UTL_WSA_Close Process

This exercise will show how to create a process to close the Worksoft Web Sample Application.

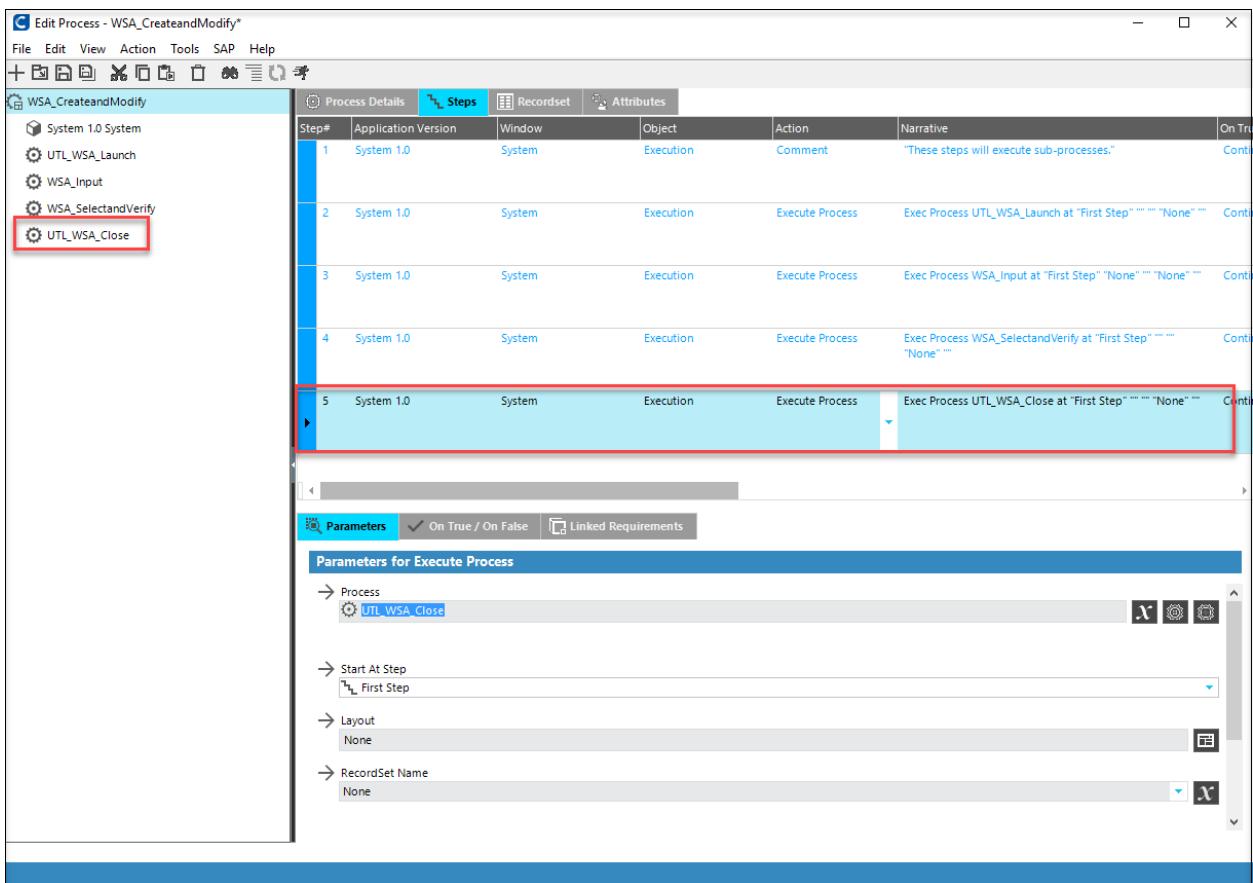
Step	Action
------	--------

1. In the Navigation Tree, select your **WSA_CreateandModify** folder.
2. Right-click in the Summary Pane, and select **New Process**.
3. Name the process: **UTL_WSA_Close**.
4. Enter the description: **This utility will close the browser window of the Worksoft Web Sample Application.**
5. Create the two steps shown below.

Step	Application Version	Window	Object	Action	
1	System 1.0	Browser	Browser	Set Page Context Timeout	
Seconds(default =20)		2			

Step	Application Version	Window	Object	Action
2	System 1.0	Browser	Browser	Close Browser

6. **Save and Close** the Process Editor.
7. Find and edit the **WSA_CreateandModify** process.
8. Add a final step to execute the **UTL_WSA_Close** process.



9. **Save and Close** the Process Editor.
10. Execute the **WSA_CreateandModify** process.

Lesson Summary

You've completed the [Advanced Layouts and Recordsets](#) lesson.

Key points to remember:

- A layout created using any method can be manually edited to add or remove variables.
- A recordset will automatically be updated with any new variables added to a layout, but you will need to provide the values.
- Recordset filters provide a dynamic way to process specific records matching a condition.
- Creating utilities to launch and close an application ensures the Application Under Test is opened and closed during the execution of your end-to-end, integrated process.

Lesson 7

Developing Advanced Processes

Overview

In this lesson, you will create advanced processes that will edit, update, and delete existing purchase orders in the Worksoft Web Sample Application by collecting data from the application and using additional Certify conditional logic and verification techniques.

Objectives

After completing this lesson, you'll be able to:

- Understand conditional logic and how it can add value to your tests.
- Plan sophisticated process validation.
- Use Certify's built-in logic tools to enhance your processes.
- Gather and store data from an application.

Advanced Processes

With processes to create and verify Purchase Orders in place, it's time to look at how to use the data created by those processes to validate the intended functionality of the Worksoft Web Sample Application.

As you have seen up to this point, creating simple tests using specific datasets is a valid way to test an application's basic functionality, but there are times when something more robust is needed; a test that can "interpret" data and act accordingly.

For example, here's a screenshot of our Web Sample Application screen for the created Purchase Orders.

Action	DocNumber	PONumber	Name	ShipTo	BillTo	Amount
 	1478629791099	122928	Mary Wilson	100 Fifth Street	1444 North Fifth Street	10500
 	1478629808127	122953	John Smith	123 Renner Ln	1455 North Main	3360

As you developed the processes that created the Purchase Orders above, you knew exactly how many Purchase Orders were created, as well as all the details. The only new information generated is the Document Number and the Amount of the Purchase Order. Developing a process to create purchase orders with known data is easy enough. However, what if you need a test that edits an existing purchase order or deletes purchase orders using a specific criterion? In such cases, you would need to develop a robust driven process.

Here's a brief comparison of simple, functional automated tests vs. robust, data-driven automated tests:

Simple Functional Test	Robust Data-Driven Test
Advantages:	Advantages:
<i>Is quicker and easier to create</i>	<i>Creates more 'thoughtful' tests</i>
<i>Requires minimal upfront thought/planning</i>	<i>Utilizes reusable functionality</i>
<i>Utilizes only basic Certify technology</i>	<i>Takes advantage of advanced Certify technology</i>
<i>Runs faster</i>	<i>Plays better with other tests</i>
<i>Meets testing and BPO requirements</i>	<i>Requires less maintenance*</i>
Disadvantages:	<i>Meets testing and BPO requirements</i>
<i>Is a static, inflexible process</i>	Disadvantages:
<i>Has only step reusability</i>	<i>Requires more upfront thought/planning</i>
<i>Every test requires complete automation</i>	<i>Initially takes more time to automate</i>
<i>Requires more maintenance*</i>	<i>Takes longer to run</i>

* more often than not

When deciding which kind of automated test to develop, there's no right or wrong choice as both types can be used to test the functionality of any application, but it's important to take the time before automating any tests to determine which option makes the most sense for your current situation.

Conditional Logic

To create a test sophisticated enough to work with unknown transactions, your process needs to be able to analyze and make decisions depending on what data it finds. In automation, this ability is called **conditional logic**. A great example and the most common conditional logic in testing are validation steps. If the result is expected, the test passed; if the result is not expected, the test failed.

Here are some common conditional logic methods. All of them start with checking for some condition and then acting accordingly:

- o **If, then else** – if the specified condition exists, do this; if not, do something else. *Compare values; if they match, do something; if they don't match, do something else.*
- o **Do while** – repeat processing as long as specified criteria is met. *If counter < 10, repeat processing.*
- o **Case** – if the current condition is this, go to this location in the process. *If Fund = Money, jump to the "Process Money" steps.*

Certify has built-in tools for adding conditional logic to your processes.

In Certify, **If, then else** is primarily managed using **On True/On False** and **Labels**. Here is an example of a Certify process using **if, then else** logic:

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Text	Compare	Verify T[Task Type] Is Equal To "Daily"	Continue	Jump
2	Service 1.0	Task Manager	Daily	[Click]	Click Daily RadioButton	Continue	Exit Process
3	System 1.0	System	Execution	Label	"Hourly"	Continue	Continue
4	Service 1.0	Task Manager	Hourly	[Click]	Click Hourly RadioButton	Continue	Continue

This process checks to see if the value of the variable "Task Type" is "Daily" (Step 1). If it is, the execution proceeds to the next step and clicks the Daily RadioButton (Step 2), and then exits the process. If "Task Type" is *not* equal to "Daily", then the execution jumps to the section of the process identified by the label "Hourly" (Step 3), and then clicks the Hourly RadioButton (Step 4).

To see how this is done, here is a screenshot of the **On True / On False** tab for Step 1:

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Text	Compare	Verify T[Task Type] Is Equal To 'Daily'	Continue	Jump

On True

Log Status As: Passed
Action: Continue

On False

Log Status As: Skipped
Action: Jump

Target Step: Hourly

The **On True** section keeps the Certify defaults, which tell the process that if the desired condition (Task Type = Daily) is true, pass the step and then continue on to the next step. The **On False** section tells the process that if the desired condition is not true, then skip (don't pass or fail) the step and jump to the label Hourly.

Step 2 also uses the **On True / On False** tab. If Task Type = Daily and then the Daily RadioButton is clicked, the process should stop. Or, when it gets to Step 4, it will click the Hourly RadioButton. Once the Daily RadioButton is successfully clicked, the process ends.

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
2	Service 1.0	Task Manager	Daily	[Click]	Click Daily RadioButton	Exit Process	Continue
3	System 1.0	System	Execution Label		"Hourly"	Continue	Continue

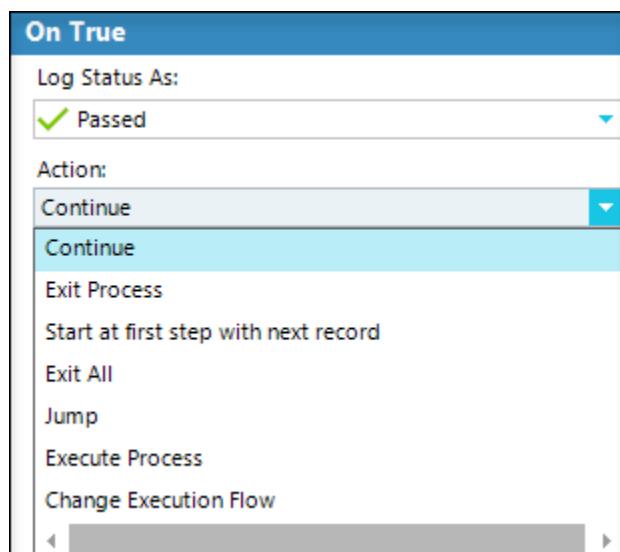
On True

Log Status As: Passed
Action: Exit Process

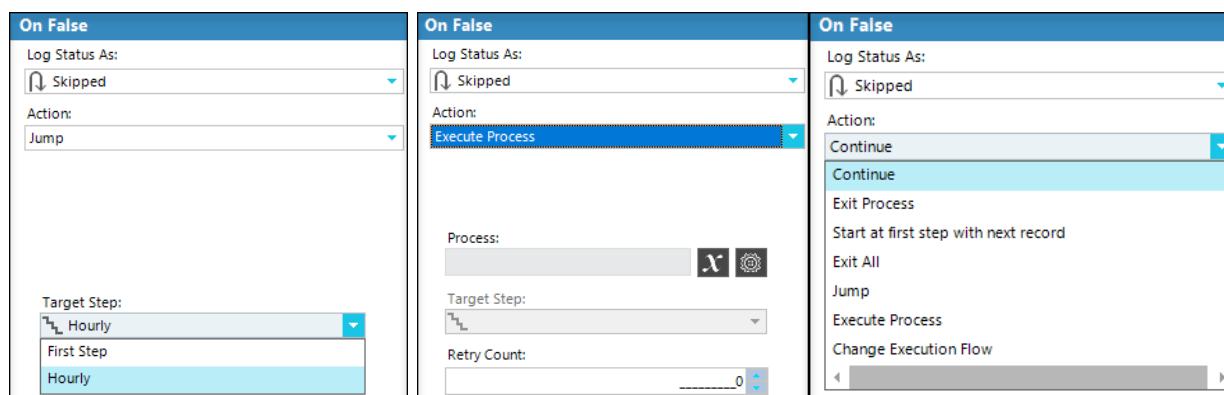
On False

Log Status As: Failed
Action: Continue

It's important to know all the **Actions** available in the **On True / On False** tab to make the best use of this feature.



Some of the **Actions** have their own options providing even more control of your processes:



There are two other automation methods featured in our validation processes:

- o **Initialize** – set/reset a variable value to *blank* (text) or "0" (number). *String = ""*
- o **Increment** – increase (or decrease) numeric values by one or more. *RowNum = RowNum + 1*
- o **Comments** – statements within a process that explain the programming but are not part of the actual functional steps. *"The following three steps create a random text string containing the current account number."*

Below is a process that utilizes **While** and **If, Then Else** conditional logic as well as **Initialize**, **Increment**, and **Comments**.

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Execution	Comment	"There is a delay until job execution status message is "Process Complete". So we will wait 10 seconds."	Continue	Continue
2	System 1.0	System	Variable	Set	Set N[Counter] = "0"	Continue	Continue
3	System 1.0	System	Execution	Label	"Press Refresh"	Continue	Continue
4	Personnel Management 1.0	(RPUBEN6 2:2000) Personnel Management	(GuiButton)	Click	Click "Single" Button (GuiButton)	Continue	Continue
5	SAP Core 1.0	SAP Main	sbar	Verify Property	Verify sbar StatusBar "Text" Contains "Processing complete"	Exit Process	Continue
6	System 1.0	System	Execution	Wait	Wait "10" seconds	Continue	Continue
7	System 1.0	System	Number	Math	Modify "" = N[Counter] "Add" N[Counter]	Continue	Continue
8	System 1.0	System	Number	Compare	Verify N[Counter] Is Less Than Or Equal To "5"	Continue	Continue
9	System 1.0	System	Execution	Label	"Processing Complete"	Continue	Continue

This process runs up to five times to determine if an SAP process has finished. The process:

- Has a **Comment** explaining the purpose of the process (Step 1);
- Initializes** the numeric variable "Counter" (Step 2);
- Has two labels (Steps 3, 9);
- Clicks a Refresh button (Step 4) and then checks for the message "Processing Complete". If it finds the message, execution jumps to the label "Processing Complete - Exit Process". If it doesn't find the message, execution continues (Step 5);
- Waits (stops processing) for 10 seconds (Step 6);
- Increments** the variable "Counter" by 1 (Step 7);
- Checks to see if "Counter" is < or = 5 (**While** "Counter" is < or = 5). If it is, the execution jumps to the label "Press Refresh"; if the counter is greater than 5, then the execution exits the process (Step 8).

Note: There's no need to worry about whether a condition is **While**, **If, Then Else**, or something else. The important thing to understand is how to create the conditional logic in Certify.

EXERCISE 7.1 — Create the WSA_EditQuantityandPrice Process

This exercise will show how to create a process to edit Purchase Orders in the Worksoft Web Sample Application.

Note: Depending on how Certify learned your version of the Web Sample Application's objects, your object names may vary from the objects shown in the following exercises.

Step	Action
1.	Create a new folder titled WSA_EditandDelete under your Sandbox folder.
2.	Create a new process using the following information:

Process Name	Description
WSA_EditQuantityandPrice	Selects a purchase order and edits a specific material's quantity and price.

Step	Application Version	Window	Object	Action
1	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Find Row (Advanced)
Store Found Row In		(V) _Row		
Column Caption 1 (overrides Column Number)		Name		
Match Attribute 1		innerText		
Match Value 1		(V) Name		
Match Criteria 1		Is Equal To		

Step	Application Version	Window	Object	Action
2	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Select Cell
Row Number		(V) _Row		
Column Caption 1 (overrides Column Number)		Action		
Click Type		Single		
Horizontal % (default = 50)		30		
Vertical % (default = 50)		50		

Step	Application Version	Window	Object	Action	
3	Worksoft Web Sample Application 1.0	WebAccountManager	TABLE_ELEMENT	Find Row (Advanced)	
Store Found Row In		(V) _Row			
Column Caption 1 (overrides Column Number)		Material			
Cell Sub-Object TagName 1		INPUT			
Match Attribute 1		Value			
Match Value 1		(V) Material			
Match Criteria 1		Is Equal To			

Step	Application Version	Window	Object	Action	
4	Worksoft Web Sample Application 1.0	WebAccountManager	TABLE_ELEMENT	Input Into Cell	
Store Found Row In		(V) _Row			
Column Caption 1 (overrides Column Number)		Quantity			
Input Type		Input Text			
Value		(V) Quantity_Numeric			
Value Criteria (only for Select From DropDownList)		Is Equal To			

Step	Application Version	Window	Object	Action	
5	Worksoft Web Sample Application 1.0	WebAccountManager	TABLE_ELEMENT	Input Into Cell	
Store Found Row In		(V) _Row			
Column Caption 1 (overrides Column Number)		Price			
Input Type		Input Text			
Value		(V) Price_Numeric			
Value Criteria (only for Select From DropDownList)		Is Equal To			

Step	Application Version	Window	Object	Action	
6	Worksoft Web Sample Application 1.0	WebAccountManager	Save	Press	
Horizontal % (default = 50)		50			
Vertical % (default = 50)		50			

Step	Application Version	Window	Object	Action
7	Worksoft Web Sample Application 1.0	WebAccountManager	status_1	Verify
Value	The Item is edited			
Criteria	Is Equal To			

Step	Application Version	Window	Object	Action
8	System	System	Operating System	Capture Screen Image
Capture Screen Type	Active Window			

3. Create the **WSA_EditQuantityandPrice** Layout and Recordset by selecting **Steps 1, 3, 4 and 5**, right-clicking, and choosing **Add to Layout**. This should give you the ability to add the **Name**, **Material**, **Quantity_Numeric**, and **Price_Numeric** variables to the layout.
4. **Save** the process.
5. Navigate to the **WSA_EditQuantityandPrice** Recordset, and add the following values:

Name	Material	Quantity_Numeric	Price_Numeric
1 John Smith	Phone	1	400
2 Mary Wilson	Watch	2	199
*			

6. Execute the process and troubleshoot any failures.

EXERCISE 7.2 — Create the UTL_WSA_ExportRecordset Process

In this exercise, we will create a utility that exports the WSA_Input recordset. We will use this data in the next process.

Step	Action
------	--------

1. In Processes, navigate to your **WSA_EditandDelete** folder.
2. Create a new process using the following information:

Process Name	Description
UTL_WSA_ExportRecordset	Exports the WSA_Input recordset to a .txt file.

Step	Application Version	Window	Object	Action
1	System	System	Record Set	Export RecordSet
Layout		WSA_Input		
RecordSet		WSA_Input		
File		Name: WSA_Input_RecordsetExport.txt Save: In location of your choosing		
Mode		Overwrite		
ColumnDelimiter		<COMMA>		
IncludeHeaders		Not Checked		
TextQualifier		<NONE>		

EXERCISE 7.3 — Create the WSA_UpdateRecordset Layout

In this exercise, we will create the **WSA_UpdateRecordset Layout**. We will import the file from the previous exercise into this layout and recordset.

Step	Action
------	--------

1. In the Navigation Taskbar, select **Data**, and navigate to your **WSA_EditandDelete** folder.
2. Create a new layout with the following information:

Layout Name	Variables
WSA_UpdateRecordset	PO Number Name Ship to Bill to Doc Number Amount (Numeric)

3. Create a recordset for the new layout with the same name, **WSA_UpdateRecordset**.
4. Create a second recordset for the new layout with the following name, **WSA_UpdateRecordset_Export**. ***Note: This recordset will be used in a later exercise.**

EXERCISE 7.4 — Create the WSA_UpdateRecordset Process

This exercise will show how to create a process to update an imported recordset. The Web Sample Application does not archive created Purchase Orders and new Purchase Orders will continually add to the PO table. Before deleting old POs, we need to export the data for recordkeeping.

Step	Action
------	--------

1. In Processes, navigate to your **WSA_EditandDelete** folder.
2. Create a new process using the following information:

Process Name	Description
WSA_UpdateRecordset	Adds each purchase order's amount to each record imported to the WSA_UpdateRecordset process and exports the data to a new .txt file.

The first step will be a system step that executes the UTL_WSA_ExportRecordset process.

Step	Application Version	Window	Object	Action
1	System	System	Execution	Execute Process
Process	UTL_WSA_ExportRecordset			

The second step will import the .txt file created in the previous step.

Step	Application Version	Window	Object	Action
2	System	System	Record Set	Import RecordSet
Layout	WSA_UpdateRecordset			
RecordSet	WSA_UpdateRecordset			
File	WSA_Import_RecordsetExport.txt			
Mode	Overwrite			
ColumnDelimiter	<COMMA>			
IncludeHeaders	Not Checked			
TextQualifier	<NONE>			

Step	Application Version	Window	Object	Action
3	System	System	Record Set	Read Record
Read Record from	WSA_UpdateRecordset			
Record Set	WSA_UpdateRecordset			
Mode	First			
Index	0			

Step	Application Version	Window	Object	Action
4	System	System	Execution	Label
Label Name	Doc Number			

Step	Application Version	Window	Object	Action
5	System	System	Record Set	Read Record
Read Record from	WSA_UpdateRecordset			
Record Set	WSA_UpdateRecordset			
Mode	Next			
Index	0			

Step	Application Version	Window	Object	Action
6	System	System	Execution	Label
Label Name	Table			

Step	Application Version	Window	Object	Action
7	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Find Row (Advanced)
Store Found Row In	(V)_Row			
Column Caption 1 (overrides Column Number)	DocNumber			
Cell Sub-Object TagName 1	N/A			
Match Attribute 1	innerText			
Match Value 1	(V) Doc Number			
Match Criteria 1	Is Equal To			

Step	Application Version	Window	Object	Action	
8	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Store Cell	
Store Cell Value In		(V) Amount			
Row Number		(V) _Row			
Column Caption (overrides Column Number)		Amount			
Store Type		Cell Text			

Step	Application Version	Window	Object	Action	
9	System	System	Record Set	Write Record	
Layout		WSA_UpdateRecordset			
RecordSet		WSA_UpdateRecordset_Export			
Mode		Append			

Step	Application Version	Window	Object	Action	
10	System	System	Execution	Label	
Label Name		Export			

Step	Application Version	Window	Object	Action	
11	System	System	Record Set	Clear RecordSet	
Layout		WSA_UpdateRecordset			
Record Set		WSA_UpdateRecordset			

Step	Application Version	Window	Object	Action	
12	System	System	Record Set	Clear RecordSet	
Layout		WSA_UpdateRecordset			
Record Set		WSA_UpdateRecordset_Export			

3. Modify the On True/On False for Step 3.

On True		On False	
Log Status As:	Passed	Log Status As:	Skipped
Action	Jump	Action	Jump
Target Step:	Table	Target Step:	Export

4. Modify the On True/On False for Step 5.

On True		On False	
Log Status As:	Passed	Log Status As:	Skipped
Action	Continue	Action	Jump
Target Step:		Target Step:	Export

5. Modify the On True/On False for Step 7.

On True		On False	
Log Status As:	Passed	Log Status As:	Skipped
Action	Continue	Action	Jump
Target Step:		Target Step:	Export

6. Modify the On True/On False for Step 9.

On True		On False	
Log Status As:	Passed	Log Status As:	Failed
Action	Jump	Action	Continue
Target Step:	Doc Number	Target Step:	

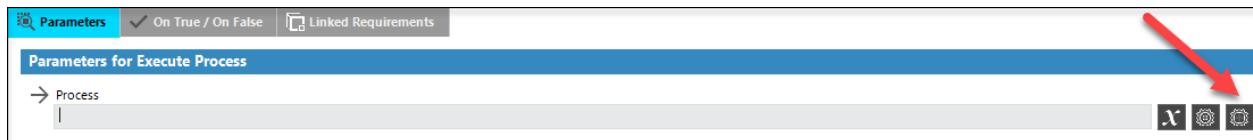
EXERCISE 7.5 — Create the WSA_UpdateRecordset_C_Export Process

This exercise will show how to create a child process to export the updated recordset to a .txt file. We are making it a child process so it only executes once after all the imported Purchase Order records are updated with an amount.

Step	Action
------	--------

- Right-click on Step #10, and select **Insert Step Below**. Modify the step as shown below.

Step	Application Version	Window	Object	Action
11	System	System	Execution	Execute Process
Process	{Select the New Process Gear}			



- Name the new process **WSA_UpdateRecordset_C_Export**, and provide the following **Description**: *This child process exports the updated recordset of the parent process to a .txt file for recordkeeping.*
- Add a **System** step to the child process.

Step	Application Version	Window	Object	Action
1	System	System	Record Set	Export RecordSet
Layout	WSA_UpdateRecordset			
RecordSet	WSA_UpdateRecordset_Export			
File	WSA_UpdateRecordset_Export.txt			
Mode	Append			
ColumnDelimiter	<COMMA>			
IncludeHeaders	Not Checked			
TextQualifier	<NONE>			

- Execute your **WSA_UpdateRecordset** process and troubleshoot any failed steps.
- Open the exported files to verify the data.

EXERCISE 7.6 — Create the WSA_DeletePO_Amount Process

This exercise will show you how to delete a purchase order from the WSA PO table by the Amount criterion. As a result, the PO table will be cleared of purchase orders below \$5000.

Step	Action
------	--------

1. Navigate to your **WSA_EditandDelete** folder.
2. Create a new process using the following information:

Process Name	Description
WSA_DeletePO_Amount	Identifies and deletes any Purchase Order with an amount less than \$5000. This will limit purchase orders in the WSA PO table by the Amount criterion.

3. Create the following steps:

Step	Application Version	Window	Object	Action
1	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Find Row (Advanced)
Store Found Row Number In		(V) _Row		
Matching Row Instance		1		
Column Caption 1 (overrides Column Number)		DocNumber		
Cell Sub-Object TagName 1		N/A		
Match Attribute 1		innerText		
Match Value 1		(V) Doc Number		
Match Criteria 1		Is Equal To		

Step	Application Version	Window	Object	Action	
2	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Store Cell	
Store Cell Value In		(V) POAmount (this must be a Number Data Type)			
Row Number		(V) _Row			
Column Caption 1 (overrides Column Number)		Amount			
Store Type		Cell Text			

Step	Application Version	Window	Object	Action	
3	System 1.0	System	Number	Compare	
Value 1		(V) POAmount			
Criteria		Is Greater Than			
Value 2		5000			

Step	Application Version	Window	Object	Action	
4	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Select Cell	
Row Number		(V) _Row			
Column Caption (overrides Column Number)		Action			
Click Type		Single			
Horizontal % (Default=50)		40			
Vertical % (Default=50)		50			
Follow-up Keystroke		None			

Step	Application Version	Window	Object	Action	
5	New Certify Web Sample App 1.0	WebAccountManager	status_2	Verify	
Value		An Item is deleted			
Criteria		Is Equal To			

Step	Application Version	Window	Object	Action
6	System 1.0	System	Operating System	Capture Screen Image
Capture Screen Type		Desktop		

Step	Application Version	Window	Object	Action
7	System 1.0	System	Execution	Label
Label Name		Amount Comparison		

4. Modify the On True/On False of **Step 1**.

On True		On False	
Log Status As:	Passed	Log Status As:	Skipped
Action	Continue	Action	Start at first step with next record

5. Modify the On True/ On False of **Step 3**.

On True		On False	
Log Status As:	Passed	Log Status As:	Skipped
Action	Jump	Action	Continue
Target Step:	Amount Comparison	Target Step:	

- 6.** Attach the **WSA_Input layout** and **WSA_Input recordset** to the process. Leave the RecordSet Mode to the default Read Only.
- 7.** Click **Save**.
- 8.** Run the process step-by-step to see the decision points, and troubleshoot any failures.

EXERCISE 7.7 — Create the WSA_DeletePO_Cleanup Process

This exercise will show you how to delete all purchase orders from the WSA PO. The static value used in the PO Number concatenation step will be used to identify the purchase order record for deletion.

Step	Action
------	--------

1. Navigate to your **WSA_EditandDelete** folder.
2. Create a new process using the following information:

Process Name	Description
WSA_DeletePO_Cleanup	Identifies and deletes all purchase orders in the PO table.

3. Create the following steps:

Step	Application Version	Window	Object	Action
1	System 1.0	System	Number	Set
To the Value		2		
Variable		(V) _Row		

Step	Application Version	Window	Object	Action
2	System 1.0	System	Execution	Label
Label Name		Loop		

Step	Application Version	Window	Object	Action
3	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Select Row
Row Number		(V) _Row		
Click Type		Single		

Step	Application Version	Window	Object	Action	
4	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Verify Cell	
Row Number		(V) _Row			
Column Caption (overrides Column Number)		PONumber			
Verify Type		Cell Text			
Criteria		Starts With			
Value		{STATIC VALUE used in PO Number concatenate step}			

Step	Application Version	Window	Object	Action	
5	System 1.0	System	Number	Math	
Value1		(V) _Row			
Operation		Add			
Value2		1			
Result		(V) _Row			

Step	Application Version	Window	Object	Action	
6	System 1.0	System	Execution	Jump	
Start At Step		Loop			

Step	Application Version	Window	Object	Action	
7	System 1.0	System	Execution	Label	
Label Name		Delete this one			

Step	Application Version	Window	Object	Action	
8	Worksoft Web Sample Application 1.0	WebAccountManager	tblList	Select Cell	
Row Number		(V) _Row			
Column Caption (overrides Column Number)		Action			
Click Type		Single			
Horizontal % (default = 50)		{May need to modify}			
Vertical % (default = 50)		{May need to modify}			

Step	Application Version	Window	Object	Action
9	System 1.0	System	Execution	Jump
Start At Step		Loop		

4. Modify the On True/ On False of **Step 3**. Modifying the True/False of Step 3 exits the process execution when there are no more purchase orders in the PO table for deletion.

On True		On False	
Log Status As:	Passed	Log Status As:	Skipped
Action	Continue	Action	Exit Process
Target Step:		Target Step:	

5. Modify the On True/ On False of **Step 4**. Modifying the True/False of Step 4 skips the step and jumps to the label “Delete this one” to delete the purchase order.

On True		On False	
Log Status As:	Skipped	Log Status As:	Passed
Action	Jump	Action	Continue
Target Step:	Delete this one	Target Step:	

6. Modify the On True/ On False of **Step 8**. Modifying the True/False of Step 8 skips the step and jumps to the first step in the process to continue execution.

On True		On False	
Log Status As:	Skipped	Log Status As:	Passed
Action	Jump	Action	Continue
Target Step:	First Step	Target Step:	

7. Click Save.
8. Run the process step-by-step to see the decision points, and troubleshoot any failures.

Lesson Summary

You've completed the [Advanced Processes](#) lesson.

Key points to remember:

Using conditional logic in automated processes allows the creation of sophisticated tests that can handle intensive business processes and broad datasets.

Key conditional logic techniques in Certify include:

- Identifying and comparing objects
- The use of labels and jumps
- On True, On False functionality/options

Lesson 8

Certify System Objects and Actions

Overview

This chapter is a reference of Certify system objects and their associated actions.

Certify System Classes and Actions

In addition to using the application under test (AUT) objects, such as SAP screens and fields, HTML objects and links, etc., Certify comes with a set of built-in non-application specific objects and actions to work with such things as date (date math, date part), Windows system commands, text manipulation (concatenate, compare, etc.), and other helpful test building options.

Reference: For a complete list of the System Classes and Actions, go to Certify Help → Classes and Actions → System Classes and Actions.

Or, on the Navigation taskbar, click **Interfaces**, then select **System**. View this area only. Modifying this list will cause problems in Certify.

Use these objects in your processes by adding a step with:

Application Version	Window	Object	Action
System 1.0	System	Desired object	Desired action

Below is a summary of the Certify system objects and their respective actions.

System Object	Action	Usage	Example
Date	Compare	Compare one date to another	Compare (V)Ship_Date to Todays date
	Get Part	Get Day, Month, or Year from a date	Get Day form Todays date Get Year from (V)Posting Date
	Date math	Add or subtract Days or Weeks from a date	Add 7 Days to today's date Subtract 2 weeks from (V)Expected Delivery date
	Set	Used to Set a variable to specific value	Set (V)Date to (V)Period Date
Execution	Comment	Used to insert comments into Certify process steps. Helps describe certain conditions or objects in the process for maintenance purpose	"Find and press the Post Goods Issue button" "Execute Process 'XXXX' if count > 5"
	Dialog Prompt	Used to prompt the Certify user at runtime to perform some action. The test is paused until the prompt is answered.	"Load check paper into the Payroll printer and press Enter when Ready..."
	Execute Process	Allows a process to call a child process. Once the child process is complete, control returns to parent process. A layout and recordset can be added to this step that may be needed to feed the child process.	OTC_StandardOrder Execute VA01_CreateOrder Layout=VA01 Recordset=Sales Execute VL01N_ShipOrder Execute VF01_InvoiceOrder
	Exit All	Will cause the test to exit regardless of what level of execution	If (V)Amount = 0 then Exit All
	Exit Process	Exit the current process being executed and return to the parent process. If the current process is the parent process, the test will exit.	If (V)Available Qty < (V)Order Qty then Exit Process
	Jump	Change flow to continue at a different step in the process rather than the next logical step. Note: works with the object Label (see below) which is the named entry point	If (V)Order Qty = 0 then Jump to Label="Enter Order Qty"
	Label	A marker in the process that step flow might be redirected to	Label="Enter Order Qty"

System Object	Action	Usage	Example
	Manual Step	A prompt that will pause the Certify test being executed and ask the user to supply a true/false status to a condition outside the control of Certify.	"Check Payroll Printer - did all 48 checks print correctly" [True/False]
	Set AUT Locale	Change locale settings for various country specific formats	Examples of valid locales are de-DE, en-GB, en-US, es-MX, fr_FR, it-IT, nl-nl
	Start at first Step with Next Record	Cause Certify to change step flow in a process to the beginning with the next record in the recordset. EOF will cause the process to end and return to the parent process or end the test if it is the parent process.	compare (V)Ship-Date = (V)system Date True=Continue False=Start at first step with next record
	Wait	Delay the process execution by X seconds between steps.	Wait 10 seconds
Execution Setting	Capture Screen Mode	Set option for whether to take screenshots of the active application or whole desktop during process execution.	CaptureActiveWindow
	Capture Screen On	Set option for when to take screenshots during the process execution.	CaptureAbortedAndFailedSteps
	Step Delay	Set optional delay between every step during execution.	Delay .5 seconds
	Stop on Failure	Set an option to stop the execution if a step fails.	Check Stop on Failure
File	Append Data	Add data to the end of a text file located outside of the Certify Database (ex: C:\orders\OrderList.txt)	Append (V)Order Number to C:\orders\OrderList.txt
	Clear File	Clears the content of a text file located outside of the Certify Database (ex: C:\orders\OrderList.txt)	Clear C:\orders\OrderList.txt
	Delete File	Deletes the text file located outside of the Certify Database (ex: C:\orders\OrderList.txt)	Delete C:\orders\OrderList.txt
	File Compare	Compare two files.	Compare One.txt with Two.txt
	File Exists	Verifies a text file located outside of the Certify Database (ex: C:\orders\OrderList.txt) is present	File Exist C:\orders\OrderList.txt (on True on False)
	Masked File Compare	Compares two .csv, .xls, or .xlsx files with masked cells.	Compare One.xls with Two.xls
	Overwrite File	Clears and appends data to the end of a text file located outside of the Certify Database (ex: C:\orders\OrderList.txt)	Overwrite File (V)Order Number C:\orders\OrderList.txt

System Object	Action	Usage	Example
	Read File	Read the contents of a text file located outside of the Certify Database (ex: C:\orders\OrderList.txt) into a variable	Read File (V)Order Number C:\orders\OrderList.txt into (V)Order List
	Store File Property	Store properties of a file into a variable.	Store File "c:\temp\tempfile.txt" Property "CreationTime" into D[Created On]
Number	Compare	compare ValueA against ValueB with various compare criteria	Compare (V)Amount Is Greater Than 0 Compare (V)Amount Is Less Than or Equal To (V)Net Value
	Get Random Number	Generate a Positive integer random number between MIN and MAX range	Get Random Number between 100 and 9999 and store in (V)Random Number
	Math	Perform standard Math operations on numeric values	Add 1 to (V)Counter and store in (V)Counter Multiple (V)Amount by (V)QTY and store in (V)Total
	Round	Round up a value based on the stated decimals value	Round 1.50 to 0 decimals and store in (V)Value [would be stored as 2] Round 1.6899 to 2 decimals and store in (V)Value [would be stored as 1.69]
	Set	Used to Set a variable to specific value	Set (V)Qty to 1
	Set Random Seed	Used to seed the Random Number to be more random. Not normally used.	Set seed value to 2
	Sum	Sum up to 25 numbers	Sum 10 20 30 40
	Truncate	Drop off the decimal value depending on the number of decimals specified	(V)Temp Number=21.87 Truncate (V)Temp Number to 0 Decimal [(V)Temp Number = 21] (V)Temp Number=21.87 Truncate (V)Temp Number to 1 Decimal [(V)Temp Number = 21.8]
	Database Store	Store a table value from an external database using an ODBC connection and standard SQL statements	ODBC Database Store Select VALUEX From TABLEX Where CONDITIONX is true and store in (V)Table Value
ODBC Data	Database Verify	Verify a table value in an external database using a ODBC connection and standard SQL statements	ODBC Database Verify Select VALUEX From TABLEX Where CONDITIONX is true and store in (V)Table Value
	Capture Screen Image	Take a screen shot of the desktop or the topmost window	Capture Current Screen Capture Active Window

System Object	Action	Usage	Example
	Execute Application	Executes an external Windows application by providing the application name and any command line options	Execute application C:\Sap\Sapshcut.exe /User=SA Paccount / Password=SAPpassword
	Get Environmental Variable	Store the value of a system environment variable.	Variable = Path
	DOS Command	Enter in a standard DOS command	TASKKILL /f C:\windows\notepad.exe
Record Set	Clear RecordSet	Used to remove the contains of a named layout and recordset	Clear Recordset Layout=VA01, Recordset=TempRecSet
	Export RecordSet	Will export the contains of a Certify Recordset out to a text file with selected formatting	Export Recordset Layout=VA01, Recordset=TempRecSet to C:\temp\TempRecSet, comma delimited
	Flush RecordSet	Flush will update the database with any changed values in a recordset rather than waiting until the end of the Certify Process run so it can be available immediately	Flush Layout=VA01_SalesOrder, Recordset=ExportOrders
	Import RecordSet	Import data from a file into a specified layout/recordset	Import Jan2013Accounts.csv
	Read Record	Will cause Certify to read a record as a step operation outside the normal process of layout/recordset attachment	Read Record Layout=VA01_SalesOrder, Recordset=ControlData, Index=2
	Refresh RecordSet	Will update the recordset values with any variable changed in a process	Refresh Recordset Layout=VA01_SalesOrder, Recordset=ControlData
	SQL to RecordSet	Imports any number of rows or columns into a layout and recordset from a SQL Server Select Statement.	SQL "SELECT top 100 [Name].[CreatedBy] FROM [Process] where [CreatedBy]<>1" to SQL to RecordsetSample
	Verify RecordSet Existence	If the recordset does exist then subsequent steps (e.g. Execute Process) using that recordset can be executed correctly.	Verify RecordSet Existence RecordsetSample
	Write Record	Writes data into a single record of a layout and recordset. Options are Overwrite and Append	Write recordset Layout=VA01_SalesOrder_Temp, Recordset=OrderNumbers [Overwrite Append]
Text	Compare	Compare TextA against TextB with various criteria and case sensitivity options	Compare (V)TaxCode Is Greater Than G Compare (V)LastName Is Less Than or Equal To (V)Temp with Case Matching
	Concatenate	Join up to 15 different text elements into one text variable	Concatenate (V)Full_Name with (V)First_Name+(V)Middle_Name +(V>Last_Name

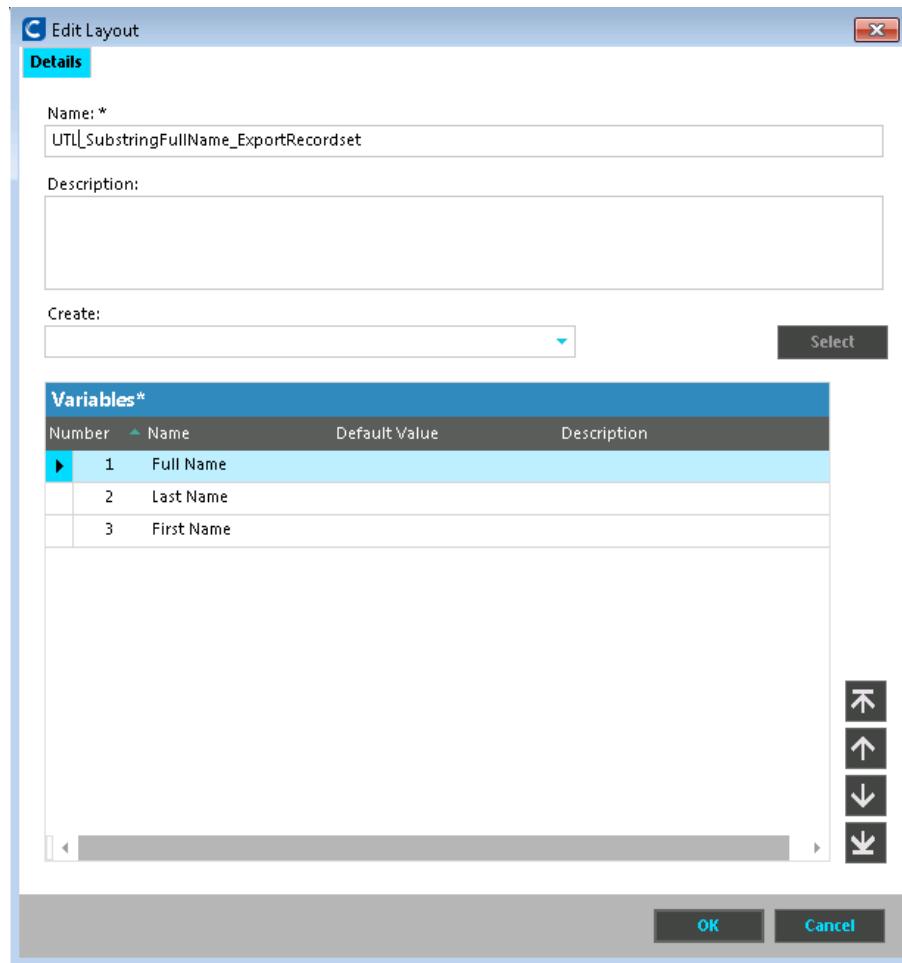
System Object	Action	Usage	Example
	Convert Case	Switch text value between upper or lower case	Convert Case (V)Last Name to Upper Case
	Find Substring	Use to grab a portion of a text string and place in a text variable based on a known starting and ending text value.	1) Set (V)String="Record 1234 created" 2) Find Substring Starts with Record and ends with created and place in (V)Temp 3) (V)Temp=1234 Note: If special characters are part of the search string (ex: ABC*1234) then use the \ to escape the * - ex: ABCD-**)
	Length	Find the number of characters in a string	1) (V)Temp="123ABC#" 2) length (V)Temp store in (V)Temp 3) (V)Temp=7
	Replace	Find and replace text within up to two strings and save to variables	Day = Find and replace 'Tuesday' with 'Wednesday'
	Substring	Use to grab a portion of a text string and place in a text variable based on a starting and ending index number	1) Set (V)String = "ABCD1234#" 2) Substring Starts with Index 5 and ending with Length 4 and place in (V)Temp 3) (V)Temp=1234
	Substring Index	Find the Index of a text string	1) Set (V)String = "ABCD1234#" 2) Substring Index (V)String with Substring "1" place in (V)Temp Number 3) (V)Temp Number=5
	Trim	Trim off leading and/or trailing spaces	1) Set (V)String = " ABCD " 2) Trim Leading (V)String and place in (V)Temp 3) (V)Temp = " ABCD " 1) Set (V)String = "ABCD " 2) TrimTrailing (V)String and place in (V)Temp 3) (V)Temp = " ABCD" 1) Set (V)String = " ABCD" 2) Trim Both (V)String and place in (V)Temp 3) (V)Temp = " ABCD"
	Verify Length	Verify the lengths of up to two strings of text.	Verify String1 = 17
Variable	Convert	Change text to number or number to text. Note: change text to numeric value will only work if the text string is a number	(T)String = "00023"; (T)String = (N)TempNo; (N)TempNo = 23

System Object	Action	Usage	Example
	Set	Assign a value to a variable	(N)Counter = 0 (T)Name = "Toby"

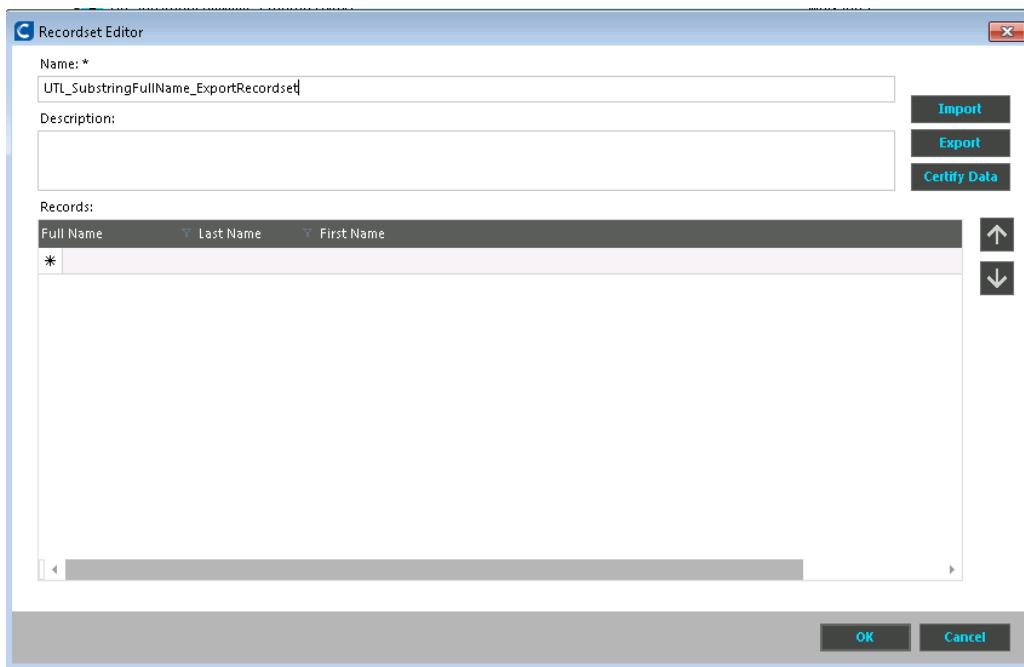
EXERCISE 8.1 — Create System Command Process

In this exercise, you will create a layout and recordset manually. Then a process will be created using several System commands to interact with the recordset.

1. Create a layout manually with the following name and variables:



2. Create an empty recordset named UTL_SubstringFullName_ExportRecordset. No rows of data will be given to the recordset at this point.



3. Create the process as shown below:

Process Name	Description
UTL_SubstringFullName_ExportRecordset	This process substrings first name and last name from the full name and exports the recordset to a .CSV file.

Set "John Smith" equal to variable Full Name.

Step	Application Version	Window	Object	Action
1	System 1.0	System	Variable	Set
Variable	(V) Full Name			
Value	John Smith			

Find the location of the space in text "John Smith". Enter a space for substring parameter.

Step	Application Version	Window	Object	Action
2	System 1.0	System	Text	Substring Index
Result	(V) index (This must be a Number data type format.)			
Input Text	(V) Full Name			
Substring	{SPACE}			

Subtract 1 from (V) index to determine length of first name.

Step	Application Version	Window	Object	Action	
3	System 1.0	System	Number	Math	
Value1		(V) index			
Operation		Subtract			
Value2		1			
Result		(V) index			

Substring "John" from (V) Full Name and assign to (V) First Name.

Step	Application Version	Window	Object	Action	
4	System 1.0	System	Text	Substring	
Result		(V) First Name			
Input Text		(V) Full Name			
Starting Index		1			
Length		(V) index			

Replace empty value for "John" and assign result to (V) Last Name.

Step	Application Version	Window	Object	Action	
5	System 1.0	System	Text	Replace	
Input Text		(V) Full Name			
Find Text		(V) First Name			
Replace					
Result		(V) Last Name			

Delete leading empty spaces.

Step	Application Version	Window	Object	Action	
6	System 1.0	System	Text	Trim	
Value		(V) Last Name			
Trim Type		Trim Left			
Result		(V) Last Name			

Write variable values into UTL_SubstringFullName_ExportRecordset recordset

Step	Application Version	Window	Object	Action	
7	System 1.0	System	Recordset	Write Record	
Layout		UTL_SubstringFullName_ExportRecordset			
Recordset		UTL_SubstringFullName_ExportRecordset			
Mode		Append			

Export recordset into CSV file. File parameter can be any path on your local computer.

Step	Application Version	Window	Object	Action
8	System 1.0	System	Recordset	Export Recordset
Layout		UTL_SubstringFullName_ExportRecordset		
Recordset		UTL_SubstringFullName_ExportRecordset		
File		C:\Users\worksoft3\Desktop\Training3.csv		
Mode		Overwrite		
ColumnDelimiter		<COMMA>		
IncludeHeaders		Checked		
TextQualifier		<NONE>		

4. Navigate to the folder specified in your **file** parameter on Step 8 of your process. Verify .CSV file was exported successfully.

Steps in this exercise should match the following:

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Variable	Set	Set T[Full Name] to "John Smith".	Continue	Continue
2	System 1.0	System	Text	SubstringIndex	Store the position of "" in T[Full Name] into N[index].	Continue	Continue
3	System 1.0	System	Number	Math	Store the result of N[index] "Subtract" "1" into N[index].	Continue	Continue
4	System 1.0	System	Text	Substring	Store the N[index] characters in T[Full Name] beginning at character "1".	Continue	Continue
5	System 1.0	System	Text	Replace	Replace T[First Name] with "" in T[Full Name].	Continue	Continue
6	System 1.0	System	Text	Trim	Remove the whitespace from T[Last Name].	Continue	Continue
7	System 1.0	System	Record Set	Write Record	"Append" the UTL_SubstringFullName_ExportRecordset/UTL_SubstringFullNam	Continue	Continue
8	System 1.0	System	Record Set	Export RecordSet	"Overwrite""C:\Users\worksoft5\Desktop\Training.bat"with the UTL_SubstringFullName_ExportRecordset/UTL_SubstringFullNam	Continue	Continue

Section 2

Lesson 1

Introduction to Worksoft Certify for SAP

Overview

This lesson discusses the basics of using Certify to automate business processes in SAP GUI.

Objectives

After completing this lesson, you will be able to:

- Configure SAP to work with Certify.
- Navigate Certify Capture while recording a business process.

Naming Process Folders

As discussed in the Certify Basics class, an organized, well-structured folder structure is critical to finding and reusing your automated processes. **As a best practice, you should plan on organizing your processes into the following folder structure:**

Integration Tests — Contains folders for the areas of integration tests for SAP, for example: Order to Cash. These folders would contain the root or parent processes (in our example above that might be **OTC_StandardOrderProcess**).

Transactional Tests — Contains folders for transactions, named by the first two characters of the transaction (i.e., VA01 folder is titled VA). This structure provides convenience for finding transactional processes that you may need to use or copy.

Utilities — Smaller single-function processes used to support more complex processes. An example would be **UTL_SAP_Logon** – a utility that does nothing but log into SAP. Includes sub-folders like Date Utilities and General SAP Utilities that contain appropriate utility processes.

Sandbox — Contains folders for each user. **The best practice is to use sub-folders for projects.** For example, for our training processes, the Sandbox folder would look similar to the following:

```
Sandbox folder
  JSmith (user name folder)
    OTC_StandardOrderProcess folder
      VA01_CreateStandardOrder
      VA01_CreateStandardOrder_C_EnterMaterials
      OTC_StandardOrderProcess
```

Once the processes are reviewed and signed off, they should be moved to their respective location. For example:

```
Integration Tests
  Order to Cash (OTC)
    OTC_StandardOrderProcess
Transactional Tests
  VA
    VA01_CreateStandardOrder_C_EnterMaterials
    VA01_CreateStandardOrder
Utilities
  General SAP Utilities
    Utilities_Auto-AnswerInformationalModals
Sandbox
  JSmith
    OTC_StandardOrderProcess
```

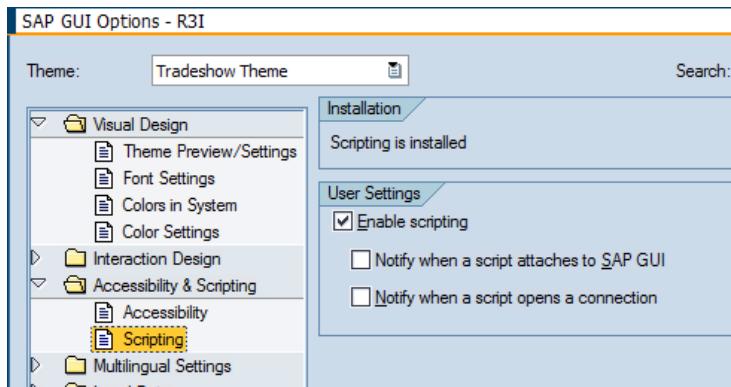
Note: The bolded text indicates folders.

Exercise 1A: Configuring Your SAP Environment

Before you can build your processes in Certify, your SAP environment must be configured to work with Worksoft Certify.

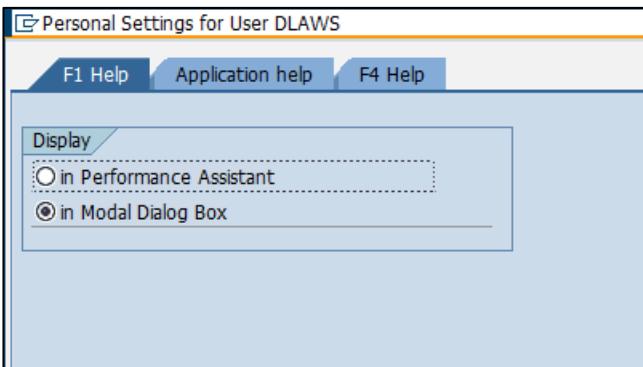
Step	Action
------	--------

1. Log on to SAP.
2. From the SAP GUI interface, click the **Customize Local Layout**  button on the toolbar.
3. The Settings menu appears.
4. Click Options...
5. The SAP GUI Options dialog box appears.



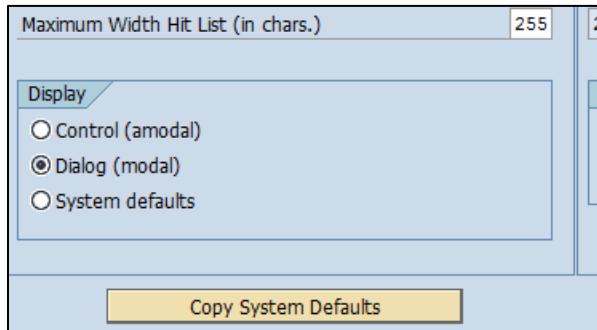
6. In the Theme drop-down, select the **Classic** theme or the **Enjoy** theme while using Certify Capture.
7. In Accessibility & Scripting > Scripting, verify that **Enable scripting** field is checked.
8. Verify the two sub-settings, Notify when a script attaches to SAP GUI and Notify when a script opens a connection, are not checked.
9. Click **OK** to close the **Options** dialog box.

10. From the SAP GUI toolbar, click **Help -> Settings**.



11. On the **F1 Help** tab, select the **in Modal Dialog Box** option.

12. Click the **F4 Help** tab.



13. In the **Display** section, select **Dialog (modal)**.

14. Click the **Save** button to close the dialog box.

Reference: For additional information on configuring your SAP environment, refer to the *Worksoft Certify Installation Guide*.

Certify Capture Overview

The **Certify Capture** feature records your actions while you work through your SAP business process and automatically converts your activity into an automated process. Once Certify Capture is started, all interactions with SAP GUI are captured.

Beginning with Certify 9.0.3 the Capture interface within Certify has changed to better align with the Process Capture standalone application. Capture can be used with SAP, HTML, and Mobile applications.

Certify processes can contain steps created using any combination of methods including Certify Capture and LiveTouch. Once created, there are no differences between these steps.

When do I use Certify Capture versus LiveTouch? What are the differences?

- LiveTouch is a tool where you select objects and those objects are used to create Certify steps. If you need to enter data or move to another page you need to Pause LiveTouch before doing those actions. Any mouse clicks are considered selections.
- Certify Capture is a tool where you interact with an AUT such as SAP GUI in a normal way, with Capture working quietly in the background. You only need to pause Capture if you don't want your actions to be included in your process.
- While you are using Certify Capture your keystrokes and actions are being interpreted by SAP. LiveTouch identifies the objects you are selecting but does not actually interact with them. For example, if you select a Save button in LiveTouch the button is not pressed by LiveTouch and SAP does not attempt to Save the document. Whereas in Certify Capture if you click the Save button then SAP will attempt to Save the document as well as capture the object.
- Certify Capture can better capture steps that involve drop-down menus, trees and other complex objects. LiveTouch may create a fairly generic step identifying these objects but generally does not have the level of detail that Certify Capture creates.
- LiveTouch may be faster and more intuitive when creating verification steps or a single step.
- If you want to work through a scenario for several screens then Certify Capture is a good option.
- If you want to insert a single step then LiveTouch may be a better option.

Using Certify Capture

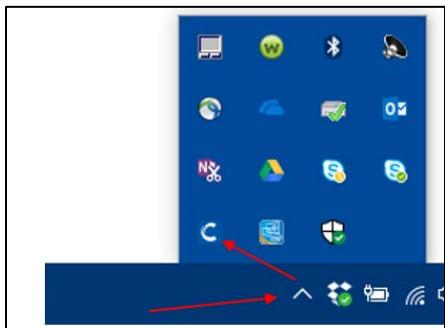
Certify Capture can be used in two ways:

- **To Capture an entire Process** – This method allows you to create a new process using your captured steps.
In Certify, right-click in the Processes Summary Pane and select **New Process Using Certify Capture**.
- **To Insert Captured steps into an existing Process** – As with Certify LiveTouch, this method allows you to insert a few captured steps into a process.
In the Steps tab of the Process Editor, right-click on the step before your desired updates and select **Insert Steps Using Certify Capture**.

Important: SAP GUI must be open and logged in before creating or updating a process using Certify Capture. *For best results, make sure no applications other than SAP GUI and Certify are running.*

General steps for using Certify Capture

- Make sure SAP GUI is open, logged in, and visible on the screen.
- Start Certify Capture using **New Process Using Certify Capture** or **Insert Steps Using Certify Capture**
Certify minimizes, revealing the SAP GUI screen. When the Certify Capture icon appears in the taskbar with the message Capture Running, you are ready to begin the Capture process. It may take a minute before the icon appears.
- Work through your SAP business process using the various Certify Capture options as needed.
- After you complete your session, right-click on the Certify Capture icon in the taskbar and select End Capture. You can also press <Ctrl> <Alt> <End> to stop the Capture process.



Note: Your captured process *may* execute successfully but it is likely that changes will be required before it will pass. Do not be discouraged!

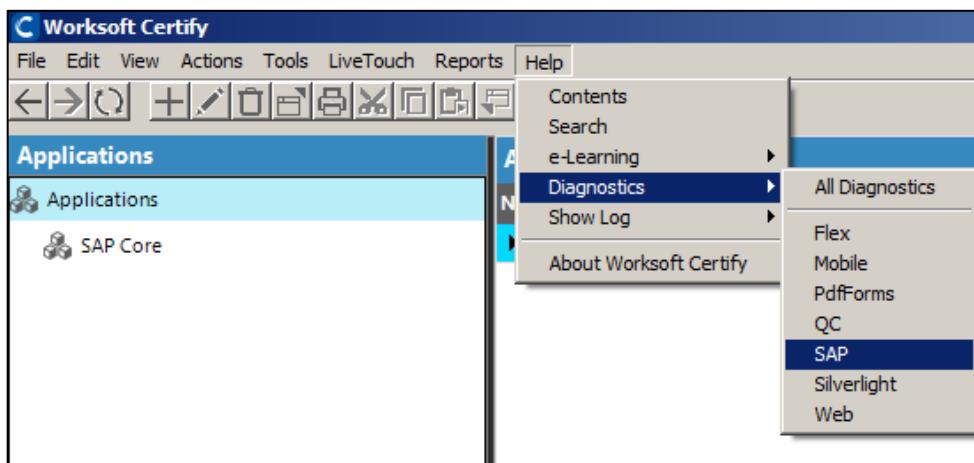
Best Practices for using Certify Capture

- Capture only creates steps for actions you perform in your test. For example, fields that are modified, check boxes that are checked. If there are 25 input fields on the screen only the few that you actually enter data into will be used to create steps. Any pre-filled values must be modified in some way so that Capture will create a step. For example, change one letter in a field and change it back to the original value.
- Be organized and plan for your capture session. You do not want to end up with a great number of extra steps that will need to be removed later.
- Close all extra SAP GUI sessions. Extra sessions will cause extra steps to be added to your process or may cause Capture to work improperly.
- Automation in general is faster using input data rather than a control. For example, rather than selecting F4 Search and selecting an item from the list, simply type the value in the input field.
- In some cases you can either press a button or use keystrokes to perform an action such as Save or Enter. Either will create a perfectly valid step, however pressing a button may result in a clearer step than using keystrokes.
- Pressing the Save Button will create a step with Save as the object and Press as the action.
- Pressing Control+S will create a step with the window as the object and Send VKey as the action.
- To make your automated process easy to read/understand, use Comments liberally during the Capture session.
- During manual testing you may have been using the Back button to return to the main SAP window. In automated testing, we use **/n** before the SAP transaction (e.g. /nVA01) to force a new transaction.

EXERCISE 1.1 — Configuring Certify Capture and Creating a Folder for Training

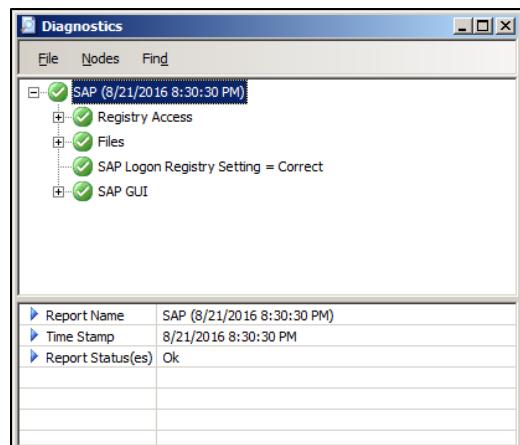
In this exercise, you will configure Certify Capture and create a process. The Certify Diagnostic tool will verify that Certify can connect to SAP GUI session before you begin using Certify Capture.

Step	Action
1.	Log into Certify.
2.	In the Certify Help menu, click Help > Diagnostics .
3.	Select SAP.

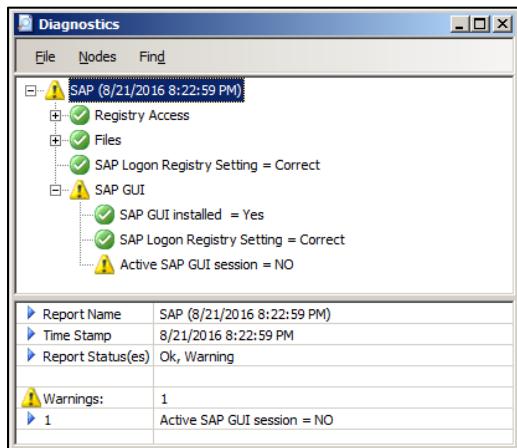


4. The Diagnostics window will display.

*If Certify **can** find the SAP GUI session correctly the screen will display with all green checks.*



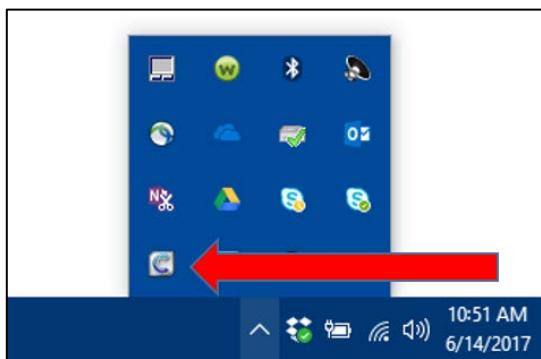
If Certify **cannot** find the SAP GUI session the screen will display with a Yellow warning or a Red Error. You cannot start using Capture until this Diagnostics screen is all green checks.



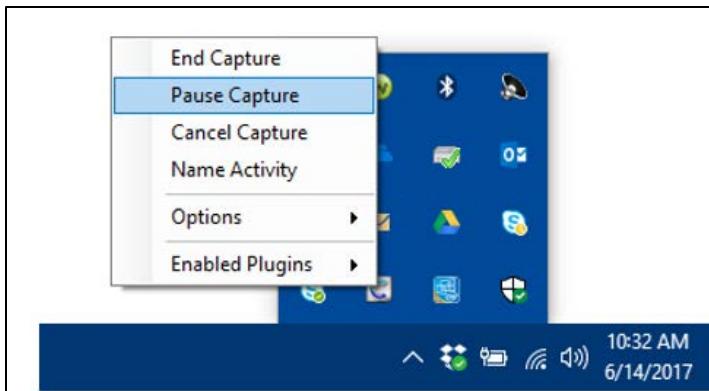
Troubleshooting tips:

- Check to see if there is **one** active SAP GUI session. This should be a QA or Test environment not Production.
- Check that the SAP GUI settings were changed as described in the Configuring Your SAP Environment section.
- Check with your SAP Basis administrator to make sure the SAP system has scripting enabled as described in the Certify Installation Guide.
- Check that your SAP security settings allow you to do scripting.

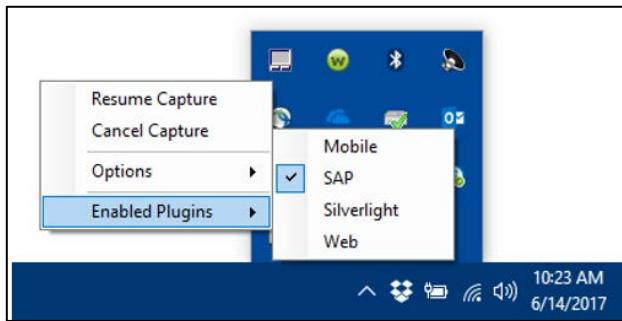
5. Close the Diagnostics window.
6. On the Navigation taskbar, click **Processes**.
7. In the Navigation Tree, navigate to your Sandbox folder.
8. Right-click on your folder and select **New Folder**.
9. In the **Name** field, type **SAP Training**.
10. Click **OK**.
11. In the Navigation Tree, click your **SAP Training** folder.
12. Right-click in the Summary Pane and select New Process Using Certify Capture.
Certify minimizes, revealing the SAP GUI screen. When the Certify Capture icon appears in the taskbar with the message Capture Running, you are ready to begin the Capture process. It may take a minute before the icon appears.
13. Click in the taskbar to see the Worksoft Process Capture icon.



14. Right click on the Worksoft Process Capture icon and select **Pause Capture**. This will prevent Capture from capturing actions while we are configuring Capture.

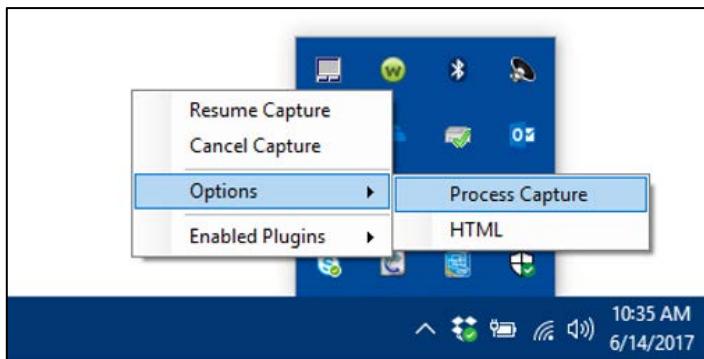


15. Right click on the Worksoft Process Capture icon and select **Enabled Plugins**. It is best to disable any plugins that are not being used in the current capture. In this exercise we will leave SAP and Web enabled.

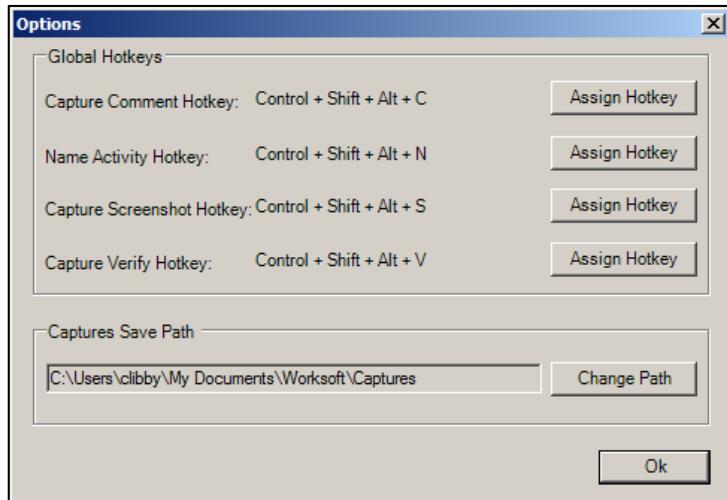


Important: The lessons in this class are designed to work when **SAP** and **Web** are the only **Enabled Plugins**.

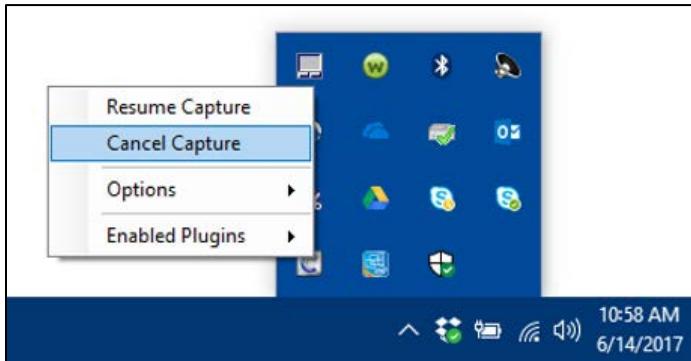
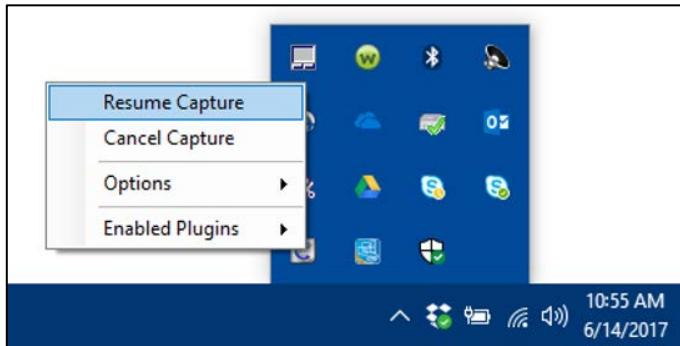
16. Right click on the Worksoft Process Capture icon and select **Options -> Process Capture**.



The options screen displays.



17. The Hotkeys shown will be used to interact with Process Capture. You can change the keys if you prefer a different combination or if the combination conflicts with another application.
- **Capture Comment Step** – Allows you to insert a Certify comment step as you are capturing your process. After you press the hotkey combination, a dialog box appears where you can enter a comment and then click **OK** to add the comment step into your process.
 - **Name Activity Step** – Allows you to provide a name to designate a business activity. This is useful while capturing applications such as web pages where it isn't easy to differentiate activities. While capturing SAP GUI the Name Activity steps are not normally used.
 - **Capture Screenshot Step** – Creates a screen capture step in your process. Note this is not an actual screen capture at the moment of capture but is inserting a step that will take a screen capture during execution. You can select either the **Active Window** or the entire **Desktop**.
 - **Capture Verify Step(s)** – Initiates a LiveTouch-like session allowing you to select specific objects for validation, such as the SAP status bar. A verify step will be inserted in the resulting process for each object selected.
 - **Captures Save Path** – This indicates where captured processes will be stored as files if Standalone Process capture is used rather than Capture Process within Certify.
18. Press **OK** to close the Options dialog.
19. Right click on the Worksoft Process Capture icon and select **Resume Capture**. This will allow Capture to begin capturing actions. If you decide not to capture your process you can use **Cancel Capture** at any time.



Lesson Summary

You've completed the [Introduction to Worksoft Certify for SAP](#) lesson.

Key points to remember:

- You must configure SAP to work with Certify.
- Certify Capture can be used to record SAP business processes.
- Following best practices for capture sessions will ease the capture process and eliminate redundant steps.

Lesson 2

Defining and Developing Processes for Order to Cash

Overview

In this lesson, you will build processes for an Order to Cash (OTC) business scenario.

Objectives

After completing this lesson, you will be able to:

- Use Best Practices to enhance your SAP testing.
- Use Certify Capture to capture a SAP GUI process in Certify.
- Understand when and how to use child processes.

Note: This lesson creates a basic Order to Cash process as an example of an automated SAP business process test. The specific process steps and data required to work at your training location may not exactly match the training guide.

Defining the SAP OTC Processes

For the SAP Order to Cash process, you will create an integrated process, **OTC_StandardOrder**, and the sub-processes listed below in Table 1.

Table 1 — OTC_StandardOrder Processes

Critical Business Process	Sub-Processes	Description
OTC_StandardOrder	VA01_CreateStandardOrder	Creates a basic standard order
	VA01_CreateStandardOrder_C_EnterMaterials	Enters material for the order
	VL01N_CreateOutboundDelivery	Creates an outbound delivery
	VL02N_PostGoodsIssue	Posts Goods Issue
	VF01_CreateBillingDocument	Creates a billing document

Data Needed for the OTC_StandardOrder Process

The table below shows an example of data that may be used for the OTC_StandardOrder process. You may need more or less data depending on your SAP configuration.

Variable	Sample Value	Data Type
Sales Organization	3000	Text
Sales Order Type	OR	Text
Distribution Channel	10	Text
Division	00	Text
Sales Office	3010	Text
Sold-to party	4055	Text
Ship-to party	4055	Text
PO Number	<i>Unique Text</i>	Text
PO Date	<i>System Date</i>	Date
Desired delivery date	<i>System Date + 14 Days</i>	Date
Material	1470	Text
Quantity	2	Number
Shipping Point	3000	Text
Selection Date	<i>System Date + 14 Days</i>	Date
Order Number	<i>Created by SAP</i>	Text
Outbound Delivery Number	<i>Created by SAP</i>	Text
Billing Document	<i>Created by SAP</i>	Text

Note: The data provided here is pulled from the Worksoft Training database. Exact values for your own SAP system will be needed if not working in a Worksoft Training database.

Creating the Order to Cash Process Folder

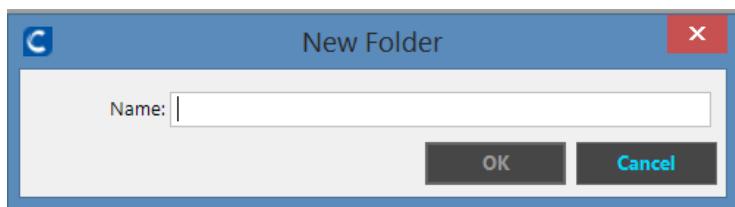
In Certify, creating and using folders is a way to organize project components. Before you create the processes for this class, create a folder to store your work.

EXERCISE 2.1 — Creating a Folder for the SAP Order to Cash Processes

In this exercise, you will create a folder to hold your SAP Order to Cash processes.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, navigate to your Sandbox folder.
3.	Right-click on your SAP Training folder (created in previous lesson), and select New Folder .

The New Folder dialog box appears.



4. In the **Name** field, type **OTC_StandardOrder**.
5. Click **OK**.

The new folder appears under your name in the Navigation Tree.

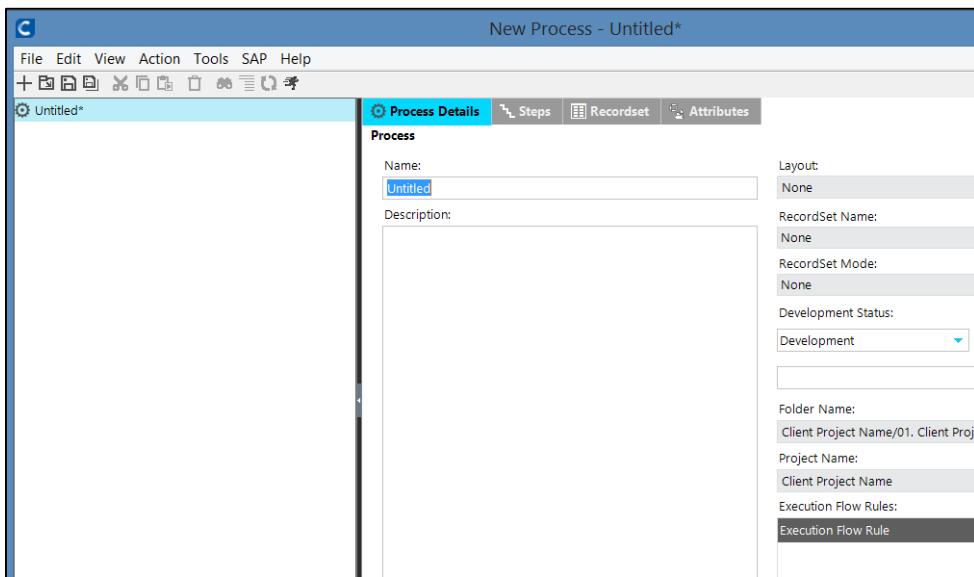
EXERCISE 2.2 — Creating the OTC_StandardOrder Integrated Process

In this exercise, you will create the integrated process, OTC_StandardOrder.

Step	Action
------	--------

1. In the Navigation Tree, click your **OTC_StandardOrder** folder.
2. Right-click in the **Summary Pane** and select **New Process**.

The Process Editor appears.



3. Give the process a name and description:
 - a. Name = **OTC_StandardOrder**.
 - b. Description = **This process creates and processes a standard sales order.**
4. Save the process.

Important: It is important to save often while creating processes. You can save by: 1) clicking the **Save** icon, 2) Selecting **Save** from the **File** menu, or 3) Pressing **Ctrl+S**.

5. Close the Process Editor.

We will return to this process when the sub-processes have been completed.

Creating Processes Using Certify Capture

The Certify Basics course focused on using LiveTouch to help create processes and Lesson 1 of this course covered how to record SAP GUI business processes with Certify Capture. The exercises in this guide will utilize both Certify Capture and LiveTouch to create processes.

EXERCISE 2.3 — Use Certify Capture to Create a Process in SAP GUI

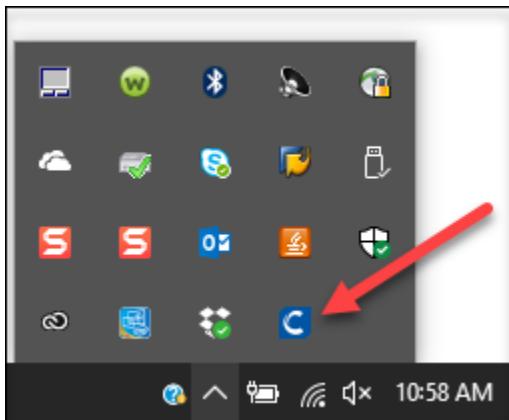
Sub-Processes	Description
VA01_CreateStandardOrder	Creates a basic standard order
VA01_CreateStandardOrder_C_EnterMaterials	Enters material for the order
VL01N_CreateOutboundDelivery	Creates an outbound delivery
VL02N_PostGoodsIssue	Posts Goods Issue
VF01_CreateBillingDocument	Creates a billing document

In this exercise, you will use the Certify Capture feature to capture the initial steps of transaction VA01 to create the VA01_CreateStandardOrder process.

Step	Action
1.	Verify that you have an SAP session open and your working folder is open in Certify.
2.	Right-click in the Summary Pane and select New Process Using Certify Capture .

Certify minimizes, revealing the SAP GUI screen. When the Certify Capture icon appears in the taskbar with the message Capture Running, you are ready to begin the Capture process. It may take a minute before the icon appears.

3. Click in the taskbar to see the Worksoft Process Capture icon.



4. In SAP, in the Command Input box, type /nVA01 and then click the Enter button.

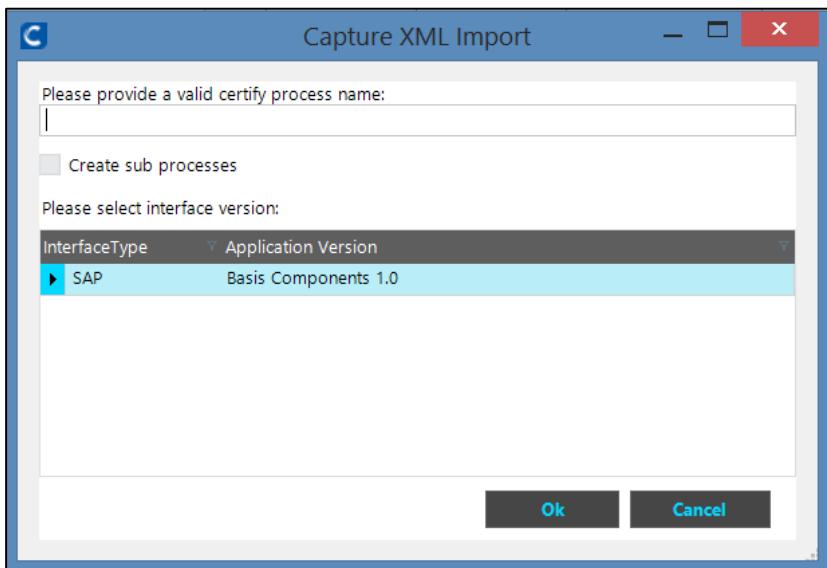
The Create Sales Order: Initial Screen window appears.

5. Enter the values in the following fields (if used in your implementation):
- Order Type
 - Sales Organization
 - Distribution Channel
 - Division
 - Sales Office
 - Any other required fields

A screenshot of the SAP 'Create Sales Order: Initial Screen' dialog. The title bar says 'Create Sales Order: Initial Screen'. Below it is a toolbar with various icons. The main area has tabs for 'Create with Reference', 'Sales', 'Item overview', and 'Ordering party'. On the left, there's a section for 'Organizational Data' with fields for 'Sales Organization', 'Distribution Channel', 'Division', 'Sales Office', and 'Sales Group', all of which have their respective input boxes highlighted with red boxes. On the right side of the screen, there is a large, mostly empty text area.

6. Press the **Enter** key.

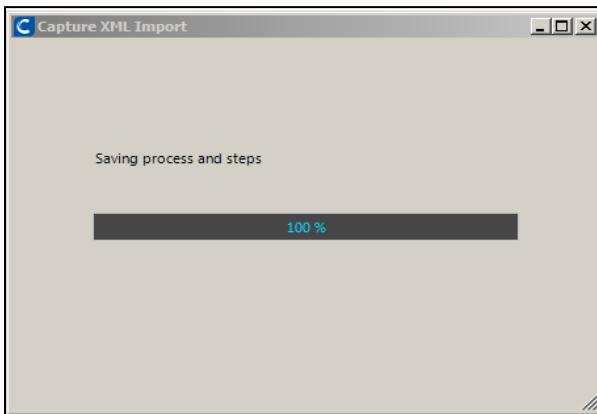
7. Right click on the Worksoft Process Capture icon and select **End Capture**.



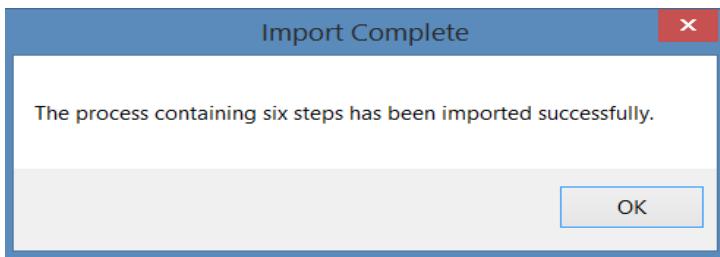
The Capture XML Import screen displays.

8. In the text field, type **VA01_CreateStandardOrder**.
9. Click **OK**.

A dialog box will display for several seconds or minutes as the steps you created are evaluated and a process is created.



The SAP session closes and the Certify Capture dialog box appears stating that the process was successfully created.



10. Click **OK**. *The new process is added to your folder in Certify containing the steps that were performed.*
11. In the Processes pane, double-click the **VA01_CreateStandardOrder** process. *The Process Editor window appears.*
12. Type in the **Description: This process creates a standard sales order.**
13. Click the **Steps** tab.

The steps you performed in SAP are shown. From here, you can edit and/or delete any of the steps.

Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	Basis Components 1.0	SAPMain	okcd	[Input]	Input the value "/NVA01" into the okcd	Continue	Continue
2	Basis Components 1.0	SAPMV45 (VBAK-A0101 AUART) O		[Input]	Input "OR" into instance "1" of the (VBAK-AUART) Order Type CTextField and send Key "	Continue	Continue
3	Basis Components 1.0	SAPMV45 (VBAK-A0101 VKORG) S		[Input]	Input "3000" into instance "1" of the (VBAK-VKORG) Sales Organization CTextField and sen	Continue	Continue
4	Basis Components 1.0	SAPMV45 (VBAK-A0101 VTWEG)		[Input]	Input "10" into instance "1" of the (VBAK-VTWEG) Distribution Channel CTextField and s	Continue	Continue
5	Basis Components 1.0	SAPMV45 (VBAK-A0101 SPART) Di		[Input]	Input "00" into instance "1" of the (VBAK-SPART) Division CTextField and send Key "Non	Continue	Continue
6	Basis Components 1.0	SAPMV45 SAPMV45 A0101		[SendVKey]	SendVKey "Enter" to SAPMV45A0101 Window while Wait For Result is "True"	Continue	Continue

14. Right-click the last step and select **Insert Step(s) Below Using Certify Capture**.

- 15.** Enter data in the fields required for the second screen of VA01.

- Sold-to party
- Ship-to party
- PO Number
- PO Date
- Material
- Order Quantity
- *Any other required fields*

The screenshot shows the SAP interface for creating an ERP standard order. The top navigation bar includes icons for search, back, forward, and various system functions. The main title is "Create ERP Standard Order: Overview". Below the title, there are several input fields: "ERP Standard Order" (with a placeholder value), "Net value" (0.00), "Sold-to party" (highlighted with a red box), "Ship-to party" (highlighted with a red box), "PO Number" (highlighted with a red box), and "PO date" (highlighted with a red box). A toolbar below these fields contains icons for new, edit, delete, etc. The main content area is divided into tabs: Sales (selected), Item overview, Item detail, Ordering party, Procurement, Shipping, and Reason for rejection. Under the Sales tab, there are fields for "Req. deliv.date" (08/22/2014), "Deliver.Plant" (highlighted with a red box), "Complete dlv." (checkbox), "Total Weight" (0.000), "Delivery block" (dropdown), "Volume" (0.000), "Billing block" (dropdown), "Pricing date" (08/15/2014), "Payment card" (dropdown), "Exp.date" (highlighted with a red box), "Payment terms" (dropdown), "Incoterms" (dropdown), "Order reason" (dropdown), and "Sales area" (3000 / 10 / 00 USA Philadelphia, Direct Sales, Cross-division). At the bottom of the screen is a toolbar with various icons for search, print, and file operations. A table titled "All items" is visible at the bottom, showing columns for Item, Material, Order Quantity, SU, Description, and Customer Material Numb. The first row of the table is also highlighted with a red box.

- 16.** Press the **Enter** key.

- 17.** Right click on the Worksoft Process Capture icon and select **End Capture**.

The steps you just captured are in your process.

- 18.** **Save** the process.

Note: Some of the captured steps may not be listed in the same order in which you performed them. This is okay and will not affect the creation of the order.

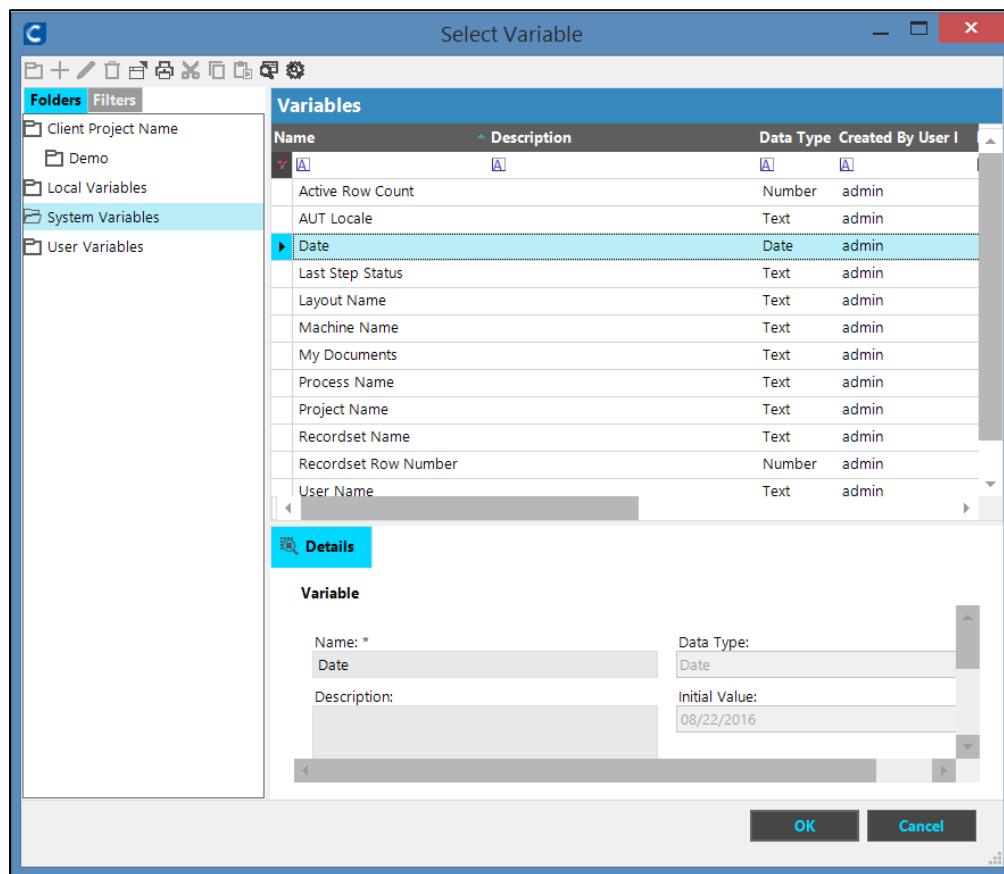
EXERCISE 2.4 — Enhance the VA01_CreateStandardOrder Process

As covered in the Certify Basics course, Certify has built-in system functionality that can enhance your processes. This exercise will use the System Date variable to populate some of your data.

Step	Action
------	--------

- In the step addressing the **PO Number** field, select a variable in the Parameters tab.

The Select Variable window appears.

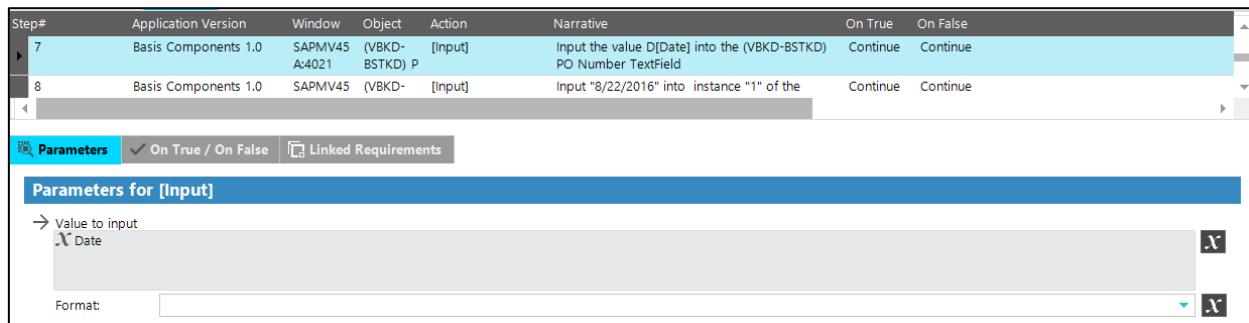


Tip: In most sales orders, PO numbers must be unique. The System Date variable with the Timestamp format creates an original PO number.

- In the tree, click the **System Variables** folder.
- In the Variables pane, select **Date**.

4. Click OK.

The variable name appears in the Value to input field.



5. Click the drop-down arrow in the Format field and select **Timestamp.**

6. Save the process.

7. In the step addressing the **PO Date field, select a variable in the Parameters tab.**

The Select Variable window appears.

8. In the tree, click the **System Variables folder.**

9. In the Variables pane, select **Date. Note: Make sure the Date is in the appropriate format for your SAP environment.**

10. Click OK.

11. Save the process.

EXERCISE 2.5 — Create a Layout and Recordset for VA01_CreateStandardOrder

In this exercise, you will create a layout and recordset for VA01_CreateStandardOrder using the Add to Layout feature.

Step	Action
------	--------

1. Navigate to your VA01_CreateStandardOrder process inside your Sandbox folder and right-click and select **Edit**.

The Edit Process window appears.

2. Click on the Steps tab and select steps 2-6 and steps 8-9.

Your step numbers could vary depending on how Certify brought them in during LiveTouch or Capture. Ensure you have the proper steps selected. Use the steps shown below.

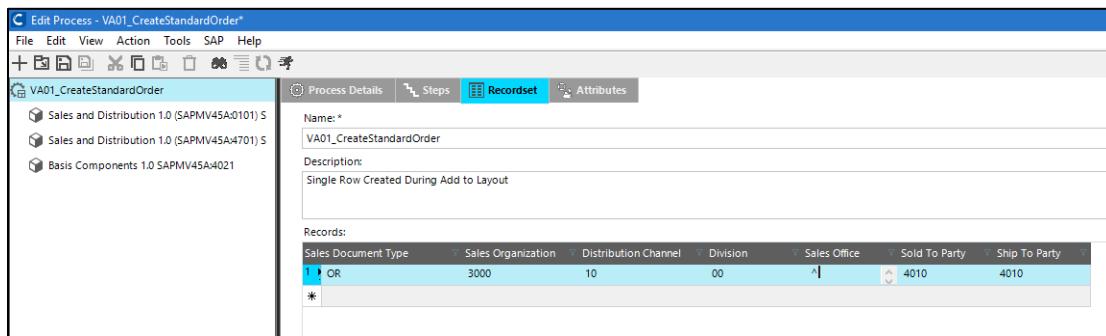
Step#	Application Version	Object	Action	Narrative
1	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	Input the value "VA01" into the okcd
2	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	Input "00" into instance "1" of the (VBAK-AUJART) Sales Document Type CTextField and send Key "None" to Window while wait for result is "True"
3	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	Input "3000" into instance "1" of the (VBAK-VKORG) Sales Organization CTextField and send Key "None" to Window while wait for result is "True"
4	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	Input "10" into instance "1" of the (VBAK-VTWEG) Distribution Channel CTextField and send Key "None" to Window while wait for result is "True"
5	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	Input "00" into instance "1" of the (VBAK-SPART) Division CTextField and send Key "None" to Window while wait for result is "True"
6	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	Input "3010" into instance "1" of the (VBAK-VKBUR) Sales Office CTextField and send Key "None" to Window while wait for result is "True"
7	Sales and Distribution 1.0 (SAPMV45A 0101)	SAPMV45A 0101	[Input]	"Press" Button btn[0] instance "1"
8	Sales and Distribution 1.0 (SAPMV45A 4701)	SAPMV45A 4701	[Input]	Input "4010" into instance "1" of the Sold-to party (GuicTextField) and send Key "None" to Window while wait for result is "True"
9	Sales and Distribution 1.0 (SAPMV45A 4701)	SAPMV45A 4701	[Input]	Input "4010" into instance "1" of the Ship-to party (GuicTextField) and send Key "None" to Window while wait for result is "True"

3. Right-click on the highlighted steps and choose **Add to Layout**.

The Add to Layout window appears.

Narrative	Parameter	Value	Action	Variable	Type	Default Value
Input "OR" into	Value to input	OR	Create Variable	Sales Document Type	Text	
Input "3000" into	Value to input	3000	Use Variable	Sales Organization	Text	
Input "10" into	Value to input	10	Use Variable	Distribution Channel	Text	
Input "00" into	Value to input	00	Use Variable	Division	Text	
Input "3010" into	Value to input	3010	Use Variable	Sales Office	Text	
Input "4010" into	Value to input	4010	Use Variable	Sold To Party	Text	
Input "4010" into	Value to input	4010	Use Variable	Ship To Party	Text	

4. Check to ensure you are using the correct *Parameter* option and select the *Variables* that should be used for each step.
5. Give the recordset a name: **VA01_CreateStandardOrder**.
6. Click **OK**.
7. Navigate to the Recordset tab and ensure your values pulled in correctly.



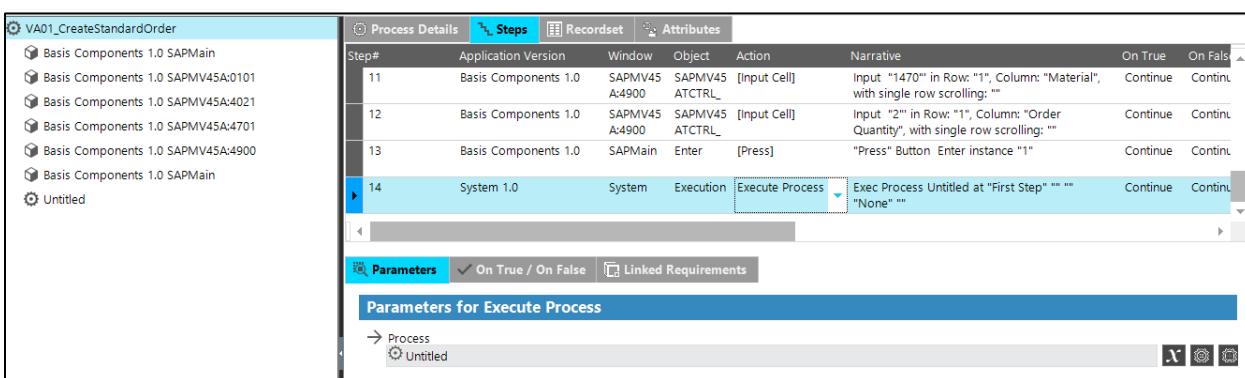
8. Update the Sales Office value to the skip character (^) as shown above to indicate that this field should be skipped during this process.

Tip: It is best practice to use the Add to Layout feature to create a layout and recordset for any fields you will be using on each new window or screen.

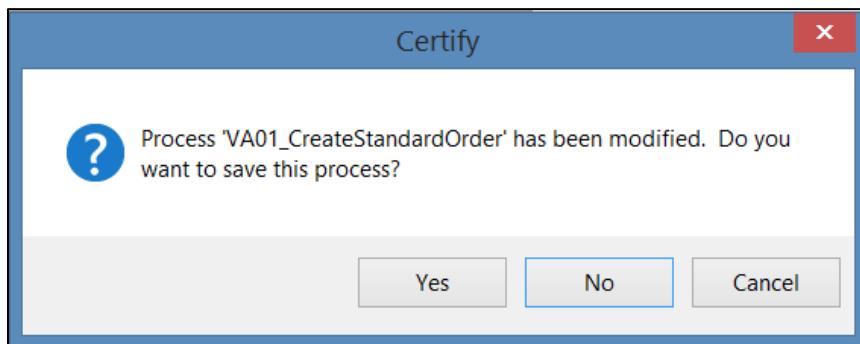
EXERCISE 2.6 — Create the VA01_CreateStandardOrder_C_EnterMaterials Process

Sub-Processes	Description
VA01_CreateStandardOrder	Creates a basic standard order
VA01_CreateStandardOrder_C_EnterMaterials	Enters material for the order
VL01N_CreateOutboundDelivery	Creates an outbound delivery
VL02N_PostGoodsIssue	Posts Goods Issue
VF01_CreateBillingDocument	Creates a billing document

When creating a sales order, there can be any number of line items. In this exercise, you will create a child process for VA01_CreateStandardOrder, called **VA01_CreateStandardOrder_C_EnterMaterials** that will handle material line items using a recordset with any number of rows.

Step	Action
1.	In the VA01_CreateStandardOrder process, right-click any step and select New .
	<i>A new System/Execution/Comment step is added to the end of your process.</i>
	Tip: Another way to insert a new step is to place your cursor in the Steps area and press Insert on your keyboard.
2.	On the new step, click the Action drop-down arrow and select Execute Process .
3.	In the Parameters tab, in the Process field, select the New Process icon.
	<i>Untitled appears in the field as well as in the tree under VA01_CreateStandardOrder.</i>
	 <p>The screenshot shows the SAP Process Details screen for the VA01_CreateStandardOrder process. The 'Steps' tab is selected. A new step, step 14, has been added at the end of the process. Step 14 is configured as an 'Execute Process' step, with the 'Process' dropdown set to 'Untitled'. The 'Parameters' tab is open, showing the 'Parameters for Execute Process' section with a single parameter named 'Untitled'. The navigation tree on the left shows the structure of the process, including various components and steps like Basis Components and SAPMV45.</p>

4. In the Navigation Tree, click **Untitled**.



A Certify dialog box appears asking to save the current process.

5. Click Yes.
6. Click the **Process Details** tab.
7. Give the process a name and description:
 - a. Name = **VA01_CreateStandardOrder_C_EnterMaterials**.
 - b. Description = **This process enters material line items into a sales order.**
8. Save the process.

Note: A best practice for naming child processes that are specific to certain transactions is ParentProcessName_C_ChildProcessDescription. (i.e., VA01_CreateStandardOrder_C_EnterMaterials.)

Certify allows you to **cut** or **copy** and then **paste** steps from one process into another. For the VA01_CreateStandardOrder_C_EnterMaterial process, we want to cut the materials table steps from the VA01_CreateStandardOrder process and paste them into the new process.

9. Navigate to the **VA01_CreateStandardOrder** process.
10. Right-click the steps captured earlier for populating the Materials table.
11. Select **Cut**.
12. Navigate to the **VA01_CreateStandardOrder_C_EnterMaterials** process.
13. Right-click in the Steps area and select **Paste**.

The steps appear in the Steps area.

VA01_CreateStandardOrder_C_EnterMaterials		Process Details	Steps	Recordset	Attributes		
Step#	Application Version	Window	Object	Action	Narrative	On True	On False
1	Basis Components 1.0 SAPMV45A:4900	SAPMV45	SAPMV45	[Input Cell] A:4900 ATCTRL	Input "1470" in Row: "1", Column: "Material", with single row scrolling: ""	Continue	Continue
2	Basis Components 1.0 SAPMV45A:4900	SAPMV45	SAPMV45	[Input Cell] ATCTRL	Input "2" in Row: "1", Column: "Order Quantity", with single row scrolling: ""	Continue	Continue

14. Right-click on **Step 1** and select **Insert Step Above**.
15. Click the **Action** drop-down arrow and select **Search Empty Row**.
16. In the Parameters tab, in the **Row Number** field, select the variable **_Row**.
17. In the **Column 1 (Name or Number)** field, type **Material**.

*Note: This tells the step to populate **_Row** with the number of the first row where the Material column is empty.*

Tip: Whenever possible, we want to use a column's **Name** rather than number because tables and grids can be customized by the user. This guarantees that we will always have the correct column. Keep in mind that this object/action requires an exact column name match, so make sure to expand the column to see the entire name.

18. Modify the steps with the [Input Cell] action (in this example **Steps 2 – 3**) to use the **_Row** variable as the **Row Number**.
19. **Save** the process.
20. Navigate to the **VA01_CreateStandardOrder** process.
21. **Save** the process.

Now you need steps in VA01_CreateStandardOrder to save the sales order and verify it was created successfully.

22. Use LiveTouch to create steps at the end of VA01_CreateStandardOrder to press the **Save**  button and read the **Status Bar**.

Process Details	Steps	Recordset	Attributes				
Step #	Application Version	Window	Object	Action	Narrative	On True	On False
11	Basis Components 1.0	SAPMain	Enter	[Press]	"Press" Button Enter instance "1"	Continue	Continue
12	System 1.0	System	Execution	Execute Process	Exec Process VA01_CreateStandardOrder_C_EnterMaterials a	Continue	Continue
13	SAP Core 1.0	SAP Main	Save	[Press]	"Press" Button Save instance "1"	Continue	Continue
14	SAP Core 1.0	SAP Main	sbar	Verify Property	Verify sbar StatusBar "Text" Is Equal To ""	Continue	Continue

23. In the last (Status Bar) step, make sure the **Action** is **Verify Property**.
24. In the Parameters tab:
 - a. Property = Text
 - b. Criteria = Contains
 - c. Value = **has been saved**
25. **Save** the process.

Now we want to save the new order number shown in the status bar into a variable so we can use it in our OTC process.

26. Right-click the last step and click **Insert Step Below**.
27. In the new last step, make sure the **Action** is **[Store Parameter]**.
28. In the Parameters tab:
 - a. Index = 2
 - b. Variable = (V)Order Number

Note: The status bar object contains eight index parameters where an SAP developer can store bits of data. A single status bar message can use up to all eight of the parameters. We know that for a VA01 sales order, the new order number is in index 2, but different screens may store info you want in other index parameters. In these cases, some trial and error may be required.

29. Save the process and close the Process Editor.
30. At this point, process VA01_CreateStandardOrder should be self-contained and ready to run. Highlight the process  in the Processes Summary Pane and click the **Run**  icon.

Tip: Check the status in the Result Viewer to see if the process passed or failed. If it failed, expand the Result Viewer tree to determine where the failure occurred.

31. Troubleshoot as needed using the Results Log until the process runs completely without errors.

EXERCISE 2.7 — Create the VL01N_CreateOutboundDelivery Process

Sub-Processes	Description
VA01_CreateStandardOrder	<i>Creates a basic standard order</i>
VA01_CreateStandardOrder_C_EnterMaterials	<i>Enters material for the order</i>
VL01N_CreateOutboundDelivery	Creates an outbound delivery
VL02N_PostGoodsIssue	Posts Goods Issue
VF01_CreateBillingDocument	Creates a billing document

In this exercise, you will create a new process that creates an outbound delivery for our sales order. Use any combination of Certify Capture, LiveTouch, and manual steps that works best for you for creating this process. Be sure to refer to your earlier processes as needed to get help.

Important: When working with multiple transactions in a single SAP session, some data is retained and carried across transactions. Since you're creating processes that can be reused in other test scenarios, you must assume these fields will be blank and include steps to populate them. If you're not sure which fields these are, log out and then back into SAP between transactions.

Step	Action
1.	Create a new process:
	<ul style="list-style-type: none"> a. Process Name = VL01N_CreateOutboundDelivery b. Description = This process creates an outbound delivery.
2.	Enter data in the required fields.
	<ul style="list-style-type: none"> • Transaction = VL01N • Shipping Point • Selection Date (two weeks after today) • Order = <i>sales order created in VA01_CreateStandardOrder</i> • <i>Any other required fields</i>

- 3.** Press **Enter**.

The Delivery Create: Overview screen is displayed.

- 4.** Click **Save**.

Note: SAP has several status bar message types. These include **S – Successful**, **W – Warning**, **E – Error**. Using **Verify Property > MessageType** is an easy way to check for generic messages.

- 5.** Check the status bar to make sure the delivery document was created successfully:
 - a.** Action = Verify Property
 - b.** Property = MessageType
 - c.** Criteria – Is Equal To
 - d.** Value = S

- 6.** Save the Delivery number to a variable for use later. Hint – [Store Parameter]



- 7.** Update the step for the Selection Date field to use the System Date. Use math to add 14 days to the system date.
- 8.** Convert the static values in your process to variables and create your layout and recordset using the Add to Layout feature.
- 9.** **Save** the process.

Important: Certify can determine if SAP is currently executing a transaction in the foreground. However, if SAP is processing in the background there is no indicator to Certify. You should include verification steps to handle the case where the Sales Order cannot be processed and the error message contains "is currently being processed".

EXERCISE 2.8 — Create the VL02N_PostGoodsIssue Process

Sub-Processes	Description
VA01_CreateStandardOrder	Creates a basic standard order
VA01_CreateStandardOrder_C_EnterMaterials	Enters material for the order
VL01N_CreateOutboundDelivery	Creates an outbound delivery
VL02N_PostGoodsIssue	Posts Goods Issue
VF01_CreateBillingDocument	Creates a billing document

In this exercise, you will create a short process that posts a goods issue against our sales order. Use any combination of Certify Capture, LiveTouch, and manual steps that works best for you for creating this process. Be sure to refer to your earlier processes as needed to get help.

- | Step | Action |
|------|---|
| 1. | Create new process:
a. Process Name = VL02N_PostGoodsIssue
b. Description = This process posts a goods issue. |
| 2. | Enter data in the required fields.
• Transaction = VL02N
• Outbound Delivery = <i>outbound delivery created in VL01N_CreateOutboundDelivery</i>
• <i>Any other required fields</i>
• Press Enter |



3. Click **Post Goods Issue**.
4. Check the status bar to make sure the delivery document was saved successfully.

Update this step in Certify and use the same variable you used in the previous process for Outbound Delivery number.

 Delivery 80024248 has been saved

5. Save the process.

EXERCISE 2.9 — Create the VF01_CreateBillingDocument Process

Sub-Processes	Description
VA01_CreateStandardOrder	Creates a basic standard order
VA01_CreateStandardOrder_C_EnterMaterials	Enters material for the order
VL01N_CreateOutboundDelivery	Creates an outbound delivery
VL02N_PostGoodsIssue	Posts Goods Issue
VF01_CreateBillingDocument	Creates a billing document

In this exercise, you will create a billing document as the final sub-process for our OTC process. Use any combination of Certify Capture, LiveTouch, and manual steps that works best for you for creating this process. Be sure to refer to your earlier processes as needed to get help.

Step	Action
1.	Create new process:
a.	Process Name = VF01_CreateBillingDocument
b.	Description = This process creates a billing document.
2.	Enter data in the required fields.
	<ul style="list-style-type: none"> Transaction = VF01 Document = <i>outbound delivery created in VL01N_CreateOutboundDelivery</i>
3.	Click Save .

4. Check the status bar to make sure the delivery document was saved successfully.

5. Save the billing document number.

Tip: In this transaction, the document number in the status bar is not in index 2.

 Document 90037671 has been saved

6. Save the process.
7. Run the process and view your results.

Note: Review your results to ensure the correct index was used in your Store Parameter step.

Important: If you receive a Pass result, you cannot assume the process is complete and accurate. You should always check your results' parameters to ensure correct data was stored, i.e. correct index numbers, and all layouts and recordsets have been added.

EXERCISE 2.10 — Assemble the OTC_StandardOrder Process

Sub-Processes	Description
VA01_CreateStandardOrder	Creates a basic standard order
VA01_CreateStandardOrder_C_EnterMaterials	Enters material for the order
VL01N_CreateOutboundDelivery	Creates an outbound delivery
VL02N_PostGoodsIssue	Posts Goods Issue
VF01_CreateBillingDocument	Creates a billing document

With all the sub-processes created, you need to put them into your integrated process created earlier in this lesson. In this exercise, you will assemble the OTC_StandardOrder integrated process.

- | Step | Action |
|------|--|
| 1. | Open OTC_StandardOrder for editing. |
| 2. | Add steps to execute each of the following process in order: |
| | <ul style="list-style-type: none"> • VA01_CreateStandardOrder • VL01N_CreateOutboundDelivery • VL02N_PostGoodsIssue • VF01_CreateBillingDocument |

Step #	Application Version	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System	Execution	Execute Process	Exec Process VA01_CreateStandardOrder at "First Step" *** "None" ***	Continue	Continue
2	System 1.0	System	Execution	Execute Process	Exec Process VL01N_CreateOutboundDelivery at "First Step" *** "None" ***	Continue	Continue
3	System 1.0	System	Execution	Execute Process	Exec Process LT03_PickGoodsViaTransferOrder at "First Step" *** "None" ***	Continue	Continue
4	System 1.0	System	Execution	Execute Process	Exec Process VL02N_PostGoodsIssue at "First Step" *** "None" ***	Continue	Continue
5	System 1.0	System	Execution	Execute Process	Exec Process VF01_CreateBillingDocument at "First Step" *** "None" ***	Continue	Continue

3. Save the process and close the Process Editor.
4. At this point, the process is ready to run. Execute the process and troubleshoot as needed using the Results Log until the process runs completely without errors.

Using Layouts with the OTC_StandardOrder process

The OTC process creates a sales order using static (hardcoded) data with a few variables for calculated fields and the newly created documents. If you wanted to run the OTC process with various sets of data, you could use layouts/recordsets for the various data entry points in:

- VA01_CreateStandardOrder
- VA01_CreateStandardOrder_C_EnterMaterials

Additional system variables can be used for calculating dates if needed.

VA01_CreateStandardOrder Layout/Recordset

Sales Order Type	Sales Organization	Distribution Channel	Division	Sales Office	Sold-to party	Ship-to party
OR	3000	10	00	^	4055	4055

VA01_CreateStandardOrder_C_EnterMaterials Layout/Recordset

Material	Quantity
1470	1

Sample completed OTC_StandardOrder process

The OTC process will vary depending on your SAP system. The following are **samples** of the process as created using the Worksoft SAP environment.

VA01_CreateStandardOrder Sample

Step	Application Version	Window	Object	Action
1	Account Management 1.0	SAPMain	okcd	[Input]
Value to input		/nVA01		
Instance (row)		1		
2	Sales and Distribution 1.0	(SAPMV45A:0101) SAPMV45A 0101	(VBAK –AUART) Sales Document Type	[Input]
Value to input		OR		
Key		None		
Instance (row)		1		
3	Sales and Distribution 1.0	(SAPMV45A:0101) SAPMV45A 0101	(VBAK –VKORG) Sales Organization	[Input]
Value to input		1000		
Key		None		
Instance (row)		1		
4	Sales and Distribution 1.0	(SAPMV45A:0101) SAPMV45A 0101	(VBAK –VTWEG) Distribution Channel	[Input]
Value to input		10		
Key		None		
Instance (row)		1		
5	Sales and Distribution 1.0	(SAPMV45A:0101) SAPMV45A 0101	(VBAK –SPART) Division	[Input]
Value to input		00		
Key		None		
Instance (row)		1		
6	Account Management 1.0	SAPMain	Enter	[Press]
Type		Press		
Instance (row)		1		
7	Sales and Distribution 1.0	(SAPMV45A:4701) SAPMV45A 4701	Sold-to party (GuiCTextField)	[Input]
Value to input		1000		
Key		None		
Instance (row)		1		

Lesson 2: Defining and Developing Processes for OTC Transactions

Step	Application Version		Window	Object	Action
8	Sales and Distribution 1.0		(SAPMV45A:4701) SAPMV45A 4701	Ship-to party (GuiCTextField)	[Input]
Value to input	1000				
Key	None				
Instance (row)	1				
9	Sales and Distribution 1.0		(SAPMV45A:4021) SAPMV45A 4021	(VBKD-BSTKD) PO Number	[Input]
Value to input	V(_ROW)				
Key	None				
Instance (row)	1				
10	Sales and Distribution 1.0		(SAPMV45A:4021) SAPMV45A 4021	(VBKD-BSTKD) PO Date	[Input]
Value to input	V(_Date)				
Key	None				
Instance (row)	1				
11	System 1.0		System	Execute	Execute Process
Process	VA01_CreateStandardOrder_C_EnterMaterials				
Start At Step	First Step				
12	SAP Core 1.0		SAP Main	Save	[Press]
Type	Press				
Instance (row)	1				
13	SAP Core 1.0		SAP Main	sbar	[Verify Property]
Property	Text				
Criteria	contains				
Value	has been saved				
14	SAP Core 1.0		SAP Main	sbar	[Store Parameter]
Index(1 to 8)	2				
Variable	(V) Order Number				

VA01_CreateStandardOrder_C_EnterMaterials Sample Process

Step	Application Version	Window	Object	Action	
1	Sales and Distribution 1.0	(SAPMV45A:4900) SAPMV45A 4900	(SAPMV45ATCTRL_U_ERF_AUFTRAG)	Search Empty Row	
Row Number		(V)_Row			
Column1(Name or Number)		Material			
2	Sales and Distribution 1.0	(SAPMV45A:4900) SAPMV45A 4900	(SAPMV45ATCTRL_U_ERF_AUFTRAG)	[Input Cell]	
Value		p-100			
Row Number		(V)_Row			
Column (Name or Caption)		Material			
3	Sales and Distribution 1.0	(SAPMV45A:4900) SAPMV45A 4900	(SAPMV45ATCTRL_U_ERF_AUFTRAG)	[Input Cell]	
Value		1			
Row Number		(V)_Row			
Column (Name or Caption)		Order Quantity			
4	Sales and Distribution 1.0	(SAPMV45A:4900) SAPMV45A 4900	(SAPMV45ATCTRL_U_ERF_AUFTRAG)	[Input Cell]	
Value		PC			
Row Number		(V)_Row			
Column (Name or Caption)		SU			
5	SAP Core 1.0	SAP Main	Enter	[Press]	
Type		Press			
Instance (row)		1			

VL01N_CreateOutboundDelivery Sample Process

This process uses a technique we call “intelligent waiting”.

- A status is checked. If SAP is not ready to continue:
- Wait 10 seconds
- Increment Counter variable
- Verify that the Counter hasn't exceeded the number of attempts.
- Jump back to check status again

Step	Application Version	Window	Object	Action
1	System1.0	System	Number	Set
To the Value	1			
Variable	(V) Counter			

Step	Application Version	Window	Object	Action
2	System1.0	System	Execution	Label
Label Name	Create Outbound Delivery check			

Step	Application Version	Window	Object	Action
3	SAP Core1.0	SAP Main	okcd	[Input]
Value	/nVL01N			
Wait For Result (Do not give result till the end of execution)	Checked			
Instance (row)	1			

Step	Application Version	Window	Object	Action
4	Sales and Distribution 1.0	SAPMV50A:4001	(LIKPV-STEL) Shipping point	[Input]
Value to input	(V) Shipping point			
Key	None			
Wait For Result	Unchecked			
Instance (row)	1			

Step	Application Version	Window	Object	Action
5	System1.0	System	Date	Math
Variable	(V) Date			

Lesson 2: Defining and Developing Processes for OTC Transactions

Operation	Add
Amount	2
Result	(V) Delivery Date
Units	Weeks

Step	Application Version	Window	Object	Action
6	Sales and Distribution 1.0	SAPMV50A:4001	(LV50C-DATBI) Selection date	[Input]
Value to input	(V) Delivery Date			
Key	None			
Wait For Result	Unchecked			
Instance (row)	1			

Step	Application Version	Window	Object	Action
7	Sales and Distribution 1.0	SAPMV50A:4001	(LV50C-VBELN) Order	[Input]
Value to input	(V) Order Number			
Key	None			
Wait For Result	Unchecked			
Instance (row)	1			

Step	Application Version	Window	Object	Action
8	SAP Core1.0	SAP Main	Enter	[Press]
Type	Press			
Instance (row)	1			

Lesson 2: Defining and Developing Processes for OTC Transactions

Step	Application Version	Window	Object	Action
9	SAP Core1.0	SAP Main	sbar	Verify Property
Property	Text			
Criteria	Does Not Contain			
Value	is currently being processed			

Step	Application Version	Window	Object	Action
10	System1.0	System	Execution	Wait
Delay	10			

Step	Application Version	Window	Object	Action
11	System1.0	System	Number	Math
Value1	(V) Counter			
Operation	Add			
Value2	1			
Result	(V) Counter			

Step	Application Version	Window	Object	Action
12	System1.0	System	Number	Compare
Value1	(V) Counter			
Criteria	Is Less Than Or Equal To			
Value2	5			

Modify Step 12 to change On True/On False Tab as shown below.

On True		On False	
Log Status as:	Passed	Log Status as:	Failed
Action:	Jump	Action:	Exit Process
Target Step:	Create Outbound Delivery check		

Lesson 2: Defining and Developing Processes for OTC Transactions

Step	Application Version	Window	Object	Action
13	System1.0	System	Execution	Label
Label Name	Document can be processed			

Step	Application Version	Window	Object	Action
14	SAP Core1.0	SAP Main	Save (Ctrl+S)	[Press]
Type	Press			
Instance (row)	1			

Step	Application Version	Window	Object	Action
15	SAP Core1.0	SAP Main	sbar	Verify Property
Property	MessageType			
Criteria	Is Equal To			
Value	S			

Step	Application Version	Window	Object	Action
16	SAP Core1.0	SAP Main	sbar	[Store Parameter]
Index (1 to 8)	2			
Variable	(V) Outbound Delivery Number			

Modify Step 9 to change On True/On False Tab as shown below.

On True		On False	
Log Status as:	Passed	Log Status as:	Skipped
Action:	Jump	Action:	Continue
Target Step:	Document can be processed		

VL02N_PostGoodsIssue Sample Process

Step	Application Version		Window	Object	Action
1	Account Management 1.0		SAPMain	okcd	[Input]
Value to input	/nVL02N				
Instance (row)	1				
2	Logistics Execution 1.0		(SAPMV50A:4004) SAPMV50A 4004	Outbound Delivery (GuiTextField)	[Input]
Value to input	(V) Outbound Delivery Number				
Instance (row)	1				
3	Logistics Execution 1.0		(SAPMV50A:4004) SAPMV50A 4004	Post Goods Issue (Shift + F8)	[Press]
Type	Press				
Instance (row)	1				
4	SAP Core 1.0		SAP Main	sbar	Verify Property
Property	Message Type				
Criteria	Is Equal To				
Value	S				

VF01_CreateBillingDocument Sample Process

Step	Application Version	Window	Object	Action
1	Account Management 1.0	SAPMain	okcd	[Input]
Value to input	/nVF01			
Instance (row)	1			
2	Sales and Distribution 1.0	(SAPMV60A:0102) SAPMV60A 0102	(SAPMV60ATCTRL_ERF_FAKT)	[Input Cell]
Value	(V) Outbound Delivery Number			
Row Number	1			
Column (Name or Caption)	Document			
3	SAP Core 1.0	SAP Main	Save	[Press]
Type	Press			
Instance (row)	1			
4	SAP Core 1.0	SAP Main	sbar	[Verify Property]
Property	Message Type			
Criteria	Is Equal To			
Value	S			
5	SAP Core 1.0	SAP Main	sbar	[Store Parameter]
Index(1 to 8)	1			
Variable	(V) Billing Document Number			

OTC_StandardOrder Sample Process

Step	Application Version	Window	Object	Action
1	System 1.0	System	Execute	Execute Process
Process	VA01_CreateStandardOrder			
Start At Step	First Step			
2	System 1.0	System	Execute	Execute Process
Process	VL01N_CreateOutboundDelivery			
Start At Step	First Step			
3	System 1.0	System	Execute	Execute Process
Process	VL02N_PostGoodsIssue			
Start At Step	First Step			
4	System 1.0	System	Execute	Execute Process
Process	VF01_CreateBillingDocument			
Start At Step	First Step			

Lesson Summary

You've completed the **Defining and Developing Processes for Order to Cash** lesson.

Key points to remember:

- Use a standardized folder structure and naming convention to keep your tests well organized.
- When creating processes for SAP GUI, work at the transaction level beginning with the transaction code and ending with an appropriate validation step.
- Use your Certify tools, Certify Capture and LiveTouch, when possible to save time and effort.
- Use child processes in transactions where an unknown number of items may be processed.
- Save your processes frequently.

Lesson 3

Defining and Developing Processes for SAP GUI using Certify Data

Overview

In this lesson, you will build and execute processes for an SAP GUI application using the Certify Data feature.

Objectives

After completing this lesson, you will be able to:

- Use the Certify Data feature to extract data.

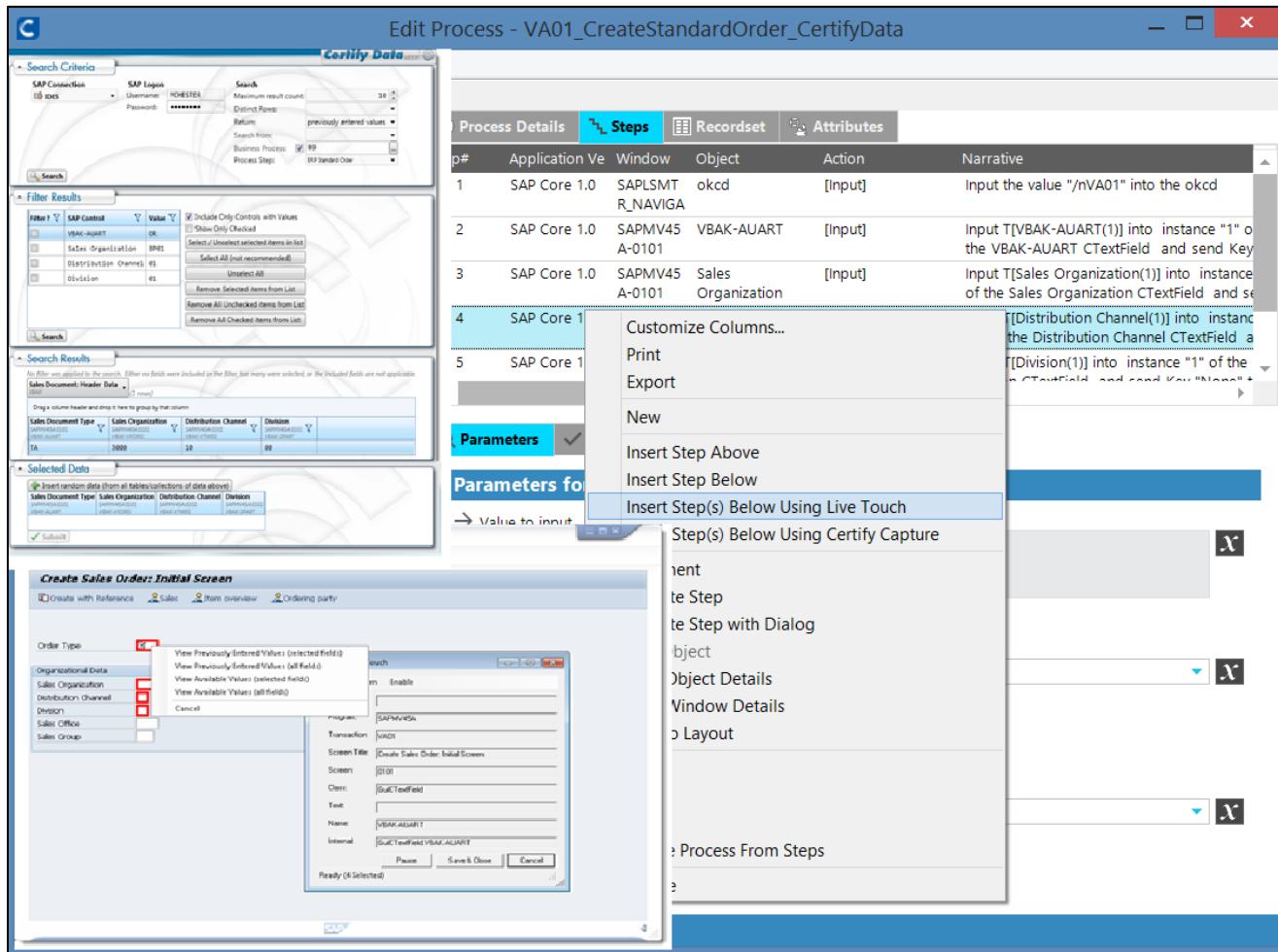
Certify Data Overview

Certify Data is a solution for extracting your own previously used SAP data and inserting it into the Certify test processes as the test steps are constructed. This can be helpful if you don't have data to use to build the test. It should also be updated using variables, layouts, and recordsets as shown in previous lessons. It can be accessed from within Worksoft Certify.

Certify Data:

- Does NOT replace data from business users
- Is used to help build process steps by
 - Facilitating automation when data is unavailable from business users
 - Adding additional data to a defined recordset (*this is temporary and does not replace master data*)

Figure 1 — How Certify Data Works



To utilize Certify Data, you must open the SAP application and navigate to the desired screen. Next, open or create a process in Certify. You add steps using LiveTouch by right-clicking a step and selecting **Insert Step Below Using LiveTouch**. Certify will minimize and the SAP application will appear, along with the Certify LiveTouch dialog box.

Once you select your objects, you can right-click on a selected object to view a drop-down list. The options are:

- View Previously Entered Values (selected fields)
- View Previously Entered Values (all fields)
- Available Values (selected fields)
- Available Values (all fields)

The **selected fields** options will retrieve anything that will take an input value and is highlighted in red. The **all fields** option will retrieve all fields on the screen.

If you select one of the **available values** options, we return only data somehow utilized or provided by the SAP program containing the search field. Often, this data is used within the SAP GUI to populate drop-down lists, provide “type-ahead” textboxes, or validate unconstrained user input against a list of defined values, for example.

If you select one of the **view previously entered values** options, whenever a user hits the ‘Save’ button (or equivalent function to commit a transaction in SAP), most of the values the user entered (as well as additional program data) are saved to various places in SAP (exactly where and what depends upon each SAP program’s implementation); we then parrot these values back to the user upon request; again, based on our knowledge of the internal workings of each SAP program.

Once you make your selection, the Certify Data window opens.

Certify Data Sections

As you can see in Figure 2, the Certify Data screen is divided into four sections:

Figure 2 —Certify Data

The screenshot shows the Certify Data application window divided into three main sections:

- Search Criteria:** This section contains SAP connection details (SAP Connection: COLOIDES02, Application Server: COLOIDES02, Router String: /H/50..59..239..25..2599/H, Instance Number: 00, Client: 808, Connection String: ASHOST=/H/50..59..239..25..2599/H/COLOIDES02 SYSNR=00 CLIENT=808), search parameters (Maximum result count: 10, Distinct Rows: None, Return: previously entered values), and a search button.
- Search Results:** This section displays a table titled "Sales Document: Header Data". The table has columns: Sales Document Type, Division, Sales Organization, and Distribution Channel. The data shows multiple rows of sales documents with various values in the columns.
- Selected Data:** This section shows a summary table with columns: Sales Document Type, Division, Sales Organization, and Distribution Channel. It includes a link to "Insert random data (from all tables/collections of data above)".

Search Criteria section is pre-populated based on the SAP context menu. You can modify the maximum number of records for the search in the **Maximum result count** field.

You can also change the return option in the **Return** field. The choices are **All Values** and **Previously Selected Values**.

Filter Results displays the SAP fields and controls from the current and previous SAP GUI screens. The Filter checkboxes display the fields and controls that are visible. You can select which SAP fields and controls you want to utilize for filtering SAP data.

By default, the Filter table displays the value for the SAP Control field. You can change the values manually by clicking the value in the **Value** column.

Search Results section shows the data organized by tables and fields as it is retrieved from SAP. In Figure 2 above, the Sales Document Header Data table returned four columns of data. For more complicated screens, the table will be bigger.

Selected Data section shows the data you selected. There are two options:

- Select a row of data from the Search Results pane and drag down to the Selected Data pane.
- Have the system automatically pick a row for you. If you have a lot of data, you don't want to manually choose the data.

Reference: For additional information on the Certify Data feature, refer to the *Using Certify Data* guide.

Using Certify Data with the VA01 Transaction

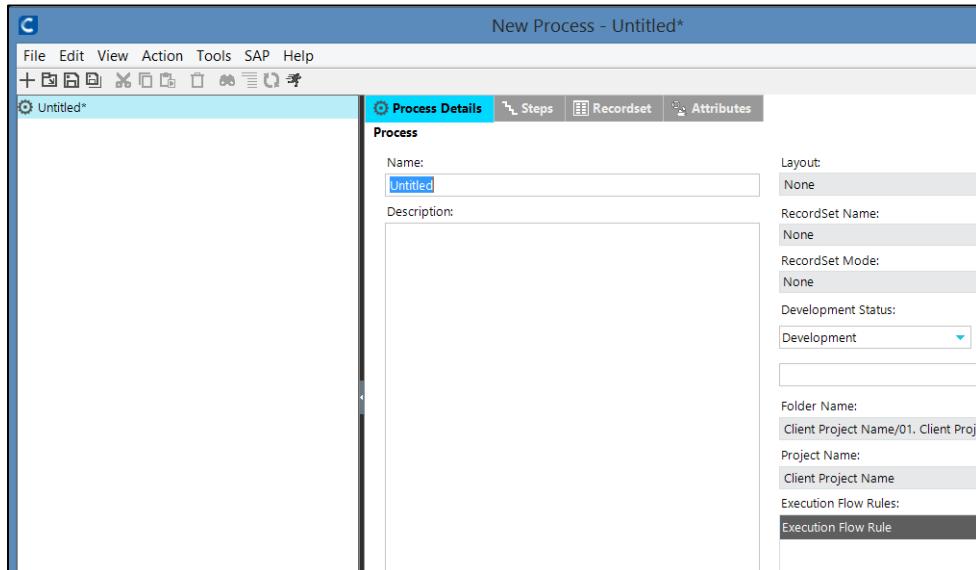
Processes are a collection of steps containing actions to be executed against objects. After completing a step, the next step is executed until the last step has completed. Create a separate process is recommended any time a step sequence is different. For example, creating a new account transaction uses different steps than deleting an account transaction.

EXERCISE 3.1 — Defining the VA01_CreateStandardOrder Process using Certify Data

In this exercise, you will define and add steps for the VA01_CreateStandardOrder sub-process using Certify Data to populate data.

- | Step | Action |
|------|--|
| 1. | In the Navigation Tree, click the Sandbox folder, and then click your name. |
| 2. | Click the OTC_StandardOrderProcess folder. |
| 3. | Right-click in the Summary Pane and select New Process . |

The Process Editor appears. This is where you will define your processes and add the steps using the SAP application.



4. Give the process a name and description:
 - a. In the Process section, in the **Name** field, type **VA01_CreateStandardOrder_CertifyData**.
 - b. In the **Description** field, type **Executes VA01 transaction and inputs initial sales data**.

- c. Click the **Steps** tab.

You are now ready to add steps to the VA01_CreateStandardOrder process.

5. Use LiveTouch to create a step to input **/nVA01** into the Command Input box:

- a. Verify that SAP is open to the SAP Easy Access window.

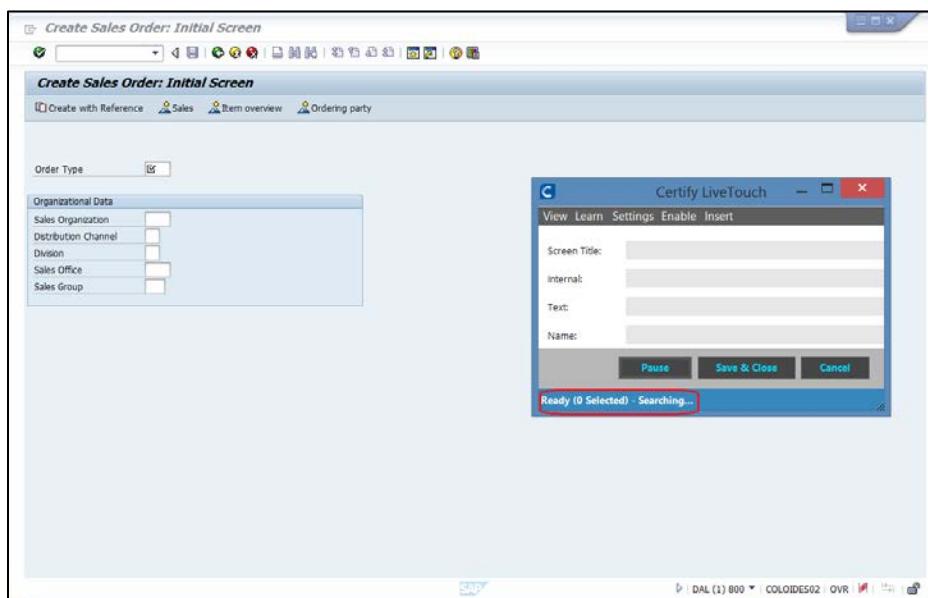
Important: Remember, to use LiveTouch, you must open the SAP application and navigate to the screen that you will use. For this example, the SAP Easy Access window.

- b. In Certify, right-click in the **Steps** area and select **New**.

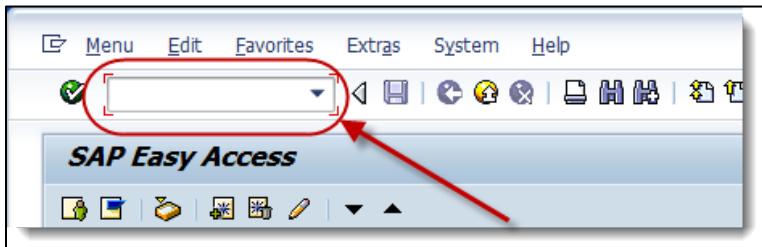
- c. Click the **Application Version** drop-down arrow and select **Select Using LiveTouch**.

Certify minimizes and the SAP window, along with the Certify LiveTouch dialog box appears.

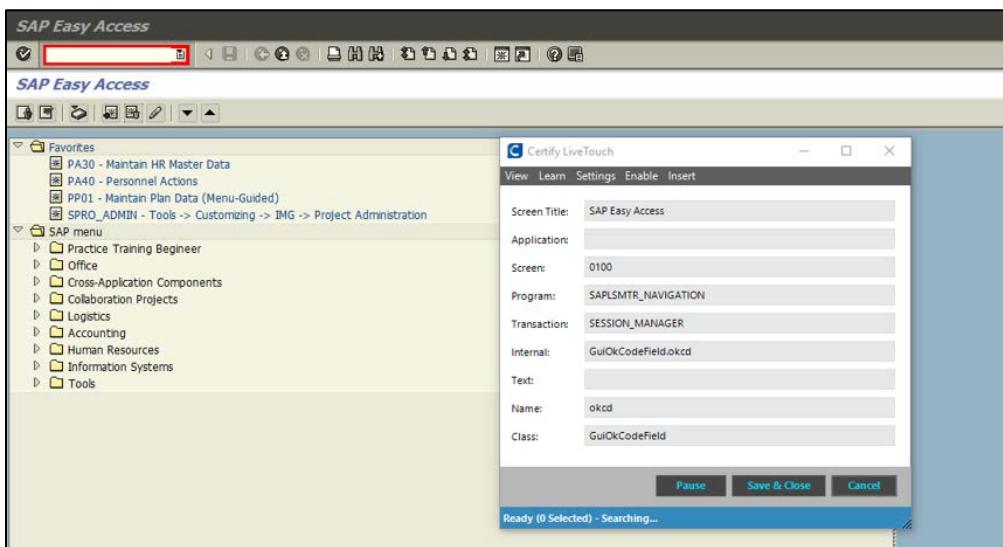
Tip: Before capturing any fields on the page, wait until LiveTouch loads and “Ready” appears at the bottom of the Certify LiveTouch dialog box.



- d. On the SAP window, click the Command Input box.

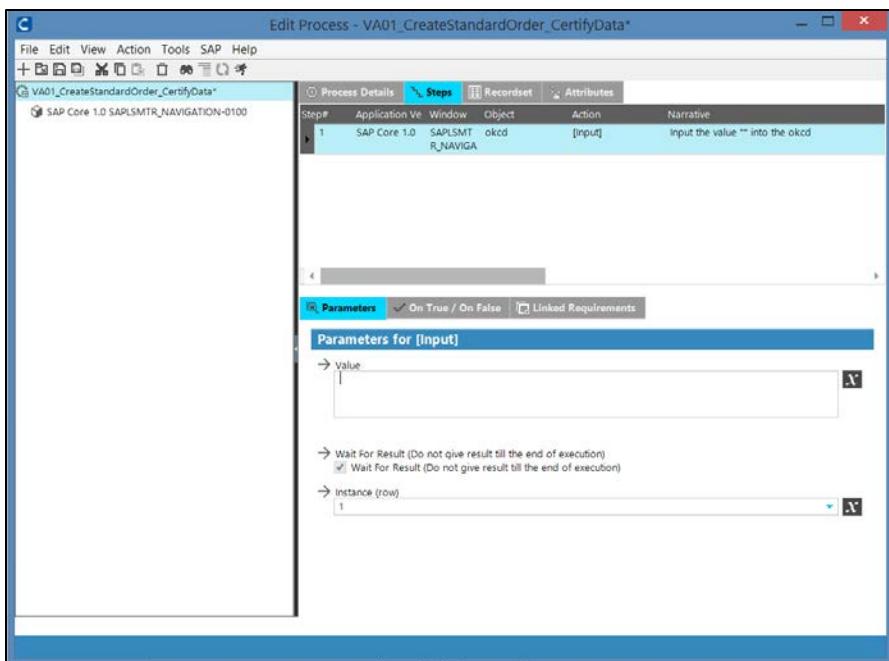


A red highlight appears around the object and the information from the object appears in the Certify LiveTouch dialog box.



- e. In the Certify LiveTouch dialog box, click **Save & Close**.

The Certify Process and Data Editor appears. In the Steps area, a step has been inserted for the object and the step is pre-populated with the application version, window, object, and action.



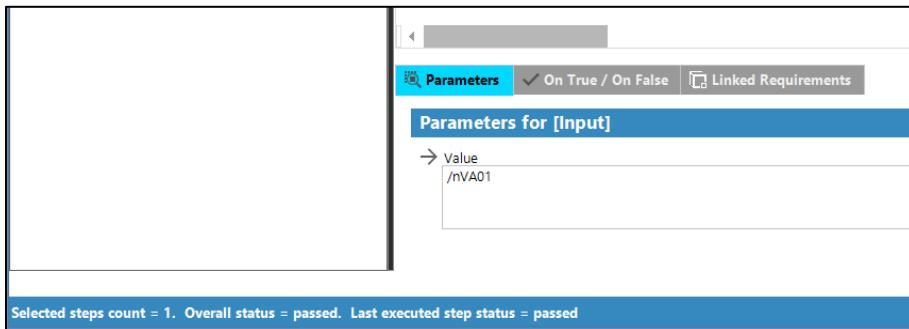
- f. At the bottom of the screen, in the Parameters tab, in the **Value** field, type **/nVA01**.

Next, we will execute the step so that SAP will be on the VA01 screen for the next series of steps. There are several ways to execute a step. Once you highlight the step, you can: 1) click **Action** from the menu and select **Execute Step**, 2) press the **F6** function key, or 3) right-click on the step and select **Execute Step**.

6. To execute the step:

- a. In the Steps area, highlight the step.
- b. Right-click on the step and select **Execute Step**.

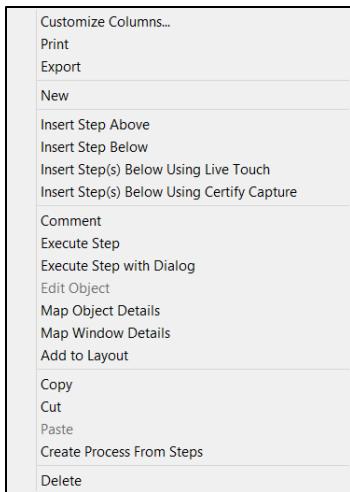
The step executes and SAP displays the Create Standard Order: Initial Screen window. In Certify, notice the status bar. It displays the results of the process. As you can see, the process passed.



7. Use **Certify Data** to extract data from fields on the **Create Sales Order: Initial Screen** window:

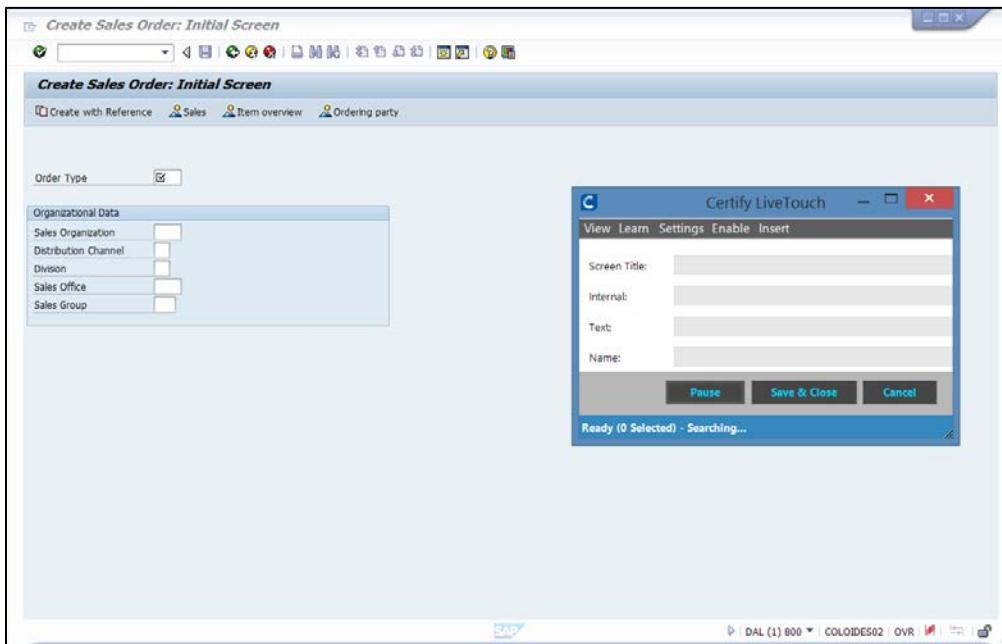
- a. Verify that SAP is open to the Create Sales Order: Initial Screen window.
- b. In Certify, in the **Steps** area, right-click **Step 1**.

A shortcut menu appears.



- c. Select **Insert Step Below Using LiveTouch**.

Certify minimizes and the SAP window, along with the Certify LiveTouch dialog box appears.

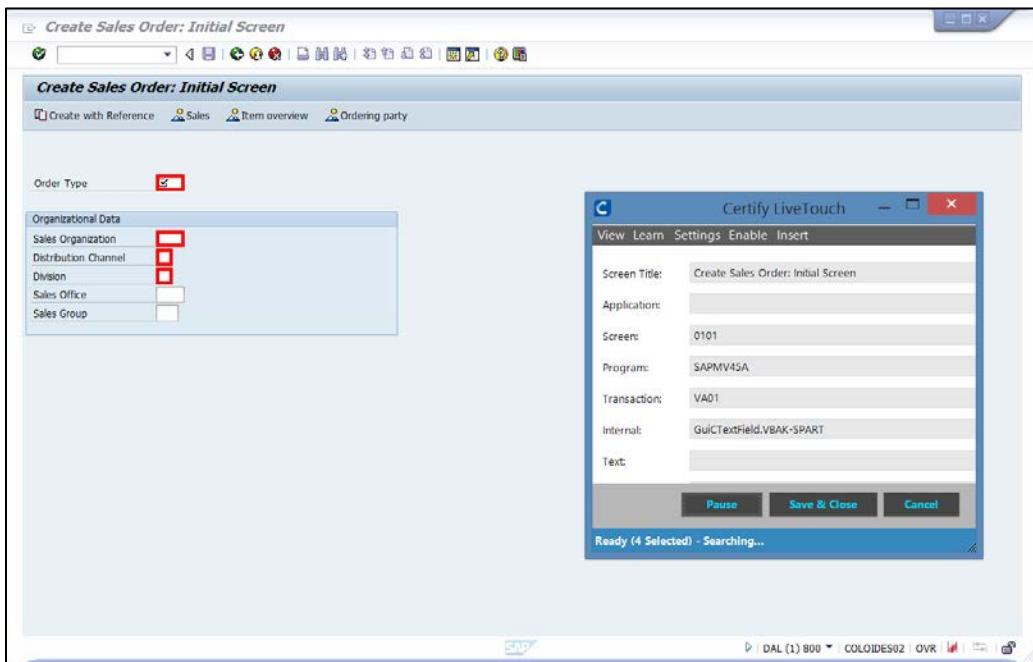


- d. Select the following object fields by clicking in the object fields:
 - Order Type

- Sales Organization
- Distribution Channel
- Division

A red highlight appears around the objects and the information from the last object selected (Enter button) appears in the Certify LiveTouch dialog box.

Now you are ready to extract the data.

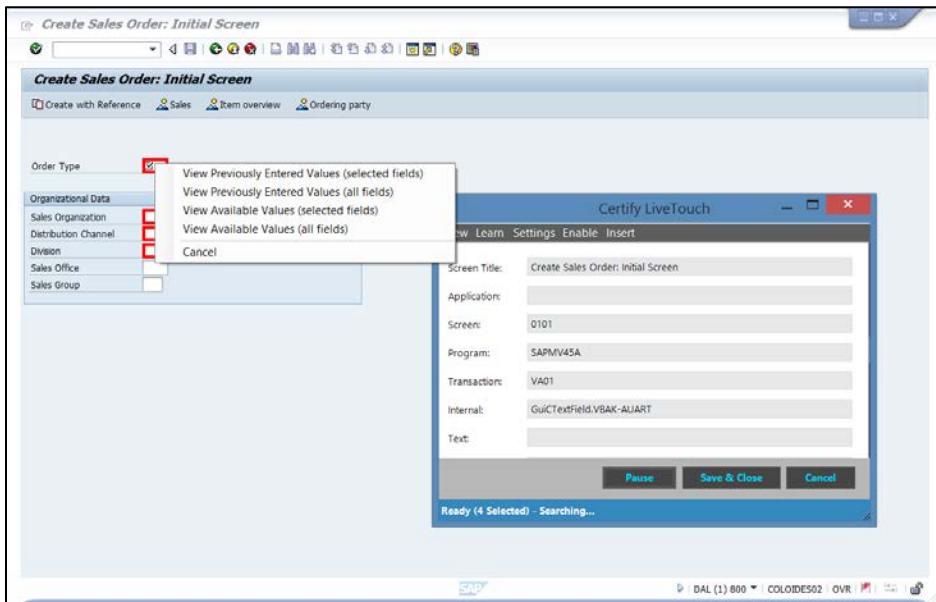


- e. Right-click on the **Order Type** field.

Tip: You can right-click on any of the highlighted fields.

A drop-down menu appears.

f. Select View Previously Entered Values (selected fields).



The Certify Data window appears.

Sales Document Type	Division	Sales Organization	Distribution Channel
SAPMV45A:0101 VBAK-AUART	SAPMV45A:0101 VBAK-SPART	SAPMV45A:0101 VBAK-VKORG	SAPMV45A:0101 VBAK-VIWEG
OR OR OR OR OR OR OR OR	00 00 00 00 00 00 00 00	1000 1000 1000 1000 1000 1000 1000 1000	10 12 12 12 12 12 12 10

Tip: You can close any of the panes by clicking on the section header. For example, to close the Search Criteria, click the **Search Criteria** header.

As previously stated, there are two options for choosing data:

1. Select a row of data from the **Search Results** section and drag down to the **Selected Data** section.
 2. Have the system automatically pick a row for you. If you have a lot of data, you don't want to manually choose the data.
- g. In the **Selected Data** pane, click **Insert random data (from all titles/collections of data above)**.

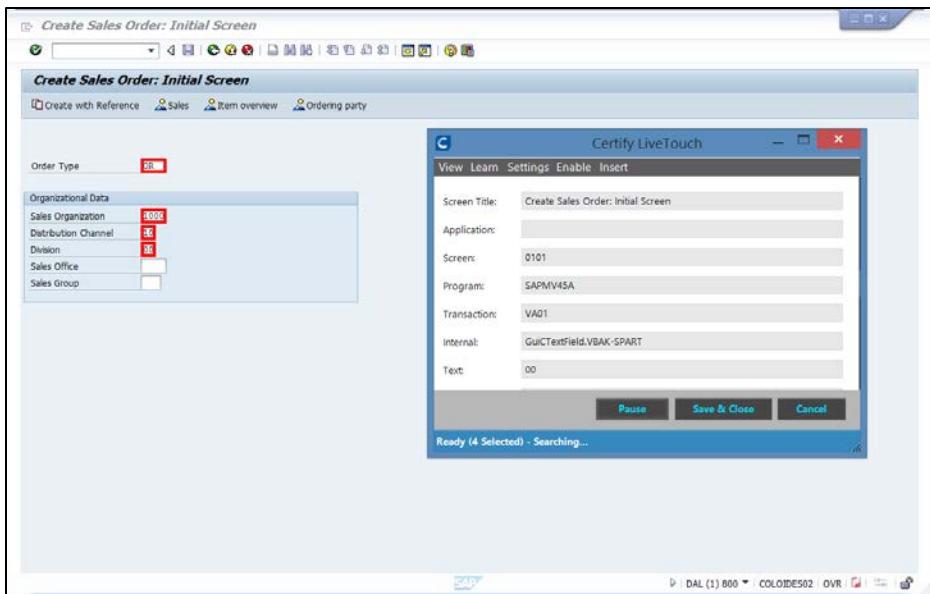
A row appears in the Selected Data pane.

The screenshot shows the Certify Data interface with three main panes:

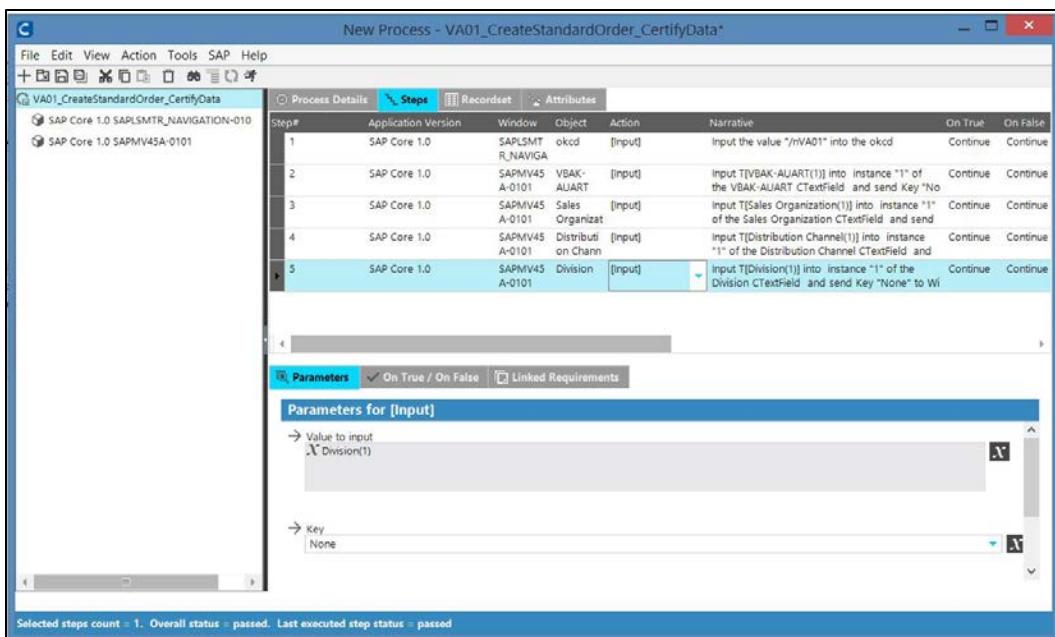
- Search Criteria:** Contains a dropdown menu "Sales Document: Header Data" set to "VBAK" and "(10 rows)". Below it is a table with columns: Sales Document Type, Division, Sales Organization, and Distribution Channel. The first row is highlighted in yellow.
- Search Results:** A table with the same four columns. The first ten rows are listed, with the first row highlighted in yellow.
- Selected Data:** A table with the same four columns. It contains one row, which is also highlighted in yellow. Below the table is a button labeled "Submit".

- h. Click **Submit**.

Certify Data closes and the SAP screen appears. As you can see, the fields are populated with data. At this point, you are back to normal LiveTouch functionality.



- To complete the LiveTouch session, in the Certify LiveTouch dialog box, click **Save & Close**.



Note: You still need to convert the static data to variables and add to a layout and recordset.

Lesson Summary

You've completed the **Defining, Developing, and Executing Processes for SAP GUI Using Certify Data** lesson.

Key points to remember:

- Certify Data is a solution for extracting SAP data and inserting it into the Certify test processes as the test steps are constructed.
- To utilize Certify Data, you must open the SAP application and navigate to the desired screen. Next, open or create a process in Certify. You add steps using LiveTouch by right-clicking a step and selecting Insert Step Below Using LiveTouch. Certify will minimize and the SAP application will appear, along with the Certify LiveTouch dialog box.

Lesson 4

Advanced Processes

Overview

In this lesson, you will learn how to create more advanced SAP processes in Certify.

Add Logic to a Process

This example of using logic will check the SAP status bar to make sure it's clear of errors or warning messages that could cause the test to fail. SAP has four statuses for its status bar:

- Green = Success (S)
- Red = Error (E)
- Yellow = Warning (W)
- Blank (nothing)

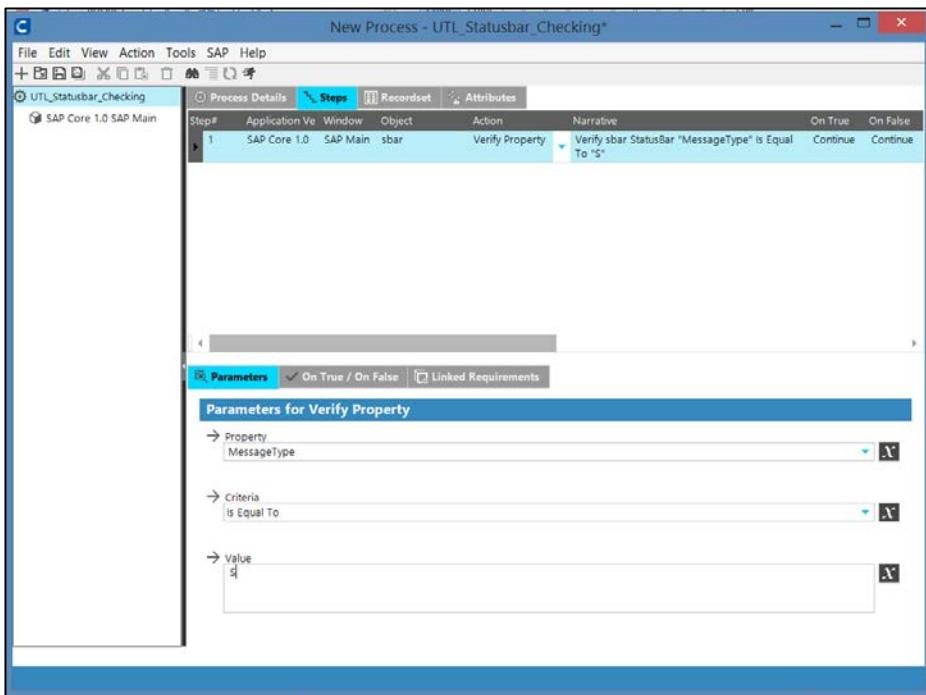
EXERCISE 4.1 — Adding Logic to a Process

This exercise walks through the steps to check the Status bar in SAP to make sure it's clear of errors or warning messages.

Step	Action
1.	In the Navigation Taskbar, click Processes .
2.	In the Navigation Tree, click your personal Sandbox folder.
3.	Create a sub-folder in your Sandbox: <ol style="list-style-type: none">Right-click and select New Folder.In the Name field, type Utilities.Click OK.
4.	In the Navigation Tree, click the Utilities folder.
5.	Right-click in the Summary Pane and select New Process . <i>The Process Editor appears.</i>
6.	Give the process a name and description: <ol style="list-style-type: none">In the Process section, in the Name field, type UTL_Statusbar_Checking.In the Description field, type This process will check the SAP status bar to make sure there are no errors or warning messages appearing that could cause the test to fail.
7.	Add Step #1 to the process. This step will check for a success (or green status bar): <ol style="list-style-type: none">Click the Steps tab.Right-click in the Steps tab and select New.

- c. Click the **Application Version** drop-down arrow and select **SAP Core 1.0**.
- d. Click the **Window** drop-down arrow and select **SAP Main**.
- e. Click the **Object** drop-down arrow and select **sbar**.
- f. Click the **Action** drop-down arrow and select **Verify Property**.
- g. At the bottom of the screen, in the Parameters tab, click the **Property** drop-down arrow and select **MessageType**.
- h. Click the **Criteria** drop-down arrow and select **Is Equal To**.
- i. In the **Value** field, type **s**.

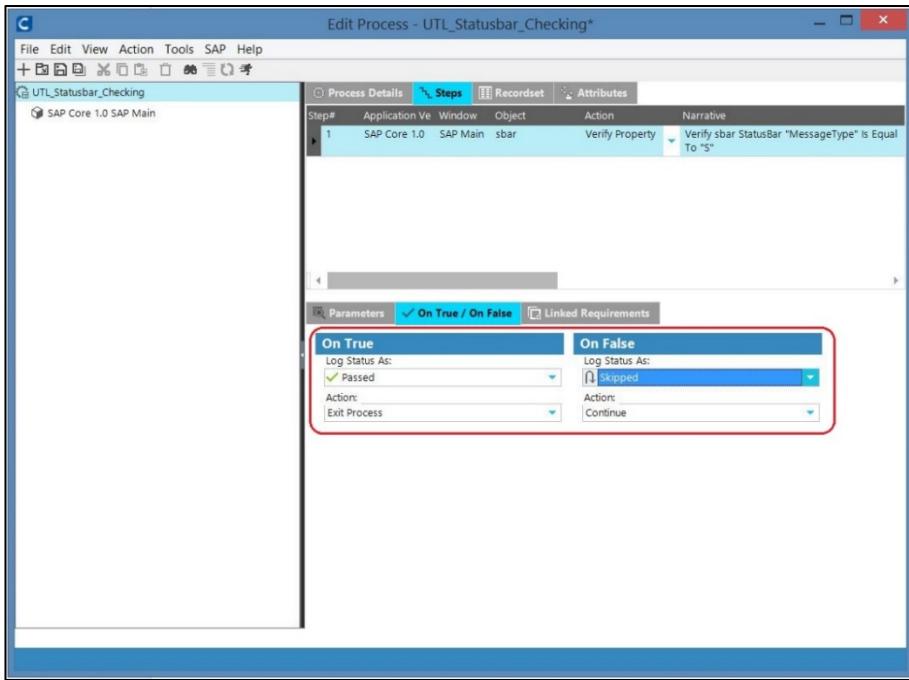
Your screen should look like this:



Next, we want to add logic to the process. If the status bar is equal to "S", we will pass and exit the process. If false, we want to skip (don't fail it) and continue. This will allow the process to continue to check the status bar for other statuses.

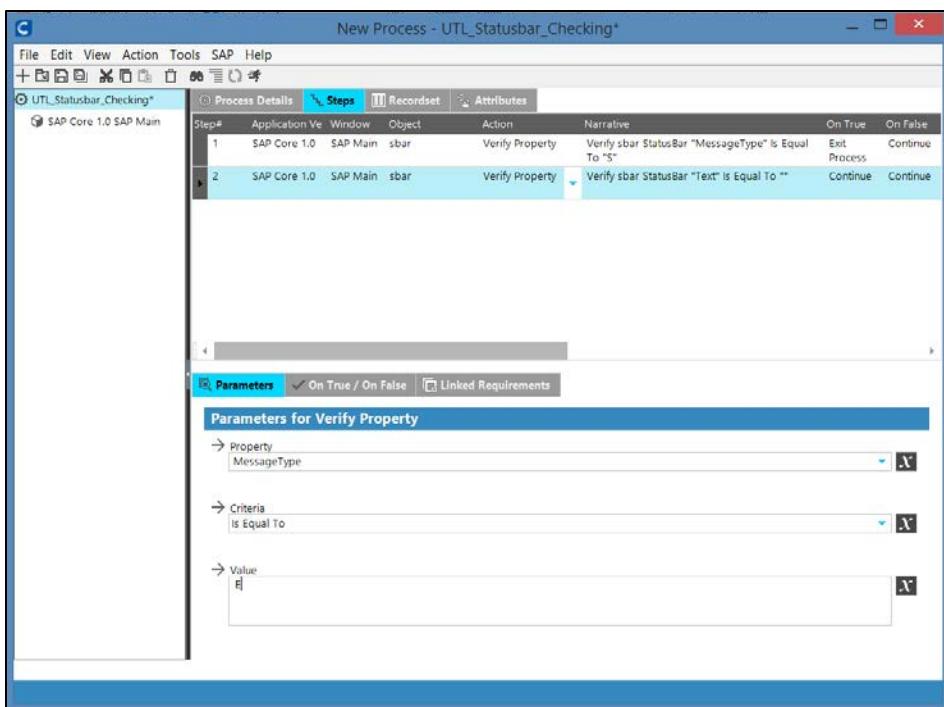
8. To add logic to the process step:
 - a. Click the **On True/On False** tab.
 - b. In the **On True** pane, click the **Action** drop-down arrow and select **Exit Process**.
 - c. In the **On False** pane, click the **Log Status As** drop-down arrow and select **Skipped**.
 - d. In the **On False** pane, click the **Action** drop-down arrow and select **Continue**.

Your screen should look like this:



9. Add **Step 2** to the process. This step will check for an error (or red status bar):
 - a. Click the **Parameters** tab.
 - b. In the Steps area, right-click step #1 and select **Insert Step Below**.
The new step is created in the Steps area. The new step contains the values of the previous step.
 - c. At the bottom of the screen, in the Parameters tab, click the **Property** drop-down arrow and select **Message Type**.
 - d. Click the **Criteria** drop-down arrow and select **Is Equal To**.
 - e. In the **Value** field, type **E**.

Your screen should look like this:

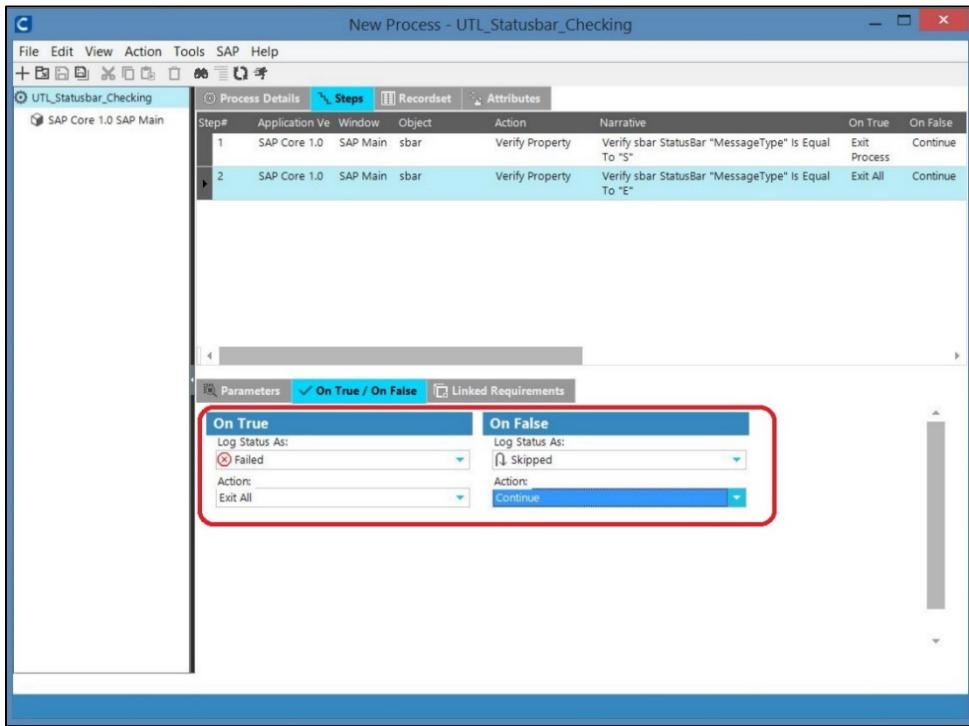


Again, we want to add logic to the process. If the status bar returns an "E", the test is done and we cannot continue. Therefore, we need to set On True to Fail and the action will Exit All to end the test. We need to set On False to Skip and the action to Continue so we can continue checking the status bar.

10. To add logic to the process step:

- a. Click the **On True/On False** tab.
- b. In the **On True** pane, click the **Log Status As** drop-down arrow and select **Failed**.
- c. In the **On True** pane, click the **Action** drop-down arrow and select **Exit All**.
- d. In the **On False** pane, click the **Log Status As** drop-down arrow and select **Skipped**.
- e. In the **On False** pane, click the **Action** drop-down arrow and select **Continue**.

Your screen should look like this:



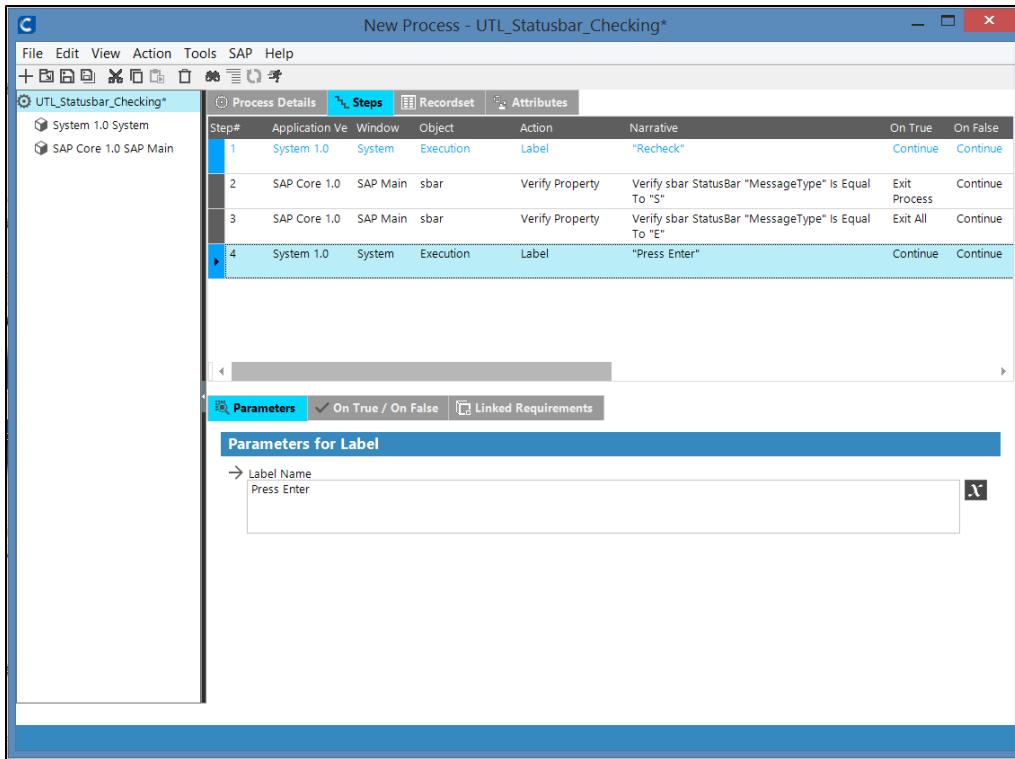
The next step is to check for a warning message (yellow). This will require a few steps. First, we will check for the "W" and if true, we will press Enter and recheck everything. To do this, we will add two steps which will be **Labels** (placeholders as they do not have any action tied to them). Label #1 will be titled "Recheck" and will be added at the beginning of the process. Label #2 will be titled "Press Enter" and will be added at the end of the process.

11. To add labels to the process:

- a. Click the **Parameters** tab.
- b. Right-click step #1 and select **Insert Step Above**.
- c. Click the **Application Version** drop-down arrow and select **System 1.0**.
- d. Click the **Window** drop-down arrow and select **System**.
- e. Click the **Object** drop-down arrow and select **Execution**.
- f. Click the **Action** drop-down arrow and select **Label**.
- g. At the bottom of the screen, in the Parameters tab, In the **Label Name** field, type **Recheck**.
- h. Right-click the last step and select **Insert Step Below**.
- i. Click the **Application Version** drop-down arrow and select **System 1.0**.
- j. Click the **Window** drop-down arrow and select **System**.

- k. Click the **Object** drop-down arrow and select **Execution**.
- l. Click the **Action** drop-down arrow and select **Label**.
- m. At the bottom of the screen, in the Parameters tab, In the **Label Name** field, type **Press Enter**.

Your screen should look like this:

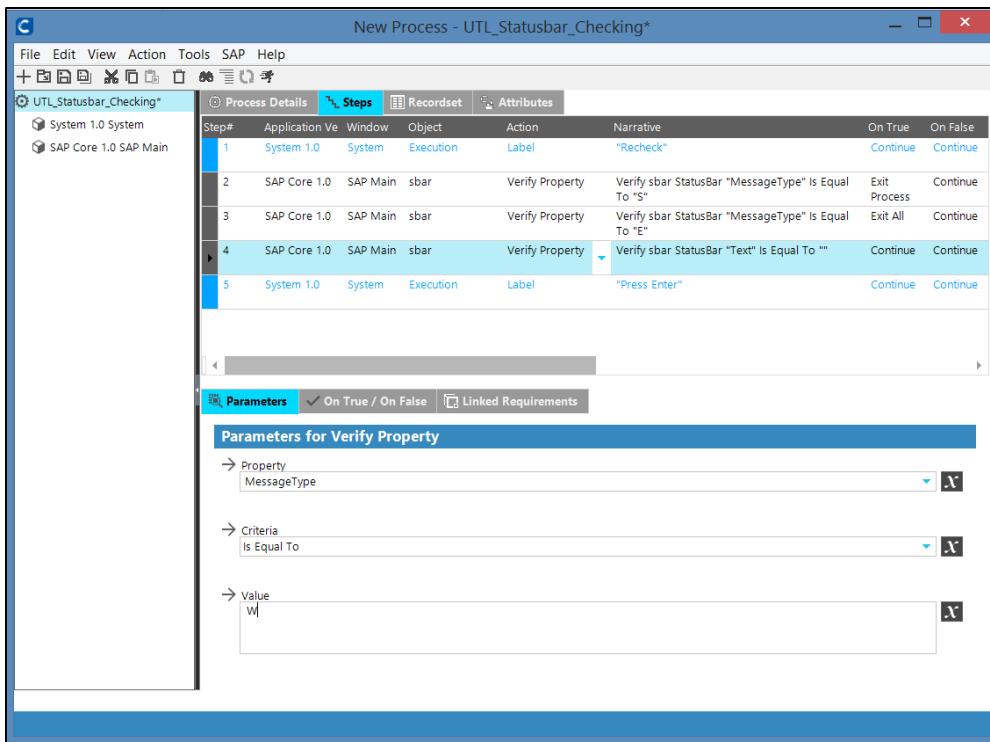


Next, we will add a step to check for a warning message (or yellow status bar). We will add this step before the last label step.

12. Add **Step 4** to the process to check for a warning message:

- a. Right-click step 3 and select **Insert Step Below**.
- b. At the bottom of the screen, in the Parameters tab, click the **Property** drop-down arrow and select **Message Type**.
- c. Click the **Criteria** drop-down arrow and select **Is Equal To**.
- d. In the **Value** field, type **w**.

Your screen should look like this:

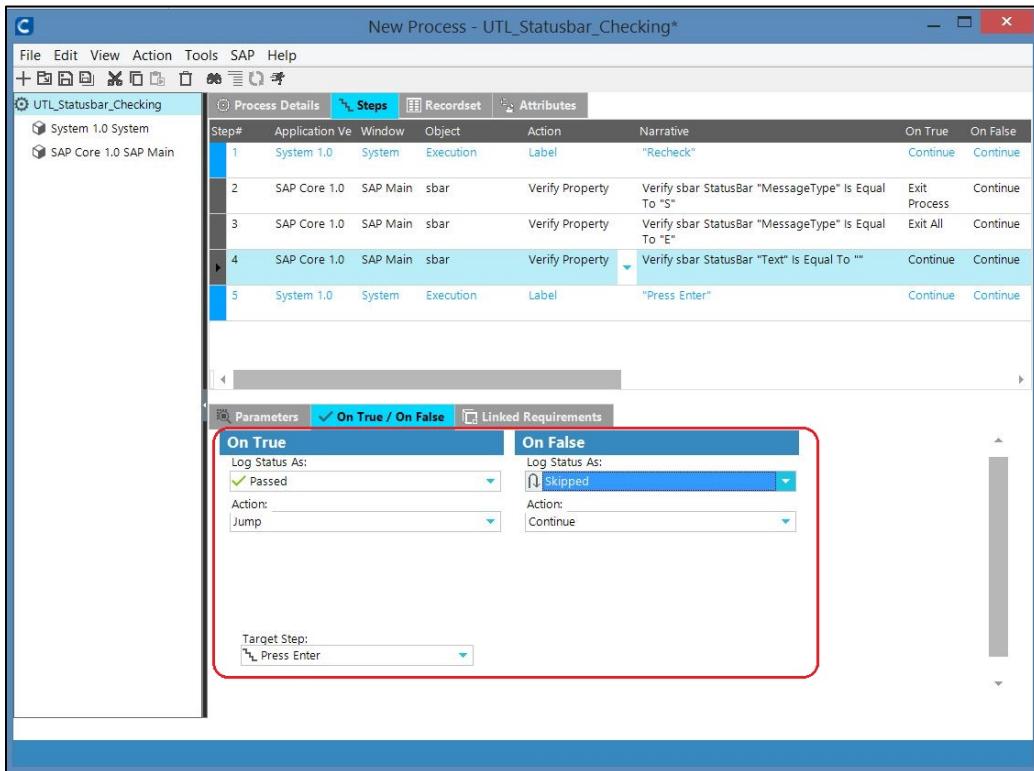


Again, we want to add logic to the process. If the status bar equals "W" the status will equal Passed, the Action will equal Jump, and the Target Step will equal Press Enter (this is populated from the labels we just created). If false, the process will be skipped.

13. To add logic to the process step:

- a. Click the **On True/On False** tab.
- b. In the **On True** pane, click the **Log Status As** drop-down arrow and select **Passed**.
- c. In the **On True** pane, click the **Action** drop-down arrow and select **Jump**.
- d. In the **On True** pane, click the **Target Step** drop-down arrow and select **Press Enter**.
- e. In the **On False** pane, click the **Log Status As** drop-down arrow and select **Skipped**.
- f. In the **On False** pane, click the **Action** drop-down arrow and select **Continue**.

Your screen should look like this:

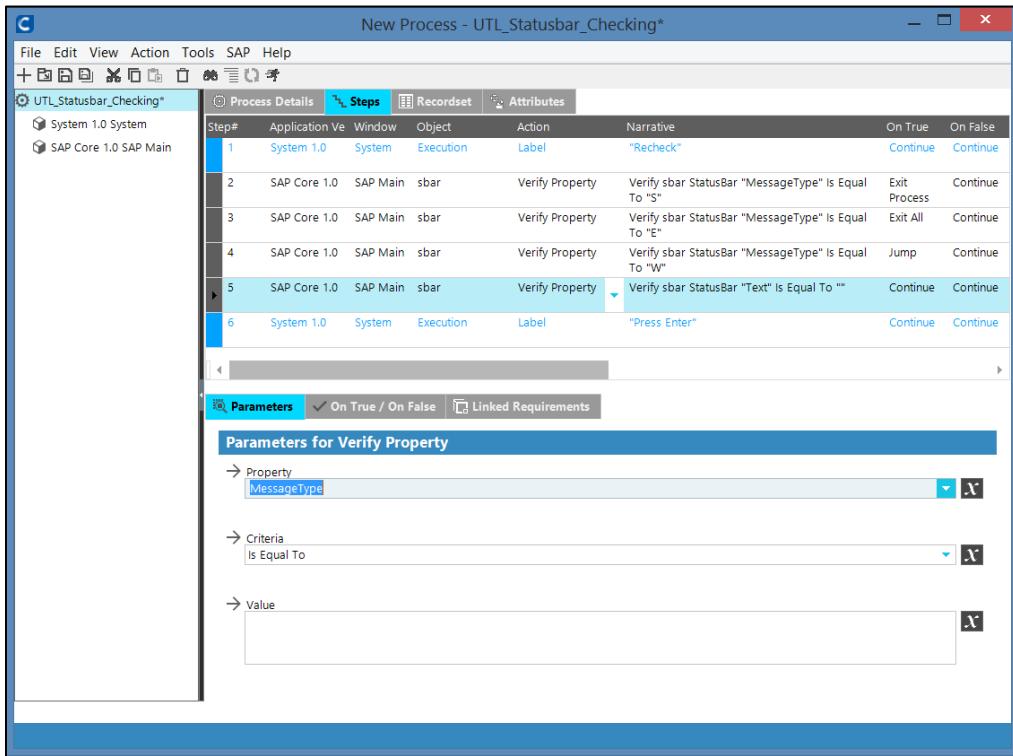


Next, we will add a step to check for a blank status on the status bar. We will add this step before the last label step.

14. Add step to check for the blank status on the status bar:

- Click the **Parameters** tab.
- Right-click step 4 and select **Insert Step Below**.
- At the bottom of the screen, in the Parameters tab, click the **Property** drop-down arrow and select **Message Type**.
- Click the **Criteria** drop-down arrow and select **Is Equal To**.
- In the **Value** field, **leave blank**.

Your screen should look like this:

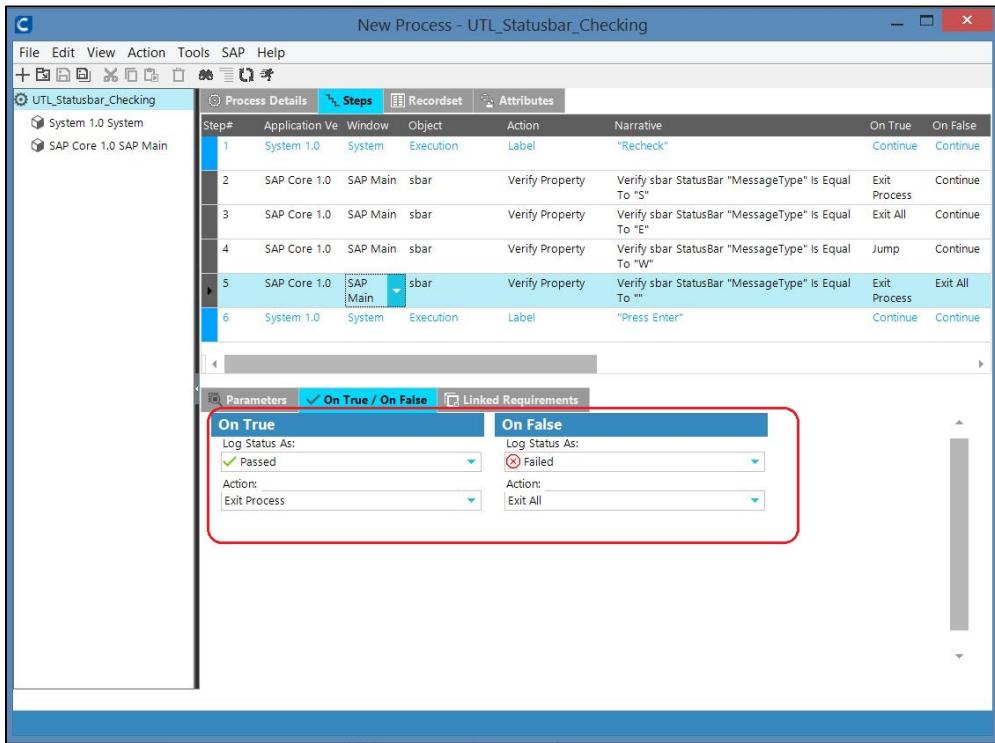


Again, we want to add logic to the process. If the status bar is blank, the test will pass and exit. Blank means there is nothing wrong with the test and we can keep testing. If false, the test will fail and exit all. The reason for this is because the test has something wrong. Usually, the test cannot see the status bar since there is a popup or something else is wrong. We would fail and exit the test since we did not account for the popup.

15. To add logic to the process step:

- Click the **On True/On False** tab.
- In the **On True** pane, click the **Log Status As** drop-down arrow and select **Passed**.
- In the **On True** pane, click the **Action** drop-down arrow and select **Exit Process**.
- In the **On False** pane, click the **Log Status As** drop-down arrow and select **Failed**.
- In the **On False** pane, click the **Action** drop-down arrow and select **Exit All**.

Your screen should look like this:



The last step is to press the **Enter** button and add logic to the step.

16. To add **Step 7** to press Enter:

- Click the **Parameters** tab.
- In the Steps area, right click step #6 and select **Insert Step Below**.
- Click the **Application Version** drop-down arrow and select **SAP Core 1.0**.
- Click the **Window** drop-down arrow and select **SAP Main**.
- Click the **Object** drop-down arrow and select **Enter**.
- Click the **Action** drop-down arrow and select **[Press]**.

Tip: You can also use Certify LiveTouch and select the Enter button.

Again, we want to add logic to the process to recheck the status bar after pressing <Enter>.

17. To add logic to the process step:

- a. Click the **On True/On False** tab.
- b. In the **On True** pane, click the **Log Status As** drop-down arrow and select **Passed**.
- c. In the **On True** pane, click the **Action** drop-down arrow and select **Jump**.
- d. In the **On True** pane, click the **Target Step** drop-down arrow and select **Recheck**.
- e. In the **On False** pane, click the **Log Status As** drop-down arrow and select **Failed**.
- f. In the **On False** pane, click the **Action** drop-down arrow and select **Exit All**.

Your screen should look like this:

Step#	Application	Ve	Window	Object	Action	Narrative	On True	On False
1	System 1.0	System		Execution	Label	"Recheck"	Continue	Continue
2	SAP Core 1.0	SAP Main	sbar		Verify Property	Verify sbar StatusBar "MessageType" Is Equal To "S"	Exit Process	Continue
3	SAP Core 1.0	SAP Main	sbar		Verify Property	Verify sbar StatusBar "MessageType" Is Equal To "E"	Exit All	Continue
4	SAP Core 1.0	SAP Main	sbar		Verify Property	Verify sbar StatusBar "MessageType" Is Equal To "W"	Jump	Continue
5	SAP Core 1.0	SAP Main	sbar		Verify Property	Verify sbar StatusBar "MessageType" Is Equal To ""	Exit All	Process
6	System 1.0	System		Execution	Label	"Press Enter"	Continue	Continue
7	SAP Core 1.0	SAP Main	Enter		[Press]	"Press" Button Enter instance "1"	Jump	Exit All

18. Click the **Save**  button.

19. From the File menu, select **Close** to exit the Process Editor.

The next part of this exercise involves checking to see if the process handles the 4 Message Types.

20. In SAP, make sure that the status bar is blank.

Run your process – it should Pass and exit execution after Step 5.

21. In SAP, trigger a failure status. For example, by entering invalid information for a transaction or trying to save before completing all fields. Make sure that the icon is the red exclamation point as shown below:

 Please enter sold-to party or ship-to party

Run your process – it should Fail and exit execution after Step 3.

22. In SAP, trigger a warning status. For example, by entering a date in the past.

 Date is in the past: Please check

Run your process – it should press Enter, which should clear the Warning message. Other warning messages may appear, but eventually the process will exit execution after clearing all warnings. Ultimately the process may fail if an failure displays.

23. In SAP, trigger a success status. For example, by successfully saving a document.

 ERP Standard Order 38964 has been saved

Run your process – it should Pass and exit execution after Step 2.

24. In SAP, type in a nonsense transaction – your name for example.

 Transaction CYNTHIA does not exist

Run your process – it should Pass and exit execution after Step 2. This is an example of a status that SAP considers successful but your process may not. Your process will need an additional verification – such as verifying the text of the message.

Handling Dynamic Windows

The purpose of this section is to show how to work with dynamic report screens in SAP GUI. These types of screens are displayed throughout SAP. Generally, the window name will be SAPMSSY0:0120.

The FPL9 transaction allows us to see the same information in a dynamic report and ALV Grid so the steps can be compared.

TCode: **FPL9**

ALV Grid Layout: **Unchecked** for Dynamic window,
Checked for ALV Grid

Note: Do not be concerned if the Application Version is not the same in your Certify database, as long as the Window and Object work correctly.

Notes:

- ALV Grids can be searched by column/row
- Dynamic screens are usually all labels. Although the basic screen is included in Certify, each specific use of the dynamic screen should not be learned and imported as maps since the contents will change for each transaction and execution.
- SAP Learn views are shown to illustrate the use of the screen. You do not need to use SAP Learn to create the steps.

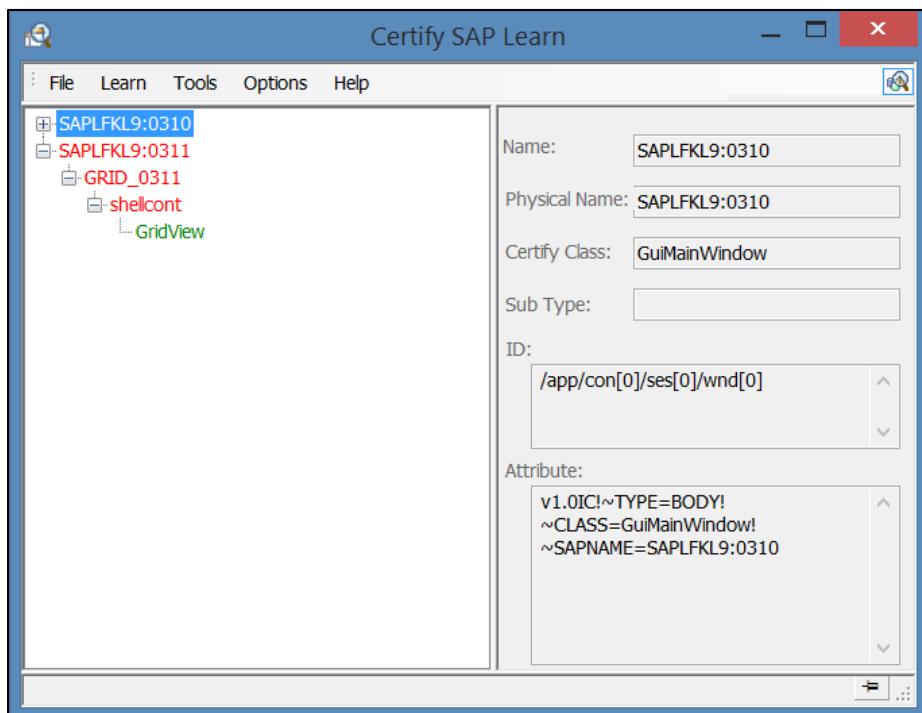
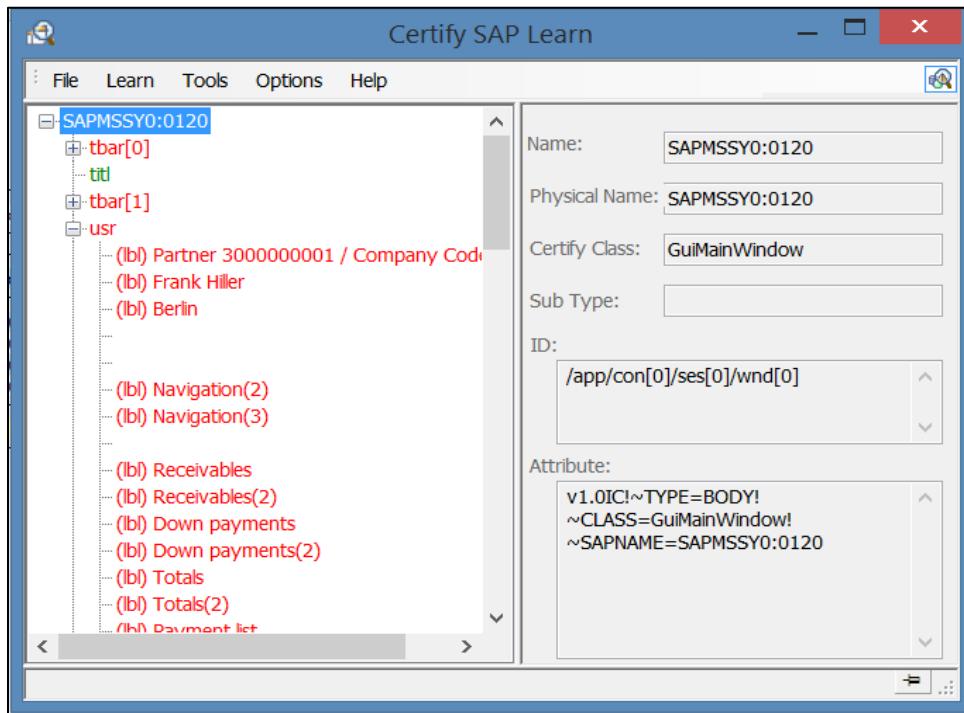
The dynamic, non-ALV Grid display:

This screenshot shows a dynamic grid display for receivables. At the top, there's a navigation bar with links like Account Balance, Edit, Goto, Settings, Environment, System, and Help. Below that is a toolbar with various icons. The main title is "Account Display: Basic List". Underneath, it shows partner information: Partner 3000000001 / Company Code 1000, Frank Hiller, Berlin. A "Navigation" button is present. Below this is a tab bar with "Receivables" (highlighted in green), Down payments, Totals, Payment list, and Chronology. The main area contains a table with columns: Doc. Number, Due, MTns, STns, Name, Clrg. Doc., and Amount. The data rows are: 110000000001 (Due 06/07/2006, Amount 996.00-), 1000000011 (Due 03/24/2006, Amount 996.00), 1000000012 (Due 03/22/2006, Amount 1,601.00), and 1000000014 (Due 04/03/2006, Amount 222.00). A summary row at the bottom shows Receivables EUR 1,823.00.

The ALV Grid display:

This screenshot shows an ALV grid display for receivables. The layout is similar to the previous one, with a navigation bar, toolbar, and "Account Display: Basic List" title. It includes a "Business Partner" search field with value 3000000001. The main area features a "Receivables and Credit Memos" section with a toolbar below it. A table displays transaction details: Tra..., Doc. No., Net Date, MTns, STr., Description, Clearing doc., and Amount. The data rows are identical to the ones in the previous screenshot: 110000000001 (Due 06/07/2006, Amount 996.00-), 1000000011 (Due 03/24/2006, Amount 996.00), 1000000012 (Due 03/22/2006, Amount 1,601.00), and 1000000014 (Due 04/03/2006, Amount 222.00).

The SAP Learn views of these screens:



EXERCISE 4.2 — Handling Dynamic Windows

This exercise walks through the steps to work with dynamic windows and how they compare with ALV grids.

Step	LiveTouch	Application Version	Window	Object	Action
1	okcd	Parameters			[Input]
			Value		/nFPL9
			Instance (row)		1
2	Business partn.	Parameters			[Input]
			Value to input		3000000001
			Key		None
			Wait For Result		Checked
			Instance (row)		1
3	Contract Acct	Parameters			[Input]
			Value to input		100000
			Key		None
			Wait For Result		Checked
			Instance (row)		1
4	Line Layout	Parameters			[Select]
			Item		FI-CA Standard Line Layout
			Criteria		Equals
			Search By Key		Unchecked
			Item Instance		1
			Instance (row)		1
5	ALV Grid	Parameters			[Set]
			State		Unchecked
			Instance (row)		1
6	Enter	Parameters			[Press]
			Type		Press
			Instance (row)		1
7		System	System	Execution	Comment

Step	LiveTouch	Application Version	Window	Object	Action
		Parameters	Comment		Store the Incoming Payment document number in the variable Payment Document Number
8		SAP Core 1.0	SAP Main	SAP Main	Store Report Text
		Parameters	Variable		(V) Payment Document Number
			Row Anchor Text		Incoming Payment
			Row Anchor Search Criteria		Equals
			Row Anchor 2		
			Search Criteria for Row Anchor 2		Equals
			Row Anchor 3		
			Search Criteria for Row Anchor 3		
			Row Anchor Instance		0
			Column Anchor Text		Doc. Number
			Column Anchor Search Criteria		Equals
			Column Anchor Instance		0
9		System	System	Execution	Comment
		Parameters	Comment		Use set reference row and set relative object to select the second document listed
10		SAP Core 1.0	SAP Main	SAP Main	Set Reference Row
		Parameters	Start Search From Current Page		Checked
			Vertical Alignment (in Pixels)		0
			Row Instance (number, last)		1
			Set Vertical Scroll Position		No

Step	LiveTouch	Application Version	Window	Object	Action
			Set Horizontal Scroll Position		No
			Control-1 Property		Text
			Control-1 Property Value		Doc. Number
			Control-1 Criteria		Contains
			Control-2 Property		Class
			Control-2 Property Value		
			Control-2 Criteria		Contains
			Control-2 Is Required		Checked
			Control-3 Property		Class
			Control-3 Property Value		
			Control-3 Criteria		Contains
			Control-3 Is Required		Checked
			Control-4 Property		Class
			Control-4 Property Value		
			Control-4 Is Required		Checked
			Control-4 Criteria		Contains
11		SAP Core 1.0	SAP Main	SAP Main	Set Relative Object
		Parameters	Navigation Direction		Down
			Number of objects to move		2
			Alignment Threshold for Vertical and Horizontal Move		0
			Count blank labels		Checked
12		System	System	Execution	Comment

Step	LiveTouch	Application Version	Window	Object	Action
		Parameters	Comment		Need to check "Count blank labels" since the stoplight doesn't have visible text
13		SAP Core 1.0	SAP Main	SAP Main	Set Relative Object
		Parameters	Navigation Direction		Left
			Number of objects to move		1
			Alignment Threshold for Vertical and Horizontal Move		0
			Count blank labels		Checked
14		System	System	Execution	Comment
		Parameters	Comment		One way to verify document status is with the IconName: - S_TL_R - stop light red - S_TL_G - stop light green
15		SAP Core 1.0	SAP Main	(CustomLabel)	Verify Property
		Parameters	Property		IconName
			Value		S_TL_R
			Criteria		Is Not Equal To
			Instance (row)		1
16		System	System	Execution	Comment
		Parameters	Comment		Another way to verify document status is with the Tooltip: - Receivable open and due - Receivable paid
17		SAP Core 1.0	SAP Main	(CustomLabel)	Verify Property
		Parameters	Property		Tooltip
			Value		Receivable paid
			Criteria		Is Equal To
			Instance (row)		1

Step	LiveTouch	Application Version	Window	Object	Action
18		SAP Core 1.0	SAP Main	SAP Main	Set Relative Object
		Parameters	Navigation Direction		Right
			Number of objects to move		1
			Alignment Threshold for Vertical and Horizontal Move		0
			Count blank labels		Checked
19		System	System	Execution	Comment
		Parameters	Comment		Display the document by double clicking on it
20		SAP Core 1.0	SAP Main	(GuiLabel)	[Click]
		Parameters	Type		Double (F4)
			Set Cursor Position		0
			Instance (row)		1
21		System	System	Execution	Comment
		Parameters	Comment		Now work with the ALV Grid
22	okcd				[Input]
		Parameters	Value		/nFPL9
			Instance (row)		1
23	Business partn.				[Input]
		Parameters	Value to input		3000000001
			Key		None
			Wait For Result		Checked
			Instance (row)		1
24	Contract Acct				[Input]
		Parameters	Value to input		100000
			Key		None
			Wait For Result		Checked
			Instance (row)		1
25	Line Layout				[Select]

Step	LiveTouch	Application Version	Window	Object	Action
		Parameters	Item		FI-CA Standard Line Layout
			Criteria		Equals
			Search By Key		Unchecked
			Item Instance		1
			Instance (row)		1
26	ALV Grid	Parameters			[Set]
			State		Checked
			Instance (row)		1
27	Enter				[Press]
		Parameters	Type		Press
			Instance (row)		1
28		System	System	Execution	Comment
		Parameters	Comment		Store the Incoming Payment document number in the variable Payment Document Number
29	GridView	Parameters			[Find Row]
			Variable		(V) _Row
			Column 1 (number or caption)		Description
			Search Property1		Text
			Criteria		Equals
			Value 1		Incoming Payment
			Column 2		
			Search Property2		Text
			Criteria		Equals
			Value 2		
			Column 3		
			Search Property3		Text
			Criteria		Equals
			Value 3		
			Column 4		
			Search Property4		Text

Step	LiveTouch	Application Version	Window	Object	Action
			Criteria		Equals
			Value 4		
			RowIndex		
30		Parameters			[Store Cell]
			Variable		(V) Payment Document Number
			Property		Text
			Column (number or caption)		Doc. No.
			Row number		(V) _Row
31		System	System	Execution	Comment
		Parameters	Comment		Verify document status with the Tooltip: - Receivable open and due - Receivable paid
32	GridView				[Verify Cell]
		Parameters	Property		Tooltip
			Row number		2
			Column (number or caption)		Traffic Light
			Criteria		Is Equal To
			Value		Receivable paid
33		System	System	Execution	Comment
		Parameters	Comment		Display the document by double clicking on it
34	GridView				[Click Cell]
		Parameters	Row number		2
			Column (number or caption)		Doc. No.
			Type		Double
			Key (optional)		Enter (No Result)

Lesson 5

SAP Classes and Actions

Overview

In this lesson, you will learn how to perform various tasks in Worksoft Certify.

General Classes and Actions

A **Class** is a description of a category of objects. A class defines how to recognize objects and the actions that can be executed against the objects. Each Certify interface has a set of classes and associated actions already defined.

An **Action** is an individual activity or operation that can be automatically executed on a class. Some actions simulate an operation and some actions check to see if a condition is true. Each action can have input parameters and output parameters. Actions are invoked when you execute a process, and if parameters are required, prompts you for the parameters. For example, entering data to a TextBox requires a parameter for the data value. Each class and action combination invokes a particular function in the interface library, and the parameters are passed into the function during execution as arguments.

Some actions simulate an operation like press, while others check to see if a condition is true such as enabled, focused, and visible.

All classes in the SAP GUI interface inherit the following generic actions:

Class	Action
Generic	Click
	Find
	Select Context Menu
	Set Property
	Store Property
	Verify Property

Reference: For a complete list of the SAP Classes and Actions, go to [Certify Help → Classes and Actions → SAP Classes and Actions](#).

Or, on the Navigation toolbar, click **Interfaces**, then select **SAP**. **View this area only.** Modifying this list will cause problems in Certify.

SAP Classes and Actions

As well as having generic actions, some classes have actions that are unique to the class. The table below lists the unique actions for each class.

Control	Description	Actions
GuiBox	A simple frame on the screen. The items inside the frame are not children of the box.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiButton	Buttons on the screen.	[Press] Click Find Select Context Menu Set Property Store Property Verify Property
GuiCheckBox	Check boxes on the screen.	[Set] [Verify] Find Select Context Menu Set Property Store Property Verify Property
GuiComboBox	Combo boxes on the screen.	[Select] [Store] [Verify] Find Select Context Menu Set Property Store Property Verify Property
GuiContainerShell	A wrapper for a set of GUI Shell objects.	Click Find Select Context Menu Set Property Store Property Verify Property

Control	Description	Actions
GuiCTextField	A text field with a button in it that can be pressed to bring up a list of options, a dialog, or other input help.	[Input] [Store] [Verify] Click Find Select Context Menu Set Property Store Property Verify Property
GuiCtrlAPOGrid	APO Grid Control	Click Cell Col Caption For Number Col Number For Caption Find Cell Find Cell in Column Find Cell in Row Find Column Have Cells Find Row Have Cells Input Cell Row Caption For Number Row Number For Caption Select All Cells Select Column Select Context Menu Select Row Store Cell Property Store Grid Property Verify Cell Property Verify Grid Property
GuiCtrlBarChart	Bar Chart	Store Property Verify Property
GuiCtrlCalendar	Select a date from the calendar and then store or verify the date. Similar to the Date Picker.	[Select Date] [Store Date] [Verify Date] Click Find Select Context Menu Set Property Store Property Verify Property

Control	Description	Actions
GuiCtrlGridView	Implemented as an ActiveX. Similar to a table control but has more features.	[Click Cell] [Find Row] [Input Cell] [Press ToolBar] [Select Row] [Store Cell] [Verify Cell] Click Find Find Column Select All Rows Select Context Menu Select Menu Set Property Store Property Verify Property
GuiCtrlHTMLViewer	See Scripts class description.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiCtrlPicture	Picture	[Click] Find Select Context Menu Set Property Store Property Verify Property
GuiCtrlTextEdit	A text field with a button in it that can be pressed to bring up a list of options, a dialog, or other input help.	[Input] [Store] [Verify] Click Find Select Context Menu Set Property Store Property Verify Property
GuiCtrlToolBar	Toolbar buttons on the screen.	[Press] Find Select Context Menu Set Property Store Property Verify Property

Control	Description	Actions
GuiCtrlTree	Tree view on the screen	[Click Cell] [Select] [Verify Node] Click Click Column Header Find PressKey Search Node Global Select Child Node Select Context Menu Select Node Select Parent Node Select Sibling Column Select Sibling Node Set Property Store Node Property Store Node Text Store Property Verify Node Property Verify Node Text Verify Property
GuiDockShell	A wrapper for the docker control.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiGOSShell	A child of the Titlebar that contains another shell. An example is a toolbar control.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiLabel	Labels on the screen. By default, labels are not enabled (imported) in the learn process unless they are explicitly activated.	[Click] [SendVKey] Find Select Context Menu Set Property Store Property Verify Property

Control	Description	Actions
GuiOkCodeField	The OkCodeField is at the top of every screen and allows a transaction or program name to be entered instead of using the menu structure. It is treated as a normal text field except the <Enter> key is automatically sent when a value is input in the field.	[Input] [Store] [Verify] Click Find Select Context Menu Set Property Store Property Verify Property
GuiPasswordField	A text field that is similar to the GUITextField object, but it is different because the Text property cannot be read for the password field.	[Input] Click Find Select Context Menu Set Property Store Property Verify Property
GuiRadioButton	Radio buttons on the screen.	[Click] [Verify] Find Select Context Menu Set Property Store Property Verify Property
GuiScrollContainer	Represents scrollable subscreens. A subscreen may be scrollable without actually having a scrollbar.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiShell	An abstract object whose interface is supported by all the controls.	[Click] Find Select Context Menu Set Property Store Property Verify Property
GuiSimpleContainer	Represents non-scrollable subscreens. A simple container does not have any functionality apart from the inherited interfaces.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiSplitterShell	A shell that, by nature, is splitting.	Click Find Select Context Menu Set Property Store Property Verify Property

Control	Description	Actions
GuiStatusBar	Represents the message displaying part of the status bar on the bottom of the SAP GUI window.	[Store Parameter] [Verify Parameter] Click Find Select Context Menu Set Property Store Property Verify Property
GuiStatusPane	Represents the status pane on the bottom of the SAP GUI window.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiTab	Tabs on the screen.	[Click] Find Select Context Menu Set Property Store Property Verify Property
GuiTableControl	Provides all the basic functionality available in most grid type controls.	[Click Cell] [Find Row] [Input Cell] [Select Row] [Store Cell] [Verify Cell] Click Display Settings Find Find Column Input Cells Page Down Page Up Search Empty Row Search Row Select Column Select Context Menu Select Menu By Id SendVKey Set Property Store Column Title Store Property UnSelect Row Verify Column Title Verify Property

Control	Description	Actions
GuiTabStrip	A container whose children are of type GUI Tab.	Click Find Select Context Menu Set Property Store Property Verify Property
GuiTextField	Text fields on the screen.	[Input] [Store] [Verify] Click Find Select Context Menu Set Property Store Property Verify Property
GuiTitleBar	Title bar of the screen.	[Verify] Click Find Select Context Menu Set Property Store Property Verify Property
Object	Used for dynamic objects on the screen.	Click Find Select Context Menu Set Property Store Property Verify Property
Scripting	Enables the use of the scripting facility provided by the SAP GUI Scripting API. It is used for those exceptional cases where the Certify classes and actions do not provide the needed functionality.	[Open] [Play] [Search Replace] [Set Variable]

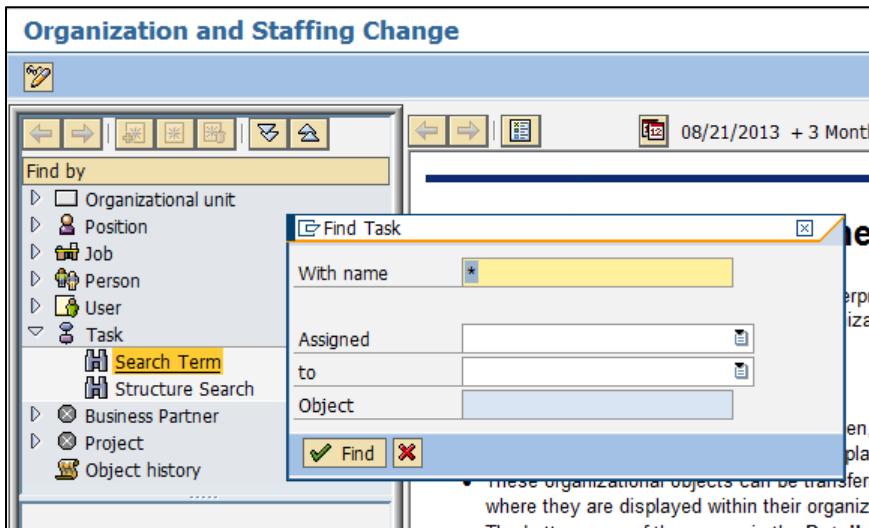
Control	Description	Actions
Window	Screen menus and Vkey functions	<ul style="list-style-type: none"> [Select Menu] [SendVKey] Activate Click Click Reference Text Click Report Text Close Popup Context Find Get Client Ignore Window Context Launch SAP Page Scroll Reset Defaults Select Menu By Id Set Client Set Instance Set Reference Row Set Relative Object Set Session Set Timeout Start Logging Start Performance Capture Stop Logging Stop Performance Capture Store Property Store Report Text Store Window Caption Verify Client Verify Property Verify Report Text Verify Window Caption Wait

Common Scenarios/Examples

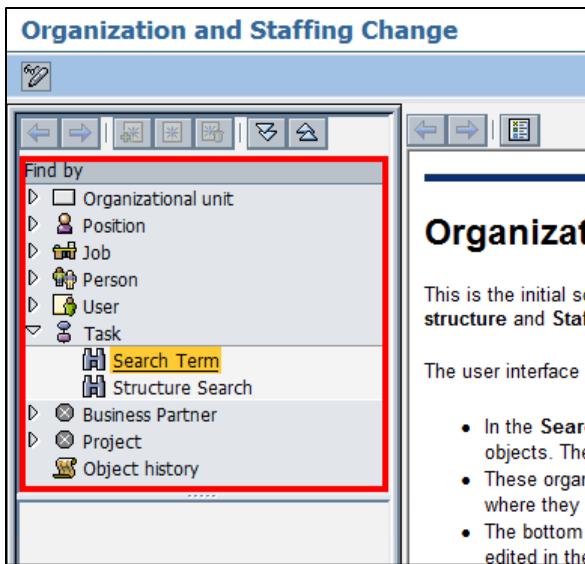
Below are some sample of using specific SAP classes and actions.

SAP Trees and Nodes

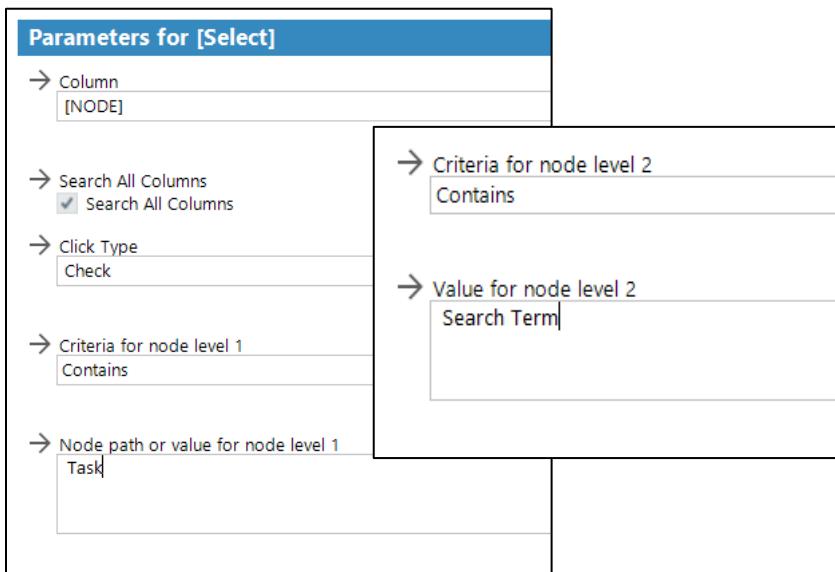
Scenario 1: you want to open a node from an SAP tree. This example uses transaction PPOME and we want to open the **Search Term** window in the **Task** node.



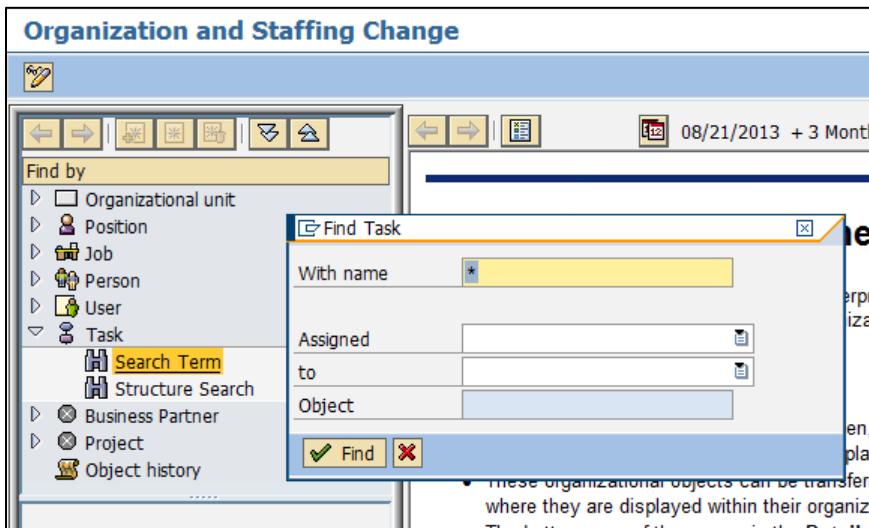
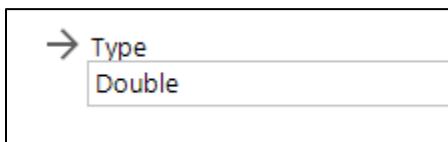
LiveTouch will find the tree, but not the node:



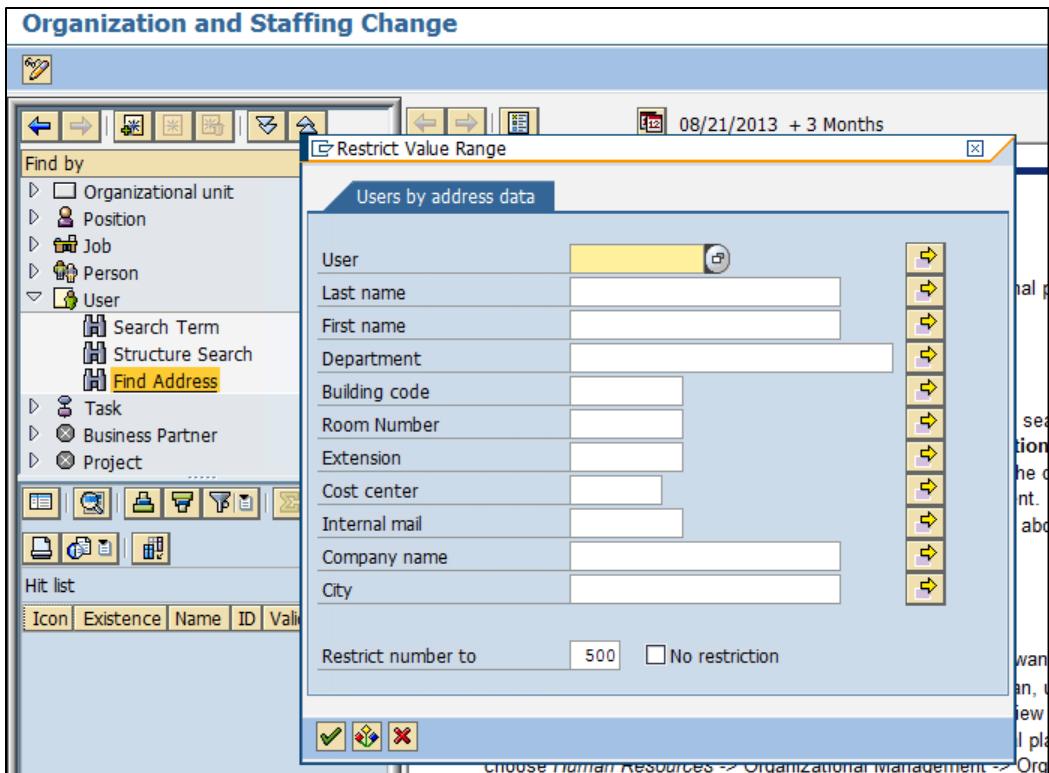
In this case, since most of the top-level nodes have a child-node called **Search Term**, we'll use the default Select Node action and tell it to look in the **Task** node. By setting the **Level 1 Node Value** to 'Task' and the **Level 2 Node Value** to 'Search Term' in the Parameters tab, we tell Certify exactly where to find the node.



And by setting the **Action Type** parameter to 'Double', Certify opens the node.

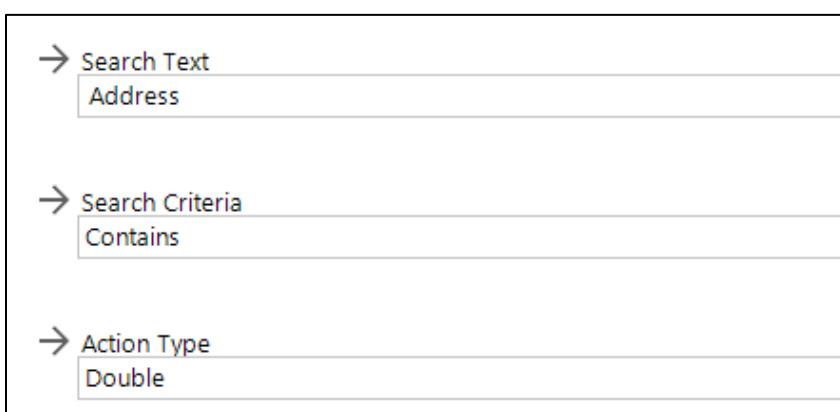


Scenario 1a: you want to open a node from an SAP tree. This example uses transaction PPOME and we want to open the **Find Address** window in the **User** node.

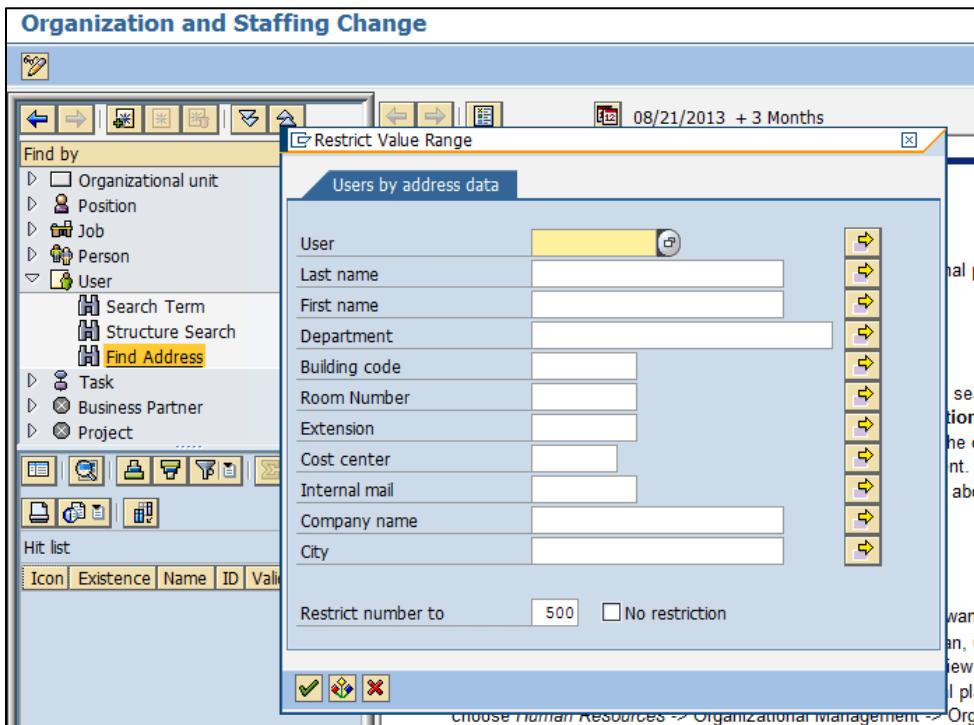


In this example, **Find Address** is a unique term within the tree so we can use the **Search Global Node** action to find and open this node.

After selecting the tree with LiveTouch, select the action **Search Global Node** and in the **Parameters** tab, set the **Search Text** to 'Address', **Search Criteria** to 'Contains' and **Action Type** to 'Double'.

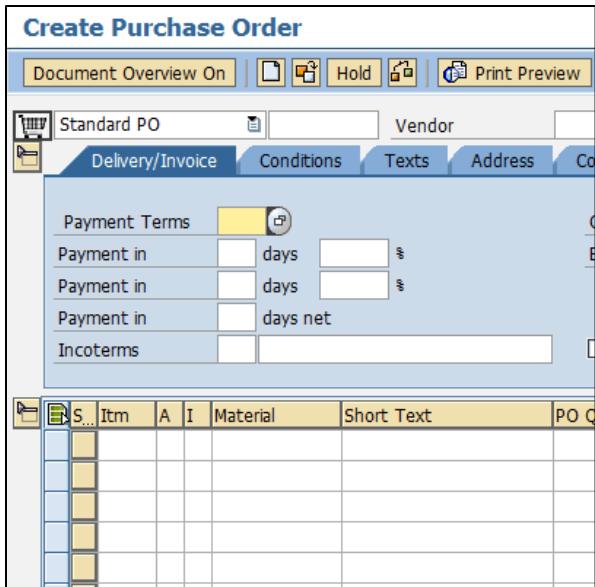


Executing the step finds the unique child-node and opens the window

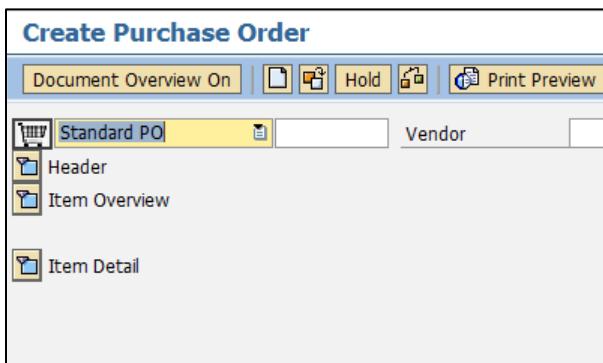


Sending Keys to SAP

Scenario 1: you want to create a purchase order using transaction ME21N. The issue here is that you may not know which of the three sections of the Create Purchase Order page are expanded, allowing Certify to find the objects you need.

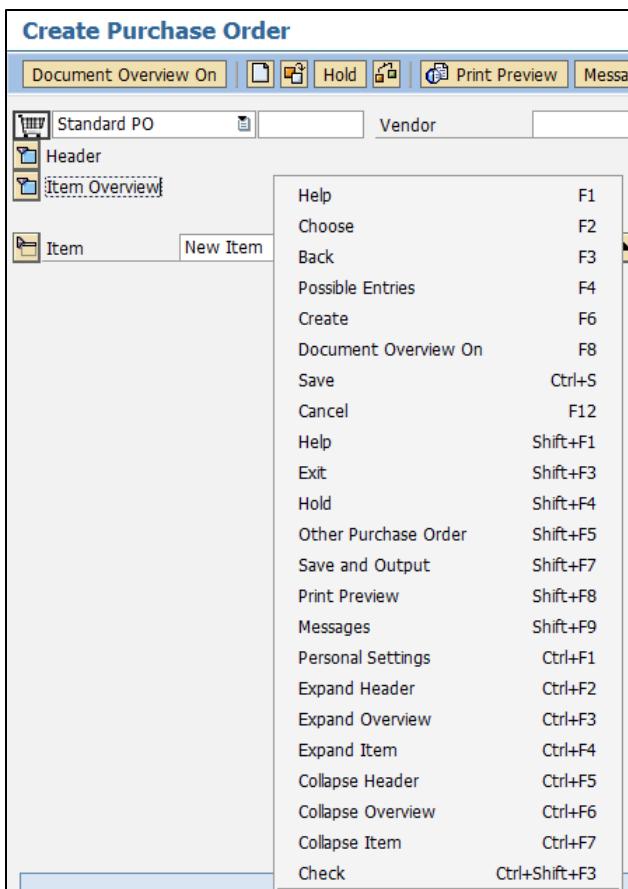


or



One possible solution for this is to have Certify send the appropriate keystrokes to SAP to make sure the areas are expanded before you start creating the PO. To do this:

1. Find the appropriate key combination by hovering over the icon with the desired functionality or by right-clicking anywhere on the screen.



2. Use Certify's SAP Core functionality of the SAP Main object with the [SendVKey] action to send the appropriate keys to the SAP window as shown here in steps 3-5.

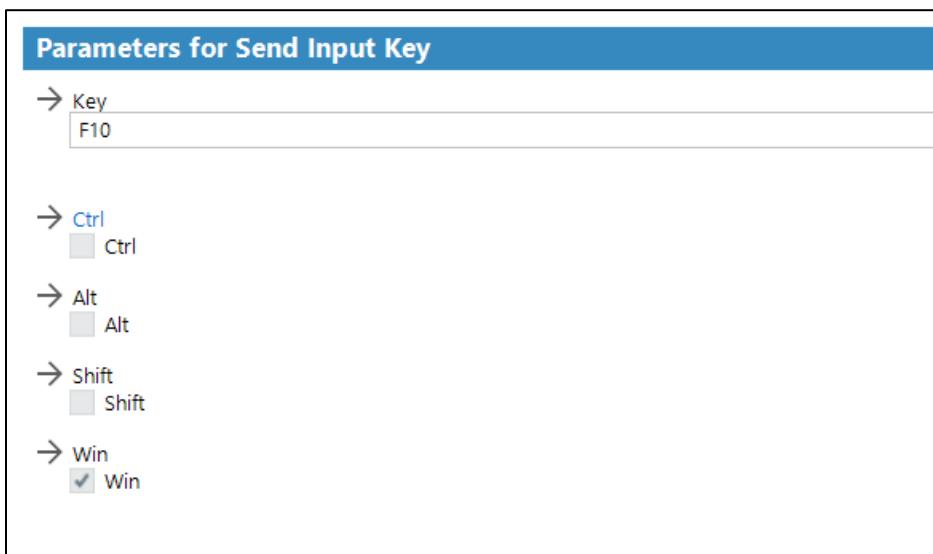
Step#	Application	Version	Window	Object	Action	Narrative	On True	On False
2	System 1.0		System	Execution	Comment	"Expand all sections on ME21N screen.."	Continue	Continue
3	SAP Core 1.0		SAP Main	SAP Main	[SendVKey]	SendVKey "Ctrl+F2" to SAP Main Window while Wait For Result is "True"	Continue	Continue
4	SAP Core 1.0		SAP Main	SAP Main	[SendVKey]	SendVKey "Ctrl+F3" to SAP Main Window while Wait For Result is "True"	Continue	Continue
5	SAP Core 1.0		SAP Main	SAP Main	[SendVKey]	SendVKey "Ctrl+F5" to SAP Main Window while Wait For Result is "True"	Continue	Continue

As these key combinations do nothing when the section is already expanded, it's okay to use the default On True and On False values.

3. Another way to send keys is to use Certify's system functionality. For example, to send the **Ctrl+Shift+F10** key to a window:

5	System 1.0	System	System	Send Input Key	Send Input Key "F10" Modifier "False" "False" "False" "True"	Continue	Continue
---	------------	--------	--------	----------------	---	----------	----------

4. This option uses standard Windows Sendkey functionality – {key} preceded by Ctrl (^), Shift (+), and/or Alt (%), and lets you specify the window by caption and timeout time.

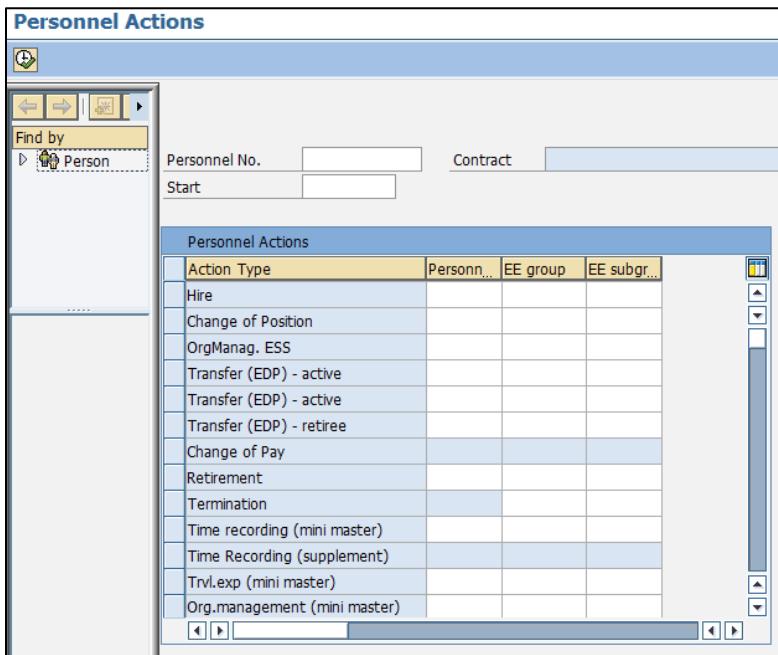


Note: Use Certify's system option to send keys to non-SAP popup windows such as **File Save As**.

Either of these send key options can be used wherever key codes can be used, such as **Ctrl-S** for saving and **F3** to go back a screen.

Common SAP Table Scenarios

Scenario 1: you want to add text to a specific row in a table. This example uses transaction PA40 where you want to add text to the **Personnel** column in the **Retirement** row.



Knowing that the arrangement of rows and columns could change, you want a process that will be able to find the right location if it does. Certify uses row numbers to navigate through tables. So your best option is to find number of the row containing **Retirement** and then putting your text at the intersecting cell of that row and the **Personnel** column.

In this example, use LiveTouch to find the table and then use the **[Find Row]** action to get the row number containing Retirement.

```

    → Column 1 (number or caption)
      Action Type

    → Criteria1
      Equals

    → Value 1
      Retirement
  
```

Be sure to use a numeric variable to store the row number.

The screenshot shows the SAP Fiori launchpad interface. A new variable is being created with the following details:

- Name:** X_Row
- Type:** Numeric
- Format:** [Empty field]

Then **Insert Step Below** and use the **[Input Cell]** action to put the text into the correct cell.

The screenshot shows the configuration of an [Input Cell] action step with the following parameters:

- Value:** 1000
- Row Number:** X_Row
- Column (number or caption):** Personal area

With these steps in place...

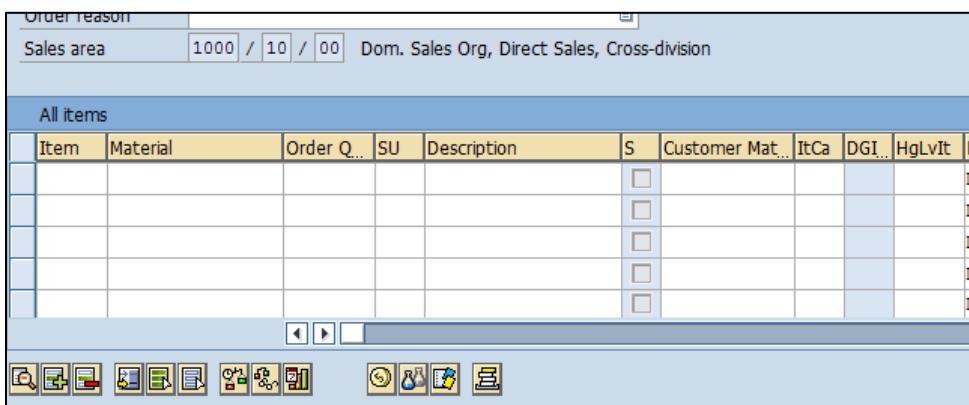
21	Human Resources 1.0	SAPMP50A	SAPMP50ATC_ME	[Find Row] NU_EVENT	Find instance "0" of row in Table having Column1 "Action Type" "Equals" "Retirement", Column2 "" "Equals" "", Column3 "" "Equals" "", Column4 ""	Continue	Continue
22	Human Resources 1.0	SAPMP50A	SAPMP50ATC_ME	[Input Cell] NU_EVENT	Input the value "1000" in Row: N_Row, Column: "Personal area"	Continue	Continue

...your processes will input the text in the right location every time.

The screenshot shows the SAP Fiori launchpad interface after the process has been executed. The table under "Personnel Actions" shows the following data:

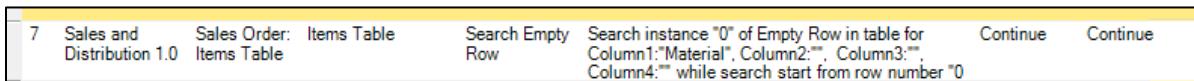
Action Type	Personn...	EE group	EE subgr...
Retirement	1000		
Termination			
Time recording (mini master)			
Time Recording (supplement)			
Trvl.exp (mini master)			
Org.management (mini master)			
Hire Applicant			

Scenario 2: you want to add data into the first blank row of a table. This example uses the Material table of transaction VA01.



Note: This example was used for the Order to Cash process as **VA01_CreateStandardOrder_C_Enter Materials** in Lesson 2 of the *Certify for SAP Training Guide*.

After LiveTouching the Material table, change the Action to **Search Empty Row**. And set the option for **Column 1** in the Parameters tab to 'Material'. This tells Certify to search the table for the first row where the Material column contains no data.



This process works for all tables where data needs to be entered into a blank row.

Section 3

Lesson 1

Introduction to Worksoft Certify for HTML

Overview

This lesson discusses the basics of using Certify to automate business processes in HTML.

Objectives

After completing this lesson, you will be able to:

- ▶ Determine the configuration needed for browsers used by Certify.
- ▶ Create an Application and Project to store your training maps and processes.

Configuring Your HTML Environment

Before you can build your processes in Certify, you must set up your HTML environment to work with Worksoft Certify.

For any additional requirements, refer to the *Worksoft Certify Installation Guide*.



LiveTouch and Learn are not supported against Chrome or Firefox.

Always refer to the Worksoft Community site for the latest information on the HTML interface.

You will use Windows Internet Explorer to build your processes. The processes can be executed on Chrome or Firefox.

Windows Internet Explorer Configuration

Windows Internet Explorer configuration was covered in the Certify Basics course.

Firefox and Chrome Configuration

To enable browser support you will need to enable the extension corresponding to the browser you wish to use for execution. The Certify Browser Extension Package can be downloaded if it was not included in your Certify installation. The extension files are in the \\Worksoft\\Certify\\Interface Client\\Worksoft\\wsTest\\HTML directory.

- ▶ Worksoft Automation Extension.crx (For Chrome)
- ▶ Worksoft Automation Extension.xpi (For Firefox)

Follow the instructions in the **Technical notes for Certify Browser Extension** to install the extension(s).

Test the installation by creating a simple Certify Web interface process and ensure that the process can be run successfully against MS Internet Explorer 9 or above. Then run the same test against Chrome or Firefox after installing the extensions.

In order to support single content, the attribute strings for the object might need to use a more lax criteria for object identification. Single content in this context refers to the ability to execute the same Certify process against different browsers, e.g., IE, Chrome and Firefox. While there are many limitations, this allows for multiple browsers to be tested.

Web interface optimizations are not supported in the Chrome or Firefox browser. Web pages also must be hosted on a web server.

Naming Process Folders

As discussed in the Certify Basics class, an organized, well-structured folder structure is critical to finding and reusing your automated processes. **As a best practice, you should plan on organizing your processes into the following folder structure:**

- ▶ **Organization Name** — While the official technical object name in Certify is “Project” this folder is the highest level folder and should reflect the name of the organization. This folder contains all of the folders for the organization (in our example on the next page that might be **Worksoft**).
- ▶ **Project Name** — Contains folders for the project, for example: Project Pronto. These folders would contain all of the applications and updates associated with a specific project (in our example on the next page that might be **Project Pronto**).
- ▶ **Application Version Name** — Contains folders for the areas of integration tests for a particular application. These folders would contain the processes for a particular application (in our example on the next page that might be **Certify 9.0.3**).
- ▶ **Integration Tests** — Contains folders for the areas of integration tests for Web, for example: Online Enrollment. These folders would contain the root or parent processes (in our example on the next page that might be **OnlineEnrollment_CustomerCreate_VerifyInvoice**).
- ▶ **Transactional Tests** — Contains folders for transactions, named by web application. This structure provides convenience for finding transactional processes that you may need to use or copy.
- ▶ **Utilities** — Smaller single-function processes used to support more complex processes. An example would be **UTL_CalculateTodayPlus30Days** – a utility that calculates date. Includes sub-folders like Date Utilities and Math Utilities that contain appropriate utility processes.
- ▶ **Sandbox** — Contains folders for each user. **The best practice is to use sub-folders for end to end processes.** For example, for our training processes, the Sandbox folder would look similar to the following:

```
Sandbox folder
  JSmith (user name) folder
    CertifyWebSampleApp_FieldCheck
      CertifyWebSampleApp_Login
      CertifyWebSampleApp.Buttons
      CertifyWebSampleApp_Edits
```

Naming Process Folders (con't)

Once the processes are reviewed and signed off, they should be moved to their respective location. For example:

Worksoft

Project Pronto

Certify 9.0.3

Integration Tests

Online Enrollment

OnlineEnrollment_CustomerCreate_VerifyInvoice

Transactional Tests

Online Enrollment

OnlineEnrollment_CustomerInformation

OnlineEnrollment_AddressInformation

Utilities

Date Utilities

UTL_CalculateTodayPlus30Days

Sandbox

JSmith (user)

OnlineEnrollment_CustomerCreate_VerifyInvoice



Lesson Summary

You've completed the **Introduction to Certify for HTML** lesson.

Key points to remember:

- ▶ Certify supports multiple browsers for execution.
- ▶ Certify supports Internet Explorer for creating processes.
- ▶ Configuration is needed for browsers used by Certify.

Process folders should be well managed to allow easy search on processes.

Lesson 2

Learning Screens and Importing Maps

Overview

In this lesson, you will learn the process for learning HTML screens and how they are imported into Certify.

Objectives

After completing this lesson, you will be able to:

- ▶ Use Certify Web Learn to learn application maps
- ▶ Modify Attributes in Web Learn
- ▶ Save maps and import into Certify projects
- ▶ Use Touch Learn feature

Application Maps Overview

To automate the testing process, Certify has to understand the contents of the application under test. To gather application contents, you use a utility provided with Certify. This utility learns and creates an application map. An application map is an inventory of an applications component, including:

- ▶ Application version — the release or build of an application
- ▶ Window — window, page, or screen
- ▶ GUI elements — object, control, field interface, element, or other class
- ▶ Type of object class
- ▶ Identification parameters _ name, handle, ordinal, index, schema, or other identifying information

The application map serves as the link between the application under test and the Certify processes. Once you have the application map, you save it to a file and import it into a Certify project. From there, you can edit the map, if needed, and begin adding processes, variables, and data to build your automated test.

Because each application map is unique, you can compare a previous map version with new versions of the application. Any changes can be cross-referenced to all affected properties in Certify. For example, if a field or object is modified in a later version, Certify performs impact analysis and identifies all processes that refer to that field or object and modifies all test steps affected by the changes. The test can be re-used without totally rebuilding it for the new application version.

The utility for creating HTML application maps is the [Certify Web Learn utility](#). Certify Web Learn maps the windows and objects used in the test application and store each object type and its attributes in application maps.

An application map file is created for each page within the application and saved in text format in a file with a .map extension. The file contains the learned window and object descriptors (physical name) and their associated attributes (property name and strings).

Creating a Project and Application for Training

Any windows or objects you wish to interact with must be identified so that Certify can work with them. This identification process is known as mapping. You create maps through LiveTouch, WebLearn or by importing a Captured process.

Once one person has mapped a window and objects every Certify user in that Project will have access to the objects. Normally this is ideal – no extra rework when using the same screens.

For training purposes, each student will create a Project and Application that will only be used by that student.

NOTE: Remember to switch back to your normal project after training. It is not easy to move processes between projects. Any processes you want to keep should **NOT** be created in the training project.



EXERCISE 2.1 — Create an Application for Training

In this exercise, you will create an Application.

Step	Action
------	--------

To create a new application and version:

1. In the Navigation pane, click **Applications**.
The Applications window appears.
2. In the Navigation tree, right-click on **Applications** and select **New Application**.
The New Application and Version dialog box appears.
3. Type the following information in the New Application and Version text fields:

Field/Options	Value/Action
Name	_Training_<your initials>
Description	
Version	1.0
Version Description	
Interfaces	Check Web
Shared	Not checked

New Application and Version

Application Details

Name: *
_Training_CLL

Description:

Application Version Details

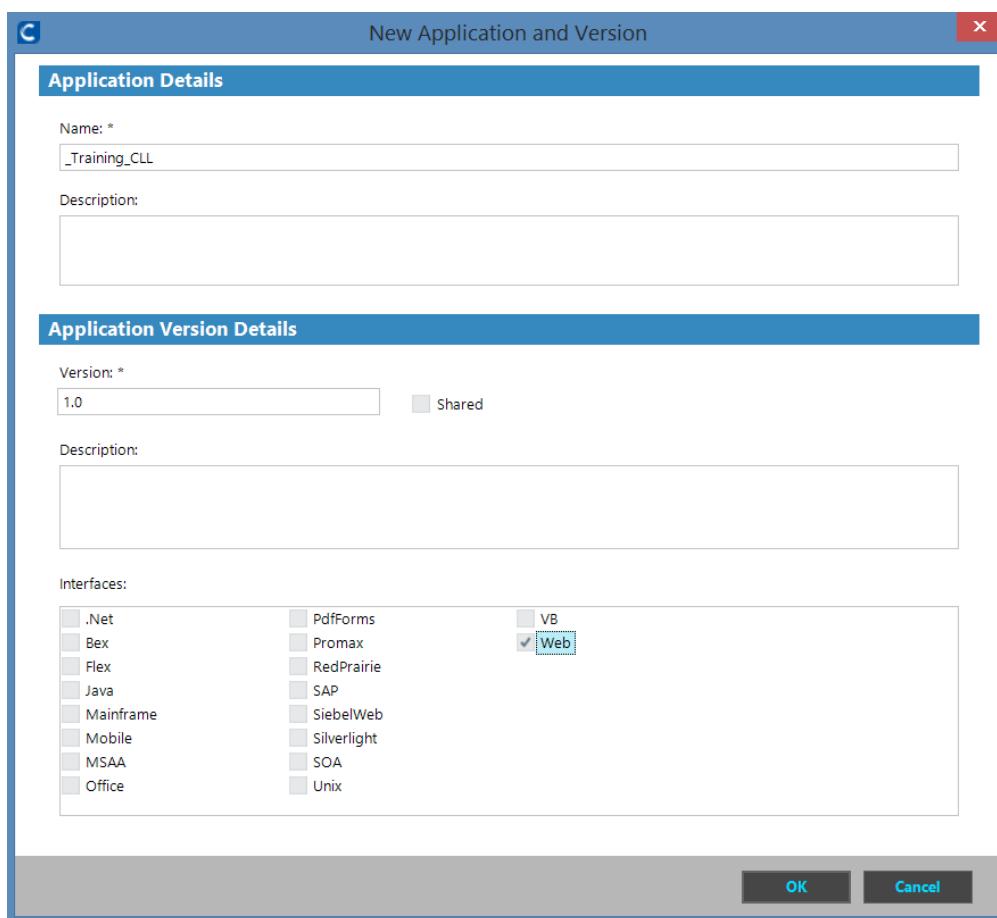
Version: *
1.0 Shared

Description:

Interfaces:

.Net	PdfForms	VB
Bex	Promax	<input checked="" type="checkbox"/> Web
Flex	RedPrairie	
Java	SAP	
Mainframe	SiebelWeb	
Mobile	Silverlight	
MSAA	SOA	
Office	Unix	

OK Cancel



5. Click **OK**.

The new application and version are created.



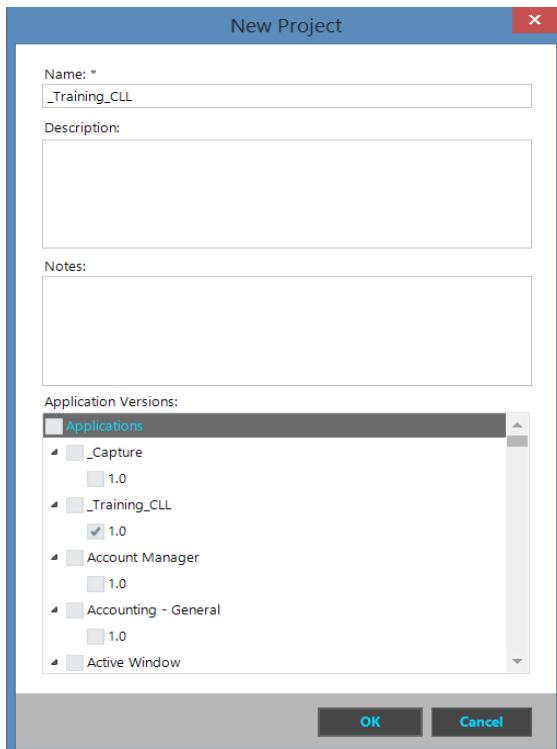
EXERCISE 2.2 — Create a Project for Training

In this exercise, you will create a Project and tie it to your Application.

Step	Action
------	--------

To create a new Project:

1. In the Navigation pane, click **Projects**.
The Projects window appears.
2. In the Navigation tree, right-click on **Projects** and select **New Project**.
The New Project dialog box appears.
3. In the Name field type in the same name as the Application you just created (e.g. _Training_CLL). Normally the project name and application name don't match – but this keeps it organized for training class.
4. In the Application Versions tree, select the checkbox next to the application you just created.



6. Click **OK**.

The new project is created.

7. In the Certify menu, select File > Open Project.
8. Select the project you just created.
9. Click **OK**.

Navigate to Processes, Variables and Data. Note that these areas are empty. Ready for class!

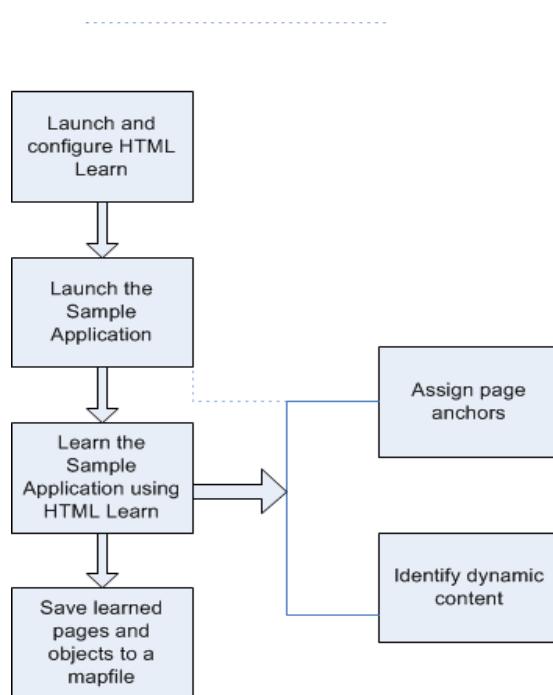
Creating Application Maps

Application maps for HTML are created using the Certify Web Learn utility. In Figure 1, you can see that in addition to learning and saving the map, there are additional steps required.

HTML applications pose a special challenge for mapping because the page contents are often dynamic. Generally within Certify, a page is mapped as a collection of objects, and these objects are learned using the Certify Web Learn utility. Issues may arise, however, if the objects displayed at runtime change due to how the page was developed, available data, or user selections. Certify provides special capabilities to deal with these situations.

While the page title is usually used to denote which set of objects are active, in some cases developers use multiple pages with the same title, then change the objects dynamically in their code. This makes it difficult to know whether the correct page is displayed.

Figure 1 — HTML Learn Process



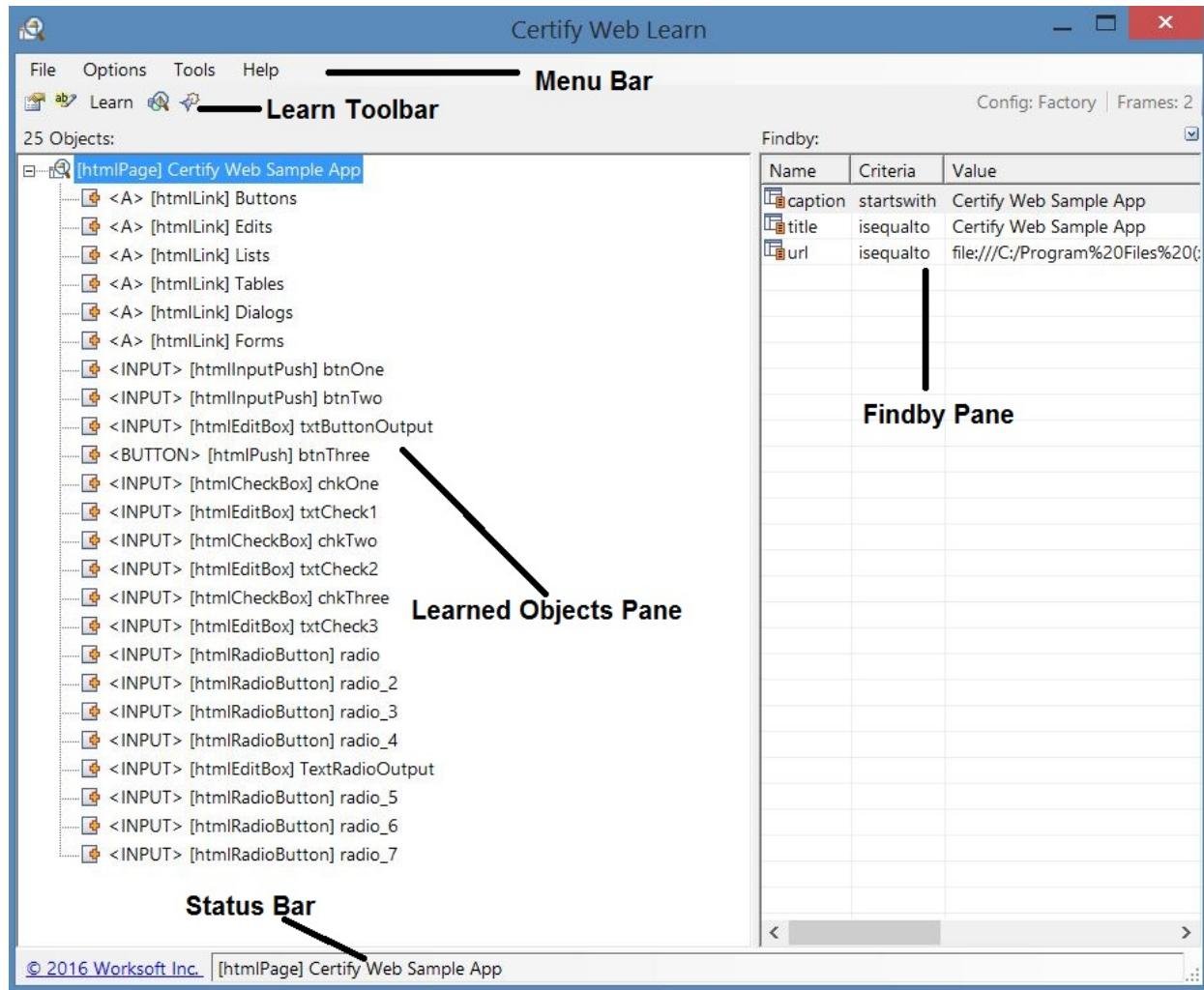
An example of objects that change due to user selection or other conditions would be an insurance policy, where the questions posed to the user depend on options previously selected. For example, if the user elects to add a special endorsement, then extra questions may appear to gather information about the desired coverage amounts. In this case, you can define the objects within the test process instead of the map by using the generic object functionality within Certify.

All of these special features are designed to allow you to use Certify to work efficiently with dynamic Web applications, and they are described in more detail within this lesson.

Certify Web Learn Overview

The Certify Web Learn utility is installed along with the HTML interface. Figure 2 shows the browser with the Account Manager application open and the Certify Web Learn utility displayed. This view occurs after the displayed page has been mapped for the first time. Note the various parts called out in the diagram.

Figure 2 — Application Manager with HTML Learn Displayed



Once the application is mapped, you can modify the hierarchy of objects as needed to make the map more user-friendly.

Learn Menu Bar

The Learn menu bar contains four menus for performing tasks within Certify Web Learn: File, Options, Tools, and Help.

Learned Browser Pane

The browser pane displays the name of the learned application, number of frames and objects, and the configuration file that was used to learn the page.

Learn Toolbar

The Learn toolbar includes four buttons:

Icon	Description
	The Properties button opens the Properties dialog box of a selected object.
	The Highlight in App button highlights an object in a Web application.
	The Hover-Find button allows you to hover over the application and find objects in the Learned Objects tree.
	<p>The Touch Learn button allows you to:</p> <ul style="list-style-type: none">▶ Add only objects that are selected with Touch Learn.▶ Add objects that dynamically appear after a full learn has completed. <p>To turn off this feature, click the Touch Learn button.</p>

Learned Objects Pane

The Learned Objects pane displays the learned objects of the application in a tree structure.

Findby Pane

When a learned object is selected in the tree, the configured recognition attributes appears in the Findby pane. All configured attributes are shown in this list, but those that have no value will not be in the attribute string when the map is saved.

Status Bar

The status bar displays the elapsed time of the learn process and other pertinent information about the object.



EXERCISE 2.3 — Launching the Certify Web Learn Utility and Sample Application

The first step in mapping an application is to open the browser and the Learn utility.

In this exercise, you will launch the Learn utility and the Certify Web Sample App

Step	Action
------	--------

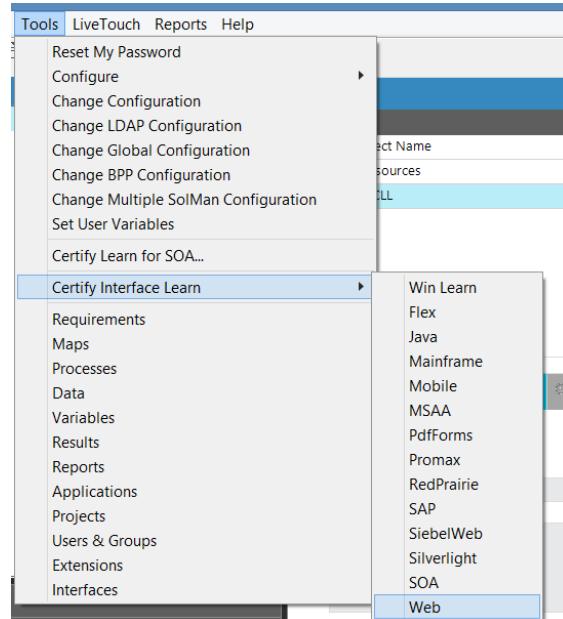
To launch the learn utility:

1. In Certify, on the menu bar, click **Tools**.

A shortcut menu appears

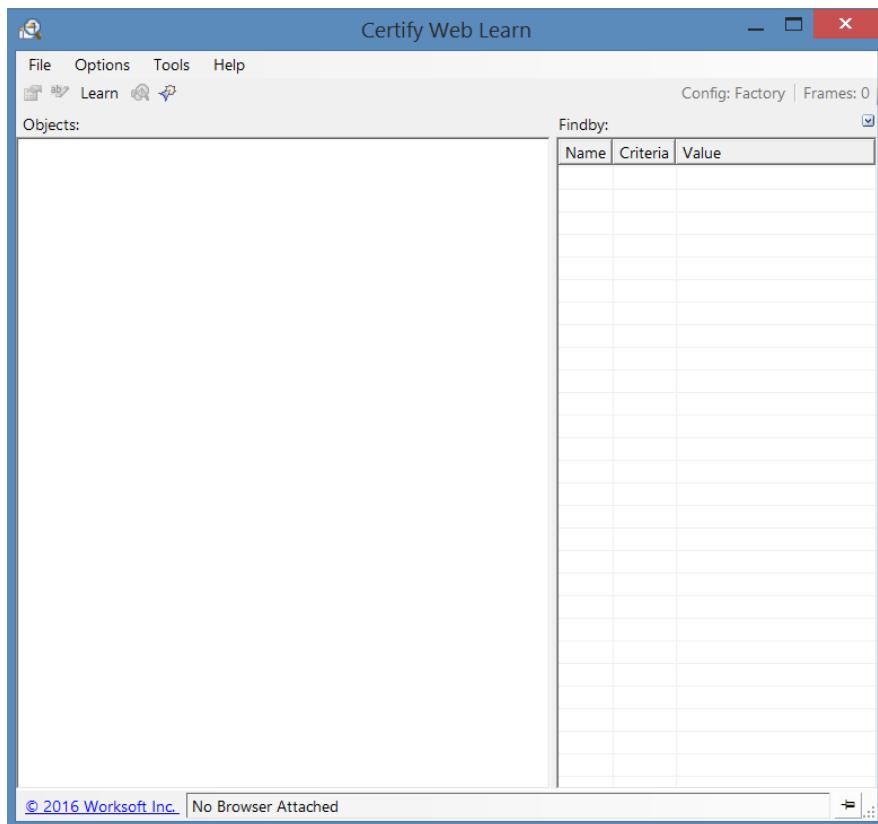
2. Select Certify Interface Learn

A drop-down list appears.



3. Select **Web**.

The Certify Web Learn Utility appears. This window will remain on top of other windows unless you minimize it or select the push pin at the bottom of the screen.



To launch the application:

4. Click **Start** and select All Programs > Worksoft > Sample Applications.
5. Select **CertifyWebSampleApp1**.

Be sure to open this page in Internet Explorer.

The Certify Web Sample App Login page appears.



Certify Web Sample Login

Username:

Password:

Username=admin Password=password

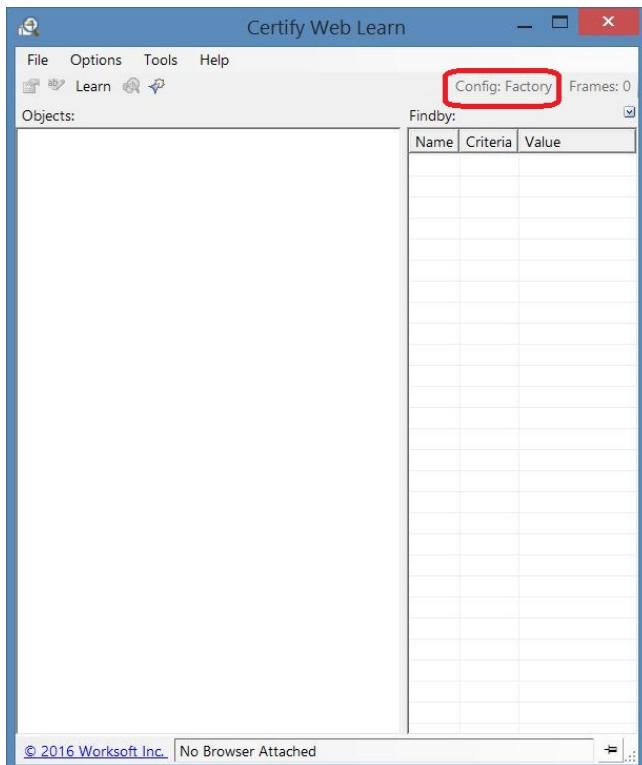
Web Configuration Manager

Web Configuration Manager allows you to change the way objects are identified by Web Live Touch, Web Learn and Capture. Certify Web Learn comes pre-installed with a Factory default configuration. You can create a new custom configuration to work with your applications and choose that as the default. After creating a configuration, it is possible to select it in Certify Web Learn using the File > Set Current Configuration menu choice.



You normally do not need to change the current setting for Web Configuration. If your administrator has created a custom configuration you will see that configuration in Web Learn rather than the standard **Factory** configuration.

The Web Configuration name appears on the right side of the menu bar.



Modifying Attributes in Certify Web Learn

We will use Certify Web Learn as a substitute for Live Touch and create our own windows utilizing Web Learn. Web Learn gives us the flexibility to quickly and easily create a new window, give it a good name, and save the window for importing into Certify. Once the window is in Certify we can return to our process and begin Live Touching all other objects on the screen and Live Touch will automatically add them to our newly created window.

Once a screen is identified, Web Learn should add the window. In the left pane is the Web Learn tree and it should contain the window along with all the objects on the page.

The right pane is the Findby pane and shows properties of the object selected in the left hand pane. The attributes listed here are how Certify will find a window or object. We will make changes to these attributes.

- ▶ The attributes should be specific enough to find the right object among many objects. For example, if we specified "find the input box on the page" there might be a dozen matches. But if we specify "find the Username input box on the page" it should be unique.
- ▶ The attributes shouldn't include information specific to the user, development environment, or browser. If we include an identifier for the QA environment in the URL then this window cannot be used in DEV or other environments.
- ▶ The attributes shouldn't include values that will most likely change in the future. For example, if the attribute for the page includes the title "Sales Order 123 Creation" it will only work the first time that order is created.

This course will teach you how to modify these values to work correctly in many situations.



EXERCISE 2.4 — Mapping the Login Page and Saving the Map File

With the Learn utility ready and the application open, you can begin the mapping process.

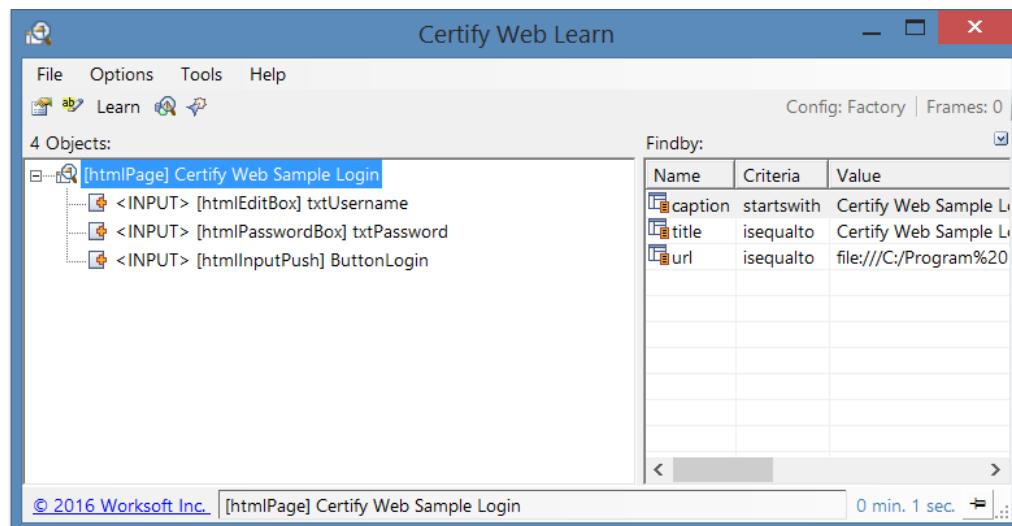
In this exercise, you will map the Login page and modify the attributes to allow execution in various types of browsers.

 IMPORTANT	<p>When learning objects, you have the option to only learn objects in a Web page that are set to "enabled". This is the default option in the Certify Web Learn utility. You can uncheck the option in the Option menu if you wish to learn all elements. When Learned Enabled Options Only is checked, it learns large pages in a short amount of time. For the exercises in this lesson, we will learn objects using both options.</p>
---	---

Step	Action
------	--------

1. Verify both Certify Web Learn and the sample application are open and in view.
2. On the Certify Web Learn menu bar, click **Learn**.

The web applications that are open on your machine displays in the drop-down list.



3. Click Certify Web Sample Login.

The Learn utility begins processing and the objects for the Certify Web Sample Login page are displayed in the Learned Objects pane. The Learned Objects pane also displays the number of objects that were mapped.



Your web configuration determines the fields shown in the Findby area in Web Learn. You may see slightly different options and may not need to make the changes shown in this exercise.

The screenshot shows the 'Certify Web Learn' application window. The menu bar includes File, Options, Tools, and Help. The toolbar has icons for Learn, Find, and other functions. The main area displays '4 Objects:' and a tree view of an HTML page structure under '[htmlPage] Certify Web Sample Login'. The tree includes nodes for <INPUT> [htmlEditBox] txtUsername, <INPUT> [htmlPasswordBox] txtPassword, and <INPUT> [htmlInputPush] ButtonLogin. To the right is the 'Findby:' pane, which contains a table with three rows:

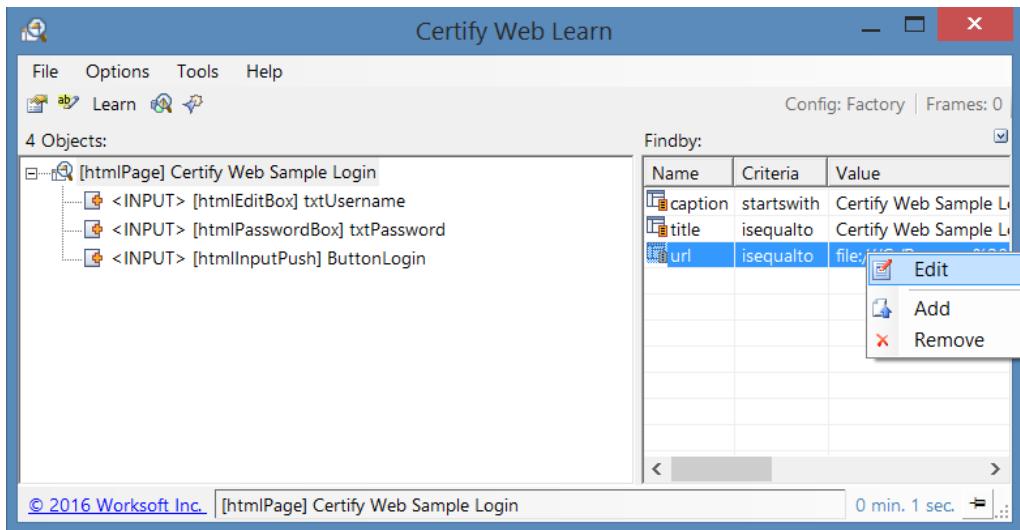
Name	Criteria	Value
caption	startswith	Certify Web Sample Lo
title	isequalto	Certify Web Sample Lo
url	isequalto	file:///C:/Program%20

4. Now we need to take a closer look at the Findby attributes of the window and modify the attributes to make it more dynamic. Right click on **caption** in the Findby pane and select **Edit**.
5. Update the caption to **Certify Web Sample Login** and change the criteria to **startswith**. Click **OK**.

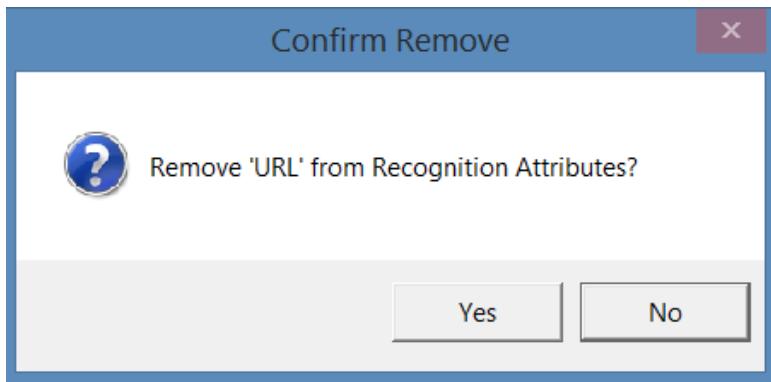
The screenshot shows the 'Attribute Editor - [caption]' dialog box. The title bar says 'Attribute Editor - [caption]'. The main area contains the text 'Certify Web Sample Login'. At the bottom, there are buttons for OK and Cancel, and a dropdown menu labeled 'Criteria = startswith'. Below the dropdown is a table with one row:

caption	startswith	Certify Web Sample Login
---------	------------	--------------------------

6. Right click on **url** and select **Remove**.



7. Confirm the remove by clicking **Yes**.



8. On the menu bar, click **File > Save**.

The Save HTML Map dialog box appears.

9. Browse to a folder where you would like to store this file. For example, My Documents.

Remember this location!

10. In the **File Name** field, verify that **Certify_Web_Sample_Login.map** is displayed.

11. Click **Save**.

Creating Search String Anchors

Anchors are added to uniquely identify windows. Anchors are mostly used in scenarios where there are two browsers open that have the same criteria for setting page context. The chosen search string should exist only on one of the two browsers and will allow Certify to differentiate between two browsers that are otherwise identical. Or, the site may use the same URL for all pages and you need to differentiate between the first page of a site and all other pages.

Add anchors to windows prior to importing into Certify. The window attribute string for each learned object must not exceed 10,000 characters. You can create up to three search strings that can be used for window recognition anchors.



EXERCISE 2.5 — Assigning Anchors to Learned HTML Pages

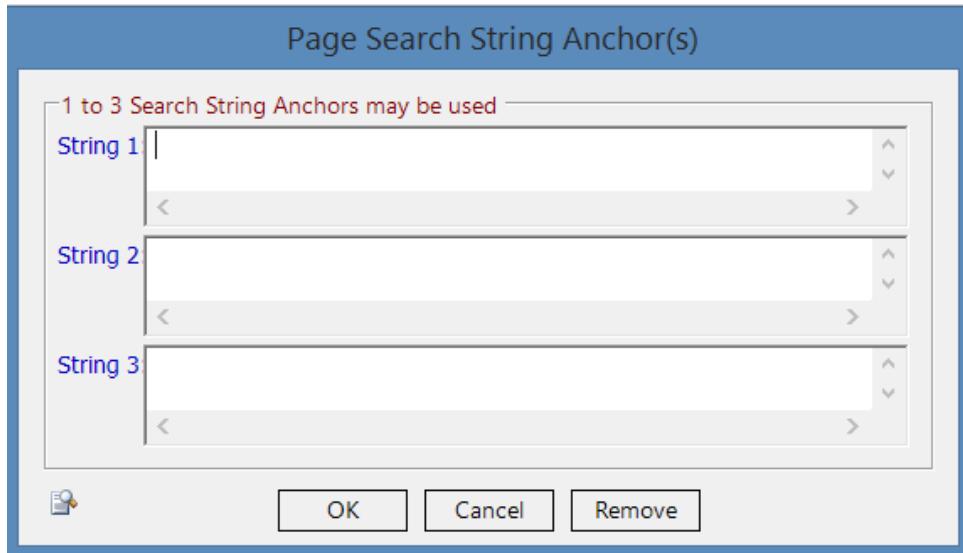
In this exercise you will create a window anchor for **Certify_Web_Sample_Login.map** to uniquely identify the window during execution. The Username field will be used as the anchor because it is unique to this page

	Verify that you are viewing the Certify_Web_Sample_Login.map page you just learned and saved in the previous exercise.
--	---

Step	Action
------	--------

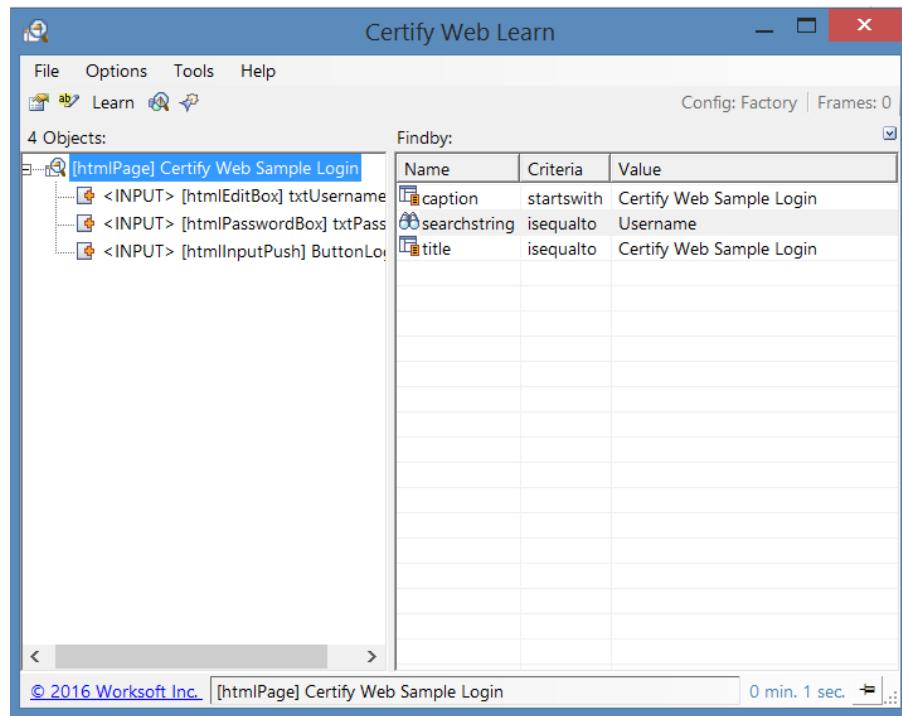
1. On the Certify Web Learn menu bar, click **Tools > Page Search String Anchors**.

The Page Search String Anchor(s) dialog box appears. Up to three search strings can be used for window recognition anchors.



2. In the **String 1** text box, type Username.
3. Click **OK**.

The window anchor appears in the FindBy pane.

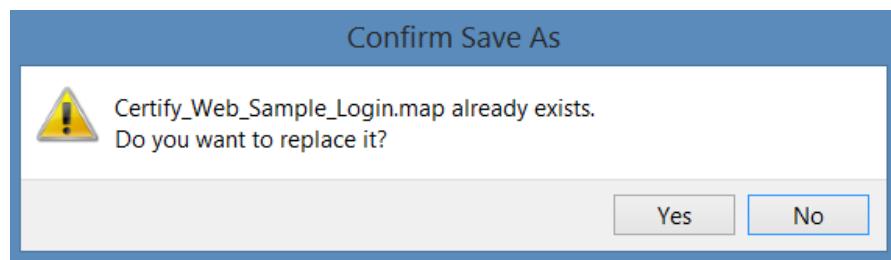


4. On the menu bar, click **File > Save**.

The Save HTML Map dialog box appears.

5. In the **Save In** field, browse to a folder where you would like to store this file. For example, My Documents.
6. In the **File Name** field, verify that **Certify_Web_Sample_Login.map** is displayed.
7. Click **Save**.

A dialog box appears asking if you want to replace the existing file.



8. Click **Yes**.

The Account Master map is saved.



EXERCISE 2.6 — Import Map into Certify Application

You will now import the map into Certify for use to create processes and tests.

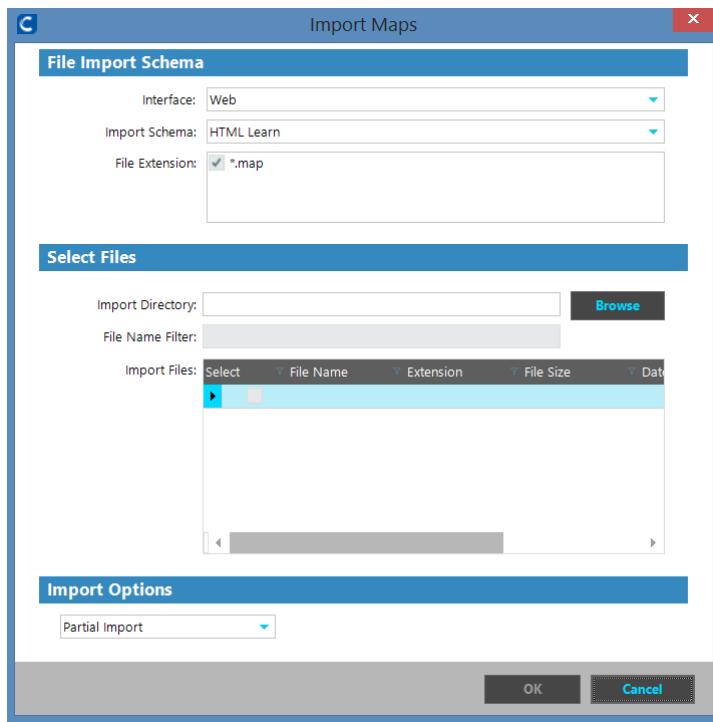
Application maps consist of windows and objects that make up the application you want to test. You mapped the sample application outside of Certify and are now ready to import the map into your Certify database. Importing the maps will make the objects available so that you can create tests or processes using the objects.



NOTE You must have an application and at least one application version created in Certify before importing an application map. If you are importing an application map from Maps, you must have a project created and opened in Certify prior to importing.

Step Action

1. In Certify, on the Navigation taskbar, click **Application**.
The Applications window appears.
2. In the Navigation tree, click the plus sign next to the application you created to expand the folder.
3. Select and right-click **1.0**.
A shortcut menu appears.
4. Select Import Maps.
The Import Maps dialog box appears.



5. Type in field values or select options as follows:

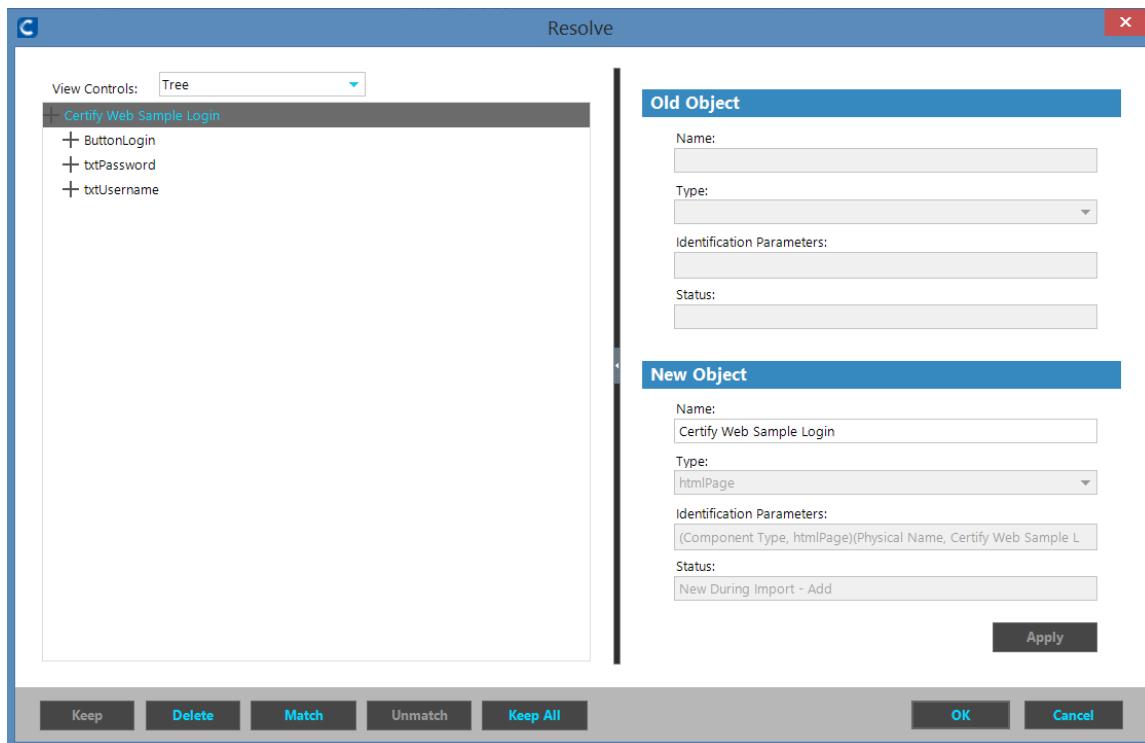
Field/Options	Value/Action
Interface	Web
Import Schema	Select HTML Learn
File Extension	Select checkbox next to .map
Import Directory/Browse	<i>The folder where you stored maps in the previous exercises. For example, My Documents.</i>
Import Files	Select checkbox for Certify_Web_Sample_Login.map
Import Options	Select Partial Import



When importing maps, make sure the Interface and Interface Schema is selected correctly. Failure to do so will prevent you from seeing the maps files in the directory.

6. Click **OK**.

The Import Progress dialog box appears indicating the map files are being processed. When the files have been processed, the Resolve dialog box appears.



Note that the parent node in the tree represents a window object, and with the parent node expanded, you can view all its child objects.

- + New window and/or object imported

When importing an application map for second or subsequent times, symbols you normally see include:

- + New import of window or object
- ✓ Exact match to the current object
- ✗ Window or object is missing from import
- ⊖ Object found is the same as another object, but the new imported object has been changed or at least one of its children has been changed.

7. Click **OK**.

All new windows and objects are added to the database and window names appear in the Summary pane. **Do not close this window. You will continue here for the next exercise.**

Touch Learn Overview

Users can enhance learning objects in an application by using Touch Learn feature. Typically the step will fail during execution because the correct property is further down in the object tree. Certify Web Learn allows you to “touch” the objects and all their properties. The Touch Learn button allows you to add only objects that are selected with Touch Learn, and add objects that dynamically appear after a full learn has completed.



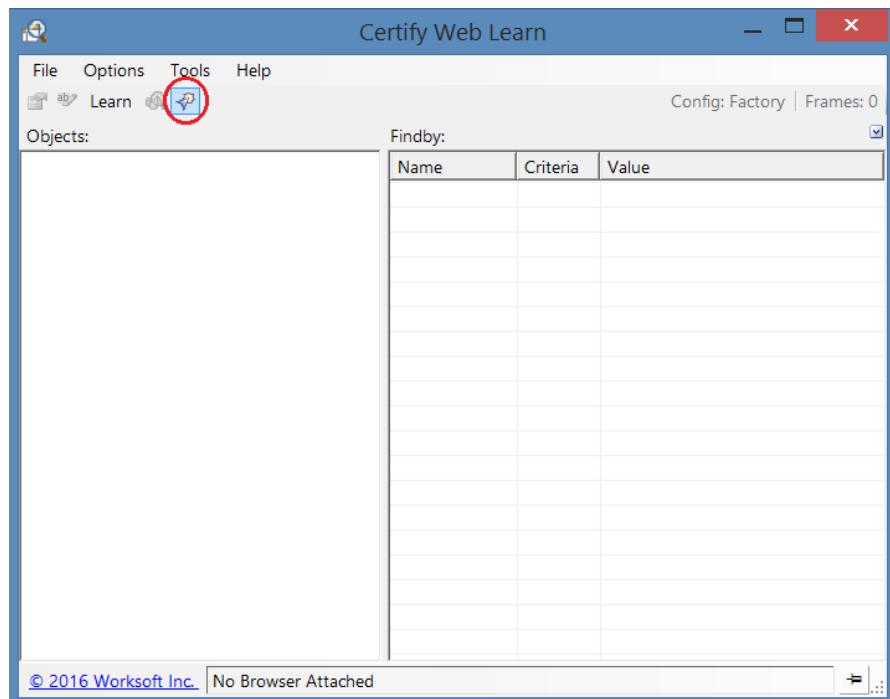
EXERCISE 2.7 — Using the Touch Learn Feature in Certify Web Learn

This exercise walks through the steps to using the Touch Learn feature. For this exercise, we will use Google as the application.

Step	Action
------	--------

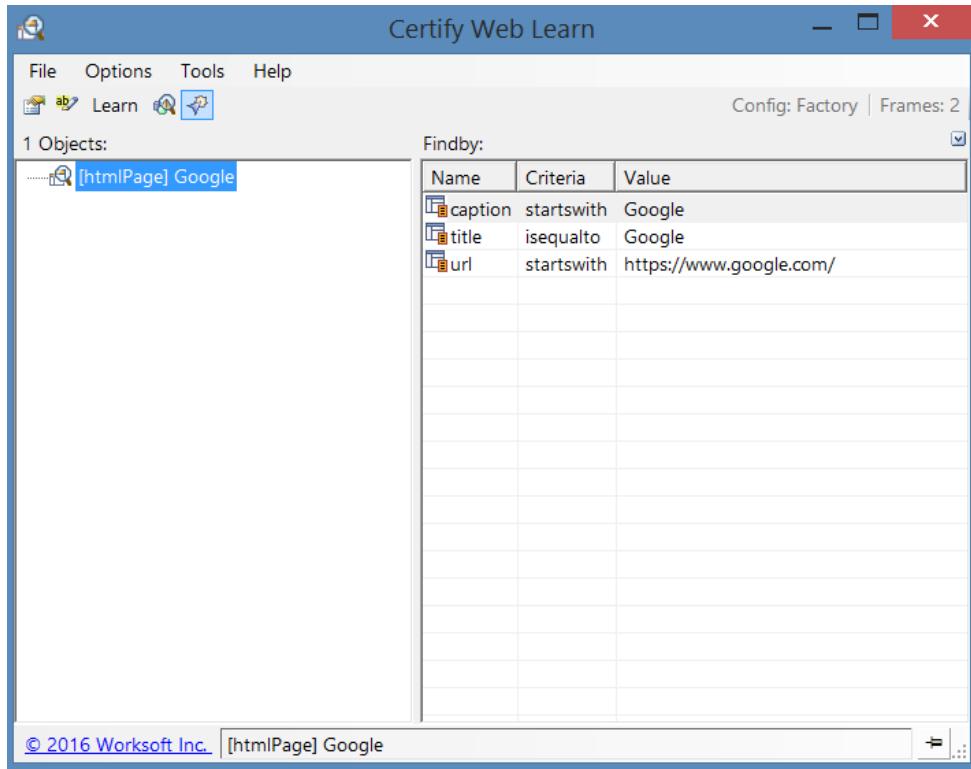
1. Open a browser and navigate to <http://www.google.com>.
2. Launch Certify Web Learn.
3. Click the **Touch Learn** button.

The feature is now ‘turned on.’



4. Click the **Learn** drop-down and select **Google**.

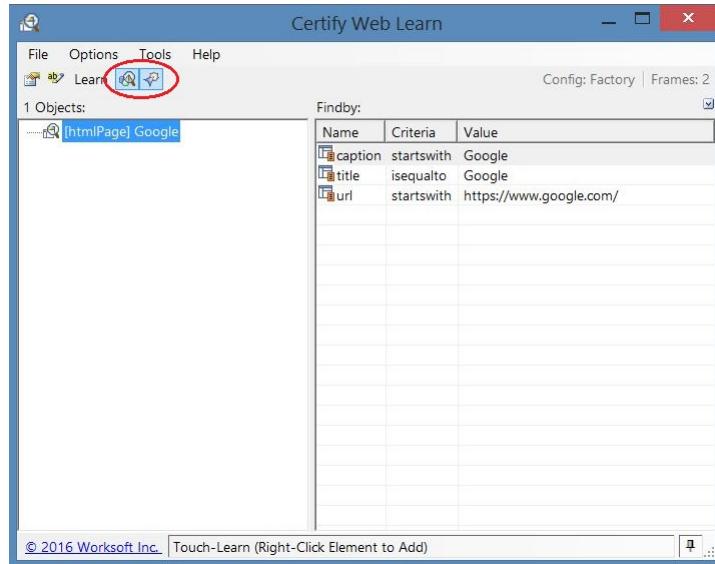
This will learn the Caption, Title, and URL.



We now need to learn the object that is giving up a problem.

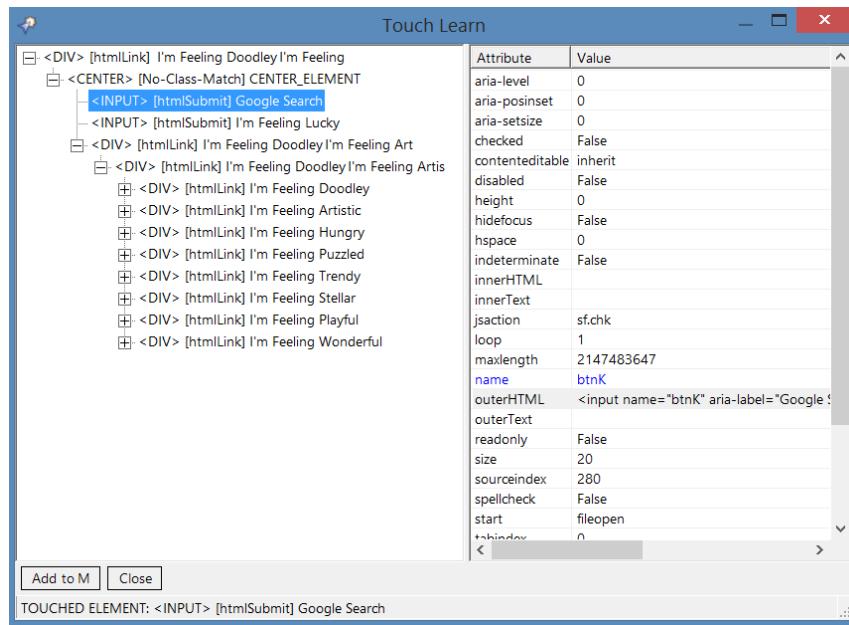
5. Click the **Hover Find** button.

Both the Hover Find and Touch Learn buttons are depressed. Notice the Touch Learn message, "Touch-Learn (Right-click Element to Add)" in the status bar at the bottom of the screen.



6. Go to the application and right-click on the **Google Search** button.

Certify Web Learn disappears and the Touch Learn dialog box appears. The left pane displays a tree structure which is the objects on the page. Typically, the object selected is the correct object.



7. Click Add to Map.
8. Click Close.

Certify Web Learn appears. You will see the objects that Certify Live Touch learned.

Lesson Summary

You've completed the Learning Screens and Importing Maps for HTML Applications lesson.

Key points to remember:

- ▶ Learn all web applications using Internet Explorer.
- ▶ Attributes for objects may need to be modified to allow Certify to identify the object even if it changes value or uses non-default properties.
- ▶ Map files are not available to Certify processes or Live Touch until they are imported into Certify.
- ▶ The Touch Learn feature is useful for identifying objects, especially if the screen has a large number of objects.

Lesson 3

Advanced Topics with HTML Applications

Overview

In this lesson, you will build processes for the Certify Web Sample App.

Objectives

After completing this lesson, you will be able to:

- ▶ Use Best Practices to enhance your HTML testing
- ▶ Understand when and how to use child processes
- ▶ Create negative tests
- ▶ Understanding the Set Attributes action
- ▶ Setting processes to test in different web browsers



When working with HTML applications, it is a good practice to keep Certify Web Learn open and mapped to the current screen under development. This will allow you to easily reference additional attributes of object properties to use in parameters of steps.

Best Practices for using the HTML Interface

The Certify HTML interface is very flexible to allow you to work with many types of HTML applications. This means there isn't one best way to work with all applications – so these Best Practices will need to be adapted for your situation.

Guidelines

These guidelines should be helpful for most applications:

- ▶ Names for Web processes can become quite lengthy when including pages/menu paths. Spend time determining how best to clearly identify processes without creating process names that are very long. For example, the Certify Web Sample Application is identified in process names using the prefix CW_ rather than CertifyWebSampleApplication_.
- ▶ When learning new applications be sure to review the Page attributes. You may need to remove or modify the URL to make it work with multiple environments. It is easy to remember to do this for pages that are learned using Web Learn, but don't forget about pages created using LiveTouch or Capture.
- ▶ Certain applications are built such that the URL, Caption and Title do not change even as the "page/tab" changes. Determine if your application should use anchors to identify unique windows.
 - ▷ Unique Page Advantage: Certify will verify that the correct page is in view before starting execution on that page.
 - ▷ Single Page Advantage: Menu bars are easily accessible from any page.
 - ▷ Single Page Disadvantage: Difficult to identify based on steps which page is currently being acted upon.
- ▶ After creating your process:
 - ▷ Close all browser sessions and rerun your process. This will help identify any session specific attributes that need to be modified.
 - ▷ Run the process against any alternative environments you expect to use (e.g. Development or QA). This will help identify any environment specific attributes that need to be modified.
 - ▷ Run the process in any browsers that should be supported by your application.
- ▶ As with all interfaces, be sure to add steps to verify that the previous step(s) completed successfully. It is always best to trigger a failure as close to the actual failure as possible. For example, verify login was successful during the login process so that any failure is shown in that process rather than the subsequent process.

Recommended Configuration Steps to Consider

The Certify HTML interface has several options that allow you to configure how Certify interacts with your application. There are many types of applications and some of the settings will not work with every application. Use care when turning on or off settings to make sure the change allows Certify to interact better with the application.

The following section gives you some ideas of actions that may be useful. A complete list of actions is included in the back of this training guide.



Use care when turning on or off settings to make sure the change allows Certify to interact better with the application. After changing the setting run through an entire process or integrated process to evaluate the effect.

Set Page Context Timeout

By default, Certify will wait up to 20 seconds for a page to load. If it loads in 5 seconds the step will complete and execution continues. There are times when your process may want to allow more or less time for a page to load. Potentially less time if you are checking for an error pop-up that most likely won't appear. More time if the page has complicated background processing. This timeout will also be used for other execution checks.

Set Object Context Timeout

The default object context timeout is 5 seconds. Similar to the Page Context timeout except used for objects. Certain applications refresh objects in a way that requires extra time.

Set Busy Check

By default Certify checks to see if a page is busy before executing the next step. Certain applications always appear to be busy, in which case you need to turn Busy Check **off** or your process will not execute properly.

Set Cursor Sync

Set Cursor Sync can be useful for any page that shows an hourglass mouse-pointer during AJAX calls. If set **On** the execution engine will check for an hourglass cursor before executing each step and wait for it to go away for up to the page context timeout.

Set Scroll Into View

This action will cause the interface to scroll the object under test towards the top, bottom or turn off scroll into view. The setting will be honored throughout the life of the test or until the action is used again to reset.

One use for setting this off is for applications that have a banner across the top of the screen. When Certify scrolls objects to the top these objects end up being under the banner and cannot be acted upon. Use care though, since at some point objects may need to be scrolled into view and you will need to turn it back to top or bottom.

Set Step Delay

Some applications require a pause between each test step executed. Normally this is not needed for an entire process, but for a set of steps. In that case, set the Step Delay to a value before the set of steps, then set back to 0 for the remainder of the process.



If you find that your application requires one or more of these settings you should consider putting them in the login process. That way they are set every time Certify starts the application.

Defining the CW_FieldCheck Processes

For the Certify Web Sample App process, you will create a parent process, **CW_FieldCheck**, and the child processes listed below in Table 1.

Table 1 — CW_FieldCheck Processes

Integrated Business Process	Sub-Processes	Description
CW_FieldCheck	UTL_CW_Login	Launch browser and login to sample application
	CW.Buttons	Validate button objects
	CW.Editable	Validate edit objects

Creating the CW_FieldCheck Process

In Certify, creating and using folders is a way to organize project components. Before you create the processes for this class, create a folder to store your work.



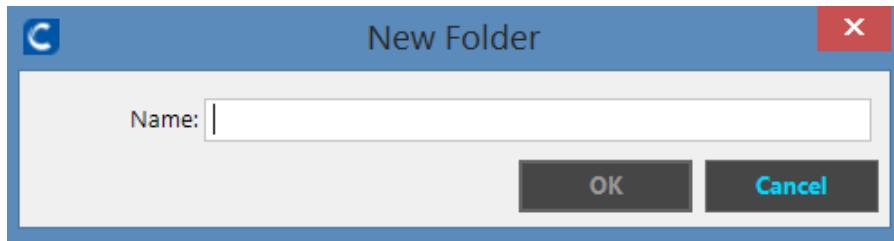
EXERCISE 3.1 — Creating a folder for the CW_FieldCheck Processes

In this exercise, you will create a folder to hold your SAP CW_FieldCheck processes.

Step	Action
------	--------

1. On the Navigation taskbar, click **Processes**.
2. In the Navigation tree, navigate to your Sandbox folder.
3. Right-click on your folder and select **New Folder**.

The New Folder dialog box appears.



4. In the **Name** field, type **CW_FieldCheck**.
5. Click **OK**.

The new folder appears under your name in the Navigation tree.



EXERCISE 3.2 — Creating the UTL_CW_Login Process

In this exercise, you will create a login process, UTL_CW_Login, using the skills learned from our Certify Basics class.

Step	Action
------	--------

1. In the Navigation tree, click your **CW_FieldCheck** folder.
2. Right-click in the **Summary** pane and select **New Process**.
3. Give the process a name and description:
 - a. Name = UTL_CW_Login.
 - b. Description = Process launches application and logs in.
4. Create the following steps:

Load browser to sample application. By using a variable for the browser parameter, we can specify the browser type in our recordset to handle execution in different browser type. URL path may be different based on operating system.

Step	Application Version	Window	Object	Action	
1	System 1.0	Browser	Browser	Load Browser	
Browser		Internet Explorer			
URL		C:\Program Files (x86)\Worksoft\Samples\CertifyWebSampleApp\Version1\CertifyWebSampleLogin.htm			



Certify has the ability to execute processes in different web browsers (Firefox, Chrome, IE). However, development should only be done in Internet Explorer.

Input **admin** into **username** field.

Step	Application Version	Window	Object	Action	
2	Certify Web Sample App 1.0	Certify Web Sample Login	txtUsername	Input	
Value		admin			
Follow-up Key		None			

Input password into password field.

Step	Application Version	Window	Object	Action	
3	Certify Web Sample App 1.0	Certify Web Sample Login	txtPassword	Input	
Value		password			

Click the Login button.

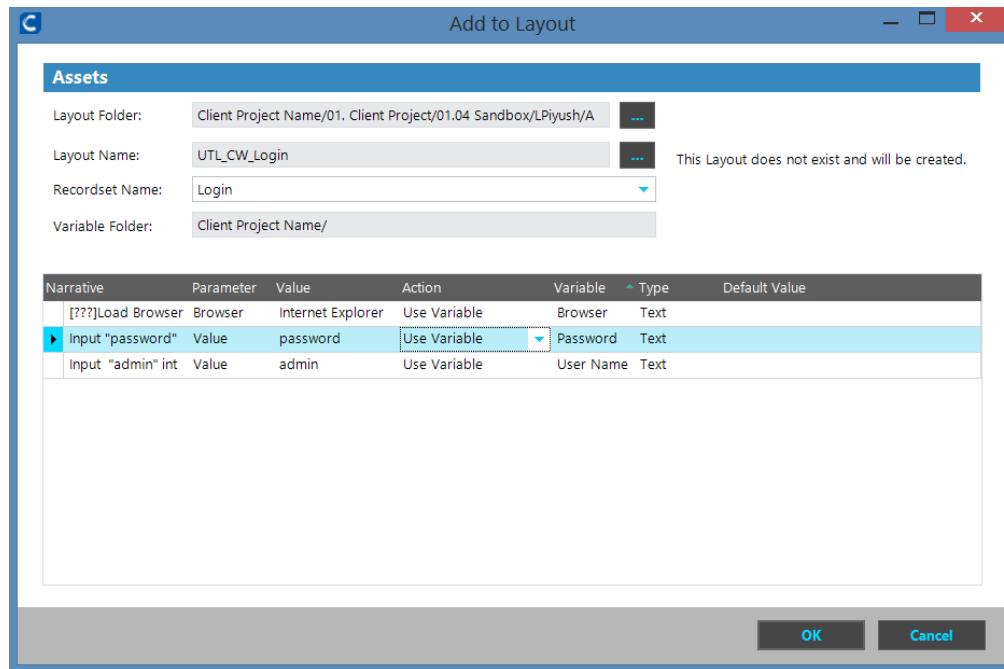
Step	Application Version	Window	Object	Action	
4	Certify Web Sample App 1.0	Certify Web Sample Login	ButtonLogin	Press	
Horizontal %		50			
Vertical %		50			

- 5.** Save the process.

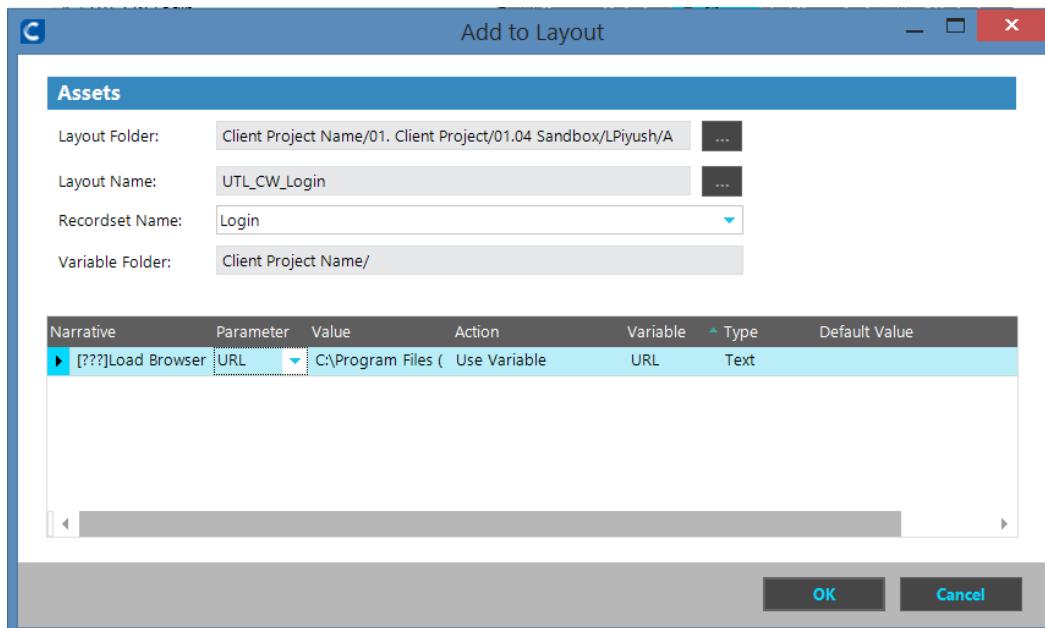


It is important to save often while creating processes. You can save by: 1) clicking the **Save** icon, 2) Selecting **Save** from the **File** menu, or 3) Pressing **Ctrl+S**.

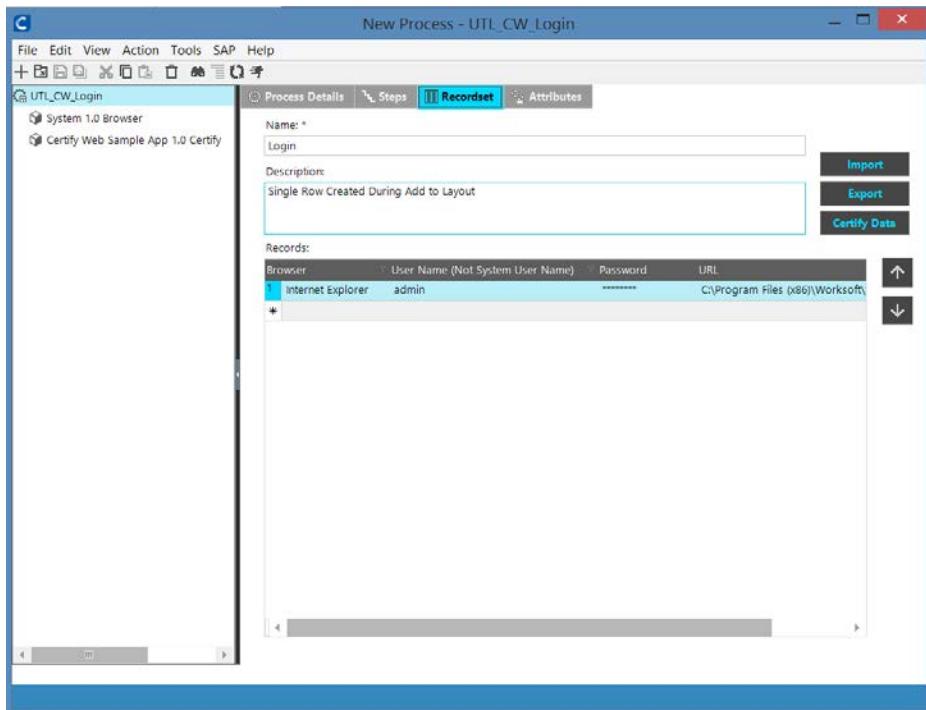
6. Select steps 1 to 3 to create a layout and recordset using **Add to Layout**. You will create a layout, **UTL_CW_Login**, and a recordset, **Login**, using existing variables.



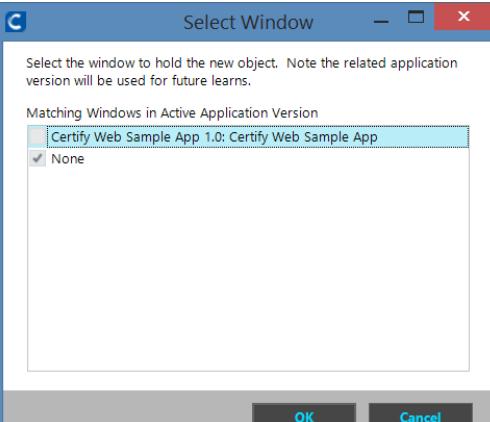
7. Select step 1 again to add the URL variable using **Add to Layout**.



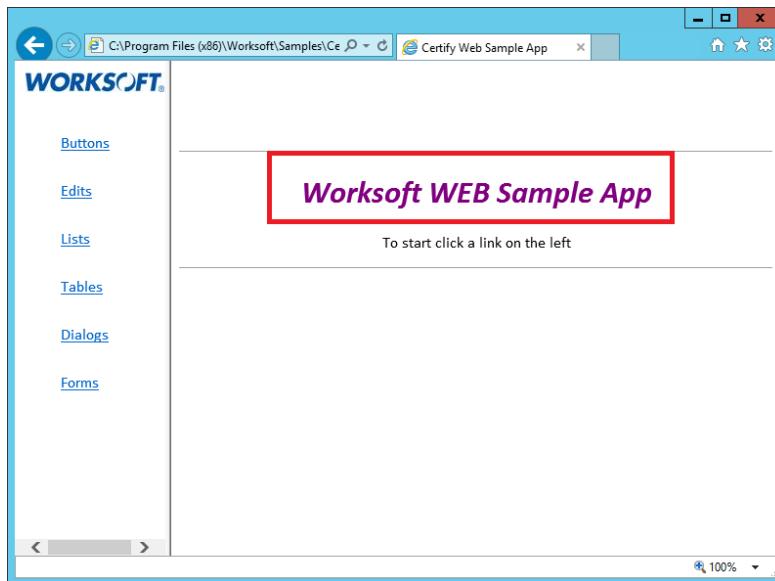
The **Login** layout/recordset should include 4 variables.



Add a step to verify that the next screen is displayed correctly. This step will be created using LiveTouch with the **Learn Objects as Needed** option turned on. This allows LiveTouch to learn and import new map objects into the database simultaneously.

 TIP	<p>When Learn Objects as Needed is enabled, Certify will prompt you to choose the matching window to hold the new objects. The selected application will be used for all future learns.</p>	 A screenshot of the "Select Window" dialog box. It contains instructions: "Select the window to hold the new object. Note the related application version will be used for future learns." Below this is a list titled "Matching Windows in Active Application Version" with "Certify Web Sample App 1.0: Certify Web Sample App" selected. There is also an unchecked checkbox for "None". At the bottom are "OK" and "Cancel" buttons.
---	---	---

8. LiveTouch the words “Worksoft WEB Sample App”.



9. Modify the step to check if the item is Visible.

Step	Application Version	Window	Object	Action
5	Certify Web Sample App 1.0	Certify Web Sample	Worksoft WEB Sample App	Verify
Value		Worksoft WEB Sample App		
Criteria		Is Equal To		

10. Insert Step Below to create a step to Maximize the window.

Step	Application Version	Window	Object	Action
6	Certify Web Sample App 1.0	Certify Web Sample	Certify Web Sample	Set Window State to “Maximize”
Window State		Maximize		

11. Save the Process and Execute to make sure it runs successfully.



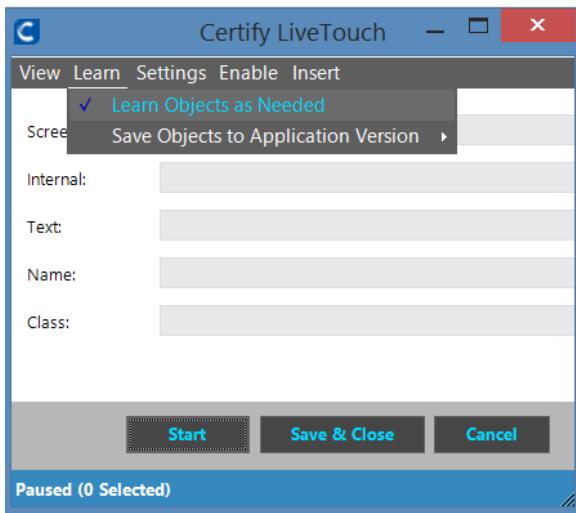
EXERCISE 3.3 — Creating the CW_Buttons Process

In this exercise, you will create a process for the Buttons page.

This exercise will also show you how to build verification steps using different attributes found in the properties of an object. Any attribute discovered with Certify Web Learn can be used for verification.

Step	Action
------	--------

1. Create a new process named **CW_Buttons**.

 NOTE	<p>Make sure LiveTouch setting Learn Objects as Needed is checked.</p> 
--	--

2. Create the following steps for standard buttons using LiveTouch when needed:

Create a comment step.

Step	Application Version	Window	Object	Action
1	System 1.0	System	Execution	Comment
Comment		Standard Buttons Steps		

*Click on the **Buttons** link on the left menu.*

Step	Application Version	Window	Object	Action
2	Certify Web Sample App 1.0	Certify Web Sample	Buttons	Press

Horizontal %	50
Vertical %	50

Press the **Button-1** button.

Step	Application Version	Window	Object	Action	
3	Certify Web Sample App 1.0	Certify Web Sample	btnOne	Press	
Horizontal %		50			
Vertical %		50			

Verify text in edit box is equal to **Button-1**.

Step	Application Version	Window	Object	Action	
4	Certify Web Sample App 1.0	Certify Web Sample	txtButtonOutput	Verify	
Value		Button-1			
Criteria		Is Equal To			

Press the **Button-2** button.

Step	Application Version	Window	Object	Action	
5	Certify Web Sample App 1.0	Certify Web Sample	btnTwo	Press	
Horizontal %		50			
Vertical %		50			

Verify **Value** property of edit box is equal to Button-2.

Step	Application Version	Window	Object	Action	
6	Certify Web Sample App 1.0	Certify Web Sample	txtButtonOutput	Verify Property	
Property		Value			
Value		Button-2			
Criteria		Is Equal To			

Press the **Button-3** button.

Step	Application Version	Window	Object	Action	
7	Certify Web Sample App 1.0	Certify Web Sample	btnThree	Press	
Horizontal %		50			
Vertical %		50			

Store the **InnerHTML** property of Button-3 to be used in Step 9.

Step	Application Version	Window	Object	Action	
8	Certify Web Sample App 1.0	Certify Web Sample	btnThree	Store Property	
Variable		(v) Text			
Property		InnerHTML			

Verify the text in the edit box matches the variable stored in step 8.

Step	Application Version	Window	Object	Action	
9	Certify Web Sample App 1.0	Certify Web Sample	txtButtonOutput	Verify	
Value		(v) Text			
Criteria		Is Equal To			

3. Save the process

Changing an Objects Findby Attributes during execution

There will be objects that you want to identify dynamically during runtime.

For example, you may have a link on the page that contains the current Sales Order Number and there are no other good attributes to find that link. If you know the Sales Order Number you can set the attribute string using a step in your process.

The “Set Attributes” action has a number of ways to modify attributes during runtime. These can be useful if there are parts of the recognition attributes for a page that are dynamic. All of the parameters perform work on the attribute string of the object in the step. The attribute string is only used by Certify to find the object at runtime. The changes these parameters make to the attribute string modify the way that the object is found. All of these parameters can be data-driven, making them a very powerful tool to find dynamic content and elements in dynamic tables.

This action can also be used to create one object that can dynamically be updated to find one of many similar objects. For example, you may wish to push one button from a large set of 10 radio buttons. Instead of having 10 radio button objects and using logic to find which one to press, you can use Set Attributes to easily identify the correct button.

In the exercises that follow we will be using the following parameters.

Substring To Replace and Replacement Substring

The “Substring To Replace” and the “Replacement Substring” parameters work together. All occurrences of the string specified in the “Substring to Replace” parameter will be replaced with the string specified in the “Replacement Substring” parameter. If “Replacement Substring” is left blank, all occurrences of “Substring To Replace” will be removed. The “Substring To Replace” parameter is case-sensitive. If the “Substring To Replace” is not found in the Attribute String, the “Set Attributes” step will fail. The “Replacement Substring” parameter will be ignored if there is no “Substring To Replace” parameter specified.

For example, you may have an attribute such as InnerText is equal to 12345 where you want to substitute the number during runtime. We normally use the word ReplaceMe in the string so it is obvious which part of the attribute is being replaced.

Substring To Replace: **ReplaceMe**

Replacement Substring: **Sales Order Number (variable)**



EXERCISE 3.4 — Continuing with the CW_Buttons Process using Set Attributes action

This exercise will work with checkboxes on the buttons page of the sample app. You will create new dynamic objects and use the Set Attributes action to set content and elements on this page.

Step	Action
------	--------

1. Continue with the CW_Buttons process by creating the following steps.

Create a comment step

Step	Application Version	Window	Object	Action	
10	System 1.0	System	Execution	Comment	
Comment		Checkboxes			

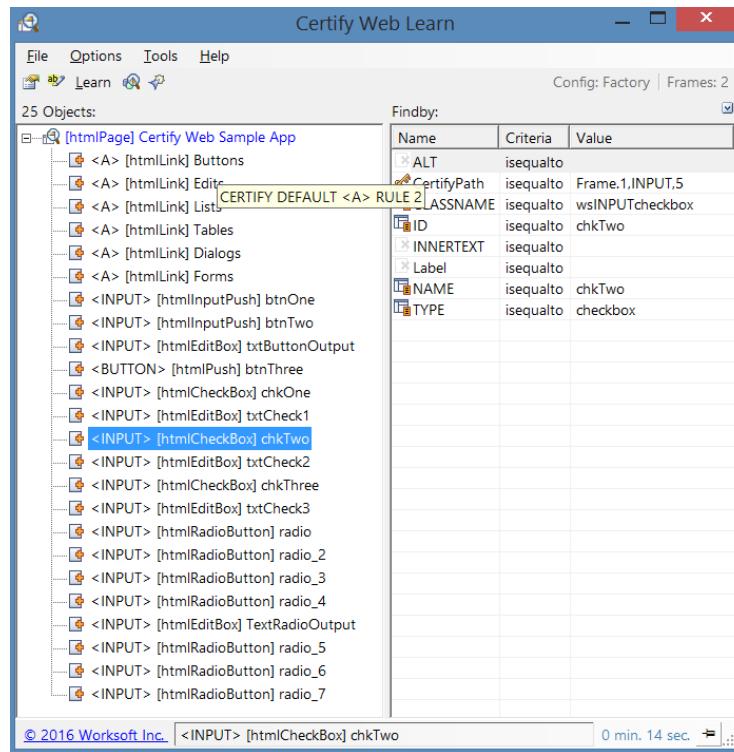
Check the **Check-1** checkbox.

Step	Application Version	Window	Object	Action	
11	Certify Web Sample App 1.0	Certify Web Sample	chkOne	Set	
State		On			
Horizontal %		50			
Vertical %		50			

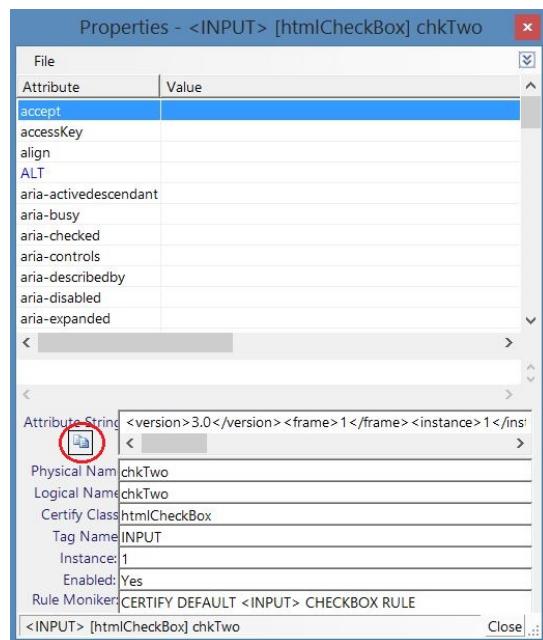
Verify **Check-1** is checked.

Step	Application Version	Window	Object	Action	
12	Certify Web Sample App 1.0	Certify Web Sample	txtCheck1	Verify	
State		On			

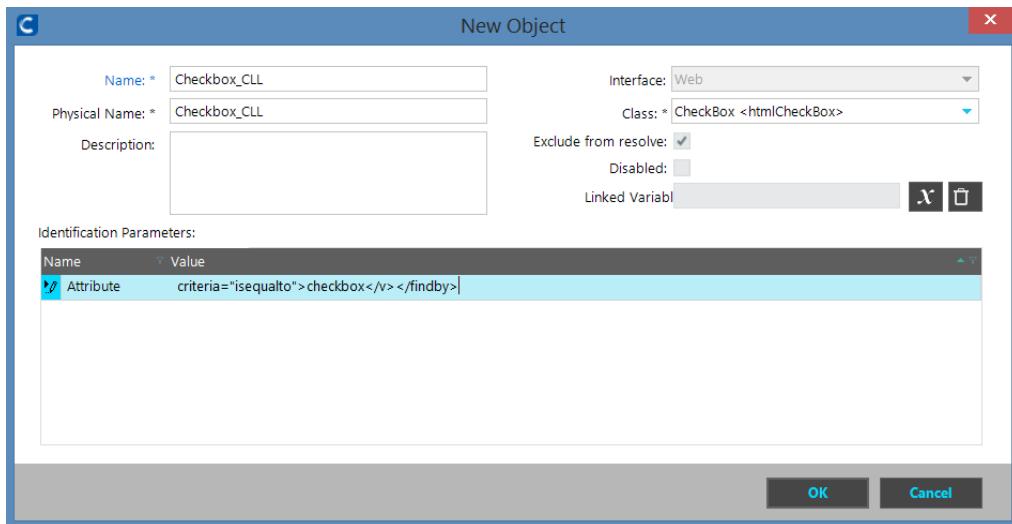
2. In Certify Web Learn, locate the **chkTwo** object in the tree.



3. Right click on the object and select **Properties**.
 4. Click on the Copy Attribute String to Clipboard button.



5. Navigate to Applications module, and expand **Certify Web Sample App -> 1.0** to select **Certify Web Sample App**
6. In the summary pane, right click and select **New Object**.
7. Create a new object with name **Checkbox_YourInitials** along with the following properties. Paste the attribute string copied from Certify Web Learn.



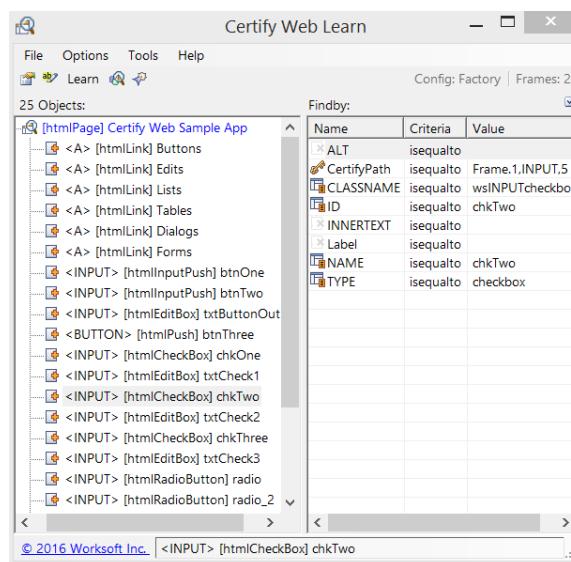
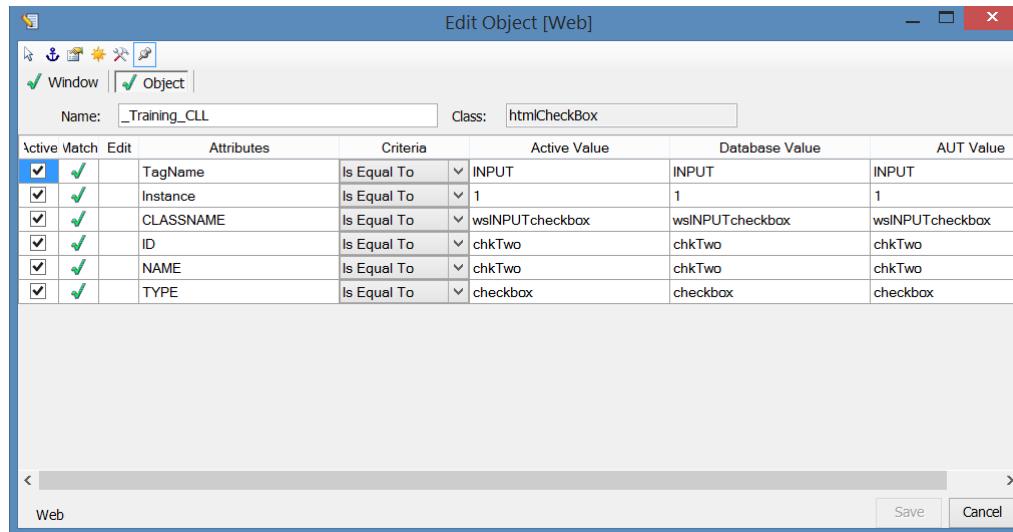
8. Click OK to save the new object.
9. Continue with your process by adding the following step. You can use LiveTouch or manually create the step.

Step	Application Version	Window	Object	Action
13	Certify Web Sample App 1.0	Certify Web Sample	Checkbox_CLL	Set
State		On		
Horizontal %		50		
Vertical %		50		

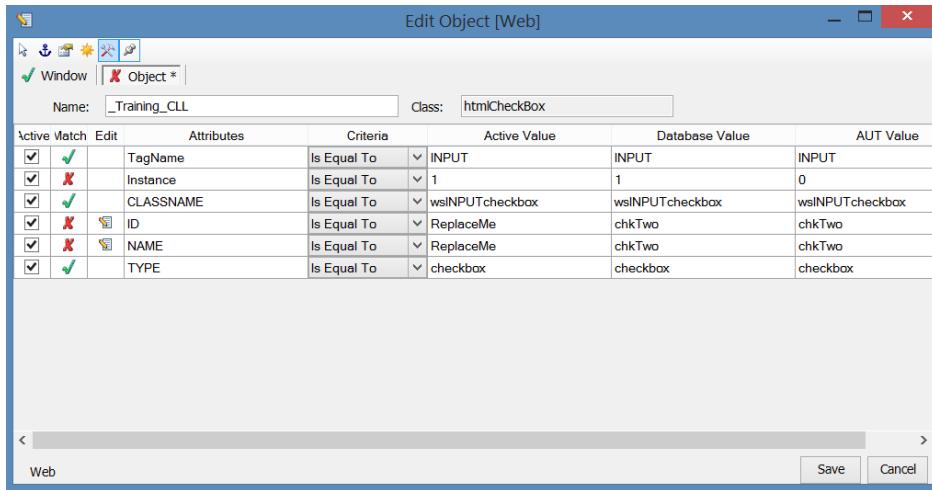
10. Execute the step to make sure that it works.

11. Right Click on the step and select **Edit Object**.

Edit Object allows you to easily update the attributes for a particular object. Note that the attributes shown in Edit Object are similar to the attributes shown in Web Learn.



12. Modify the values in Edit Object replacing chkTwo with the word ReplaceMe as shown below.



Note that the Match columns change to Red X's indicating that the object can no longer be found. That is correct for this exercise. The step will no longer execute correctly at this point.

13. Press OK to close the Edit Object window.
14. Highlight Step 13 and select Insert Step Above to create a new Step 13.
15. Modify the step as shown below so that **chkTwo** will replace **ReplaceMe**.

Step	Application Version	Window	Object	Action
13	Certify Web Sample App 1.0	Certify Web Sample	Checkbox_CLL	Set Attributes
New Attribute String				
Substring To Replace		ReplaceMe		
Replacement Substring		chkTwo		

16. Highlight Steps 13 and 14 and execute them. They should pass.
17. Save the process.



EXERCISE 3.5 — Create the CW_Buttons_C_VerifyRadioButtons Process

When creating a process, you can drive looping through recordsets. In this exercise, you will create a child process for CW_Buttons, called **CW_Buttons_C_VerifyRadioButtons** that will handle validating all the radiobuttons.

We will create one radio button object that will be used for all of the radio buttons on the page.

Step	Action
1.	In the CW_Buttons process, right-click step 14 and select Insert Step Below .
2.	Manually change the step to match the following

Step	Application Version	Window	Object	Action
15	System 1.0	System	Execution	Execute Process

3. In the Parameters tab, in the **Process** field, select the **New Process**  icon.

Untitled appears in the field as well as in the tree under CW_Buttons.

4. In the Navigation tree, click **Untitled**.

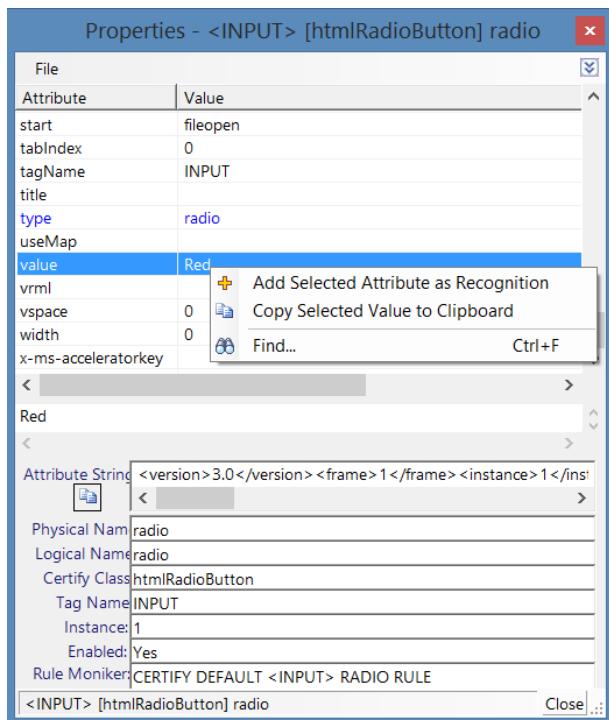
Certify dialog box appears asking to save the current process.

5. Click **Yes**.
6. Click the Process Details tab.
7. Give the process a name and description:
 - a. Name = CW_Buttons_C_VerifyRadioButtons
 - b. Description = This process validates all the radiobuttons
8. Save the process.



A best practice for naming child processes that are specific to certain transactions is ParentProcessName_C_ChildProcessDescription.

9. Navigate to Certify Web Learn and look at the Findby attributes for the **radio** object. There is nothing that identifies the radiobutton with the label. For this particular object, we do not see any reference to Red for this radio button.
10. Right click and select **Properties** for **radio** object. In the properties window, locate the attribute **value** and add it to the attribute by right clicking and selecting **Add Selected Attribute as Recognition**.
11. Close the Properties window.



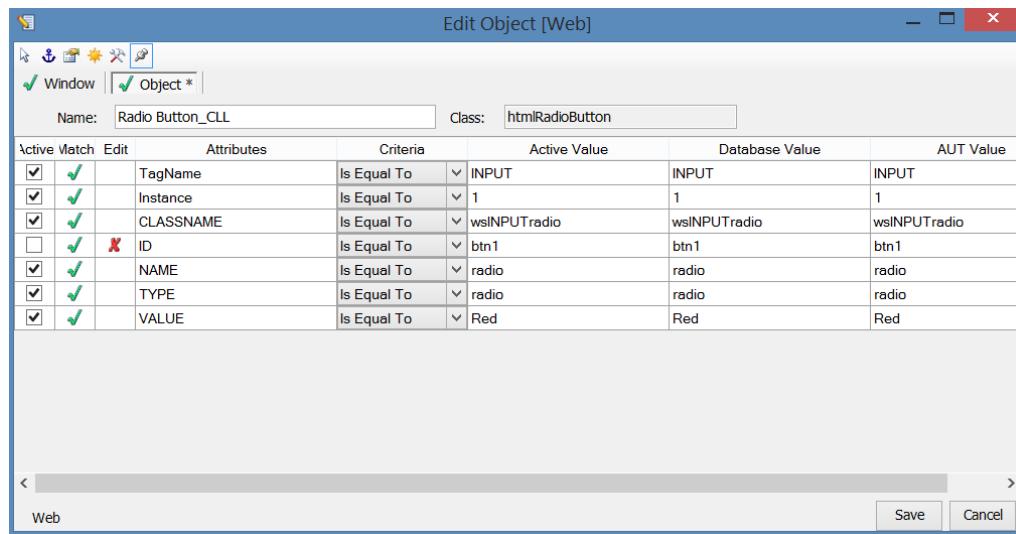
12. Scroll through the Findby attributes and now you can see the **value** attribute has been added and includes references to Red. Compare these attributes to the Findby attributes for the other Radio buttons. It looks like the difference between the buttons will be the ID and the Value attributes.

We could create an object as shown in the previous exercise. As an alternative, we are going to use LiveTouch to create the object, then Edit Object to change the attributes.

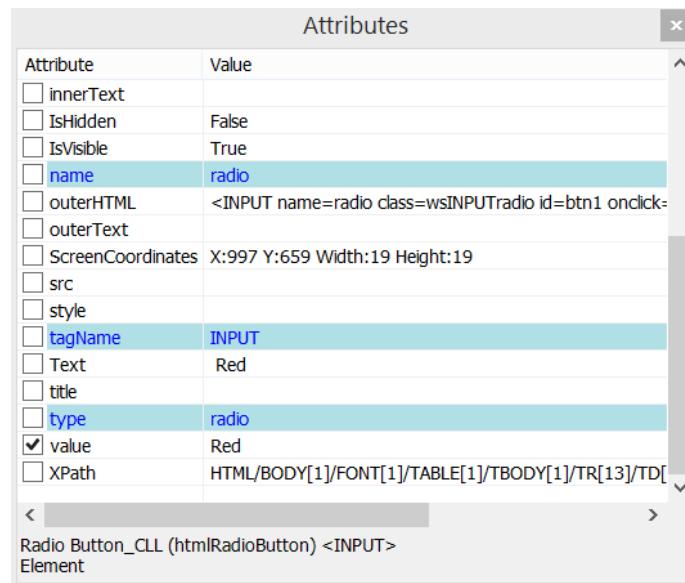
13. Insert a Step with LiveTouch and select the Red radio button.

Step	Application Version	Window	Object	Action
1	Certify Web Sample App 1.0	Certify Web Sample	radio	Press
	Horizontal %	50		
	Vertical %	50		

14. Right click on Step 1 and select **Edit Object**.
15. Name the object Radio Button_YourInitials.
16. Uncheck the box next to ID to remove this attribute from the Attribute String.



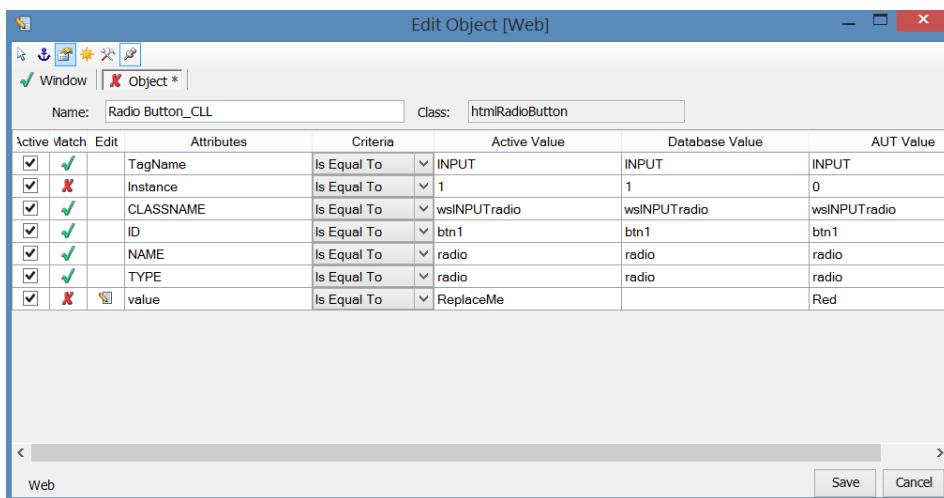
17. Press the Add Attributes button in the Edit Object window.
18. Scroll down the list and Check the box next to value to add this attribute to the Attribute String.



19. Press OK to close the Attributes window.

We want to make the color dynamic so it will work with all buttons.

20. Change **Red** to **ReplaceMe**.



21. Press Save to save and close the Edit Object window.
22. Press Save to save your process. Note that the name of the Radio object has changed to use your new name.
23. Insert a Step Above to create a new Step 1 to Set Attributes to the correct color.

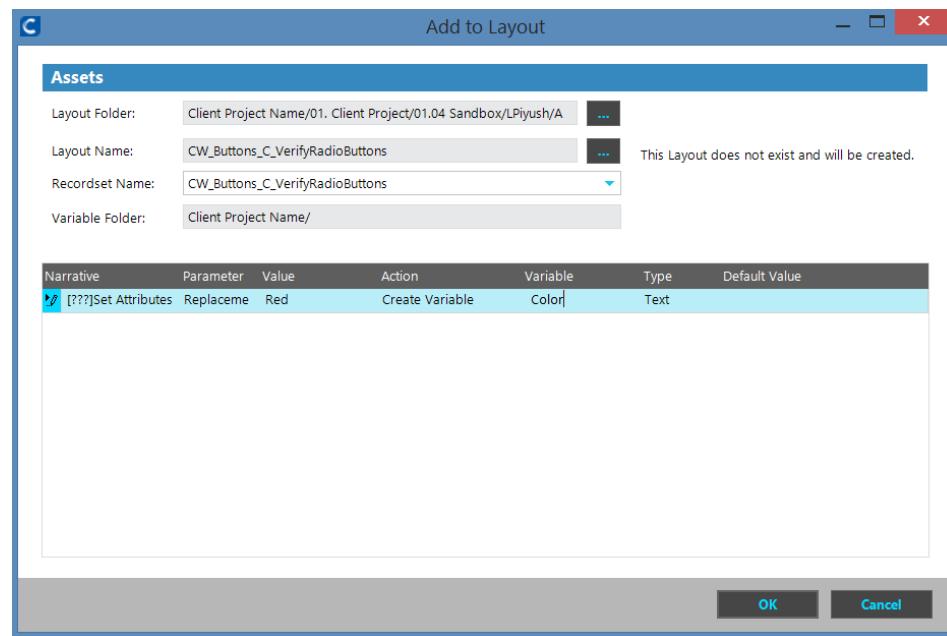
Create Set Attribute step to replace ReplaceMe with Red.

Step	Application Version	Window	Object	Action
1	Certify Web Sample App 1.0	Certify Web Sample App	Radio Button_CLL	Set Attributes
New Attribute String				
Substring To Replace		ReplaceMe		
Replacement Substring		Red		

Your step 2 should look as follows.

Step	Application Version	Window	Object	Action
2	Certify Web Sample App 1.0	Certify Web Sample App	Radio Button_CLL	Press
Horizontal %		50		
Vertical %		50		

24. Create a layout and Recordset named **CW.Buttons_C.VerifyRadioButtons** using the **Color** variable by selecting Step 1 and right clicking it to select **Add to Layout** option from drop down.



25. Modify the **recordset** to add rows of data for each color in the Certify Web Sample App.

The screenshot shows the 'Recordset' tab of a configuration interface. The 'Name:' field is set to CW.Buttons_C.VerifyRadioButtons. The 'Description:' field contains the text 'Single Row Created During Add to Layout'. Below these fields is a section labeled 'Records:' containing a table with seven rows, each representing a color: 1. Red, 2. Orange, 3. Yellow, 4. Green, 5. Blue, 6. Indigo, and 7. Violet. The row for 'Violet' is highlighted with a blue background. An asterisk (*) is present at the bottom of the record list.

- 26.** Insert a Step using LiveTouch to verify the **EditBox** matches the radio button selected. Be sure to use the variable you created above.

Step	Application Version	Window	Object	Action	
3	Certify Web Sample App 1.0	Certify Web Sample App	TextRadioOutput	Verify	
Value		(v) Color			
Criteria		Is Equal To			

- 27.** Update step 15 in the process to include the layout and recordset in the parameters.

Step	Application Version	Window	Object	Action	
15	System 1.0	System	Execution	Execute Process	
Process		CW.Buttons_C.VerifyRadioButtons			
Start At Step		First Step			
Layout		CW.Buttons_C.VerifyRadioButtons			
RecordSet Name		RadioButtons Colors			
Recordset Mode		Read Only			
RecordSet Filter		None			

- 28.** Save the process.

At this point, process **CW.Buttons_C.VerifyRadioButtons** should be self-contained and

ready to run. Highlight the process in the Processes summary pane and click the **Run** icon.



Check the status in the Result Viewer to see if the process passed or failed. If it failed, expand the Result Viewer tree to determine where the failure occurred.

Troubleshoot as needed using the Results Log until the process runs completely without errors.



EXERCISE 3.6 — Create the CW_Edits Process

In this exercise, you will create a new process that interacts with different objects. You will also learn how to create negative tests by switching the On True / On False values in the steps.

Step	Action
1.	Create a new process: a. Process Name = CW_Edits b. Description = This process verifies edit objects on screen.
2.	Create the following steps for this process: Press the Edits link.

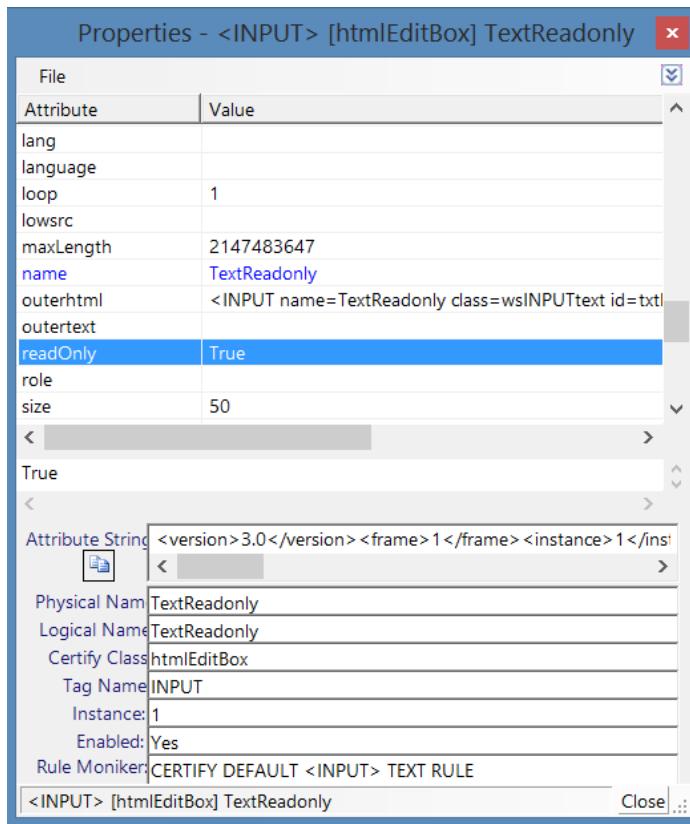
Input text into the **Editable** field..

Step	Application Version	Window	Object	Action	
2	Certify Web Sample App 1.0	Certify Web Sample	Edits	Press	
Horizontal %		50			
Vertical %		50			

Verify text in the **Read Only** field.

Step	Application Version	Window	Object	Action	
3	Certify Web Sample App 1.0	Certify Web Sample	Text_READONLY	Verify	
Value		I am read-only			
Criteria		Is Equal To			

Verify **ReadOnly** property in the attribute. This can be located using Web Learn.



ReadOnly is not in the drop-down list for Property. You can type it in the field.

Step	Application Version	Window	Object	Action
4	Certify Web Sample App 1.0	Certify Web Sample	Text_READONLY	Verify Property
Property		ReadOnly		
Value		True		
Criteria		Is Equal To		

Insert comment step

Step	Application Version	Window	Object	Action
5	System 1.0	System	Execution	Comment
Comment		Negative test to enter length greater than 5.		

Create a negative test by modifying the On True / On False settings of the step. Insert step to enter **123456** in to Maxlength=5 field.

Step	Application Version	Window	Object	Action	
6	Certify Web Sample App 1.0	Certify Web Sample	TextMaxLength	Input	
Value		123456			
Follow-up Key		None			
On True / On False	On True: Log Status As	Failed			
	Action	Continue			
	On False: Log Status As	Passed			
	Action	Continue			

Input text into **Visibility** field.

Step	Application Version	Window	Object	Action	
7	Certify Web Sample App 1.0	Certify Web Sample	TextVisibility	Input	
Value		Visibility			
Follow-up Key		None			

Click on the **Visibility** link.

Step	Application Version	Window	Object	Action	
8	Certify Web Sample App 1.0	Certify Web Sample	Visibility	Press	
Horizontal %		50			
Vertical %		50			

Negative test to input text into Visibility field. This step should return False because field is not visible after executing step 8.

Step	Application Version	Window	Object	Action	
9	Certify Web Sample App 1.0	Certify Web Sample	TextVisibility	Input	
Value		Visibility			
Follow-up Key		None			

Change the On True / On False condition to Fail on True and Pass on False.

On True: Log Status As	Failed
Action	Continue
On False: Log Status As	Passed
Action	Continue

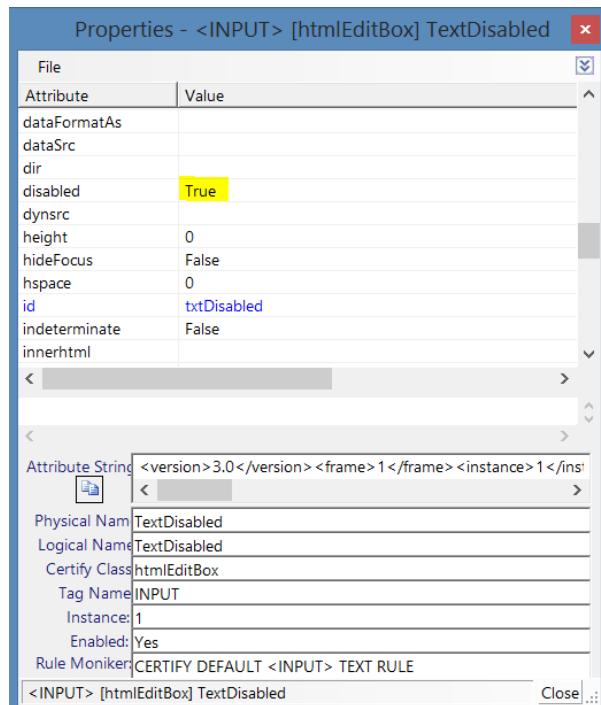
Enter text into the **Password** field. The values are masked in the application.

Step	Application Version	Window	Object	Action	
10	Certify Web Sample App 1.0	Certify Web Sample	TextPassword	Input	
Value		Password			

Verify text in the **Password** field. Even though values are masked, they can still be verified.

Step	Application Version	Window	Object	Action	
11	Certify Web Sample App 1.0	Certify Web Sample	TextPassword	Verify	
Value		Password			
Criteria		Is Equal To			

Verify **disabled** property of **TextDisabled** field is set to True. The attribute value can be found using Web Learn.



Step	Application Version	Window	Object	Action
12	Certify Web Sample App 1.0	Certify Web Sample	TextDisabled	Verify Property
Property		Disabled		
Value		True		
Criteria		Is Equal To		

Click on the **Browse** button. When you LiveTouch, the object will include the field and the browse button. Note the change in the **Horizontal %** in the parameters.

Step	Application Version	Window	Object	Action
13	Certify Web Sample App 1.0	Certify Web Sample	TextFile	Send Click
Click Type		LeftClick		
Mask		None		
Horizontal %		90		
Vertical %		50		



By default, Certify will click in the center of an object with 50/50 coordinates. By adjusting the Horizontal % and Vertical %, you can alter the placement of the click. For example, if you need to click to the right, the Horizontal % can be increased to 80%.

Close the pop up menu using the **Set Window State** System Action

Step	Application Version	Window	Object	Action	
14	System 1.0	System	System	Set Window State	
State		Close			
Target Window Caption		Choose File to Upload			
Caption Criteria		Is Equal To			

Enter text in the Dynamic field. Note that the name of the object will have the current time and date.

Step	Application Version	Window	Object	Action	
15	Certify Web Sample App 1.0	Certify Web Sample	txtDynamic_Fri Apr 29 15:10:27 CST 2016	Input	
Value		Hello			

3. Save the process



EXERCISE 3.7 — Create the CW_FieldCheck Parent Process

Integrated Business Process	Sub-Processes	Description
CW_FieldCheck	UTL_CW_Login	Launch browser and login to sample application
	CW.Buttons	Validate button objects
	CW.Editable	Validate edit objects

This exercise will string all three child processes into a parent process. UTL_CW_Login process will have a layout and recordset attached.

- | Step | Action |
|------|---|
| 1. | Create a new process:
c. Process Name = CW_FieldCheck
d. Description = This process validates the buttons and edit pages of the Web Sample App. |
| 2. | Create the following steps for this process: |

Execute **UTL_CW_Login** process with layout and recordset attached.

Step	Application Version	Window	Object	Action
1	System 1.0	System	Execution	Execute Process
Process		UTL_CW_Login		
Start at Step		First Step		
Layout		UTL_CW_Login		
Recordset Name		Login		
Recordset Mode		Read Only		
Recordset Filter				

Execute **CW_Butons** process.

Step	Application Version	Window	Object	Action	
2	System 1.0	System	Execution	Execute Process	
Process		CW_Butons			
Start at Step		First Step			
Layout					
Recordset Name					
Recordset Mode					
Recordset Filter					

Execute **CW_Edits** process.

Step	Application Version	Window	Object	Action	
3	System 1.0	System	Execution	Execute Process	
Process		CW_Edits			
Start at Step		First Step			
Layout					
Recordset Name					
Recordset Mode					
Recordset Filter					

3. Save the process



- 4.** Execute the parent process by clicking the **Run** icon.
- 5.** Troubleshoot as needed using the Results Log until the process runs completely without errors.

Lesson Summary

You've completed the [Advanced Topics for HTML Applications](#) lesson.

Key points to remember:

- ▶ Use a standardized folder structure and naming convention to keep your tests well organized.
- ▶ When creating processes for HTML, work at the component level and break test cases by pages.
- ▶ Use your Certify Web Learn and LiveTouch, when possible to save time and effort.
- ▶ Use child processes in processes where an unknown number of items may be processed.
- ▶ Develop all web applications using Internet Explorer. Processes can be executed in different supported web browsers.

Lesson 4

HTML Classes and Actions

Overview

In this lesson, you will learn the various classes and actions for the HTML interface.

General Classes and Actions

A **Class** is a description of a category of objects. A class defines how to recognize objects and the actions that can be executed against the objects. Each Certify interface has a set of classes and associated actions already defined.

An **Action** is an individual activity or operation that can be automatically executed on a class. Some actions simulate an operation and some actions check to see if a condition is true. Each action can have input parameters and output parameters. Actions are invoked when you execute a process, and if parameters are required, prompts you for the parameters. For example, entering data to a EditBox requires a parameter for the data value. Each class and action combination invokes a particular function in the interface library, and the parameters are passed into the function during execution as arguments.

Some actions simulate an operation like press, while others check to see if a condition is true such as enabled, focused, and visible.

All classes in the HTML interface inherit the following generic actions:

Class	Action
Generic	Fire Event Key Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible



For a complete list of the HTML Classes and Actions, go to the Navigation toolbar, click **Interfaces**, then select **HTML**. **View this area only**. Modifying this list will cause problems in Certify.

HTML Classes and Actions

As well as having generic actions, some classes have actions that are unique to the class. The table below lists the unique actions for each class.

Control	Description	Actions
Page	A Page class represents the page being tested	Click Find Text Select Context Menu Set Attributes Set Window State Store StatusText Store URL Verify StatusText Visible Wait Wait For Page Busy
Browser	System Browser Class	Capture Test Close Browser Create Debug Log Load Browser Load URL Page Back Page Forward Set Busy Check Set Cursor Sync Set Execution Engine Set Hidden Check Set Input Options Set Modal Watch Set Object Context Timeout Set Page Context Timeout Set Scroll Into View Set Step Delay Store Browser Version Verify Browser Version

Control	Description	Actions
Cell	The Cell represents a cell in an html table	Fire Event Key Press Left Click Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible
CheckBox	The Checkbox represents a normal html checkbox	Fire Event Key Press Send Click Set Set Attributes Set Property Store Property Type Keys Verify Verify Property Visible
ComboBox	The ComboBox represents normal dropdown lists	Fire Event Key Press Select Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible
DropDown	The DropDown represents complex multi-element dropdown lists	Select Send Click Set Attributes Store Type Keys Verify Visible Select Visible

Control	Description	Actions
EditBox	The EditBox represents an html text field	Fire Event Input Key Press Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible Fire Event Input Key Press Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible
File	The File represents an html file lookup element	Fire Event Input Key Press Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible
FloatFrame	The FloatFrame represents an html iframe	Fire Event Key Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible

Control	Description	Actions
Form	The Form represents an html form element	Fire Event Key Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
Frame	The Frame represents an html frame	Fire Event Key Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
Hidden	The Hidden represents an html element of type hidden	Fire Event Key Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
htmlHeader	The Header represents an html header element	Fire Event Key Press Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible

Control	Description	Actions
htmlReset	The Reset represents an html form reset element	Fire Event Key Press Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
Image	This Image represents an html image	Fire Event Hover Key Press Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
Link	The Link represents an html anchor object	Fire Event Hover Key Press Press Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible

Control	Description	Actions
ListBox	htmlListBox	Fire Event Key Press Select Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible
Object	htmlObject	1) Set Browser 2) Find Page Context using Frame Number 2) Find Page Context using Search String 3) Find Object using Attribute 3) Find Object using ID Property 3) Find Object using Name Property 3) Find Object using Search String 4) Get Object Child 4) Get Object Parent 4) Get Object Sibling 5) Perform Object Click 5) Perform Object Hover 5) Perform Object Input 5) Perform Object Select (List/Combo Box) 5) Perform Object Store 5) Perform Object Store Property 5) Perform Object Verify 5) Perform Object Verify Property Key Press Send Click Type Keys

Control	Description	Actions
PasswordBox	htmlPasswordBox	Fire Event Input Key Press Send Click Set Attributes Set Property Store Store Property Type Keys Verify Verify Property Visible
PostForm	htmlPostForm	Fire Event Key Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
PushButton	htmlPushButton	Fire Event Key Press Press Send Click Set Attributes Set Property Store Property Type Keys Verify Property Visible
RadioButton	htmlRadioButton	Fire Event Key Press Press Send Click Set Attributes Set Property Store Property Type Keys Verify Verify Property Visible

Control	Description	Actions
Table	htmlTable	Find Column Find Node Row Find Row Find Row (Advanced) Find Text Fire Event Input Into Cell Key Press Select Cell Select Row Send Click Set Attributes Set Property Store Cell Store Property Type Keys Verify Cell Verify Property Visible
Tree	htmlTree	Select Node Visible

Lesson 5

Resources

Overview

This section identifies useful resources for working with Worksoft Certify and contacting Worksoft.



Worksoft Certify Resources

Access Worksoft Certify Help

If you need additional information on the features in Worksoft Certify, press **F1** or click **Help** in the menu bar.

New Feature	Description
Updated Certify Capture tool	<p>Certify Capture™ now allows for you to capture actions performed against an SAP® or HTML applications under test.</p> <p>This tool's functionality has been updated. For information to use Certify Capture, see the following topics:</p> <ul style="list-style-type: none"> ◆ Adding a New Process with Certify Capture ◆ Adding Steps to a Process with Certify Capture ◆ Inserting a Comment Step with Certify Capture ◆ Inserting a Screen Capture Step with Certify Capture ◆ Inserting Verify Steps with Certify Capture ◆ Adding a Named Activity with Certify Capture
Within Certify, a Certify administrator is able to create filter rules to control the scope of the information that is being gathered by Worksoft Analyze™.	<p>For more information, see the following topics:</p> <p>Creating Worksoft Analyze Filter Rules</p> <p>Testing All Worksoft Analyze Filter Rules Together</p> <p>Editing Worksoft Analyze Filter Rules</p>
Simplify Worksoft Certify integration with the HP Quality Center®	<p>If you are using Certify v9.0.3 or later versions, you are able to configure your settings for HP Quality Center integration using the new Quality Center Configuration tool. This tool is located in the Certify Tools menu: Tools > Configure > QC.</p> <p>For more information about the configuration tool, see the Worksoft Help Portal.</p>

Certify Log File

Certify Log files contain information both on normal operation of Certify and any errors. There is a general log file and log files for specific interfaces.

In the Certify Help menu, click **Help > Show Log**

Select either "**Show Certify Log File**" or "**Show Certify Log Folder**"

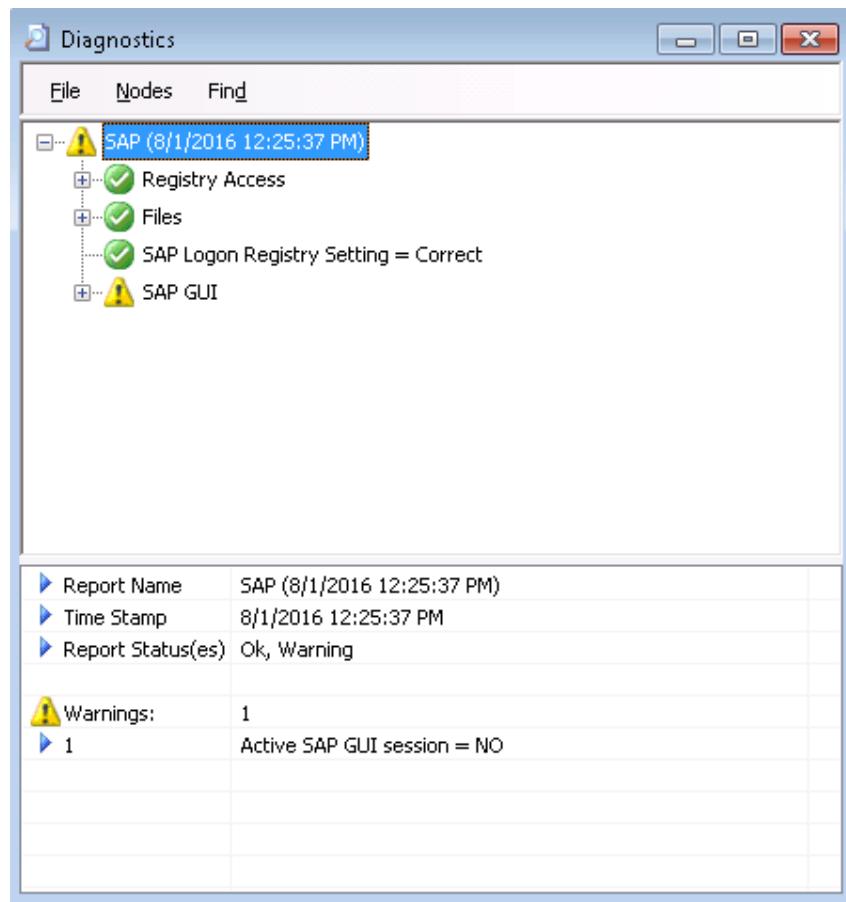
Certify Diagnostic tool

The Certify Diagnostic tool helps you diagnose issues that you encounter when using Certify by creating a report about a specific interface.

In the Certify Help menu, click **Help > Diagnostics**

Select either "**All Diagnostics**" or a specific Interface such as SAP.

Output can be saved in a file and e-mailed to Support as needed.



Worksoft Community and Customer Solutions Site

The Worksoft Community site allows you to:

- ▶ Discuss and learn about Worksoft products.
- ▶ View Worksoft published media.
- ▶ Download Worksoft product updates.
- ▶ View a knowledge base for Worksoft products.
- ▶ Submit a case.
- ▶ View answers to frequently asked questions.
- ▶ Find support-related solutions.

To utilize the Worksoft Community and Customer Solutions site, you must register:

<http://portal.worksoft.com>

To request a Portal account login, send an email to customer-portal@worksoft.com and include the following information in your email: full name, email address, department/company (and Client Name if you are a Worksoft partner), job title, phone number, and address (if different from your company's main address).

Worksoft Help Portal

The Worksoft Help Portal page provides you:

- ▶ The latest product documentation for Worksoft products and solutions.
- ▶ Helps you "Get Started".
- ▶ Provides product and solution "How To" instructions.

To utilize the Worksoft Help Portal, log on to the site: <http://docs.worksoft.com/>

Anyone at your company can have access to this site.

To request a Worksoft Support portal account login, send an email to support@worksoft.com and include the following information in your email: full name, email address, department/company/(and Client Name if you are a Worksoft partner), job title, phone number, and address (if different from your company's main address).



Job Aid: Techniques Used when Defining Processes



For convenience, you may remove these pages from your guide.

To create a process:

1. From the Processes window, in the Navigation tree, click the **1.4 Sandbox** folder, and then click your name.
2. Click the desired process folder.
3. Right-click in the **Summary** pane and select **New Process**.
4. In the **Process** section, in the **Name** field, type in the name.
5. In the **Description** field, type in the description.
6. Press the **Save** button
7. Click the **Steps** tab.

To use LiveTouch to replace current step:

1. Verify that the application is open to the screen you wish to learn.
2. Right-click in the **Steps** area and select **New**.
3. Click the **Application Version** drop-down arrow.
4. Select **Select Using LiveTouch**.
5. In the application, select the object field.

6. In the Certify LiveTouch dialog box, click **Save & Close**.
7. To Execute the step right-click the **step** and select **Execute Step**.

To use LiveTouch to insert one or many steps:

1. In Certify, in the Steps area, right-click on **Step** you want to insert after.
2. Select **Insert Step Below Using LiveTouch**.
3. In the application, select the objects by clicking in the object fields.
4. In the Certify LiveTouch dialog box, click **Save & Close**.
5. The Certify Process and Data Editor appears.
6. In Certify, in the Parameters tab, enter any parameters.
 - a. Click the **Save**  button.
7. Once all steps are entered, click the **Save**  button.
8. Click the red  in the top right corner to exit the Process Editor.

To execute the entire process:

1. In the Summary pane, right-click the **process** and select **Run**.
2. The Configuration dialog box appears.
3. Click **Start**.
4. The Execution dialog box appears.
5. Click **Run**.
6. The process executes and upon completion, the Result Viewer appears.
7. When finished reviewing the results, click the red  in the top right corner to close the Result Viewer.



Job Aid: Starting a New Integrated Process

1. Create a folder to hold all of the processes associated with this integrated process.
2. Create a top level process in the folder.
3. Create child processes by:
 - a. Using Capture Process to create the entire process.
 - b. Using “New Process” or Execute Process then use Insert with Livetouch or Insert with Capture.
4. If you haven’t already done so – link your child processes to your top level process.
5. Add a call to a utility like UTL_Login to the top of your process to login to your application.
6. Create Layout/Recordsets for each process using “Add to Layout” or one of the other Layout techniques.
7. Make sure that you have replaced any data with variables and put the variables in a layout or recordset.

Job Aid: Creating Layouts Manually

Creating a layout manually involves adding existing or creating new variables for the layout. One reason for creating layouts manually is to test a specific area of your application. For example, the Worksoft Sample Application has seven input fields including PO Number, Name, Ship to, Bill to, Material, Quantity, Price. You want to test only three input fields (Name, Ship to, Bill to) to ensure the information you are collecting is valid or invalid. In this case, you may want to create a layout with these three fields only and create a process that only tests Name, Ship to, and Bill to. One option is to create the layout manually and use the add capability to select the variables.

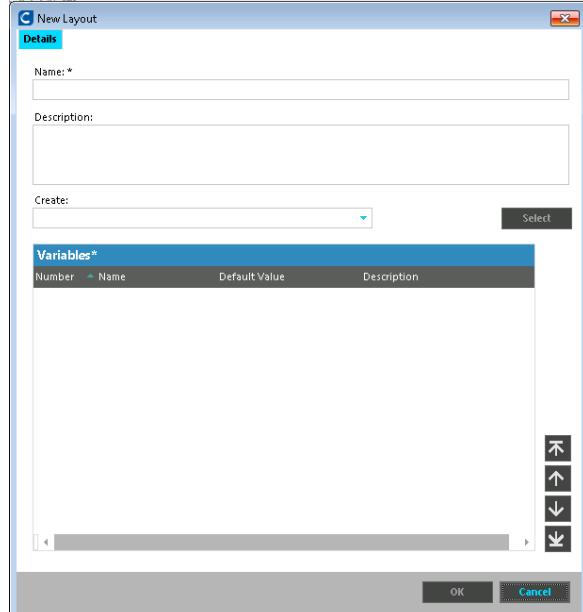
Step	Action
------	--------

1. In the Navigation taskbar, click **Data**.

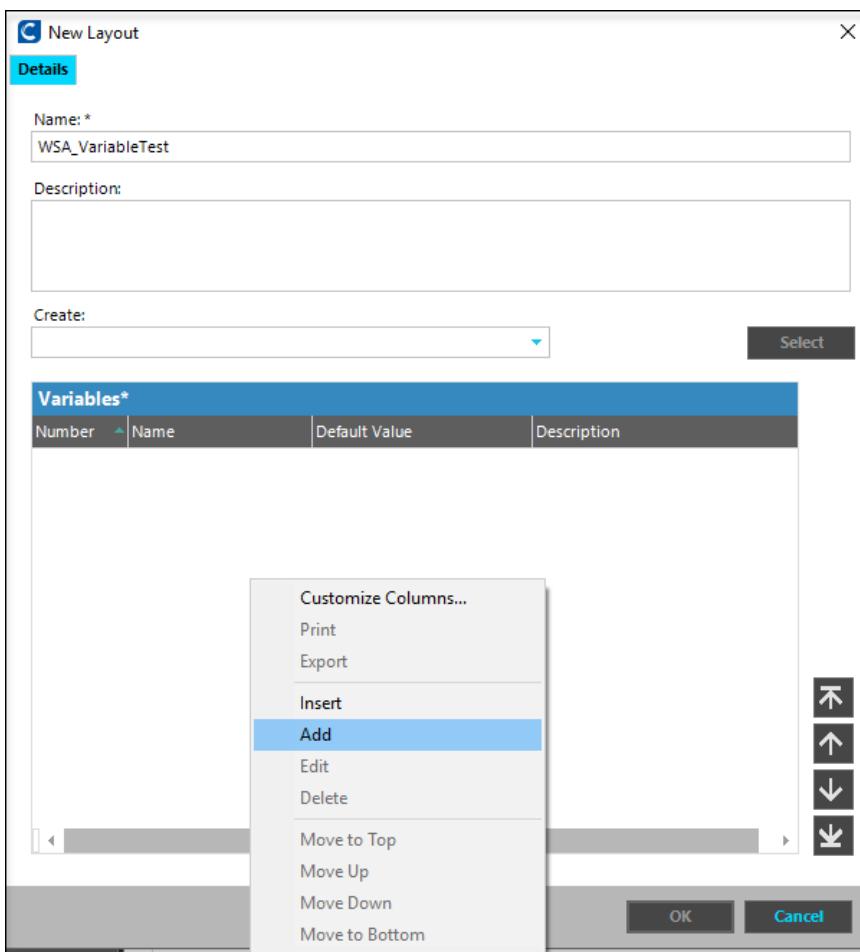
The Data window appears.

2. In the Navigation tree, select **your Sandbox** folder, and then select **your WSA_CreateandModify** folder.
3. Right-click in the Summary pane, and select **New Layout**.

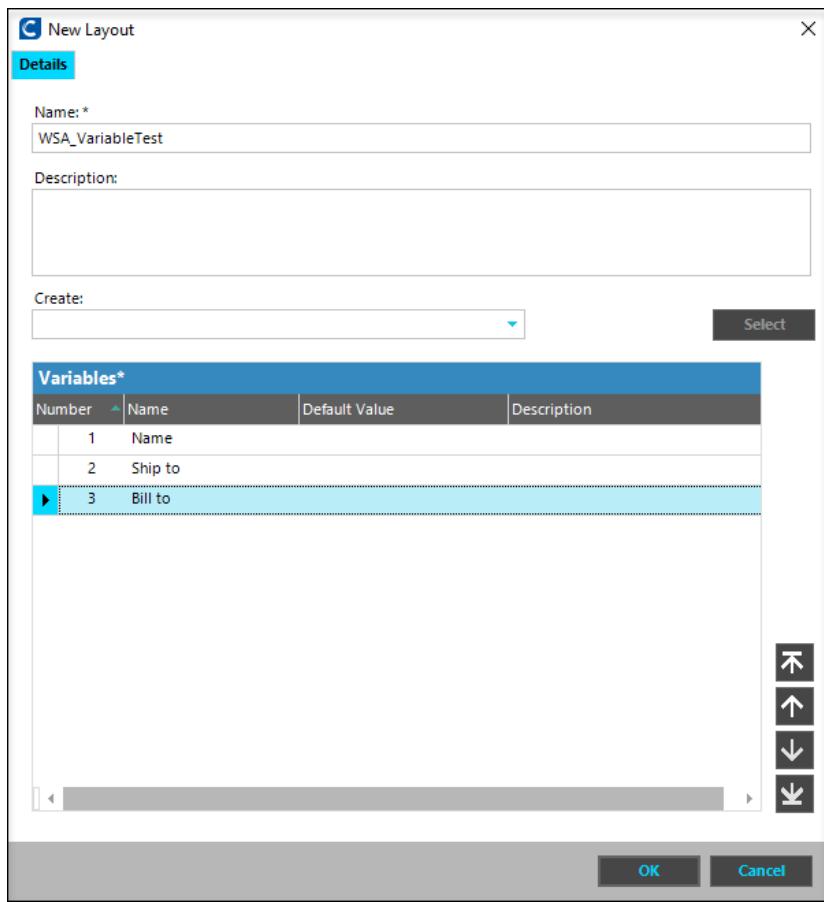
The New Layout dialog box appears.



4. In the **Name** field, type **WSA_VariableTest**.
5. Right-click in the Variables Pane, and select **Add**.



6. Select the **Name** variable.
7. Click **OK**.
8. Right-click in the Variables Pane, and select **Add**.
9. Select the **Ship to** variable.
10. Click **OK**.
11. Right-click in the Variables Pane, and select **Add**.
12. Select the **Bill to** variable.
13. Click **OK**.



14. Click **OK** to save the new layout, and close the New Layout dialog box.

Job Aid: Creating Layouts Using Processes

Creating layouts using processes involves selecting one or more processes from the Select Process dialog box. The layout is generated based on the variables used in the selected processes. In this exercise, you will create a layout called WSA_EditandDelete by selecting the WSA_EditandDelete end-to-end process. All the variables used in this process and child processes are added to the layout. ****Note: You will need to complete Lesson 7: Developing Advanced Processes to complete this exercise.**

Step	Action								
1.	In the Navigation Taskbar, click Data .								
2.	In the Navigation Tree, navigate to your WSA_EditandDelete folder.								
3.	Right-click in the Summary Pane and select New Layout .								
	<i>The New Layout dialog box appears.</i>								
4.	In the New Layout dialog box, type in field values or select options as follows:								
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #d3d3d3;"> <th style="text-align: left;">Field/Option</th><th style="text-align: left;">Value/Action</th></tr> </thead> <tbody> <tr> <td>Name</td><td>WSA_EditandDelete</td></tr> <tr> <td>Create</td><td>From Processes and Child Processes</td></tr> <tr> <td>Child Process Depth</td><td>1</td></tr> </tbody> </table>		Field/Option	Value/Action	Name	WSA_EditandDelete	Create	From Processes and Child Processes	Child Process Depth	1
Field/Option	Value/Action								
Name	WSA_EditandDelete								
Create	From Processes and Child Processes								
Child Process Depth	1								

5. Click the **Select** button.

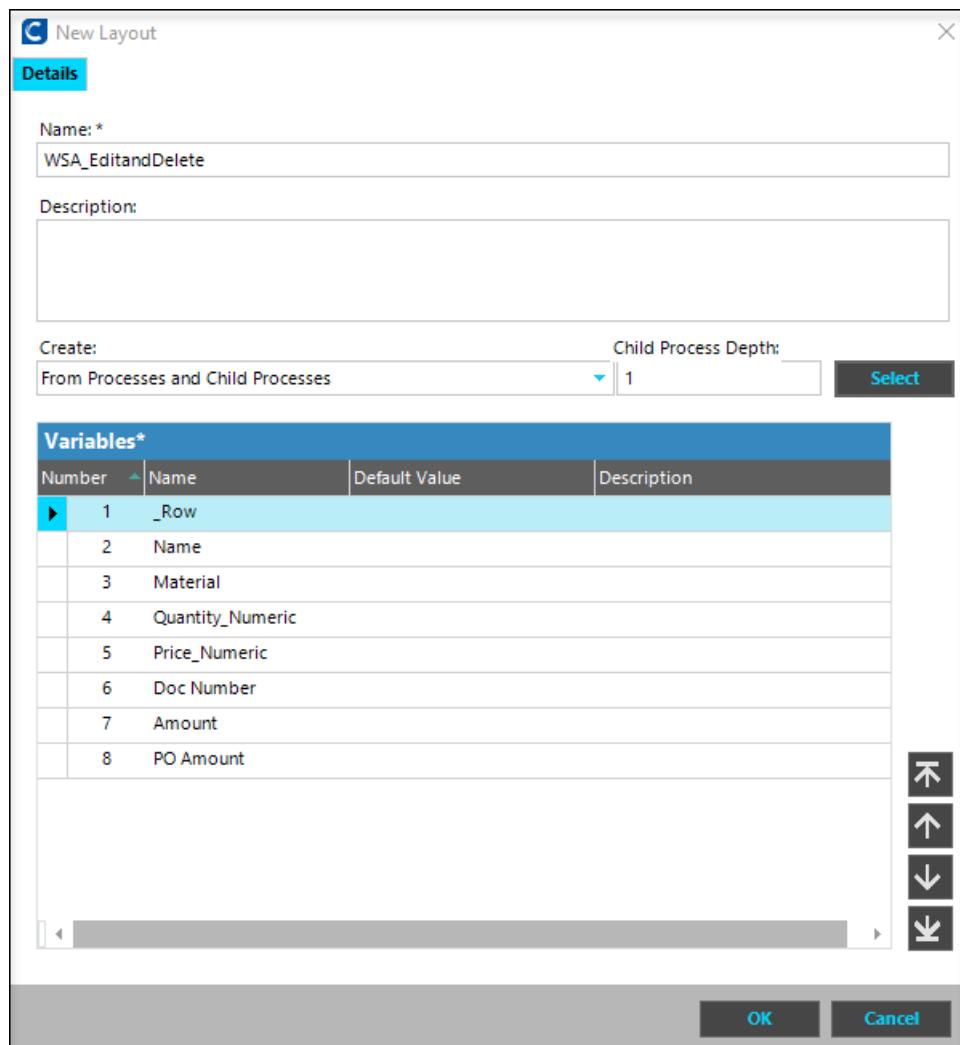
The Select Process dialog box appears.

6. In the Navigation Tree, click your **WSA_EditandDelete** folder
 7. In the Summary Pane, select the **WSA_EditandDelete** process.

The process is highlighted in the Summary Pane.

8. Click **OK**.

The Select Process dialog box closes and all variables that were used in the WSA_EditandDelete process and child processes are placed in the Variables pane in the New Layout dialog box.



9. Right-click the **_Row** variable, and delete. You do not need this variable.
10. Click **OK** to save the new layout, and close the New Layout dialog box.

Job Aid: Creating a Recordset

With the WSA_EditandDelete layout defined, you are now ready to create a recordset. In this exercise, you will select the WSA_EditandDelete layout, create a recordset called WSA_EditandDelete, and add two rows of data.

Step	Action
------	--------

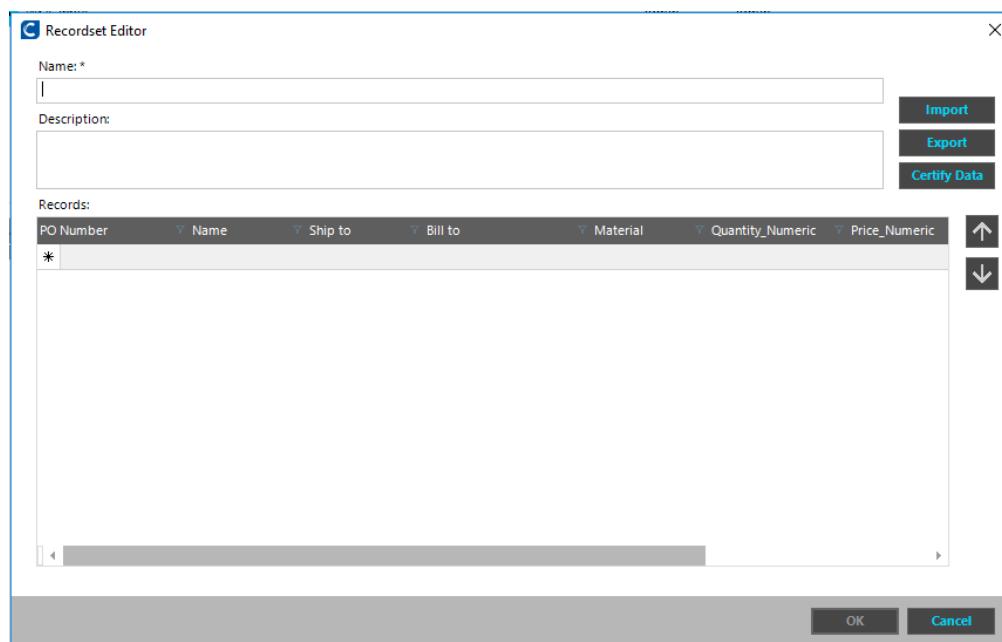
If you are continuing directly from the previous Job Aid, you can skip to step 4.

1. In the Navigation Taskbar, click **Data**.
2. In the Navigation Tree, navigate to your **WSA_EditandDelete** folder.
3. In the Summary Pane, select the **WSA_EditandDelete** layout.
4. In the Detail Pane, click the **Recordset** tab.
5. Right-click in the Recordsets tab.

A shortcut menu appears.

6. Select **New Recordset**.

The Recordset Editor dialog box appears.



7. In the New Recordset dialog box, type in field values as follows:

Field	Value
Name	WSA_EditandDelete
Description	Recordset to retain data after execution

Enter the data as shown in the following screenshot:

The screenshot shows the 'Recordset Editor' dialog box. At the top left is the title 'Recordset Editor'. To the right are three buttons: 'Import', 'Export', and 'Certify Data'. Below the title is a 'Name:' field containing 'WSA_EditandDelete'. Underneath it is a 'Description:' field containing 'Recordset to retain data after execution'. A large table titled 'Records:' follows, with columns labeled 'Name', 'Material', 'Quantity_Numeric', 'Price_Numeric', 'Doc Number', 'Amount', and 'PO Amount'. The table contains two rows of data: row 1 with 'Name' 'Mary Wilson' and row 2 with 'Name' 'John Smith'. The 'John Smith' row is currently selected, indicated by a blue highlight. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

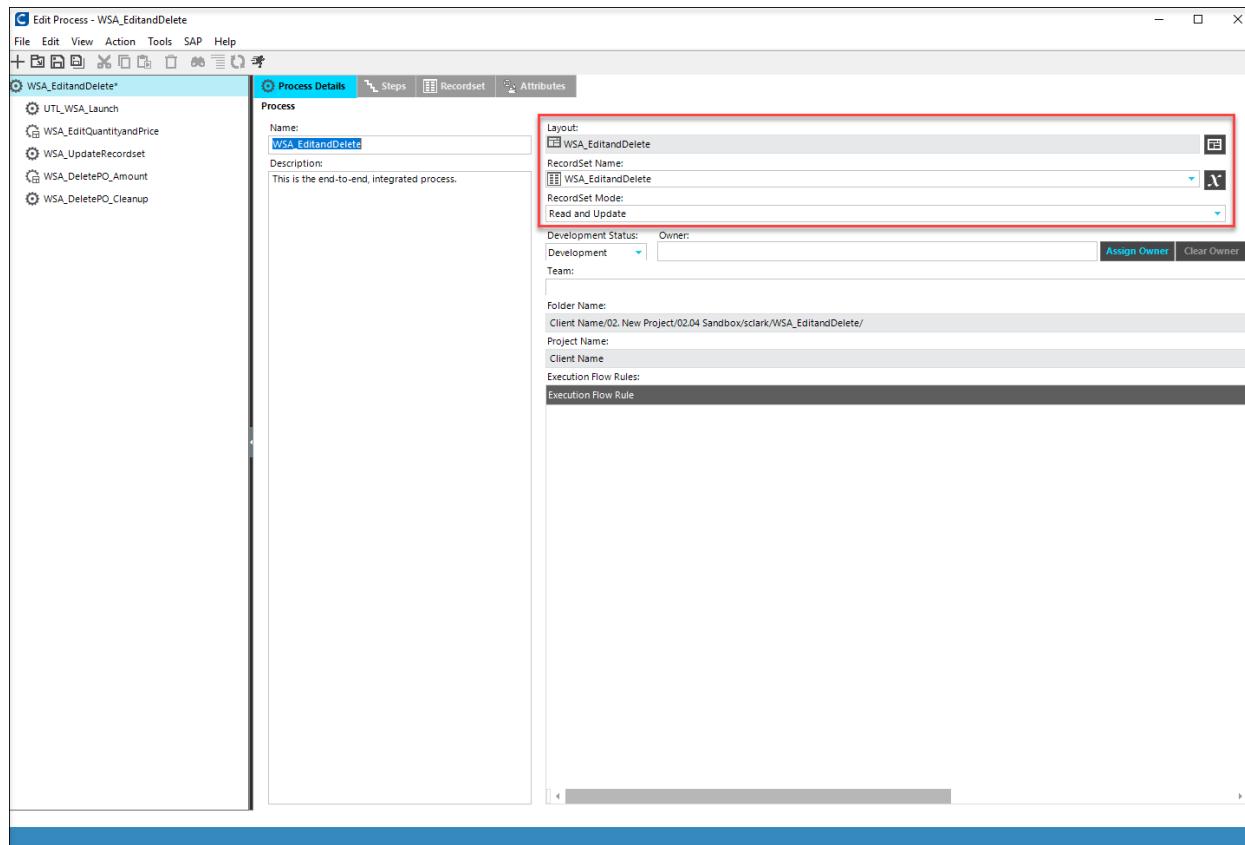
8. Click **OK** to save the recordset, and close the Recordset Editor.

Job Aid: Adding Layouts and Recordsets to an Existing Process

After defining a layout and recordset, you can attach them to processes or process steps in the Process Editor. Each process can have one layout and recordset attached. Attaching recordsets is done by selecting a layout, a recordset, and the mode of operation. The level at which you attach the layout determines whether a loop will occur, and the mode determines how it will be executed. For example, if a recordset is attached to a process in Read mode, then the process will loop once for each row of data values. In this exercise, you will apply the layout and recordset created in the previous Job Aids to the WSA_EditandDelete process.

Step	Action
1.	In the Navigation toolbar, click Processes .
	<i>The Processes window appears.</i>
2.	In the Navigation Tree, expand Your Sandbox Folder .
3.	Select your WSA_EditandDelete folder.
4.	In the Summary Pane, right-click the WSA_EditandDelete process, and select Edit .
5.	On the right side of the screen, in the Layout field, click the Select Layout  button.
	<i>The Select Layout dialog box appears.</i>
6.	In the Summary Pane, select WSA_EditandDelete layout.
7.	Click OK .

Lesson 5: Resources



8. On the right side of the screen, in the **RecordSet Name** field, verify **WSA_EditandDelete** is selected.
9. On the right side of the screen, in the **RecordSet Mode** field, verify that **Read and Update** is selected. This will read the Name field and update the other variables after execution.
10. Click the **Save**  button.
11. Click the **Close** button to exit the Process Editor.



Course Evaluation

At Worksoft, Inc. we continually strive to improve the quality of our training programs and documentation. Your feedback is important in helping us to provide the best possible training experience for our clients. Please take a moment to complete this course evaluation.

Company: _____

Course Name: _____ Your Job Role: _____

Date Attended: _____ Instructor: _____

About the Course

Overall...	Excellent	Satisfactory	Fair	Poor
How would you rate the class?				
How would you rate the course format?				
How would you rate the instructor?				
How would you rate the training materials?				

How would you rate the pace of the course?

Just Right Reasonable Too Slow Too Fast

Overall, what percentage of the covered material was completely new to you?

0 to 25% up to 50% up to 75% All

Do you feel that you are now adequately prepared to implement and/or use Certify?

Yes No

What did you like **most** about the course?

What did you like **least** about the course?

Would you prefer to have more hands-on time or more lecture time? Why?

Are there any topics covered that you would have liked to spend more time on?

Are there any topics not covered that you would like to know more about?

Would you recommend this course to others? If not, why not?

About the Instructor

Did the instructor demonstrate excellent technical expertise on the product?

Yes No

Did the instructor demonstrate industry knowledge and provided adequate explanations?

Good Average Poor

Did the instructor present the material and answer questions clearly?

Good Average Poor

Was the instructor well-prepared?

Yes No

About the Materials

Is the training guide comprehensive for the course?

Yes No

Will you use the training guide as a reference after the class?

Yes No

Additional comments or suggestions overall

Worksoft Inc.

15851 N. Dallas Parkway, Suite 855
Addison, TX 75001
www.worksoft.com