Foxtrot® powered by ENABLESOFT

FOXTROT ESSENTIALS

For Beginners

Intended for users just starting to learn Foxtrot. This guide will cover topics available for use in Foxtrot's 'Beginner' user level.

Foxtrot Suite Version 14.0.1

Table of Contents

Product Introduction

EXAMPLE USE CASES

- Data Entry
  - → Creating new customer accounts
  - → Posting debits & credits
  - → Applying late fees
  - → Ordering or closing debit & credit cards
  - → Budget input
  - → Establishing employee benefit options
- Data Maintenance and Cleansing
  - → Changing & updating codes
  - → Changing overdraft fees
  - → Consolidating accounts
  - → Adjusting interest rates
  - → Data validation & cleanup
- Data Application / Integration
  - → Integrating card ordering system with core
  - → Integrating multiple third-party software solutions
- Data Migration
  - → Loan onboarding
  - → General ledger posting
  - → Integrating cash management functions
- Data Mining and Aggregation
  - → Researching and reporting on competitive rates
  - → Ad hoc reporting
  - → Compiling transaction history
- Data Testing
  - → Monitoring online banking systems and trends
  - → Application monitoring and error reporting
- Other Data-Related Tasks
  - → Monitoring server status
  - → Automating user provisioning
  - → Automating file transfers
  - → Uploading & downloading to/from an FTP server
  - → Scheduled maintenance of files & folder
  - → Much, much more.

THE FOXTROT SUITE

- Foxtrot works like an automated employee, intelligently performing data entry, maintenance, integration, migration and virtually anything else that can be performed with a mouse and keyboard, automatically and with total accuracy. Foxtrot can be used in tandem with any web, windows or legacy application.

- FoxBot is used to execute any projects created within Foxtrot. It will allow you to scale up automation in your organization. These runtime bots do one thing – work until the work is done.

- FoxHub allows you to manage your workforce of FoxBots, and is the central place to orchestrate your automation Jobs. View all of your Bots and know what they are doing at all times. Put them to work by adding a Job for an individual Bot, or by adding a team Job for multiple Bots to complete.

- Foxtrot Administrator is a control center used to manage licenses, security, settings, and global preferences.

ACTIVATING FOXTROT LICENSES

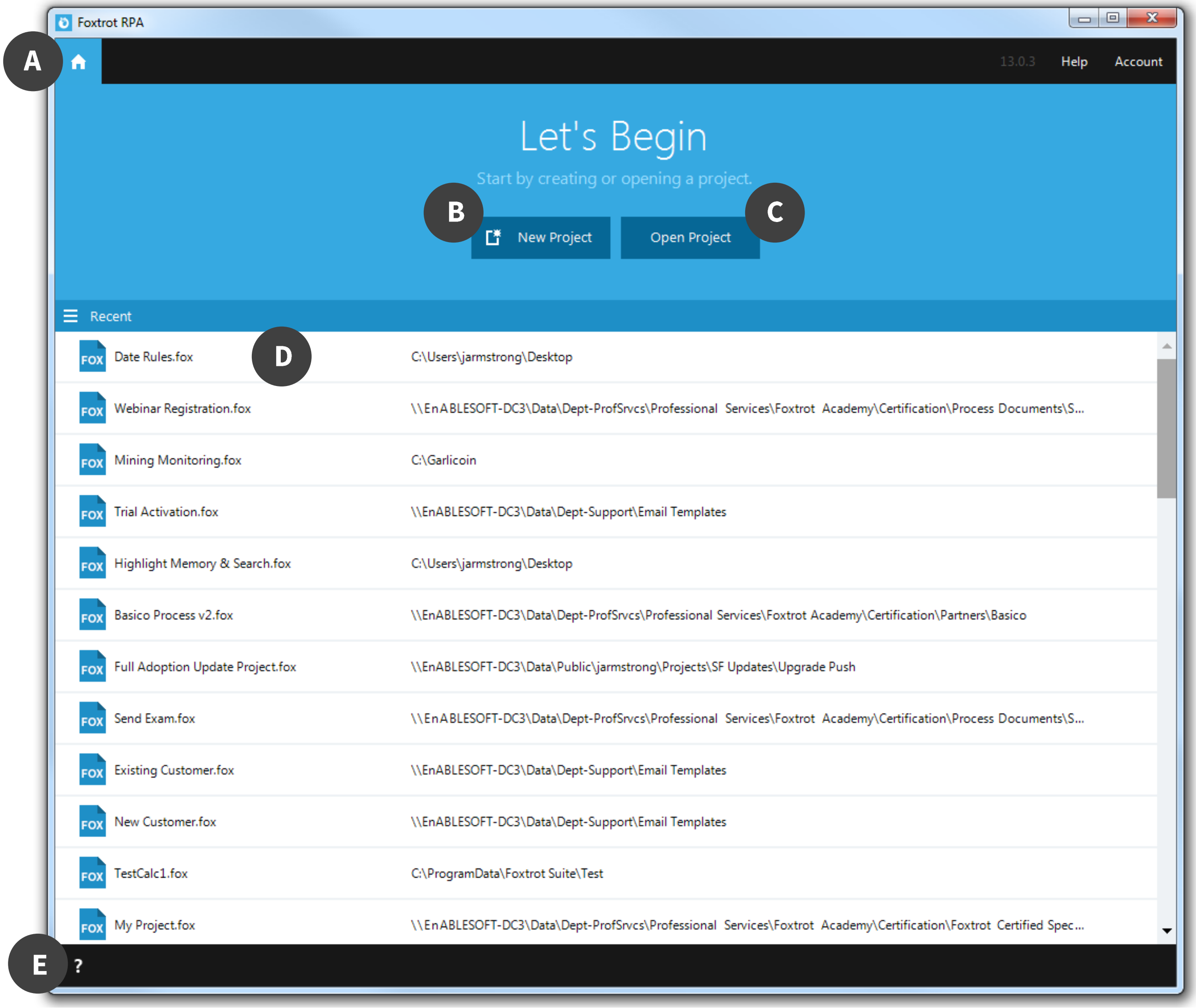
Before starting a project, any Foxtrot Suite license(s) must first be activated. First, determine what type of license model works best within the institution. Foxtrot Enterprise supports both local and central deployment models. There are slight differences in the activation process between these two models.

To learn more, please reference the Foxtrot Deployment Guide, linked on the Latest Version page on our website.

Meet Foxtrot

HOME SCREEN

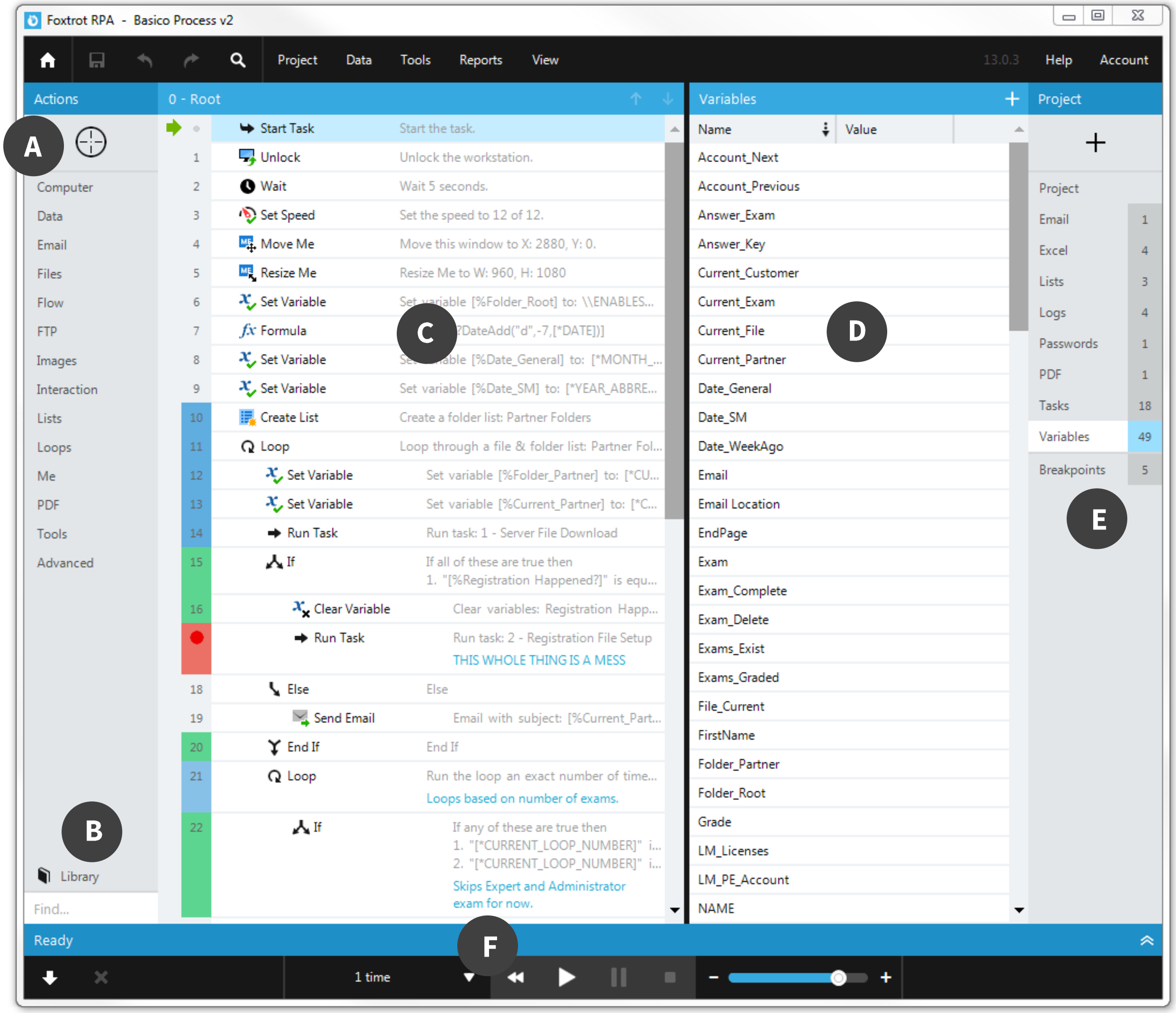
Welcome to Foxtrot. Easy automation begins at the Home Screen. You can access this at any time with the Home Button (A). Click on New Project (B) to create something new, or Open Project (C) to automate something you've already created. Your recently viewed projects can be found in the Recent Projects (D) list. And if you get stumped with anything, you can click the Help Button (E) to access Foxtrot's Help menu.



Date Rules.fox	C:\Users\jarmstrong\Desktop
Webinar Registration.fox	\\EnABLESOFT-DC3\Data\Dept-ProfSrvcs\Professional Services\Foxtrot Academy\Certification\Process Documents\S...
Mining Monitoring.fox	C:\Garlicoin
Trial Activation.fox	\\EnABLESOFT-DC3\Data\Dept-Support\Email Templates
Highlight Memory & Search.fox	C:\Users\jarmstrong\Desktop
Basico Process v2.fox	\\EnABLESOFT-DC3\Data\Dept-ProfSrvcs\Professional Services\Foxtrot Academy\Certification\Partners\Basico
Full Adoption Update Project.fox	\\EnABLESOFT-DC3\Data\Public\jarmstrong\Projects\SF Updates\Upgrade Push
Send Exam.fox	\\EnABLESOFT-DC3\Data\Dept-ProfSrvcs\Professional Services\Foxtrot Academy\Certification\Process Documents\S...
Existing Customer.fox	\\EnABLESOFT-DC3\Data\Dept-Support\Email Templates
New Customer.fox	\\EnABLESOFT-DC3\Data\Dept-Support\Email Templates
TestCalc1.fox	C:\ProgramData\Foxtrot Suite\Test
My Project.fox	\\EnABLESOFT-DC3\Data\Dept-ProfSrvcs\Professional Services\Foxtrot Academy\Certification\Foxtrot Certified Spec...

MAIN INTERFACE

Once a project is opened, Foxtrot will provide access to all of its controls and functionalities.



	Action	Description
	Start Task	Start the task.
1	Unlock	Unlock the workstation.
2	Wait	Wait 5 seconds.
3	Set Speed	Set the speed to 12 of 12.
4	Move Me	Move this window to X: 2880, Y: 0.
5	Resize Me	Resize Me to W: 960, H: 1080
6	Set Variable	Set variable [%Folder_Root] to: \\ENABLES...
7	Formula	?DateAdd("d",-7,[*DATE])]
8	Set Variable	Set variable [%Date_General] to: [*MONTH_...
9	Set Variable	Set variable [%Date_SM] to: [*YEAR_ABBRE...
10	Create List	Create a folder list: Partner Folders
11	Loop	Loop through a file & folder list: Partner Fol...
12	Set Variable	Set variable [%Folder_Partner] to: [*CU...
13	Set Variable	Set variable [%Current_Partner] to: [*C...
14	Run Task	Run task: 1 - Server File Download
15	If	If all of these are true then 1. "[%Registration Happened?]" is equ...
16	Clear Variable	Clear variables: Registration Happ...
●	Run Task	Run task: 2 - Registration File Setup THIS WHOLE THING IS A MESS
18	Else	Else
19	Send Email	Email with subject: [%Current_Part...
20	End If	End If
21	Loop	Run the loop an exact number of time... Loops based on number of exams.
22	If	If any of these are true then 1. "[*CURRENT_LOOP_NUMBER]" i... 2. "[*CURRENT_LOOP_NUMBER]" i... Skips Expert and Administrator exam for now.

Name	Value
Account_Next	
Account_Previous	
Answer_Exam	
Answer_Key	
Current_Customer	
Current_Exam	
Current_File	
Current_Partner	
Date_General	
Date_SM	
Date_WeekAgo	
Email	
Email Location	
EndPage	
Exam	
Exam_Complete	
Exam_Delete	
Exams_Exist	
Exams_Graded	
File_Current	
FirstName	
Folder_Partner	
Folder_Root	
Grade	
LM_Licenses	
LM_PE_Account	
NAME	

Project	
Email	1
Excel	4
Lists	3
Logs	4
Passwords	1
PDF	1
Tasks	18
Variables	49
Breakpoints	5

A) Actions are created from the **Action Panel**. All actions are listed here, either after targeting something with the Bullseye or in the list below it.

B) The **Action Library** will list every action Foxtrot offers, and provides a link to the Help article for each. The **Library Search** lets you search through all Actions for what you need.

C) The **Task Pane** displays actions in the current active Task within the project.

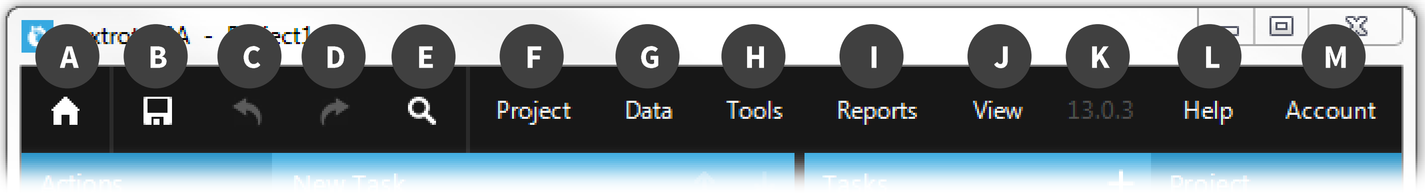
D) Items such as Tasks, Variables and Data are found in the **Project Pane**.

E) The **Item List** provides access to all items within your project, such as variables, tasks and connections.

F) The **Run Bar** provides Run options and Statistics for your project.

MENU BAR

Foxtrot's Menu Bar provides a variety of menus allowing you to access various commands. Some menu items will be disabled until a project is opened. Some menu items may also be disabled depending on the permission level set up on your Foxtrot user profile.

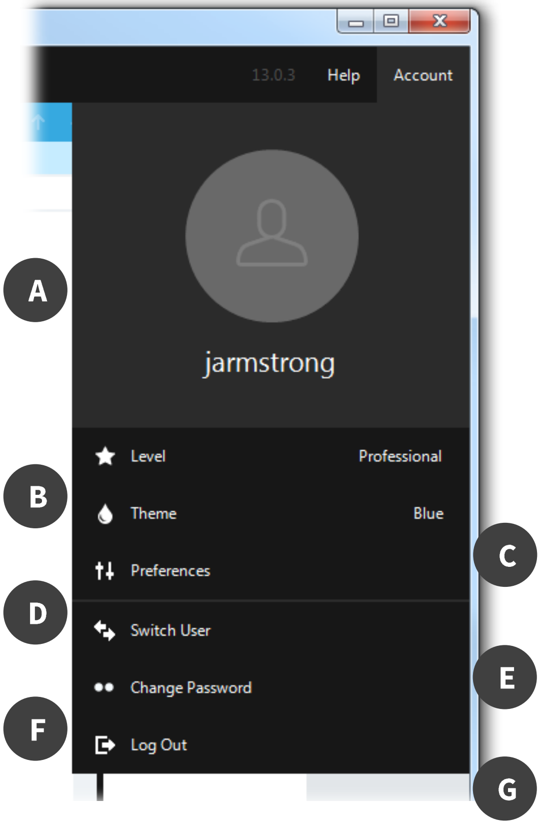


A) **Home** returns to Foxtrot's home screen.
B) **Save** will save the current project. Supports Ctrl+S keyboard shortcut.
C) **Undo** your most recent changes. Supports Ctrl+Z keyboard shortcut.
D) **Redo** your previously undone action. Supports Ctrl+Y keyboard shortcut.
E) **Find** an item in your project. Supports Ctrl+F keyboard shortcut.
F) **Project** to create a new project, open a project save your current project or exit Foxtrot.
G) **Data** is where to open and close data, as well as mark and unmark records in your current data.
H) **Tools** gives access to the Image Editor and Mouse Recorder
I) **Reports** will create a PDF report of tasks, variables or passwords within the current project.
J) **View** provides options to alter the view of Foxtrot.
K) **Current Version** displays the current version of Foxtrot.
L) **Help** to access the help file, check for updates or the look at new features.
M) **Account** to alter information about the current account, including the profile picture, mode and theme.

ACCOUNT PANEL

Remember, Foxtrot is your personal automated employee. You can customize the look and feel of Foxtrot. The Account Panel can be accessed through the Account menu.

A) A **Profile Image** may be added to a Foxtrot account. By clicking the default image, the Image Editor will appear. You can simply upload an image, and modify it as needed.



jarmstrong

Level Professional

Theme Blue

B) There are three **Levels** to operate Foxtrot: Beginner, Professional and Expert. When first opening Foxtrot you will be prompted to choose your comfort level. You can change this at any time. Beginner Mode provides access to the essential features of Foxtrot. Professional is a step above, and unlocked additional features and actions. The Expert level gives access to programming tools within Foxtrot that a "normal" user might not need.

C) The color of Foxtrot may be changed using the **Theme** option. There are 3 themes available: the default Blue, Raspberry and Dark. This does not affect performance.

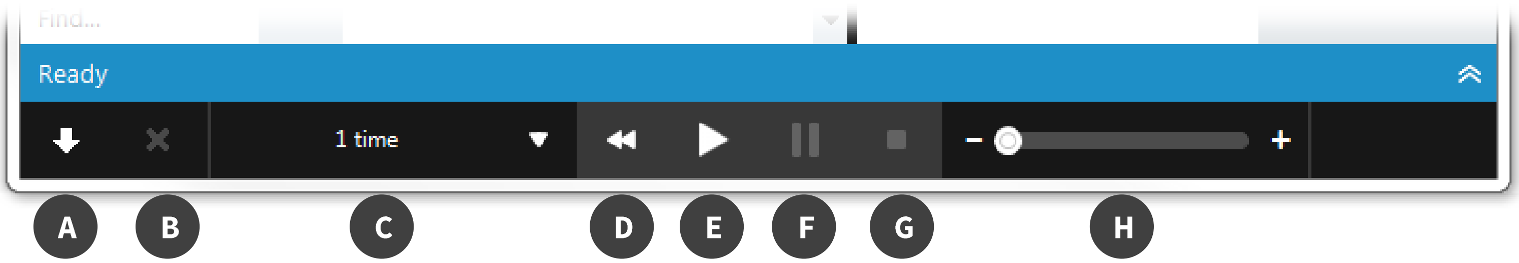
D) All action defaults are fully customizable through Foxtrot's **Preferences**. Default folder locations, run settings, timing and targeting options can be configured here.

E) If user logins are enabled, the **Switch User** option will be presented. This will allow a user to switch to another account without closing the project and data.

F) If user logins are enabled, the **Change Password** option will allow a user to change their password.

G) If user logins are enabled, the **Log Out** option will return to the user login screen.

RUN BAR



A) The **Execute** button will execute the current action without running the entire process.

B) The **Break** button will stop executing the action in progress and stop a running project.

C) The **Run Options** provide access to additional run settings.

D) The **Rewind** button moves the execution arrow to the beginning of the project.

E) The **Play** button runs the project from the location of the execution arrow

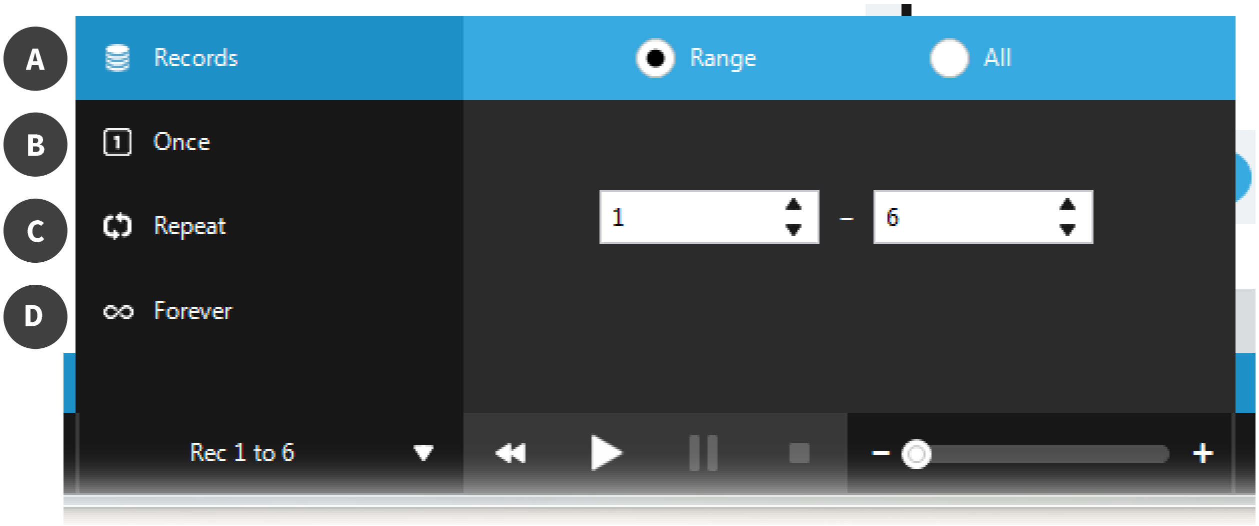
F) The **Pause** button is only active when a project is running. When pressed, the project will stop immediately.

G) The **Stop** button will stop the project when it reaches the End Task action in the Startup Task.

H) The **Speed Slider** controls the speed of a running project. The speed may be adjusted by using the increase or decrease speed buttons. The handle on the slider can also be used to quickly change the speed.

RUN OPTIONS

Run Options alter the way Foxtrot runs a project. There are four modes to choose from.



A) **Records:** a database must be imported into Foxtrot in order to use this option. This will run a project through the imported database. Depending on the parameters set, Foxtrot may run all of the records or just a subset. When Foxtrot reaches the *End Task* action, it will automatically do the following:

   A) Mark the record as being Done.
   B) Go to the next Pending record in the database (if any).
   C) Navigate to the beginning of the project.

   After all records have been processed, Foxtrot will automatically stop the project.

B) **Once:** runs a project one time. There are no additional options, as this will NOT mark any records. If there is data imported, Foxtrot will process the current record.

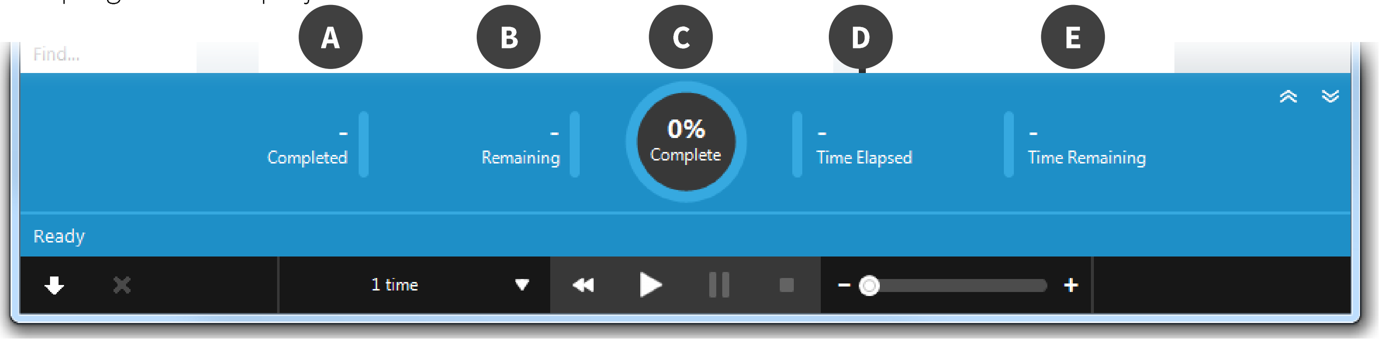
C) **Repeat:** runs the project a set number of times. This option will NOT mark any records. If there is data imported, Foxtrot will process the current record.

D) **Forever:** if this option is selected, the project will only stop if one of the following scenarios is encountered:

- The user stops the project manually
- An error occurs in the project causing the project to stop
- When one of the following Actions are encountered:
  - → *Close Me*
  - → *Lock* (if the action is configured to stop the project)
  - → *Log Off*
  - → *Reboot*
  - → *Shut Down*
  - → *Stop Task*

STATS BAR

While running through a set of data, Foxtrot will track many real-time stats about the progress of the project.



A) **Completed** displays the number of records or iterations the project has successfully completed.
B) **Remaining** shows the number of records or iterations the project has remaining before completed
C) **% Complete** shows the current percentage of records or iterations completed
D) **Time Elapsed** displays how long the project has been running.
E) **Time Remaining** supplies an estimation of remaining time until the entire process is completed

Importing Data

There are two methods to automate a process driven by a pre-existing dataset – by importing data into Foxtrot, or by working directly with the Excel Workbook. The latter will be reviewed in the Excel Automation chapter. This chapter will review the supported data formats for import as well as the methods required for import.

SUPPORTED DATA FORMATS

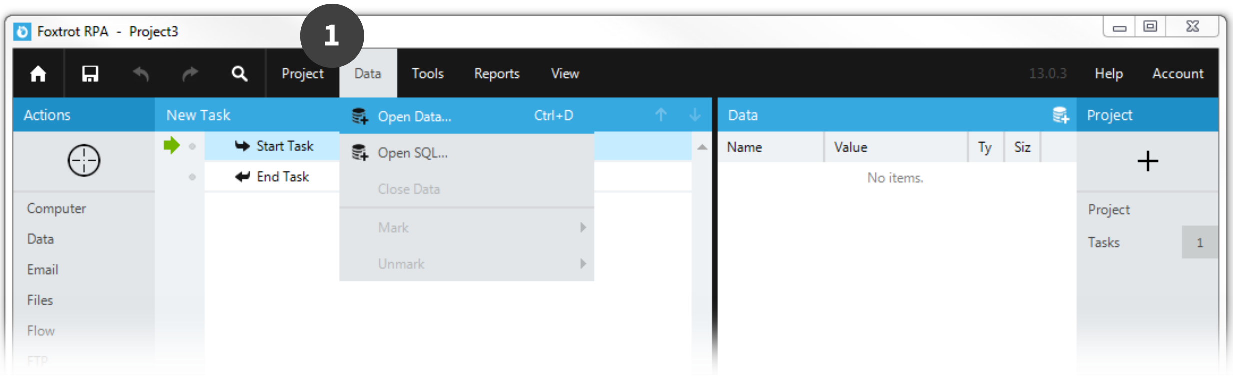
Foxtrot supports the following files for import:

- Microsoft Access databases (.ACCDB or .MDB)
- Microsoft Excel workbooks (.XLSX, .XLS, .XLSM)
- Comma separated values (.CSV)
- Text files (.TXT)
- Dbase file (.DBF)
- XML file (.XML)
- Microsoft SQL Server and MySQL Databases

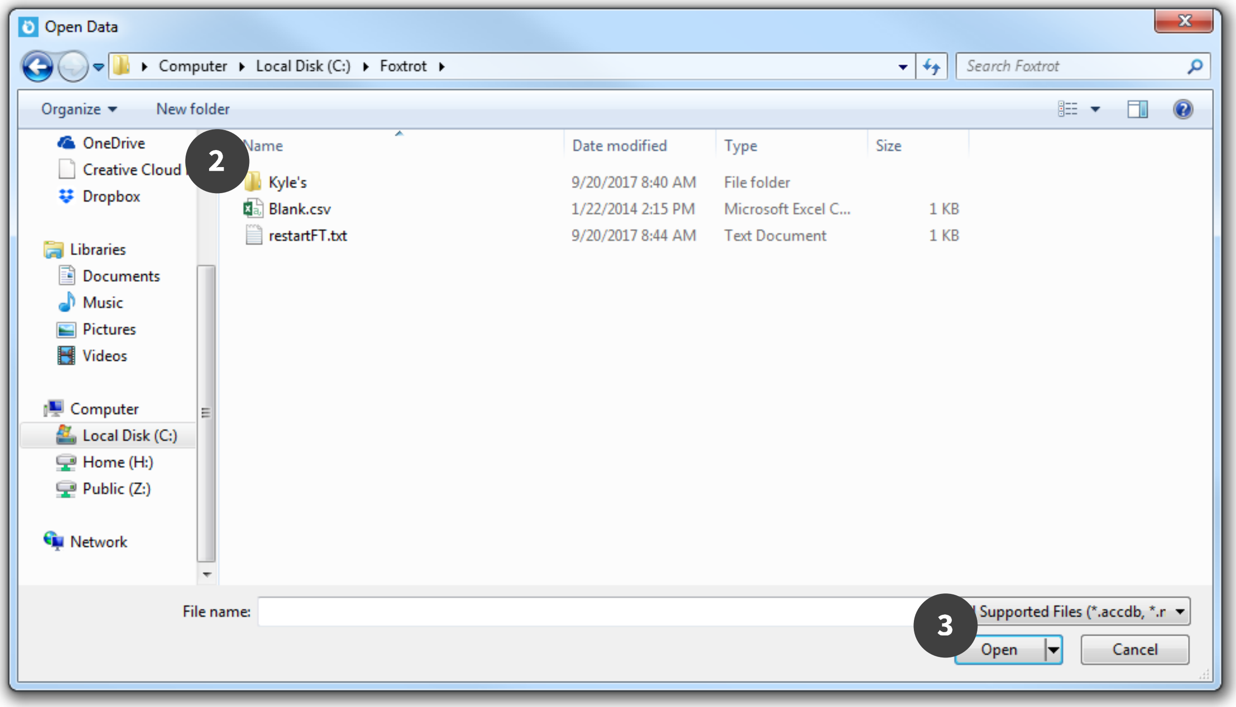
Foxtrot will work directly with .ACCDB, .MDB and .DBF files in their original format. For all other formats, Foxtrot will read the original file and create its own .ACCDB file. Foxtrot will not alter the original database for non-database file types.

IMPORTING EXCEL FILES INTO FOXTROT

1) On the top menu bar, click *Data* followed by *Open Data.* Alternatively, you can press *Ctrl+D*.



2) Navigate to and select the correct file to import.

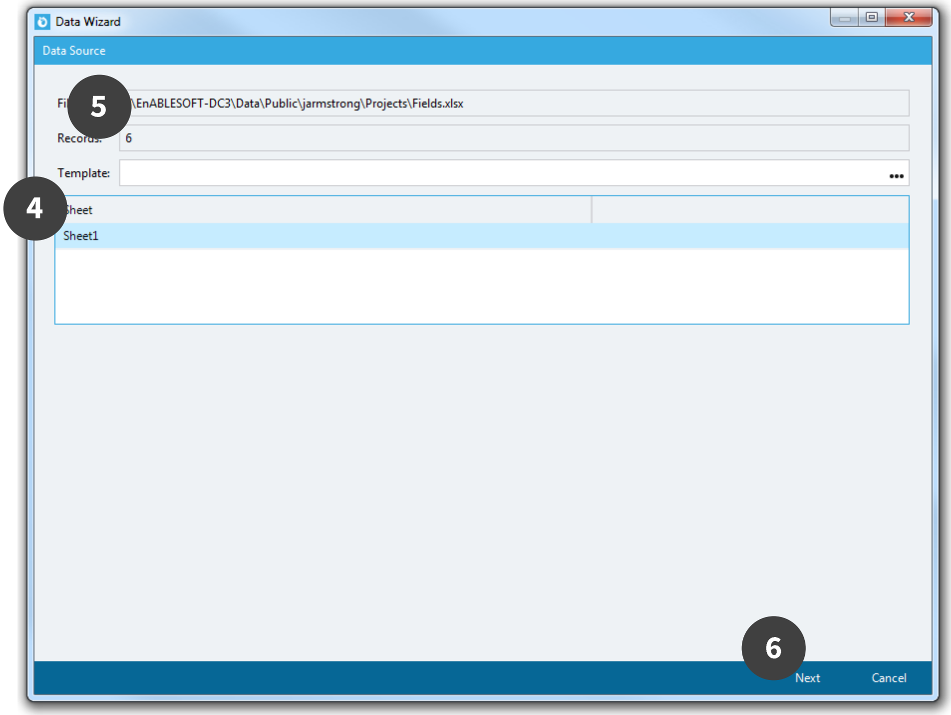


Name	Date modified	Type	Size
Kyle's	9/20/2017 8:40 AM	File folder	
Blank.csv	1/22/2014 2:15 PM	Microsoft Excel C...	1 KB
restartFT.txt	9/20/2017 8:44 AM	Text Document	1 KB

3) Click *Open* to display the Data Wizard.

4) Select the worksheet to import.

5) Verify the total number of records.



Data Wizard

Data Source

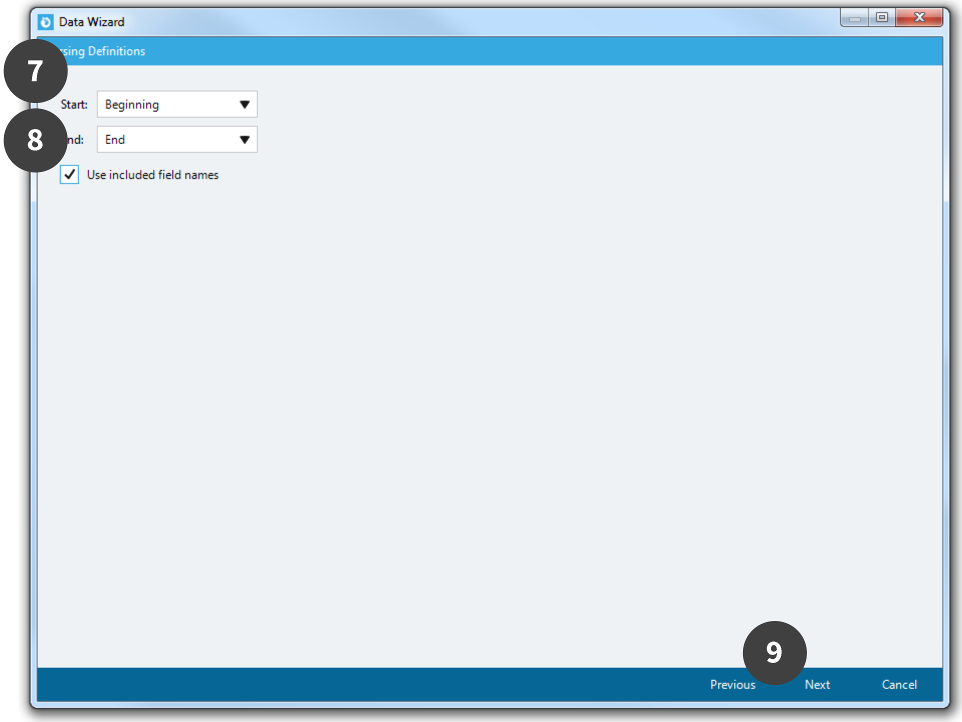
Fi \EnABLESOFT-DC3\Data\Public\jarmstrong\Projects\Fields.xlsx

Records: 6

Template:

heet	
Sheet1	

6) Click *Next* to continue to Step 2: Parsing Definitions.



Data Wizard

sing Definitions

Start: Beginning

nd: End

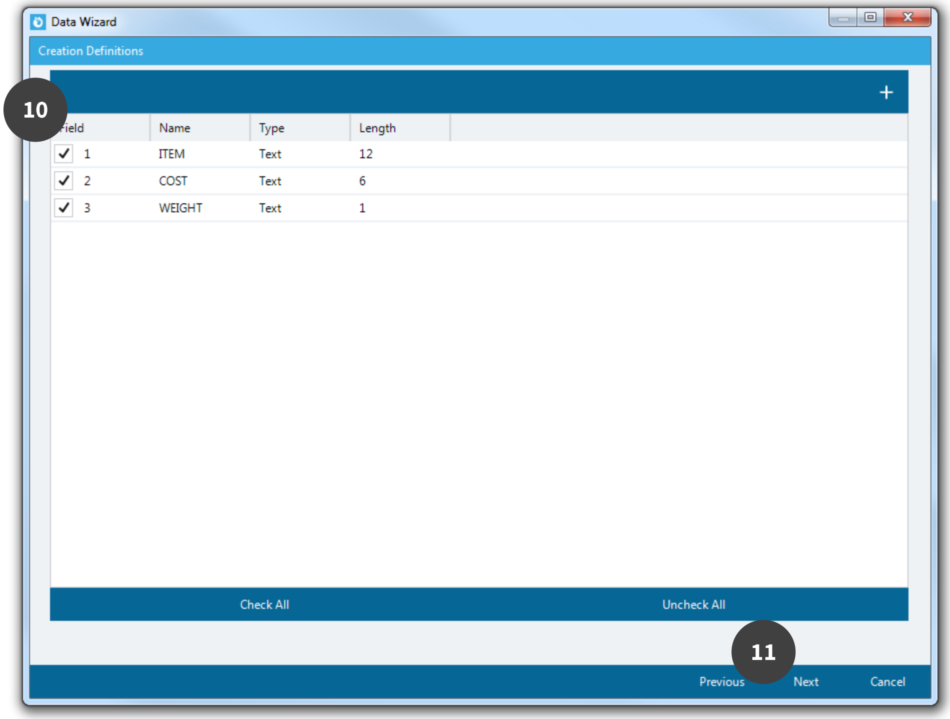
☑ Use included field names

7) Confirm where you want your data import to Start and End.

8) Check the box name *Use included field name* to use the first row of your data as field headers.

9) Click *Next* to continue to Step 3: Creation Definitions.

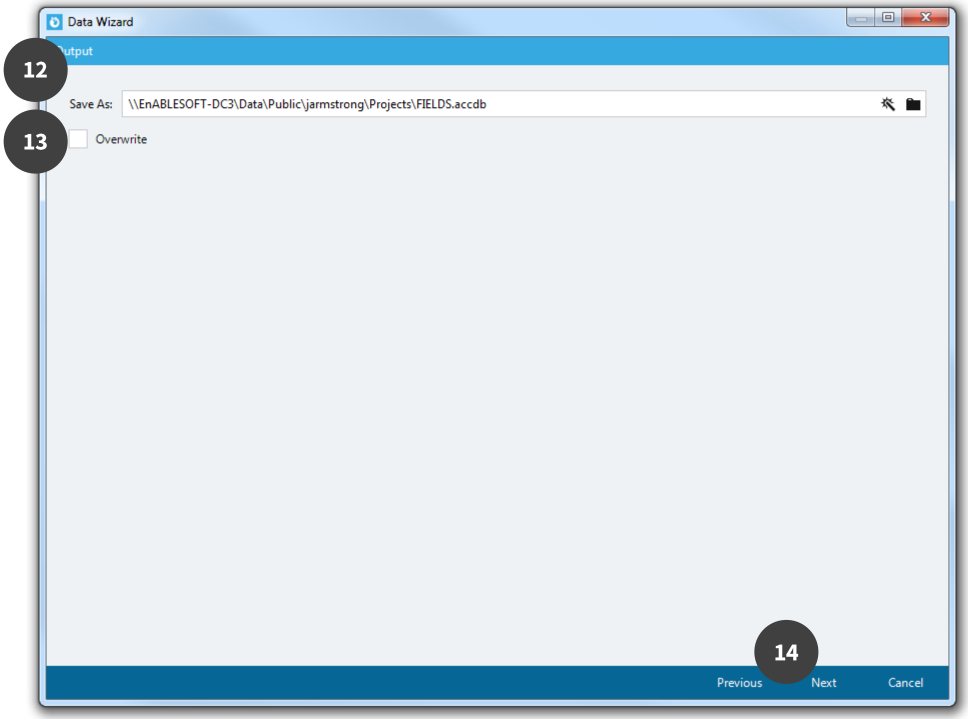
10) Verify the field names, types and sizes. If changes are needed, select the field and click the pencil to *Edit* the field definitions. You may use the checkboxes to the left of each field to exclude fields from importing.



Creation Definitions

	Field	Name	Type	Length
✓	1	ITEM	Text	12
✓	2	COST	Text	6
✓	3	WEIGHT	Text	1

11) Click *Next* to continue to Step 4: Output.



utput

Save As: \\EnABLESOFT-DC3\Data\Public\jarmstrong\Projects\FIELDS.accdb

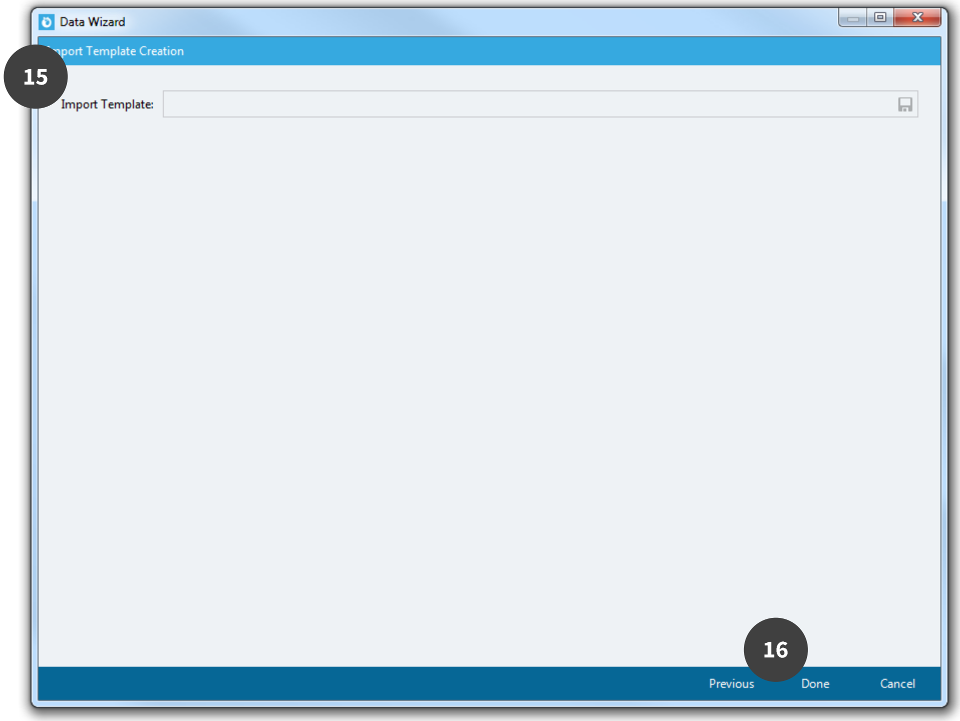
☐ Overwrite

12) Choose the output path where the converted ACCDB file will be saved. This will default to the same location the original file was pulled from.

13) Check the box for *Overwrite* if the output file already exists.

14) Click *Next* to continue to Step 5: Import Template Creation.

15) You can choose to save your import definitions in an Import Template. To do so, click the disk icon to *Save* the template. This will save the definitions that were just configured for use at a later time.



16) Click *Done* to complete the Data Wizard.

FOXTROT'S SOURCE FILE – ACCDB

Upon completion of the Data Wizard, an Access database file is created in which Foxtrot will use as the main data source. This file:

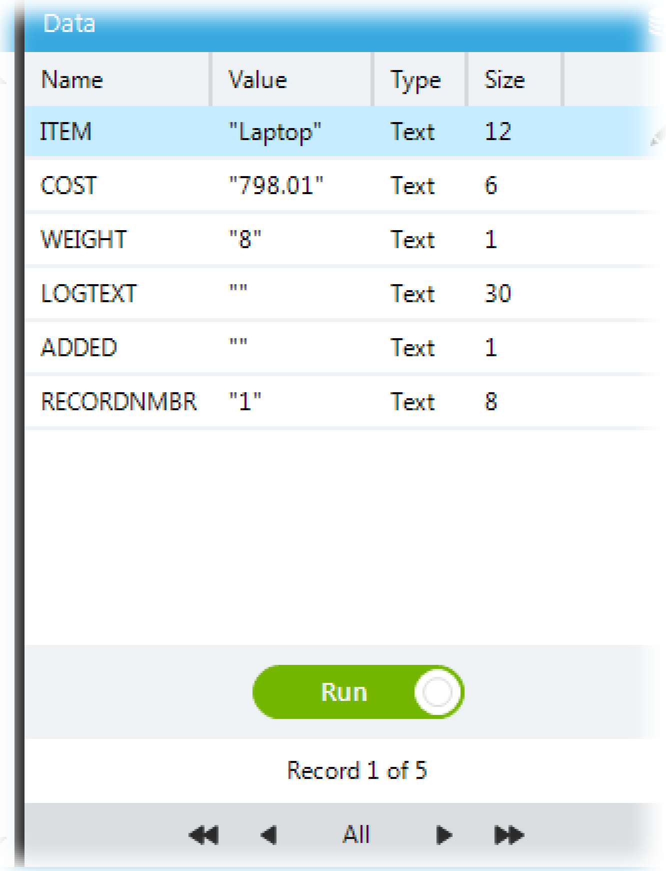
- Is (by default) saved in the same location as the original data source
- Is (by default) named using the first eight characters of the original file name
- Will add three additional fields
- Will remember which records have been processed, and set as *Don't Run*. This allows the Access database file (.accdb) to be reloaded at another time, rather than going through the entire import process again.

Any data that has been imported can be viewed by selecting the Data item in the Project Pane. In addition to the original dataset, Foxtrot will add three additional fields:

- LOGTEXT is a free field that may store any information (up to 30 characters). This field is typically used to log errors with a particular record. Records with any information in this field will be considered *Tagged*. If the field remains blank, it is *Untagged*.

- ADDED tracks which records have been processed, and which are still pending. After a record is processed Foxtrot will denote the field with an *A*, classifying it as *Don't Run*. Records that have not been processed are denoted as *Run*, and the field will remain empty. This field cannot be changed by a user.

- RECORDNMBR tracks the current record number in the database. This field cannot be changed by a user.

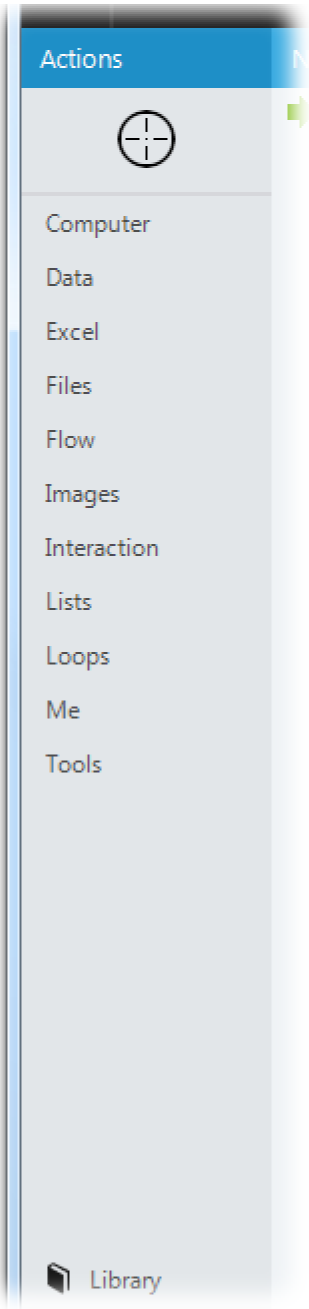


Name	Value	Type	Size
ITEM	"Laptop"	Text	12
COST	"798.01"	Text	6
WEIGHT	"8"	Text	1
LOGTEXT	""	Text	30
ADDED	""	Text	1
RECORDNMBR	"1"	Text	8

Introduction to Actions

ACTIONS IN FOXTROT

Projects in Foxtrot use *Actions* to drive the process. Some Actions require Targets (i.e. anything outside of Foxtrot, like a browser or Windows application) that a user can interact with. Clicking a button, or copying and sending data to a field are examples of this. Simply drag-and-drop the Selector, or Bullseye, to the desired target to access these actions. Foxtrot will only present actions that can be performed with the targeted object.



For example, actions such as *Click*, *Send Value*, and *Get Value* will always be presented. However, when targeting something like a dropdown menu for instance, Foxtrot will provide actions such as *Search* or *Select Item*. Targeting the title bar of an application will provide actions such as *Open App*, *Close Window* and more. Because it would not make much sense to select a specific row in an Excel Workbook when targeting a hyperlink in Internet Explorer, Foxtrot's smart technology shows actions that are relevant to your target.

And some actions do not require targets. Instead, they are found in the Actions Panel directly below the selector. The most widely-used from this list is the *If* action, found under the *Flow* category. This action is used to make logical decisions based on criteria specified by the user. It can also control the flow and speed of projects. *Formula*, *Loop*, and *If* are all examples of these types of actions.

ACTION LIBRARY AND SEARCH

These features are found at the bottom of the Action Panel. The Action Library shows every possible action that Foxtrot has. When an action is selected from the Library, the action's Help menu article tells you everything you need to know about it. The Action Search lets you search for an exact action or describe what you want to do. It will show any actions that are related to your search terms.

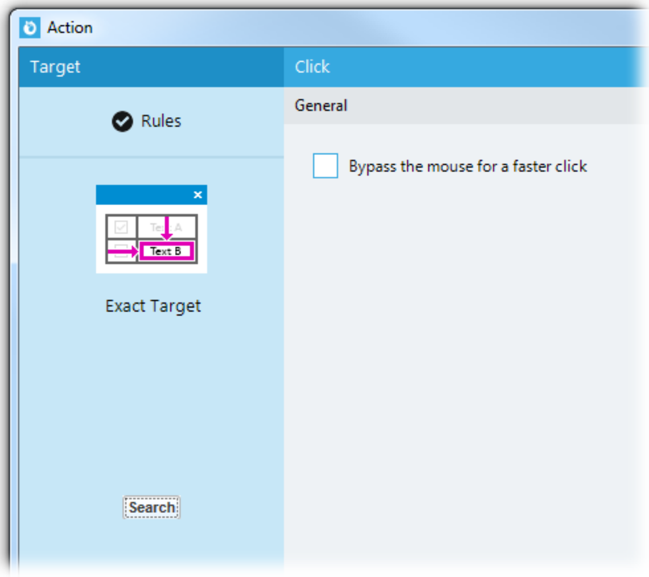
Fundamental Actions

CLICK

Click

A *Click* is used to click an item on the screen. Use it to click a button, a link, a checkbox; virtually anything that can be clicked, Foxtrot can click.

When targeting a web browser, Foxtrot may send an invisible click called a *Bypass Mouse Click*. If there is any chance that, as a result of the click, a message box or any other popup window may appear, this option should not be used.



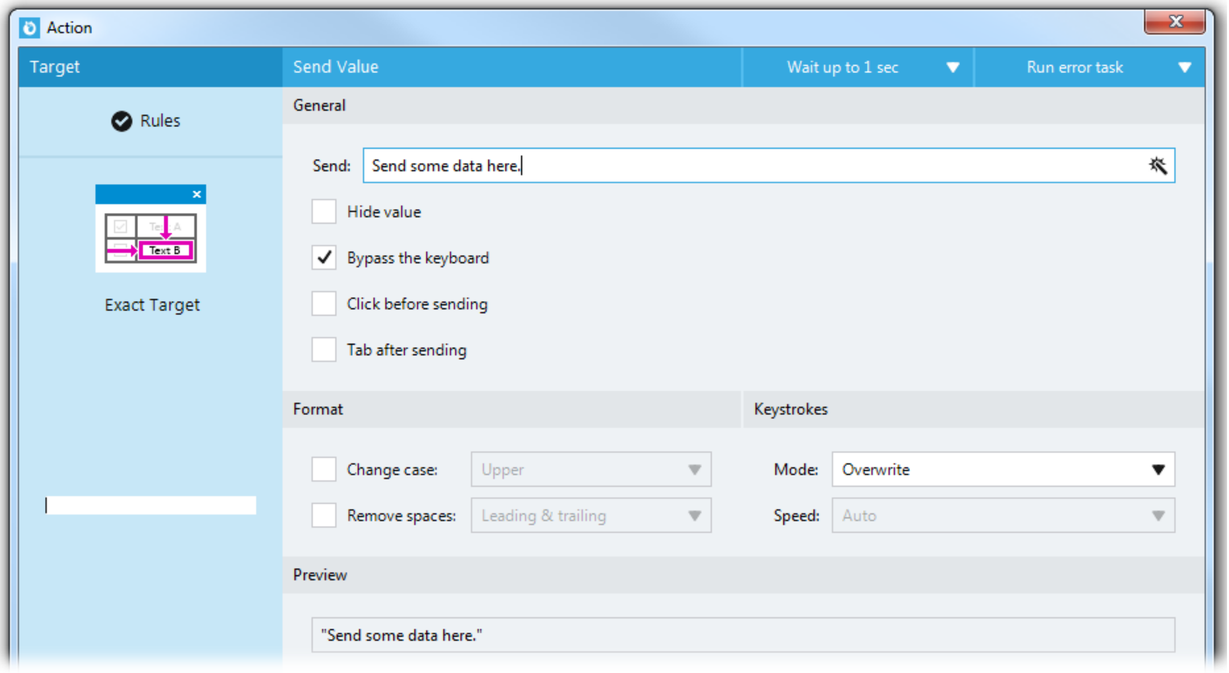
General

☐ Bypass the mouse for a faster click

SEND VALUE

Send Value

The *Send Value* action is used to send information to the screen or application.



Wait up to 1 sec ▼ | Run error task ▼

General

Send: Send some data here.

☐ Hide value

☑ Bypass the keyboard

☐ Click before sending

☐ Tab after sending

Format

☐ Change case: Upper ▼

☐ Remove spaces: Leading & trailing ▼

Keystrokes

Mode: Overwrite ▼

Speed: Auto ▼

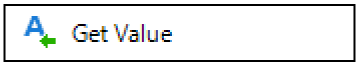
Preview

"Send some data here."

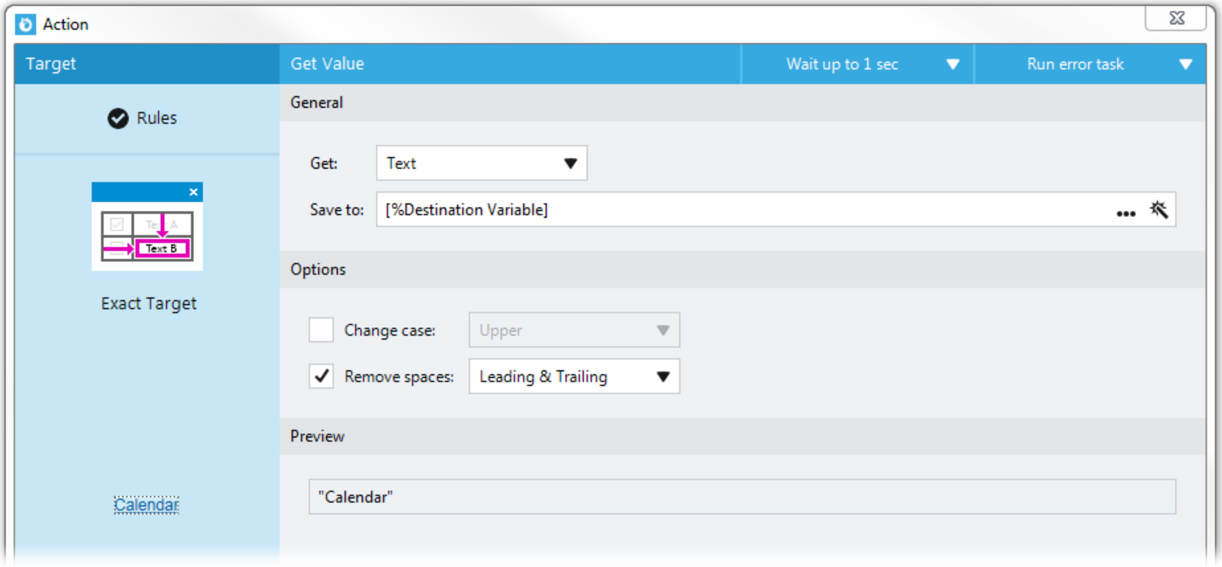
Any special formatting or keystroke options may be configured.

- General
  - Hide Value
  - Bypass the keyboard
  - Click before sending
  - Tab after sending
- Format
  - Change case
  - Remove spaces
- Keystrokes
  - Mode
  - Speed

GET VALUE



The *Get Value* action is used to copy information from the screen, to either a field in a preloaded dataset or to a temporary storage area within Foxtrot called a Variable. Some users may wish to copy information to verify its value, make comparisons, or store for a later use. The **Format** options described for *Send Value* also apply to this action, and will be applied prior to retrieving the value.



General

Get: Text

Save to: [%Destination Variable]

Options

□ Change case: Upper

☑ Remove spaces: Leading & Trailing

Preview

"Calendar"

Foxtrot requires a location for the copied value to be store, typically a Variable. The next section will explain how to create a variable, as well as an introduction to other Tokens. Before a value can be copied into a variable, an understanding of Tokens is necessary. This guide will continue the *Get Value* process in a later section.

Expression Builder

TOKENS

The Expression Builder may be used to access data resources, referred to as Tokens. Tokens may be combined in any arrangement, and even embedded within each other. The types of tokens available will depend on the action to be performed, and what has been targeted. Each time a token is selected, it is added into the Expression box at the bottom of the window. There are fourteen different Token categories:

- Application: information about the Foxtrot application.
- Computer: information regarding the workstation
- Data: values of the current record's imported data
- Date & Time: current, future and past dates and times.
- Folders: common folder directories
- Formula: a list of all formulas
- Image: information about the image currently open in the Image Editor
- Keyboard: a full keyboard
- Emulator Keys: keys that may be sent to an emulator. This replaces the Keyboard category when targeting an emulator
- Lists: access to any lists that have been created
- Loop: information regarding the current Loop. This will only be displayed when the Execution Arrow is on an action within a Loop.
- Project: information regarding the current project
- Random: random numbers, letters and names
- Variables: access to any variables that have been created in this project.

VARIABLES

Variables are most widely used to save or manipulate information from the screen, like an on-the-fly data source. Unlike data that has been imported however, variables have no length limitations. Further, a project may contain an infinite number of variables. Some typical uses for variables may include, but are not limited to:

- Internal data storage
- Calculations
- Comparisons
- List referencing

Before creating a variable, it is important to understand the different types:

- Text can store anything – a word, number, paragraph, date, etc. When in doubt, choose this.
- Number will store any whole number. Any decimal values will be rounded to the nearest integer. For example, 3.14 will be saved as 3.
- Decimal will store any number, including decimal places.

- Date can only contain a value date. The format for this variable will be a four-digit year. The month and day will not be zero-filled. January 1st, 2014 will be saved as 1/1/2014.
- True/False holds a true or false value only. This is most commonly used when storing results of a Search action.

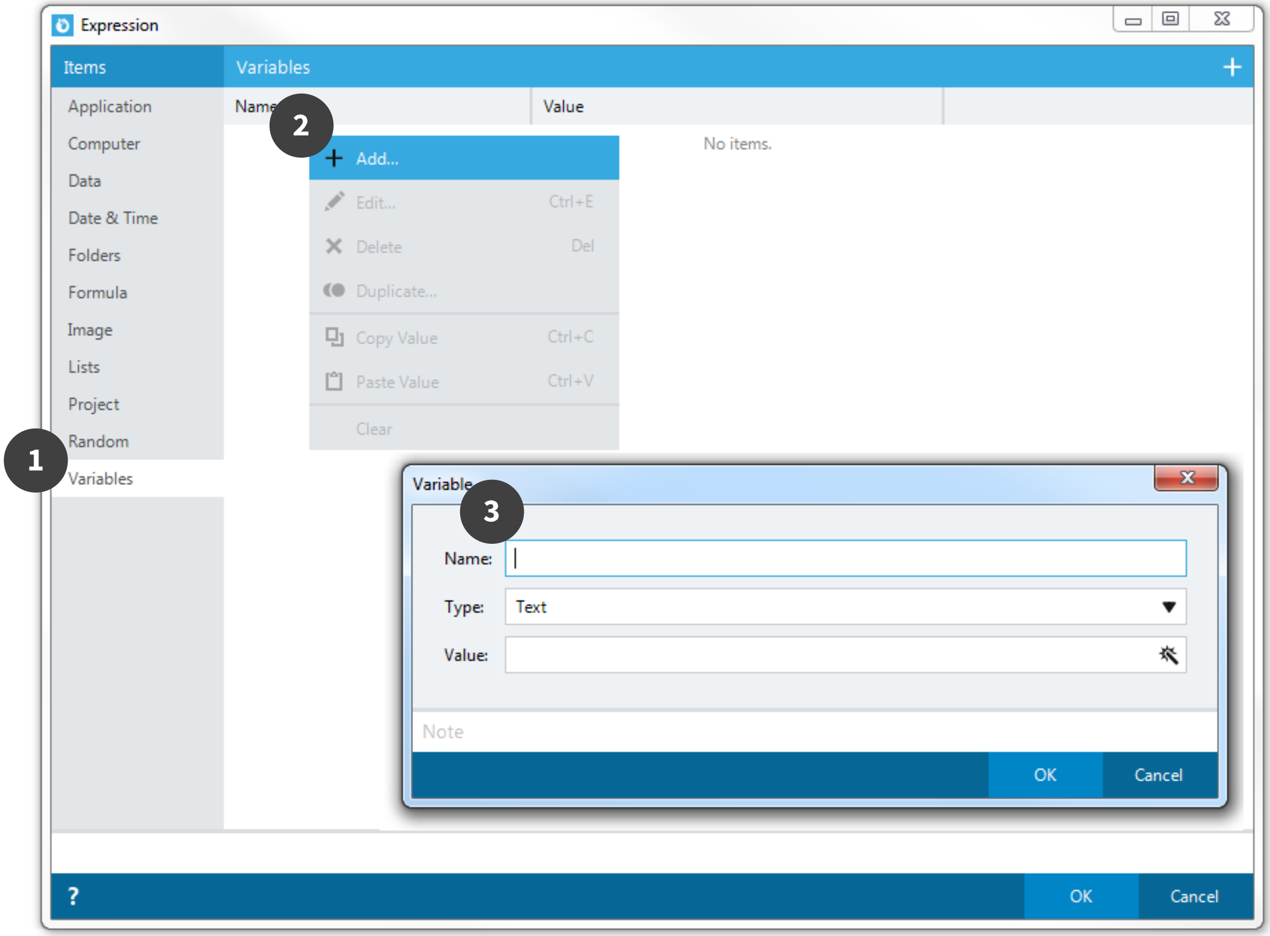
CREATING A VARIABLE

There are two techniques used to create a new variable.

On the Fly

Creating a variable on the fly is probably the quickest and easiest way to do so. It prevents creating too many variables, and enables the user to create and save information in one fluid motion. The following required the expression builder to be opened. This can be accomplished by clicking either the magic wand icon or the ellipses button.

1) From the expression builder, select the Variables category
2) Right click anywhere in the white space and select Add. You may also click the + in the upper right corner.
3) Specify a Name and a Type for your new variable, and click OK.

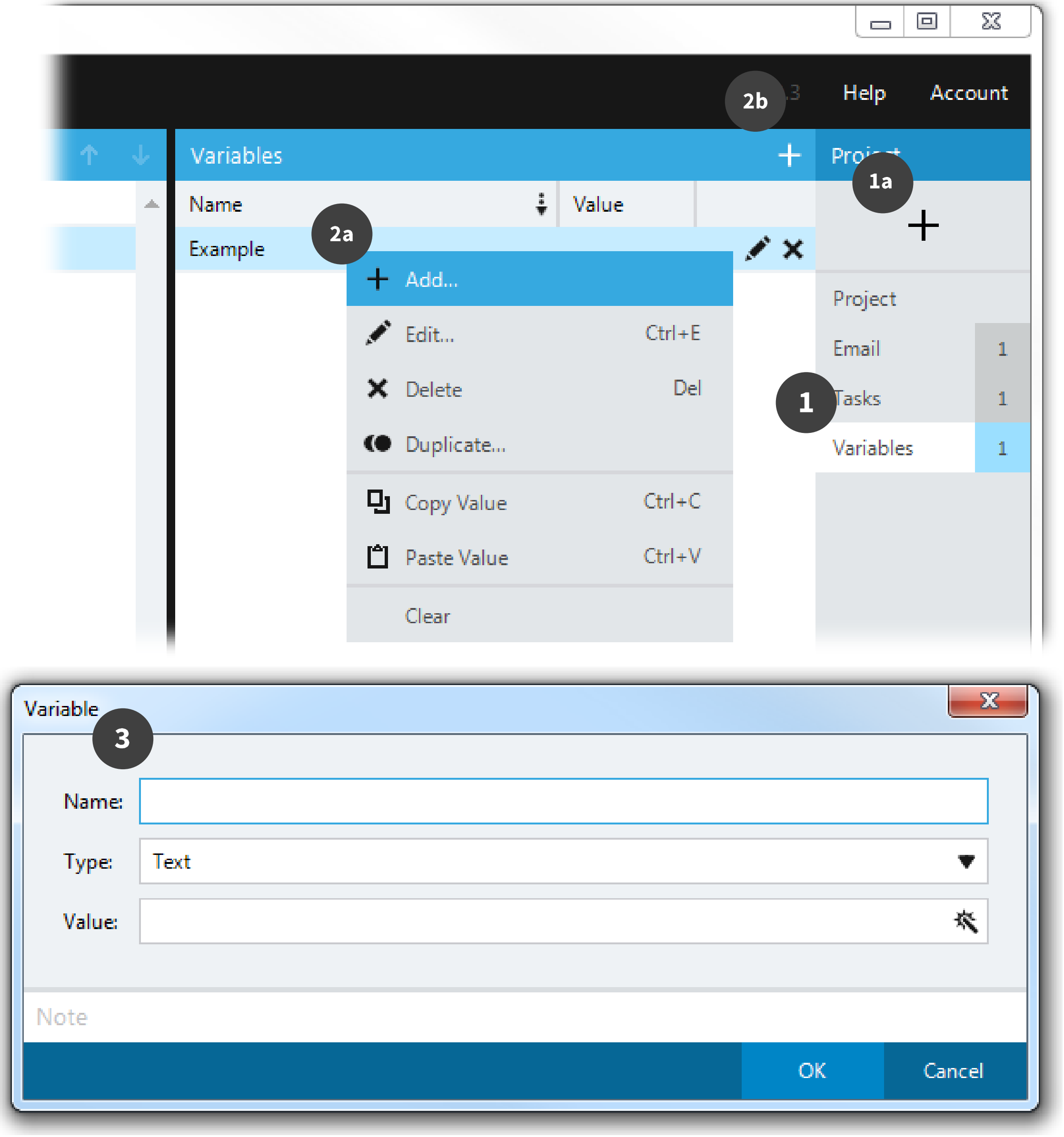


Type: Text

From the Project Pane

Creating a variable from the Project Pane is probably the more common approach, but lacks the ease and convenience of creating one on the fly. There are two ways to create a variable from within the project pane:

1) Start by switching to the Variables list in the project pane.
    a. If you don't already have a variable within your project, you will need to click on Add (the big +), on top of the items list, and select Variable. Skip to step 3.
2) From here, a Variable may be created by either:
    a. Right clicking anywhere in the white space and selecting Add
    b. Click the small +, in the blue space on the right side of the project pane.
3) Specify a name and a type for the variable, and click OK.



Name:

Type: Text

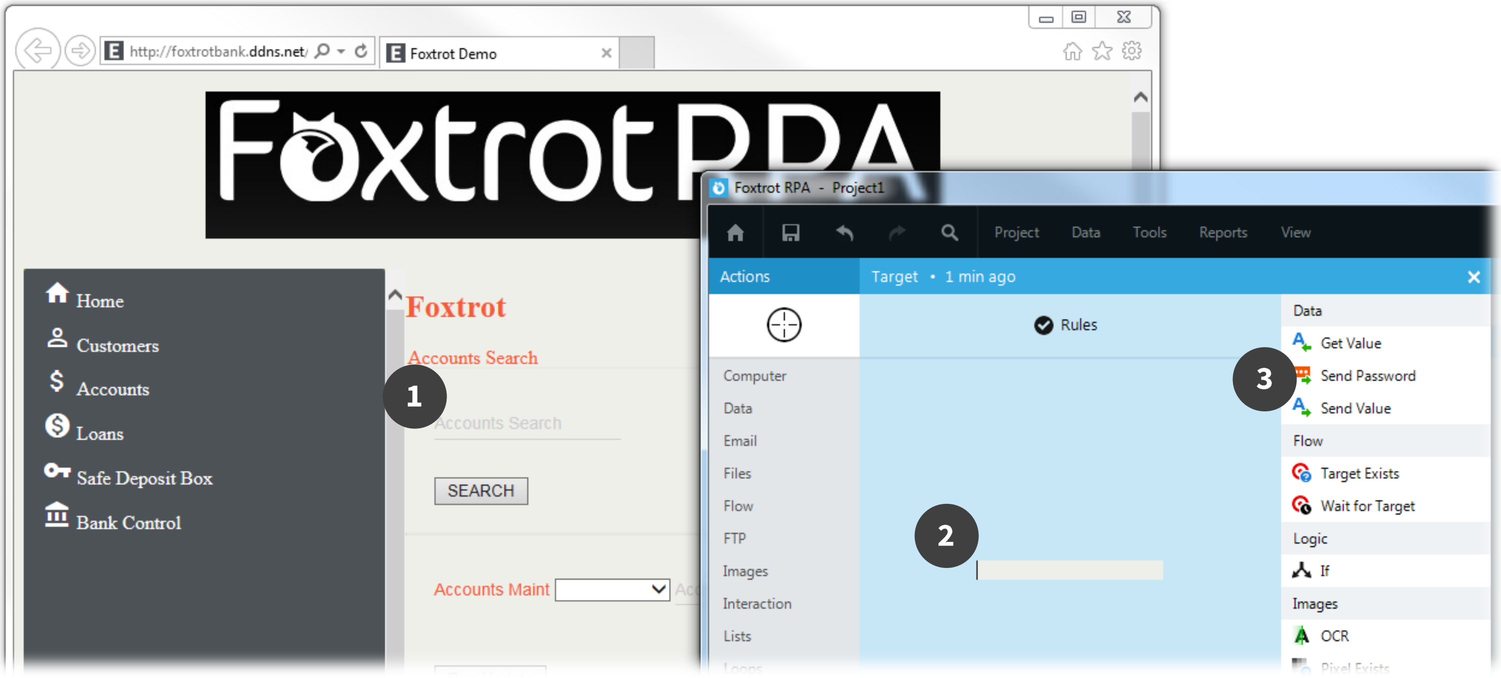
Value:

Using Tokens and Actions

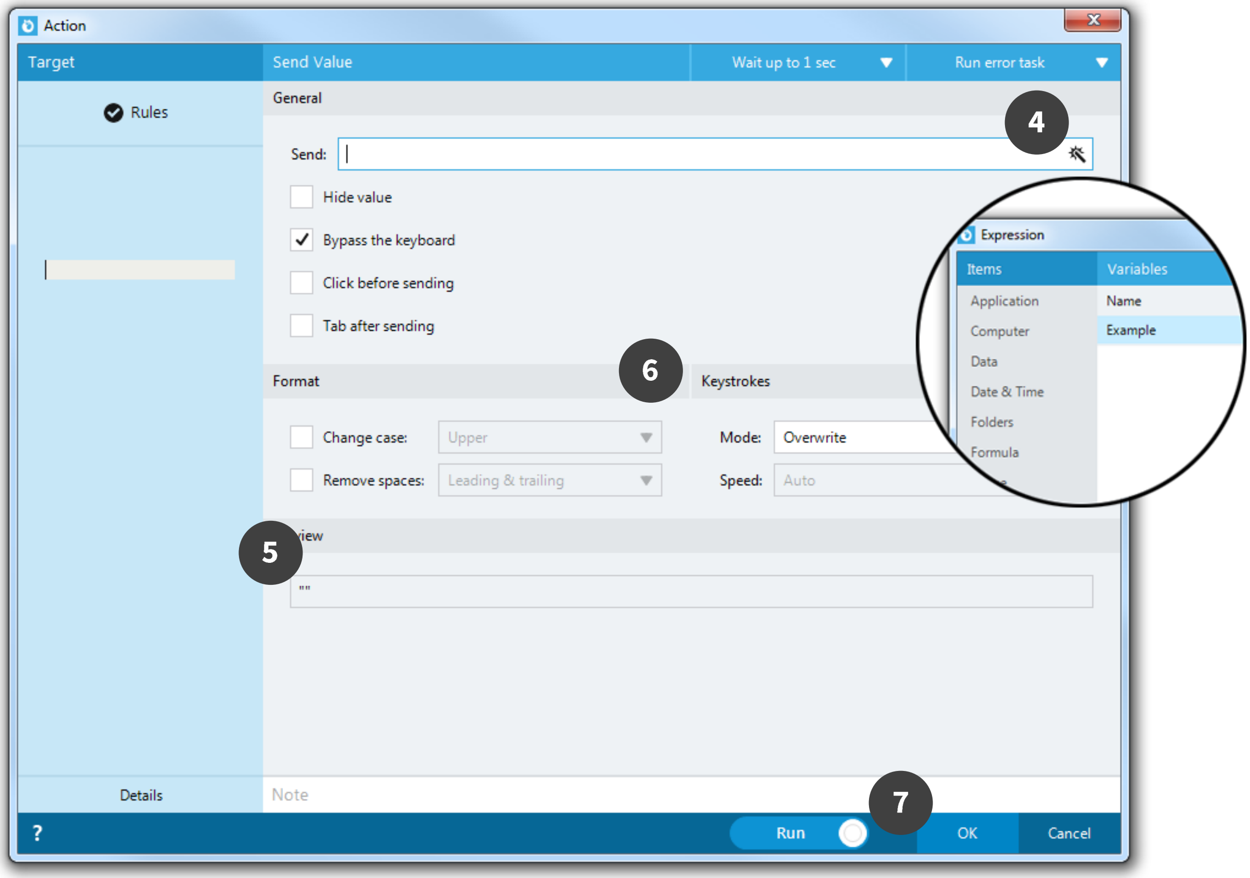
After becoming familiar with accessing Tokens and creating variables, it is time to apply them to a functional script. Below are two examples, one in which Foxtrot will send data to a webpage, the other in which Foxtrot will copy information from the screen into a variable.

SENDING DATA TO A TEXT FIELD

1) Target the desired text field.
2) Verify the preview matches the intended target.
3) Choose the *Send Value* action.



4) Open the Expression Builder and select the token to send.
5) Verify the preview of the token in the Action Builder
6) Provide any format or keystroke options as desired
7) Click OK to create the action.



Send:

□ Hide value

☑ Bypass the keyboard

□ Click before sending

□ Tab after sending

□ Change case: Upper

□ Remove spaces: Leading & trailing

Mode: Overwrite

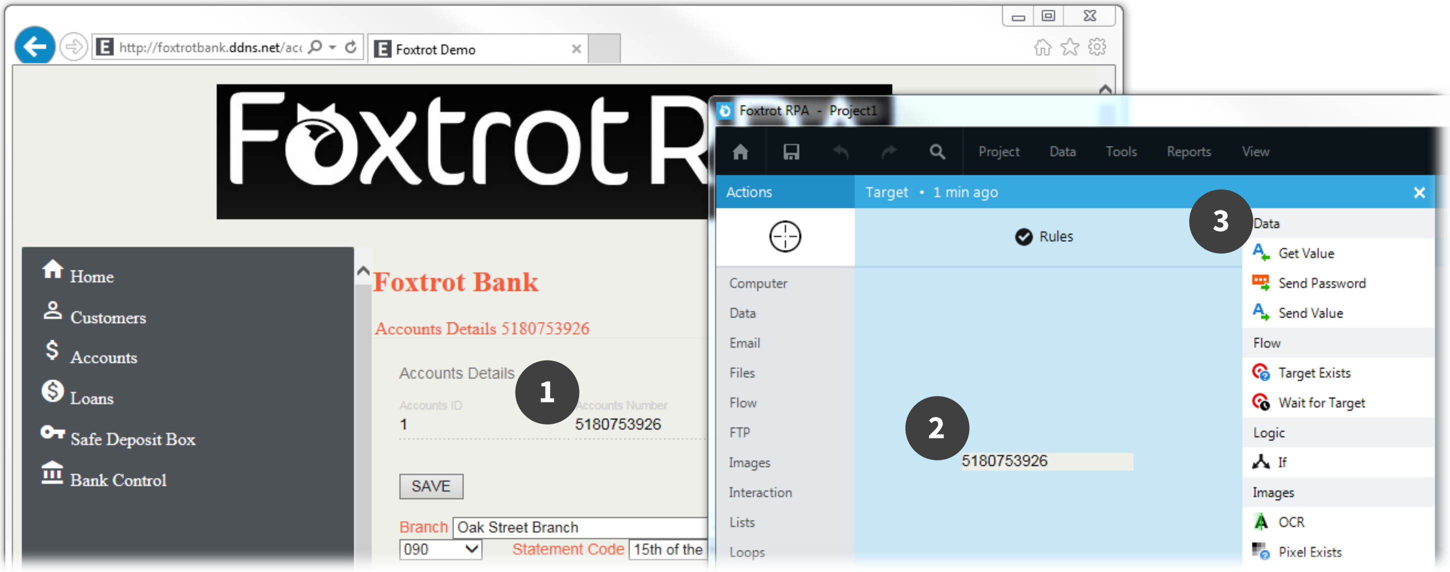
Speed: Auto

Preview

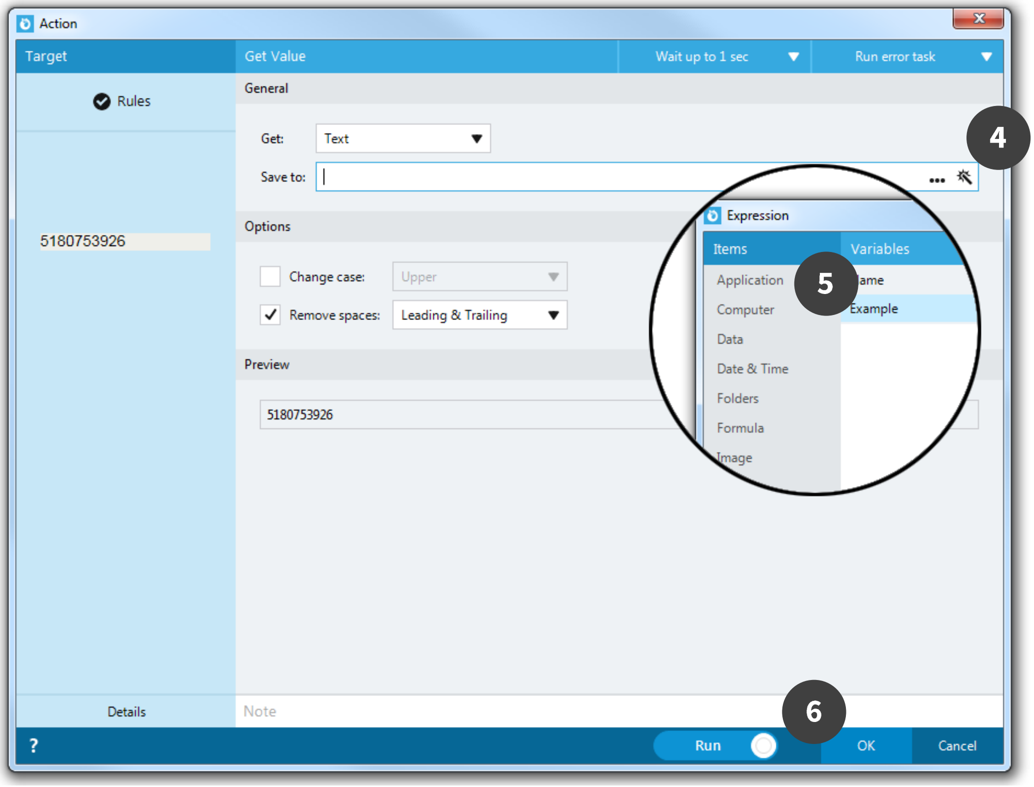
""

COPYING A VALUE INTO A VARIABLE

1) Target the desired item to be copied.
2) Verify that the target preview matches what is expected.
3) Choose the *Get Value* action.



4) Click on the expression builder icon
5) Choose either a Data field or Variable token to save the value to
6) Click OK on the Action Builder to create the action.



Get Value

General

Get: Text

Save to:

Options

☐ Change case: Upper

☑ Remove spaces: Leading & Trailing

Preview

5180753926

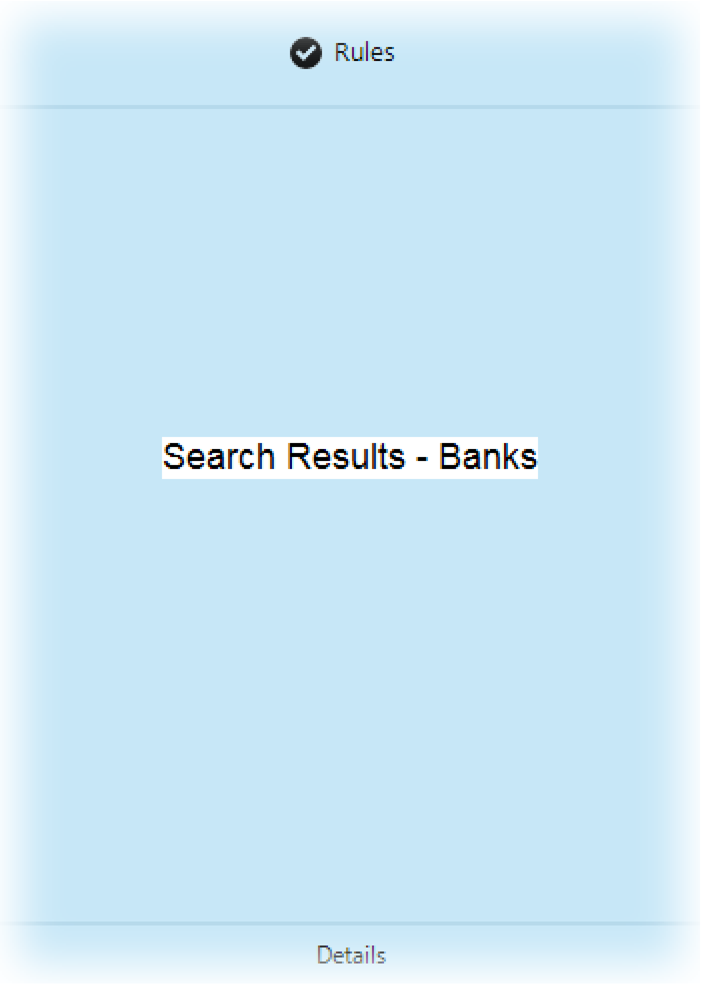
Using Highlights

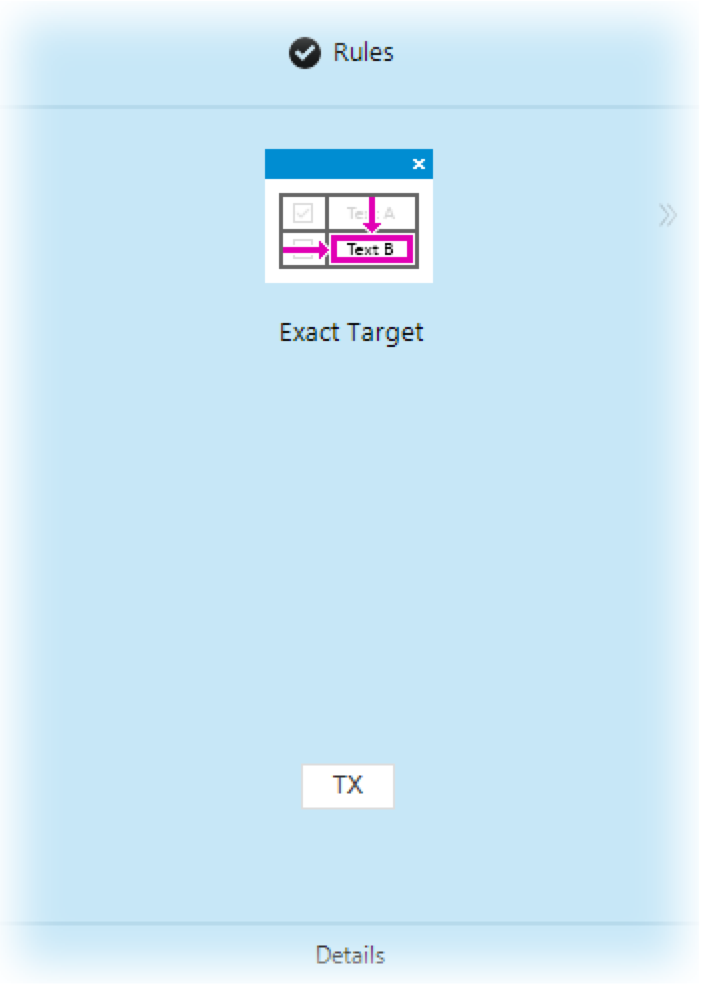
When working in a web environment a process may require Foxtrot to interact with a single target that could change its absolute positioning, or multiple targets in the same table. Foxtrot's highlight technology is a branch of Smart Actions that are only available when targeting a web table. The purpose of the Highlight is to dynamically interact with targets in various locations within a table. Highlights can be used to:

- Search for a static or dynamic value in a table
- Interact with a target relative to a particular location
- Perform identical maintenance to multiple record in the search table

IDENTIFYING A WEB TABLE

Notice the different between the two Target Previews pictured. When targeting a web table, in addition to a preview of the target, Foxtrot will display a generic table indicating that highlighting technology may be used. Pictured on the right is an example of what it would look like when targeting a web table, while on the left there is no web table.





HIGHLIGHTING ACTIONS AVAILABLE

Start Highlight will highlight a specific cell in a web table. After execution of this action, the targeted cell will appear yellow, as if Foxtrot has taken a highlighter to that cell's text. After creating a highlight, it may be moved or cleared. The Start Highlight action is available when the *Exact Target* option is selected, and only visible when targeting a web table.

Move Highlight will move the current highlight to a new location, relative to the current highlighted cell. This action will only appear if a highlight has already been created from a Start Highlight or Search action, and is presented only when targeting a *Highlighted Target*. The highlight may be moved up, down, left or right.

Clear Highlight removes the yellow color from the highlighted cell. This is purely for cosmetic purposes. Foxtrot will no longer be able to interact with a *Highlighted Target* until a new highlight is created. This action is also only available when the *Highlighted Target* option is selected.

Search allows Foxtrot to search a web table for a value, regardless of the location. This action is available when *Table* targeting option is selected, and only visible when targeting a web table. Foxtrot will search for an exact match, unless otherwise specified in the action builder. Match criteria can be tailored by selecting any one, or combination, of the following options:

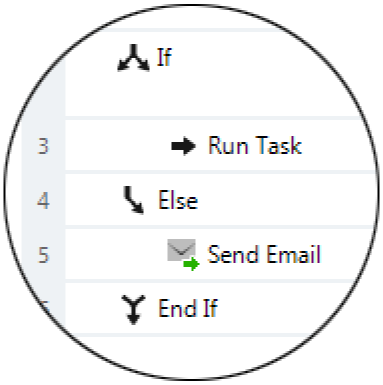
- Ignore case
- Ignore surrounding spaces
- Partial match
- Use wildcards

The search results can also be stored in a variable by using the Results field at the bottom of the action builder. If enabled, Foxtrot will set a variable to True if the value to be searched is found in the table, and False otherwise.

INTERACTING WITH THE HIGHLIGHTED TARGET

Once the desired target is highlighted, Foxtrot's smart targeting can be used to interact directly with the highlighted target. When targeting the highlight again, Foxtrot will automatically select the *Highlighted Target* option. Performing a *Send Value* action, for instance, sends data to the *Highlighted Target*. For example, if performing a *Get Value* action to the *Highlighted Target*, the value of the highlight will be copied, regardless of the cell location from record to record. All actions that are available to an *Exact Target* will also be available when working with highlights.

If Actions



Quite often, a process may require more logic than simple data entry; a process may require a set of actions *if* some condition is met, or *if* a particular target is found. If actions are one of the most widely used actions, as they allow for customized logical commands to drive the direction of a script. Every *If* action will conclude with an *End If*. Any action(s) to be performed if the condition is "true" should be included between the *If* and the *End If* end-caps.

An *Else* clause may also be included, to take action if the original condition does not hold. In the example, a task will be run if some condition is met, otherwise (else), Foxtrot will send an email.

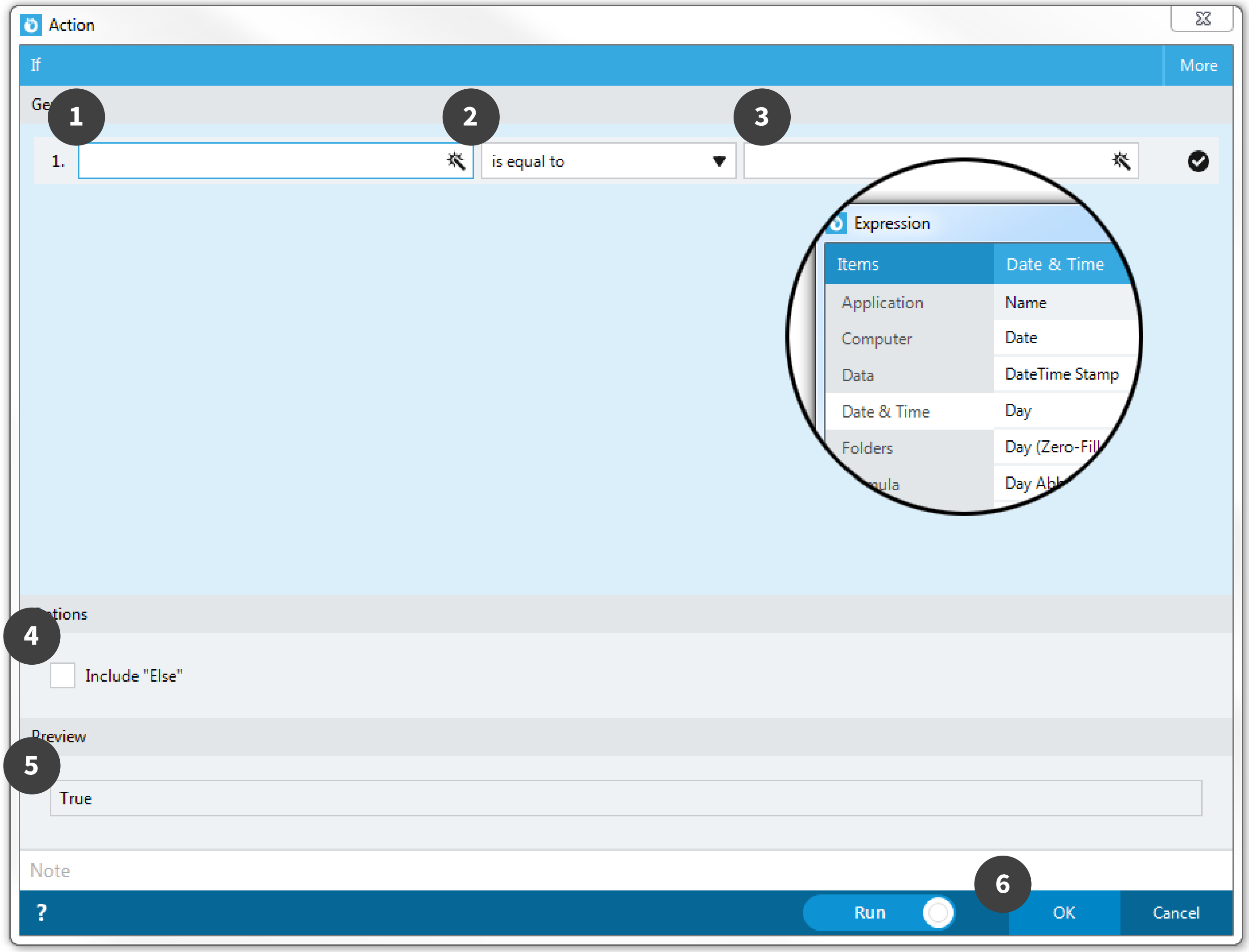
Foxtrot has five different types of If actions, below are examples of the four most commonly used.

IF STATEMENTS

If Statements can be found in the *Flow* section of the Actions List. They allow simple or complex decisions to be made. Foxtrot can compare hard coded values, or Tokens within a project. Comparison types include:

- Is blank
- Is not blank
- Is equal to
- Is greater than
- Is less than
- Contains
- Begins with
- Ends with
- Is numeric
- Is not numeric
- Is not equal to
- Is greater than or equal to
- Is less than or equal to
- Does not contain
- Does not begin with
- Does not end with

The values to be compared are to be specified by the user. To create an *If* Statement:

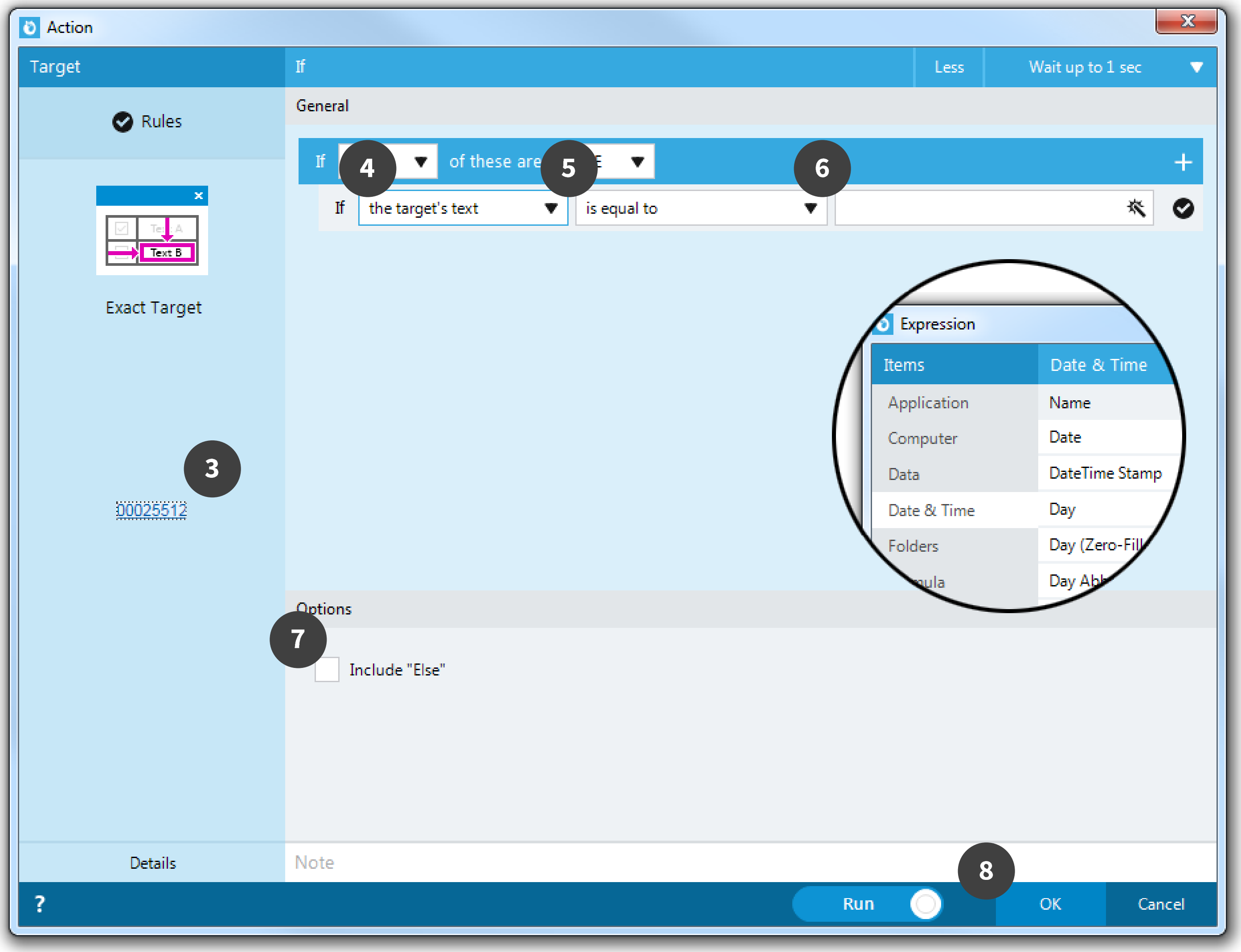


1) Make a selection for the value to be compared.
2) Choose a comparison type.
3) Make a selection for the value to be compared against.
4) Optional: select the Else clause to specify a series of steps to be completed if the condition does not hold.
5) Verify the preview result.
6) Click OK to create the *If* statement.

IF THE TARGET'S TEXT

A similar comparison with a value on the screen may be made. Using *If the target's* text will do just that, regardless of an *Exact Target* or a *Highlighted Target*. The comparison options are the same as our standard *If* statements.

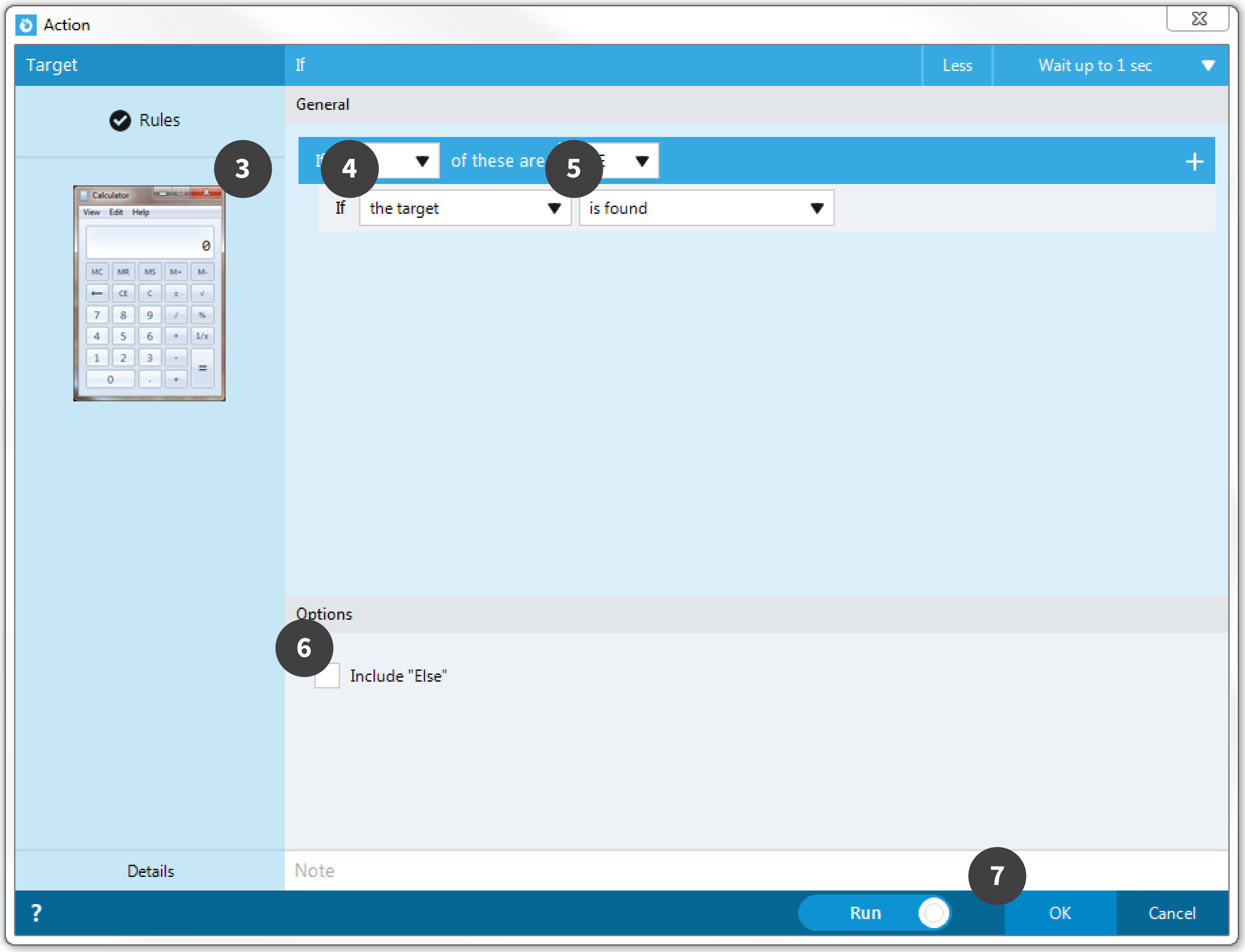
To create this type of *If* action:



1) Use the bullseye to target some value on the screen.
2) Select the *If* action from the action list.
3) Verify the target's preview.
4) Change the first condition to "The target's text".
5) Choose a comparison type.
6) Make a selection for the value to be compared against.
7) Optional: select the Else clause to specify a series of steps to be completed if the condition does not hold.
8) Click OK to create the If action.

IF THE TARGET IS FOUND OR NOT FOUND

The existence (or not) of a particular target on the screen may be important. For example, a simple click to save changes may have worked previously, but for certain records might result in an error message from the application. Foxtrot can determine if a target exists. By targeting the element in question, and selecting the If action from the action builder, Foxtrot can determine if the target is either found or not found. To do so:



If the target is found

□ Include "Else"

1) Use the bullseye to target the desired element on the screen.
2) Select the *If* action from the action list.
3) Verify the target's preview.
4) Change the value of the first condition to "the target".
5) Choose a comparison type, either "is found" or "is not found".
6) Optional: select the Else clause to specify a series of steps to be completed if the condition does not hold.
7) Click OK to create the If action.

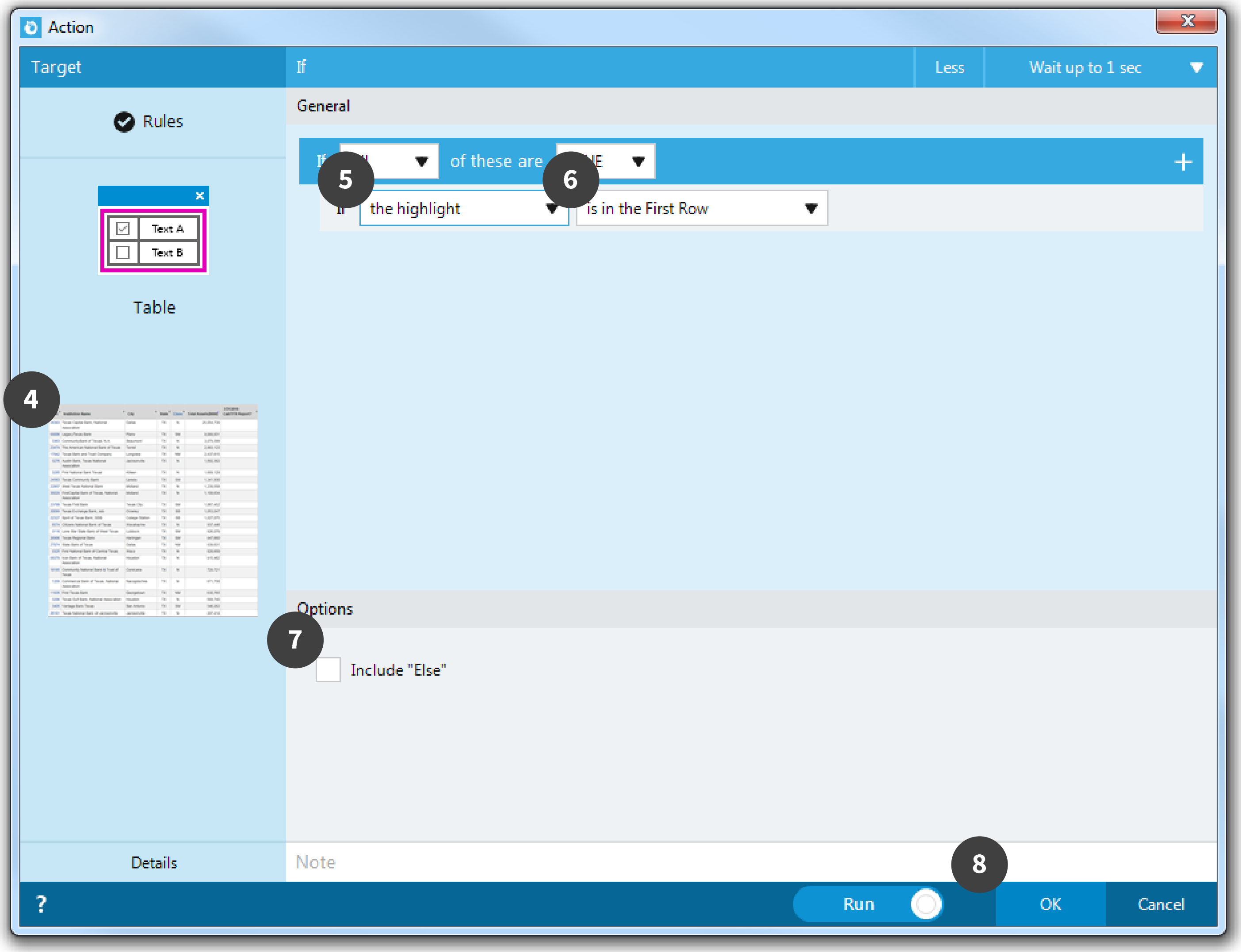
IF THE HIGHLIGHT

Foxtrot can make a decision based on the location of a highlight. Note that this particular type of *If* action is only available when targeting a web table, and is made visible by selecting the *Table* target option.

Foxtrot can determine if the highlight:

- Is in the First Row
- Is in the First Column
- Is in the Last Row
- Is in the Last Column
- Is not in the First Row
- Is not in the First Column
- Is not in the Last Row
- Is not in the Last Column

To determine the location of the highlight:



If ll of these are UE

If the highlight is in the First Row

□ Include "Else"

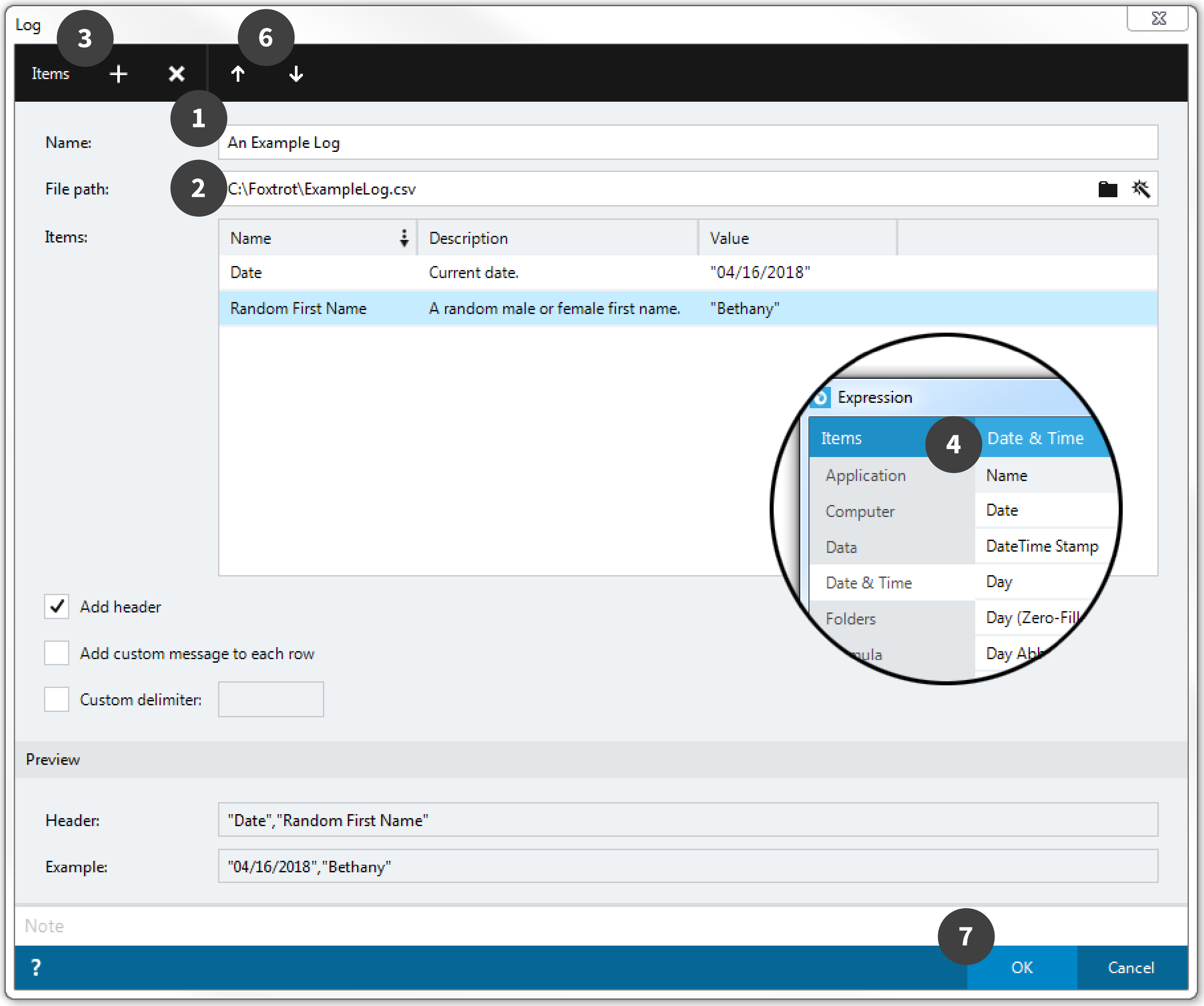
1) Use the bullseye to target a web table
2) Select the *Table* target option
3) Select the *If* action from the action list.
4) Verify the target's preview.
5) Change the value of the first condition to "the highlight".
6) Choose a comparison type
7) Optional: select the Else clause to specify a series of steps to be completed if the condition does not hold.
8) Click OK to create the If action.

Logs

Logs allow for the recording of certain events or information from a project in real time. Foxtrot can record information gathered from the screen to create a success, error or exception report. A project can even attach this file to an email, or open it internally as a new data source. Note that the log must be in a .csv file format and it is recommended that the file is saved on a local drive. All Foxtrot needs to know is what information to store in the file, and when to record that information.

CREATING A LOG

To create a Log, start by navigating to the Item List. Next, click Add (the large + at the top), and then select Log. This will display the Log Wizard.



Name: An Example Log

File path: C:\Foxtrot\ExampleLog.csv

Items:

Name	Description	Value
Date	Current date.	"04/16/2018"
Random First Name	A random male or female first name.	"Bethany"

☑ Add header

□ Add custom message to each row

□ Custom delimiter:

Preview

Header: "Date","Random First Name"

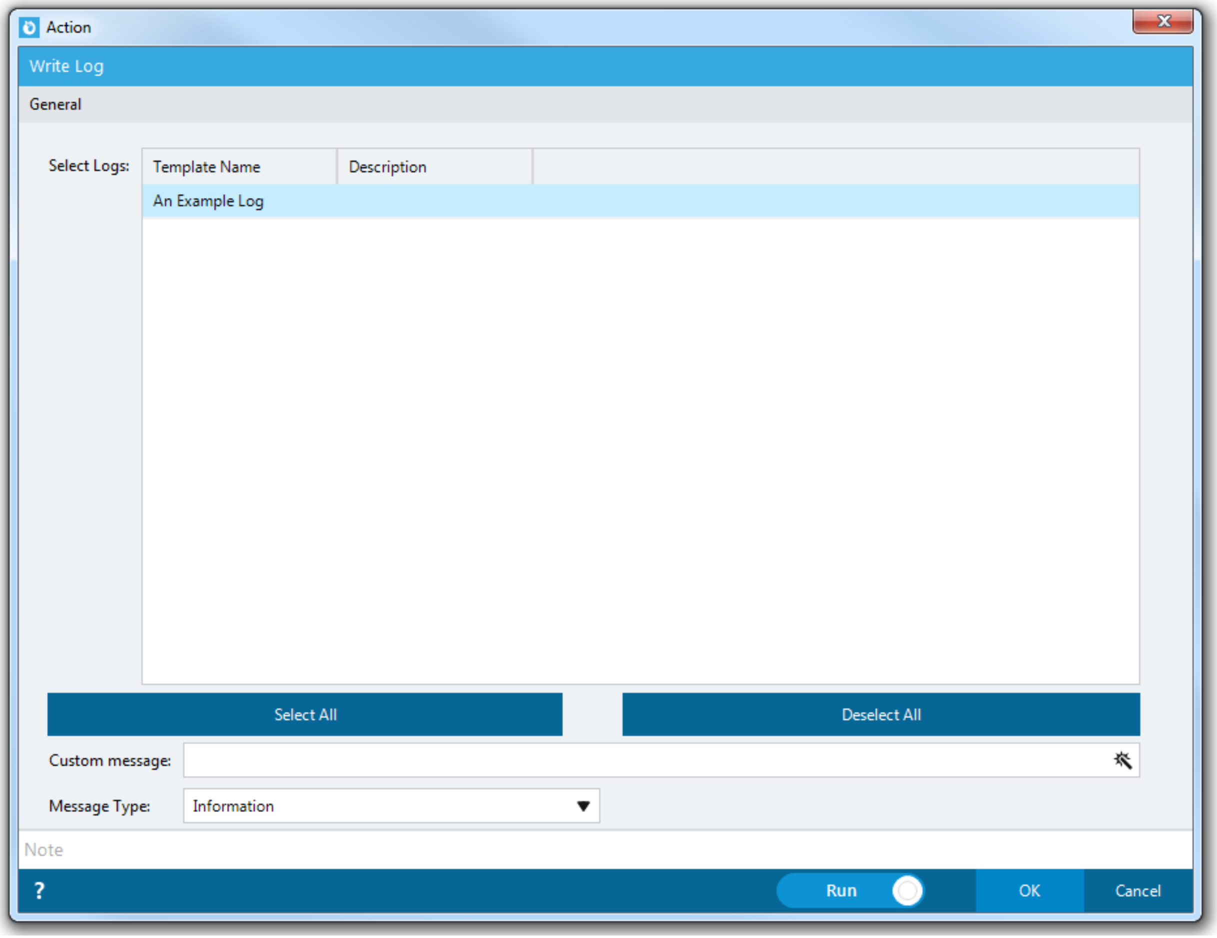
Example: "04/16/2018","Bethany"

1) Define a Name unique to the project
2) Provide a file path to save the log to
3) Click the + to add tokens to your Log.
4) One at a time, select each desired Token the file will include. Click OK after each selection.
5) Repeat steps 3 & 4 as needed.
6) After adding all desired tokens, items may be rearranged to how they should be displayed in the file. Highlight the item you wish to move, then use the Up & Down arrows to change its position.
7) Click OK to create the Log.

Foxtrot will only write to this file when a *Write Log* action is executed.

WRITING TO A LOG

Once a Log is created, the project may write to it at any time. For example, within an *If* action, or when logging an error within an error task (discussed later in this guide). In any case, the *Write Log* action will be used to log the desired data to the file. This action is found in the *Data* section of the action list. Once selected, specify the Log to write to. If the file does not exist, Foxtrot will create it. If the file does exist, Foxtrot will append to the current file. Note that Foxtrot cannot write to a file if it is open, make sure the file is closed while running the associated project.



Select Logs:

Template Name	Description
An Example Log	

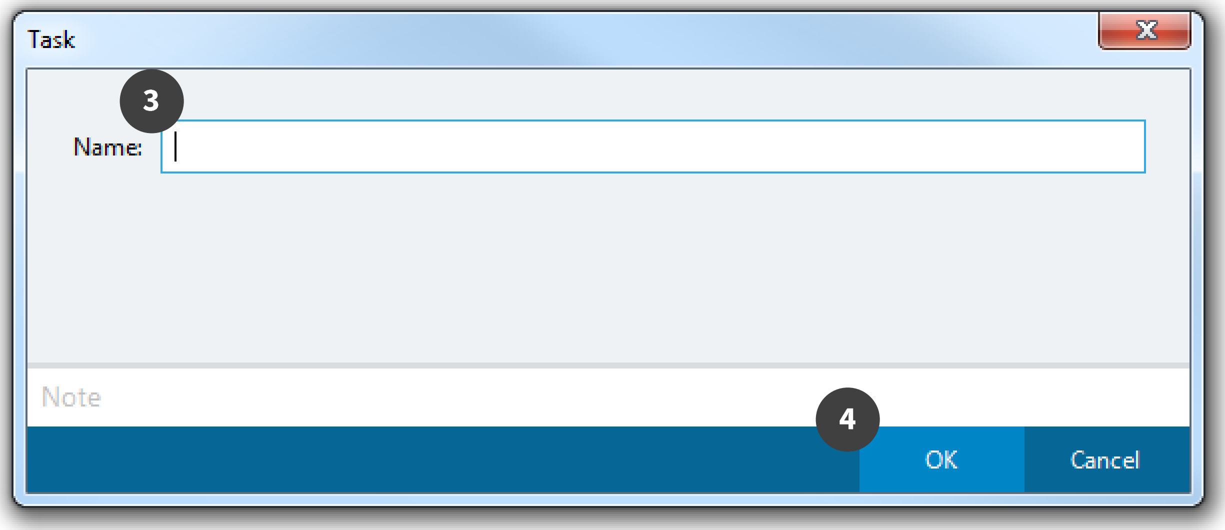
Custom message:

Message Type: Information

Tasks

Tasks are used to incorporate an organizational flow within a project. Tasks can be used to store a sub routine of actions that may be called upon at a later time. The project pane will allow a user to view, edit, add, delete or duplicate any Task. There are three types of Tasks, all of which have their own unique use.

CREATING A TASK



1) Click Add (the large +) at the top of the item list
2) Select Task from the list
3) Give the Task a unique name
4) Click OK to create the Task.

START HERE TASK

A project must contain one and only one Start Here (sometimes called a Start Up) Task. In fact, when creating a brand new project, Foxtrot will provide the project with a Start Here Task, named “New Task”. The Start Up Task serves as the beginning and ending of every project; the *Start Task* action in a project will be the very first action in the Start Here Task. The very last action will be the *End Task* action in the Start Here task. A project’s Start Up Task may be changed at any time by right-clicking and editing the task.

TASK

A Task is a subroutine of commands that can be called upon at any time, using the *Run Task* action. A project can include an infinite number of tasks that may be called upon as needed. Think of a task as a set of actions that are packaged into one slick *Run Task* action. A Task may be called upon from a Start Here Task or another Task.

Once a task has reached the *End Task* action, Foxtrot will return to the task in which the *Run Task* action was executed. To end a task prior to reaching the end, an *Exit Task* action may be used. From there, Foxtrot will know to proceed with the action following the *Run Task* action.

RUN ON ERROR

By default, in the event of an error Foxtrot will automatically stop the project and display an error message. However, if a project contains an Error Task, Foxtrot will instead call this Task automatically and perform the action(s) specified by the script to handle the error. A Task set to Run on Error is a special type of Task, in which certain rules and restrictions apply:

- Only one Error Task may be used at a time
- Certain *Flow* actions such as *Run Task* and *Skip to End* cannot be used
- A *Mouse Sequence* action cannot be used.

Once Foxtrot has handled the error as specified within the Task, a project may be resumed to complete the process, or continue updating records that do not exhibit an error. Foxtrot may continue running the script following an error with a *Resume* action. A project may be resumed nine different ways:

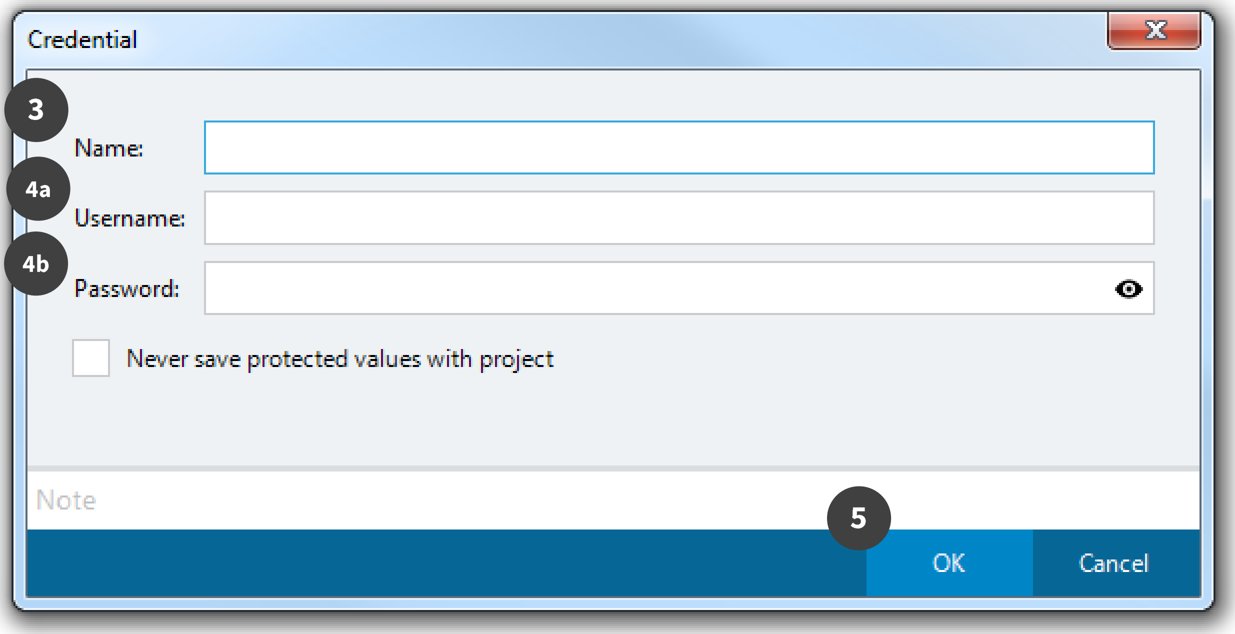
- Skip the Action
- Retry the Action
- Go to Start
- Go to Top
- Go to Bottom
- Go to End
- Go to Label
- Go Forward
- Go Backward

By resuming the script, Foxtrot will automatically continue running the process, regardless of an error message. It is recommended to log these errors into a log file.

Credentials

A Credential can contain a username and an encrypted password that Foxtrot can use throughout your project. You can use these Credentials to interact with your core applications, to add FTP connections, open secure Excel Workbooks, and much more. Any action that requires a password will require you to create a Credential beforehand.

CREATING A CREDENTIAL



1) Click Add (the large +) at the top of the item list
2) Select Credential from the list
3) Give the Credential a unique name.
4) At least one of the following is required:
    a. Enter the Username you want to be associated with this Credential.
    b. Enter the Password you want to be associated with this Credential
5) Click OK to create this Credential.

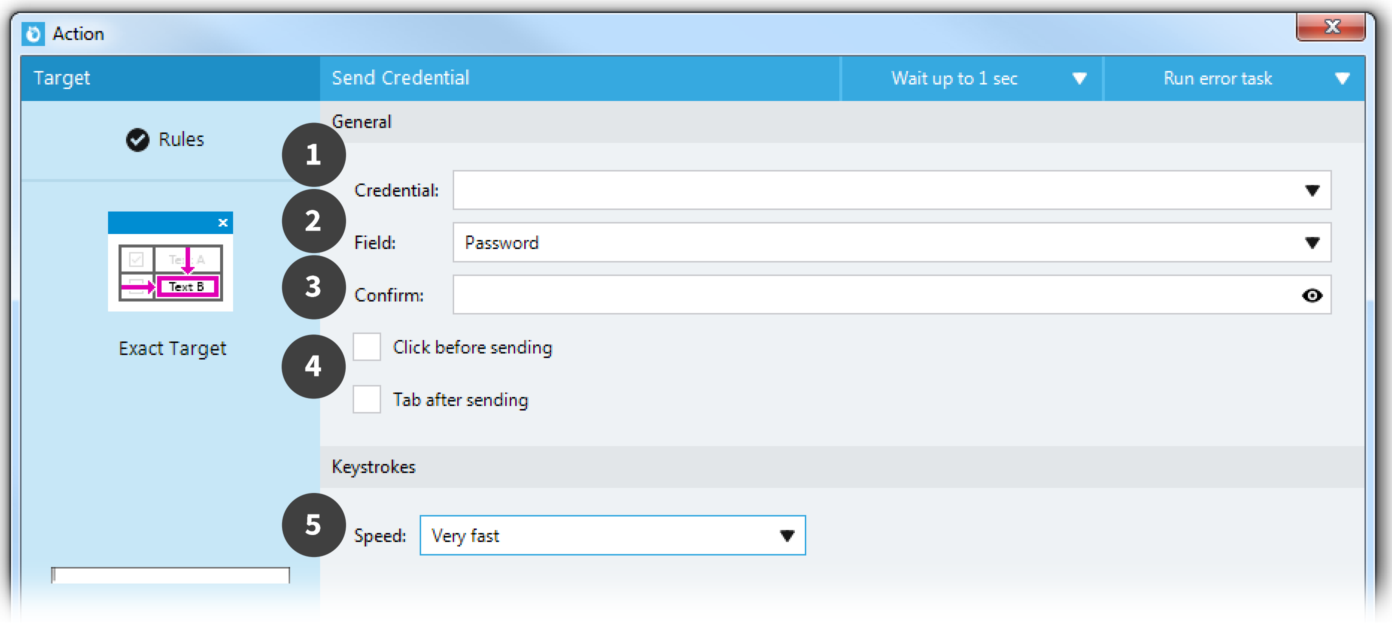
After this Credential has been created, you will be able to utilize the *Send Credential* action as well as using the Credential elsewhere in your project.

SEND CREDENTIAL ACTION

Send Credential

The *Send Credential* action functions the same as a *Send Value* action, but with a few additional steps to set it up.

1) Select a Credential to use.
2) Choose which field from that credential should be sent to the target
3) If it is a password, verify the password.
4) Optional: enable 'Click before sending' and 'Tab after sending'
5) Optional: adjust the 'Speed' to something other than the default value.



1 Credential:

2 Field: Password

3 Confirm:

4 □ Click before sending

□ Tab after sending

5 Speed: Very fast

Connections

Foxtrot is smart enough to interact with applications that are used on a daily basis in many institutions. Thus far, the features and methodology taught in this guide have been directed towards Internet Explorer. Aside from highlights (which are unique to a browser environment), everything covered in this guide up to this point can also be applied not only to Windows applications, but also to Excel workbooks, PDF forms and certain supported emulators (also called green screens). One key difference between targeting in previous chapters, and an Excel workbook for example, is the inclusion of Connections.

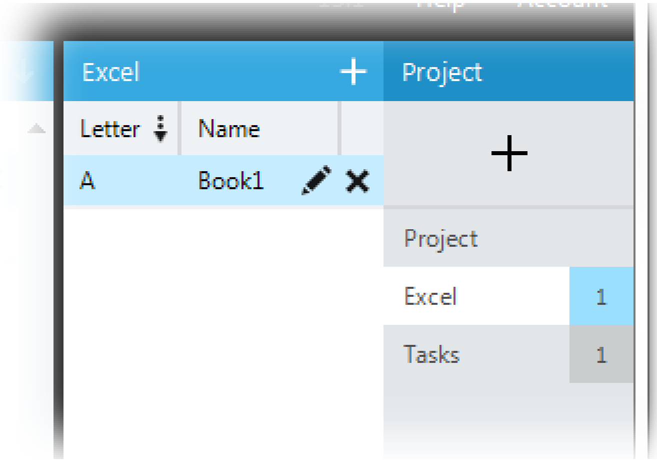
WHAT IS A CONNECTION?

A Connection is a unique letter identified used to establish communication between Foxtrot and the target application. With a Connection, additional actions may be used as they relate to the target application. As this guide will describe in subsequent sections, such technologies have unique actions. The following technologies require a connection:

- Emulator
- Excel
- PDF
- FTP
- SQL
- SMTP & IMAP Email

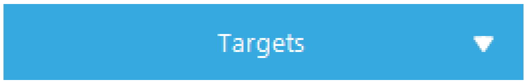
CREATING CONNECTIONS

For the most part, the methods of creating a Connection are universal for emulator, Excel and PDF technologies – simply drag and drop over the emulator screen, Excel workbook or PDF document. While doing so, a blue banner will appear, indicating that a Connection has been made. Foxtrot will automatically assign connection letter with the next letter available, starting with A and ending with Z.



Letter	Name
A	Book1

Project	
Excel	1
Tasks	1



Releasing the mouse anywhere over the document or session will allow you to create an Action. Once the desired Action is created, a unique Connection will appear in the appropriate Project Pane.

In this example, a Connection to an Excel workbook was made. The methods for connecting to an emulator session or PDF form are exactly the same. However, the method for connecting to FTP, SQL and Email are different. These will not be covered in this guide.

Excel Automation

As described above, Foxtrot can interact directly with Microsoft Excel workbook to automate any job. Rather than importing a dataset, Foxtrot can work directly with your spreadsheet. Before Foxtrot can communicate with the workbook, a connection must be established. The previous chapter provides more detail on creating connection. This chapter is dedicated to outlining the features and functionality of Foxtrot and Excel.

TARGETING OPTIONS

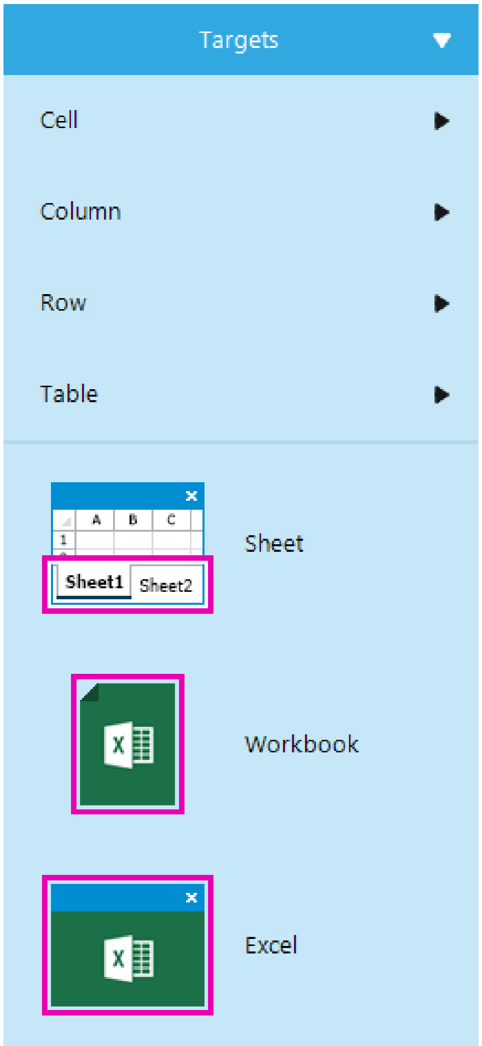
Targeting an Excel workbook will allow Foxtrot to lock in on each cell, row, column or sheet targeted. Hover over the blue *Targets* bar will display all available targeting options.

- Cell provide actions to interact with specific cells within the workbook. Most Excel automation processes will involve some degree of cell interaction. Common cell actions include *Send Value*, *Copy Value*, *Move Selection*, and *Search Cells*. There are four different targeting options when working with a cell.

- Column will display actions unique for interacting with a particular column in the workbook. Popular column actions include *Format Column*, *Select Column*, and *Insert Column*. Columns have three additional targeting options.

- Row provides row-specific actions. Similar to column actions, rows have three additional targeting options available.

- Table provides actions specific to working with tables. This has two additional targeting options.



- Sheet allow for the management and selection of a specific worksheet within the workbook. Actions such as *Add Sheet*, *Copy Sheet*, and *Select Sheet* are available with this target option.

- Workbook displays actions in respect to the workbook itself. Some widely used actions for this include *Excel Macro*, *Select All Cells*, and *Save Workbook*.

- Excel will display actions regarding the Excel application. Actions such as *Open App*, *Set Connection*, and *Open Workbook* are found here.

MORE TARGETING OPTIONS

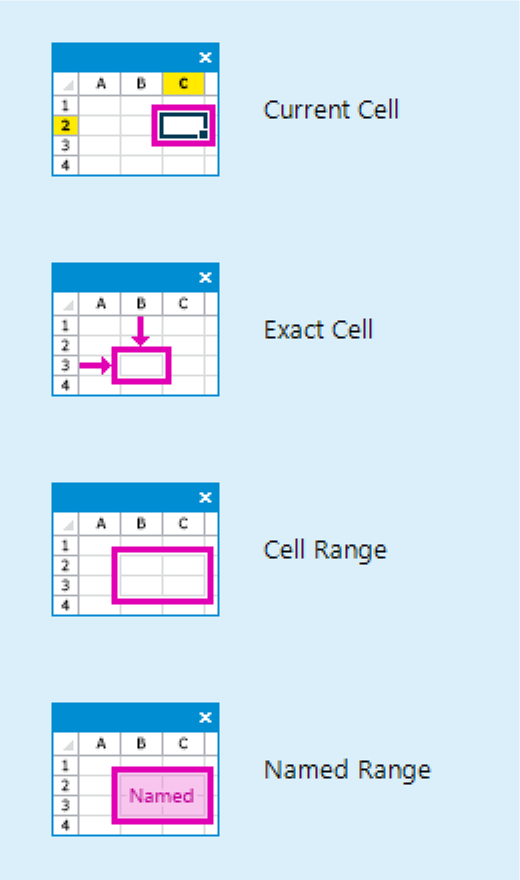
As mentioned previously, cells, columns and rows all have additional targeting options available. Hovering over these options will display the targeting options available. The targeting option can always be changed before you create an action.

- Current dynamically interacts with the active cell, column or row. Foxtrot will not ask for an exact input, because it will always interact with what is currently active.

- Exact interacts with a single static target. After choosing the Exact option, Foxtrot will require the user to input the desired cell, column or row letter or number. Going forward, Foxtrot will always know to interact with that particular choice. You may also use tokens in these fields.

- Range interacts with a range of cells, columns or rows. Choosing the Range option will require the input parameters of the desired range. Notice that tokens may be used to specify some range.

- Named Range is only available when targeting cells. This allows the user to choose a cell range that is specified by some custom name as it would appear in a workbook.

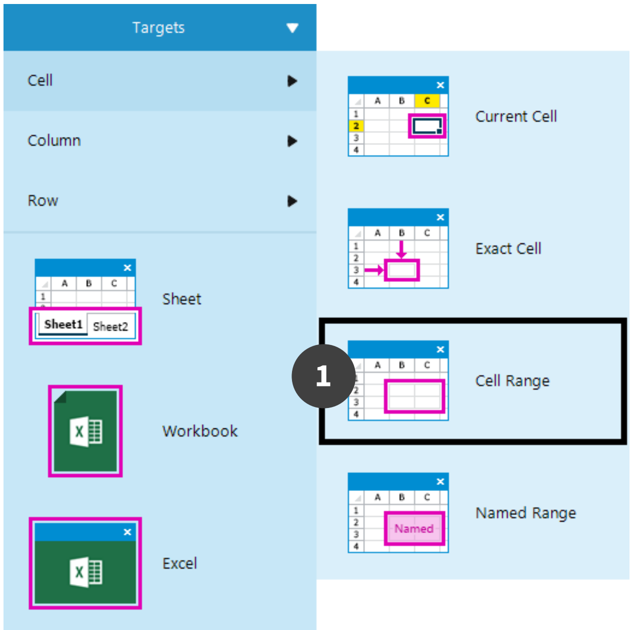


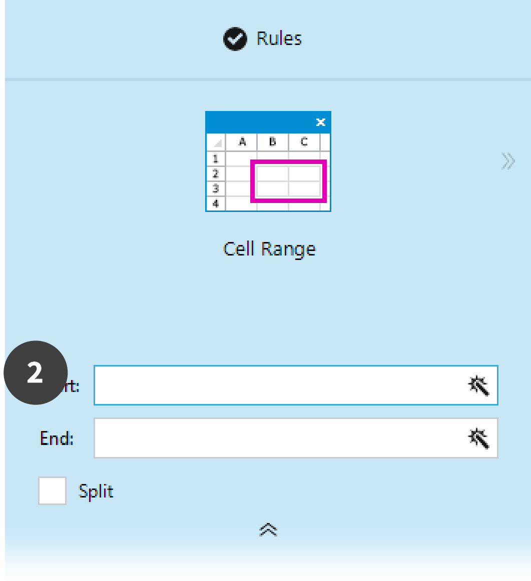
INTERACTING WITH A WORKBOOK

The list of things Foxtrot can do within Excel is virtually endless and usually requires a combination of targeting options. A simple, yet popular example may provide some insight on how to properly use Foxtrot's Excel technology.

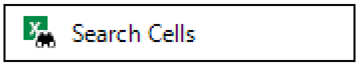
For this example, a user needs to update the ASSETS field in their workbook. It is assumed that a connection exists, and the asset information is already copied from another source and stored into a variable.

1) Hover over the blue targets bar, then Cell, followed by selecting Cell Range.
2) Specify the range in which to search.



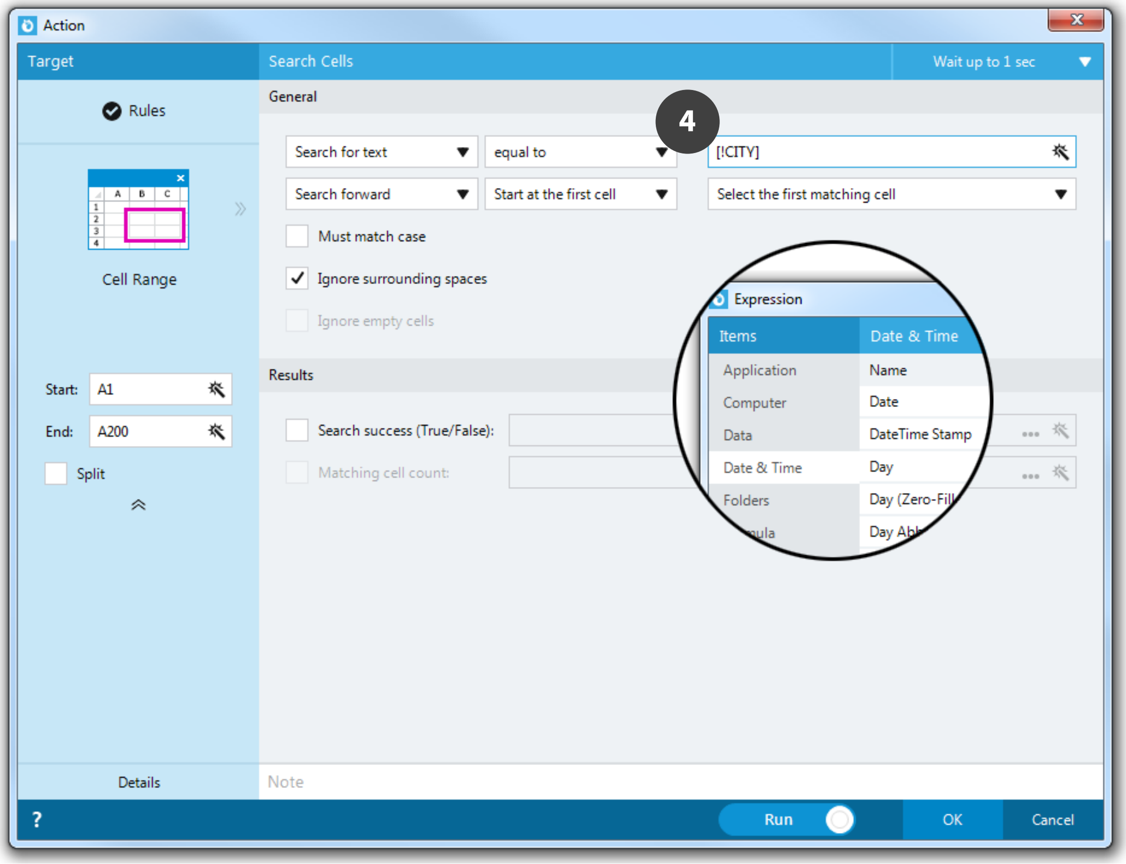


3)



Choose the *Search Cells* action.

4) Input the desired value to search.



Wait up to 1 sec

Start: A1

End: A200

□ Split

Search for text | equal to | [!CITY]

Search forward | Start at the first cell | Select the first matching cell

□ Must match case

☑ Ignore surrounding spaces

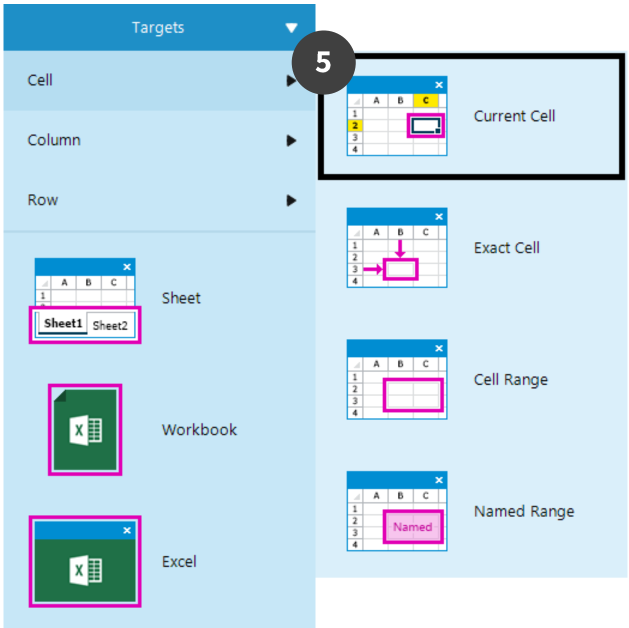
□ Ignore empty cells

□ Search success (True/False):

□ Matching cell count:

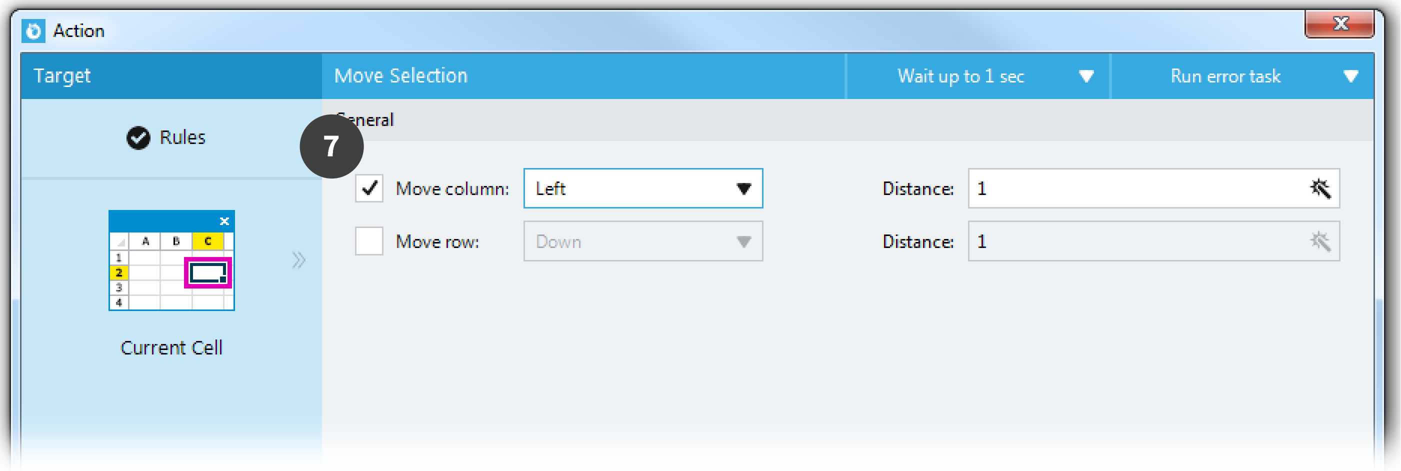
By using a *Search Cells* action, Foxtrot will select the first cell that matches the value of the [!CITY] field token. This cell will then become the Current Cell. The purpose of the search is to efficiently update the proper record according to the city in the dataset. Next, move the current cell to the left 1.

5) Hover over the blue targets bar, then Cell, followed by selecting Current Cell. Foxtrot will now interact with the cell that contains the value that was searched for.



6) Move Selection Choose the *Move Selection* action.

7) Specify Foxtrot to move 1 column to the left.



Move Selection | Wait up to 1 sec | Run error task

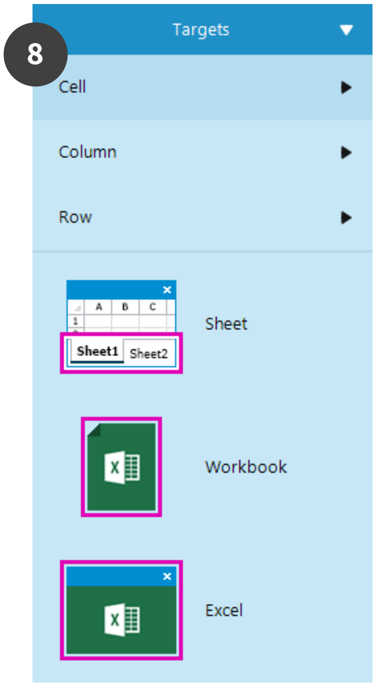
General

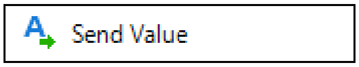
☑ Move column: Left Distance: 1

□ Move row: Down Distance: 1

By moving the current cell to the left 1 column, a new current cell will be established. Finally, input the desired variable to the current cell.

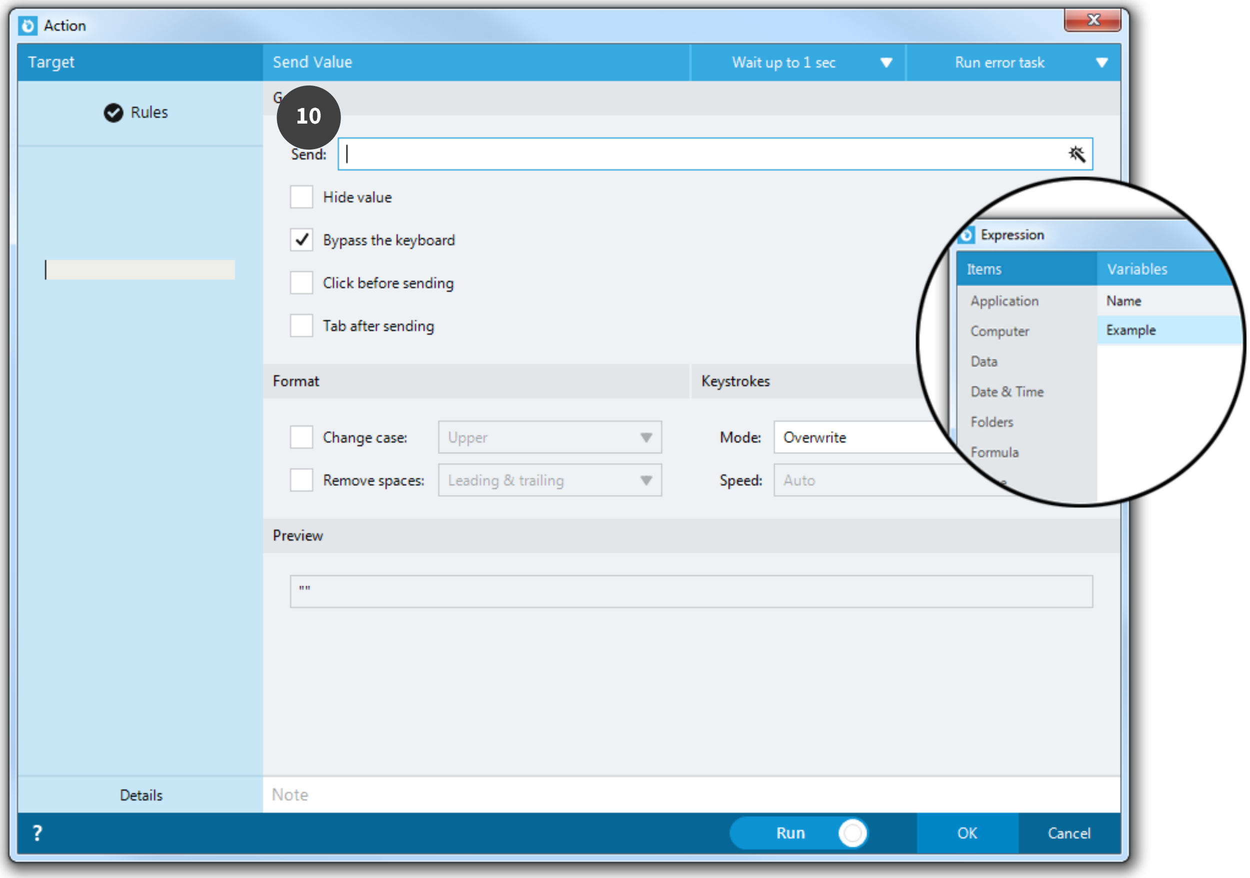
8) Hover over the blue targets bar, then select Cell. Selecting Cell is the same as choosing the Current Cell option, just a fraction faster.



9) 

Choose the *Send Value* action.

10) Input the appropriate variable.



Send: |

□ Hide value

☑ Bypass the keyboard

□ Click before sending

□ Tab after sending

Format

□ Change case: Upper

□ Remove spaces: Leading & trailing

Keystrokes

Mode: Overwrite

Speed: Auto

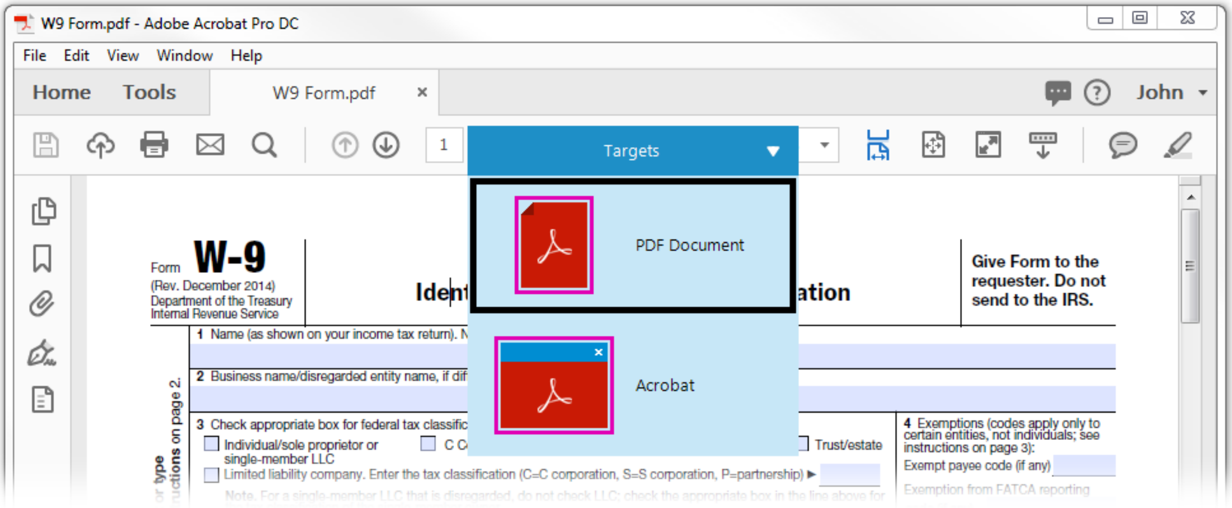
Preview

""

Note

PDF Automation

Working with PDF forms is easy – simply drag and drop the bullseye onto the PDF document to connect, similar to that of Excel. You can target individual fields, the entire document, or Acrobat itself to create various actions.


W9 Form.pdf - Adobe Acrobat Pro DC
File
Edit
View
Window
Help
Home
Tools
W9 Form.pdf
John
1
Targets
PDF Document
Acrobat
Form W-9
(Rev. December 2014)
Department of the Treasury
Internal Revenue Service
Ident
ation
Give Form to the requester. Do not send to the IRS.
1 Name (as shown on your income tax return). N
2 Business name/disregarded entity name, if dif
3 Check appropriate box for federal tax classific
Individual/sole proprietor or single-member LLC
C C
Trust/estate
Limited liability company. Enter the tax classification (C=C corporation, S=S corporation, P=partnership) ▶
4 Exemptions (codes apply only to certain entities, not individuals; see instructions on page 3):
Exempt payee code (if any)
Exemption from FATCA reporting

TARGETING OPTIONS

Foxtrot will allow you to interact with forms themselves, or the application as a whole. The following are available when hovering over the *Targets* icon that sits atop the PDF document.

- PDF Document will allow for the interaction with the document itself. Actions such as *Close PDF*, *Save PDF*, and *Print PDF* will be available when targeting this option.
- Acrobat will provide actions such as *Open PDF* and *Set Connection*, similar to the actions Excel offers.

Aside from the options available under the *Targets* icon, you can target a field, paragraph, dropdown menu, etc. just as you would with a web or Windows application.

Emulator Automation

Foxtrot is also able to interact within an emulator or "green screen" environment to automate data-related tasks. Advanced users may wish to toggle between emulator and browser environments to perform more complex tasks. Similar to Excel and PDF technology, Foxtrot will make a direct Connection to the emulator and perform the actions written by the user.

SUPPORTED EMULATORS

Before beginning a project within an emulator environment, first determine if Foxtrot supports the desired emulator. As of this writing, the following emulators are supported:

- Attachmate EXTRA! X-treme
- Attachmate INFOConnect
- Attachmate Reflection Workspace
- Aviva
- CAIL Tandem
- Esker TUN
- IBM Client Access
- IBM iAccess
- IBM iAccess – ACS (Java)
- INFO Connect
- Jolly Giant QWS3270 PLUS
- Passport
- MicroFocus Rumba Desktop
- NetManage Rumba
- Century Software TinyTERM Plus
- Open Text HostExplorer
- Rocket BlueZone
- Wall Data Rumba 2000 AS400

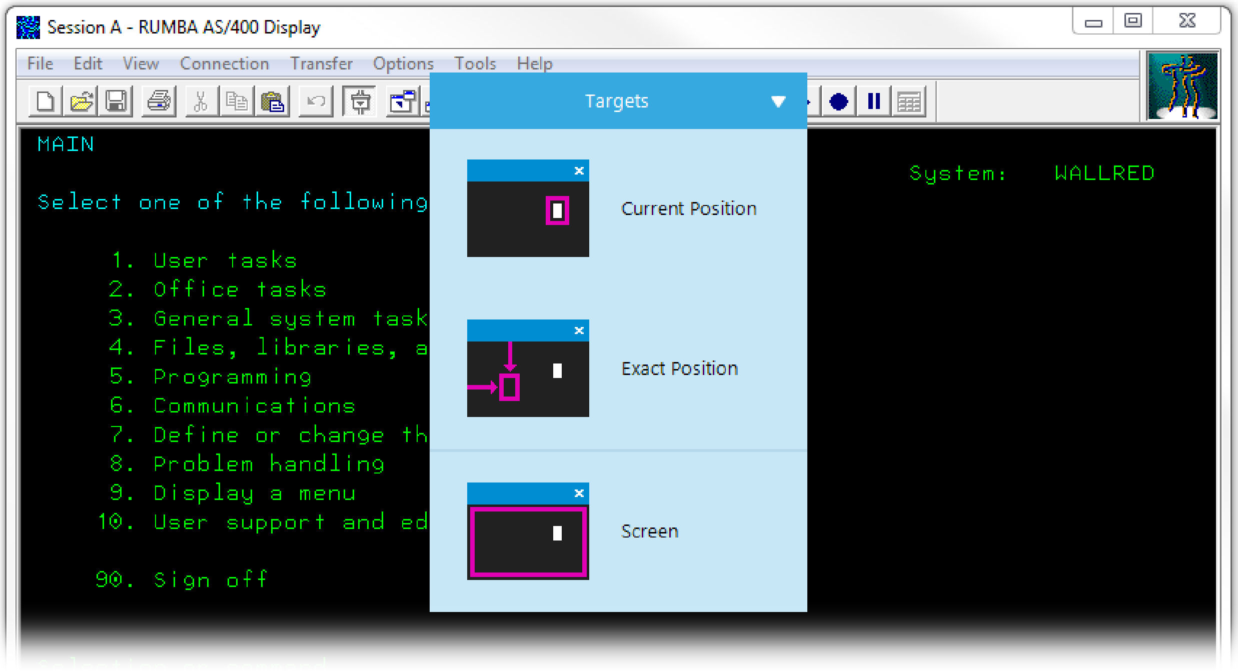
CREATING CONNECTIONS

Creating a Connection to an emulator is similar to that of connecting to Excel, but with one extra step: the emulator must have a unique short name assigned. The methods for configuring the short name are different for every emulator. Please contact a system administrator to assist in setting up a short name, or verifying that one is set up if unsure.

Once the short name is properly configured and verified, create a connection by targeting anywhere over the emulator screen and creating an action. The *Emulator* view in the Project Pane will display all emulator connections in the current project.

TARGETING OPTIONS

Targeting the emulator will allow for the targeting of a specific coordinate on the screen. Hovering over the blue *Targets* banner will display all additional targeting options.


Session A - RUMBA AS/400 Display
File Edit View Connection Transfer Options Tools Help
Targets
Current Position
Exact Position
Screen

```
MAIN
                                                              System:   WALLRED
Select one of the following

     1. User tasks
     2. Office tasks
     3. General system task
     4. Files, libraries, a
     5. Programming
     6. Communications
     7. Define or change th
     8. Problem handling
     9. Display a menu
    10. User support and ed

    90. Sign off
```

- Cursor Position interacts with the current position of the cursor. If an action is configured to the cursor position, Foxtrot could potentially interact with Row 1, Column 1 in one instance but then Row 1, Column 16 in another instance, depending on the current active cursor position.

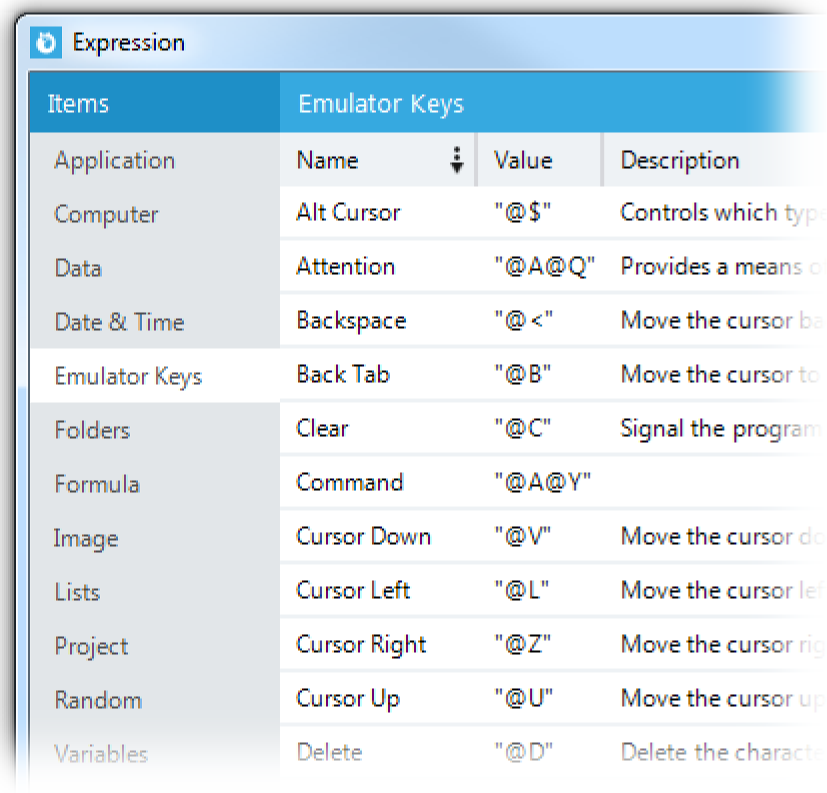
- Exact Position will interact with a fixed coordinate. Choosing this option will require manually entering a specific row and column to interact with. Note that Tokens may be used as well. If the coordinate is unknown, simply target the desired position on the screen and let Foxtrot do the rest.

- Screen allows for the interaction of the entire emulator screen. Targeting the screen option allows Foxtrot to unlock the screen, or search the screen for specific text.

EMULATOR KEY TOKENS

When opening the Expression Builder while in an emulator screen, the Keyboard tokens are replaced with Emulator Keys. Multiple steps of an entire process may be arranged into one *Send Value* action.

There is no limit to the number of tokens that can be combined.



Expression

Items

Application
Computer
Data
Date & Time
Emulator Keys
Folders
Formula
Image
Lists
Project
Random
Variables

Emulator Keys

Name	Value	Description
Alt Cursor	"@$"	Controls which typ
Attention	"@A@Q"	Provides a means o
Backspace	"@<"	Move the cursor ba
Back Tab	"@B"	Move the cursor to
Clear	"@C"	Signal the program
Command	"@A@Y"	
Cursor Down	"@V"	Move the cursor do
Cursor Left	"@L"	Move the cursor lef
Cursor Right	"@Z"	Move the cursor rig
Cursor Up	"@U"	Move the cursor up
Delete	"@D"	Delete the characte

Lists

Lists are an extreme powerful, simple to use tool. They can be used to avoid having multiple *If* actions, as they allow for countless complex decisions to be made at once. Lists can be used to:

- Reference an alternate name or value
- Interact with a collection of files or folders
- Store a collection of information
- Data mapping

There are two different types of lists; a Custom List and a Files & Folders List. For the former, a user will create rows and columns of information that can be referenced at any time within the script. The latter is used to obtain information from a collection of files or folders in a particular directory.

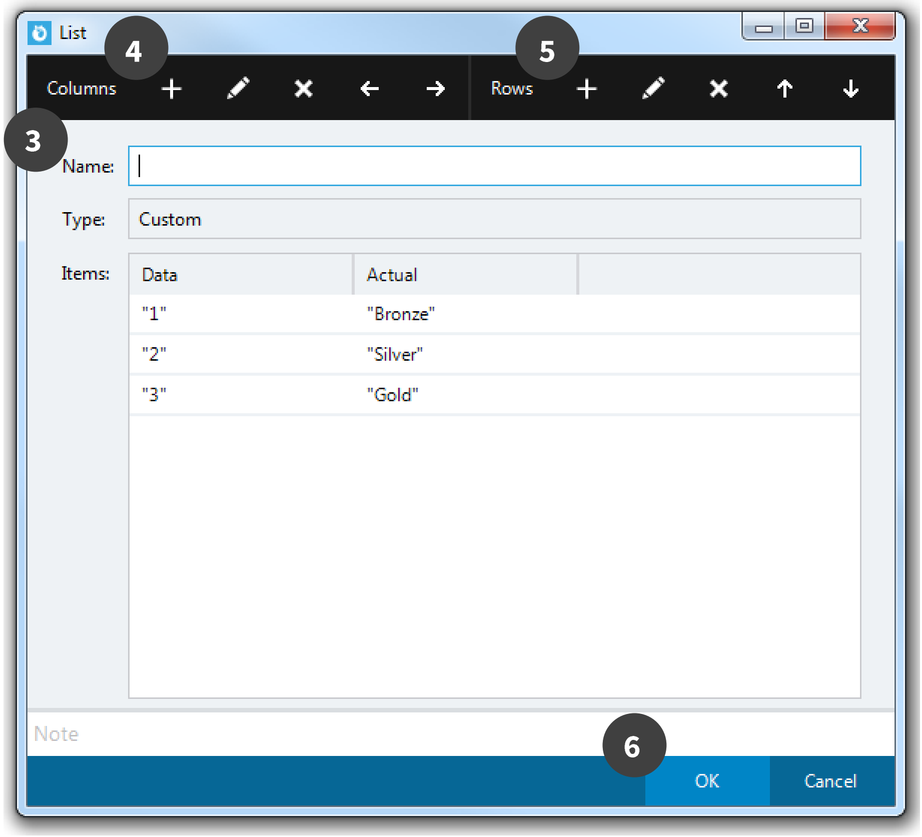
CUSTOM LISTS

A Custom List is used to reference an alternate value, or hold a collection of information that meets some certain criteria. For instance, you could create 20 *If* actions to determine various area codes in respect to their city, or you could search a List for an area code and obtain the corresponding value in a single action. At a minimum, a Custom List must have two columns and one row. When creating a Custom List, first think about what information to include in the List, and the purpose of it. Essentially, Foxtrot will use Lists to search one particular column, and return another.

For example, a customer may purchase one of three plans of product support: Bronze, Silver or Gold. In a spreadsheet, this information is displayed as 1, 2 or 3, where 1 is Bronze, 2 is Silver and 3 is Gold. When sending the information to the screen, the application only recognizes the text “Bronze”, “Silver”, or “Gold”. These values must be translated before they are sent to the screen.

CREATING A CUSTOM LIST

1) Click Add (the large +) at the top of the item list
2) Select List from the list
3) Give the List a unique name
4) Click the + next to Columns, and name the column. For example: “Data”, and the second column: “Actual”
5) Click the + next to Rows. In the example, the first row would have the values “1” and “Bronze” in the respective columns, and so on.
6) When all desired values have been entered, click OK. If following the example, it should look similar to what is pictured.



Name:

Type: Custom

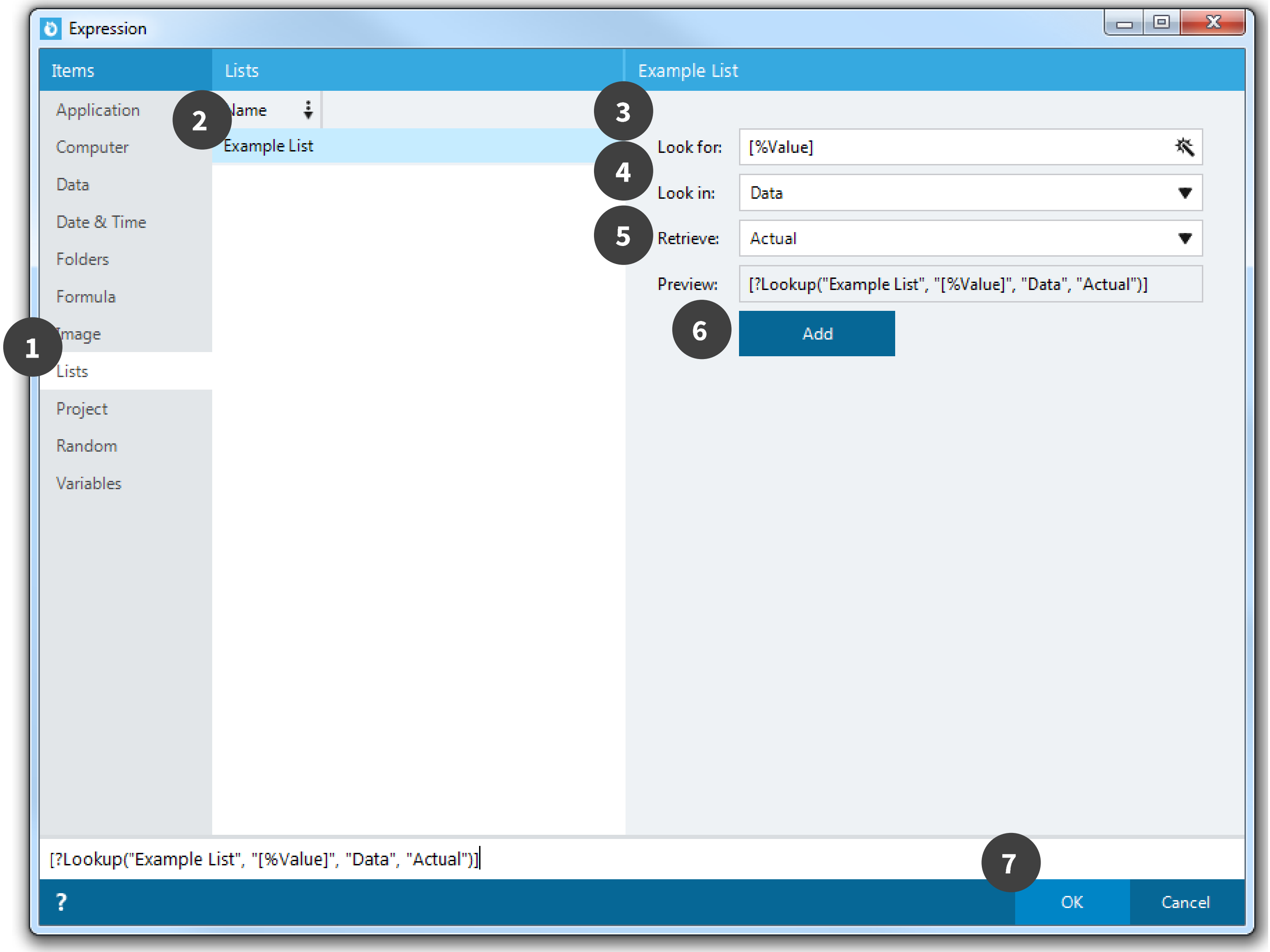
Items:

Data	Actual
"1"	"Bronze"
"2"	"Silver"
"3"	"Gold"

USING A CUSTOM LIST

Any List can be accessed through the Expression Builder.

1) In the Expression Builder, choose the Lists item from the left.
2) Choose the desired List to search
3) Enter a value to *Look for*. This will commonly be a data or variable token.
4) Select a column to *Look in*.
5) Select a column to *Retrieve*.
6) Click Add to confirm the token in the expression builder.
7) Click OK to use the token in your action.



Look for: [%Value]

Look in: Data

Retrieve: Actual

Preview: [?Lookup("Example List", "[%Value]", "Data", "Actual")]

[?Lookup("Example List", "[%Value]", "Data", "Actual")]

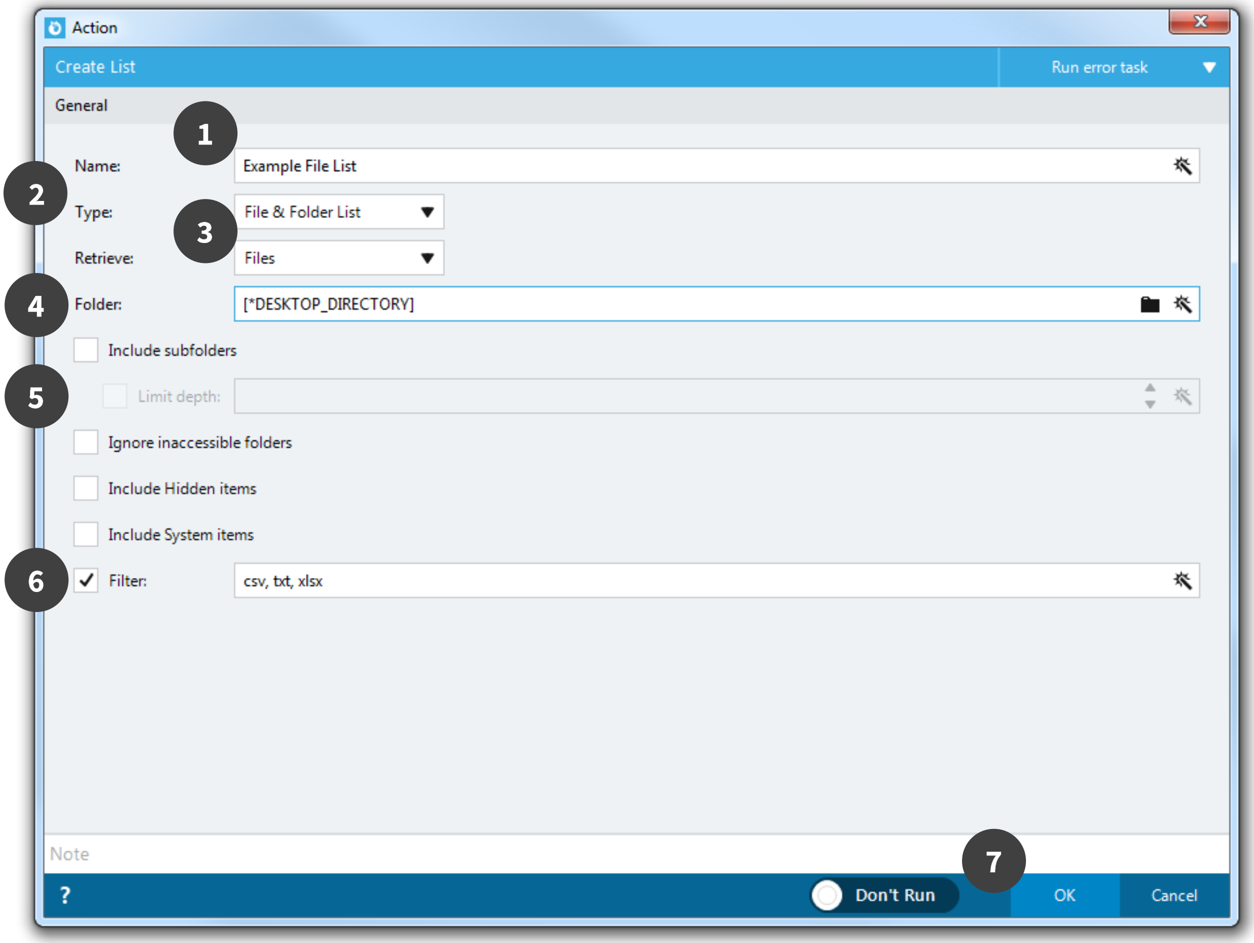
FILE & FOLDER LISTS

File & Folder Lists allow Foxtrot to access collections of files or folder on a local or remote machine. Such Lists can be customized to filter certain file types, include hidden items, items in subfolders, and more. Files actions work almost exclusively with File & Folder Lists. File & Folder Lists and their respective Loop tokens go hand-in-hand. Before jumping too far ahead, a File & Folder list must first be created.

CREATING A FILE & FOLDER LIST

1)

Select the Create List action, located in the Action List under the Lists category.
2) Give the List a unique name.
3) Keep the default *Type* of File & Folder List
4) Choose to retrieve only File, only Folders, or both Files & Folders
5) Choose a folder. This will be the location in which Foxtrot looks for the desired files and/or folders.
6) Select any options, or filter file types as necessary. Separate any filtered file types with a comma.
7) Click OK to create the action.



Name: Example File List

Type: File & Folder List

Retrieve: Files

Folder: [*DESKTOP_DIRECTORY]

□ Include subfolders

□ Limit depth:

□ Ignore inaccessible folders

□ Include Hidden items

□ Include System items

☑ Filter: csv, txt, xlsx

USING FILE & FOLDER LISTS

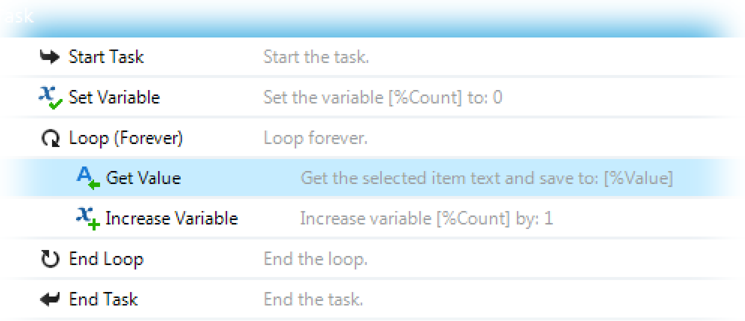
For File & Folder Lists, the number of items is the same as the number of files or folders in the specified folder. When editing the List, you will notice a number of attributes about each item within the list; the name, location, size, etc.

A File & Folder List may be referenced the same way as a Custom List, however, referencing a value is much easier within a Loop. This will be covered in a later chapter.

Loops

A *Loop* gives the ability to repeat a set of actions a predetermined number of times. In fact, a project running on records is essentially a *Loop* which runs against the database. A *Loop* can be as simple or complex as desired, as it allows for the repetition of steps to be performed. *Loops* can be run to:

- Navigate a web table or a List
- Repeat a series of actions or an entire process



Start Task	Start the task.
Set Variable	Set the variable [%Count] to: 0
Loop (Forever)	Loop forever.
Get Value	Get the selected item text and save to: [%Value]
Increase Variable	Increase variable [%Count] by: 1
End Loop	End the loop.
End Task	End the task.

Creating a *Loop* action will also create the closing *End Loop* action. Anything between these two actions will be looped according to the loop type specified. There are three different types of loops.

LOOPING A SPECIFIC NUMBER OF TIMES

Creating a *Loop* will prompt the user to choose the *Loop type*. Running a Loop 15 times, for example, will run the actions between the *Loop* and *End Loop* 15 times. Of course, a Token may also be used to set the number of times a *Loop* should run. Once the *Loop* has reached the last iteration, the project will exit the Loop automatically and continue with the actions following the End Loop.

LOOPING FOREVER

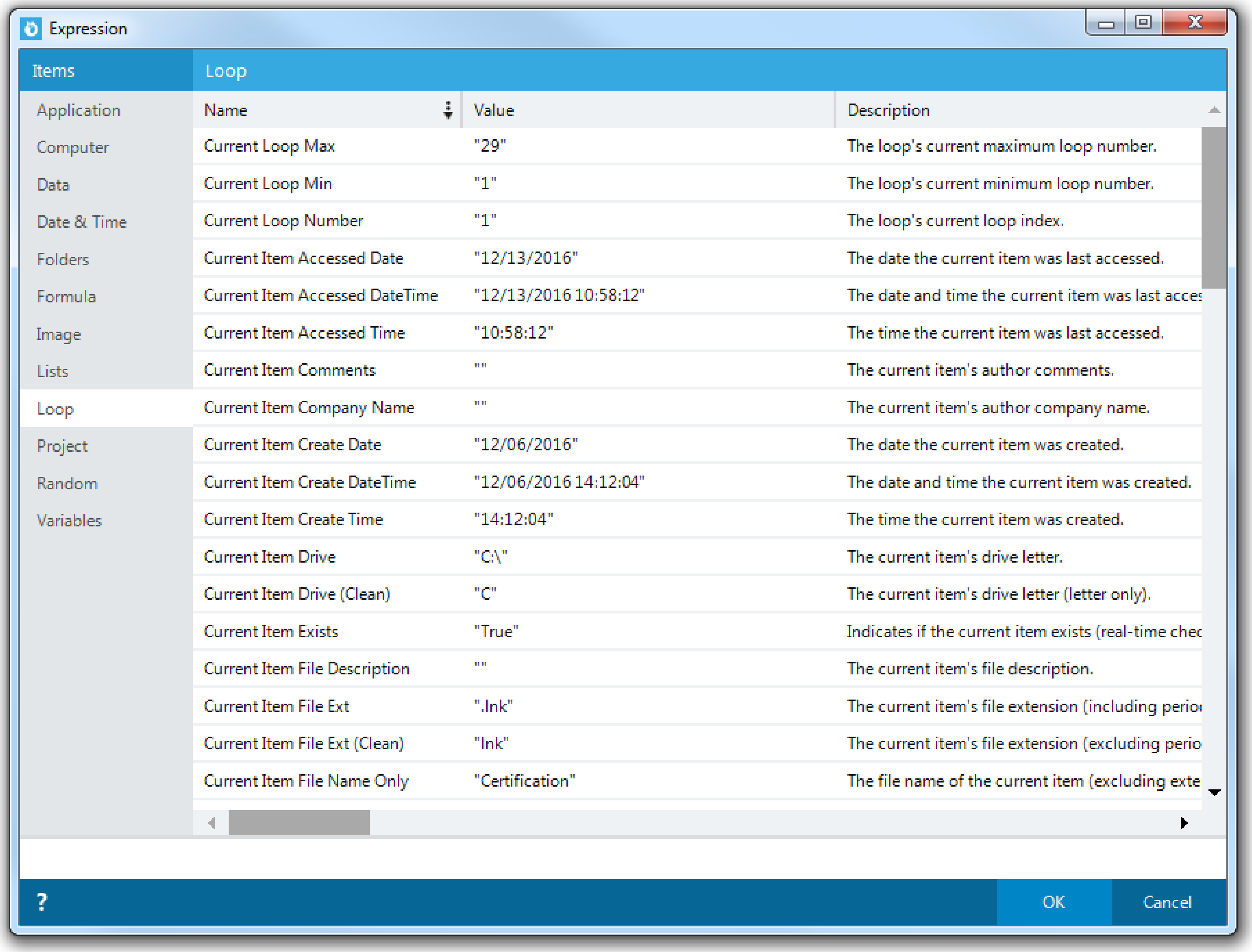
Running a Loop forever will essentially run the actions between the *Loop* and *End Loop* forever – but more practically, until an *Exit Loop* action is encountered (this will probably be triggered by an *If* action). By running a Loop forever, and specifying when to end, a Loop may run dynamically until something of interest occurs like a Target appearing on the screen, a file appearing a directory, or a Variable reaching a desired value.

LOOPING THROUGH A LIST

A *Loop* may also run against a List. Doing so will first require that a List is created. When executed, Foxtrot will run the Loop according to the number of entries in the List; if the List has 6 entries, the Loop will run 6 times. When using a File & Folder List, Foxtrot offers special tokens to access the information contained in this unique List type.

LOOP TOKENS

All Loop types offer 3 special Tokens: Current Loop Min, Current Loop Max, and Current Loop Number. These tokens offer information about the loop itself, and can be utilized in many ways.



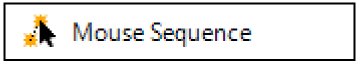
Name	Value	Description
Current Loop Max	"29"	The loop's current maximum loop number.
Current Loop Min	"1"	The loop's current minimum loop number.
Current Loop Number	"1"	The loop's current loop index.
Current Item Accessed Date	"12/13/2016"	The date the current item was last accessed.
Current Item Accessed DateTime	"12/13/2016 10:58:12"	The date and time the current item was last acces
Current Item Accessed Time	"10:58:12"	The time the current item was last accessed.
Current Item Comments	""	The current item's author comments.
Current Item Company Name	""	The current item's author company name.
Current Item Create Date	"12/06/2016"	The date the current item was created.
Current Item Create DateTime	"12/06/2016 14:12:04"	The date and time the current item was created.
Current Item Create Time	"14:12:04"	The time the current item was created.
Current Item Drive	"C:\"	The current item's drive letter.
Current Item Drive (Clean)	"C"	The current item's drive letter (letter only).
Current Item Exists	"True"	Indicates if the current item exists (real-time chec
Current Item File Description	""	The current item's file description.
Current Item File Ext	".lnk"	The current item's file extension (including perio
Current Item File Ext (Clean)	"lnk"	The current item's file extension (excluding perio
Current Item File Name Only	"Certification"	The file name of the current item (excluding exte

When using a File & Folder Loop, you will notice many more tokens available to use. These tokens provide access to all of the information provided in the List, without having to use a List lookup token. These tokens will reference the current item in the List. Values for List item 1 are visible when the Loop number is 1, and values for List item 2 are visible when the Loop index is 2, and so on.

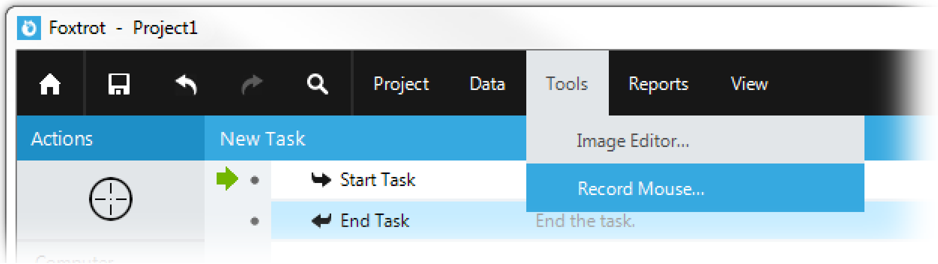
Mouse Sequences

A process may require a series of uninterrupted mouse movements to better interact with a target. Hovering over a menu to display its contents, or performing a right-click to display an options menu are perfect examples of appropriate times to utilize a *Mouse Sequence*.

CREATING A MOUSE SEQUENCE



To start creating a *Mouse Sequence*, navigate to *Tools* in the menu bar. Clicking *Record Mouse* will display Foxtrot's *Mouse Recorder*, in which every mouse action is recording, logging all clicks and hovers as they are executed.

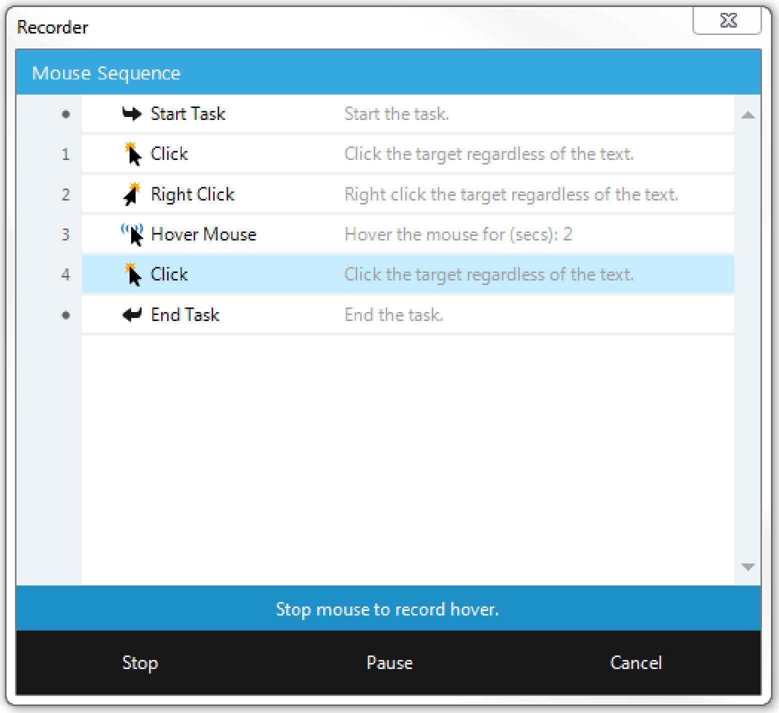


The following Actions are recorded within a *Mouse Sequence*:

- *Hover Mouse*
- *Right Click*
- *Click*

To record a *Hover Mouse* Action, simply hover over the specific target for five seconds. Foxtrot will display an informational banner within the Mouse Recorder window while you are hovering. After five seconds, a *Hover Mouse* action will be recorded.

When a specific sequence has been performed, clicking Done on the Mouse Recorder will create a *Mouse Sequence* Action.



●	Start Task	Start the task.
1	Click	Click the target regardless of the text.
2	Right Click	Right click the target regardless of the text.
3	Hover Mouse	Hover the mouse for (secs): 2
4	Click	Click the target regardless of the text.
●	End Task	End the task.

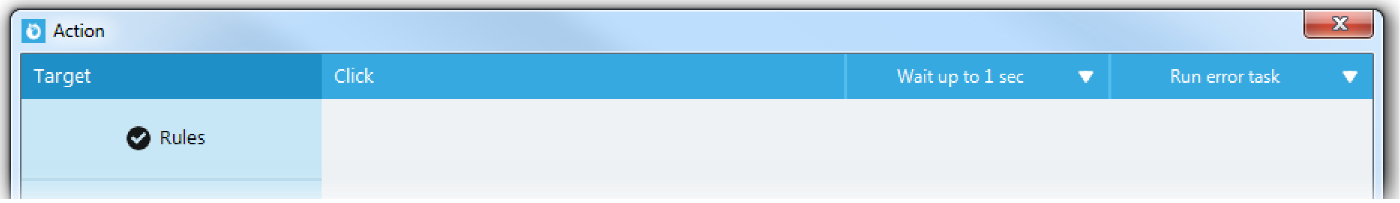
Stop mouse to record hover.

EDITING A MOUSE SEQUENCE

After you've recorded and saved your *Mouse Sequence* action, you may edit the Actions within the Sequence. Each element of a *Mouse Sequence* may be edited just like any other Action. When editing a *Mouse Sequence* action, more advanced users may wish to add additional Actions, such as *Wait* or an *If* Action or two. Notice that only certain Actions are available while editing a *Mouse Sequence*.

The Action Builder

Each Action has different options that are completely customizable depending on the specifications of the project. Upon creating or editing most Actions, the Action Builder will display three additional options along the top: Rules, Time, and On Error.



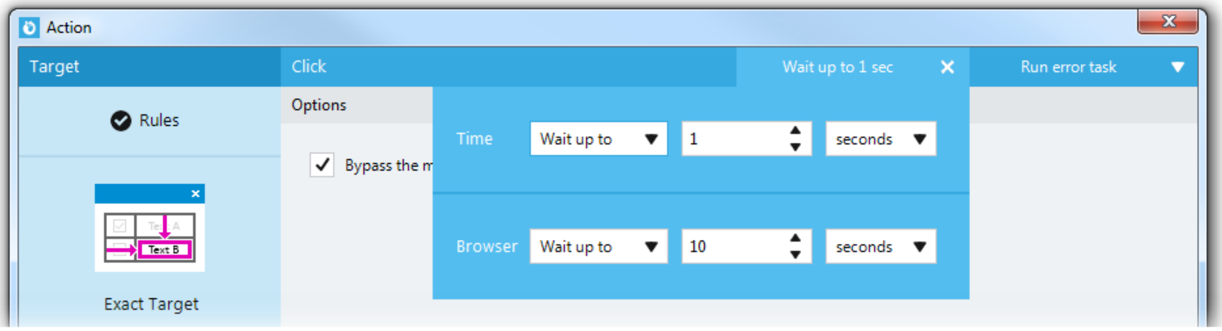
Action

Target | Click | Wait up to 1 sec ▼ | Run error task ▼

TIME

The *Time* menu provides options regarding how long Foxtrot will wait for the target to appear before an error occurs. Depending on the Action, there may be no timing options available. The following are available under the *Time* menu:

- Do Not Wait will not wait for a target to appear/disappear, but instead will continue with the next Action in the project.
- Wait Up To is the default selection. Foxtrot will up to X seconds for the target, where X is the specified value. The default is to wait up to 3 seconds.
- Wait Forever will wait until the target is found, no matter how long it takes.



Action

Target | Click | Wait up to 1 sec ✕ | Run error task ▼

Options

☑ Bypass the m

Time: Wait up to ▼ | 1 | seconds ▼

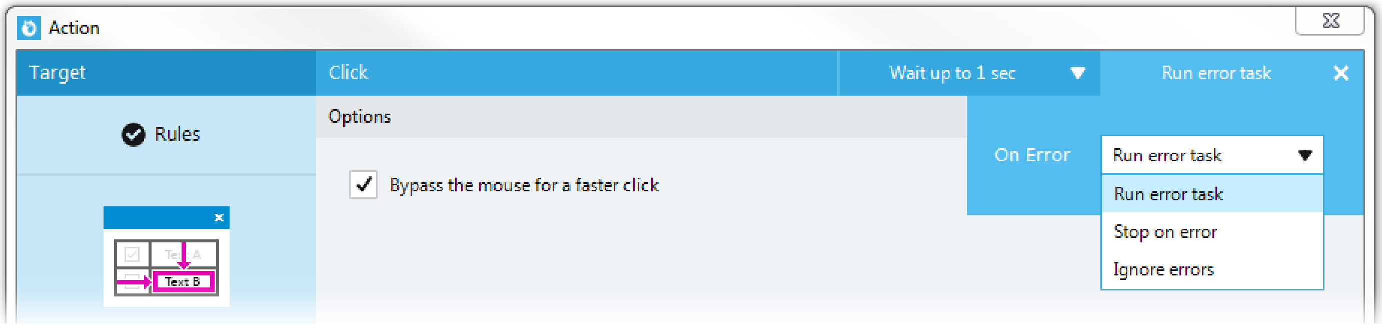
Browser: Wait up to ▼ | 10 | seconds ▼

Actions interacting with a web browser will have an additional option called *Browser*. While navigating from page-to-page, Foxtrot will know to wait for the browser to "catch up". In some cases, the browser never reports that it is ready. Foxtrot may be instruction to automatically continue if the browser is busy for a certain amount of time.

ON ERROR

Some Actions also have error handling options within the *Errors* tab. If an error occurs, Foxtrot can:

- Stop running the project and display an error message
- Ignore and Continue to disregard the error and continue running the project
- Run Error Task is the default setting, and will run an Error Task if one is present. If there is no Error Task, this option will instead behave like the *Stop* option.



Action

Target | Click | Wait up to 1 sec | Run error task

Rules

Options

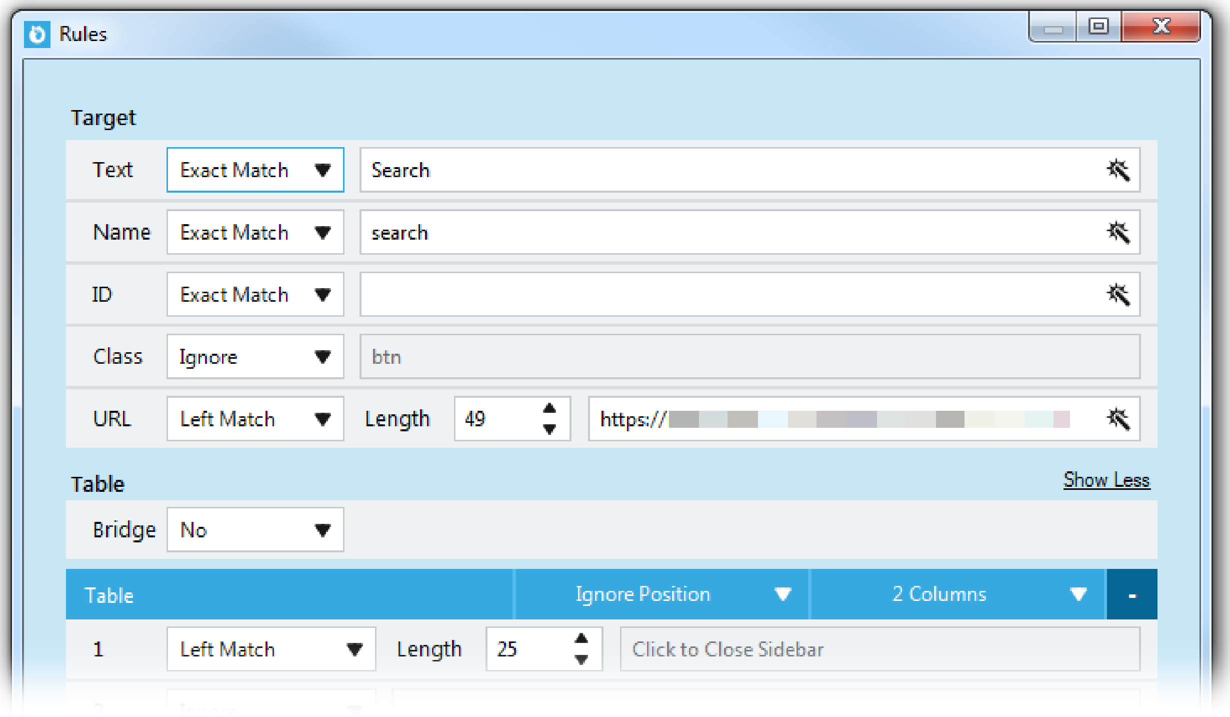
☑ Bypass the mouse for a faster click

On Error: Run error task

- Run error task
- Stop on error
- Ignore errors

RULES

The *Rules* button will only have options for some Actions. The options in the *Rules* menu will instruct Foxtrot to either restrict the information used to locate a target, or more tightly define it. Additionally, there may be different options within the *Rules* menu, depending on what has been targeted. For the simplicity of basic training, this guide will only review the *Text* option.



Rules

Target

Text	Exact Match	Search	
Name	Exact Match	search	
ID	Exact Match		
Class	Ignore	btn	
URL	Left Match	Length 49	https://[redacted]

Table

Show Less

Bridge: No

Table | Ignore Position | 2 Columns | -

1	Left Match	Length 25	Click to Close Sidebar

The *Text* option tells Foxtrot what the text of the object reads. Foxtrot may use this text to help locate a target. If the text changes on a record-to-record basis, it is wise to change this option to *Ignore*. Additionally, this option may be modified to only use the leading characters of the *Text*. Notice that Token values may be used in these fields, as well.

Dictionary of Terms

- Actions – Instructions provided to perform a process.
- Action Builder – The Action Builder contains specific parameters available for the selected Action.
- Breakpoint – a predetermined pause within a Project. When a breakpoint is encountered, Foxtrot will automatically pause the project at the Action containing the breakpoint (this Action will not be executed). Breakpoints are especially helpful when testing projects.
- Bypass Keyboard – A method used to Send Value without use of the keyboard. This option will only appear when the target resides in a browser window. Using this option can increase the speed of your project, if it is supported by the target application.
- Bypass Mouse - A method used to Click without use of the mouse. This option will only appear when the target resides in a browser window. Using this option can increase the speed of your project, if it is supported by the target application.
- Client Machine – Typically referred to as a machine that points to a License Host.
- Comparison – A parameter used within an *If* Action to compare two values.
- Concurrent License – Foxtrot may be installed on an endless number of machines, but the number of simultaneous running instances is limited to the number of licenses.
- Connection – A unique letter used to establish communication between applications (Excel, PDF, and emulators) or systems.
- Current Row/Column/Cell – The current active row, column or cell in an Excel spreadsheet.
- Current Position – The current active position of the cursor in an emulator or “green screen” environment.
- Custom List – A collection of information that is used to map values to a respective counterpart.
- Don’t Run – To classify a record to “Don’t Run”, or completed.
- Else – A clause used within an *If* Action that allows for an alternate set of Actions to be performed in the event that an *If* Action does not evaluate as true. If an *If* Action evaluates as false, and an *Else* clause is used, the Actions between *Else* and *End If* will be performed.
- Error Task – A Task that is designated for handling errors. In the event of an error, Foxtrot will automatically run the Actions within the Error Task.
- Exact Position – A cursor position within an emulator or “green screen” that is specified by a user or Token.
- Exact Cell/Column/Row – A cell, column, or row position within an Excel spreadsheet that is specified as a hard coded value or Token.
- Exact Target – A targeting option available when targeting a web table. When the Exact Target option is used, Foxtrot will not use table technology, unless using a Start Highlight Action. Instead, Foxtrot will interact with whatever the user has targeted.
- Expression Builder – Access to Tokens.
- Field – A column within your database.

- Files & Folders List – A collection of files or folders in a directory.
- Highlight – A method used to dynamically interact with a set of targets within a web table.
- Highlighted Target – A targeting option available when targeting a web table, specifically, when targeting a pre-existing Highlight. When the Highlighted Target option is used, Foxtrot will interact with the cell of the table that is currently highlighted.
- License Host – The location of the licenses. Typically, this machine will also maintain audit trails and security logins.
- List – Allows for the lookup of alike values. For example, a List provides the power to map state capitals to their time zone.
- Log – A csv file that is created by Foxtrot with the use of a *Write Log* Action.
- Managing Machine – Machine that maintains all licensing information. This is the only machine that can add, remove, or migrate licenses.
- Mapped Drive – A method used to access shared storage area on another computer over a network.
- Mouse Recorder – Special technology that will record a fluid sequence of mouse movements.
- Mouse Sequence – A collection of mouse movements that are consolidated into a single Action.
- Normal Task – A type of Task which serves as a collection of Actions. The *Run Task* Action is used to call upon a Normal Task.
- Project – An automated job that Foxtrot understands as hand-given instructions.
- Project Pane – Home to different Views within your project including any Data views or created Variables.
- Record – A row within your database.
- Run – To classify a record to “Run” or not completed.
- Run Arrow – A green arrow located within the Task Pane that notifies a user which Action will be performed next. By clicking “Run”, Foxtrot will begin performing Actions beginning with the Action that this arrow points to.
- Run Bar – Foxtrot’s control center – manages the running of your project.
- Selector/Bullseye/Targeting Tool – Tool used to create Actions with a drag & drop method.
- Smart Actions – Actions that are presented dependent upon the target. For example, Excel Actions are only available when targeting an Excel Spreadsheet.
- Start Here Task – Is considered the main Task within a project. The very first Action within a project will be the *Start Task* Action within the Start Here Task, and the very last Action is the *End Task* Action at the end of the Start Here Task. Rewinding a project will bring the Run Arrow to the top of the Start Here Task.
- Stepper – Allows for the single execution of the selected Action.
- Table – A targeting option available when targeting a web table. When the Table option is used, Foxtrot can interact with the entire table itself, as opposed to a single cell in the Highlighted Target option, or a single Target in the Exact Target option.

- Target – An element on the screen native to the Target Application that Foxtrot interacts with in the form of an Action. A button to be clicked or a field to send data to are examples of Targets.
- Target Application – The application(s) that Foxtrot interacts with.
- Target Caption – The caption of the targeted object that Foxtrot uses to locate a target.
- Tasks – Sub-routines within a project that are used to maintain organization of a project.
- Task Pane – Center of the application, home to all available and created Actions.
- Token – Individual values displayed within the Expression Builder. Tokens are defined codes that are translated from supplied information such as data field values, system dates, Variables, and the Windows clipboard value.
- Toolbar – Manages your project and data
- UNC Path – Universal Naming Convention, specifies a unique location in a file system.
- Variables – Custom Tokens that are used primarily to copy and evaluate data.