

Instructor Guide

Automating Tasks Using IBM Robotic Process Automation with Automation Anywhere

Course code WB502 / ZB502 ERC 3.0



March 2019 edition

Notices

This information was developed for products and services offered in the US.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

© Copyright International Business Machines Corporation 2018, 2019.

This document may not be reproduced in whole or in part without the prior written permission of IBM.

US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Trademarks	xiii
Course description	xiv
Agenda	xv
Unit 1. Introducing robotic process automation and bots	1-1
Unit objectives	1-2
Topics	1-3
1.1. What is Robotic Process Automation (RPA)?	1-4
What is Robotic Process Automation (RPA)?	1-5
What is Robotic Process Automation (RPA)?	1-6
Software robots are similar to physical robots	1-7
Benefits of robotic process automation	1-8
Challenges of Robotic Process Automation	1-9
1.2. Use cases and industries	1-10
Use cases and industries	1-11
Common use cases (1 of 2)	1-12
Common use cases (2 of 2)	1-13
Good uses for Robotic Process Automation	1-14
Industries that use Robotic Process Automation (1 of 2)	1-15
Industries that use Robotic Process Automation (2 of 2)	1-16
1.3. Introducing IBM Robotic Process Automation with Automation Anywhere	1-17
Introducing IBM Robotic Process Automation with Automation Anywhere	1-18
IBM partnership with Automation Anywhere	1-19
IBM Robotic Process Automation with Automation Anywhere architecture	1-20
1.4. Robotic Process Automation and Business Automation Workflow	1-21
Robotic Process Automation and Business Automation Workflow	1-22
RPA and the IBM Digital Business Automation platform	1-23
Comparing IBM RPA with Automation Anywhere and IBM BAW	1-24
Unit summary	1-25
Review questions	1-26
Review answers	1-27
Unit 2. Introducing IBM Robotic Process Automation with Automation Anywhere	2-1
Unit objectives	2-2
Topics	2-3
2.1. IBM Robotic Process Automation with Automation Anywhere product overview	2-4
IBM Robotic Process Automation with Automation Anywhere product overview	2-5
IBM Robotic Process Automation with Automation Anywhere components (1 of 5)	2-6
IBM Robotic Process Automation with Automation Anywhere components (2 of 5)	2-7
IBM Robotic Process Automation with Automation Anywhere components (3 of 5)	2-8
IBM Robotic Process Automation with Automation Anywhere components (4 of 5)	2-9
IBM Robotic Process Automation with Automation Anywhere components (5 of 5)	2-10
High-level architecture	2-11
Topology overview	2-12
2.2. Introducing the Control Room	2-13
Access to IBM Robotic Process Automation with Automation Anywhere components and features	2-14
Introducing the Control Room	2-15
Control Room overview (1 of 2)	2-16

Control Room overview (2 of 2)	2-17
Dashboard features	2-18
Home dashboard	2-19
Bots dashboard	2-20
Audit dashboard	2-21
Bot management features (1 of 2)	2-22
Bot management features (2 of 2)	2-23
Workload management features (1 of 3)	2-24
Workload management features (2 of 3)	2-25
Workload management features (3 of 3)	2-26
Server settings	2-27
Administration features (1 of 2)	2-28
Administration features (2 of 2)	2-29
2.3. User management	2-30
User management	2-31
Creating roles and users	2-32
Creating a user (1 of 4)	2-33
Creating a user (2 of 4)	2-34
Creating a user (3 of 4)	2-35
Creating a user (4 of 4)	2-36
Unit summary	2-37
Review questions	2-38
Review answers	2-39
Exercise 1: Exploring the Control Room	2-40

Unit 3. Introducing the Enterprise Client	3-1
Unit objectives	3-2
Topics	3-3
3.1. Creating bots	3-4
Creating bots	3-5
What is a bot?	3-6
Types of bots	3-7
How to create a bot	3-8
Steps for creating bots	3-9
3.2. Overview of the Enterprise Client	3-10
Overview of the Enterprise Client	3-11
Enterprise Client: Dashboard	3-12
Dashboard: Task-relevant activities panel	3-13
Accessing bot files	3-14
Enterprise Client: Workbench	3-15
3.3. Overview of recorders	3-16
Overview of recorders	3-17
Overview of recorders	3-18
Screen recorder	3-19
Smart recorder	3-20
Web recorder	3-21
Using the Workbench with a recorder	3-22
3.4. Coding in the Workbench	3-23
Coding in the Workbench	3-24
Creating a bot	3-25
The Workbench	3-26
The Actions List	3-27
Working with the Commands list	3-28
Naming conventions	3-29
Good versus challenging formats for bot development	3-30
Unit summary	3-31

Review questions	3-32
Review answers (1 of 2)	3-33
Review answers (2 of 2)	3-34
Exercise 2: Creating a basic bot	3-35
Exercise introduction (1 of 2)	3-36
Exercise introduction (2 of 2)	3-37
Unit 4. Working with variables, loops, and strings	4-1
Unit objectives	4-2
Topics	4-3
4.1. Overview	4-4
Overview	4-5
Overview	4-6
Variables, Loop commands, and String Operation commands	4-7
4.2. Variables	4-8
Variables	4-9
Types of variables	4-10
Variable Manager	4-11
Creating variables (1 of 2)	4-12
Creating variables (2 of 2)	4-13
Guidelines for naming variables	4-14
Using variables in bot commands	4-15
Variable Operation command	4-16
4.3. Loop commands	4-17
Loop commands	4-18
Overview of Loop commands	4-19
Loop structures	4-20
Start Loop subcommands (1 of 4)	4-21
Start Loop subcommands (2 of 4)	4-22
Start Loop subcommands (3 of 4)	4-23
Start Loop subcommands (4 of 4)	4-24
Working with Loop commands (1 of 2)	4-25
Working with Loop commands (2 of 2)	4-26
4.4. String Operation commands	4-27
String Operation commands	4-28
String Operation command overview (1 of 2)	4-29
String Operation command overview (2 of 2)	4-30
String Operation subcommands (1 of 4)	4-31
String Operation subcommands (2 of 4)	4-32
String Operation subcommands (3 of 4)	4-33
String Operation subcommands (4 of 4)	4-34
Working with String Operation commands	4-35
4.5. Exercise command focus: Excel command	4-36
Exercise command focus: Excel command	4-37
Introduction to the Excel commands	4-38
Introduction to the Excel commands	4-39
Excel subcommands (1 of 5)	4-40
Excel subcommands (2 of 5)	4-41
Excel subcommands (3 of 5)	4-42
Excel subcommands (4 of 5)	4-43
Excel subcommands (5 of 5)	4-44
4.6. Introducing the exercise case study	4-45
Introducing the exercise case study	4-46
Introduction to the exercise case study (1 of 2)	4-47
Introduction to the exercise case study (2 of 2)	4-48
Case study bot development exercises overview (1 of 3)	4-49

Case study bot development exercises overview (2 of 3)	4-50
Case study bot development exercises overview (3 of 3)	4-51
Command matrix: Commands covered	4-52
Command matrix: Commands not covered	4-53
Unit summary	4-54
Review questions	4-55
Review answers	4-56
Exercise 3: Writing data from a text file to an Excel spreadsheet	4-57
Exercise objectives	4-58
Exercise introduction	4-59
Unit 5. Working with the Smart Recorder and the Object Cloning command	5-1
Unit objectives	5-2
Topics	5-3
5.1. Working with the Smart Recorder.	5-4
Working with the Smart Recorder	5-5
Smart Recorder overview	5-6
Starting the Smart Recorder	5-7
Recording actions (1 of 2)	5-8
Recording actions (2 of 2)	5-9
Working with recorded actions (1 of 2)	5-10
Working with recorded actions (2 of 2)	5-11
5.2. Working with Object Cloning commands	5-12
Working with Object Cloning commands	5-13
Object Cloning command overview	5-14
Using the Object Cloning command window	5-15
Object Cloning play mode options	5-16
Defining the Object Cloning actions	5-17
Editing an Object Cloning command	5-18
5.3. Exercise command focus: Files/Folders commands	5-19
Exercise command focus: Files/Folders commands	5-20
Files/Folders commands	5-21
Files/Folders command: Advanced View	5-22
Files subcommands (1 of 3)	5-23
Files subcommands (2 of 3)	5-24
Files subcommands (3 of 3)	5-25
Folders subcommands (1 of 2)	5-26
Folders subcommands (2 of 2)	5-27
5.4. Exercise command focus: Database commands	5-28
Exercise command focus: Database commands	5-29
Database commands	5-30
Database subcommands (1 of 3)	5-31
Database subcommands (2 of 3)	5-32
Database subcommands (3 of 3)	5-33
Unit summary	5-34
Review questions	5-35
Review answers	5-36
Exercise 4: Automating data entry to a business application and a database	5-37
Exercise objectives	5-38
Exercise introduction (1 of 2)	5-39
Exercise introduction (2 of 2)	5-40
Unit 6. Working with emails, conditional logic, and triggers	6-1
Unit objectives	6-2
Topics	6-3
6.1. Sending emails	6-4

Sending emails	6-5
Sending emails (1 of 2)	6-6
Sending emails (2 of 2)	6-7
6.2. Implementing conditional logic	6-8
Implementing conditional logic	6-9
Working with conditions (1 of 2)	6-10
Working with conditions (2 of 2)	6-11
If/Else Command (1 of 2)	6-12
If/Else Command (2 of 2)	6-13
6.3. Pasting and copying from the Clipboard	6-14
Pasting and copying from the Clipboard	6-15
Pasting and copying from the Clipboard	6-16
6.4. Extending the Variable Operation command	6-17
Extending the Variable Operation command	6-18
Extending the Variable Operation command (1 of 2)	6-19
Extending the Variable Operation command (2 of 2)	6-20
6.5. Extending the Loop command	6-21
Extending the Loop command	6-22
Extending the Loop command with nested looping	6-23
6.6. Introducing bot triggers	6-24
Introducing bot triggers	6-25
Introduction to bot triggers (1 of 4)	6-26
Introduction to bot triggers (2 of 4)	6-27
Introduction to bot triggers (3 of 4)	6-28
Introduction to bot triggers (4 of 4)	6-29
6.7. Introducing debugging	6-30
Introducing debugging	6-31
Debugging (1 of 2)	6-32
Debugging (2 of 2)	6-33
Unit summary	6-34
Review questions (1 of 2)	6-35
Review questions (2 of 2)	6-36
Review answers (1 of 2)	6-37
Review answers (2 of 2)	6-38
Exercise 5: Creating a bot to sum check declines, query a database, and send an email	6-39
Exercise objectives	6-40
Exercise introduction (1 of 7)	6-41
Exercise introduction (2 of 7)	6-42
Exercise introduction (3 of 7)	6-43
Exercise introduction (4 of 7)	6-44
Exercise introduction (5 of 7)	6-45
Exercise introduction (6 of 7)	6-46
Exercise introduction (7 of 7)	6-47
Unit 7. Working with email and PDFs	7-1
Unit objectives	7-2
Topics	7-3
7.1. Downloading attachments from email	7-4
Downloading attachments from email	7-5
Email Automation command (1 of 3)	7-6
Email Automation command (2 of 3)	7-7
Email Automation command (3 of 3)	7-8
7.2. Extracting data from a PDF document	7-9
Extracting data from a PDF document	7-10
PDF Integration command	7-11
PDF Integration command: Extract Text	7-12

PDF Integration command – Extract form fields (1 of 3)	7-13
PDF Integration command – Extract form fields (2 of 3)	7-14
PDF Integration command – Extract form fields (3 of 3)	7-15
PDF Integration command – PDF Viewer (1 of 5)	7-16
PDF Integration command – PDF Viewer (2 of 5)	7-17
PDF Integration command – PDF Viewer (3 of 5)	7-18
PDF Integration command – PDF Viewer (4 of 5)	7-19
PDF Integration command – PDF Viewer (5 of 5)	7-20
PDF Integration command: Caveats (1 of 2)	7-21
PDF Integration command: Caveats (2 of 2)	7-22
7.3. Extending the String Operation command to trim variables	7-23
Extending the String Operation command to trim variables	7-24
Extending the String Operation command to trim variables	7-25
7.4. Extending the String Operation command to get length of variables	7-26
Extending the String Operation command to get length of variables	7-27
Extending the String Operation command to get length of variables	7-28
Unit summary	7-29
Review questions	7-30
Review answers	7-31
Exercise 6: Creating a bot to evaluate data from a PDF and send an email	7-32
Exercise objectives	7-33
Exercise introduction (1 of 7)	7-34
Exercise introduction (2 of 7)	7-35
Exercise introduction (3 of 7)	7-36
Exercise introduction (4 of 7)	7-37
Exercise introduction (5 of 7)	7-38
Exercise introduction (6 of 7)	7-39
Exercise introduction (7 of 7)	7-40
Unit 8. Creating interactive bots	8-1
Unit objectives	8-2
Topics	8-3
8.1. Attended versus unattended bots	8-4
Attended versus unattended bots	8-5
Attended bots (1 of 2)	8-6
Attended bots (2 of 2)	8-7
Comparing attended and unattended bots	8-8
Examples of attended bot implementations (1 of 3)	8-9
Examples of attended bot implementations (2 of 3)	8-10
Examples of attended bot implementations (3 of 3)	8-11
8.1. Using interactive elements	8-12
Using interactive elements	8-13
Using interactive elements: Prompt	8-14
Using interactive elements: Message	8-15
8.2. Using the Web Recorder command to extract a table from a web page	8-16
Using the Web Recorder command to extract a table from a web page	8-17
Using the Web Recorder command to extract a table from a web page (1 of 7)	8-18
Using the Web Recorder command to extract a table from a web page (2 of 7)	8-19
Using the Web Recorder command to extract a table from a web page (3 of 7)	8-20
Using the Web Recorder command to extract a table from a web page (4 of 7)	8-21
Using the Web Recorder command to extract a table from a web page (5 of 7)	8-22
Using the Web Recorder command to extract a table from a web page (6 of 7)	8-23
Using the Web Recorder command to extract a table from a web page (7 of 7)	8-24
8.3. Extending the Excel command to find a value in a spreadsheet	8-25
Extending the Excel command to find a value in a spreadsheet	8-26
Extending the Excel command to find a value in a spreadsheet	8-27

8.4. Introduction to bot hardening	8-28
Introduction to bot hardening	8-29
Introduction to bot hardening	8-30
Unit summary	8-31
Review questions	8-32
Review answers	8-33
Exercise 7: Creating an interactive bot to check values in disparate systems	8-34
Exercise objectives	8-35
Exercise introduction (1 of 7)	8-36
Exercise introduction (2 of 7)	8-37
Exercise introduction (3 of 7)	8-38
Exercise introduction (4 of 7)	8-39
Exercise introduction (5 of 7)	8-40
Exercise introduction (6 of 7)	8-41
Exercise introduction (7 of 7)	8-42
Unit 9. Introducing MetaBots	9-1
Unit objectives	9-2
Topics	9-3
9.1. Overview of MetaBots	9-4
Overview of MetaBots	9-5
Overview of MetaBots	9-6
9.2. Benefits of MetaBots	9-7
Benefits of MetaBots	9-8
Benefits of MetaBots (1 of 2)	9-9
Benefits of MetaBots (2 of 2)	9-10
9.3. Introduction to the MetaBot Designer	9-11
Introduction to the MetaBot Designer	9-12
Introduction to the MetaBot Designer	9-13
Introduction to the MetaBot Designer: Assets tab	9-14
Introduction to the MetaBot Designer – Logic tab	9-15
Creating a MetaBot (1 of 5)	9-16
Creating a MetaBot (2 of 5)	9-17
Creating a MetaBot (3 of 5)	9-18
Creating a MetaBot (4 of 5)	9-19
Creating a MetaBot (5 of 5)	9-20
Passing parameters to a MetaBot (1 of 2)	9-21
Passing parameters to a MetaBot (2 of 2)	9-22
AI Sense	9-23
9.4. Accessing credentials in a MetaBot	9-24
Accessing credentials in a MetaBot	9-25
Accessing credentials in a MetaBot (1 of 2)	9-26
Accessing credentials in a MetaBot (2 of 2)	9-27
9.5. Running a MetaBot from a task bot	9-28
Running a MetaBot from a task bot	9-29
Running a MetaBot from a task bot (1 of 2)	9-30
Running a MetaBot from a task bot (2 of 2)	9-31
Unit summary	9-32
Review questions	9-33
Review answers	9-34
Exercise 8: Creating a login MetaBot	9-35
Exercise objectives	9-36
Unit 10. Working with web services	10-1
Unit objectives	10-2
Topics	10-3

10.1. SOAP Web Service command	10-4
SOAP Web Service command	10-5
SOAP Web Service command	10-6
10.2. REST Web Service command	10-7
REST Web Service command	10-8
REST Web Service command (1 of 3)	10-9
REST Web Service command (2 of 3)	10-10
REST Web Service command (3 of 3)	10-11
Unit summary	10-12
Review questions	10-13
Review answers	10-14
Unit 11. Exceptions and error handling	11-1
Unit objectives	11-2
Topics	11-3
11.1. Error handling concepts	11-4
Error handling concepts	11-5
Error handling concepts	11-6
11.2. Error Handling command	11-7
Error Handling command	11-8
Error Handling command	11-9
11.3. Error View	11-10
Error View	11-11
Error View	11-12
11.4. Evaluate error codes that are received in a web service call	11-13
Evaluate error codes that are received in a web service call	11-14
Evaluate error codes that are received in a web service call	11-15
11.5. Display a variable as currency	11-16
Display a variable as currency	11-17
How to display a variable as currency (1 of 2)	11-18
How to display a variable as currency (2 of 2)	11-19
Unit summary	11-20
Review questions	11-21
Review answers	11-22
Exercise 9: Working with web services and error handling	11-23
Exercise objectives	11-24
Exercise introduction (1 of 2)	11-25
Exercise introduction (2 of 2)	11-26
Unit 12. Hardening bots against exceptions	12-1
Unit objectives	12-2
Topics	12-3
12.1. Bot hardening concepts	12-4
Bot hardening concepts	12-5
Bot hardening concepts	12-6
12.2. Bot hardening scenarios and suggested practices	12-7
Bot hardening scenarios and suggested practices	12-8
Bot hardening example scenarios (1 of 7)	12-9
Bot hardening example scenarios (2 of 7)	12-10
Bot hardening example scenarios (3 of 7)	12-11
Bot hardening example scenarios (4 of 7)	12-12
Bot hardening example scenarios (5 of 7)	12-13
Bot hardening example scenarios (6 of 7)	12-14
Bot hardening example scenarios (7 of 7)	12-15
Running a bot from another bot	12-16
Good practices for running a bot from another bot	12-17

Run Task command	12-18
Challenge: Create an Error Handler bot (1 of 4)	12-19
Challenge: Create an Error Handler bot (2 of 4)	12-20
Challenge: Create an Error Handler bot (3 of 4)	12-21
Challenge: Create an Error Handler bot (4 of 4)	12-22
Unit summary	12-23
Review questions	12-24
Review answers	12-25
Exercise 10: Hardening the Account Opening bot	12-26
Exercise objectives	12-27
Exercise introduction	12-28
Unit 13. Bot management and reporting	13-1
Unit objectives	13-2
Topics	13-3
13.1. Deploying bot dependencies	13-4
Deploying bot dependencies	13-5
Deploying bot dependencies (1 of 2)	13-6
Deploying bot dependencies (2 of 2)	13-7
Performing a code comparison	13-8
13.2. Bot management tools	13-9
Bot management tools	13-10
Bot management tools	13-11
Bot management tools: Repository (1 of 2)	13-12
Bot management tools: Repository (2 of 2)	13-13
Bot management tools: Schedule Manager (1 of 4)	13-14
Bot management tools: Schedule Manager (2 of 4)	13-15
Bot management tools: Schedule Manager (3 of 4)	13-16
Bot management tools: Schedule Manager (4 of 4)	13-17
Bot management tools: Trigger Manager (1 of 4)	13-18
Bot management tools: Trigger Manager (2 of 4)	13-19
Bot management tools: Trigger Manager (3 of 4)	13-20
Bot management tools: Trigger Manager (4 of 4)	13-21
13.3. Bot performance monitoring tools	13-22
Bot performance monitoring tools	13-23
Bot performance monitoring tools	13-24
Bot performance monitoring tools: Notifications (1 of 3)	13-25
Bot performance monitoring tools: Notifications (2 of 3)	13-26
Bot performance monitoring tools: Notifications (3 of 3)	13-27
Bot performance monitoring tools: System logs (1 of 4)	13-28
Bot performance monitoring tools: System logs (2 of 4)	13-29
Bot performance monitoring tools: System logs (3 of 4)	13-30
Bot performance monitoring tools: System logs (4 of 4)	13-31
Bot performance monitoring tools: ROI Calculator	13-32
Bot performance monitoring tools: Report Designer (1 of 4)	13-33
Bot performance monitoring tools: Report Designer (2 of 4)	13-34
Bot performance monitoring tools: Report Designer (3 of 4)	13-35
Bot performance monitoring tools: Report Designer (4 of 4)	13-36
Unit summary	13-37
Review questions	13-38
Review answers	13-39
Exercise 11: Managing bots	13-40
Exercise objectives	13-41
Exercise introduction	13-42

Unit 14. Administering bots through the Control Room.....	14-1
Unit objectives	14-2
Topics	14-3
14.1. Workload management.....	14-4
Workload management	14-5
Queues and device pools	14-6
Create a device pool	14-7
Create a queue	14-8
Queue-based workload management	14-9
Managing work items	14-10
Queue orchestration	14-11
Using work queues	14-12
14.2. Running and scheduling bots from the Control Room	14-13
Running and scheduling bots from the Control Room	14-14
Run bot now (1 of 5)	14-15
Run bot now (2 of 5)	14-16
Run bot now (3 of 5)	14-17
Run bot now (4 of 5)	14-18
Run bot now (5 of 5)	14-19
Schedule a bot for execution (1 of 5)	14-20
Schedule a bot for execution (2 of 5)	14-21
Schedule a bot for execution (3 of 5)	14-22
Schedule a bot for execution (4 of 5)	14-23
Schedule a bot for execution (5 of 5)	14-24
Run bot with queue (1 of 5)	14-25
Run bot with queue (2 of 5)	14-26
Run bot with queue (3 of 5)	14-27
Run bot with queue (4 of 5)	14-28
Run bot with queue (5 of 5)	14-29
14.3. Bot insights overview	14-30
Bot insights overview	14-31
Bot insights overview (1 of 4)	14-32
Bot insights overview (2 of 4)	14-33
Bot insights overview (3 of 4)	14-34
Bot insights overview (4 of 4)	14-35
14.4. Bot lifecycle management.....	14-36
Bot lifecycle management	14-37
Bot lifecycle management overview (1 of 2)	14-38
Bot lifecycle management overview (2 of 2)	14-39
Bot lifecycle management workflow	14-40
Exporting bot deployment package	14-41
Importing bot deployment package	14-42
Unit summary	14-43
Review questions	14-44
Review answers	14-45
Exercise 12: Administering bots in the Control Room	14-46
Exercise objectives	14-47
Unit 15. Course summary	15-1
Unit objectives	15-2
Course objectives	15-3
Command matrix: Commands covered	15-4
Command matrix: Commands not covered	15-5
To learn more on the subject	15-6
Additional resources	15-7
Earn an IBM Badge	15-8

Enhance your learning with IBM resources	15-9
Unit summary	15-10
Course completion	15-11

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

DB™
Insight®
400®

FileNet®
Notes®

Global Business Services®
Tivoli®

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

VMware is a registered trademark or trademark of VMware, Inc. or its subsidiaries in the United States and/or other jurisdictions.

Other product and service names might be trademarks of IBM or other companies.

Course description

Automating Tasks Using IBM Robotic Process Automation with Automation Anywhere

Duration: 5 days

Purpose

This course is intended to teach the skills that are needed to work with the IBM Robotic Process Automation with Automation Anywhere Enterprise Client and Control Room to develop and manage bots.

Audience

This course is intended for developers who use IBM Robotic Process Automation with Automation Anywhere.

Prerequisites

- Practical knowledge of data structures
- Understanding of SQL syntax
- Basic understanding of web services
- Experience with modern programming techniques

Objectives

- Describe the IBM Robotic Process Automation with Automation Anywhere Control Room and Enterprise Client
- Work with the various Enterprise Client Recorders and commands to create and manage bots

Agenda



Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

- (00:15) Course introduction
- (00:30) Unit 1. Introducing robotic process automation and bots
- (01:00) Unit 2. Introducing IBM Robotic Process Automation with Automation Anywhere
- (00:30) Exercise 1. Exploring the Control Room
- (01:00) Unit 3. Introducing the Enterprise Client
- (00:45) Exercise 2. Creating a basic bot
- (01:00) Unit 4. Working with variables, loops, and strings
- (01:45) Exercise 3. Writing data from a text file to an Excel spreadsheet

Day 2

- (01:00) Unit 5. Working with the Smart Recorder and the Object Cloning command
- (02:00) Exercise 4. Automating data entry to a business application and a database
- (01:00) Unit 6. Working with emails, conditional logic, and triggers
- (02:00) Exercise 5. Creating a bot to sum check declines, query a database, and send an email
- (01:00) Unit 7. Working with email and PDFs

Day 3

- (02:00) Exercise 6. Creating a bot to evaluate data from a PDF and send an email
- (01:00) Unit 8. Creating interactive bots
- (01:30) Exercise 7. Creating an interactive bot to check values in disparate systems
- (01:00) Unit 9. Introducing MetaBots
- (00:30) Exercise 8. Creating a login MetaBot
- (01:00) Unit 10. Working with web services

Day 4

- (01:00) Unit 11. Exceptions and error handling
- (02:30) Exercise 9. Working with web services and error handling
- (01:00) Unit 12. Hardening bots against exceptions
- (02:30) Exercise 10. Hardening the Account Opening bot

Day 5

- (01:00) Unit 13. Bot management and reporting
- (01:00) Exercise 11. Managing bots
- (01:00) Unit 14. Administering bots through the Control Room
- (00:30) Exercise 12. Administering bots in the Control Room
- (00:30) Unit 15. Course summary

Unit 1. Introducing robotic process automation and bots

Estimated time

00:30

Overview

In this unit, you learn about robotic process automation (RPA) and what RPA bots can do.

How you will check your progress

- Checkpoint

Unit objectives

- Describe robotic process automation (RPA)
- Describe an RPA bot

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-1. Unit objectives

Topics

- What is Robotic Process Automation (RPA)?
- Use cases and industries
- Introducing IBM Robotic Process Automation with Automation Anywhere
- Robotic Process Automation and Business Automation Workflow

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-2. Topics

1.1. What is Robotic Process Automation (RPA)?

What is Robotic Process Automation (RPA)?

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-1. What is Robotic Process Automation (RPA)?

What is Robotic Process Automation (RPA)?

- Automation of a wide range of administrative tasks
- Uses specific software algorithms that interact with multiple applications and computer-centric processes
- Runs transactional processes at the user interface (UI) level



Software robots (bots)

- Software that provides a “**Virtualized full-time equivalent (FTE)**”
- More functionality than a desktop script or macro
- Can:
 - Manipulate, operate, and orchestrate other applications
 - Follow business rules
 - Run transactions
 - Execute program APIs and other program objects



Software that mimics human actions

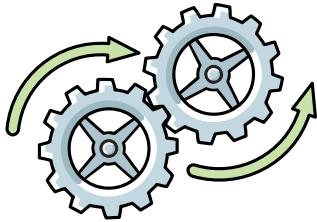
- Taught to “drive” applications the way that a human does through the UI
- “Robot” is programmed to:
 - Follow predetermined computer pathways between the screen and data repositories
 - Move or populate data between locations
 - Conduct calculations
 - Perform actions
 - Trigger downstream activities

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-2. What is Robotic Process Automation (RPA)?

Software robots are similar to physical robots

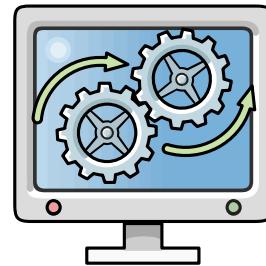


Physical Robots

Perform repetitive **physical** tasks

Example:

- Pick eight chocolates from assembly line
- Assemble a finished box of chocolates



Software Robots

Perform repetitive **software** tasks

Example:

- Log in to four different systems
- Browse the UI of established applications to retrieve data
- Use data to open new account in system of record

Figure 1-3. Software robots are similar to physical robots

Benefits of robotic process automation



Accelerate time-to-value

Create, test, and deliver new automations in days or weeks



Reduce human error

Eliminate copy and paste mistakes that are introduced by swivel chair integration



Increase throughput

Fulfill automated tasks in seconds or minutes, 24x7



Decrease development costs

Develop bots quickly with simple record and playback functions

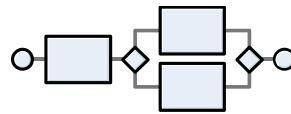
Figure 1-4. Benefits of robotic process automation

Challenges of Robotic Process Automation



Fragility and exception handling

- Bots are designed to work in a specific way
- Bots are not designed for change or agility
- 30% of bots need exception handling



Requires process analysis

- Must understand processes and select the right activities to automate
- Bots do not have human task, case, or complex rule capabilities
- RPA is not always the correct way to address a process automation problem



Requires business and IT sponsorship

- Business and IT must work together
- Governance of new business and IT changes is required
- Must have a Center of Excellence (CoE) for all automation technologies, not just RPA

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-5. Challenges of Robotic Process Automation

1.2. Use cases and industries

Use cases and industries

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-1. Use cases and industries

Common use cases (1 of 2)



Multiple data sources

- Data transfer between systems
- Matching data between systems



Swivel chair

- No back-end integration available
- Integration is possible only through the UI



System migration

- Integration with established or mature systems
- Migration of data between systems with no integration

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-2. Common use cases (1 of 2)

Common use cases (2 of 2)



Unstructured inputs

- Extract data from documents such as email, faxes, and paper forms
- Convert unstructured text to structured data



Pre-checking

- Automate high volume of checks
- Filter out simple cases
- Refer to users for exceptions



New systems

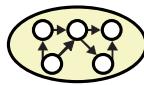
- New systems with missing functionality
- Impossible or too expensive to customize
- Workarounds needed

Figure 1-3. Common use cases (2 of 2)

Good uses for Robotic Process Automation



High volume



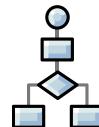
Repetitive tasks



Manual data entry



Multiple established systems



Logic and rules-based processes



High number of full-time equivalent (FTE) employees

The ideal use for RPA is when an organization has a **large-scale use of human labor** for general knowledge process work, where people are performing **high-volume, highly transactional** process functions through **highly repetitive tasks**.

Industries that use Robotic Process Automation (1 of 2)



Insurance

- Claims processing
- Updating user data
- Audit management
- Enrollment and eligibility
- Data integration
- Billing and compliance
- Data Transfer Claims
- Processing Eligibility
- Data Verification
- Provider Updates
- Reconciliation



Banking

- Verification and auditing
- Bank reconciliation
- Compliance processing
- Account management
- Data migration
- Audit trail
- Statements and reporting
- Loan processing
- Payment cancellations
- Credit decisioning



Financial

- Order to cash
- Credit processing
- Customer master data management
- Order entry
- Procure to pay
- Payment process
- Duplicate payment tracking
- Claim processing
- Financial analysis
- Record to report
- Financial statements
- Tax services

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-5. Industries that use Robotic Process Automation (1 of 2)

Industries that use Robotic Process Automation (2 of 2)



Human resources

- Offer letters
- Onboarding
- Appraisal process
- Payroll status
- Position management
- Reporting line changes
- Payment summaries
- Employment type updates
- Service desk reports
- Leave amendments



Network and IT

- FTP management
- Automated installations
- Monitoring and alert management
- Service desk management
- Notification and escalation
- Data movement
- Provisioning
- Configuration management
- Routine maintenance



Supply chain

- Order Prioritization
- Invoice verification
- Receipt confirmation
- Scheduling processes
- Production information capture
- Inbound processing
- Inventory management
- Billing
- Freight costing
- Reporting

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-6. Industries that use Robotic Process Automation (2 of 2)

1.3. Introducing IBM Robotic Process Automation with Automation Anywhere

Introducing IBM Robotic Process Automation with Automation Anywhere

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-1. Introducing IBM Robotic Process Automation with Automation Anywhere

IBM partnership with Automation Anywhere



- IBM and Automation Anywhere joined forces to create a new RPA offering within the IBM Digital Business Automation portfolio

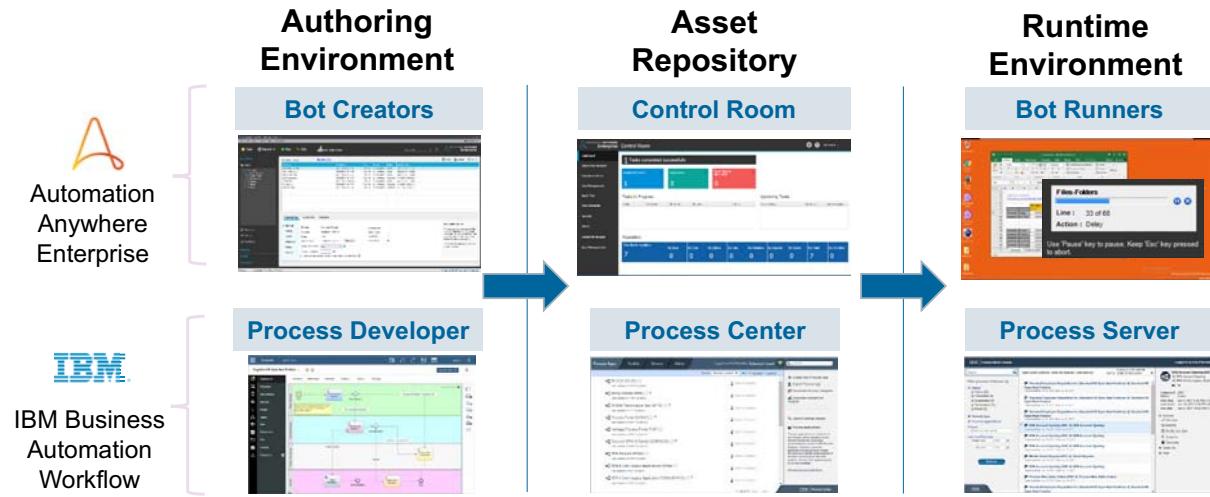
Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-2. IBM partnership with Automation Anywhere

IBM Training

IBM Robotic Process Automation with Automation Anywhere architecture



Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-3. IBM Robotic Process Automation with Automation Anywhere architecture

1.4. Robotic Process Automation and Business Automation Workflow

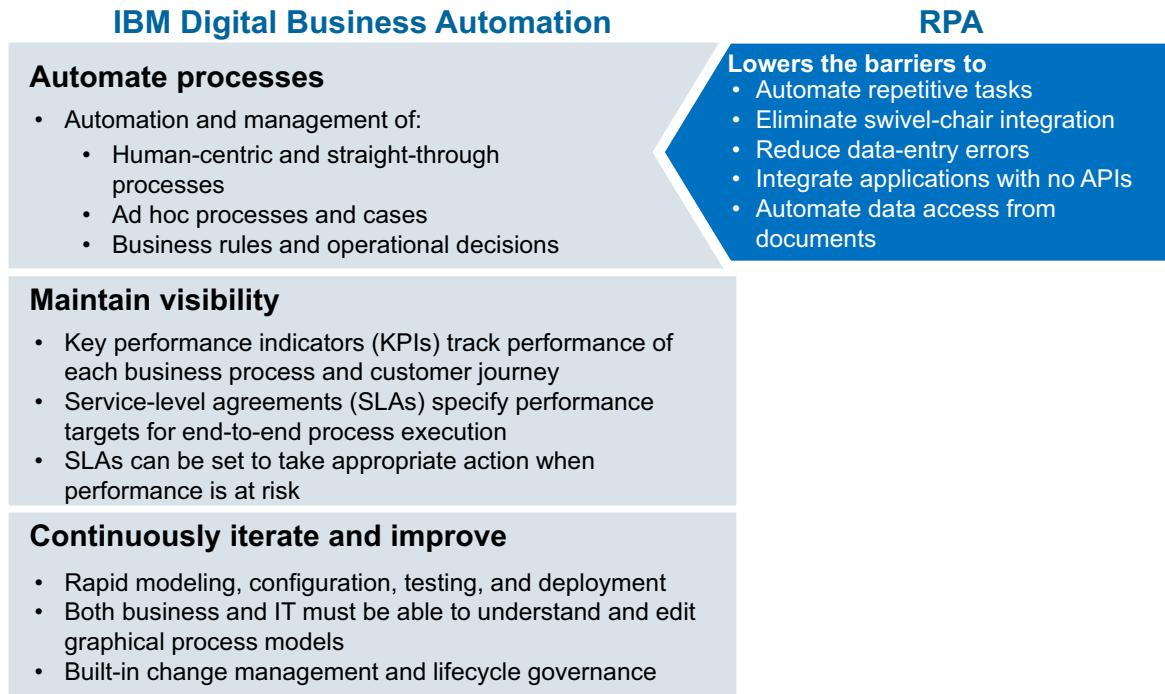
Robotic Process Automation and Business Automation Workflow

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-1. Robotic Process Automation and Business Automation Workflow

RPA and the IBM Digital Business Automation platform



Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-2. RPA and the IBM Digital Business Automation platform

IBM Training



Comparing IBM RPA with Automation Anywhere and IBM Business Automation Workflow

	 IBM RPA with AA	 IBM BAW
Goal	<ul style="list-style-type: none"> • Human task automation • Mimic human interaction with systems 	<ul style="list-style-type: none"> • Process optimization • Straight-through processing (STP) when possible
Scope	<ul style="list-style-type: none"> • Usually a tactical solution • Address specific process problems • Can be used strategically when RPA is the only way to integrate with a stable backend system 	<ul style="list-style-type: none"> • Recognizes that not all process can or should be automated
Value Proposition	<ul style="list-style-type: none"> • Provide tactical fix to a process problem • No need to change business operations • Increase throughput • Reduce workforce • Free human resources for knowledge work 	<ul style="list-style-type: none"> • Generally more strategic adoption to maximize value and ROI • Enterprise-wide • Use for long-running, stateful processes

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-3. Comparing IBM RPA with Automation Anywhere and IBM Business Automation Workflow

Unit summary

- Describe robotic process automation (RPA)
- Describe an RPA bot

Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-4. Unit summary

Review questions



1. The following characteristics are a benefit of RPA. Select all that apply.
 - A. Reduction in human error
 - B. Increase of throughput
 - C. Decrease in development costs
 - D. RPA bots never break

2. Which of the following examples is not a common use case for RPA?
 - A. Pre-checking
 - B. Multiple data sources
 - C. Process optimization
 - D. Swivel chair

Review answers

1. The following characteristics are a benefit of RPA.
Select all that apply.
 - A. Reduction in human error
This answer is true.
 - B. Increase of throughput
This answer is true.
 - C. Decrease in development costs
This answer is true.
 - D. Robots never break
This answer is false. Bots are not coded for agility or change.

2. Which of the following examples is not a common use case for RPA?
 - A. Pre-checking
 - B. Multiple data sources
 - C. Process optimization
 - D. Swivel chair

The answer is C.



Introducing robotic process automation and bots

© Copyright IBM Corporation 2018, 2019

Figure 1-6. Review answers

Unit 2. Introducing IBM Robotic Process Automation with Automation Anywhere

Estimated time

01:00

Overview

In this unit, you learn about IBM Robotic Process Automation with Automation Anywhere with an introduction to the Control Room.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe the main components and features of IBM Robotic Process Automation with Automation Anywhere
- Explain the client/server architecture model that IBM Robotic Process Automation with Automation Anywhere uses
- Describe the Control Room and its features

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-1. Unit objectives

Topics

- IBM Robotic Process Automation with Automation Anywhere product overview
- Introducing the Control Room
- User management

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-2. Topics

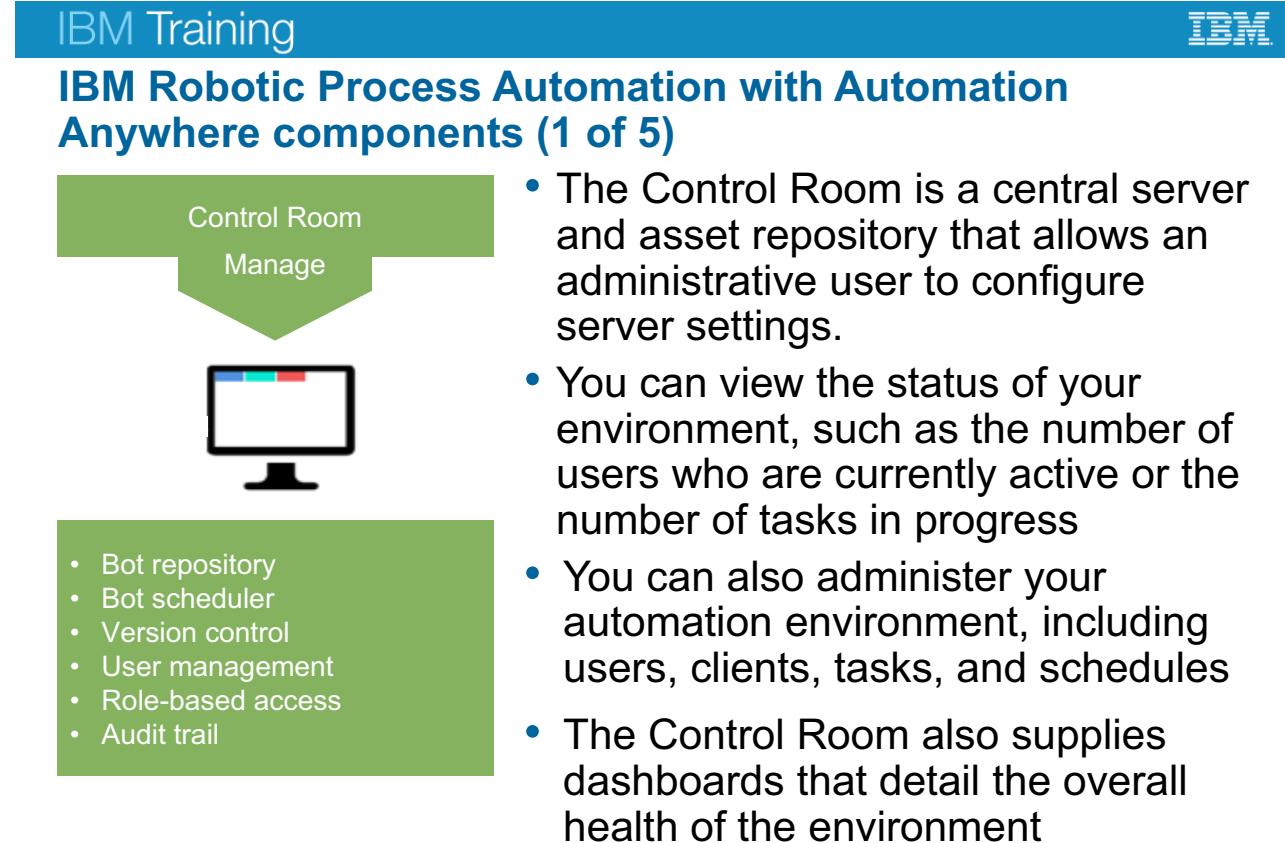
2.1. IBM Robotic Process Automation with Automation Anywhere product overview

IBM Robotic Process Automation with Automation Anywhere product overview

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-3. IBM Robotic Process Automation with Automation Anywhere product overview



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-4. IBM Robotic Process Automation with Automation Anywhere components (1 of 5)

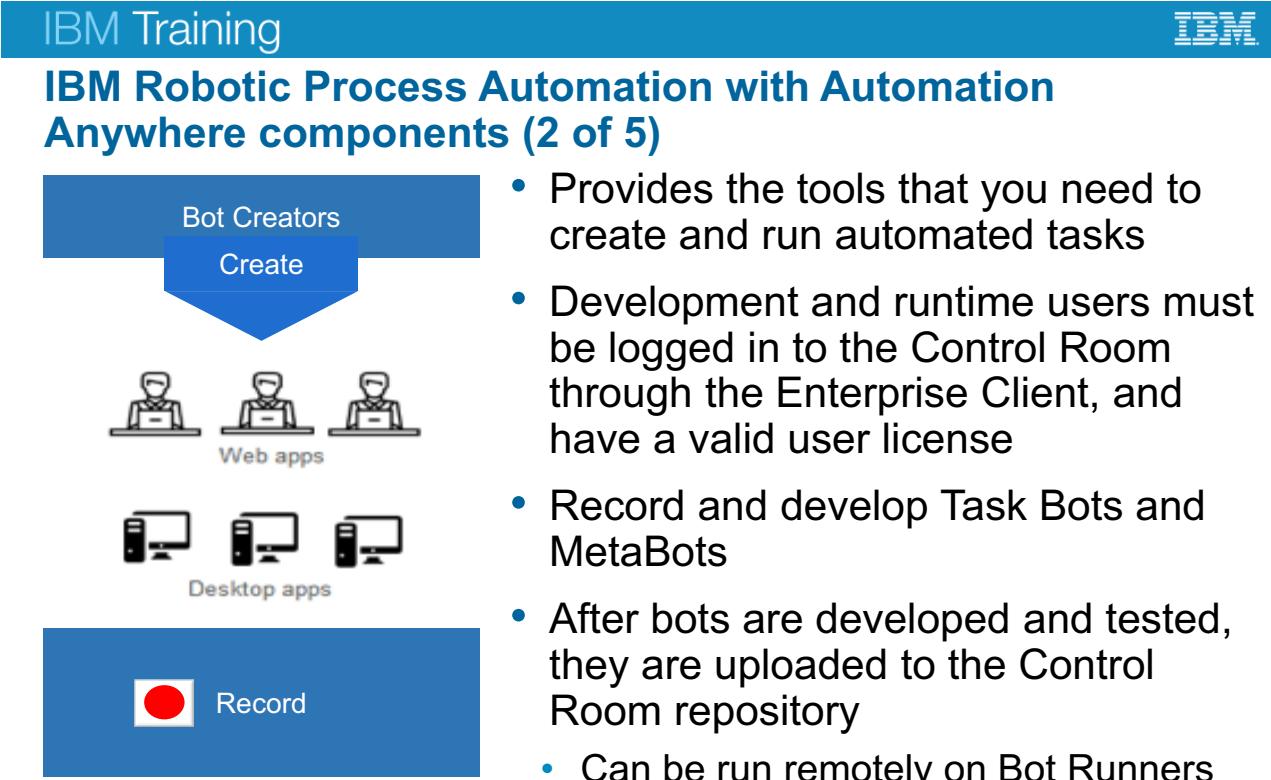


Figure 2-5. IBM Robotic Process Automation with Automation Anywhere components (2 of 5)

IBM Training



IBM Robotic Process Automation with Automation Anywhere components (3 of 5)



Unattended Bots:

- Unattended bots run tasks and interact with applications independent of human involvement
- Unattended bots are either scheduled or triggered by an event
- Both unattended and attended bots can be deployed within the same environment
- The combination of attended and unattended RPA provides for a “complete” RPA solution

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-6. IBM Robotic Process Automation with Automation Anywhere components (3 of 5)

IBM Training



IBM Robotic Process Automation with Automation Anywhere components (4 of 5)



Attended Bots:

- Must run on the user's workstation using the user's Windows credentials to perform work on their behalf
- Can either be started by a user or triggered by an event that occurs on the users workstation
- Makes individuals more productive
- Simpler to develop, quicker to implement
- Faster and less costly to deploy
- Less costly to operate
- Less disruptive to existing work

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

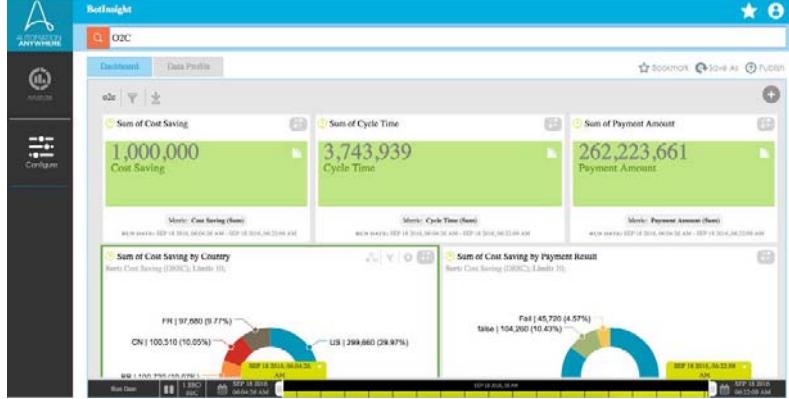
Figure 2-7. IBM Robotic Process Automation with Automation Anywhere components (4 of 5)

IBM Training

IBM Robotic Process Automation with Automation Anywhere components (5 of 5)

Bot Insights

- Bot Insight is now available for IBM RPA Enterprise deployment with IBM RPA V11.0.0.3 release and beyond. Bot Insight is not available with IBM RPA Express.



The screenshot shows the Bot Insight dashboard with three main KPIs: Sum of Cost Saving (1,000,000), Sum of Cycle Time (3,743,939), and Sum of Payment Amount (262,223,661). Below these are two donut charts showing the distribution of cost saving by payment result (97,680, 100,510, 45,720) and a table of detailed data.

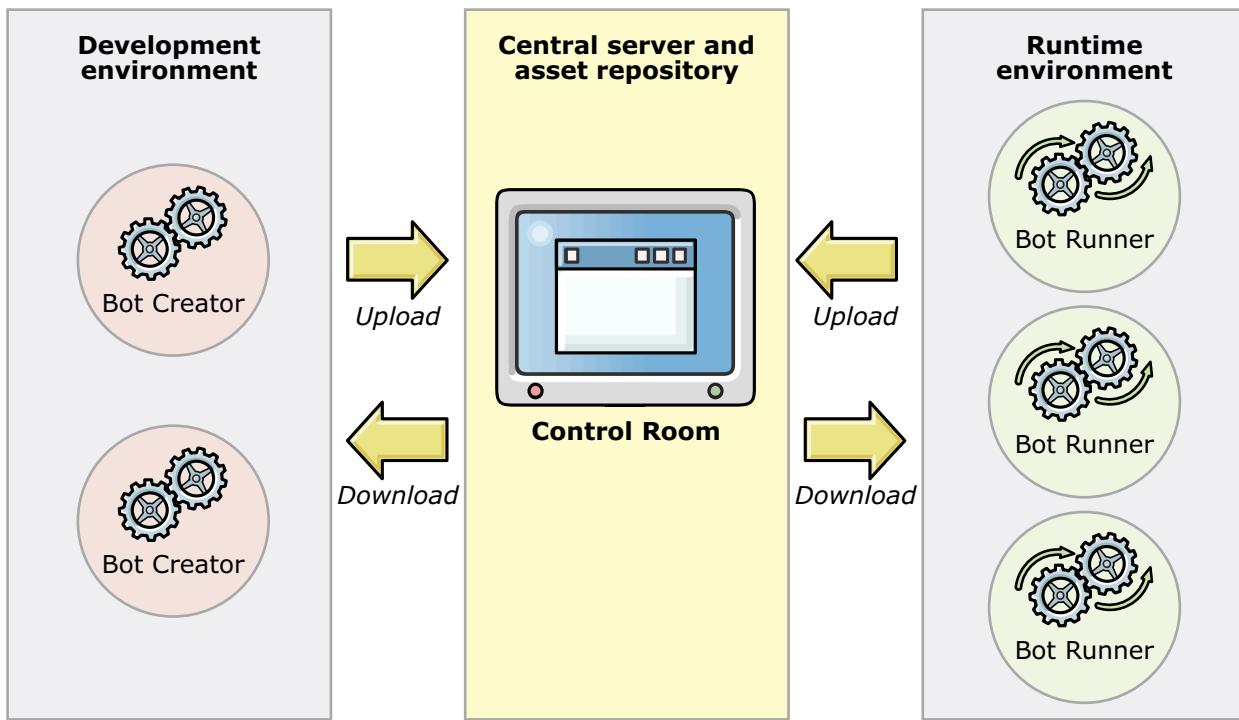
- Bot Insight is activated by default and can be used to report on bot performance without requiring a separate license.
- Customers need to buy appropriate subscriptions for the Bot Insight capability and for Bot Insight users before they are entitled to use variable tagging, Bot Insight APIs, and Bot Insight dashboards for business operations.

[Introducing IBM Robotic Process Automation with Automation Anywhere](#)

© Copyright IBM Corporation 2018, 2019

Figure 2-8. IBM Robotic Process Automation with Automation Anywhere components (5 of 5)

High-level architecture



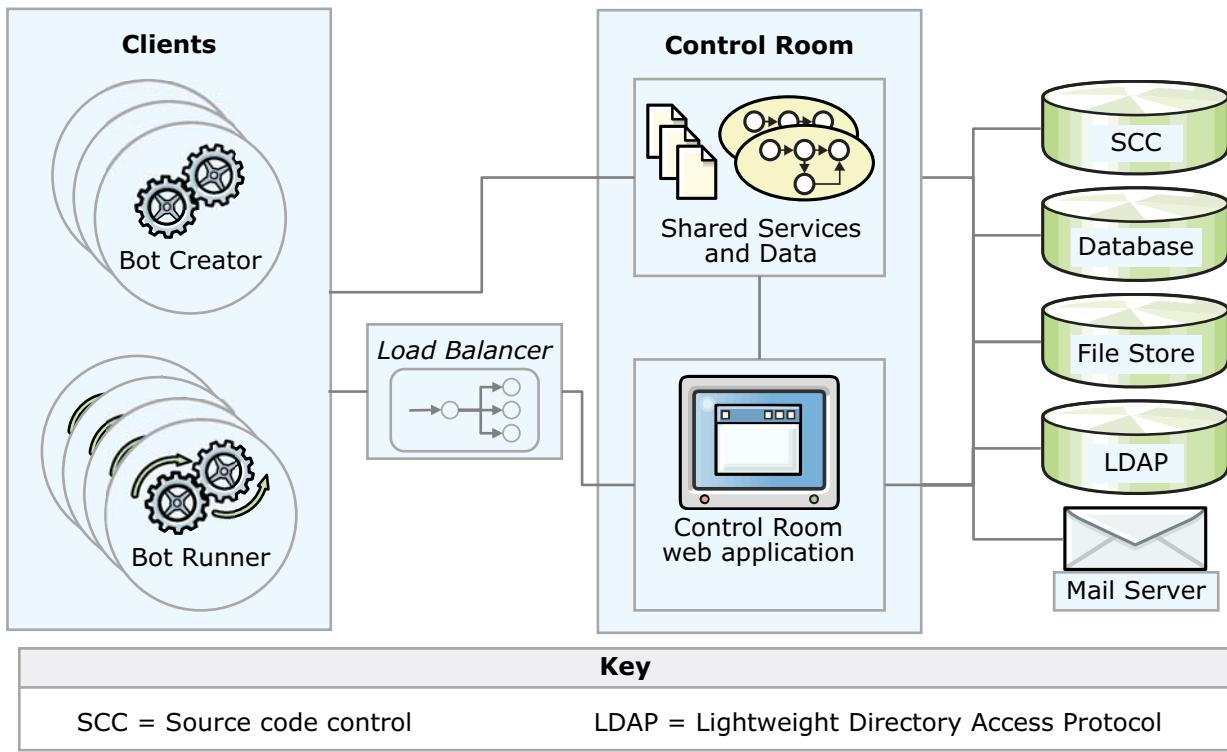
Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-9. High-level architecture



Topology overview



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-10. Topology overview

Access to IBM Robotic Process Automation with Automation Anywhere components and features

- Access to the various features in IBM Robotic Process Automation with Automation Anywhere components depends on user roles and licenses
- Main administrative user has “super user” capabilities
 - Other administrative users have more limited rights
- User roles
 - Assigned by administrators
 - Default user roles are available for specific features
 - Administrative users can create custom user roles
 - A user can have more than one role, depending on what they need to do
- Licenses
 - Assigned by administrators
 - Development license: Grants access to the bot development tools in the Enterprise client
 - Runtime license: Grants access to the runtime tools in the Enterprise client

2.2. Introducing the Control Room

Introducing the Control Room

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-12. Introducing the Control Room

Control Room overview (1 of 2)

The Control Room is a central server and asset repository that allows an administrative user to:

- View Dashboards:
 - View bot performance data
 - View active users, registered clients, bot schedules and more
- Manage bots:
 - Monitor bot activity
 - Upload and download bots
 - Run and schedule bots from a central location
 - Manage credentials for bots from a central location
- Manage Resources
 - Devices and device pools
 - Workload queues
 - Monitor Service Level Agreements
 - Monitor audit trail

Control Room overview (2 of 2)

The Control Room is a central server and asset repository that allows an administrative user to:

- Configure server settings
 - Control room database
 - Bot version control
 - Client configuration
 - Credentials master key
 - Email notifications
- Administer users and security:
 - Create and assign user roles
 - Create users and assign licenses
 - Migrate data from older version

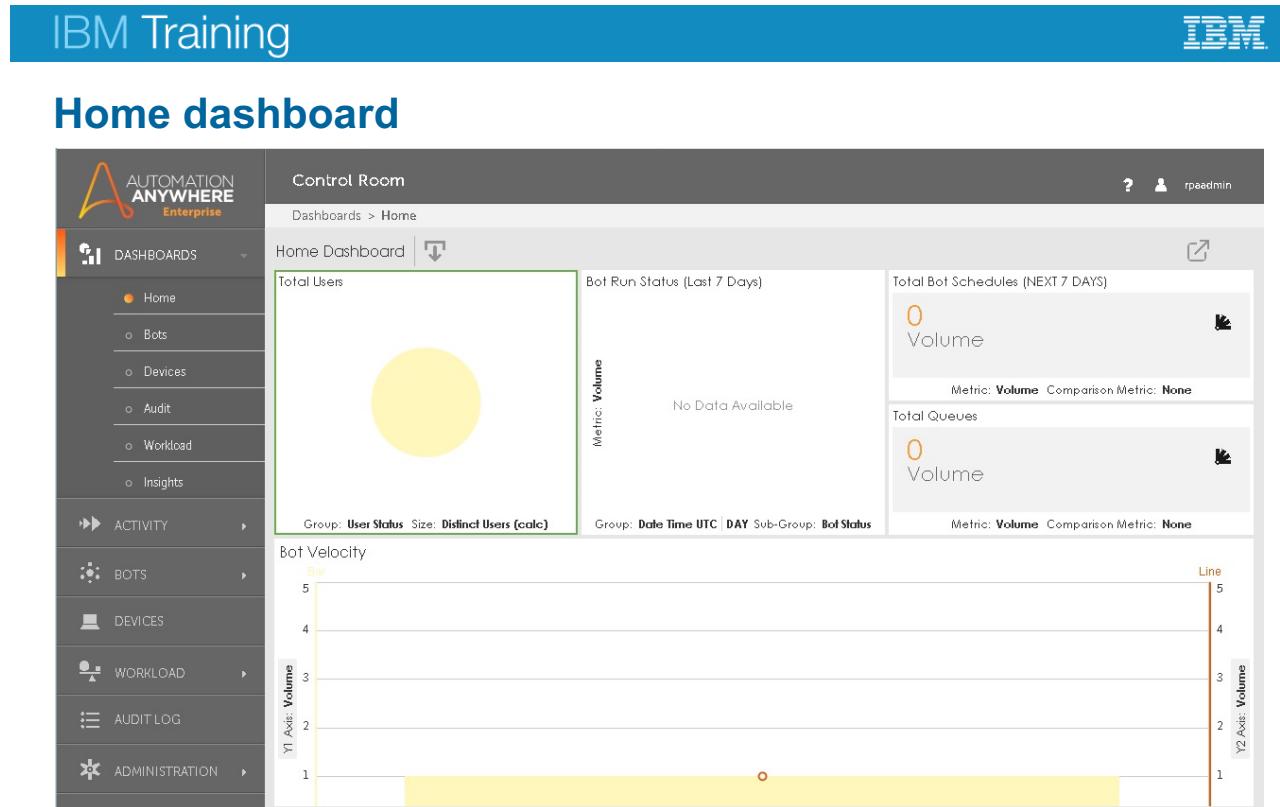
Dashboard features

- Home Dashboard
 - Provides a high-level view that includes bot run status, total users, and more
- Bots Dashboard
 - Provides a high-level view of overall bot status and heartbeat
- Devices Dashboard
 - Provides a high-level view of bot runner machines and their status
- Audit Dashboard
 - Provides a snapshot of audit information along with a visual representation
- Workload Dashboard
 - Provides workload status of device pools, queues, and work items
- Insights Dashboard
 - Additional configuration necessary to run

[Introducing IBM Robotic Process Automation with Automation Anywhere](#)

© Copyright IBM Corporation 2018, 2019

Figure 2-15. Dashboard features



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-16. Home dashboard

IBM Training

Bots dashboard



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-17. Bots dashboard

IBM Training

Audit dashboard



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-18. Audit dashboard

Bot management features (1 of 2)

- Monitor bot activity
 - In progress
 - Scheduled
 - Historical
- Upload and download bots
 - When you upload bots, documents, and other artifacts to the Control Room, they can be viewed under the My bots section.
- Run and schedule bots from a central location
 - You can also import, export, run, and schedule bot executions from the My bots section.
- Manage credentials for bots from a central location
 - **Standard** credentials hold credential values.
 - **User-specific** credentials must be supplied at run time.
 - Credentials must be in a locker to access at run time.
 - Sensitive information is not stored on the client machine. All data is encrypted by using AES-256 and RSA-2048 algorithms and stored on the server.
 - Support for integration with other Access Management by using CyberArk.

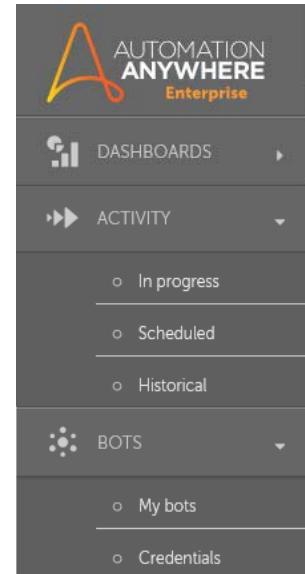


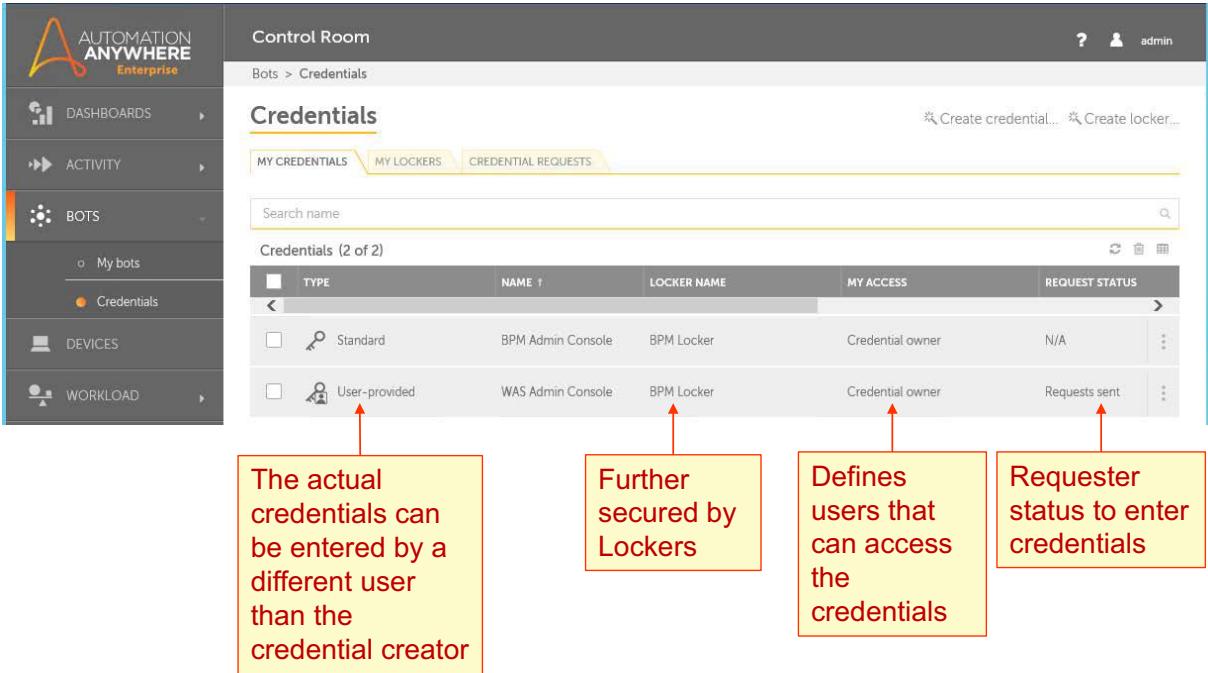
Figure 2-19. Bot management features (1 of 2)

IBM Training



Bot management features (2 of 2)

Credentials section screen capture



The screenshot shows the 'Control Room' interface with the 'Bots > Credentials' path selected. The 'Credentials' tab is active, showing two entries:

Type	Name	Locker Name	My Access	Request Status
Standard	BPM Admin Console	BPM Locker	Credential owner	N/A
User-provided	WAS Admin Console	BPM Locker	Credential owner	Requests sent

Annotations explain the features:

- The actual credentials can be entered by a different user than the credential creator** (points to the first row)
- Further secured by Lockers** (points to the 'Locker Name' column)
- Defines users that can access the credentials** (points to the 'My Access' column)
- Requester status to enter credentials** (points to the 'Request Status' column)

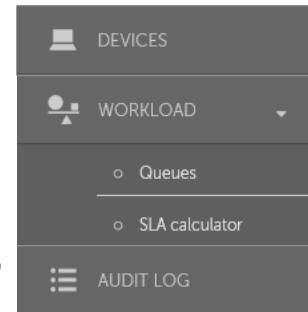
Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-20. Bot management features (2 of 2)

Workload management features (1 of 3)

- Devices
 - Run bots on specific devices
 - View status of bot runners and bot creators
- Device Pools
 - Device pools provide logical groupings of similar Bot Runners with the work item from a queue. For example, you can group devices of a particular department and create a device pool for it.
- Workload Queues
 - You can use queues to distribute work across multiple bot runners.
 - Managing workload by dividing work into small logical units helps to ensure service level agreements (SLAs) are met and resource usage is optimized.
 - Auto-distribution of workload to bot runners achieves best optimization of resources.



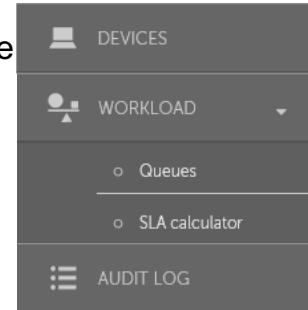
[Introducing IBM Robotic Process Automation with Automation Anywhere](#)

© Copyright IBM Corporation 2018, 2019

Figure 2-21. Workload management features (1 of 3)

Workload management features (2 of 3)

- Monitor Service Level Agreements
 - How long does it take to process items, based on device pool size? Or
 - How many devices will you need to meet a target SLA?
- Monitor audit trail
 - The Audit Log captures and provides read-only records of all the important actions that are performed by users for the Control Room and Client.



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-22. Workload management features (2 of 3)



IBM

Workload management features (3 of 3)

Audit Log section screen capture

All Control Room activities are logged. For example: installing bots, exporting bots, running bots on bot runners, admin settings changes.

Export entries to CSV file

STATUS	TIME	ACTION TYPE	ITEM NAME	ACTION TAKEN BY	SOURCE DEVICE
Unsuccessful	13:07:17 EDT 2018-05-17	Import bots	BLM Package_2018051...	admin	fe80::e189:ba76:844f:
Successful	13:06:15 EDT 2018-05-17	Import bots	BLM Package_2018051...	admin	fe80::e189:ba76:844f:
Successful	13:00:20 EDT 2018-05-17	Export bots	BLM Package_2018051...	admin	fe80::e189:ba76:844f:
Successful	12:53:22 EDT 2018-05-17	User multiple log in	N/A	admin	fe80::e189:ba76:844f:
Unsuccessful	08:44:43 EDT 2018-05-17	Edit User	runner	admin	fe80::e189:ba76:844f:
Unsuccessful	08:43:24 EDT 2018-05-17	Edit User	super	admin	fe80::e189:ba76:844f:
Successful	12:44:19 EDT 2018-05-17	User multiple log in	N/A	admin	fe80::e189:ba76:844f:

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

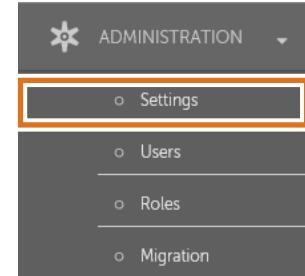
Figure 2-23. Workload management features (3 of 3)



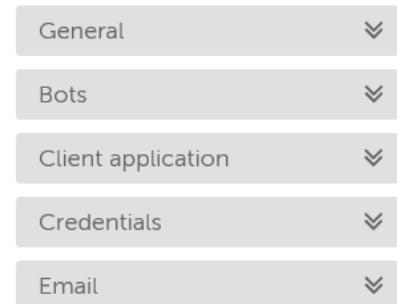
IBM

Server settings

- Control Room database
 - Website security and configuration
 - Control room database and repository configuration
 - Deployment settings and user password requirements
- Bot version control
 - Applicable only if a version control system is installed.
- Client configuration
 - Secure recording setting
 - Product help URLs
 - Device health check configuration
- Credentials master key
 - Credential connection mode
- Email notifications
 - Email notification configuration



Settings



[Introducing IBM Robotic Process Automation with Automation Anywhere](#)

© Copyright IBM Corporation 2018, 2019

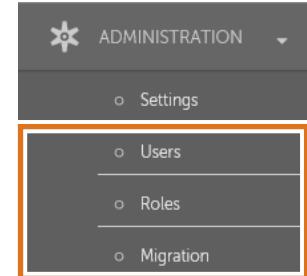
Figure 2-24. Server settings

Administration features (1 of 2)

- Create and assign user roles
 - The Roles section supports the administrative user in creating, copying, modifying, disabling, or deleting roles.

- Create users and assign licenses
 - The Users section supports the administrative user in creating, modifying, disabling, or deleting users.
 - The administrative user can also assign licenses (Bot runner or Bot creator) and roles to users from this screen.

- Migrate data from older version
 - A data migration tool is built in within the Control Room to migrate control room from V10.3 or higher to V11.X.
 - Connect to the source Control Room and migrate data from the source database to the destination Control Room database selecting settings, roles, bots, schedules, and credentials.



[Introducing IBM Robotic Process Automation with Automation Anywhere](#)

© Copyright IBM Corporation 2018, 2019

Figure 2-25. Administration features (1 of 2)



Administration features (2 of 2)

- Roles section screen capture

TYPE	ROLE NAME	# OF USERS	LAST MODIFIED	MODIFIED BY
System-created	AAE_Admin	2	12:17:29 EDT 2018-05-01	System
System-created	AAE_Basic	5	12:17:29 EDT 2018-05-01	System
System-created	AAE_Locker Admin	6	12:17:29 EDT 2018-05-01	System
System-created	AAE_Pool Admin	6	12:17:29 EDT 2018-05-01	System
System-created	AAE_Queue Admin	6	12:17:29 EDT 2018-05-01	System
System-created	AAE_Bot Insight Admin	0	12:17:29 EDT 2018-05-01	System
System-created	AAE_Bot Insight Consumer	0	12:17:29 EDT 2018-05-01	System
System-created	AAE_Bot Insight Expert	0	12:17:29 EDT 2018-05-01	System
User-created	SuperUser	6	17:13:16 EDT 2018-05-04	admin

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-26. Administration features (2 of 2)

2.3. User management

User management

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-27. User management

Creating roles and users

- Roles determine the activities that the assigned users can perform in the Control Room
 - Use the predefined roles in Control Room
 - Create custom roles
 - Assign permissions to the roles, for example, for task scheduling
- After the roles are defined, you can create users, assign them roles and permissions, and allocate licenses to them

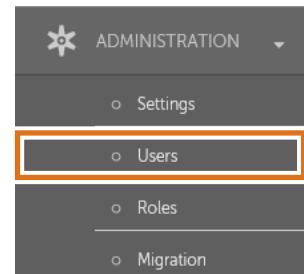
Examples

- Bot Creator role requires a development license
- Bot Runner role requires a runtime license



Creating a user (1 of 4)

- Add new users from the **User** section
- In the Actions menu, click **Create User...**



All users

Username	Search username
botuser1	
devuser1	
rpaadmin	

Users (3 of 3)

User Type	Username	First Name	Last Name	Description	Roles	Auto Login	Last Modified
Bot runner	botuser1	Bot	User1	--	AAE_Basic	Can auto login	16:08:42 EDT 2018-08-15
Bot creator	devuser1	Dev	User1	--	BotCreator	Can auto login	16:02:34 EDT 2018-08-15
Admin	rpaadmin	--	--	--	AAE_Admin	N/A	11:45:06 EDT 2018-08-15

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-29. Creating a user (1 of 4)

IBM Training

Creating a user (2 of 4)

- New users are assigned a predefined role
- You can assign multiple roles to a single user
- There are system-created roles and user-created roles
- You can also allocate a Development or Runtime license

Select roles

Select one or more roles

Search name

Available roles (6 of 8)		Selected (2)	
<input type="checkbox"/>	NAME ↑	<input type="checkbox"/>	NAME ↑
<input type="checkbox"/>	AAE_Admin	<input type="checkbox"/>	AAE_Basic
<input type="checkbox"/>	AAE_IQ Bot Services	<input type="checkbox"/>	BotCreator
<input type="checkbox"/>		<input type="checkbox"/>	
<input type="checkbox"/>		<input type="checkbox"/>	

Allocate a device license to this user?

Device licenses are only applicable if the user does not have the "Admin" or the "BotFarm admin" role.

A device, or Client UI, cannot connect to the Control Room until the user that logs into it has a device license. If you change from a Bot runner license to a Bot creator license, any schedules associated with this username will be deleted.

<input type="radio"/> Bot runner	Requires a Development license, which enables the user to create AND run Task Bots.
<input checked="" type="radio"/> Bot creator	<input checked="" type="checkbox"/> Enable auto login This allows the Client UI to remember the password and automatically log into the Control Room.
<input type="radio"/> None	

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-30. Creating a user (2 of 4)



Creating a user (3 of 4)

- When the Control Room is configured to send email, the Control Room automatically sends a notification to the email address of the newly created user
- The user must confirm the account by clicking a link in the email and then create a password

Email	
Notifications Edit...	
Send email notifications	
From this email address	rpaAdmin@sib.com
Email server host	smtp.sib.com
My server does not use a secure connection (SSL/TLS)	25
My server does not require authentication	
Send an email when	No events chosen
Modified by	Last modified
rpaadmin	12:25:06 EDT 2018-08-15

[Introducing IBM Robotic Process Automation with Automation Anywhere](#)

© Copyright IBM Corporation 2018, 2019

Figure 2-31. Creating a user (3 of 4)

Creating a user (4 of 4)

- To access the Enterprise Client, the new user is prompted to create a new password and set of security questions
 - Users must log in to the Control Room with their credentials
 - User authorization for the Enterprise Client is managed through the Control Room
 - Features that you can access through the Control Room depend on:
 - The permissions that are granted to the security role that your user account is assigned to
 - The type of license that is assigned to your user account

Welcome! First things first...

Save and log in

Create a password

Please create a password and note down your username.

PASSWORD

Username: mydevuser

Password: *****

SECURITY QUESTIONS

Password

••••••••••

8-15 characters; a-z, A-Z, 0-9, @, ., _, !, #, \$, %, &, and . allowed.

Confirm password

••••••••••

Next >

Welcome! First things first...

[Save and log in](#)

PASSWORD	
Username: mydevuser	
Password: *****	
SECURITY QUESTIONS	

Create a password

Please create a password and note down your username.

Password

8-15 characters; a-z, A-Z, 0-9, @, ., _, !, #, \$, %, &, and . allowed.

Confirm password

[Next >](#)

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-32. Creating a user (4 of 4)

Unit summary

- Describe the main components and features of IBM Robotic Process Automation with Automation Anywhere
- Explain the client/server architecture model that IBM Robotic Process Automation with Automation Anywhere uses
- Describe the Control Room and its features

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-33. Unit summary

Review questions



1. What is the function of the Control Room?
 - A. It is the bot development environment that is used by Developers (Bot Creators).
 - B. It is the central server and repository that is used for bot and user management.
 - C. It is the runtime environment for bot users (Bot Runners).
2. True or false. A bot Developer who is using the Enterprise Client must be logged in to the Control Room.

Figure 2-34. Review questions

Review answers



1. What is the function of the Control Room?
 - A. It is the bot development environment that is used by Developers (Bot Creators).
 - B. It is the central server and repository that is used for bot and user management.
 - C. It is the runtime environment for bot users (Bot Runners).

The answer is B.
2. True or false. A bot Developer who is using the Enterprise Client must be logged in to the Control Room
The answer is True.

Exercise 1: Exploring the Control Room

Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-36. Exercise 1: Exploring the Control Room

Exercise Introduction

- Describe the features of the Control Room
- Create an IBM Robotic Process Automation with Automation Anywhere user and allocate an appropriate license



Introducing IBM Robotic Process Automation with Automation Anywhere

© Copyright IBM Corporation 2018, 2019

Figure 2-37. Exercise Introduction

Unit 3. Introducing the Enterprise Client

Estimated time

01:00

Overview

This unit provides an overview of the Enterprise Client.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe the main features of the Enterprise Client
- Explain the difference between the Web Recorder and the Smart Recorder
- Explain how to work with the Commands list
- Describe good practices for coding bots

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-1. Unit objectives

Topics

- Creating bots
- Overview of the Enterprise Client
- Overview of recorders
- Coding in the Workbench

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-2. Topics

3.1. Creating bots

Creating bots

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-3. Creating bots

What is a bot?

- Software robot, called a *bot*, captures, and replicates human interactions with desktop or web applications
- Use bots to automate repetitive activities that people perform
- Bots integrate with application user interfaces by mimicking keyboard and mouse actions that people use

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-4. What is a bot?

Types of bots

- IBM Robotic Process Automation with Automation Anywhere supports:
 - Bots
 - Metabots
- Bots
 - Bots are the basic building block of RPA
 - Bots can call MetaBots
 - Also referred to as Task bots
 - Easier to build and more resilient to change
- MetaBots
 - Reusable logic block
 - Can be used to create custom commands
 - Can replace scripting or Visual Basic macros

How to create a bot

- You can create bots in two ways:
 - Record human interaction with application user interfaces
 - Manually code the bot
- Recording
 - Use one of the recorders: Smart Recorder, Screen Recorder, or Web Recorder
 - The steps that are captured by the recorder are saved as commands in the Workbench
 - Recorded steps can also be manually edited in the Workbench
- Manually
 - Add the commands directly in the Workbench

Steps for creating bots

1. Record a task or manually code the task in the Workbench
 - Capture human interaction with a user interface
 - Steps of the tasks are recorded as a set of commands
2. Edit the bot in the Workbench
 - Recorded tasks must be edited to include more steps, such as error handling, retrieving data from external services, and other actions
3. Test the bot
 - Develop incrementally and test to verify behavior

After bots are developed and tested, they can be uploaded to the Control Room for deployment on Bot Runners.

3.2. Overview of the Enterprise Client

Overview of the Enterprise Client

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-8. Overview of the Enterprise Client

IBM Training

The screenshot shows the IBM Automation Anywhere Enterprise Client dashboard. The interface is divided into several sections:

- Top Bar:** Includes 'File', 'Edit', 'View', 'Tools', 'Help', and 'Feedback' menus, along with 'New', 'Record', 'Run', 'Edit', and 'ROI' buttons.
- Header:** Features a search bar ('Search Help...'), a bell icon for notifications, and the 'AUTOMATION ANYWHERE Enterprise' logo.
- Left Sidebar (AUTOMATE):** Contains a tree view of 'Tasks' (My Tasks, IBM, Sample Tasks), 'My Scripts', 'My Exes', and 'My Docs'. It also includes sections for 'MetaBots' and 'Workflows'.
- Middle Left Panel:** Titled 'My Tasks', it lists two entries in a table:

File Name	Type	Repeat	Status	Last Run Time
Data Consistency Solution.atmx	Task File	Do not Repeat	Complete	08/16/2018 12:09:41
Transactions Solution.atmx	Task File	Do not Repeat	Complete	08/16/2018 15:01:31
- Bottom Left Panel (Properties):** Shows task details for 'Transactions Solution.atmx' under the 'General' tab:

Repeat	File Name : Transactions Solution.atmx	Task Report
Created at	01/03/2018 04:25:42	Mouse Clicks : 0
Status	Complete	Keystrokes : 0
Notification	Last Run Time : 08/16/2018 15:01:31	Total Clicks : 0
Hotkey	Priority (for queuing) : Low	
Security	Timeout (in minutes) : 0	
<input type="checkbox"/> Enable this task to run with other similar tiles or window titles		
- Right Side (Actions):** Includes 'Delete', 'Upload', and 'Actions' buttons.
- Bottom Navigation:** Shows 'Introducing the Enterprise Client' on the left and '© Copyright IBM Corporation 2018, 2019' on the right.

Five yellow circles with numbers 1 through 5 are overlaid on the interface to highlight specific elements:

- On the 'Run' button in the top menu bar.
- On the search bar in the header.
- On the 'My Tasks' table in the middle-left panel.
- On the 'Actions' buttons in the bottom-right panel.
- On the 'Actions' tab in the bottom-left panel properties.

Figure 3-9. Enterprise Client: Dashboard



Dashboard: Task-relevant activities panel

PROPERTIES	SCHEDULE	TRIGGER
General		
Repeat	File Name : Files-Folders.atmx	Task Report Mouse Clicks 0 Keystrokes 1339 Total Clicks 1339
Speed	Created at : 09-25-2014 21:25:29	
Notification	Status : N/A	
Hotkey	Last Run Time : N/A	
Security	Priority (for queuing) : Low	
	Timeout (in minutes) : 0	
<input type="checkbox"/> Enable this task to run with other similar files or window titles		

PROPERTIES	SCHEDULE	TRIGGER
Launch Task : Please Select a Schedule		
Please Select a Schedule One Time Only		

PROPERTIES	SCHEDULE	TRIGGER
Trigger Type : Please Select a Trigger Type		
Please Select a Trigger Type Window File Folder Performance Process Service Email Message		

Displays task-related Properties, Schedule, and Triggers in the form of tabs

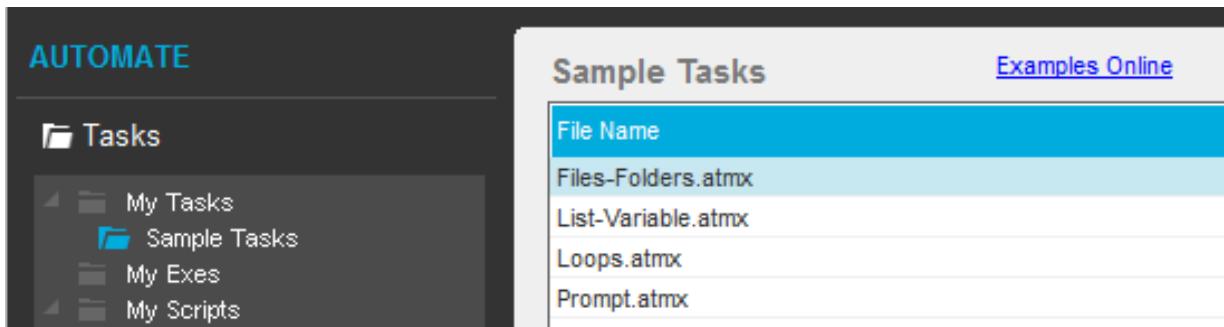
Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-10. Dashboard: Task-relevant activities panel

Accessing bot files

- Bots are saved as **.atmx** files
 - By default, bot files are saved in the following directory:
C:\Users\Administrator\Documents\Automation Anywhere Files\Automation Anywhere\My Tasks
- In the Enterprise Client Dashboard, you can access your bot files from the **Automate** pane **Tasks** folders
- The **Sample Tasks** folder contains some simple example tasks (bots)



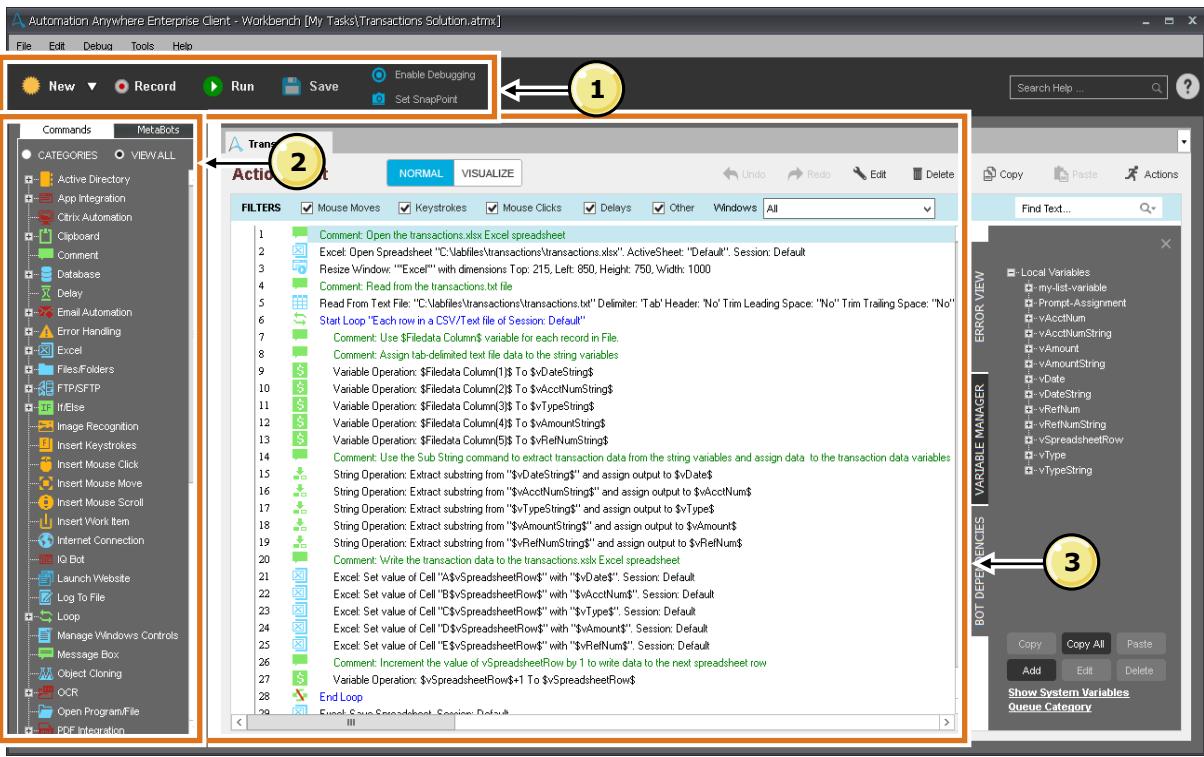
Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-11. Accessing bot files



Enterprise Client: Workbench



Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-12. Enterprise Client: Workbench

3.3. Overview of recorders

Overview of recorders

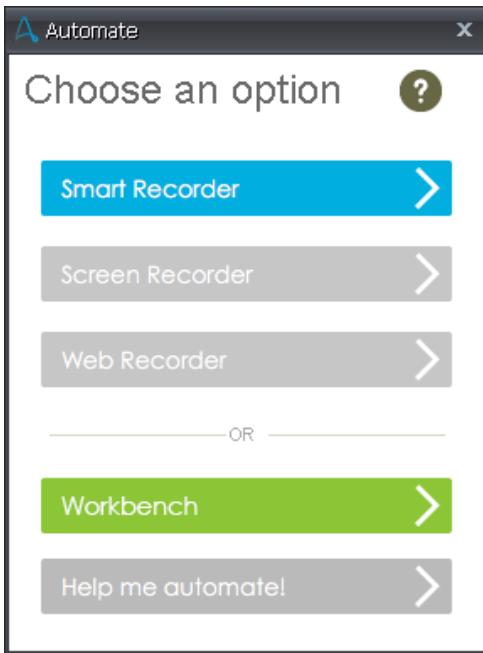
Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-13. Overview of recorders



Overview of recorders



- Smart Recorder
 - Captures objects from various applications that use different technologies, such as HTML, .Net, Windows Presentation Foundation, Java, Flex, and Silverlight
 - Saved as a set of Object Cloning commands
 - Can locate moved controls
- Screen Recorder
 - Records tasks that need to be run on the same computer in which it is recorded
 - Records mouse clicks and keyboard operations
- Web Recorder
 - Understands all web controls
 - Applicable when recording web-only tasks
 - Can locate moved controls
 - If the properties of the web controls change, the task can be updated

Introducing the Enterprise Client

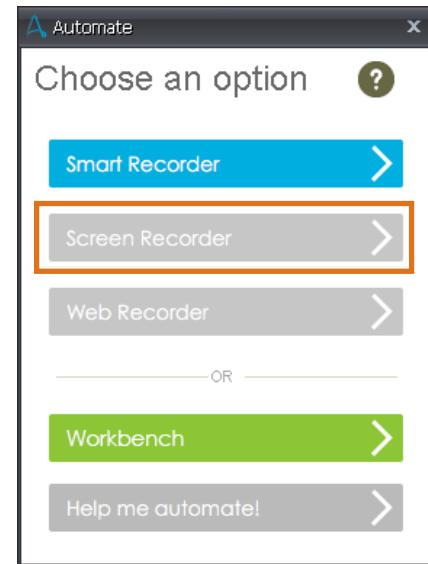
© Copyright IBM Corporation 2018, 2019

Figure 3-14. Overview of recorders



Screen recorder

- Easiest way to create simple automation processes
- Captures mouse clicks and keystrokes (X, Y coordinates)
- Requires that the screen resolution is the same as of the operating system
- Limited to static screens and simple desktops or web applications
- When recording:
 - The Recording toolbar is displayed in the lower-right corner of your computer screen.
 - Complete the actions that you want to record in the task.
 - When you are finished recording, click **Stop** on the Recording toolbar.
 - Save the task by specifying a name, and optionally select a folder in which to store it.



Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

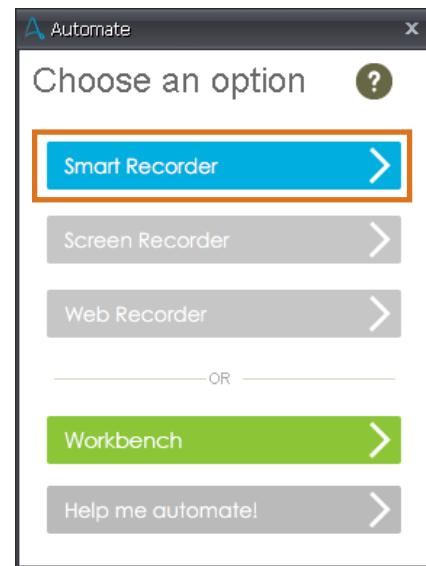
Figure 3-15. Screen recorder



Smart recorder

- Advanced recorder that is used to capture actions and objects in desktop applications
- Capture interactions with Windows objects and controls:
 - Text from drop-down menus
 - Buttons
 - Mouse-click actions
 - Statuses from radio buttons and check boxes
- Can also use to record a task from applications that use HTML, Java, Windows Presentation Foundation, Flex, or Silverlight
- Works on object, coordinates, or image playback mode
- Features adjustable object identification criteria

- To minimize runtime errors:
 - Use the Workbench to add **Open Program / File** commands
 - Avoid double-clicking application icons



[Introducing the Enterprise Client](#)

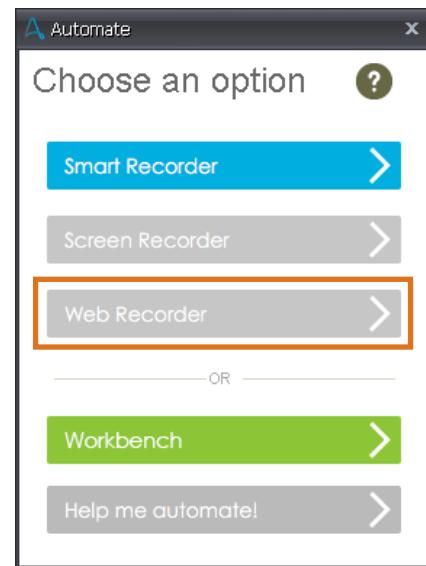
© Copyright IBM Corporation 2018, 2019

Figure 3-16. Smart recorder



Web recorder

- Use when you need to extract website data for process tasks that are based on web
- Extracts single data, pattern-based data, and tables
- Can select web controls
 - If web controls move to a different location on the page, the Web Recorder can locate the new position
- Commonly used for:
 - Repeatedly extracting data from multiple web pages
 - Extracting data from tables on web pages
 - Completing web forms



Introducing the Enterprise Client

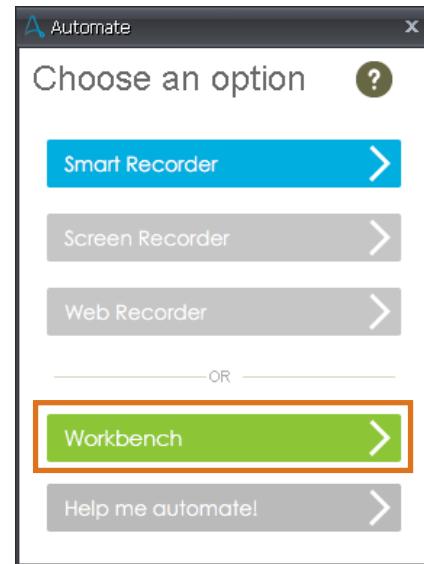
© Copyright IBM Corporation 2018, 2019

Figure 3-17. Web recorder



Using the Workbench with a recorder

- Most powerful feature of the product
- After recording a task with a Recorder, changes can be made to it by using the Workbench
- In many cases, recording and editing a task is much faster than creating the task manually by using the Workbench



Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-18. Using the Workbench with a recorder

3.4. Coding in the Workbench

Coding in the Workbench

Introducing the Enterprise Client

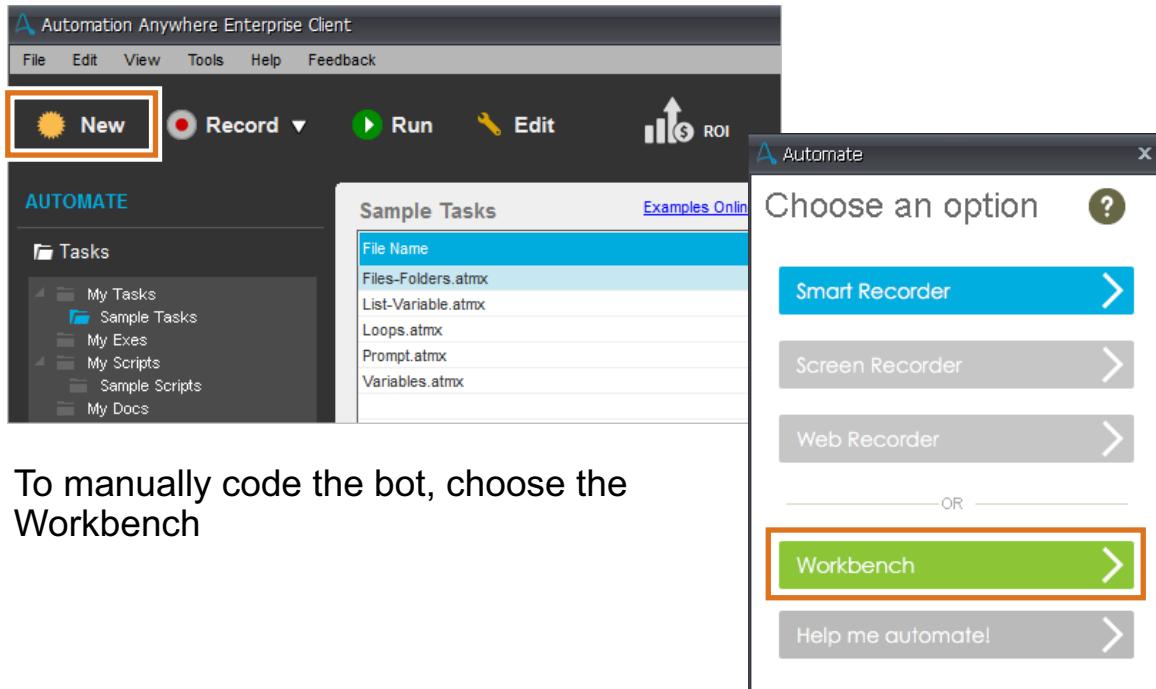
© Copyright IBM Corporation 2018, 2019

Figure 3-19. Coding in the Workbench



Creating a bot

- From the Enterprise Client main window, click **New**



- To manually code the bot, choose the Workbench

Introducing the Enterprise Client

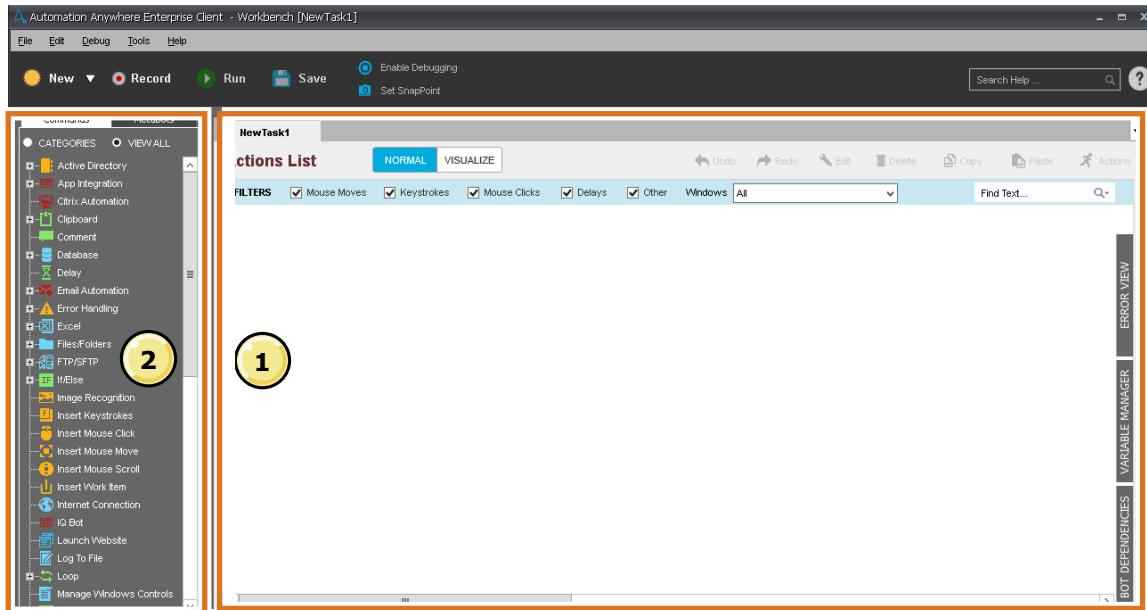
© Copyright IBM Corporation 2018, 2019

Figure 3-20. Creating a bot



The Workbench

- Workbench is the bot development environment
 1. The Actions List is the editor where you build bots
 2. The Commands list contains automation task commands



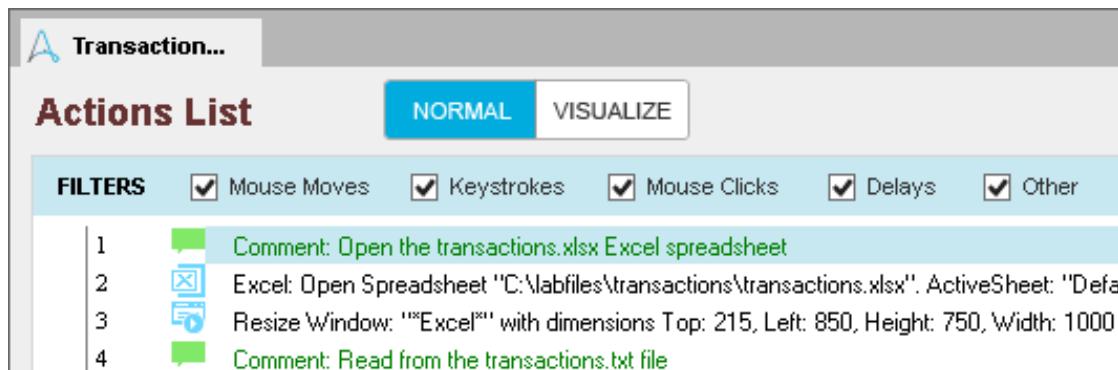
Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-21. The Workbench

The Actions List

- Editor where you add and edit commands
- Uses a proprietary coding environment for building automation tasks
 - Add and edit commands from the Commands list
 - Use a recorder (Smart, Web, or Screen) to add and edit recorded actions



Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

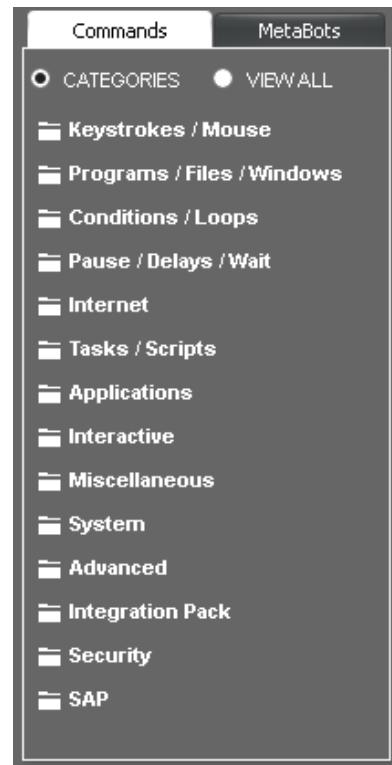
Figure 3-22. The Actions List

Working with the Commands list

- In the Commands list, you can view all the commands in alphabetical order or switch to the Categories view to sort the list

- To add commands from the Commands list:
 - Select the command with your mouse and drag it to the Actions List
 - OR Double-click the command to add it

- Commands are run sequentially
 - To edit the order of commands, you can select a command in the Actions List and drag it to another location in the list



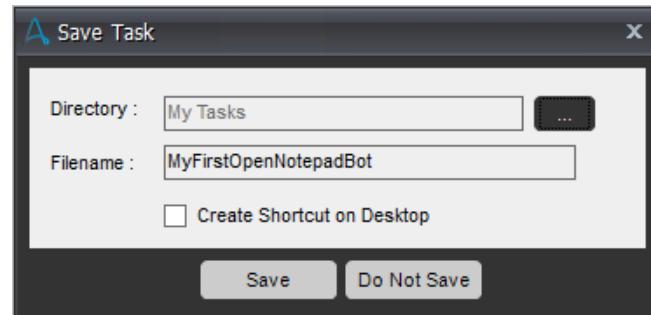
Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-23. Working with the Commands list

Naming conventions

- When you save your bot, you are prompted for a name



- Follow these naming convention guidelines:
 - Use Pascal Case (such as `HelloWorld`)
 - Use numbers in bot names when bots are to be run sequentially and manually
 - Use verbs to describe the bot action
 - Use the word “Helper” for subbots that are called by other scripts
 - Placing a “v” in front of user-defined variables can be helpful in differentiating between user-defined and system-defined variables

Figure 3-24. Naming conventions

Good versus challenging formats for bot development

Good formats	Challenging formats
<ul style="list-style-type: none">• Text fields• Combination boxes• Checkboxes• Radio buttons• First-level menu items• Tables• Excel spreadsheets• CSV and text files• Databases• Copying and pasting through the clipboard• Importing and exporting text• Screens that use the Tab key• Hot keys	<ul style="list-style-type: none">• Handwritten documents• Multiple windows with the same title• Multiple monitors• Monitors that have different screen resolutions• Different versions of the following software:<ul style="list-style-type: none">• Windows operating systems• Internet Explorer• Microsoft Office• Interactive Flash forms• Windows without titles• Tasks that involve dragging items to a different location

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-25. Good versus challenging formats for bot development

Unit summary

- Describe the main features of the Enterprise Client
- Explain the difference between the Web Recorder and the Smart Recorder
- Explain how to work with the Commands list
- Describe good practices for coding bots

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-26. Unit summary

Review questions



1. Which of the following formats are good candidates for task automation? Select all that apply.
 - A. Text fields.
 - B. Handwritten documents.
 - C. Multiple monitors that have different resolutions.
 - D. Excel spreadsheets.
2. True or false. The Workbench is the Enterprise Client tool that you use to create and edit bots.
3. What is the main purpose of using the Smart Recorder?
 - A. Extracting data from web pages.
 - B. Recording basic interactions that involve mouse clicks or keyboard actions in simple desktop applications.
 - C. Capturing interactions with objects and controls in desktop applications that use HTML, Java, Windows Presentation Foundation, Flex, or Silverlight.

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-27. Review questions

Review answers (1 of 2)



1. Which of the following formats are good candidates for task automation? Select all that apply.
 - A. [Text fields](#).
 - B. Handwritten documents.
 - C. Multiple monitors that have different resolutions.
 - D. [Excel spreadsheets](#).

The answers are [A](#) and [D](#).
2. True or false. The Workbench is the Enterprise Client tool that you use to create and edit bots.
The answer is [true](#).

Review answers (2 of 2)



3. What is the main purpose of using the Smart Recorder?
- A. Extracting data from web pages.
 - B. Recording basic interactions that involve mouse clicks or keyboard actions in simple desktop applications.
 - C. Capturing interactions with objects and controls in desktop applications, especially if they use HTML, Java, Windows Presentation Foundation, Flex, or Silverlight.

The answer is **C**.

Figure 3-29. Review answers (2 of 2)

Exercise 2: Creating a basic bot

Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-30. Exercise 2: Creating a basic bot

Exercise introduction (1 of 2)

- Explore the Enterprise Client interface
- Work with the Workbench to add and edit bot commands
- Create a bot that opens a text editor, creates a text file, and saves the file



Introducing the Enterprise Client

© Copyright IBM Corporation 2018, 2019

Figure 3-31. Exercise introduction (1 of 2)

Exercise introduction (2 of 2)

The following commands are used in this exercise:



Command	Purpose
Insert keystrokes	Insert keystrokes into Notepad.
Open program/file	Open Notepad.
Window Actions	Close Notepad window.

Figure 3-32. Exercise introduction (2 of 2)

Unit 4. Working with variables, loops, and strings

Estimated time

01:00

Overview

In this unit, you learn about the IBM RPA with Automation Anywhere implementation of variables and loops, and how they are defined in the Workbench bot coding environment. You learn how to work with data through String Operation commands, and how to use Excel commands to work with spreadsheets. You are also introduced to the exercise case study.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Explain how to define and use variables
- Describe how to set up and use the Loop commands to automate repetitive tasks
- Describe how to use the String Operation commands to work with string data
- Explain how to use Excel commands to work with spreadsheets

Topics

- Overview
- Variables
- Loop commands
- String Operation commands
- Exercise command focus: Excel command
- Introducing the exercise case study

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-2. Topics

4.1. Overview

Overview

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-3. Overview

Overview

- The Enterprise Client Workbench uses a proprietary coding environment that you use during bot development to:
 - Define variables
 - Define task loops
 - Work with string data
- Variables, loops, and string operations are some of the basic building blocks for automating tasks

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-4. Overview

Variables, Loop commands, and String Operation commands

- Variables hold data that is used by bots to complete tasks, and enable bots to pass data between applications
- Loop commands enable bots to run repetitive actions
- String Operation commands enable bots to work with data and assign the output to a variable, such as:
 - Extracting a number from a message (find a substring)
 - Determining the length of a variable value (find the length of a string)

4.2. Variables

Variables

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-6. Variables

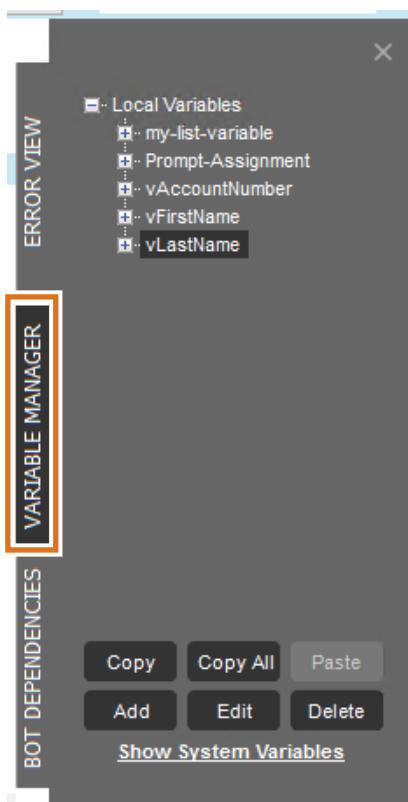
Types of variables

- Variables hold data that bots use to complete tasks
 - Variable names are demarcated in bot code by a pair of dollar signs (\$)
 - Examples: \$vName\$, \$Counter\$
- Two kinds of variables: system and user
- System variables
 - Predefined variables for data in various categories
 - Data types: Date/Time, Loop, Excel, Email, Trigger, PDF, and System
 - Can be used by different bots
- User (or local) variables
 - Created and defined by the bot Developer for use in a specific task
 - Scoped locally to the bot in which they are defined
 - However, you can copy and paste variable definitions from one bot to another

IBM Training 

Variable Manager

- Part of the Enterprise Client Workbench
 - Click the **Variable Manager** vertical tab on the right side of the Workbench window
 - Variable Manager** tab expands into the Workbench window
- Use to:
 - Add and edit user variable definitions
 - Delete user variables
 - Copy and paste variable definitions between different bot files
 - View system variables



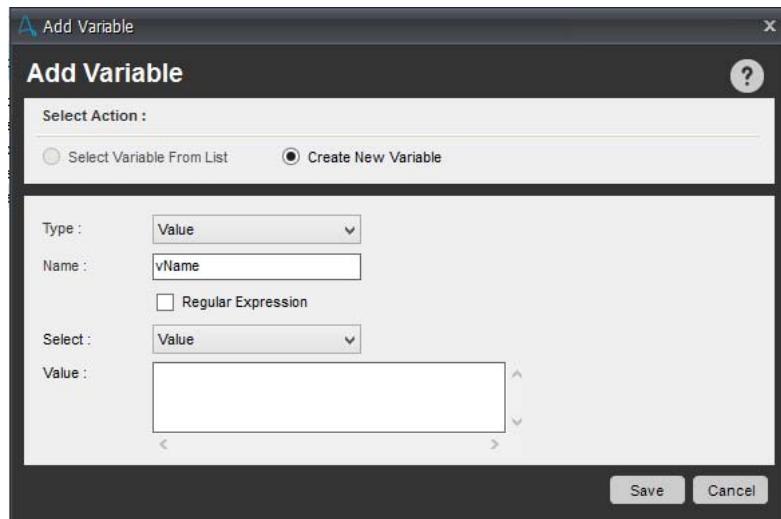
Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-8. Variable Manager



Creating variables (1 of 2)



- Click **Add** in the Variable Manager window
- Use the Add Variable window to define the variable details

Figure 4-9. Creating variables (1 of 2)

Creating variables (2 of 2)

- Define variable type, name, and value
- Type
 - Value: A single value
 - List: Multiple values, one-dimensional
 - Array: Multiple values, two-dimensional (rows and columns)
 - Random: A random string or a random number
- Value
 - Different options for different variable types
 - Value type: Enter a value or leave blank for an initial value of null
 - List type: Enter a list or read values from a text file
 - Array type: Initialize column and row values, or read from a text, CSV file, Excel spreadsheet, or database
 - Random: For a random string, specify the string length; for a random number, enter a number range

Guidelines for naming variables

- Use a lowercase v to prefix user variable names
 - This practice helps you identify user variables in bot code
 - Keeps user variables distinct from system variables
- Use camel case
 - Examples: vReferenceNumber, vListOfUserOptions
- Use descriptive names for variables
 - Variable names should describe the data that the variable holds
- Do not use single character variable names

Using variables in bot commands

- You can assign variables in many bot commands
- You can enter a variable in a bot command by clicking the field and pressing F2
 - Depending on your computer and operating system, you might need to press Fn + F2
 - Opens the Insert Variable window
- Select the variable from the list
 - Can be a user variable or system variable

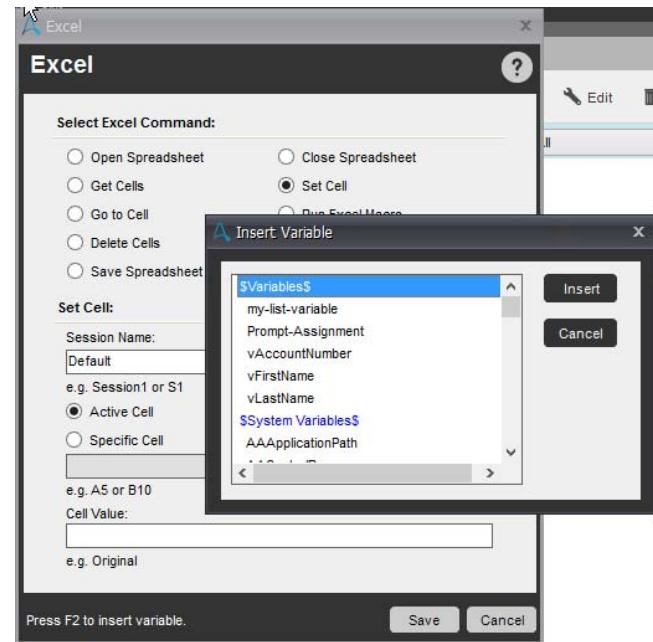
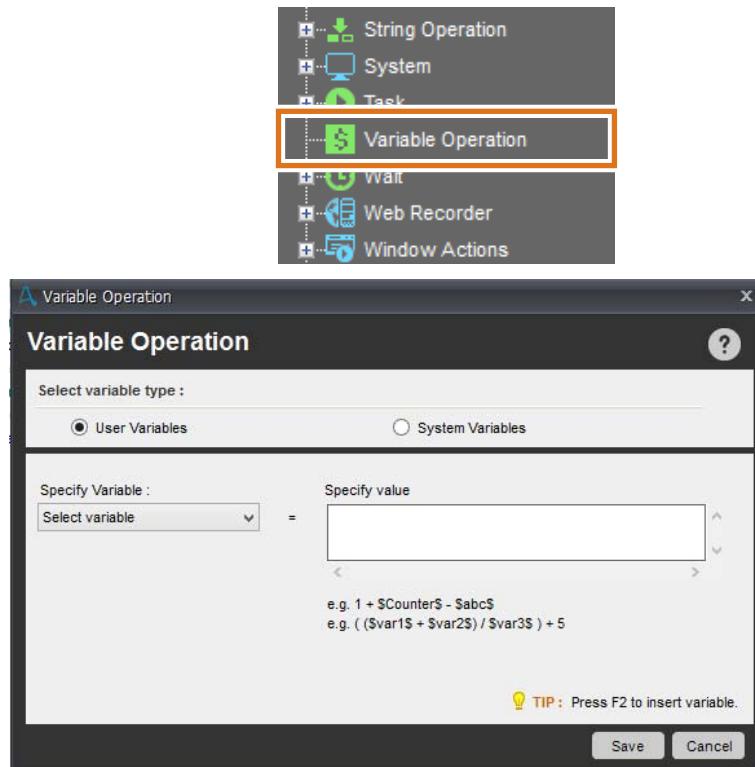


Figure 4-12. Using variables in bot commands



Variable Operation command

- Add from the Commands list
- Use in a task to assign a value to a user variable or a system variable



Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-13. Variable Operation command

4.3. Loop commands

Loop commands

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-14. Loop commands

Overview of Loop commands

- IBM Robotic Process Automation with Automation Anywhere provides a comprehensive set of Loop commands
- Loop commands are in the Commands list
- Four main loop structures:
 - Start Loop
 - End Loop
 - Exit Loop
 - Continue Loop

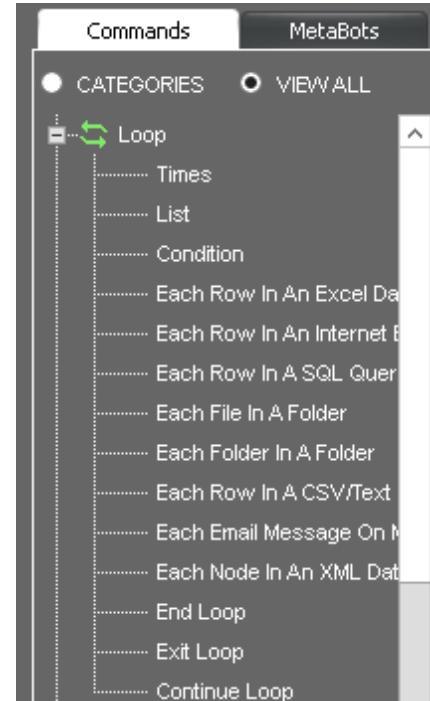
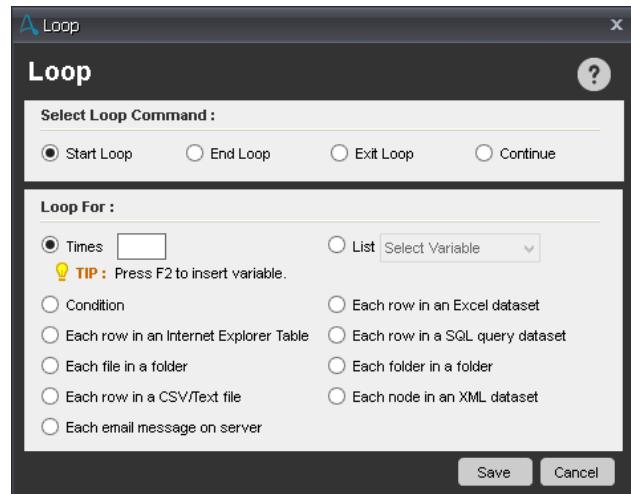


Figure 4-15. Overview of Loop commands

Loop structures

- Start Loop
 - Begins the loop
 - Has 11 different subcommands to handle different kinds of tasks
 - See next slides for more details about the Start Loop subcommands
- End Loop
 - Ends the loop
- Exit Loop
 - Break out of loop
 - Can exist only between a Start Loop command and an End Loop command
- Continue Loop
 - Continue the next iteration when an action or condition occurs
 - If a running bot reaches a Continue Loop command, the remaining commands in the loop are ignored and the next loop iteration starts



Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-16. Loop structures

Start Loop subcommands (1 of 4)

Start Loop subcommand	Description
Times	<ul style="list-style-type: none"> • Use when you know the exact number of times an action must happen • Specify a number of loop iterations
List	<ul style="list-style-type: none"> • Use when you do not know the exact number of times an action must happen • Specify a list variable to iterate through
Condition	<ul style="list-style-type: none"> • Use when actions must run under a specific condition (such as File exists) • Select a condition • Select file to work on while the condition exists • If the condition is a variable, you can define more conditions (and/or conditions)

Figure 4-17. Start Loop subcommands (1 of 4)

Start Loop subcommands (2 of 4)

Start Loop subcommand	Description
Each row in an Excel dataset	<ul style="list-style-type: none"> • Use with Excel commands to work on Excel spreadsheets • Specify an Excel spreadsheet session • Iterates through all rows in the Excel spreadsheet
Each row in an Internet Explorer Table	<ul style="list-style-type: none"> • Use with Extract Table Web Recorder command • Specify an Internet Explorer session • Iterates through all rows in the web table
Each row in an SQL query dataset	<ul style="list-style-type: none"> • Use with database commands • Specify an SQL dataset • Iterates through all rows in the data set

Figure 4-18. Start Loop subcommands (2 of 4)

Start Loop subcommands (3 of 4)

Start Loop subcommand	Description
Each file in a folder	<ul style="list-style-type: none"> • Use to perform actions on all files in a folder • Specify a folder • Iterates through all files
Each folder in a folder	<ul style="list-style-type: none"> • Use to perform actions on all folders in a folder • Specify a folder • Iterates through all folders in the parent folder • Folders are retrieved in alphabetical order
Each row in a CSV/Text file	<ul style="list-style-type: none"> • Use to perform actions on CSV or text file data • Specify a CSV or text file session • Iterates through all rows in the file

Figure 4-19. Start Loop subcommands (3 of 4)

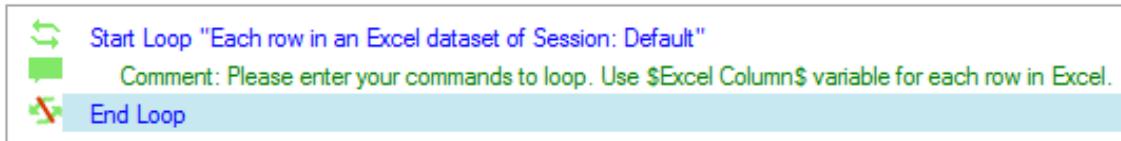
Start Loop subcommands (4 of 4)

Start Loop subcommand	Description
Each node in an XML dataset	<ul style="list-style-type: none"> • Specify an XML session • Iterates through all XML nodes to retrieve data from multiple nodes
Each email message on server	<ul style="list-style-type: none"> • Opens Email Automation window • Specify details: <ul style="list-style-type: none"> • Email server host name • Email account user name and password • Server type: IMAP or POP3, port number • All messages, read messages, or unread messages • Message format: HTML or text • Option to save attachments in a folder that you specify

Figure 4-20. Start Loop subcommands (4 of 4)

Working with Loop commands (1 of 2)

- Adding a Loop command to the Actions List adds a set of Start Loop and End Loop commands



- Each loop must have both a Start Loop and an End Loop command
 - If you accidentally delete an End Loop command, you can add it again from the Commands list

Working with Loop commands (2 of 2)

- Place the actions that you want to repeat inside the Start Loop and End Loop commands
 - These actions are repeated for each item that the loop operates on

```

$ Start Loop "Each row in a CSV/Text file of Session: Default"
$ Comment: Use $Filedata Column$ variable for each record in File.
$ Comment: Assign tab-delimited text file data to the string variables
$ Variable Operation: $Filedata Column(1)$ To $vDateString$
$ Variable Operation: $Filedata Column(2)$ To $vAcctNumString$
$ Variable Operation: $Filedata Column(3)$ To $vTypeString$
$ Variable Operation: $Filedata Column(4)$ To $vAmountString$
$ Variable Operation: $Filedata Column(5)$ To $vRefNumString$
```

- You can nest loops in a bot
 - Child loops are indented one level to visually differentiate them from the parent loop

Figure 4-22. Working with Loop commands (2 of 2)

4.4. String Operation commands

String Operation commands

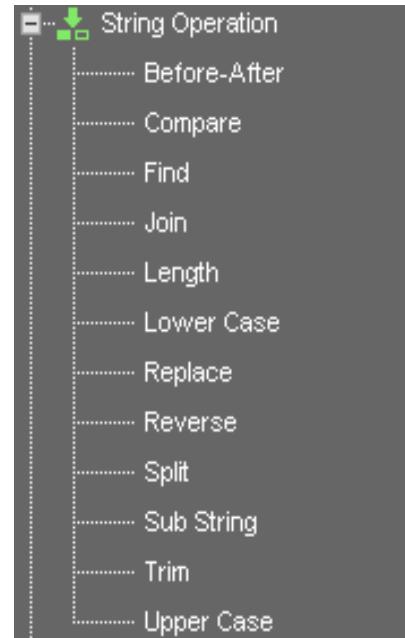
Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-23. String Operation commands

String Operation command overview (1 of 2)

- Use String Operation commands to:
 - Manipulate and work with text strings
 - Assign the result to a variable
- In the Commands list
- Usage examples:
 - Extract an error code from a message window
 - Find the length of a user name
 - Remove unneeded leading spaces from text that was copied from a document
- Useful for working with variable values



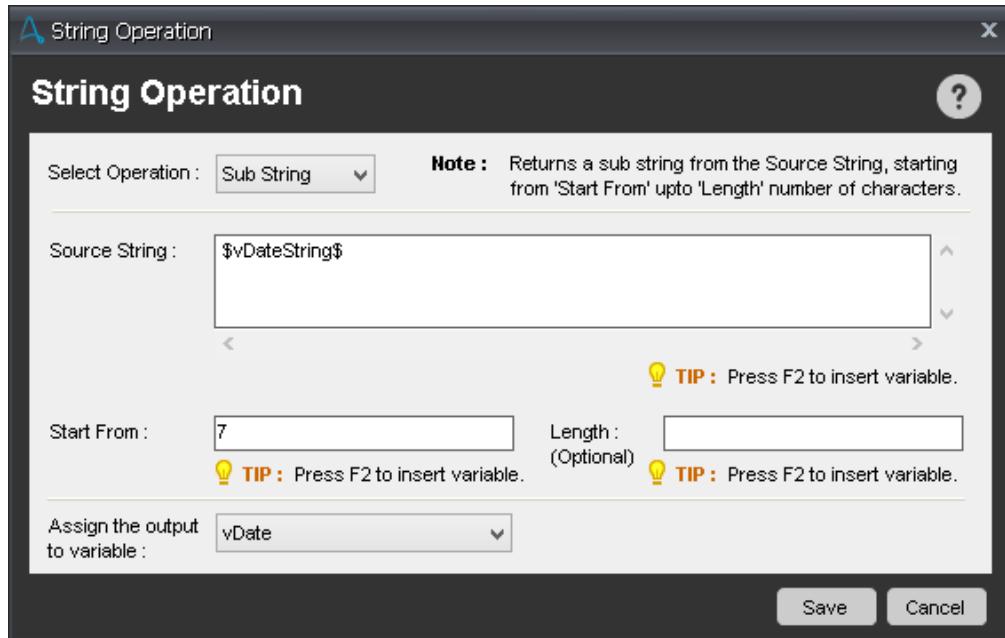
Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-24. String Operation command overview (1 of 2)

String Operation command overview (2 of 2)

- Useful for working with variable values



Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-25. String Operation command overview (2 of 2)

String Operation subcommands (1 of 4)

String Operation subcommand	Description
Before/After	<ul style="list-style-type: none"> Use to extract a substring from a string Define Before and After keywords that serve as substring boundaries <ul style="list-style-type: none"> Can use variables for the Before and After keyword definitions Can use AND/OR operators with Before and After keyword searches Optional: Trim spaces or Enter symbols from web text, specify the number of characters to be extracted If no match is found, can return either source string or an empty (null) string
Compare	<ul style="list-style-type: none"> Use to compare two different strings Specify the two strings to compare Returns a Boolean value Can match cases

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-26. String Operation subcommands (1 of 4)

String Operation subcommands (2 of 4)

String Operation subcommand	Description
Find	<ul style="list-style-type: none"> • Use to find a substring in a string • Define the substring to find • Define a Start parameter <ul style="list-style-type: none"> • The number of the character in the string that you want to start the search from • Can match cases and use this command with regular expressions
Join	<ul style="list-style-type: none"> • Use to join separate elements of a list variable • Specify the list variable • Specify the delimiter <ul style="list-style-type: none"> • Example: comma (,)
Length	<ul style="list-style-type: none"> • Use to find the length of the string • Specify the source string

Figure 4-27. String Operation subcommands (2 of 4)

String Operation subcommands (3 of 4)

String Operation subcommand	Description
Lower Case	<ul style="list-style-type: none"> • Use to convert a string to lowercase text • Specify the source string
Replace	<ul style="list-style-type: none"> • Use to replace part of a source string with other text • Specify the source string • Specify the replacement text • Optional: Specify character start position or number of times the target should be replaced • Can match cases and use with regular expressions
Reverse	<ul style="list-style-type: none"> • Use to reverse a string • Specify the source string to reverse
Split	<ul style="list-style-type: none"> • Use to split a string • Specify the source string • Specify a delimiter • Optional: Specify the number of delimiters to account for • Can match case

Figure 4-28. String Operation subcommands (3 of 4)

String Operation subcommands (4 of 4)

String Operation subcommand	Description
Sub String	<ul style="list-style-type: none"> • Use to retrieve a substring • Specify the source string • Specify the character start position • Option: Specify the length of the substring
Trim	<ul style="list-style-type: none"> • Use to remove spaces from a string • Specify source string • Specify trim options <ul style="list-style-type: none"> • From left • From right • From both left and right
Upper Case	<ul style="list-style-type: none"> • Use to convert a string to uppercase • Specify the source string

Figure 4-29. String Operation subcommands (4 of 4)

Working with String Operation commands

- The output of a string operation command is typically assigned to a variable
 - Select the variable to assign output to when you define the String Operation command
- Can usually use variables as source strings or to define other String Operation parameters
 - Use F2 to insert variables into fields
- Can also use keystrokes to define some parameters, such as delimiters
 - Use IBM Robotic Process Automation with Automation Anywhere keystroke syntax
 - Use square brackets ([]) to denote the keystroke
 - Enter the key name in uppercase
 - Examples: [ENTER] or [TAB]

4.5. Exercise command focus: Excel command

Exercise command focus: Excel command

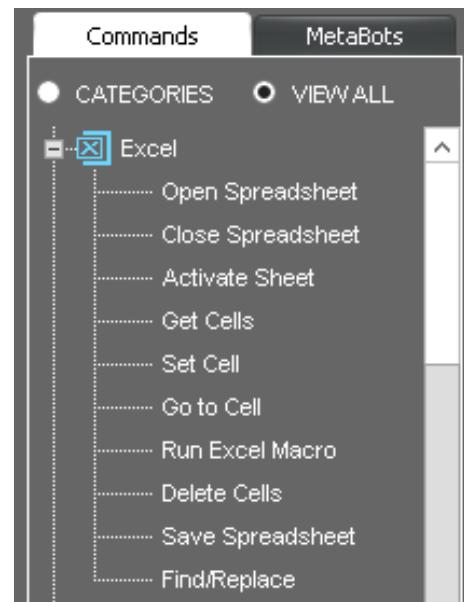
Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-31. Exercise command focus: Excel command

Introduction to the Excel commands

- The exercise introduces you to working with variables, loops, and String Operation commands
- The exercise also introduces the Excel commands
 - Use these commands to work with Microsoft Office Excel spreadsheets
 - Non-Excel spreadsheets are not supported



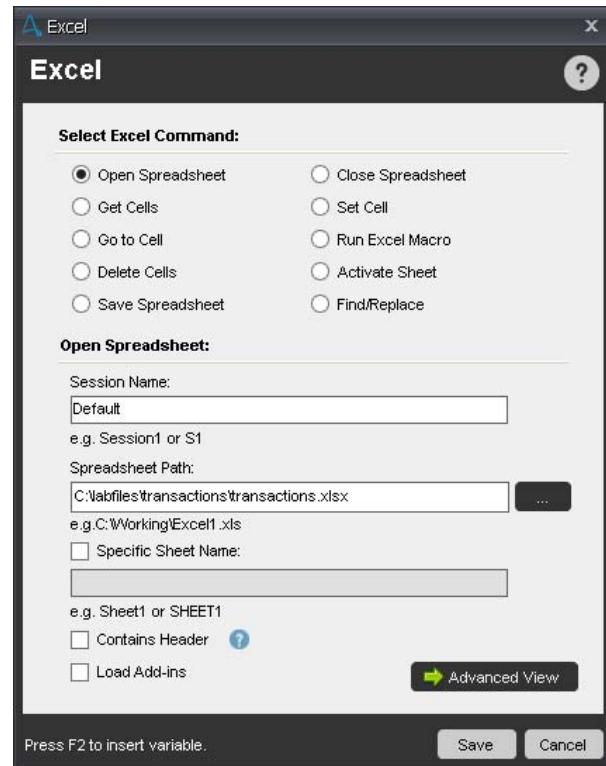
Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-32. Introduction to the Excel commands

Introduction to the Excel commands

- Includes subcommands for common spreadsheet-related tasks, such as:
 - Opening or closing a spreadsheet file
 - Getting or setting the value of cells
 - Saving the spreadsheet file



Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-33. Introduction to the Excel commands

Excel subcommands (1 of 5)

Excel subcommand	Description
Open Spreadsheet	<ul style="list-style-type: none"> • Use to open an Excel spreadsheet file • If the task involves multiple spreadsheet files, you can specify the spreadsheet session name to make sure the bot works with the correct file • Other parameters to define: <ul style="list-style-type: none"> • Spreadsheet Path (directory path) • Specific sheet name • Whether the spreadsheet contains a header row • Whether the bot needs to load Excel add-ins
Close Spreadsheet	<ul style="list-style-type: none"> • Use to close an Excel spreadsheet • Specify the session name of the spreadsheet that you want to close • Can also specify whether you do not want to save any changes

Figure 4-34. Excel subcommands (1 of 5)

Excel subcommands (2 of 5)

Excel subcommand	Description
Activate Sheet	<ul style="list-style-type: none"> • Use to activate a specific spreadsheet in an open Excel workbook • Specify the spreadsheet session name • Specify the sheet by index number or by name
Get Cells	<ul style="list-style-type: none"> • Use to get the value of a cell • Specify the get operation: <ul style="list-style-type: none"> • Get Single Cell • Get Multiple Cells • Get All Cells • Specify the spreadsheet session name • Specify whether to get the value from the active cell or from a specific cell <ul style="list-style-type: none"> • For a specific cell, identify it by column letter and row number, such as C2 or E8 • Can assign the value to a variable

Figure 4-35. Excel subcommands (2 of 5)

Excel subcommands (3 of 5)

Excel subcommand	Description
Set Cells	<ul style="list-style-type: none"> • Use to set the value of a cell • Specify the spreadsheet session name • Specify whether to set the value of the active cell or a specific cell <ul style="list-style-type: none"> • For a specific cell, identify it by column letter and row number • Specify the cell value to set <ul style="list-style-type: none"> • Can assign a variable to the Cell Value field
Go to Cell	<ul style="list-style-type: none"> • Use to move to a specific cell • Specify the spreadsheet session name • Specify a specific cell to move to, or define movement options • Movement options include: <ul style="list-style-type: none"> • Moving one cell to the left, right, above, or below • Beginning of the row or the column • End of the row or the column

Figure 4-36. Excel subcommands (3 of 5)

Excel subcommands (4 of 5)

Excel subcommand	Description
Run Excel Macro	<ul style="list-style-type: none"> • Use to run an Excel macro in a spreadsheet • Spreadsheet must be opened by the Open Spreadsheet subcommand • Specify the spreadsheet session name • Specify the macro name • Optional: Specify macro arguments
Delete Cells	<ul style="list-style-type: none"> • Use to delete the values of a cell • Spreadsheet must be opened by the Open Spreadsheet subcommand • Specify the spreadsheet session name • Specify whether to delete the active cell or a specific cell • Deletion options: <ul style="list-style-type: none"> • Shift cells left or up • Delete entire row or entire column

Figure 4-37. Excel subcommands (4 of 5)

Excel subcommands (5 of 5)

Excel subcommand	Description
Save Spreadsheet	<ul style="list-style-type: none"> • Use to save an open spreadsheet • Specify the spreadsheet session name
Find/Replace	<ul style="list-style-type: none"> • Use to find a value and replace it with a different value • Specify cell range <ul style="list-style-type: none"> • Default is entire spreadsheet (from beginning until end), but you can specify a range of cells • Specify the spreadsheet session name • Specify the value to find • Specify the search options <ul style="list-style-type: none"> • Search by rows or by columns • Can match case or match entire cell contents • Specify the replacement value • Can assign cell addresses to a variable

Figure 4-38. Excel subcommands (5 of 5)

4.6. Introducing the exercise case study

Introducing the exercise case study

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-39. Introducing the exercise case study

Introduction to the exercise case study (1 of 2)

- In the exercises in this training, you work with the Control Room and the Enterprise Client to manage and automate various tasks for a fictional financial services firm.



Smart Investment Bank (SIB)

Smart Investment Bank, or SIB, is a small bank that handles various financial transactions for its customers.

SIB recently acquired MyBank, another financial services firm, and as part of the merger, SIB and MyBank plan to integrate their systems.

However, the SIB and MyBank systems are not integrated yet. SIB wants to use IBM RPA to automate some of their tasks.

Introduction to the exercise case study (2 of 2)



Smart Investment Bank (SIB)

The exercise case study starts after the SIB automation team completed the project planning phase and identified high-priority processes with tasks that are suitable for automation.

As part of the SIB development team, you must develop bots for the following processes:

- Collecting MyBank transaction data
- Opening an account
- Booking share trades
- Ensuring that account data is consistent across multiple systems
- Summing declined checks and emailing a Margin Clerk
- Evaluating data from a PDF and sending an email

Case study bot development exercises overview (1 of 3)

- The following exercises cover bot development and are based on the exercise case study
- In some of the later exercises, you extend the functionality of a Task bot that you developed in a previous exercise

Number	Title	Description
3	Writing data from a text file to an Excel spreadsheet	Transfer transaction data from one source to another
4	Automating data entry to a business application and a database	Transfer account data from a spreadsheet to a separate Windows application and a Db2 database
5	Creating a bot to sum check declines, query a database, and send an email	Sum check values from a CSV file, check a database for the customer's membership level, and send an email
6	Creating a bot to evaluate data from a PDF and send an email	Extract data from a trade receipt PDF and use conditional logic to determine which user to email

Figure 4-42. Case study bot development exercises overview (1 of 3)

Case study bot development exercises overview (2 of 3)

Number	Title	Description
7	Creating an interactive bot to check values in disparate systems	User triggers an interactive bot to retrieve account data from a web table, an Access database, and an Excel spreadsheet, and display it in a message window.
8	Creating a login MetaBot	Based on the bot from Exercise 7. Replace Access database login with a MetaBot that uses a credential from the Credential Manager.
9	Working with web services and error handling	Based on the bot from Exercise 6. Add a REST call to the bot and add error handling commands.
10	Hardening the Account Opening bot	Based on the bot from Exercise 4. Add commands to handle known issues with the exercise spreadsheet, and create a separate error-handling bot.

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-43. Case study bot development exercises overview (2 of 3)

Case study bot development exercises overview (3 of 3)

- The following exercises cover Bot management and administrative topics

Number	Title	Description
11	Managing bots	<p>Build a task bot that passes the Account Number as a parameter to another bot. Deploy the bot, perform a code comparison, and run the bot.</p> <p>View system logs and notifications. Build a report by using the Report Designer</p>
12	Administering bots in the Control Room	This exercise provides an overview of central bot administration tasks that you can configure and access through the Control Room. You learn how an administrator can use various Control Room features to manage, schedule, and run bots on remote user (BotRunner) machines.

Figure 4-44. Case study bot development exercises overview (3 of 3)

Command matrix: Commands covered

- This command matrix displays all commands that are used in this course during bot development
- Exercises 3-10 cover bot development
 - Most of this course focuses on bot development
 - The ratio in which the commands are used follow typical development patterns
 - For example, Looping and the If/Else commands are used extensively, which are also used extensively in real-world situations
- Exercises 4-5 use the Smart Recorder
- Exercise 7 uses the Web Recorder
- Exercise 8 is a MetaBot exercise
- Exercises 11 and 12 deal with bot management and administrative topics

COMMAND	EXERCISE LAB USAGE									
	3	4	5	6	7	8	9	10		
Clipboard			X							X
Comment	X	X	X	X	X				X	X
Database		X	X							
Email Automation					X					
Error handling									X	X
Excel	X	X				X			X	
Files/folders					X				X	
If/else		X	X	X	X				X	X
Insert keystrokes			X							X
Loop	X	X	X	X	X					
Message Box							X			
Object cloning	X	X				X				X
Open program file	X	X				X				
PDF integration					X					
Prompt							X			
Read from CSV/Text	X		X							
REST web service									X	
Send email			X	X					X	X
String operation	X	X			X				X	X
Task							X			
Variable operation	X	X	X			X			X	
Web recorder								X		
Window actions	X	X				X				X

© Copyright IBM Corporation 2018, 2019

Working with variables, loops, and strings

Figure 4-45. Command matrix: Commands covered

Command matrix: Commands not covered

- This command matrix displays commands that are NOT used in this course during bot development

COMMAND	COMMAND
Active Directory	Manage windows controls
App Integration	OCR
Citrix Automation	PGP
Delay	Play sound
FTP/SFTP	Printers
Image recognition	Run script
Insert mouse click	SAP integration
Insert mouse move	Screen capture
Insert mouse scroll	Services
Insert Work Item	SNMP
Internet connection	SOAP web service
IQ Bot	System
Launch website	Terminal Emulator
Log to file	Wait
	XML

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-46. Command matrix: Commands not covered

Unit summary

- Explain how to define and use variables
- Describe how to set up and use the Loop commands to automate repetitive tasks
- Describe how to use the String Operation commands to work with string data
- Explain how to use Excel commands to work with spreadsheets

Review questions



1. Which of the following statements apply to variables in IBM Robotic Process Automation with Automation Anywhere?
Select all that apply.
 - A. User variables are globally scoped to all bots.
 - B. System variables are predefined variables that handle certain Date/Time, Loop, Excel, Email, Trigger, PDF, and System data.
 - C. You can copy and paste user variable definitions from one bot file to another.
2. True or false. A loop must have both a `Start Loop` and an `End Loop` command.
3. True or false. You can assign the output of a String Operation command to a variable.

Review answers



1. Which of the following statements apply to variables in IBM Robotic Process Automation with Automation Anywhere? Select all that apply.

- A. User variables are globally scoped to all bots.
- B. System variables are predefined variables that handle certain Date/Time, Loop, Excel, Email, Trigger, PDF, and System data.
- C. You can copy and paste user variable definitions from one bot file to another.

The answers are **B** and **C**.

2. True or false. A loop must have both a Start Loop and an End Loop command.

The answer is True.

3. True or false. You can assign the output of a String Operation command to a variable.

The answer is True.

Exercise 3: Writing data from a text file to an Excel spreadsheet

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-50. Exercise 3: Writing data from a text file to an Excel spreadsheet

Exercise objectives

- Define user variables
- Create a loop to iterate through a spreadsheet
- Work with String Operation commands



Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-51. Exercise objectives

Exercise introduction

The following commands are used in this exercise:



Command	Purpose
Comment	Insert code annotations
Excel	Open the transactions.xlsx spreadsheet and sets value of cells to values extracted from a text file Close the spreadsheet
Loop	Loop for each row in a CSV/text file
Read From CSV/Text	Read from the text file
String Operation	Extract substring data
Variable Operation	Transfer extracted data from system variables <code>\$Filedata Column(n) \$</code> to custom variables Increment counter
Window Actions	Resize the spreadsheet window

Working with variables, loops, and strings

© Copyright IBM Corporation 2018, 2019

Figure 4-52. Exercise introduction

Unit 5. Working with the Smart Recorder and the Object Cloning command

Estimated time

01:00

Overview

In this unit, you learn how to work with the Smart Recorder and the Object Cloning command to capture and edit bot interactions with other applications.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe how to capture bot interactions with the Smart Recorder
- Describe how to capture bot interactions with the Object Cloning command
- Explain how to edit Object Cloning commands in the Workbench

Topics

- Working with the Smart Recorder
- Working with Object Cloning commands
- Exercise command focus: Files/Folders commands
- Exercise command focus: Database commands

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-2. Topics

5.1. Working with the Smart Recorder

Working with the Smart Recorder

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-3. Working with the Smart Recorder

Smart Recorder overview

- Records interactions with application objects, such as:
 - Buttons
 - Fields
 - Check boxes
- Use for interacting with applications that use HTML, .Net, Windows Presentation Foundation (WPF), Java, Flex, and Silverlight
- Recorded actions are saved in the Task Actions List as Object Cloning commands
 - Object Cloning commands are covered in the next topic of this unit
- Smart Recorder supports Secure Recording mode
 - Must enable – is not enabled by default
 - Does not record images or data
 - Useful for organizations that handle sensitive data, such as customer information

[Working with the Smart Recorder and the Object Cloning command](#)

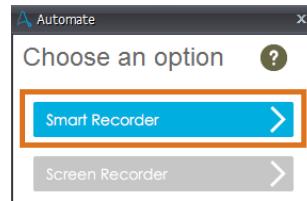
© Copyright IBM Corporation 2018, 2019

Figure 5-4. Smart Recorder overview

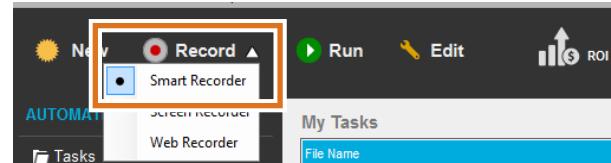
Starting the Smart Recorder

- To start the Smart Recorder, use one of these methods:

- When you start a new task, click **Smart Recorder**



- In the Enterprise Client, click **Record > Smart Recorder**



- In the Workbench, click **Record**

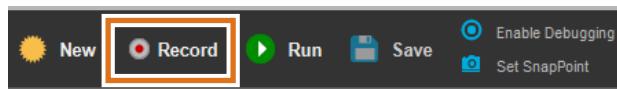
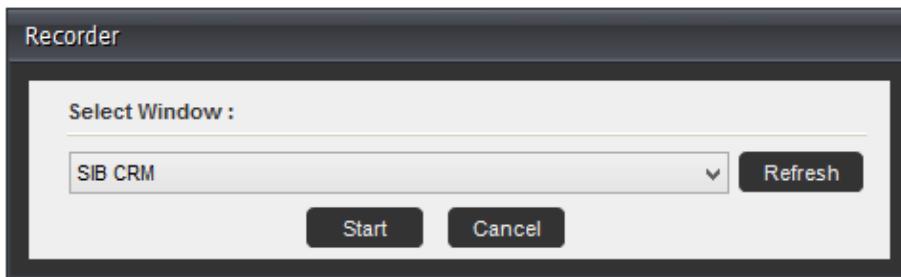


Figure 5-5. Starting the Smart Recorder

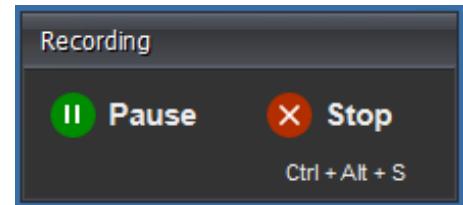


Recording actions (1 of 2)

- When you start the Smart Recorder:
 - Recorder window opens
 - Select the window that you want to work with and click **Start**
 - If you do not see the window that you need, click **Refresh**



- Recording window opens with **Pause** and **Stop** options
 - Default window location is lower-right area of the screen
 - You can move location of Recording window



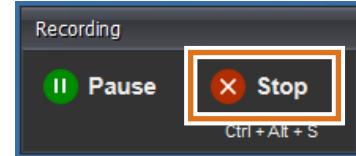
[Working with the Smart Recorder and the Object Cloning command](#)

© Copyright IBM Corporation 2018, 2019

Figure 5-6. Recording actions (1 of 2)

Recording actions (2 of 2)

- Smart Recorder records all user interactions with the window
 - Button clicks
 - Keystrokes (such as typing in a field),
 - Selecting radio buttons or check boxes
- As you use the Smart Recorder, you might record extra actions
 - Examples: Accidental clicks or keystrokes
 - These actions can be deleted later in the Workbench
- You can either record all task interactions in one recording or record each action individually
- To stop recording, in the Recording window, click **Stop**



Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-7. Recording actions (2 of 2)

Working with recorded actions (1 of 2)

- Actions that are recorded with the Smart Recorder are saved as Object Cloning commands in the Task Actions List



- Delete unnecessary or extra recorded actions from the Task Actions List
 - Right-click the unneeded Object Cloning command, and click **Delete**

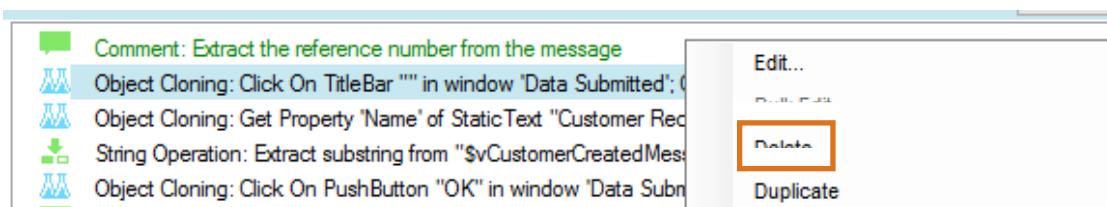


Figure 5-8. Working with recorded actions (1 of 2)

Working with recorded actions (2 of 2)

- You can edit a recorded Object Cloning command in the Workbench
- Example:
 - You need to copy data from a form field
 - The Smart Recorder recorded extra keystrokes when you used the keyboard to capture a Ctrl+C action for the field contents
 - You can edit the Object Cloning command to update the captured keystrokes so that the command includes only the keystrokes that you need

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-9. Working with recorded actions (2 of 2)

5.2. Working with Object Cloning commands

Working with Object Cloning commands

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-10. Working with Object Cloning commands

Object Cloning command overview

- Add from Commands list or through the Smart Recorder
- Add Object Cloning commands manually to bot
- Editing an Object Cloning command that was added by using the Smart Recorder opens the Object Cloning window
 - You can edit various attributes of the recorded task

Using the Object Cloning command window

1. Specify the application window
2. Capture the object
3. If needed, define or update relevant object properties
 - For Object mode: Properties such as object name and type
 - For Coordinates mode: Object X, Y coordinates, and object title
 - For Image mode: Image of object
4. Select actions to be performed on the object

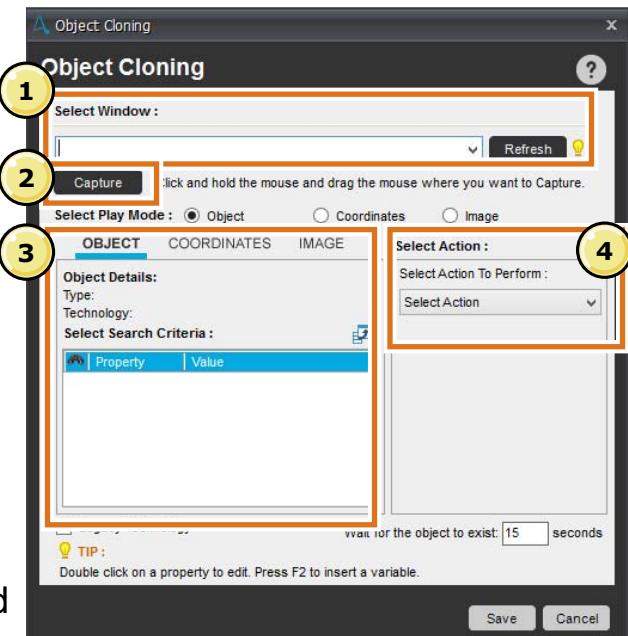
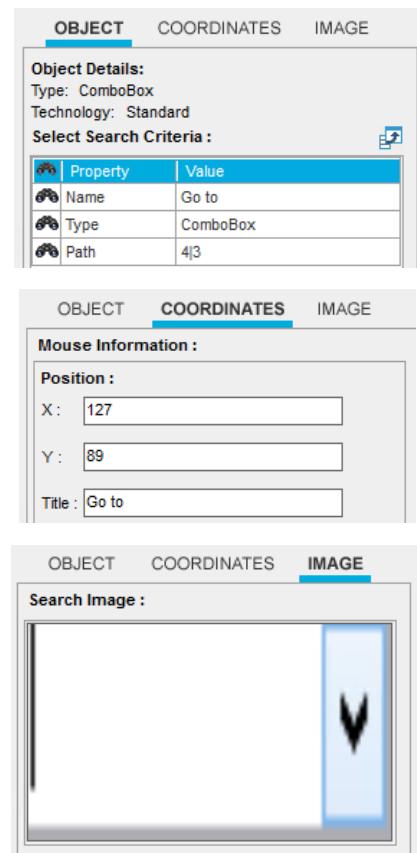


Figure 5-12. Using the Object Cloning command window

Object Cloning play mode options

When you run a bot, it can interact with the captured object in three different ways:

- **Object:** Work with the captured object and its properties
- **Coordinates:** Object is defined by location of screen coordinates
- **Image:** Object is defined by an image of the object



The figure consists of three separate windows from the Smart Recorder application, each demonstrating a different play mode option:

- OBJECT Play Mode:** This window shows "Object Details" for a "ComboBox" type object with "Standard" technology. It includes a "Select Search Criteria" section with a table:

Property	Value
Name	Go to
Type	ComboBox
Path	4 3

- COORDINATES Play Mode:** This window shows "Mouse Information" with fields for Position (X: 127, Y: 89) and Title (Go to).
- IMAGE Play Mode:** This window shows a search for an image, displaying a preview of the object's visual representation.

Working with the Smart Recorder and the Object Cloning command

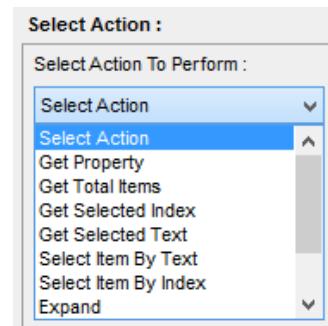
© Copyright IBM Corporation 2018, 2019

Figure 5-13. Object Cloning play mode options

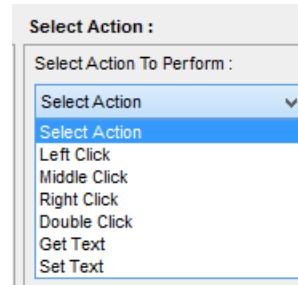
Defining the Object Cloning actions

- Select an action for the object
 - Object action options depend on Play Mode that you use

- Object mode actions
 - Get: Property, Total Items, Selected Index, Selected Text
 - Select: Item By Text, Item By Index
 - Expand
 - Click: Left or Right
 - Double Click



- Coordinates and Image mode actions:
 - Click: Left, Right, Middle
 - Double Click
 - Get Text
 - Set Text



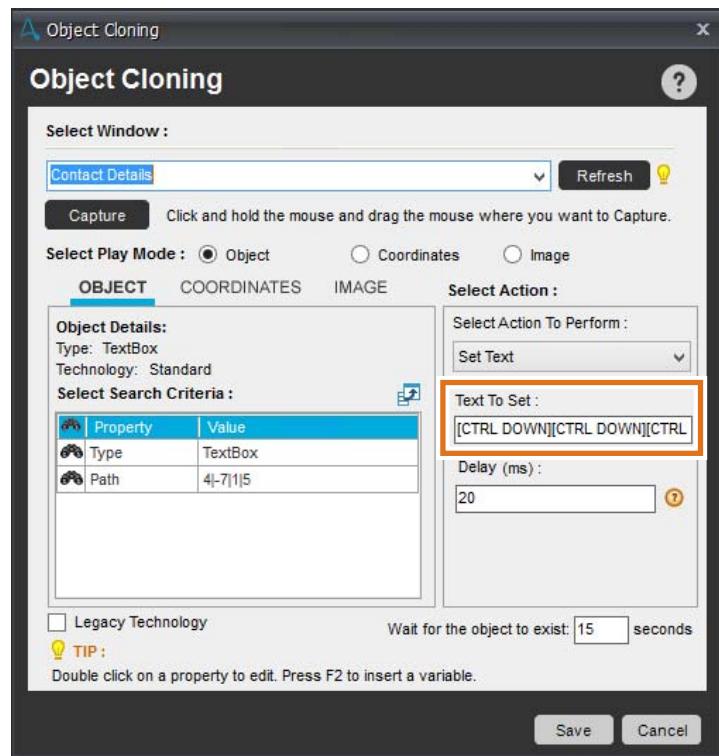
[Working with the Smart Recorder and the Object Cloning command](#)

© Copyright IBM Corporation 2018, 2019

Figure 5-14. Defining the Object Cloning actions

Editing an Object Cloning command

- Double-click the Object Cloning command in the Task Actions List to open the Object Cloning window
- You can edit or update the fields as needed
- Example:
 - The **Text To Set** field contains extra [CTRL DOWN] keystrokes
 - Can edit the input in this field to include only the keystrokes that are needed for the Ctrl+C action



Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-15. Editing an Object Cloning command

5.3. Exercise command focus: Files/Folders commands

Exercise command focus: Files/Folders commands

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-16. Exercise command focus: Files/Folders commands



Files/Folders commands

Basic commands to work with either files or folders

The image shows two views of the 'File/Folders' command. On the left is a list of commands under the 'File/Folders' category. On the right is a detailed configuration dialog for the 'File/Folders' command.

Commands View:

- CATEGORIES: Files/Folders
 - Copy Files
 - Rename Files
 - Delete Files
 - Create File
 - Zip Files
 - Unzip Files
 - Open File
 - Create File Shortcut
 - Copy Folder
 - Rename Folder
 - Delete Folder
 - Create Folder
 - Open Folder
 - Create Folder Shortcut
 - Print File
 - Print Multiple Files

Configuration Dialog:

File/Folders

File or Folders Command:

File Folder

Select Command:

<input checked="" type="radio"/> Copy Files	<input type="radio"/> Rename Files
<input type="radio"/> Delete Files	<input type="radio"/> Create File
<input type="radio"/> Zip Files	<input type="radio"/> Unzip Files
<input type="radio"/> Print File	<input type="radio"/> Print Multiple Files
<input type="radio"/> Open File	<input type="radio"/> Create File Shortcut

Source File(s): e.g. C:\MyDoc*.doc **TIP:** Press F2 to insert variable.

Destination File(s) / Folder: e.g. C:\Backup\ **TIP:** Press F2 to insert variable.

Overwrite Files/Folders

Working with the Smart Recorder and the Object Cloning command

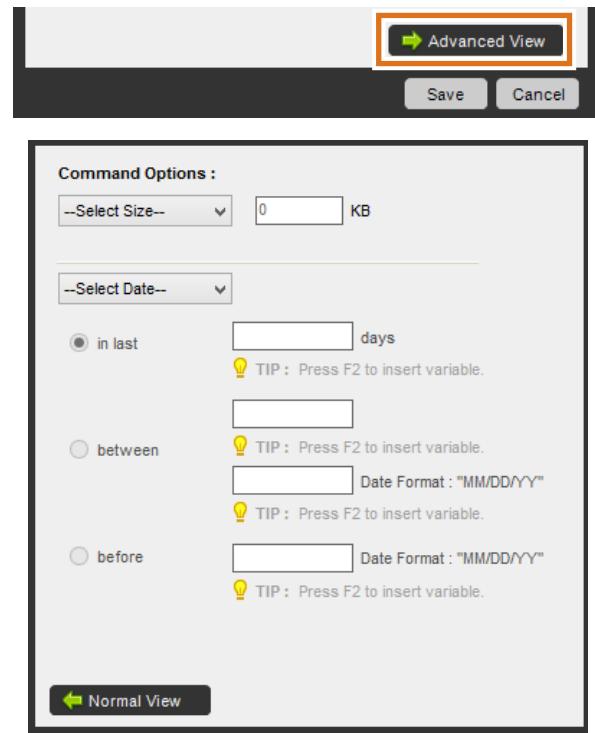
© Copyright IBM Corporation 2018, 2019

Figure 5-17. Files/Folders commands



Files/Folders command: Advanced View

- Most Files/Folders commands have advanced options
 - Exceptions: Create commands or Open commands
 - Click **Advanced View** to open
- Can filter files by size or by date
- Size options:
 - Atleast
 - Atmost
 - Exact
- Date options:
 - Modified Date
 - Created Date



Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-18. Files/Folders command: Advanced View

Files subcommands (1 of 3)

Files subcommand	Description
Copy Files	<ul style="list-style-type: none"> • Use to copy a file • Specify the source file • Specify either: <ul style="list-style-type: none"> • Destination file name or folder • Can select option to overwrite files or folder
Rename Files	<ul style="list-style-type: none"> • Use to rename a file • Specify the file directory and name • Specify the new file name
Delete Files	<ul style="list-style-type: none"> • Use to delete a file • Specify the file directory and name
Create File	<ul style="list-style-type: none"> • Use to create a file • Specify the new file directory and name • Can select option to overwrite files or folder

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-19. Files subcommands (1 of 3)

Files subcommands (2 of 3)

Files subcommand	Description
Zip Files	<ul style="list-style-type: none"> • Use to compress files into a .zip file • Specify files to compress • Specify destination .zip file name • Optional: Specify file type
Unzip Files	<ul style="list-style-type: none"> • Specify the .zip file to extract • Specify the folder to store the contents of the .zip file • Can select option to overwrite file or folder
Open File	<ul style="list-style-type: none"> • Use to open a file • Specify the file directory and name
Create File Shortcut	<ul style="list-style-type: none"> • Use to create a shortcut that you can use to open a file • Specify the file directory and name • Specify the destination folder for the shortcut

Files subcommands (3 of 3)

Files subcommand	Description
Print File	<ul style="list-style-type: none">• Use to print a file• Specify the file directory and name
Print Multiple Files	<ul style="list-style-type: none">• Use to print all of the files in a folder• Specify the folder name• Optional: Specify File Type• Can select option to include subfolders

Figure 5-21. Files subcommands (3 of 3)

Folders subcommands (1 of 2)

Files subcommand	Description
Copy Folder	<ul style="list-style-type: none"> • Use to copy a folder • Specify the source folder • Specify the destination folder • Can select option to overwrite files or folder
Rename Folder	<ul style="list-style-type: none"> • Use to rename a folder • Specify the folder path and name • Specify the new folder name
Delete Folder	<ul style="list-style-type: none"> • Use to delete a folder • Specify the folder path and name
Create Folder	<ul style="list-style-type: none"> • Use to create a file • Specify the new folder path and name • Can select option to overwrite files or folder

Figure 5-22. Folders subcommands (1 of 2)

Folders subcommands (2 of 2)

Files subcommand	Description
Open Folder	<ul style="list-style-type: none">• Use to open a folder• Specify the folder path and name
Create Folder Shortcut	<ul style="list-style-type: none">• Use to create a shortcut that you can use to open a folder• Specify the folder path and name

Figure 5-23. Folders subcommands (2 of 2)

5.4. Exercise command focus: Database commands

Exercise command focus: Database commands

Working with the Smart Recorder and the Object Cloning command

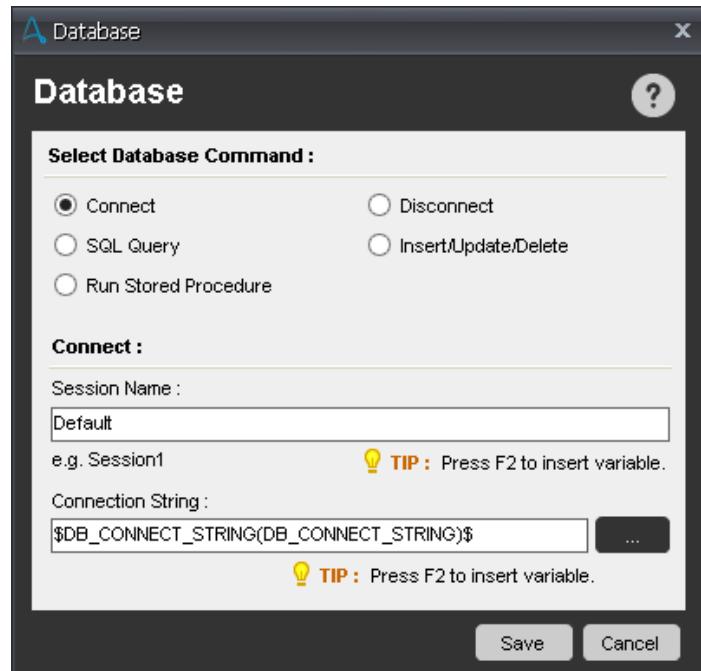
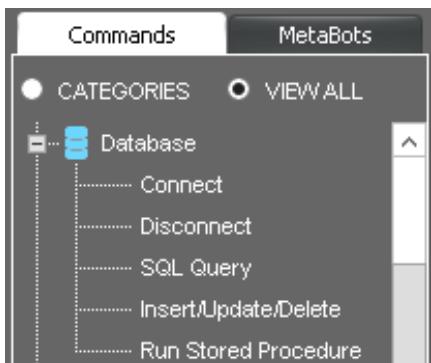
© Copyright IBM Corporation 2018, 2019

Figure 5-24. Exercise command focus: Database commands



Database commands

- Use when you need to integrate database operations in your bot
 - Import or export data in various formats
 - Convert data from one format to another
 - Transfer data from one database to another



Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-25. Database commands

Database subcommands (1 of 3)

Database subcommand	Description
Connect	<ul style="list-style-type: none"> • Use to connect to a database • Specify the database session name • Specify the connection string <ul style="list-style-type: none"> • Can define the connection string manually or use a system variable that is associated with the Control Room Credential Manager <p>To define the connection string manually:</p> <ul style="list-style-type: none"> • Specify the Object Linking and Embedding Database (OLE DB) provider • Define the connection settings <ul style="list-style-type: none"> • Data source name or connection string • User name and password for database server • Initial catalog to use • Optional: Test the database connection to make sure that it is valid

Figure 5-26. Database subcommands (1 of 3)

Database subcommands (2 of 3)

Database subcommand	Description
Disconnect	<ul style="list-style-type: none"> • Use to disconnect from a database • Specify the database session name
SQL Query	<ul style="list-style-type: none"> • Use to retrieve data from a database • Specify the database session name • Specify the SQL SELECT statement • Optional: <ul style="list-style-type: none"> • Specify maximum number of records to fetch • Specify query timeout in seconds • Export the data to a CSV file
Insert/ Update/ Delete	<ul style="list-style-type: none"> • Use to add, update, or delete records to a database • Specify the database session name • Enter the SQL statement for adding, updating, or deleting records • Optional: <ul style="list-style-type: none"> • Specify the query timeout in seconds

Figure 5-27. Database subcommands (2 of 3)

Database subcommands (3 of 3)

Database subcommand	Description
Run Stored Procedure	<ul style="list-style-type: none"> • Use to run a procedure that is stored in your database • Specify the database session name • Specify the procedure name • Can add parameters to the procedure <ul style="list-style-type: none"> • Specify parameter name • Specify parameter type: Input, Output, InputOutput • Can also update or delete existing parameters • Can export data to a CSV file • Optional: <ul style="list-style-type: none"> • Specify the query timeout in seconds

Unit summary

- Describe how to capture bot interactions with the Smart Recorder
- Describe how to capture bot interactions with the Object Cloning command
- Explain how to edit Object Cloning commands in the Workbench

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-29. Unit summary

Review questions



1. When you record user interactions with the Smart Recorder, what type of command are they saved as in the Task Actions List?
 - A. Window Actions commands
 - B. Manage Windows Controls commands
 - C. Object Cloning commands
 - D. It depends on the action that you record
2. True or false. When you use an Object Cloning command, you do not need to set an action to perform on the object.

Review answers



1. When you record user interactions with the Smart Recorder, what type of command are they saved as in the Task Actions List?
 - A. Window Actions commands
 - B. Manage Windows Controls commands
 - C. **Object Cloning commands**
 - D. It depends on the action that you record

The answer is **C.**
2. True or false. When you use an Object Cloning command, you do not need to set an action to perform on the object.

The answer is **False**. All Object Cloning commands must perform an action on the object.

Exercise 4: Automating data entry to a business application and a database

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-32. Exercise 4: Automating data entry to a business application and a database

Exercise objectives



- Use the Smart Recorder and the Object Cloning command to capture interface components and actions
- Work with variables to pass data between different applications
- Build loops to automate repetitive tasks across several different applications
- Use the Object Cloning command and the String Operation command to capture and extract data from a message window

Working with the Smart Recorder and the Object Cloning command

© Copyright IBM Corporation 2018, 2019

Figure 5-33. Exercise objectives

Exercise introduction (1 of 2)

The following commands are used in this exercise:



Command	Purpose
Comment	Insert code annotations.
Database	Connect to the database. Insert data into database. Disconnect from the database.
Excel	Opens the new_customers.xlsx spreadsheet and gets values of cells. Saves and closes the spreadsheet.
If/Else	Evaluate whether the SIB CRM application is already open.
Loop	Loop for each row in an Excel dataset

Exercise introduction (2 of 2)

The following commands are used in this exercise:



Command	Purpose
Object cloning	Sets the text values in the SIB CRM application. Gets value of pop-up dialog box. Click Submit push button in the dialog box.
Open program/file	Open SIB CRM application.
String Operation	Extracts the customer reference number substring from the dialog box.
Variable Operation	Increment row number
Window Actions	Close SIB CRM window.

Unit 6. Working with emails, conditional logic, and triggers

Estimated time

01:00

Overview

In this unit, you learn how to configure schedules and triggers to bots so they run automatically.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Enable a bot to send email
- Explain how to implement conditional logic
- Use the clipboard
- Concatenate variables
- Work with nested loops
- Describe how to use a bot trigger
- Describe how to use debugging

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-1. Unit objectives

Topics

- Sending emails
- Implementing conditional logic
- Pasting and copying from the Clipboard
- Extending the **Variable Operation** command
- Extending the **Loop** command
- Introducing bot triggers
- Introducing debugging

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-2. Topics

6.1. Sending emails

Sending emails

Working with emails, conditional logic, and triggers

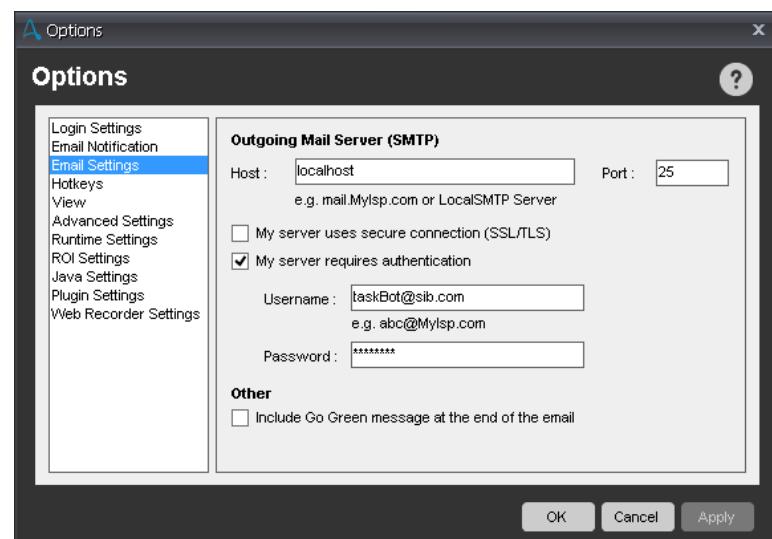
© Copyright IBM Corporation 2018, 2019

Figure 6-3. Sending emails



Sending emails (1 of 2)

- To enable a bot to send email, configure the **Outgoing Mail Server (SMTP)** in the Enterprise client
- Select **Tools > Options > Email Settings** from the menu to configure the email server
 - This setting allows bots to send outgoing emails
- To send email notifications that are related to bot updates, configure **Tools > Options > Email Notification**



Working with emails, conditional logic, and triggers

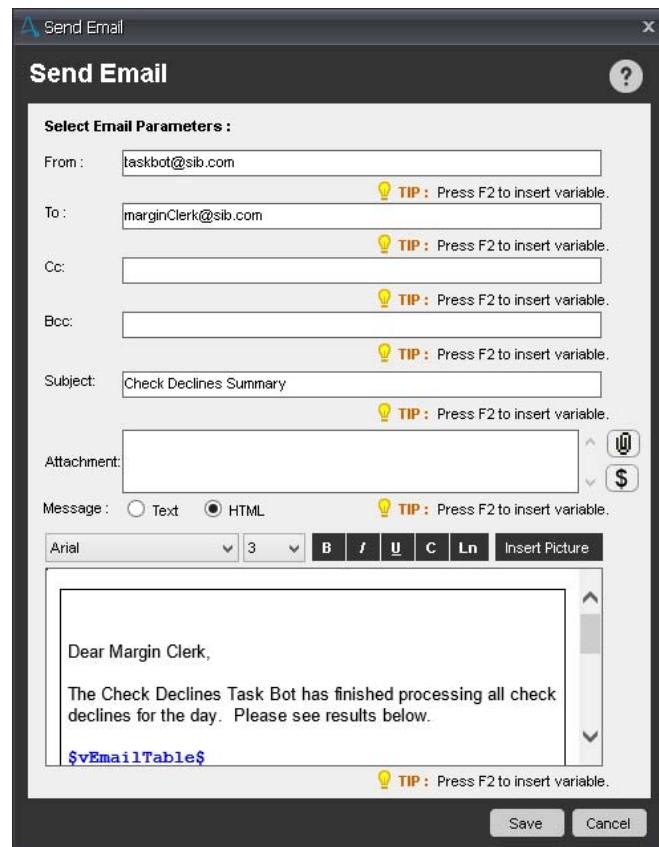
© Copyright IBM Corporation 2018, 2019

Figure 6-4. Sending emails (1 of 2)



Sending emails (2 of 2)

- After the **Outgoing Mail Server (SMTP)** is configured in the Enterprise client, use the **Send Email** command to code bots to send emails
- The **Send Email** command allows the bot to:
 - Send multiple emails at one time; use a semi-colon to separate emails
 - Include attachments with the email
 - Format the message in HTML
- Any value can be hardcoded or a variable
- Variables can be included in the body of the email



Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-5. Sending emails (2 of 2)

6.2. Implementing conditional logic

Implementing conditional logic

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-6. Implementing conditional logic

Working with conditions (1 of 2)

- Conditional processing can be used for many things, including:
 - Evaluation whether the end of a list was reached
 - Determine whether a variable is equal to another variable
 - Determine whether a Window is open
 - Determine whether a file exists, and so on
- Conditional processing and looping are often used together to determine what data is pulled and then posted
- Use the **If/Else** command to implement conditional processing
 - If/Else** command structure: **If <Condition> Then, Else If, End If**

```

IF   vCheckDeclines($Counter$.1)$ Not Equal To (<>) $vCheckDeclines($vNextAcctNum$.1): Then
    Comment: If this is the last check for the account: - Add final check value - Append to result set - Increment Account
    $vCheckSum($vAcctRow$.1)$ To $vCheckSum($vAcctRow$.1)$
    Variable Operation: $vCheckSum($vAcctRow$.1)$ | $vCheckSum($vAcctRow$.2)$ To $vResultSet($vAcctRow$)$
    $vAcctRow$ + 1 To $vAcctRow$
    Variable Operation: $vTotalAccts$ + 1 To $vTotalAccts$
    Variable Operation: $vTotalAccts$ + 1 To $vTotalAccts$
End If
  
```

Figure 6-7. Working with conditions (1 of 2)

Working with conditions (2 of 2)

- **If/Else** commands can be nested
- Use **Else** or **Else If** subcommands to enable more complex conditional processing
- When **If/Else** evaluation returns true
 - The commands immediately following the **If/Else** evaluation are run
- When **If/Else** evaluation returns false
 - The commands immediately following the **End If** command are run

```

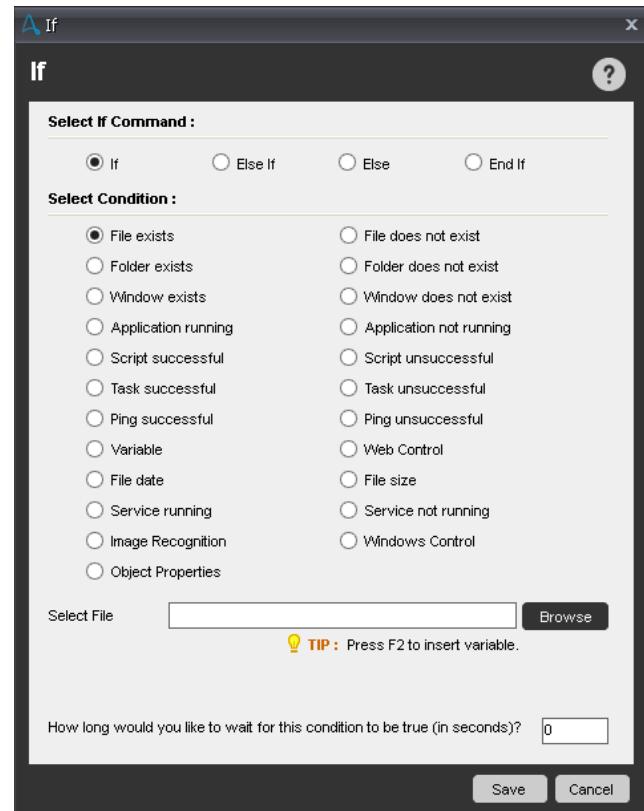
IF      If $vCheckDeclines($Counter$.1)$ Not Equal To (<>) $vCheckDeclines($vNextAcctNum$.1)$ Then
      Comment: If this is the last check for the account: - Add final check value - Append to result set - Increment Account
      Variable Operation: $vCheckDeclines($Counter$.1)$ To $vCheckSum($vAcctRow$. 1)$
      Variable Operation: $vCheckSum($vAcctRow$.1)$ | $vCheckSum($vAcctRow$.2)$ To $vResultSet($vAcctRow$)$
      Variable Operation: $vAcctRow$ + 1 To $vAcctRow$
      Variable Operation: $vTotalAccts$ + 1 To $vTotalAccts$
      End If
  
```

Figure 6-8. Working with conditions (2 of 2)



If/Else Command (1 of 2)

- The **If/Else** command allows the bot to run specific commands when certain conditions exist
- When using many of the subcommands, you can specify how long to wait for a condition to exist before taking another action
- If you use the default wait time of 0, the condition is checked immediately and actions ran
 - If the condition does not exist, the actions are skipped



Working with emails, conditional logic, and triggers

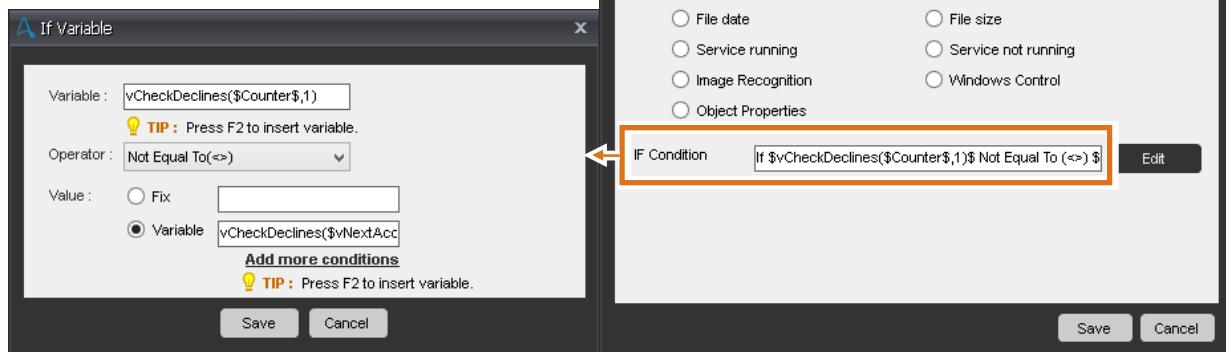
© Copyright IBM Corporation 2018, 2019

Figure 6-9. If/Else Command (1 of 2)



If/Else Command (2 of 2)

- The **Variable** subcommand is one of the most common conditional commands
- This command is used to evaluate the value of a variable and run commands based on the result of that evaluation



Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-10. If/Else Command (2 of 2)

6.3. Pasting and copying from the Clipboard

Pasting and copying from the Clipboard

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-11. Pasting and copying from the Clipboard

Pasting and copying from the Clipboard

- Sometimes it makes sense to paste and copy from the clipboard
- For example, if the bot is building a result set that cannot be held in a variable, the bot can paste the results to Notepad for each row in a result set.
 - After all rows are pasted, it can then copy it to a variable
- The **Clipboard** command is used to clear, assign to, or assign from the clipboard
- The system variable, named `$Clipboard$`, can be used to retrieve the contents of the clipboard
- If you copy or paste from the clipboard while within a loop, the clipboard needs to be cleared before each iteration in case the data copied is empty

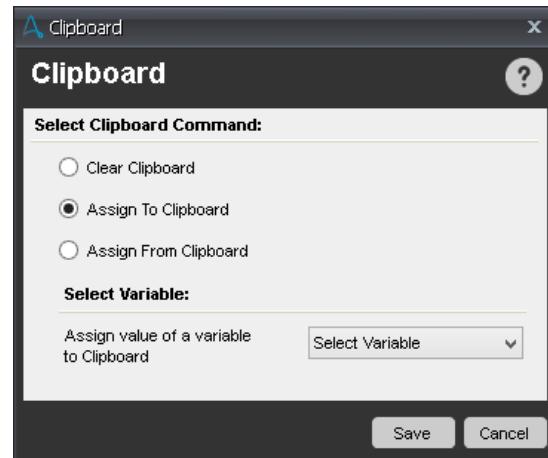


Figure 6-12. Pasting and copying from the Clipboard

6.4. Extending the Variable Operation command

Extending the Variable Operation command

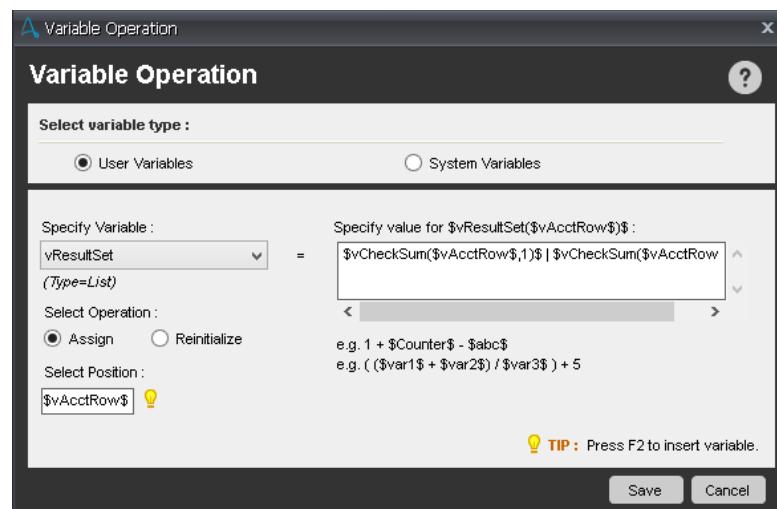
Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-13. Extending the Variable Operation command

Extending the Variable Operation command (1 of 2)

- When you use an array with a **Loop** command, the position within the array must be known to use it. The position can be a variable or fixed
- Concatenation can be done by using the **Variable Operation** command
 - Each variable is enclosed by \$
 - Other characters within the specified value are taken literally and concatenated to the variables
- Example:**
“\$Variable1\$ -
\$Variable2\$”
yields the result:
“Variable1Value -
Variable2Value”



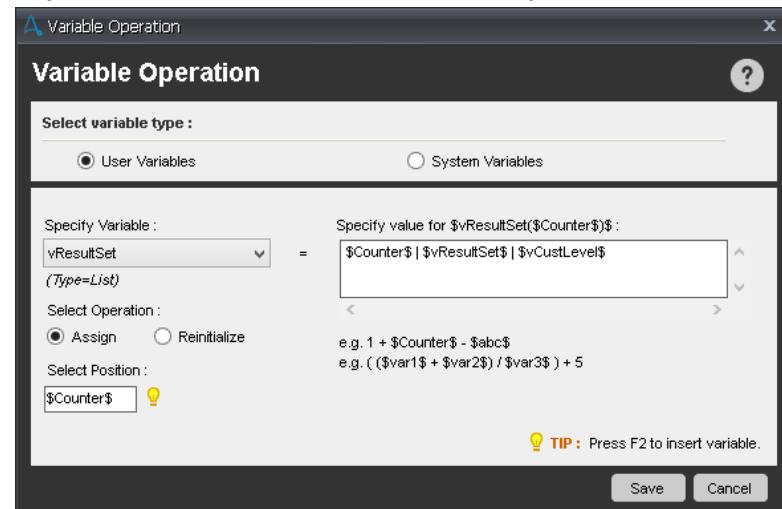
Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-14. Extending the Variable Operation command (1 of 2)

Extending the Variable Operation command (2 of 2)

- More complex concatenation when you combine values from disparate sources might require multiple concatenations to be completed
- A result set can be built then concatenated to another result set
 - If the original result set is not reused, the variable can be reused



- Example:
 - The resulting value from the concatenation:
\$Counter\$ | \$vResultSet\$ | \$vCustLevel\$ yields the result:
“1 | 1234 | 80000 | Silver”

Figure 6-15. Extending the Variable Operation command (2 of 2)

6.5. Extending the Loop command

Extending the Loop command

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-16. Extending the Loop command

Extending the Loop command with nested looping

- Looping enables the use of system variables that would not otherwise be available when you pull data from source systems, including (but not limited to):
 - Counter
 - Excel Cell
 - Dataset Column
- Sometimes looping commands are nested
 - Nested looping might need to be done to access system variables not available in the parent loop
 - For example: A nested loop can be placed in a parent loop to provide access to system variables such as:
\$Dataset Column (Counter\$) \$
 - Then, the bot can access the variables in the parent loop along with the values made available in the child loop
- Nested looping might be needed when you concatenate values together to form a result set taken from multiple sources

6.6. Introducing bot triggers

Introducing bot triggers

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-18. Introducing bot triggers



Introduction to bot triggers (1 of 4)

- Configure a bot trigger on the **Trigger** tab of the **Task relevant activities panel** in the Enterprise client
- Types of triggers:

- Window
- File
- Folder
- Performance
- Process
- Service
- Email
- Message

The screenshot shows the Automation Anywhere Enterprise Client interface. On the left, there's a sidebar with sections for AUTOMATE (Tasks, My Tasks, Sample Tasks, My Execs, My Scripts, My Docs), MANAGE, and DEFINE. The main area is titled "My Tasks" and lists a single task named "dtest - working.atmx". Below this is a table with columns for File Name, Type, Repeat, and Status. The "File Name" row shows "dtest - working.atmx", "Task file", "Do not Repeat", and "Complete". To the right of the table are buttons for Delete, Upload, and Actions. At the bottom of the table are "Search Help...", a magnifying glass icon, and the "AUTOMATION ANYWHERE Enterprise" logo.

The central part of the screen shows the properties of the task. It has tabs for PROPERTIES, SCHEDULE, and TRIGGER. The TRIGGER tab is currently selected and highlighted with an orange box. Under the TRIGGER tab, there are several sections: General, File Name (dtest - working.atmx), Created at (10/10/2017 11:15:34), Status (Complete), Last Run Time (10/10/2017 11:16:11), View Log, Hotkey, Priority (Low), Security, Task Report (Mouse Clicks 0, Keystrokes 0, Total Clicks 0), and a checkbox for "Enable this task to run with other similar files or window titles". There's also a "Description / Notes" section with a text input field.

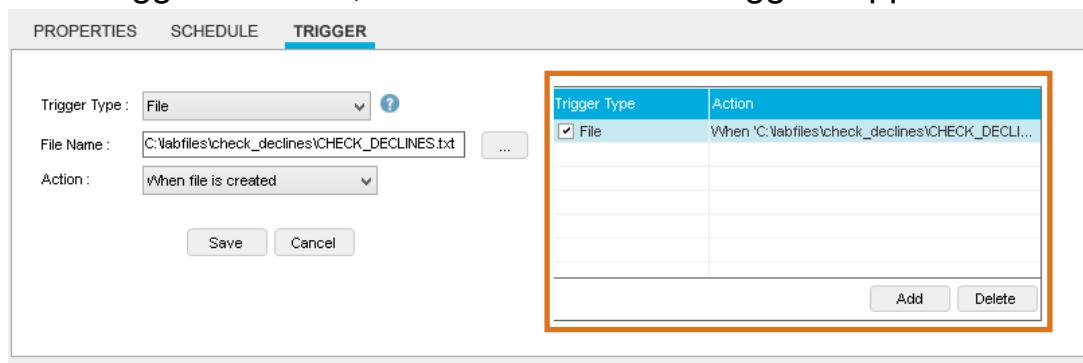
Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-19. Introduction to bot triggers (1 of 4)

Introduction to bot triggers (2 of 4)

- After you define the trigger type, you can configure the action to be taken
- For example, when you use the **File** trigger type, you can select one of the following actions:
 - When file is created
 - When file is deleted
 - When file is renamed
 - When file is modified
- When the trigger is saved, a table of all defined triggers appears



Trigger Type	Action
File	When file is created

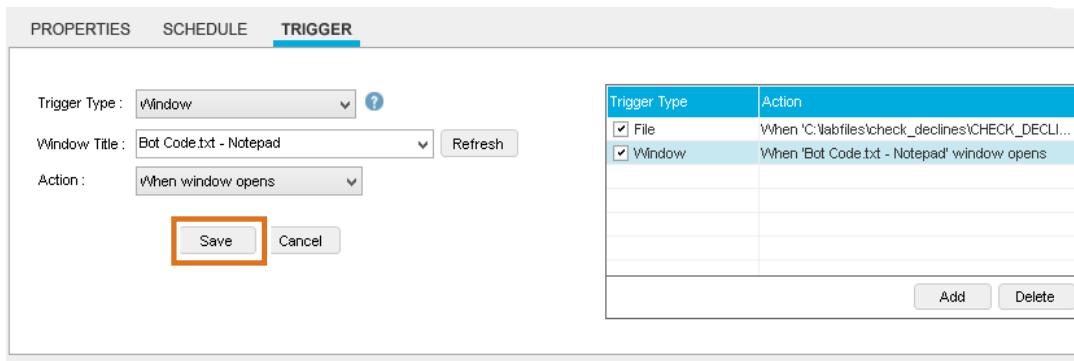
Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-20. Introduction to bot triggers (2 of 4)

Introduction to bot triggers (3 of 4)

- Bots can have multiple triggers
- You can enable or disable existing triggers by checking or clearing the trigger in the Trigger table
- The portion to the left in the Trigger pane is for configuring new triggers
- The table on the right exists only if there is at least one trigger that is configured, whether it is enabled or disabled
- To save the trigger after configuration, click **Save**



Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-21. Introduction to bot triggers (3 of 4)

Introduction to bot triggers (4 of 4)

- Bot triggers do not work on computers when it is in a logged off state. If the computer is in a locked state, triggers can work if auto-login is enabled
- Triggers apply to the local instance only of the Client
- Triggers are not uploaded with the task to the Control Room and thus cannot be deployed
- Bots and triggers have a many-to-many relationship
 - You can have one trigger that triggers several task bots or you can have one task bot with multiple triggers
- You can manage all triggers by using the Trigger Manager in the Control Room

6.7. Introducing debugging

Introducing debugging

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-23. Introducing debugging



Debugging (1 of 2)

- In the Workbench, a debugging feature can be used to help correct issues with a bot



- Debugging can be especially useful for larger, more complex bots
- Debugging is toggled on or off
 - Click **Disable Debugging** to close the debugging facility
- Features of debugging include:
 - Variable Watch Table**
 - Displays current variable values while the bot is running
 - Breakpoints**
 - Set breakpoints to have the bot pause during execution
 - Use the **Step Over (F10)** function to continue processing
 - Snap points**
 - Capture images during bot execution

Variable(s) Watch Table	
Name	Value
vAcctRow	4
vCheckDeclines(10...	3456
vChecksum(3,1)	3456
vResultSet	1 1234 80000 Silver
vTotalAccts	3
<input type="checkbox"/> Select All	
<input type="button" value="Add"/> <input type="button" value="Remove"/>	

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-24. Debugging (1 of 2)

Debugging (2 of 2)

- While in debug mode, each line of bot code is highlighted in the Workbench as it is executed one line at a time

```

1  Comment: Pull check values from file and add to running total for each account then add to result set.
2  Read From CSV file: "C:\labfiles\check_declines\CHECK_DECLINES.csv" Delimiter: "Comma" Header: "No" Trim Leading $| 
3  Start Loop "Each row in a CSV/Text file of Session: Default"
4  Comment: Add check values to running total.
5  Variable Operation: $vCheckSum($vAcctRow$.2$) + $vCheckDeclines($Counter$.2$) To $vCheckSum($vAcctRow$.2$)
6  If $vCheckDeclines($Counter$.1$) Not Equal To (<>) $vCheckDeclines($vNextAcctNum$.1$) Then
7    Comment: If this is the last check for the account: - Add final check value - Append to result set - Increment Account nu
8    Variable Operation: $vCheckDeclines($Counter$.1$) To $vCheckSum($vAcctRow$.1$)
9    Variable Operation: $vCheckSum($vAcctRow$.1$) | $vCheckSum($vAcctRow$.2$) To $vResultSet($vAcctRow$.2$)
10   Variable Operation: $vAcctRow$ + 1 To $vAcctRow$
11   Variable Operation: $vTotalAccts$ + 1 To $vTotalAccts$
12 End If
13 Comment: Increment account number
14 Variable Operation: $vNextAcctNum$ + 1 To $vNextAcctNum$
15 End Loop

```

- Debugging is CPU intensive and can lead to performance issues
 - If you experience performance problems when you run the task in debug mode, try running the task without using debug mode to verify that the bot has no errors

Figure 6-25. Debugging (2 of 2)

Unit summary

- Enable a bot to send email
- Explain how to implement conditional logic
- Use the clipboard
- Concatenate variables
- Work with nested loops
- Describe how to use a bot trigger
- Describe how to use debugging

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-26. Unit summary

Review questions (1 of 2)



1. True or False: Using the following code snippet, lines 13 and 14 are run only if the **If <Condition> Then** is evaluated to true.

```

6   IF      If $vCheckDeclines($Counter$,1)$ Not Equal To [<>] $vCheckDeclines($vNextAcctNum$,1)$ Then
7   |       Comment: If this is the last check for the account - Add final check value - Append to result set - I
8   |       Variable Operation: $vCheckDeclines($Counter$,1)$ To $vCheckSum($vAcctRow$, 1)$
9   |       Variable Operation: $vCheckSum($vAcctRow$,1)$ | $vCheckSum($vAcctRow$,2)$ To $vResult$E
10  |      Variable Operation: $vAcctRow$ + 1 To $vAcctRow$E
11  |      Variable Operation: $vTotalAccts$ + 1 To $vTotalAccts$E
12  |      End If
13  |      Comment: Increment account number.
14  |      Variable Operation: $vNextAcctNum$ + 1 To $vNextAcctNum$E
15  |      End Loop

```

2. True or False: The **Send Email** command can only be used if an outgoing mail server has been configured.
3. True or False: The **Variable Operation** command can be used for simple concatenation.
4. True or False: Some system variables that hold data that is retrieved from a source are only available within a loop.

Figure 6-27. Review questions (1 of 2)

Review questions (2 of 2)

5. Choose the best answer for the following scenario:



A bot Developer is creating a bot and building a result set for each record in a loop. The bot Developer needs to query a database to add database values to the result set.

What would be the best approach to solve this issue?

- A. Directly access the **\$Dataset Column\$** system variable after querying the database.
- B. Query the database first. Then, include a nested loop that accesses the **\$Dataset Column\$** system variable for each time the parent loop completes.
- C. Use the **Variable Operation** command to parse the results from the database query.
- D. None of the above.

Review answers (1 of 2)

1. True or False: Using the following code snippet, lines 13 and 14 are run only if the **If <Condition> Then** is evaluated to true.

The answer is False.



2. True or False: The **Send Email** command can only be used if an outgoing mail server has been configured.

The answer is True.

3. True or False: The **Variable Operation** command can be used for simple concatenation.

The answer is True.

4. True or False: Some system variables that hold data that is retrieved from a source are only available within a loop.

The answer is True.

Review answers (2 of 2)



5. Choose the best answer for the following scenario:

A bot Developer is creating a bot and building a result set for each record in a loop. The bot Developer needs to query a database to add database values to the result set.

What would be the best approach to solve this issue?

- A. Directly access the **\$Dataset Column\$** system variable after querying the database.
- B. Query the database first. Then, include a nested loop that accesses the **\$Dataset Column\$** system variable for each time the parent loop completes.
- C. Use the **Variable Operation** command to parse the results from the database query.
- D. None of the above.

The answer is B.

Exercise 5: Creating a bot to sum check declines, query a database, and send an email

Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-31. Exercise 5: Creating a bot to sum check declines, query a database, and send an email

Exercise objectives

- Extract values from files
- Use variables to perform calculations
- Work with the Email command
- Configure a bot trigger

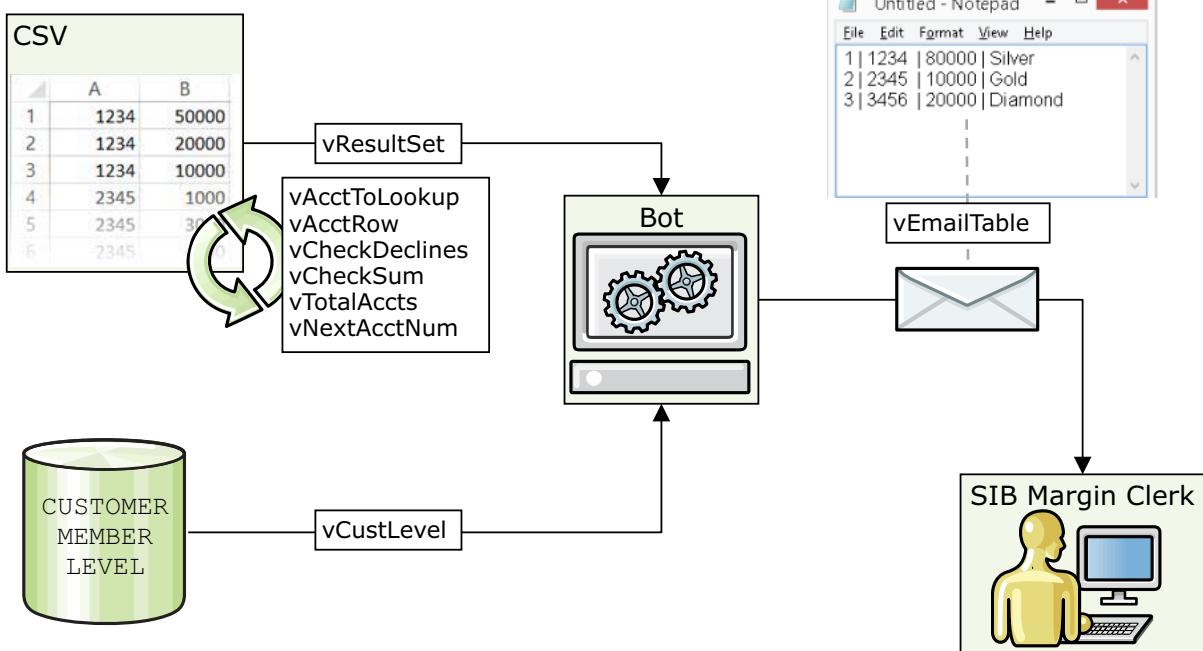
Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-32. Exercise objectives

Exercise introduction (1 of 7)

In this exercise, you play the role of a Bot Developer to create a Task bot to perform the following steps



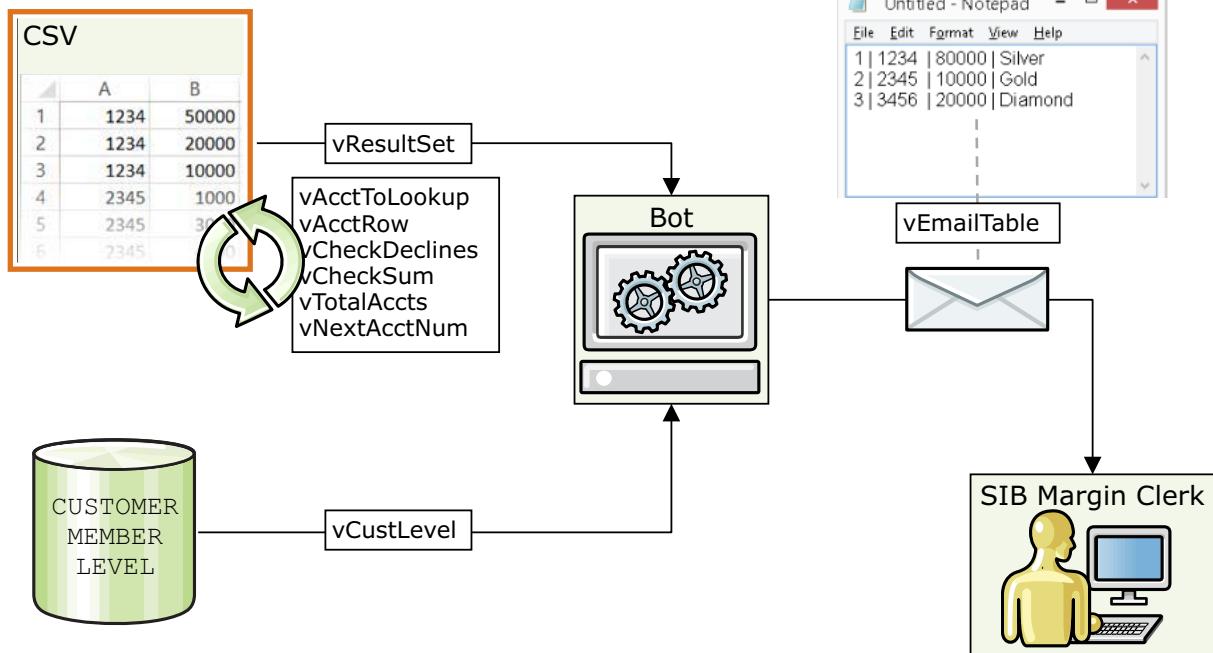
Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-33. Exercise introduction (1 of 7)

Exercise introduction (2 of 7)

- Pull check decline values from a CSV file and sum for each account



Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

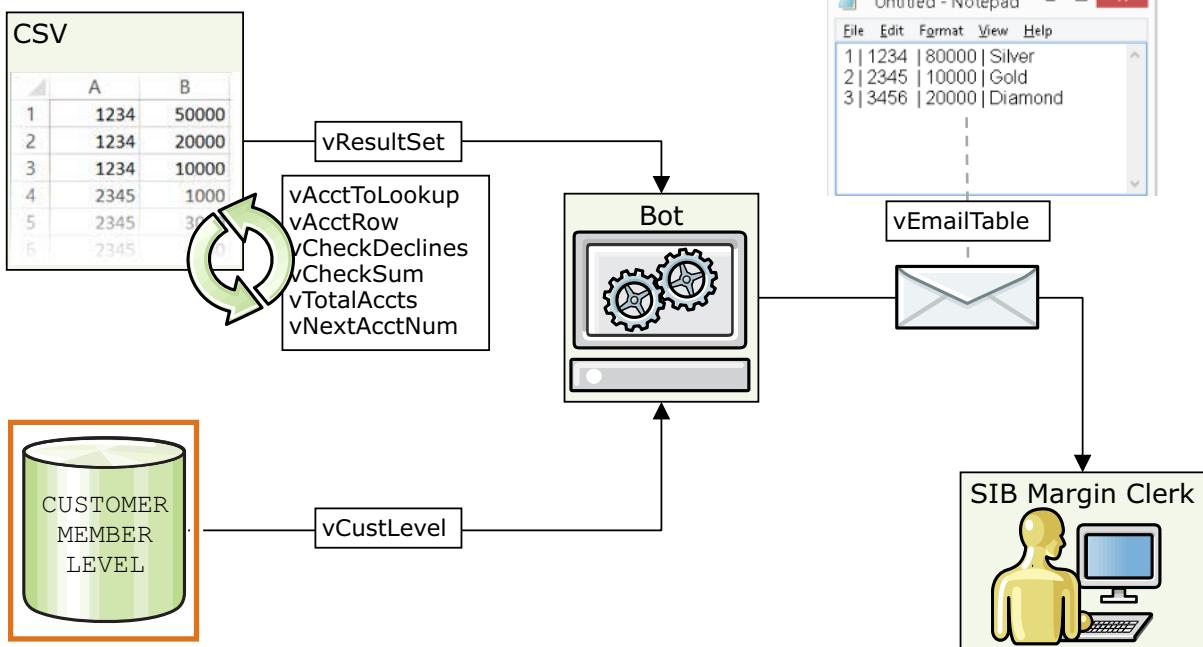
Figure 6-34. Exercise introduction (2 of 7)

IBM Training



Exercise introduction (3 of 7)

- Pull customer membership level from a database for each account



Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

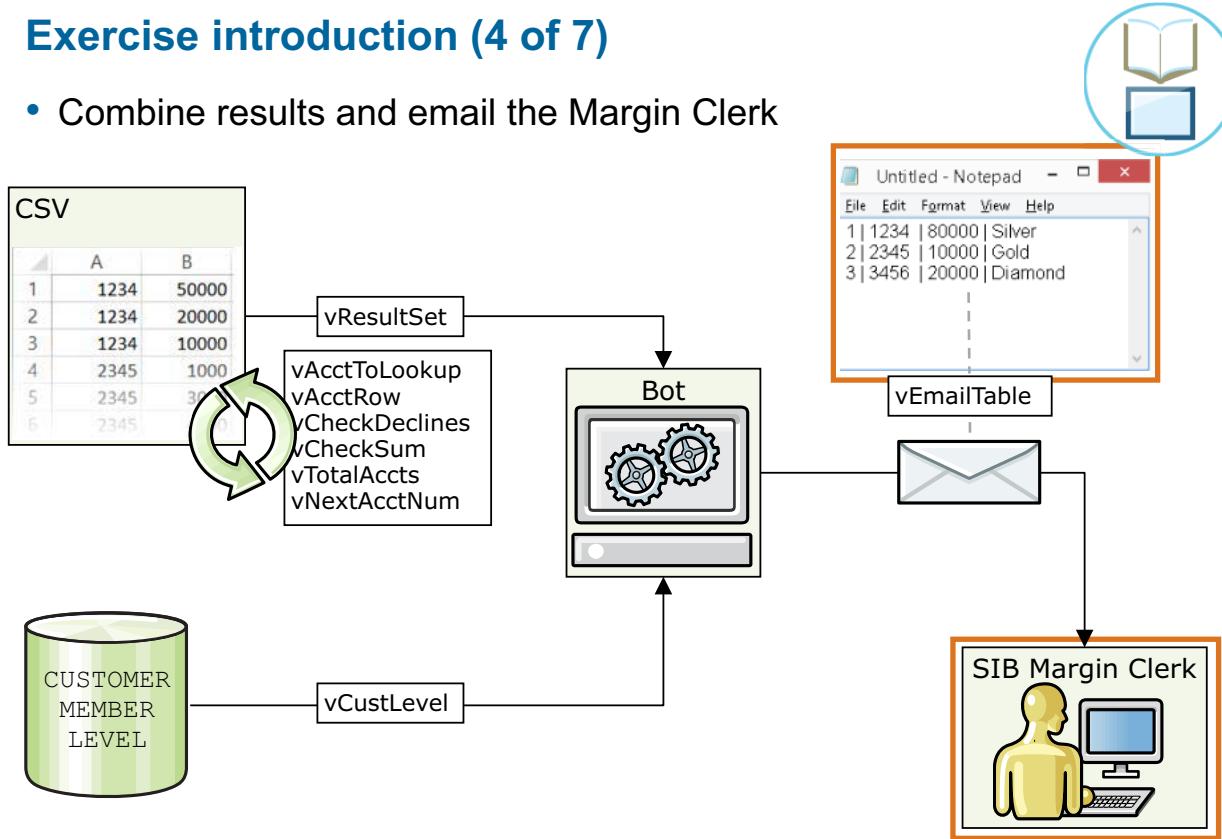
Figure 6-35. Exercise introduction (3 of 7)



IBM

Exercise introduction (4 of 7)

- Combine results and email the Margin Clerk



Working with emails, conditional logic, and triggers

© Copyright IBM Corporation 2018, 2019

Figure 6-36. Exercise introduction (4 of 7)

Exercise introduction (5 of 7)



- To build the bot, you create a set of variables, develop bot code, and verify correct functioning of the bot
- During the building of the bot, you:
 - Import a CSV file and assign variables
 - Use system variables and define custom variables
 - Access values in a two-dimensional array
 - Retrieve a record from a database that uses a parameter
 - Implement nested looping
 - Use three types of looping:
 - Each row in a CSV/Text file
 - Each row in SQL data set Session
 - Loop <variable> times
 - Perform string concatenation by using the **Variable Operation** command
 - Use the **Variable Operation** to sum by account
 - Use the clipboard to house a text table
 - Perform conditional processing by using an **If\Else** statement
- You also set a trigger on the bot to run when the file arrives

Exercise introduction (6 of 7)

The following commands are used in this exercise:



Command	Purpose
Clipboard	Assign content from the clipboard to a variable
Comment	Insert code annotations
Database	Connect to the database Query the database by using Account Number Disconnect from the database
If/Else	Evaluate whether the bot reached the last check decline for the current Account Number
Insert keystrokes	Insert keystrokes into Notepad
Loop	Loop for each row in a CSV/Text file Loop for each row in an SQL query data set Loop <n> times

Exercise introduction (7 of 7)

The following commands are used in this exercise:



Command	Purpose
Object cloning	This command is used to interact with the dialog box when closing Notepad
Open program/file	Open Notepad.
Read From CSV/Text	Read from the text file
Send email	Send an email to the Margin Clerk after processing the check declines
Variable Operation	Assign values to variables
Window Actions	Close Notepad window

Unit 7. Working with email and PDFs

Estimated time

01:00

Overview

This unit covers how to implement conditional (if/else) logic in a bot to handle different situations.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Download an attachment from an email
- Extract data from a PDF file
- Work with extracted text strings
- Determine the length and value of variables

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-1. Unit objectives

Topics

- Downloading attachments from email
- Extracting data from a PDF document
- Extending the String Operation command to trim variables
- Extending the String Operation command to get length of variables

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-2. Topics

7.1. Downloading attachments from email

Downloading attachments from email

Working with email and PDFs

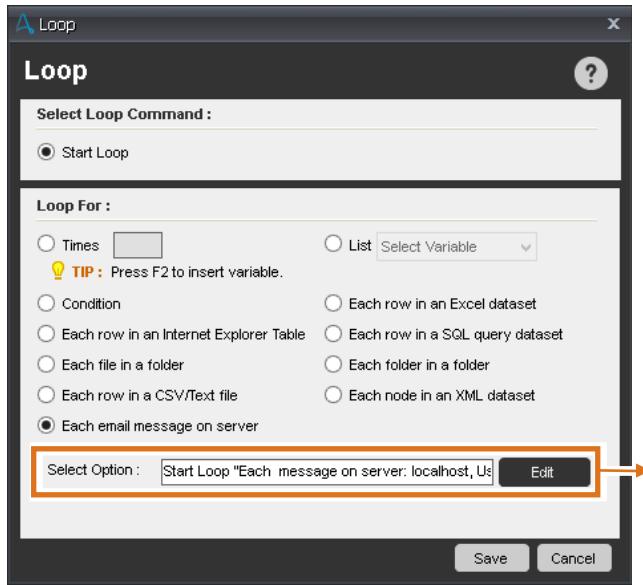
© Copyright IBM Corporation 2018, 2019

Figure 7-3. Downloading attachments from email

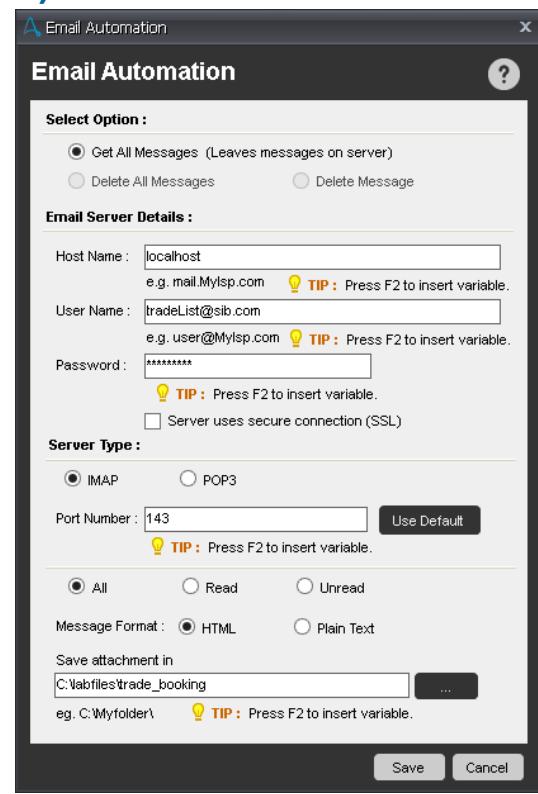


Email Automation command (1 of 3)

- Use the **Email Automation** command to download email attachments from an email inbox while within a loop



Working with email and PDFs



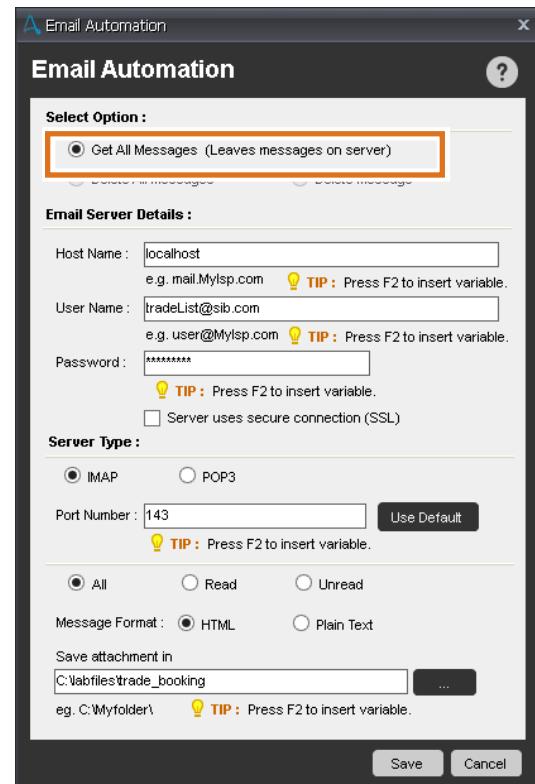
© Copyright IBM Corporation 2018, 2019

Figure 7-4. Email Automation command (1 of 3)

Email Automation command (2 of 3)

- The **Email Automation** command provides the subcommands:
 - Get All Messages
 - Delete All Messages
 - Delete Message
- With the **Email Automation** command, bots can:
 - Delete email messages
 - Monitor email activity
 - Extract email data (such as Subject, Message, and attachments) to other applications or folders
 - Organize email based on status when downloading them
- An email server must be configured before you use the **Email Automation** command

[Working with email and PDFs](#)

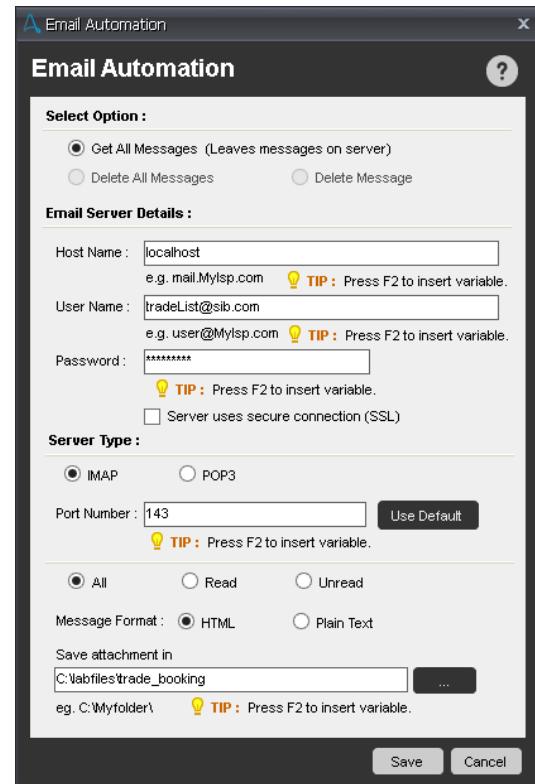


© Copyright IBM Corporation 2018, 2019

Figure 7-5. Email Automation command (2 of 3)

Email Automation command (3 of 3)

- To configure the **Email Automation** command, you provide:
 - Host name
 - User name
 - Password
 - SSL connection (optional)
 - Server type
 - Port number
 - Scope (All, Read, Unread)
 - Message format
 - Save attachment location (optional)
- All values can be a parameter (variable)
- Credentials from the Control Room can be used for the **User Name** and **Password**
- Ensure that the email server has timeout that is configured to keep the user from being logged out during email processing



© Copyright IBM Corporation 2018, 2019

Working with email and PDFs

Figure 7-6. Email Automation command (3 of 3)

7.2. Extracting data from a PDF document

Extracting data from a PDF document

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-7. Extracting data from a PDF document

PDF Integration command

- The **PDF Integration** command supports the extraction of data from a PDF document
- The **PDF Integration** command supports the following subcommands:
 - PDF to Image
 - Extract Form Fields
 - Extract Text
 - Merge Documents
 - Split Document
 - Encrypt Document
 - Decrypt Document
- Two methods to extract data from a PDF:
 - Extract Text
 - Extract Form Fields
 - Two methods for extracting Form Fields

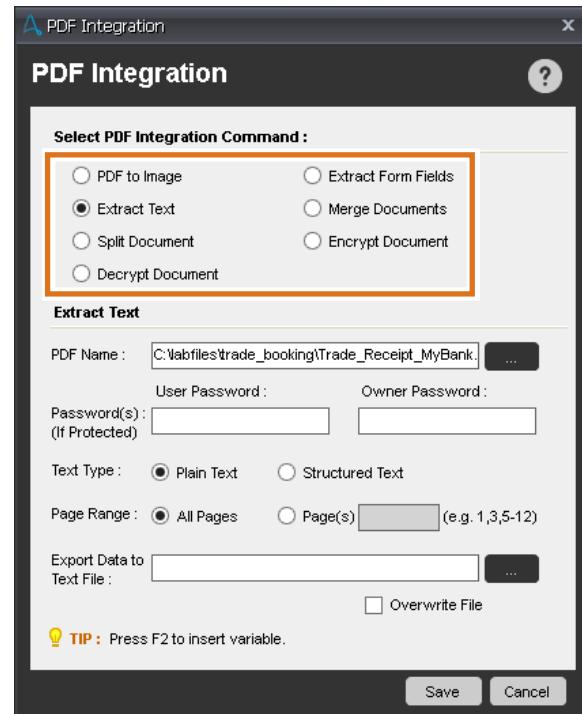
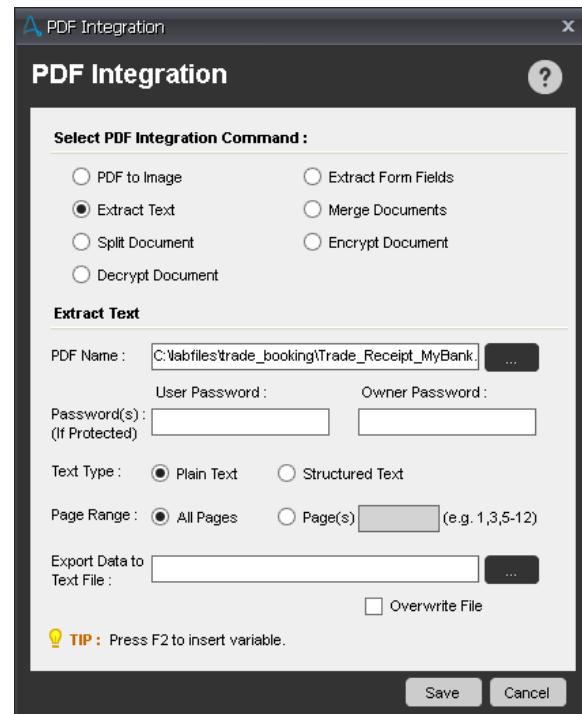


Figure 7-8. PDF Integration command

PDF Integration command: Extract Text

- The **Extract Text** subcommand of the **PDF Integration** command extracts all text from a PDF and pastes it into a text file
- You can configure the extract for a specific page range
- You can configure whether text formatting (spacing) is included in the export by selecting **Structured Text** as the **Text Type**
- Note, the extracted text is not in a variable and more data manipulation is needed to use the data



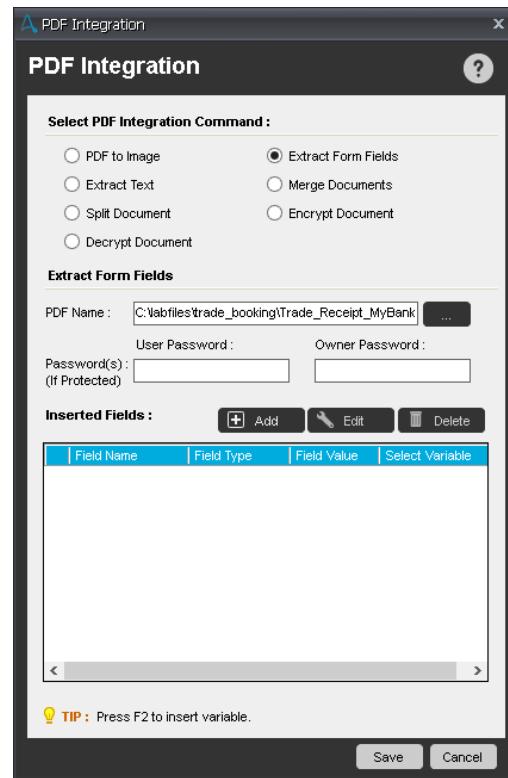
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-9. PDF Integration command: Extract Text

PDF Integration command – Extract form fields (1 of 3)

- Two methods to extract form fields from a PDF
 - The methods vary based on the type of data in the PDF
 - In either case, you need to define the area of the PDF to be extracted for each field
- For PDFs with form fields, the fields are found with red dashes.
- For PDFs with static text, the field areas need to be highlighted manually
- When **Add** is clicked, a **PDF Viewer** window appears to allow the highlighting of fields to be extracted



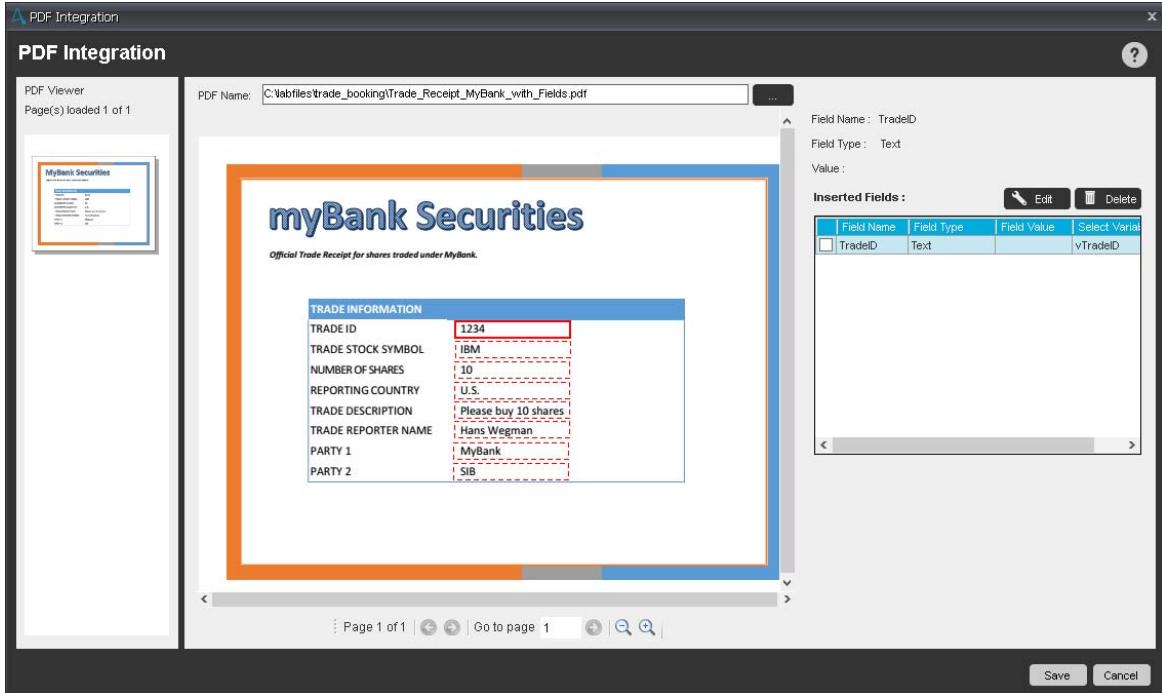
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-10. PDF Integration command – Extract form fields (1 of 3)

PDF Integration command – Extract form fields (2 of 3)

- PDF Viewer window with form fields highlighted



Working with email and PDFs

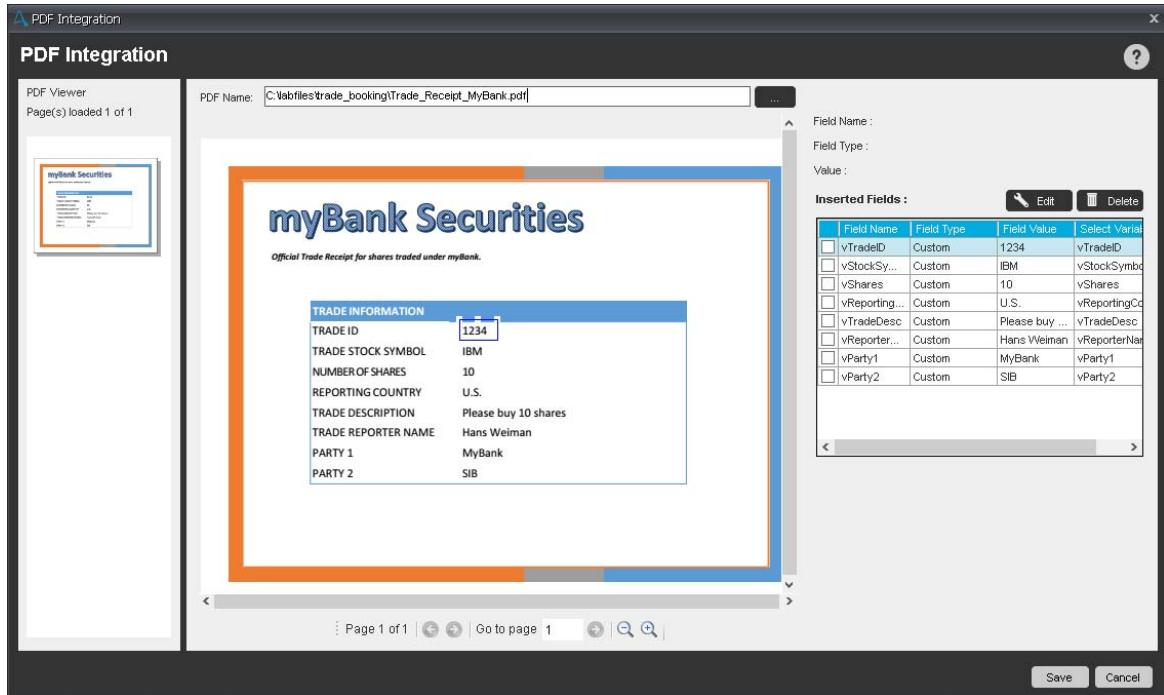
© Copyright IBM Corporation 2018, 2019

Figure 7-11. PDF Integration command – Extract form fields (2 of 3)



PDF Integration command – Extract form fields (3 of 3)

- PDF Viewer window with static data



Working with email and PDFs

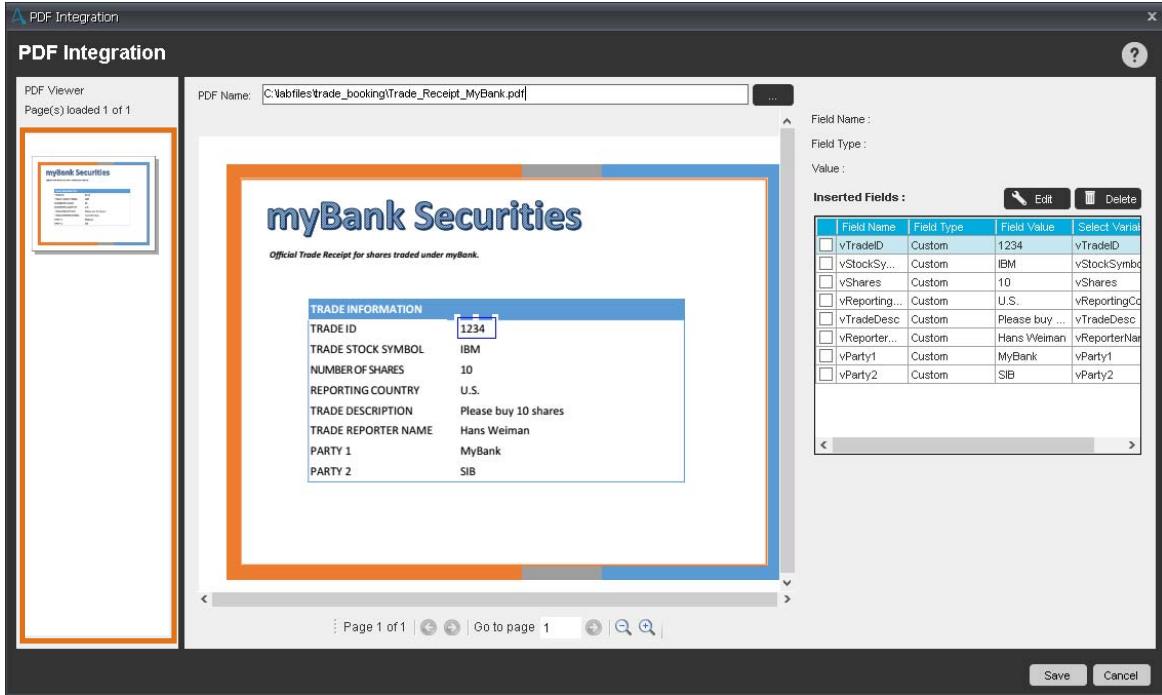
© Copyright IBM Corporation 2018, 2019

Figure 7-12. PDF Integration command – Extract form fields (3 of 3)



PDF Integration command – PDF Viewer (1 of 5)

- PDF Viewer: all pages in the PDF appear here



Working with email and PDFs

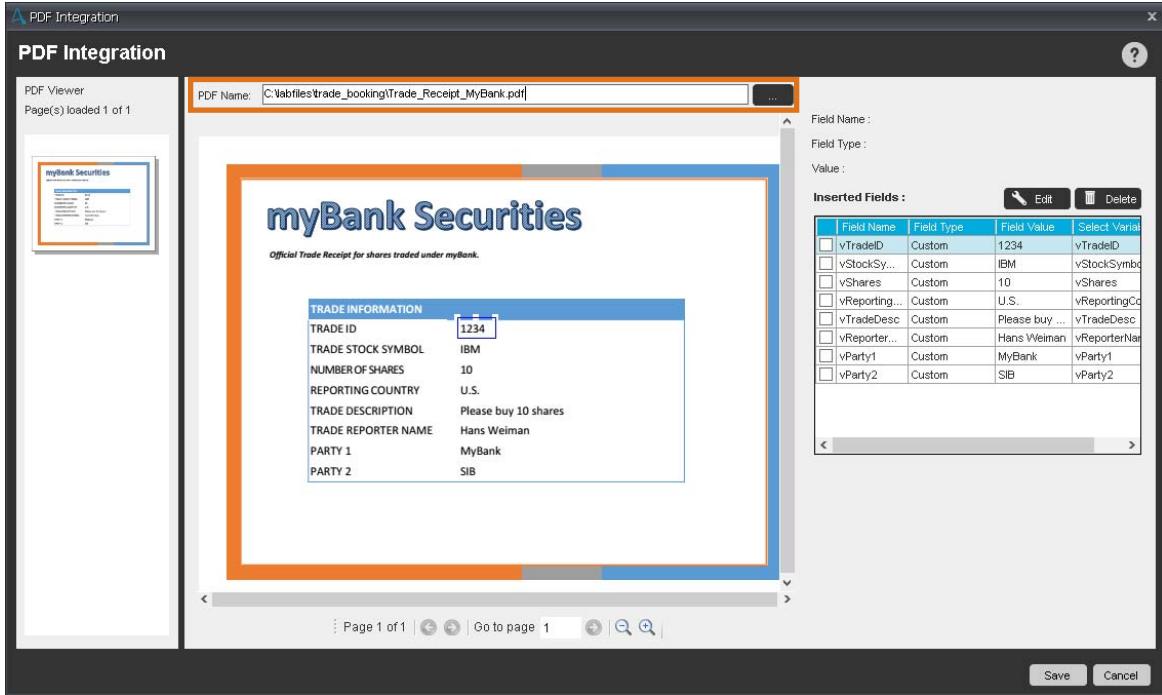
© Copyright IBM Corporation 2018, 2019

Figure 7-13. PDF Integration command – PDF Viewer (1 of 5)



PDF Integration command – PDF Viewer (2 of 5)

- PDF Name: Location of the PDF on the computer



Working with email and PDFs

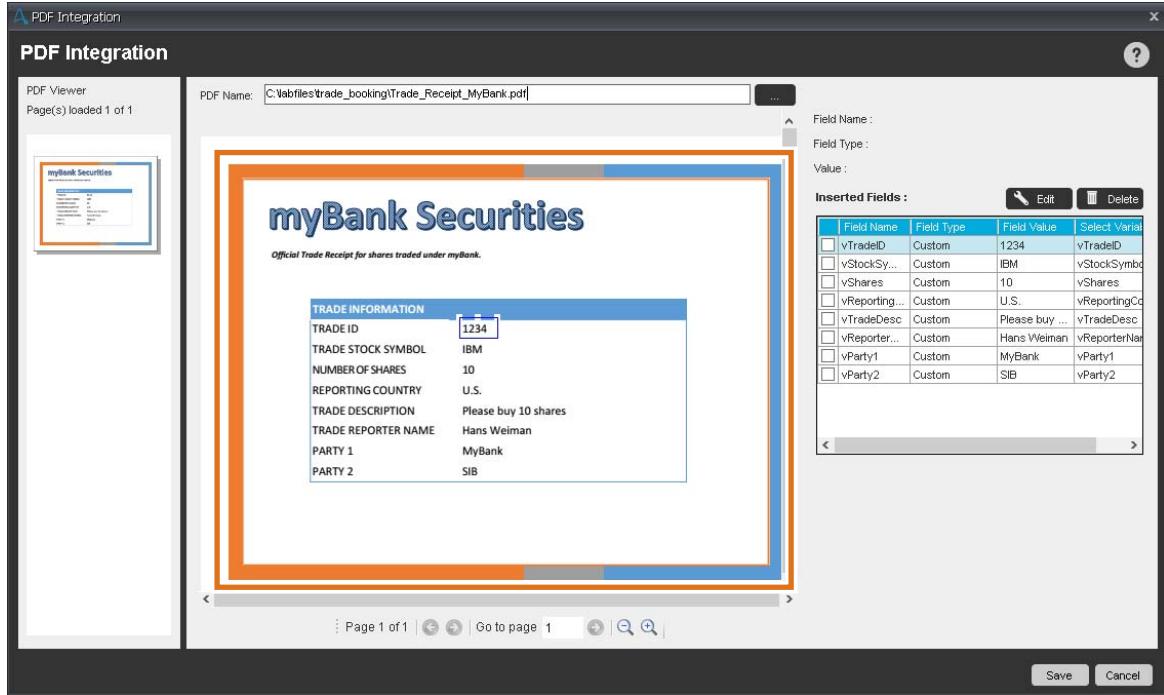
© Copyright IBM Corporation 2018, 2019

Figure 7-14. PDF Integration command – PDF Viewer (2 of 5)



PDF Integration command – PDF Viewer (3 of 5)

- PDF Viewer: Highlight area, where all fields are outlined



Working with email and PDFs

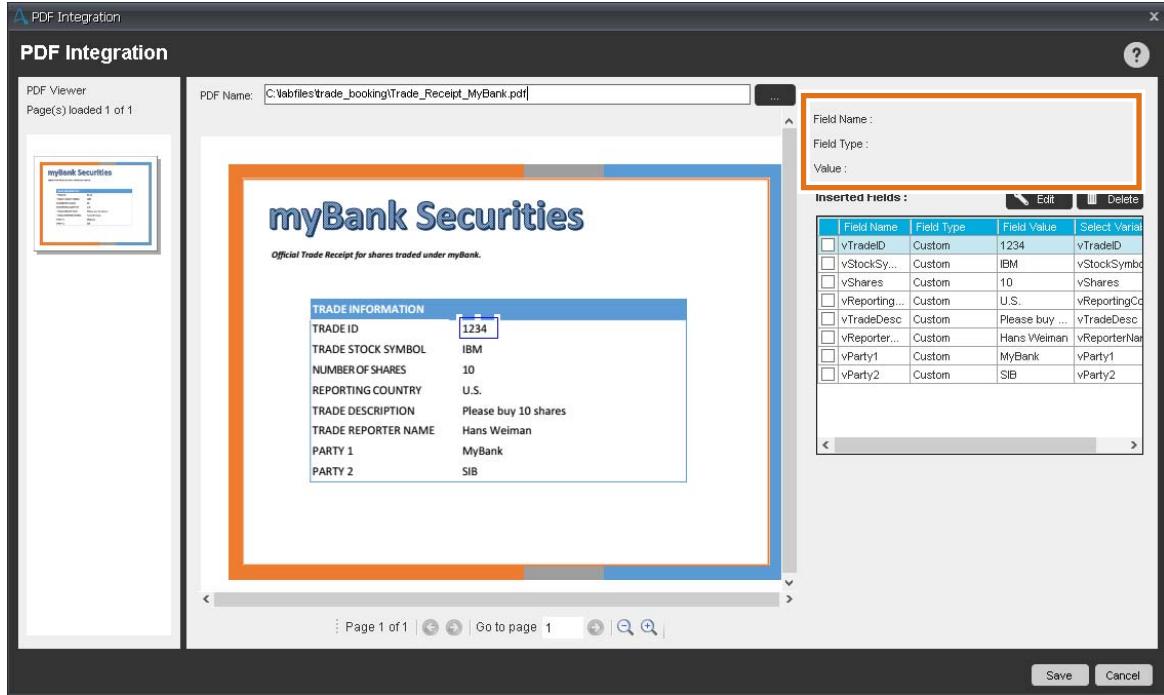
© Copyright IBM Corporation 2018, 2019

Figure 7-15. PDF Integration command – PDF Viewer (3 of 5)



PDF Integration command – PDF Viewer (4 of 5)

- PDF Viewer: Field name of extracted form field



Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-16. PDF Integration command – PDF Viewer (4 of 5)



PDF Integration command – PDF Viewer (5 of 5)

- Inserted Fields: List of all fields that are configured and the variables assigned

The screenshot shows the 'PDF Integration' application window. On the left, there's a preview pane showing a trade receipt from 'myBank Securities'. The receipt has a blue header and contains sections like 'TRADE INFORMATION' with details such as TRADE ID (1234), TRADE STOCK SYMBOL (IBM), NUMBER OF SHARES (10), REPORTING COUNTRY (U.S.), TRADE DESCRIPTION (Please buy 10 shares), TRADE REPORTER NAME (Hans Welman), PARTY 1 (MyBank), and PARTY 2 (SIB). On the right, there's a 'Field List' panel with three tabs: 'Field Name', 'Field Type', and 'Value'. Below it is a table titled 'INSERTED FIELDS' with columns: Field Name, Field Type, Field Value, and Select Variable. The table lists eight fields with their corresponding values and variable names:

Field Name	Field Type	Field Value	Select Variable
vTradeID	Custom	1234	vTradeID
vStockSy...	Custom	IBM	vStockSymbol
vShares	Custom	10	vShares
vReporting...	Custom	U.S.	vReportingCo
vTradeDesc	Custom	Please buy ...	vTradeDesc
vReporter...	Custom	Hans Welman	vReporterName
vParty1	Custom	MyBank	vParty1
vParty2	Custom	SIB	vParty2

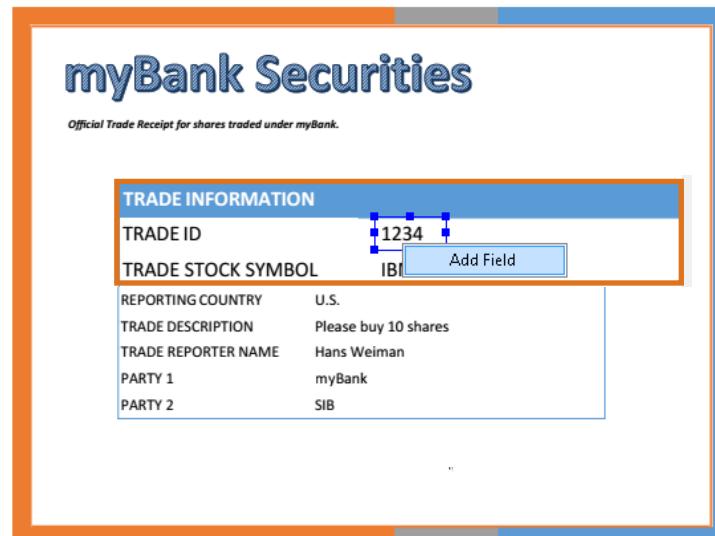
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-17. PDF Integration command – PDF Viewer (5 of 5)

PDF Integration command: Caveats (1 of 2)

- It can be tricky to outline fields in the PDF when highlighting the static data to be retrieved
- The outline can vary based on the field and potential field values. It is a good practice to ensure that the outline is large enough to accommodate the largest value possible
- You create an outline of the area to be extracted by using the mouse
- After the outline is created, right-click the highlighted value and click **Add Field**
- Assign the value to a value variable
- The field is then added to the list of **Inserted Fields**



Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-18. PDF Integration command: Caveats (1 of 2)

PDF Integration command: Caveats (2 of 2)

- There are limitations that are helpful to know as you outline each field:
 - After you add the field and it appears in the Inserted Fields list to the right, the box remains
 - Instead of re-creating an outline for the next field, you can move the outline to the next field and adjust as needed
 - The outline does not display while moving or dragging it
 - You can outline only one value at a time
 - When you highlight the outline, squares appear that allow you to either maneuver or stretch the outline
 - You cannot edit the placement of the outline after it is an inserted field



TRADE INFORMATION	
TRADE ID	1234
TRADE STOCK SYMBOL	IBI
REPORTING COUNTRY	U.S.
TRADE DESCRIPTION	Please buy 10 shares
TRADE REPORTER NAME	Hans Weiman
PARTY 1	myBank
PARTY 2	SIB

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-19. PDF Integration command: Caveats (2 of 2)

7.3. Extending the String Operation command to trim variables

Extending the String Operation command to trim variables

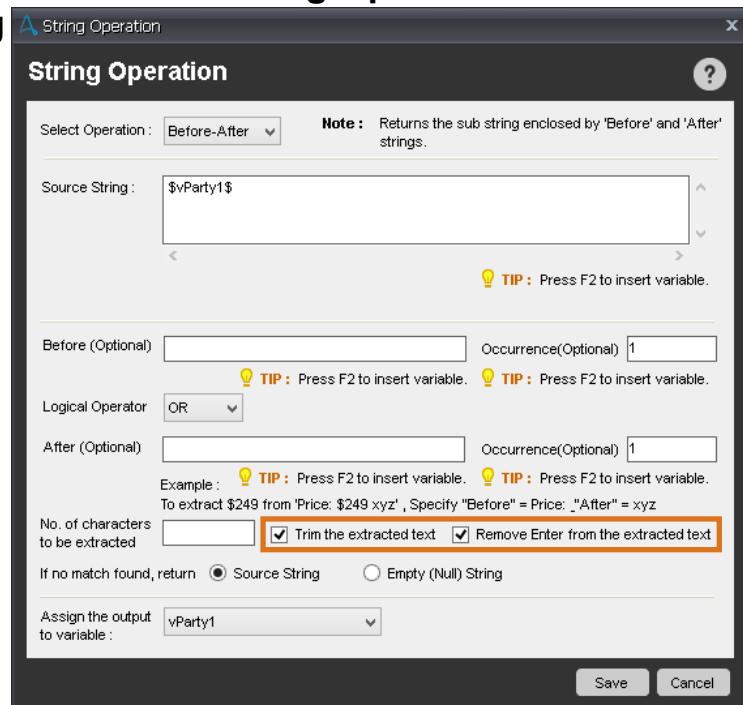
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-20. Extending the String Operation command to trim variables

Extending the String Operation command to trim variables

- The **Before-After** subcommand of the **String Operation** command is used to return a substring
- This command is also used to remove Enter characters from an extracted field
- When you extract fields from a PDF, the Enter key is extracted as part of the value
- This command is useful when used along with the **PDF Integration** command to ensure that the extracted text does not contain any superfluous characters



Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-21. Extending the String Operation command to trim variables

7.4. Extending the String Operation command to get length of variables

Extending the String Operation command to get length of variables

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-22. Extending the String Operation command to get length of variables

Extending the String Operation command to get length of variables

- The **Length** subcommand of the **String Operation** command is used to return the length of the source string
- The result is saved to another value variable
- The result might include characters that are not visible in the source string such as an Enter key character
- For example, if the source string has no value but a length of one, it's possible the Enter character is in the source



Figure 7-23. Extending the String Operation command to get length of variables

Unit summary

- Download an attachment from an email
- Extract data from a PDF file
- Work with extracted text strings
- Determine the length and value of variables

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-24. Unit summary

Review questions

1. True or False: The **Email Automation** command is used to enable the bot to send email.
2. True or False: When you use the **PDF Integration** command, you cannot edit the placement of the outline after it is an inserted field.
3. Choose the best answer for the following scenario:
A bot Developer is extracting data from a PDF and needs to remove the Enter key character from the variables.
Which command is used to accomplish this task?
 - A. Variable Operation
 - B. String Operation
 - C. PDF Integration
 - D. None of the above



Review answers

1. True or False: The **Email Automation** command is used to enable the bot to send email.
The answer is False.
2. True or False: When you use the **PDF Integration** command, you cannot edit the placement of the outline after it is an inserted field.
The answer is True.
3. Choose the best answer for the following scenario:
A bot Developer is extracting data from a PDF and needs to remove the Enter key character from the variables.
Which command is used to accomplish this task?
 - A. Variable Operation
 - B. String Operation
 - C. PDF Integration
 - D. None of the above.**The answer is B.**



Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-26. Review answers

Exercise 6: Creating a bot to evaluate data from a PDF and send an email

Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

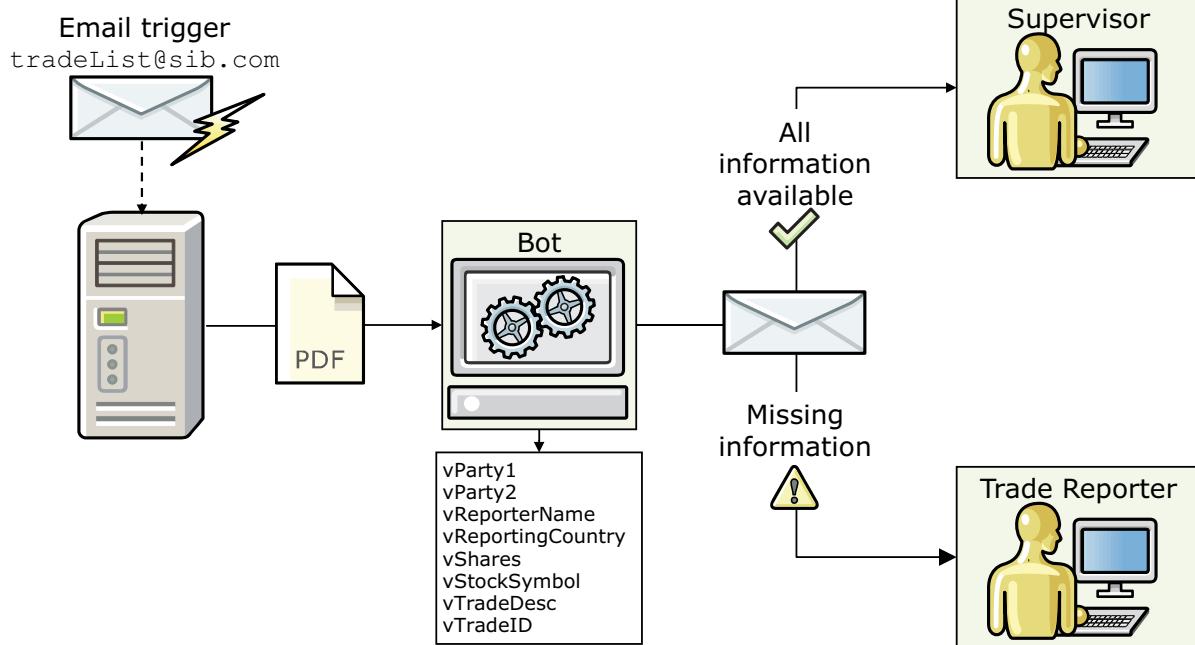
Figure 7-27. Exercise 6: Creating a bot to evaluate data from a PDF and send an email

Exercise objectives

- Use the **Email Automation** command to pull email attachments and save them to a local drive
- Use the **PDF Integration** command to extract values from a PDF
- Use the **String Operation** command to remove the Enter key from the extracted text and get length of each variable
- Use the **If/Else** command to send an email based on a variable evaluation
- Clean up the environment by deleting email messages and files used
- Define a bot trigger to start the bot based on the receipt of an email

Exercise introduction (1 of 7)

In this exercise, you play the role of a Bot Developer to create a bot to perform the following steps.



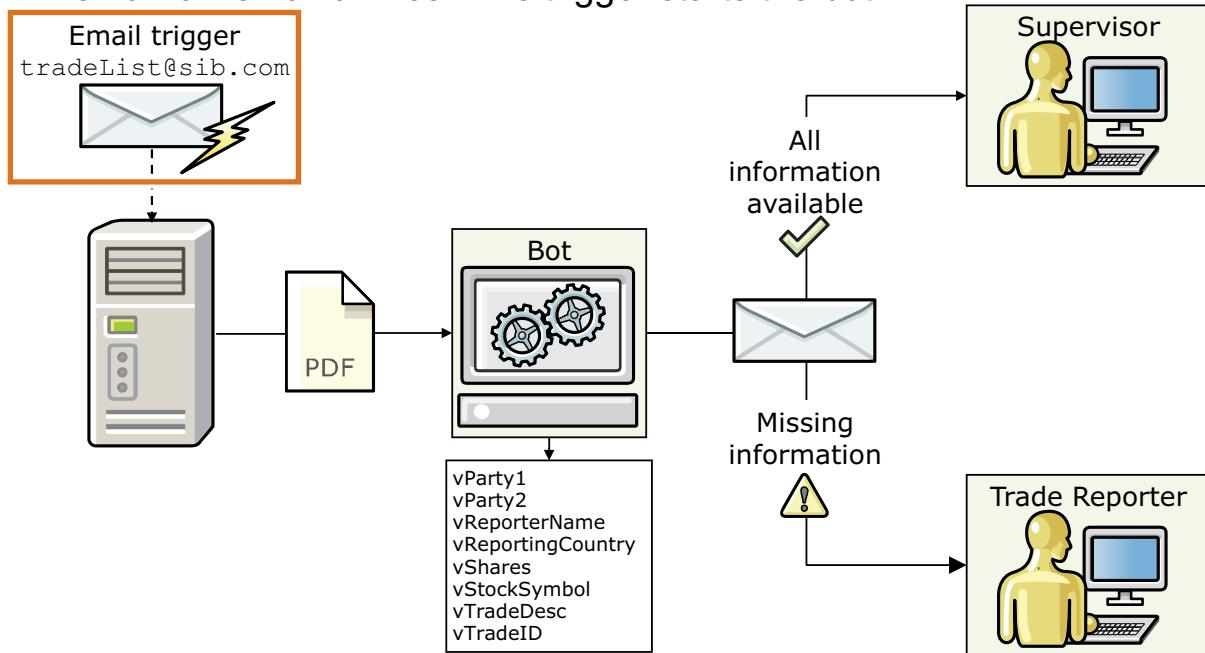
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-29. Exercise introduction (1 of 7)

Exercise introduction (2 of 7)

After you build the bot code, a trigger is configured to occur when a new email arrives. This trigger starts the bot.



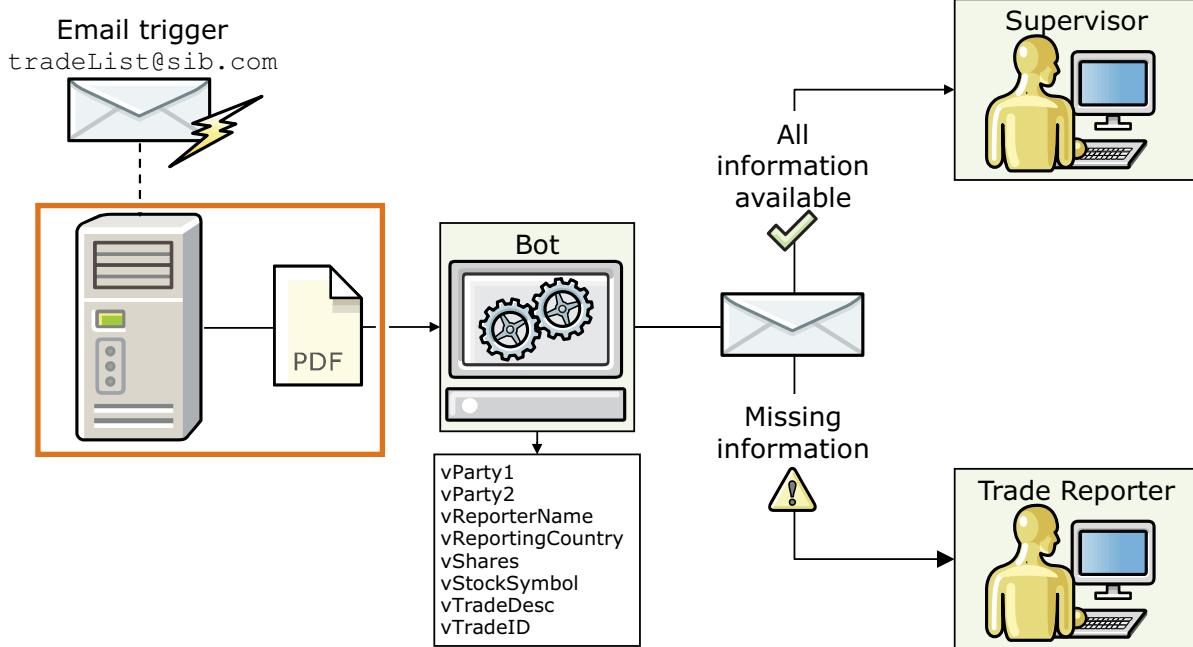
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-30. Exercise introduction (2 of 7)

Exercise introduction (3 of 7)

After the bot is triggered, it pulls the attachment from the email, extracts data from the PDF, and trims all variables.



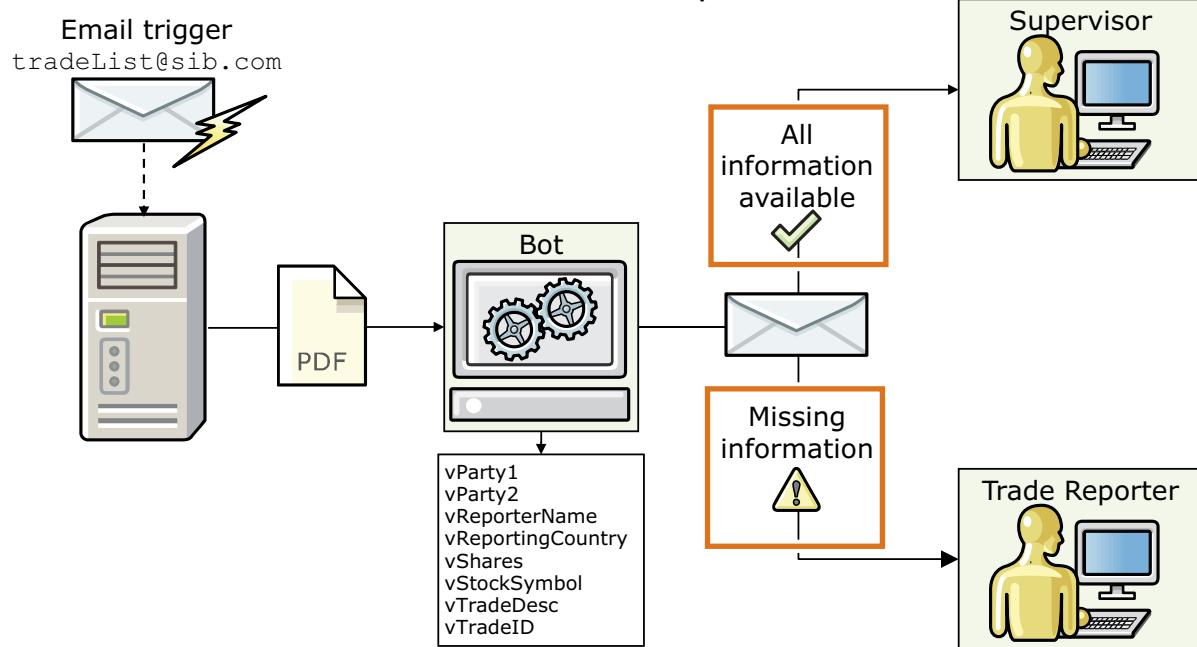
Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-31. Exercise introduction (3 of 7)

Exercise introduction (4 of 7)

The bot then evaluates the data that is extracted and sends an email based on whether all information is present.



Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-32. Exercise introduction (4 of 7)

Exercise introduction (5 of 7)

- To build the bot, you create a set of variables, develop bot code, and verify correct functioning
- When you build the bot, you perform the following:
 - Create a new bot and add custom variables
 - Pull trade receipts from an email, save to a local folder, and extract values from the PDF
 - Trim the variables and determine their length
 - Evaluate the data and send an email based on one of two situations
 - Cleanup: Delete all emails and files in folder



Working with email and PDFs

© Copyright IBM Corporation 2018, 2019

Figure 7-33. Exercise introduction (5 of 7)

Exercise introduction (6 of 7)

The following commands are used in this exercise:



Command	Purpose
Comment	Insert code annotations
Email Automation	Get all messages from email server Delete all messages from email server
Files/Folders	Delete PDF after bot completes.
If/Else	Evaluate whether there are variables with a length of zero (missing information)
Loop	This command is used with the Email Automation command to get all messages from the email server

Figure 7-34. Exercise introduction (6 of 7)

Exercise introduction (7 of 7)

The following commands are used in this exercise:



Command	Purpose
PDF Integration	Extracts data from the PDF and assigns to value variables
Send Email	Send email to either Trade Reporter or Trade Reporter Supervisor
String Operation	This command is used to trim variables and assign their length to another variable

Figure 7-35. Exercise introduction (7 of 7)

Unit 8. Creating interactive bots

Estimated time

01:00

Overview

This unit covers how to add interactive elements, such as message windows and text entry prompts, to a bot.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe the interactive components that you can configure in a bot
- Describe how to implement interactive components in a bot

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-1. Unit objectives

Topics

- Attended versus unattended bots
- Using interactive elements
- Using the Web Recorder command to extract a table from a web page
- Extending the Excel command to find a value in a spreadsheet
- Introduction to bot hardening

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-2. Topics

8.1. Attended versus unattended bots

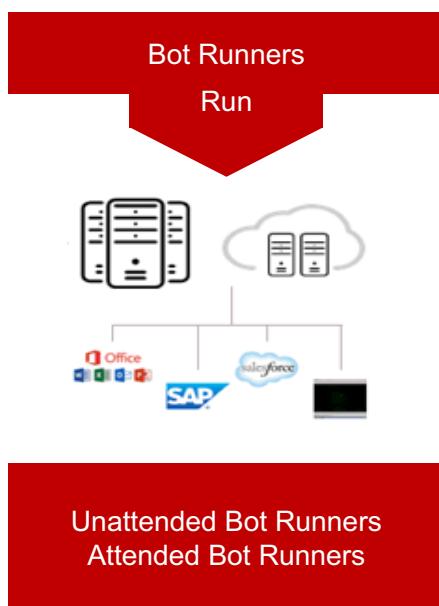
Attended versus unattended bots

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-3. Attended versus unattended bots

Attended bots (1 of 2)



Restrictions:

- Must run on the user's workstation using the user's Windows credentials to perform work on their behalf
- Cannot be automatically scheduled
- Can trigger other attended bots, but cannot trigger an unattended bot either directly or indirectly
- Can either be started by a user or triggered by:
 - Event such as a hot key
 - Event that occurs on users workstation (such as an email arrival or document being placed on the users desktop)

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-4. Attended bots (1 of 2)

Attended bots (2 of 2)



Benefits:

- Makes individuals more productive
 - Eliminates tedious, lengthy error prone manual desktop tasks
- Simpler to develop, quicker to implement
 - Does not require robust error handling
- Faster and less costly to deploy
 - Does not require provisioning of bot runners
- Less costly to operate
 - User can recover from failures
- Less disruptive to existing work
 - Addresses specific problem to enhance end-users work

Figure 8-5. Attended bots (2 of 2)

Comparing attended and unattended bots

- In the last two exercises, you created unattended bots with a pre-configured trigger.
- In the next exercise, you create an attended bot.
- This unit covers some of the interactive elements you can use with attended bots.

	Attended Bot	Unattended Bot
What	Completes work for an individual	Completes work in an unattended manner
How	Triggered by individual or even on users workstation	Pre-configured trigger (no human involved)
When	Individual triggers bot to run	Bot is scheduled to run by pre-configured triggers or schedule
Best Fit	Front-office work, for example: Service Desks, Help Desks, and Call Centers	Back-office work that is prone to automation

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-6. Comparing attended and unattended bots

Examples of attended bot implementations

Virtual assistant

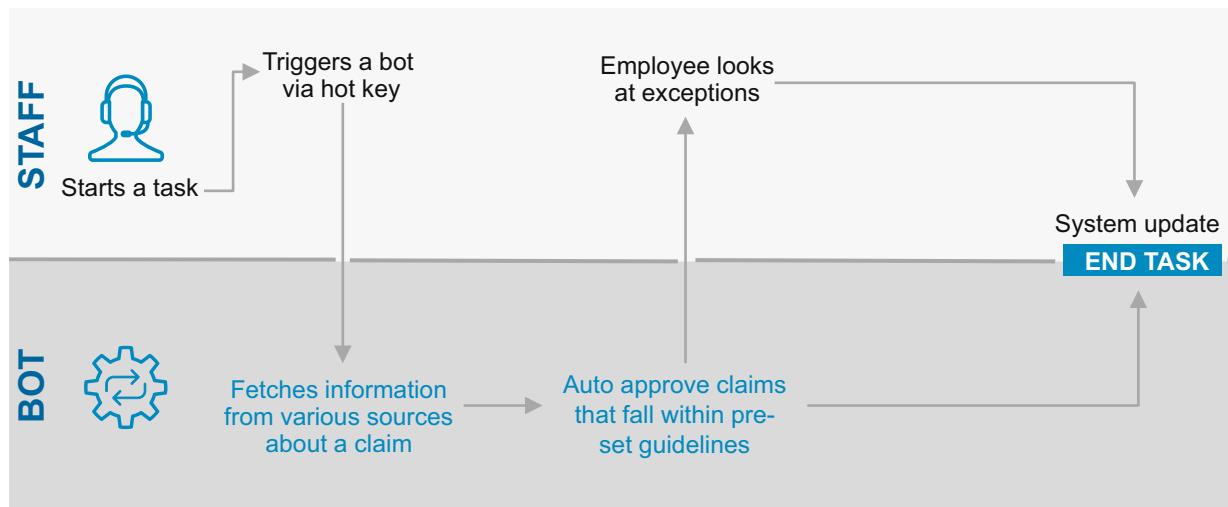


Figure 8-7. Examples of attended bot implementations (1 of 3)

Examples of attended bot implementations

Guidance and assistance

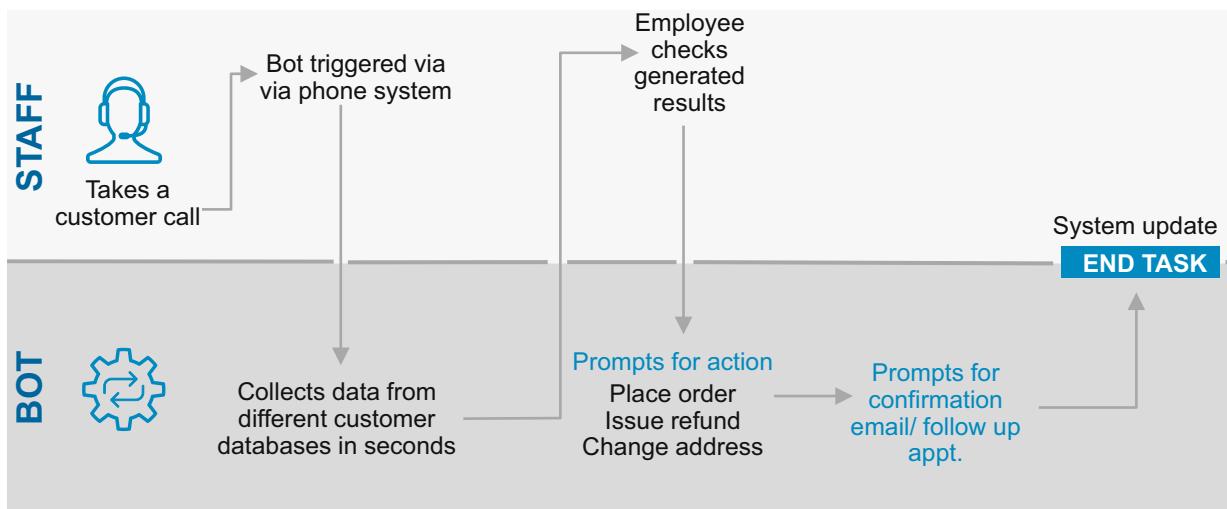
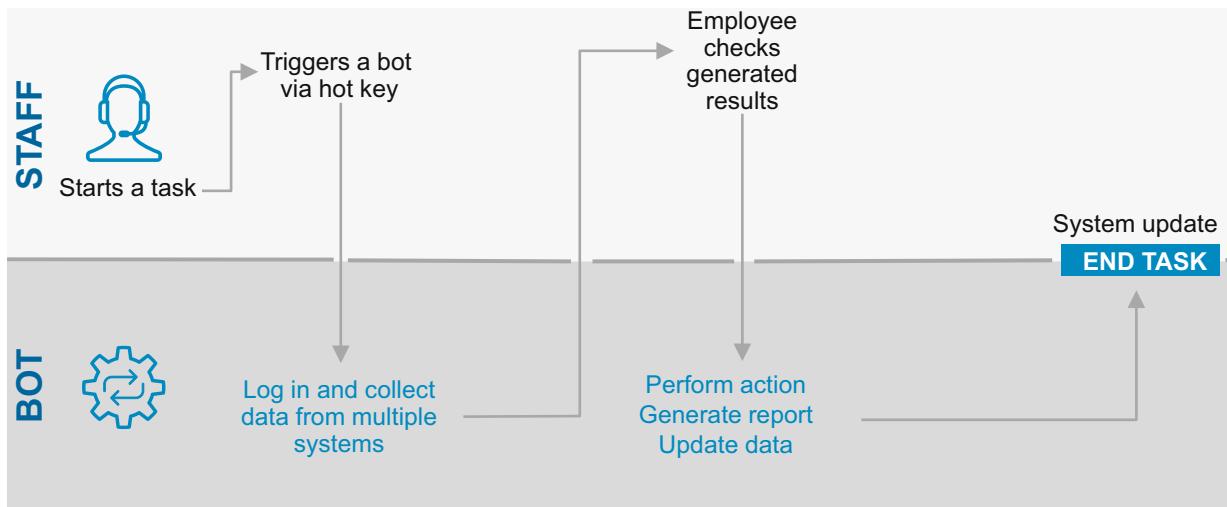


Figure 8-8. Examples of attended bot implementations (2 of 3)

Examples of attended bot implementations

On-demand data processing



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-9. Examples of attended bot implementations (3 of 3)

8.2. Using interactive elements

Using interactive elements

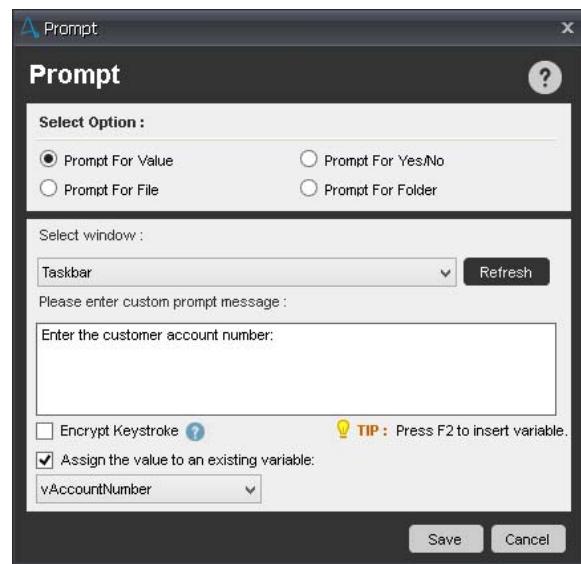
Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-10. Using interactive elements

Using interactive elements: Prompt

- Sometimes, a bot might need to receive input from a user.
- The **Prompt** command is used to interact with the user.
- The prompt can be used for:
 - A value
 - A file
 - Yes/No
 - A Folder
- The value can be saved to an existing variable.
- If the bot prompts the user for a password or other sensitive information, the keystrokes can be encrypted. However, you cannot assign a variable if the keystrokes are encrypted.
- Alternatively, you can right-click any **Insert Keystrokes** command and convert to a Prompt.



Creating interactive bots

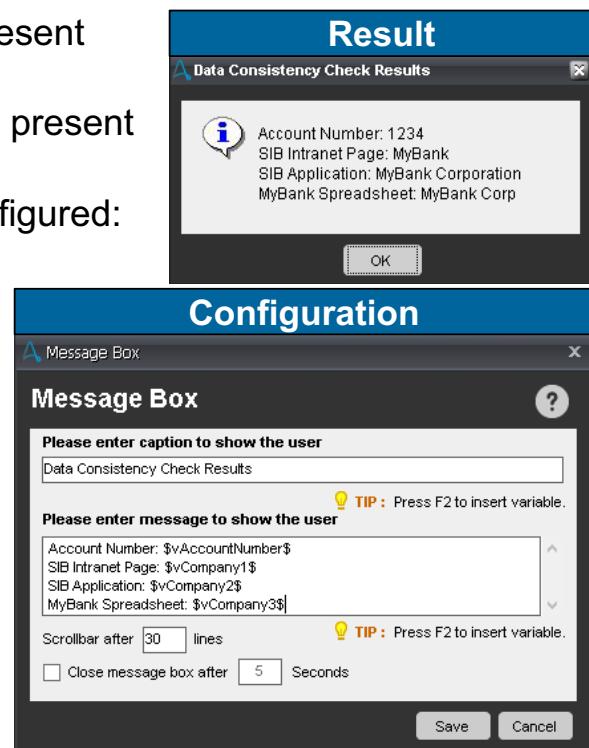
© Copyright IBM Corporation 2018, 2019

Figure 8-11. Using interactive elements: Prompt



Using interactive elements: Message

- Sometimes, a bot might need to present information to the user.
- The **Message** command is used to present a message to the user.
- The following elements can be configured:
 - Caption
 - Message
 - Scroll bar
 - Timeout
- Each value can be a variable.
- Message boxes are always presented with the “i” bubble and an OK button by default.



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-12. Using interactive elements: Message

8.3. Using the Web Recorder command to extract a table from a web page

Using the Web Recorder command to extract a table from a web page

Creating interactive bots

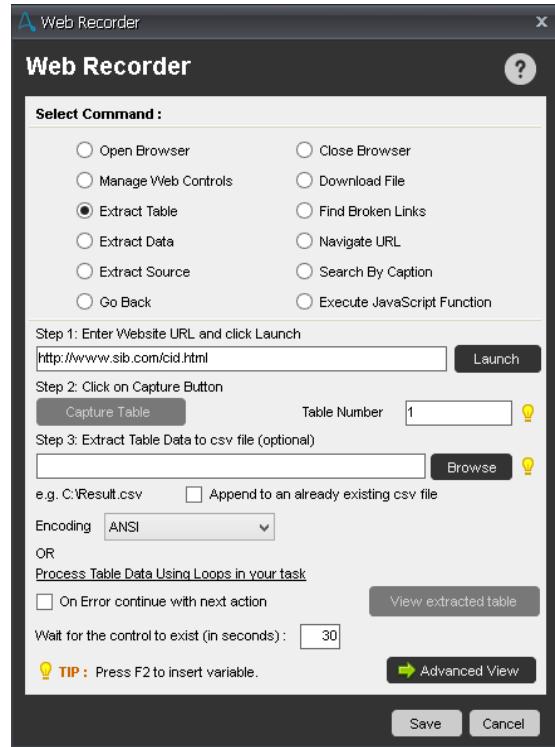
© Copyright IBM Corporation 2018, 2019

Figure 8-13. Using the Web Recorder command to extract a table from a web page



Using the Web Recorder command to extract a table from a web page (1 of 7)

- You can use the Web Recorder to extract data, including tables, from web pages
- When you use the Web Recorder, the data is saved to a variable or the table is saved to a CSV file
- Within the Workbench, you can use the **Web Recorder command** to perform the same functions
- Using a **Loop** command with the **Web Recorder** command allows the bot to iterate through the values that are extracted from the table
- Use the following **Loop** subcommand: **Each row in an Internet Explorer Table**



Creating interactive bots

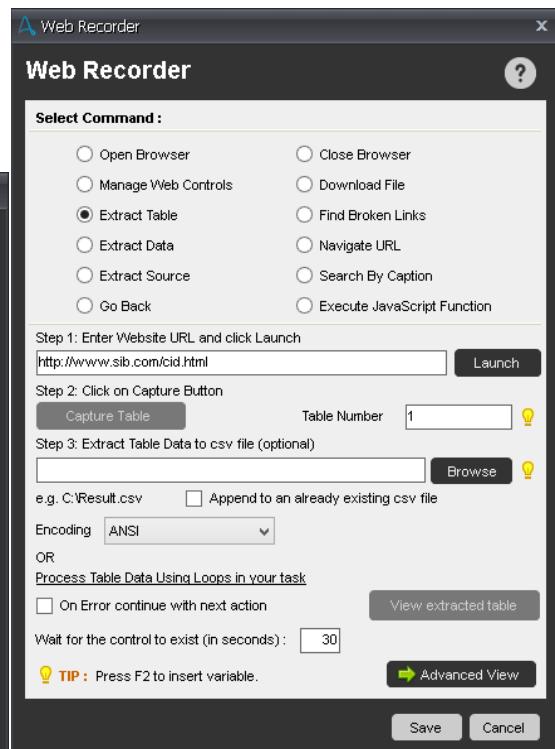
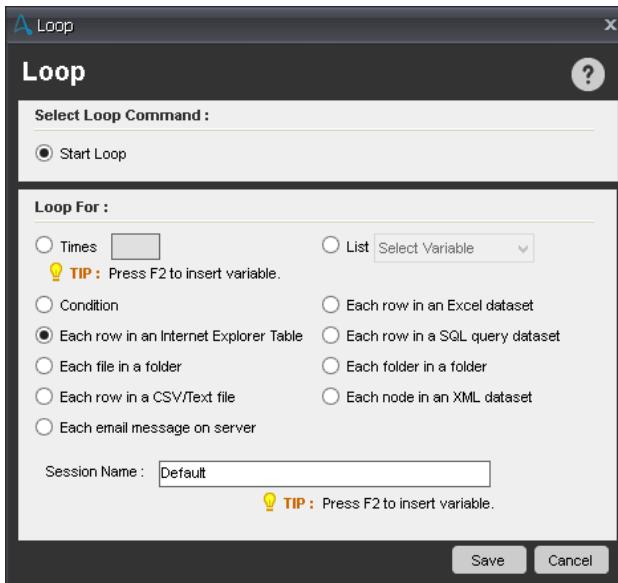
© Copyright IBM Corporation 2018, 2019

Figure 8-14. Using the Web Recorder command to extract a table from a web page (1 of 7)



Using the Web Recorder command to extract a table from a web page (2 of 7)

- Using the Loop command saves data in the table to the system variable **\$Table Column(index)\$**.



Creating interactive bots

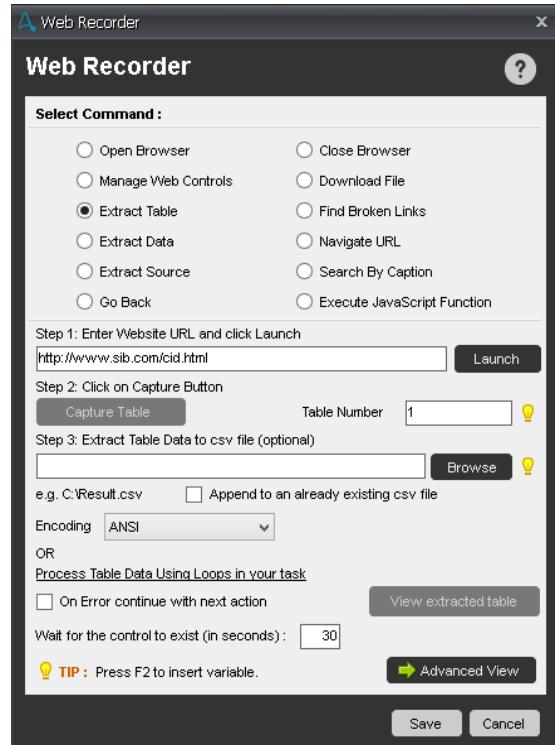
© Copyright IBM Corporation 2018, 2019

Figure 8-15. Using the Web Recorder command to extract a table from a web page (2 of 7)



Using the Web Recorder command to extract a table from a web page (3 of 7)

- The data is extracted in row-and-column format
- Use the Web Recorder command for these functions (subcommands):
 - Open Browser
 - Close Browser
 - Manage Web Controls
 - Download File
 - Extract Table
 - Find Broken Links
 - Extract Data
 - Navigate URL
 - Extract Source
 - Search by Caption
 - Go Back
 - Execute JavaScript Function



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-16. Using the Web Recorder command to extract a table from a web page (3 of 7)

IBM Training



Using the Web Recorder command to extract a table from a web page (4 of 7)

To extract a table from a web page, perform the following steps:

- Step 1: Enter website URL and click **Launch** to start the website in Internet Explorer
- Step 2: Click **Capture Table** to open the website with a green highlighter and yellow prompt, then click the table to be extracted

Account Number	Company	First Name	Last Name	Phone Number	Email Address
1234	MyBank	Meredith	Chu	650-555-1212	mchu@mybank.com
2345	Miniloan	James	Doe	202-555-1212	jdoe@miniloan.com
3456	Smart Investment Bank	Bea	Gonzalez	718-555-1212	gonzalez@sib.com
4567	MyBank	Daniel	Carson	404-555-1212	dcarson@mybank.com

- Step 3: Enter location for CSV file to house the extracted table (optional)
- Click **Advanced View** to see the remaining steps.

Creating interactive bots



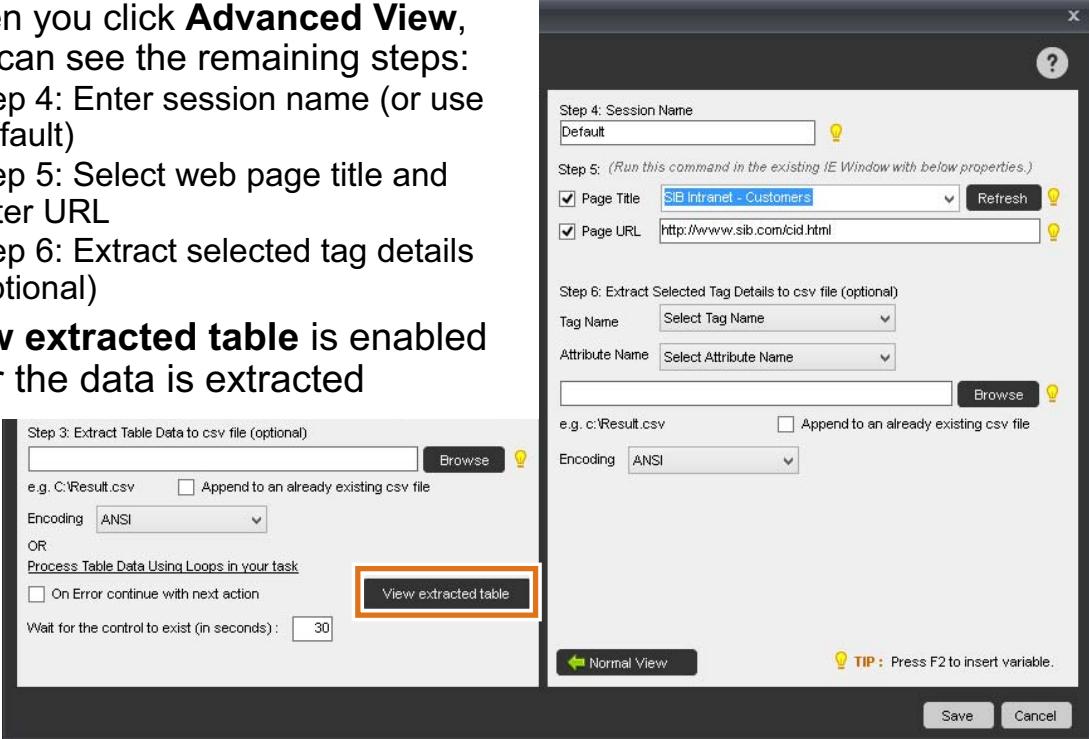
© Copyright IBM Corporation 2018, 2019

Figure 8-17. Using the Web Recorder command to extract a table from a web page (4 of 7)

IBM Training IBM

Using the Web Recorder command to extract a table from a web page (5 of 7)

- When you click **Advanced View**, you can see the remaining steps:
 - Step 4: Enter session name (or use Default)
 - Step 5: Select web page title and enter URL
 - Step 6: Extract selected tag details (optional)
- View extracted table** is enabled after the data is extracted

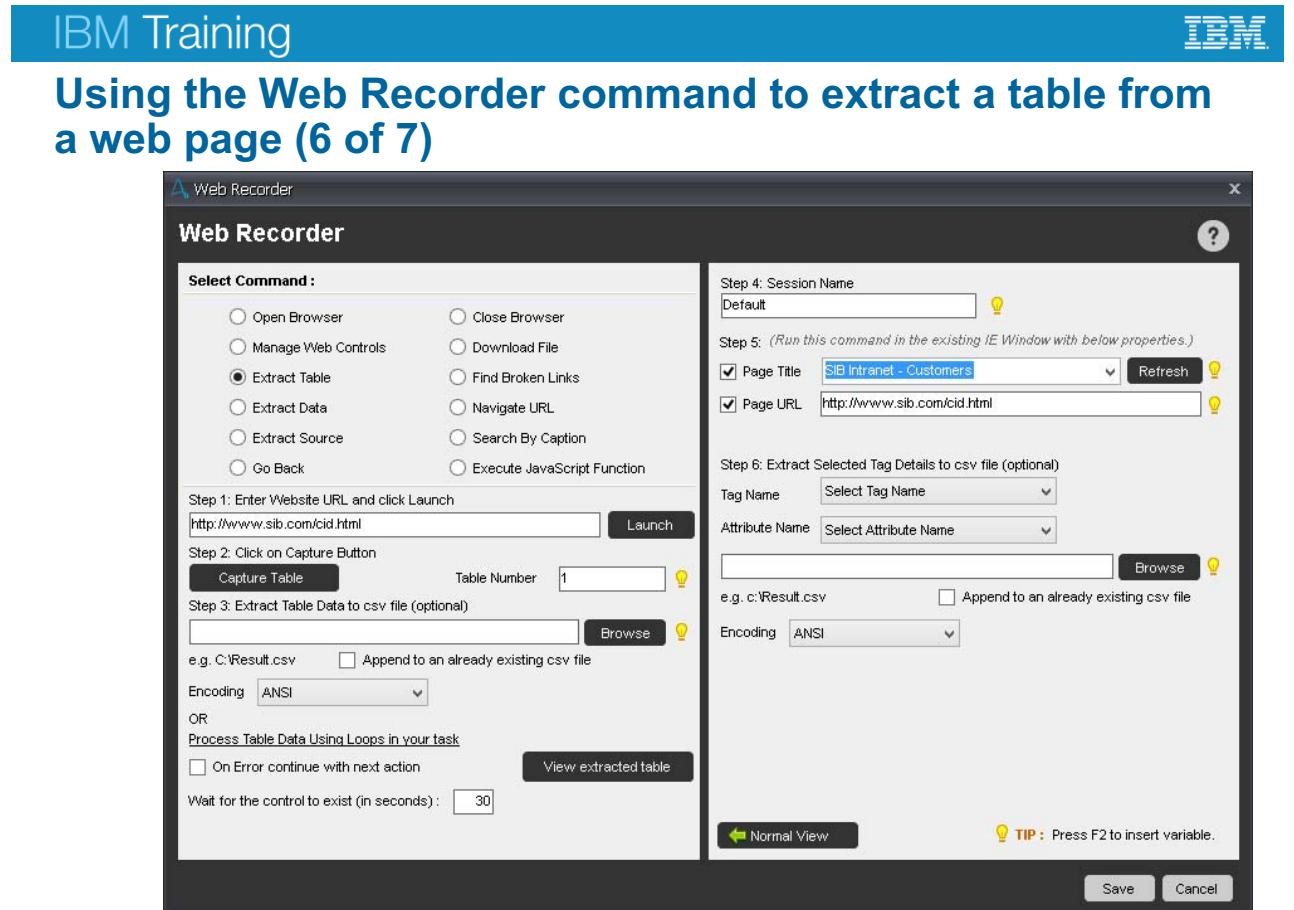


The screenshot shows the 'Web Recorder' configuration window. At the top, there are sections for 'Step 4: Session Name' (set to 'Default') and 'Step 5: (Run this command in the existing IE Window with below properties.)' where 'Page Title' is set to 'SIB Intranet - Customers' and 'Page URL' is 'http://www.sib.com/cid.html'. Below these, 'Step 6: Extract Selected Tag Details to csv file (optional)' is shown with dropdowns for 'Tag Name' and 'Attribute Name', and a 'Browse' button for the output file path 'e.g. c:\Result.csv'. A checkbox 'Append to an already existing csv file' is checked. Encoding is set to 'ANSI'. At the bottom left, there's a section for 'Process Table Data Using Loops in your task' with options for 'On Error continue with next action' and 'Wait for the control to exist (in seconds): 30'. A large orange box highlights the 'View extracted table' button. On the right, there's a 'Normal View' button, a tip 'TIP : Press F2 to insert variable.', and 'Save' and 'Cancel' buttons.

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-18. Using the Web Recorder command to extract a table from a web page (5 of 7)



Creating interactive bots

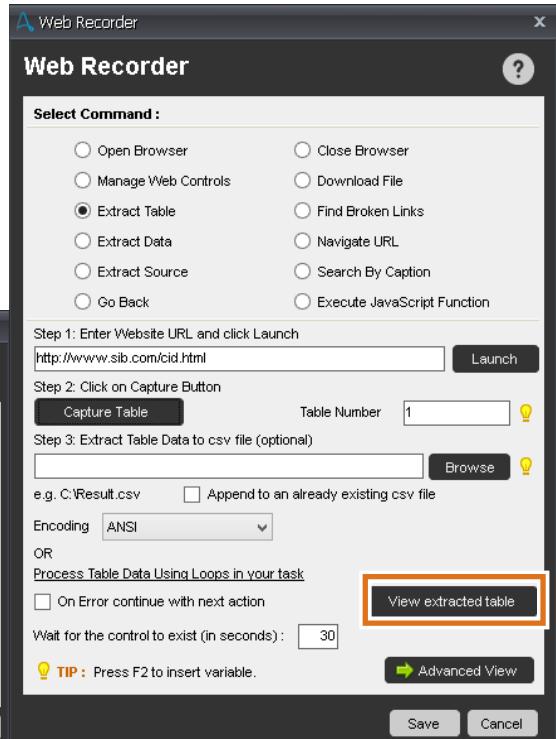
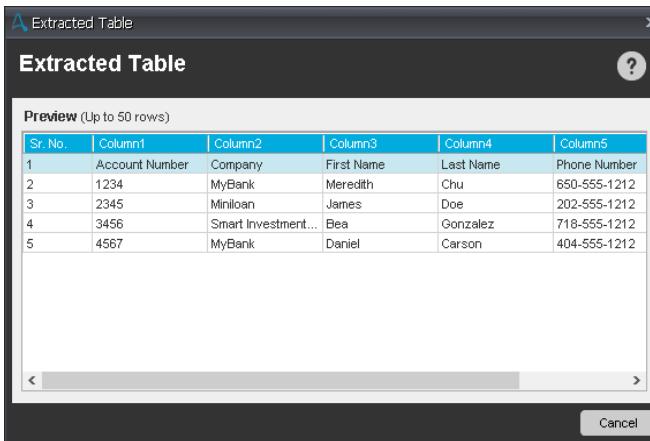
© Copyright IBM Corporation 2018, 2019

Figure 8-19. Using the Web Recorder command to extract a table from a web page (6 of 7)



Using the Web Recorder command to extract a table from a web page (7 of 7)

- It's a good practice to check the extracted table to verify that the data is correct
 - Data might be different when the bot runs
- The Extracted Table preview displays the first 50 rows only



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-20. Using the Web Recorder command to extract a table from a web page (7 of 7)

8.4. Extending the Excel command to find a value in a spreadsheet

Extending the Excel command to find a value in a spreadsheet

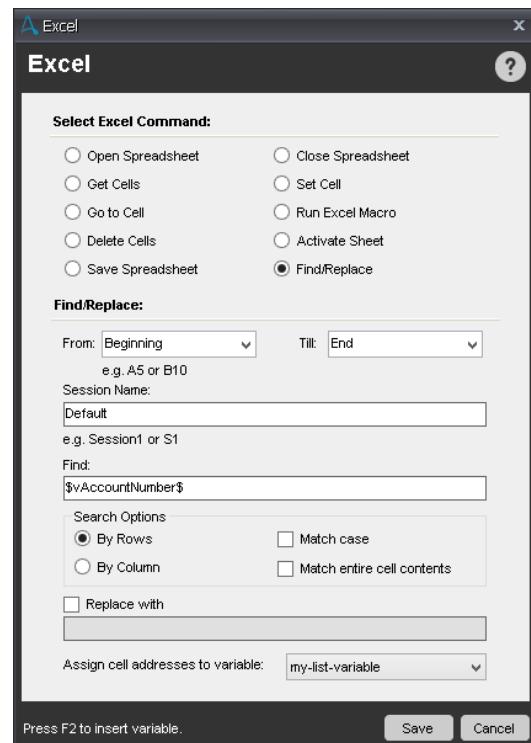
Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-21. Extending the Excel command to find a value in a spreadsheet

Extending the Excel command to find a value in a spreadsheet

- You can use the **Find/Replace** subcommand of the **Excel** command to either find or find and replace values in a spreadsheet
- You can specify the scope of the search by setting the beginning and end cells
- You can search by rows or columns
- Your search can be configured to match case or entire cell contents
- You can optionally replace the value that is found
- The cell addresses (such as A1) are saved to a list array variable



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-22. Extending the Excel command to find a value in a spreadsheet

8.5. Introduction to bot hardening

Introduction to bot hardening

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-23. Introduction to bot hardening

Introduction to bot hardening

- Hardening a software application against known vulnerabilities is a common practice in software development
- It is often used with error handling to detect known situations and respond to them gracefully
- Responding to known exceptions keeps the bot processing and ensures resiliency
- Example:
A bot correlates the company name for a particular account number across three systems. The first system to access should contain the account number. If it doesn't, it is known there is no need to continue accessing the other systems. If the bot doesn't find the account number, it presents a message box and stops the task.

```

IF If $vFound$ Not Equal To (<>) "true" Then
    Comment: Show a message that tells the user that the account number was not found, and stop the task.
    Message Box: "Account $vAccountNumber$ was not found. The automated task will end."
    Close Browser
    Stop The Current Task
End If

```

Figure 8-24. Introduction to bot hardening

Unit summary

- Describe the interactive components that you can configure in a bot
- Describe how to implement interactive components in a bot

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-25. Unit summary

Review questions

1. True or False: The **Prompt** command can be used to prompt the user for data that can be saved to a variable for further processing.

2. Which variable holds data when running a loop to extract a table from a web page?
 - A. Counter
 - B. Table Column(index)
 - C. Dataset Column(index)
 - D. Excel Column

3. Which loop subcommand is used to extract values from a web page?
 - A. Each row in an Internet Explorer Table
 - B. Each row in an SQL query data set
 - C. Each row in a web page
 - D. None of the above



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-26. Review questions

Review answers



1. **True** or False: The **Prompt** command can be used to prompt the user for data that can be saved to a variable for further processing.
The answer is True.
2. Which variable holds data when running a loop to extract a table from a web page?
 - A. Counter
 - B. Table Column(index)
 - C. Dataset Column(index)
 - D. Excel ColumnThe answer is B.
3. Which loop subcommand is used to extract values from a web page?
 - A. Each row in an Internet Explorer Table
 - B. Each row in an SQL query data set
 - C. Each row in a web page
 - D. None of the aboveThe answer is A.

Exercise 7: Creating an interactive bot to check values in disparate systems

Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-28. Exercise 7: Creating an attended bot to check values in disparate systems

Exercise objectives

- Create interactive prompts in bots
- Use the Web Recorder to extract web page data
- Test the bot interactively by running it as an attended bot

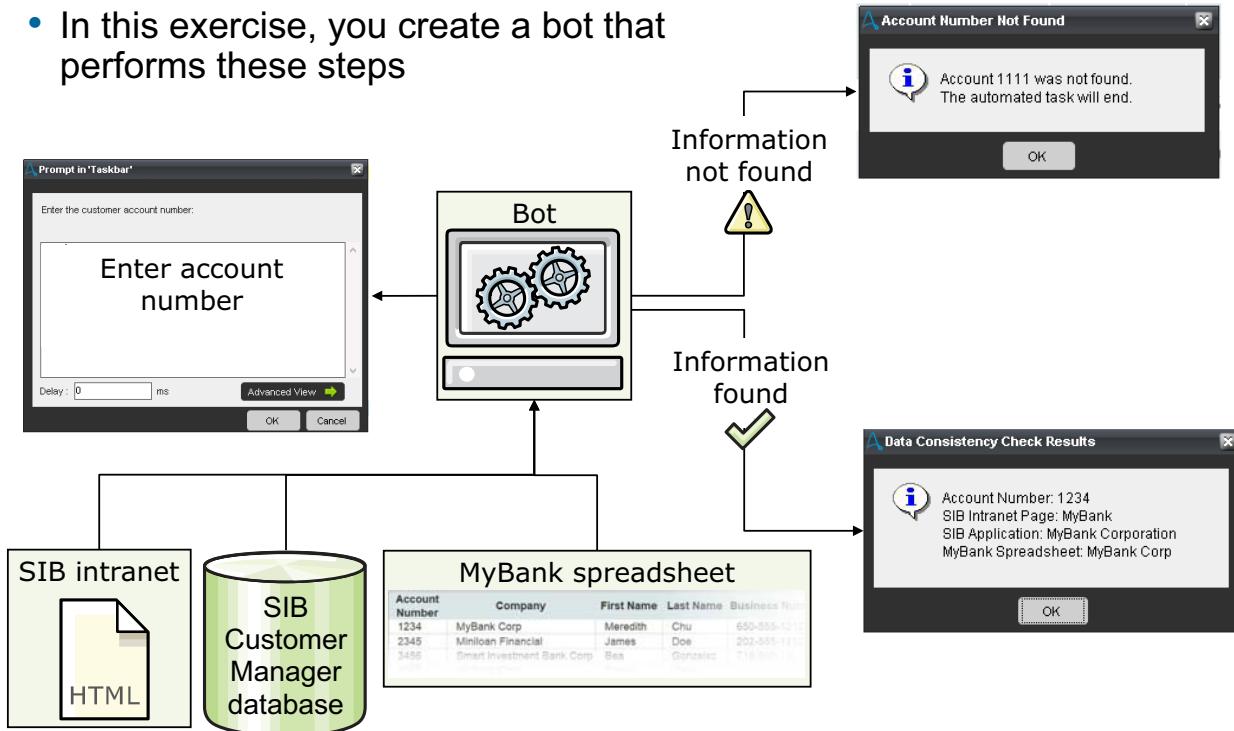
Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-29. Exercise objectives

Exercise introduction (1 of 7)

- In this exercise, you create a bot that performs these steps



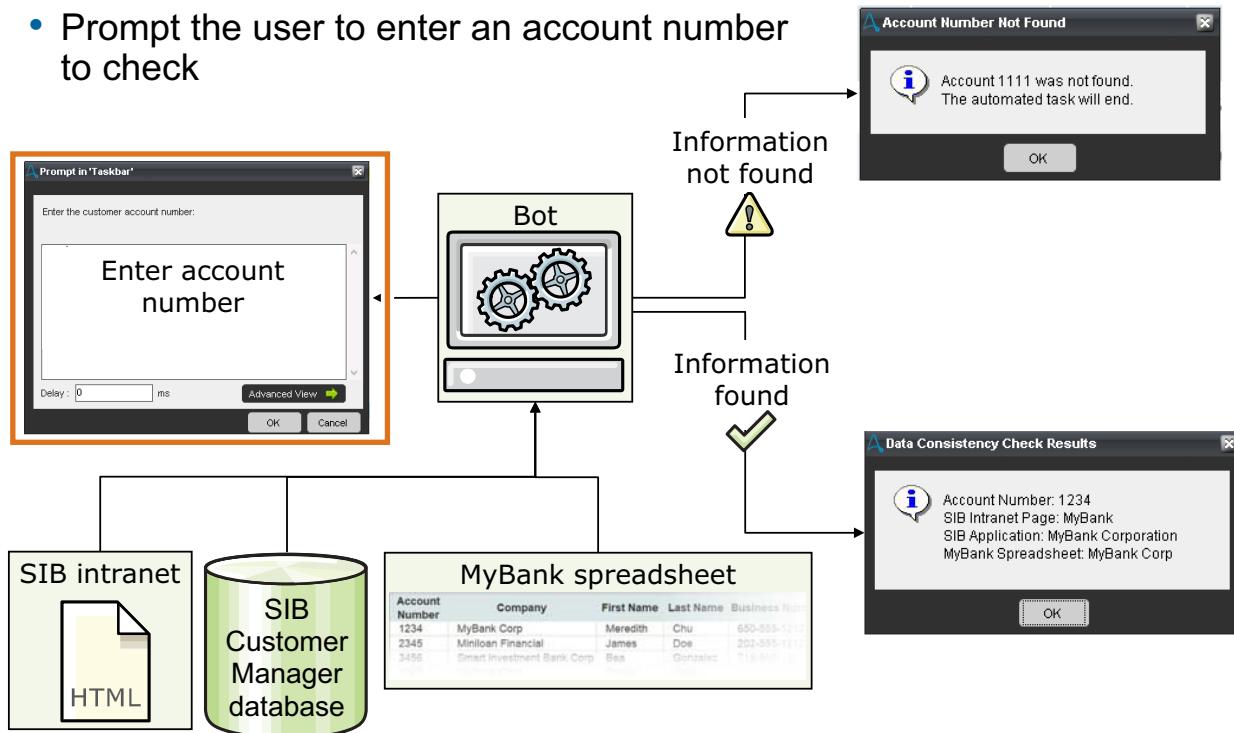
Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-30. Exercise introduction (1 of 7)

Exercise introduction (2 of 7)

- Prompt the user to enter an account number to check



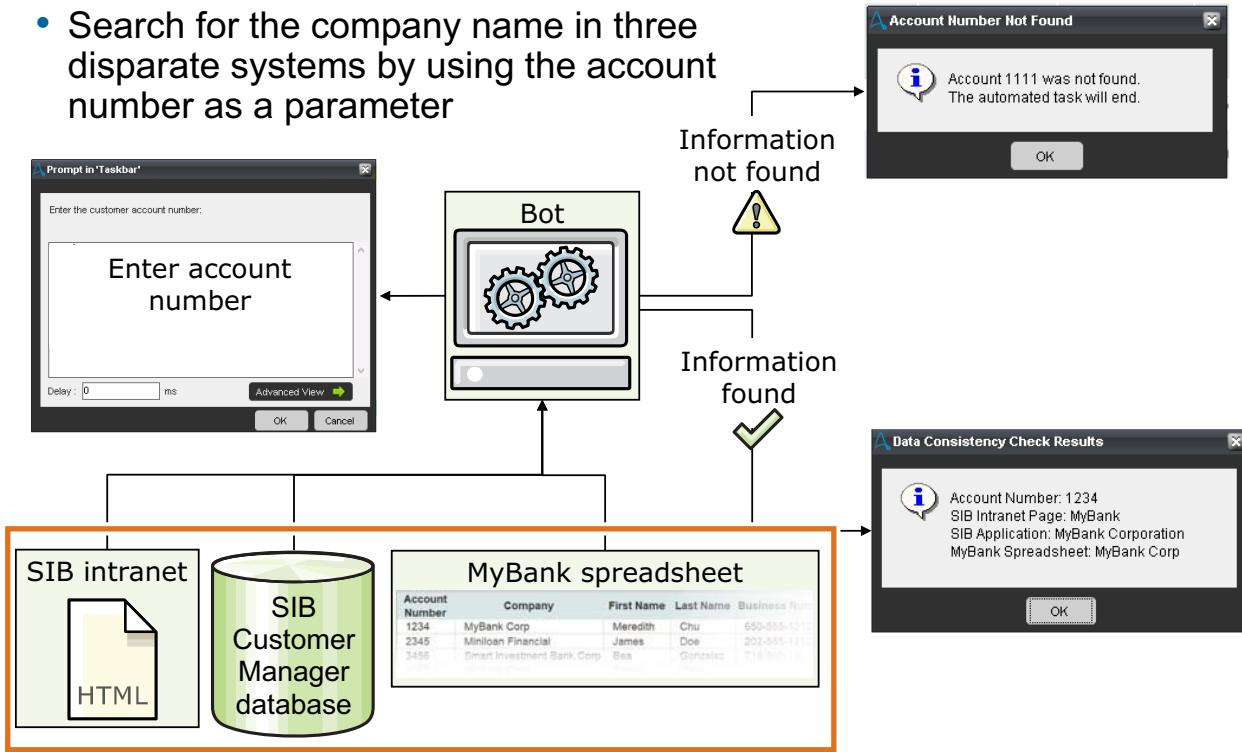
Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-31. Exercise introduction (2 of 7)

Exercise introduction (3 of 7)

- Search for the company name in three disparate systems by using the account number as a parameter



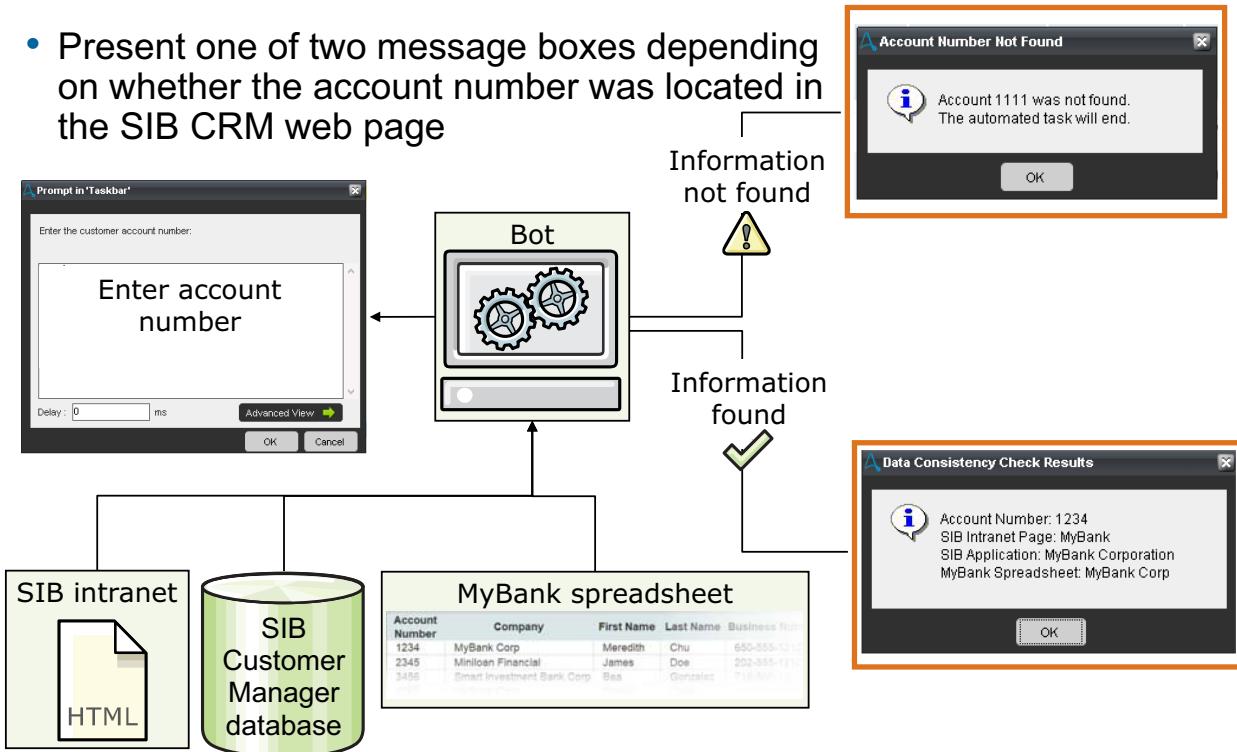
Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-32. Exercise introduction (3 of 7)

Exercise introduction (4 of 7)

- Present one of two message boxes depending on whether the account number was located in the SIB CRM web page



Creating interactive bots

© Copyright IBM Corporation 2018, 2019

Figure 8-33. Exercise introduction (4 of 7)

Exercise introduction (5 of 7)

The following commands are used in this exercise:



Command	Purpose
Comment	Insert code annotations
Excel	Opens the mybank_accounts spreadsheet, finds account number in the spreadsheet, goes one cell right, gets value of active cell and assigns to \$vCompany3\$, closes the spreadsheet
If/Else	Evaluates whether the account number is found when searching in the web page and SIB Customer Manager Access database
Loop	Used with the Web Recorder command to extract rows of data in a table from a web page

Figure 8-34. Exercise introduction (5 of 7)

Exercise introduction (6 of 7)

The following commands are used in this exercise:



Command	Purpose
Message Box	Displays the results or displays an “account not found” message if the account number is not found on the web page
Object Cloning	Logs in to the SIB Customer Manager Access database and searches for company by account number
Open Program/File	Opens the SIB Customer Manager Access database
Prompt	Prompts user for account number to check
Task	Stops the current bot (if the account number is not found on the web page)

Figure 8-35. Exercise introduction (6 of 7)

Exercise introduction (7 of 7)

The following commands are used in this exercise:



Command	Purpose
Variable Operation	Used to copy the company found in the SIB intranet application to variable \$vCompany1\$ and to copy the company found in the SIB Contacts Manager Access database to variable \$vCompany2\$
Window Actions	Resizes the SIB Customer Manager Access database Login window and closes the SIB Contacts Manager window
Web Recorder	Opens the SIB CRM website, extracts a table from the Customers page, and closes the browser

Figure 8-36. Exercise introduction (7 of 7)

Unit 9. Introducing MetaBots

Estimated time

01:00

Overview

This unit provides an overview of MetaBots and the MetaBot Designer.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe how to use MetaBots
- Describe the features and components of the MetaBot Designer
- Explain how to create a MetaBot

[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-1. Unit objectives

Topics

- Overview of MetaBots
- Benefits of MetaBots
- Introduction to the MetaBot Designer
- Accessing credentials in a MetaBot
- Running a MetaBot from a task bot

[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-2. Topics

9.1. Overview of MetaBots

Overview of MetaBots

[Introducing MetaBots](#)

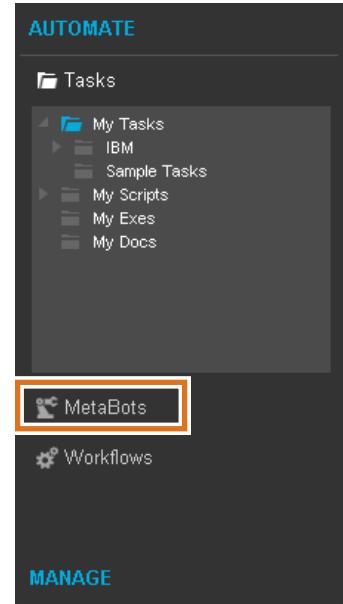
© Copyright IBM Corporation 2018, 2019

Figure 9-3. Overview of MetaBots



Overview of MetaBots

- MetaBots are reusable “blueprints”
- Build MetaBots by using the MetaBot Designer
- Build by using one of these methods:
 - Visual Captures
 - Referred to as a Screen in MetaBot Designer
 - Includes screen components of an application
 - Application APIs
 - Interfaces that support a low-level operation of an application without having to use GUI (screen) components
 - MetaBot Designer also supports the use of DLLs in Windows
 - Navigational Flows
 - Uses visual captures and APIs to create a pre-configured use case of an application
 - Known as Logic in MetaBot Designer
- Access MetaBots from the MetaBots menu in Enterprise Client



[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-4. Overview of MetaBots

9.2. Benefits of MetaBots

Benefits of MetaBots

[Introducing MetaBots](#)

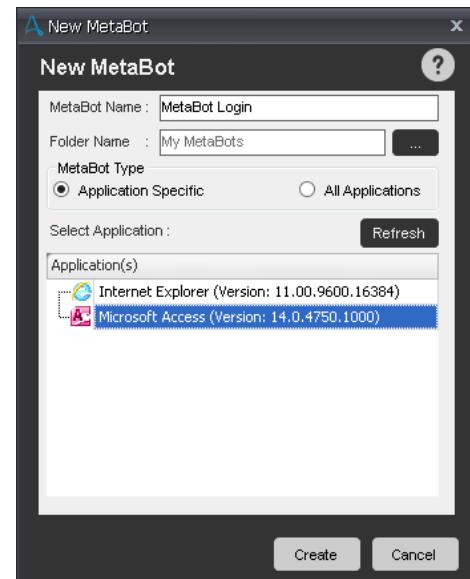
© Copyright IBM Corporation 2018, 2019

Figure 9-5. Benefits of MetaBots



Benefits of MetaBots (1 of 2)

- MetaBots are reusable
 - Can be used by any bot
- MetaBots can be used to standardize organizational-wide automation patterns
 - Helpful for meeting compliance standards
- Use MetaBots in place of scripting or macros
- Automation can take place at any level:
 - API (application programming interface)
 - Front-end
 - Back-end
 - Image recognition
- Having a shareable MetaBot library can force adherence to standard operating procedures
- After a MetaBot is created, there is no need to have access to the live application



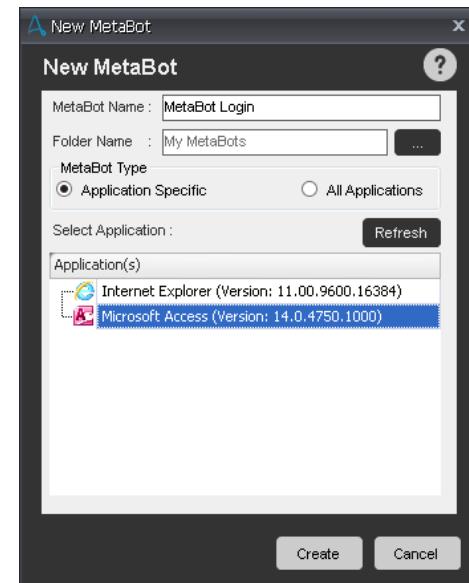
[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-6. Benefits of MetaBots (1 of 2)

Benefits of MetaBots (2 of 2)

- MetaBots can be used to build custom commands
- MetaBots can be built based on a target API from a vendor
- MetaBots contain both the object and the interaction with the object
- MetaBots can extend the command library
- Benefits of MetaBots built on screens include:
 - Reusability across bots
 - Change is encapsulated within the MetaBot
 - No need to have access to live application
 - Calibration of updated screens allows the bot to determine changes in new user interfaces



Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-7. Benefits of MetaBots (2 of 2)

9.3. Introduction to the MetaBot Designer

Introduction to the MetaBot Designer

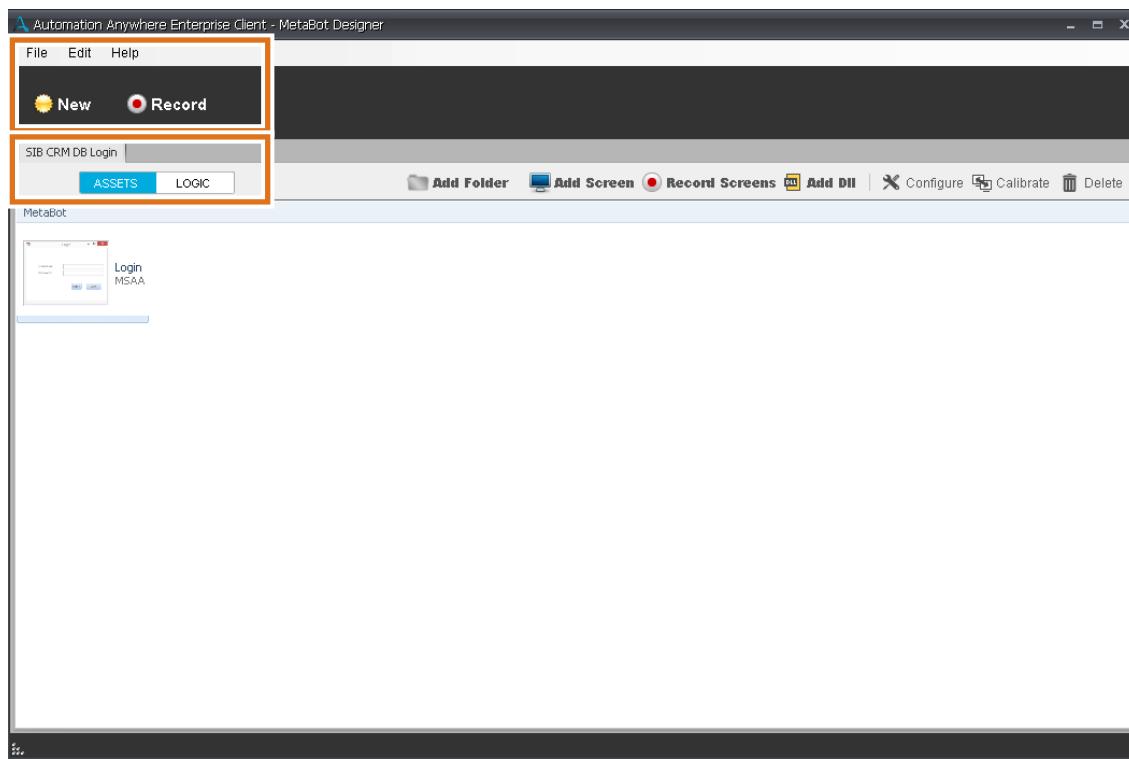
[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-8. Introduction to the MetaBot Designer



Introduction to the MetaBot Designer



Introducing MetaBots

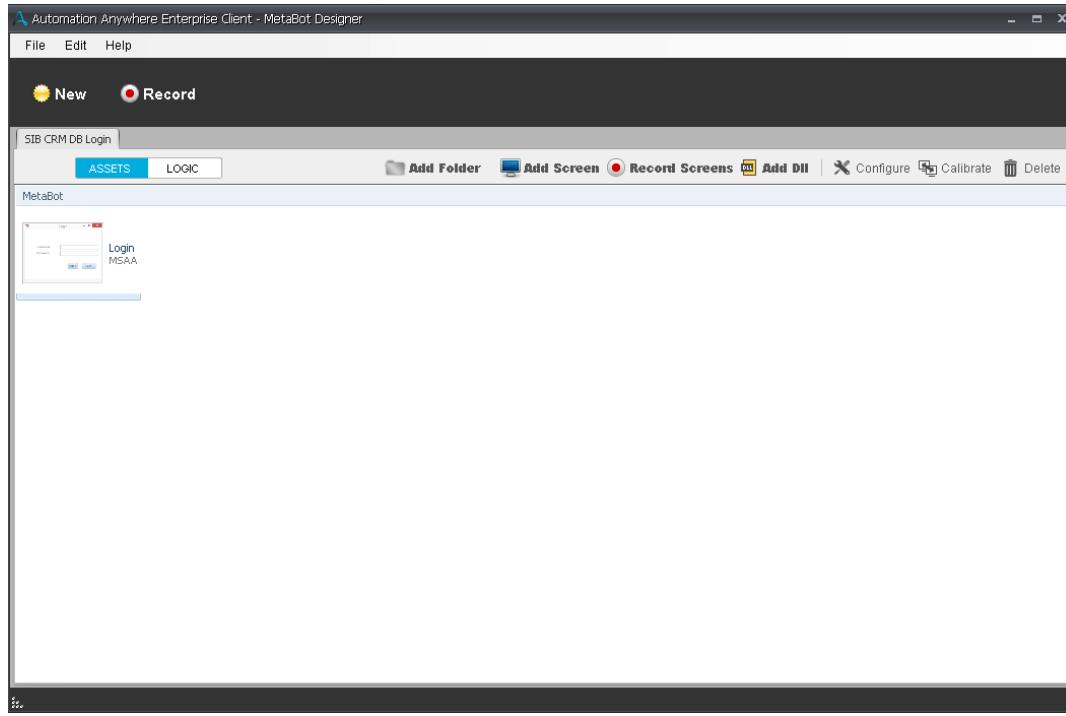
© Copyright IBM Corporation 2018, 2019

Figure 9-9. Introduction to the MetaBot Designer



Introduction to the MetaBot Designer – Assets tab

- Assets include screens, DLLs or both



Introducing MetaBots

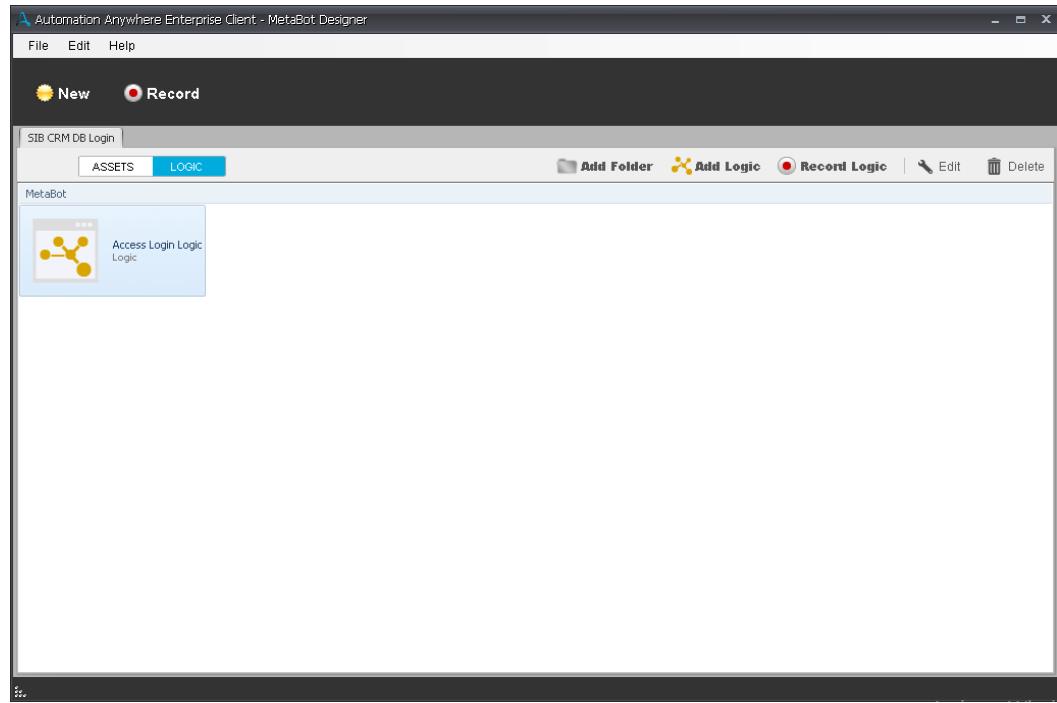
© Copyright IBM Corporation 2018, 2019

Figure 9-10. Introduction to the MetaBot Designer – Assets tab



Introduction to the MetaBot Designer – Logic tab

- Logic represents navigational flows that use screens, DLLs, or both



Introducing MetaBots

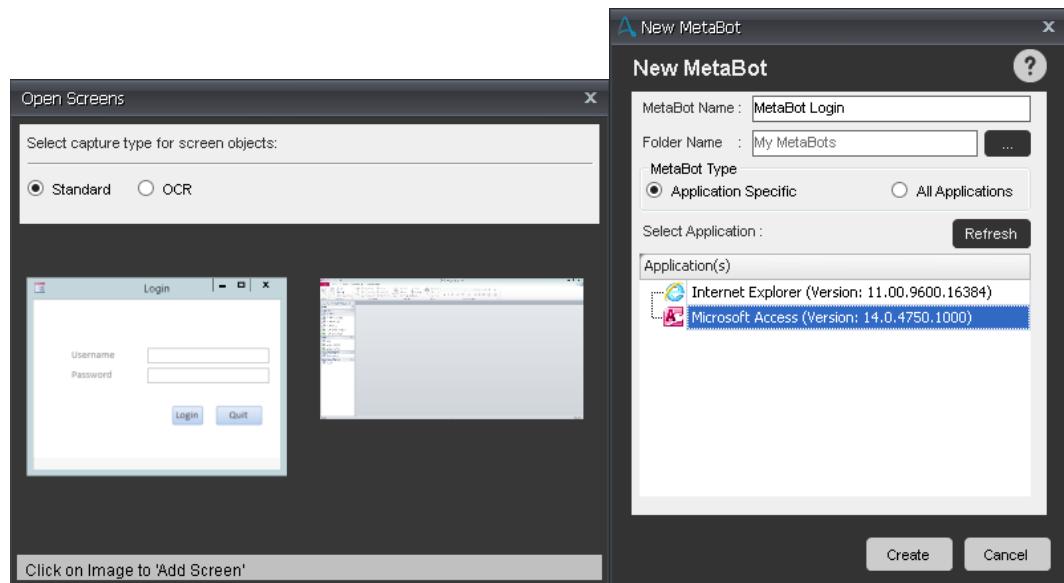
© Copyright IBM Corporation 2018, 2019

Figure 9-11. Introduction to the MetaBot Designer – Logic tab



Creating a MetaBot (1 of 5)

- Screen-based MetaBots are especially useful to encapsulate the logic necessary for screens that are reused across multiple bots
- Example: Log in screen for an application



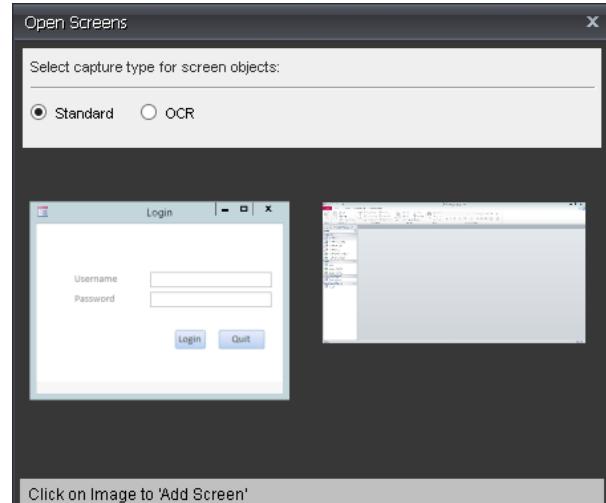
Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-12. Creating a MetaBot (1 of 5)

Creating a MetaBot (2 of 5)

- To create a new MetaBot:
 - Select the MetaBots section of the Folders pane in the Enterprise Client, then click **New**
 - Provide a name and select which application to use
 - A new MetaBot is created and the MetaBot Designer opens
- In the **Assets** tab:
 - Click **Add Screen**
 - Click the image to add the screen as an asset
 - The screen is then added as an asset



Introducing MetaBots

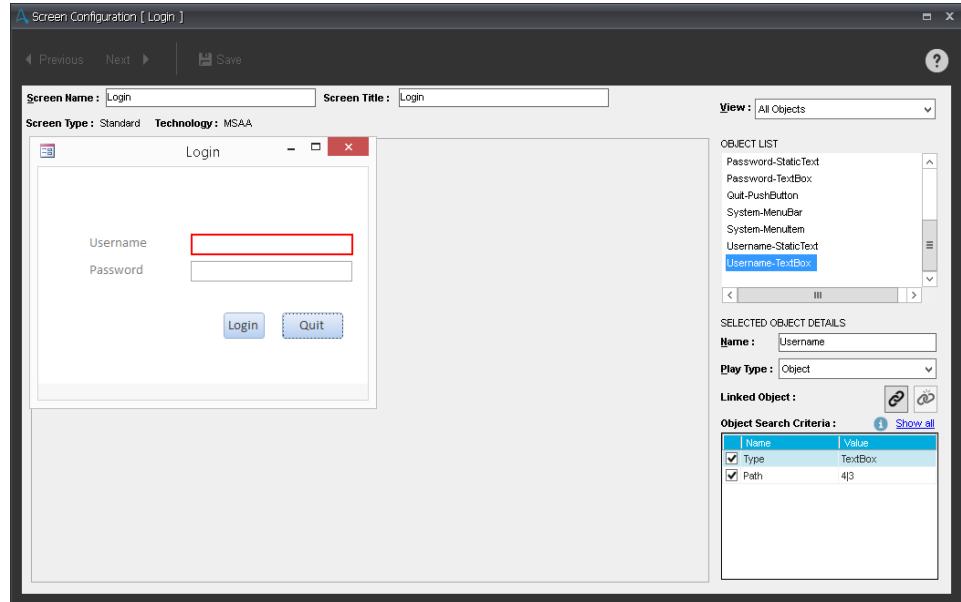
© Copyright IBM Corporation 2018, 2019

Figure 9-13. Creating a MetaBot (2 of 5)



Creating a MetaBot (3 of 5)

- Double-clicking the screen asset after it has been added opens it up in the Screen Configuration user interface
- From here, you can view object details that are found for the screen asset



[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

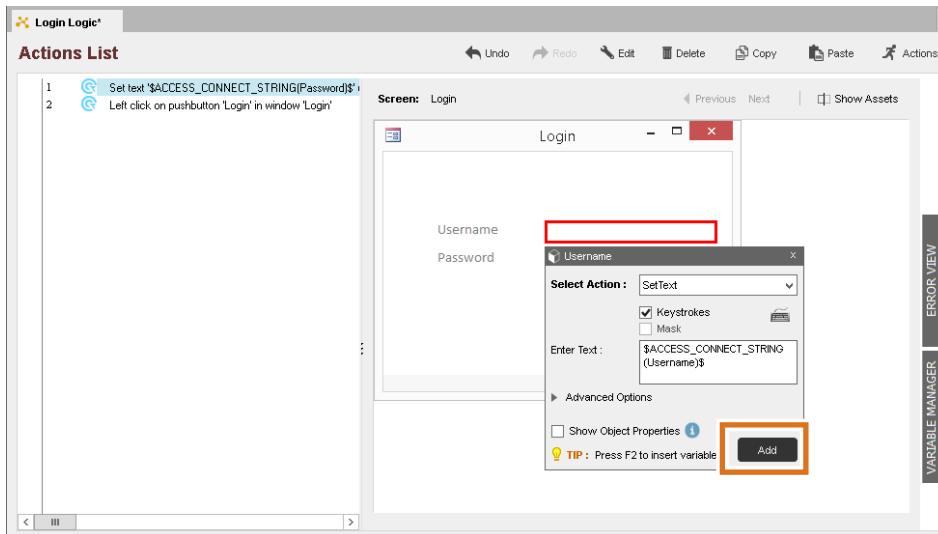
Figure 9-14. Creating a MetaBot (3 of 5)

IBM Training



Creating a MetaBot (4 of 5)

- In the Logic tab of the MetaBot Designer, click **Add Logic**
- In the Logic Editor, right-click the field to be populated and enter either static text or press **F2** to use a variable
- Click the **Keystrokes** box, and click **Add**



Introducing MetaBots

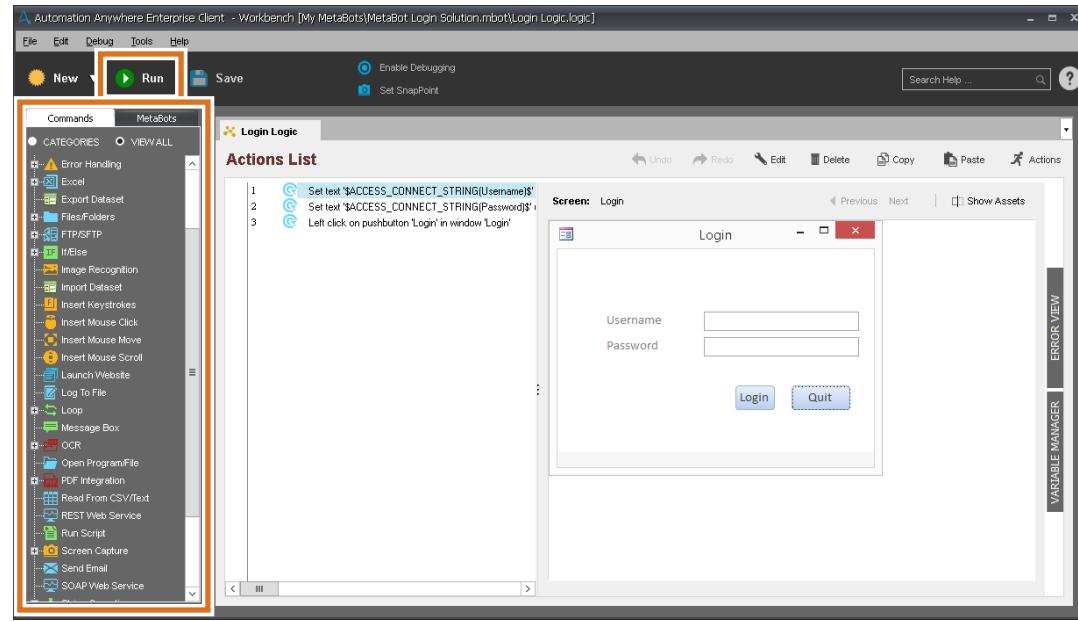
© Copyright IBM Corporation 2018, 2019

Figure 9-15. Creating a MetaBot (4 of 5)



Creating a MetaBot (5 of 5)

- After creating the necessary logic, the **Actions List** is populated
- Click **Run** to verify correct functioning
- MetaBots can use most commands available for Task bots.



Introducing MetaBots

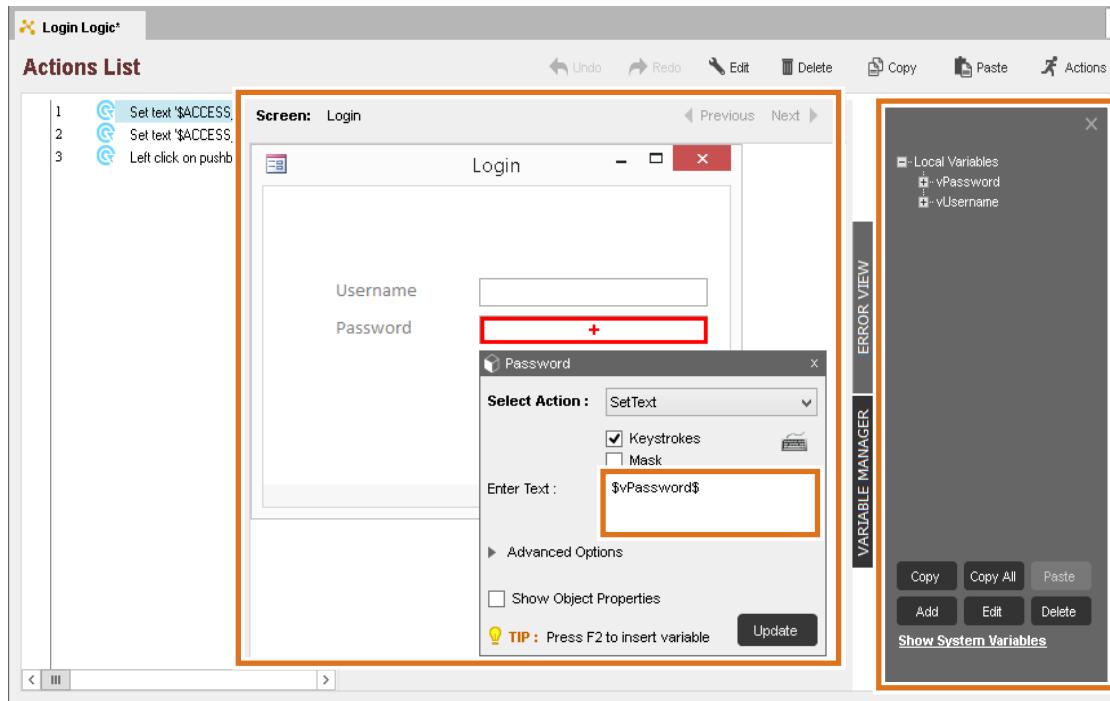
© Copyright IBM Corporation 2018, 2019

Figure 9-16. Creating a MetaBot (5 of 5)



Passing parameters to a MetaBot (1 of 2)

- MetaBots can also have variables



Introducing MetaBots

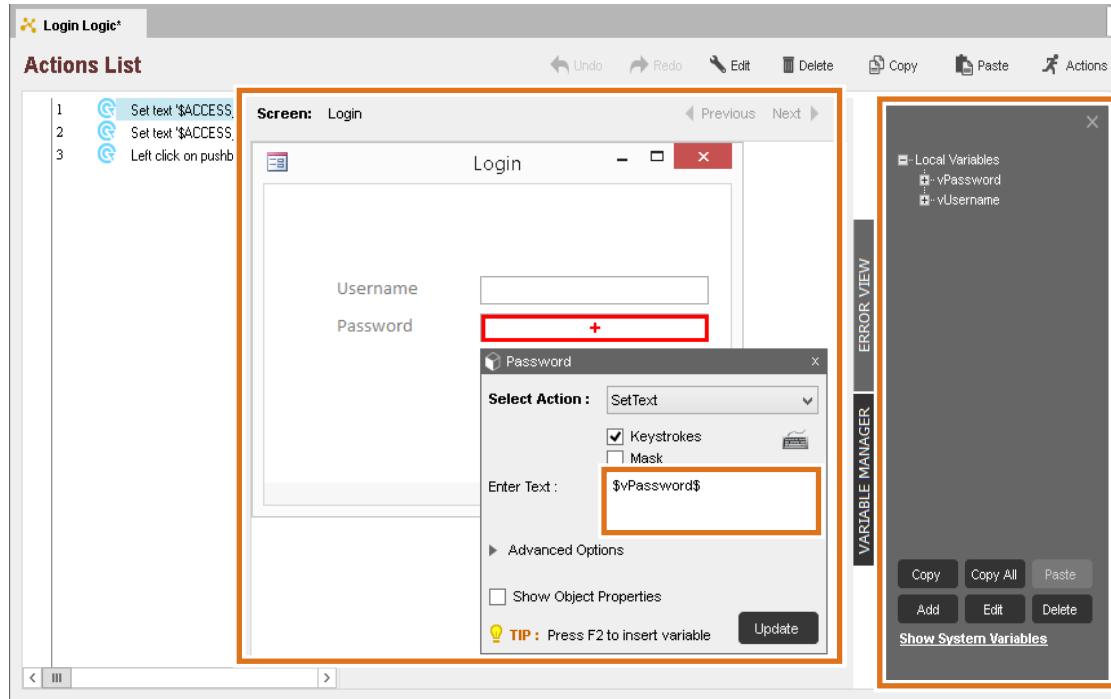
© Copyright IBM Corporation 2018, 2019

Figure 9-17. Passing parameters to a MetaBot (1 of 2)



Passing parameters to a MetaBot (2 of 2)

- Variables can be used to pass data to the MetaBot from a task bot



Introducing MetaBots

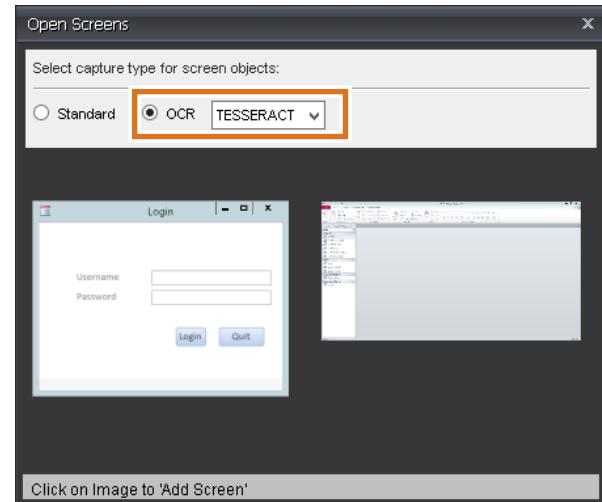
© Copyright IBM Corporation 2018, 2019

Figure 9-18. Passing parameters to a MetaBot (2 of 2)



AISense

- AISense is used in environments in which object based automation is not available or unreliable, for example: Citrix, Remote Desktop.
- AISense uses “computer vision” to identify all user interface elements and creates objects from the image.
- The UI objects are available for automation through MetaBot Logic.
- If system is not able to find the UI element by using OCR due to any problem, it automatically tries to find the same object by using image-based search.
- This saves time by not having to manually create each object.



Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-19. AISense

9.4. Accessing credentials in a MetaBot

Accessing credentials in a MetaBot

[Introducing MetaBots](#)

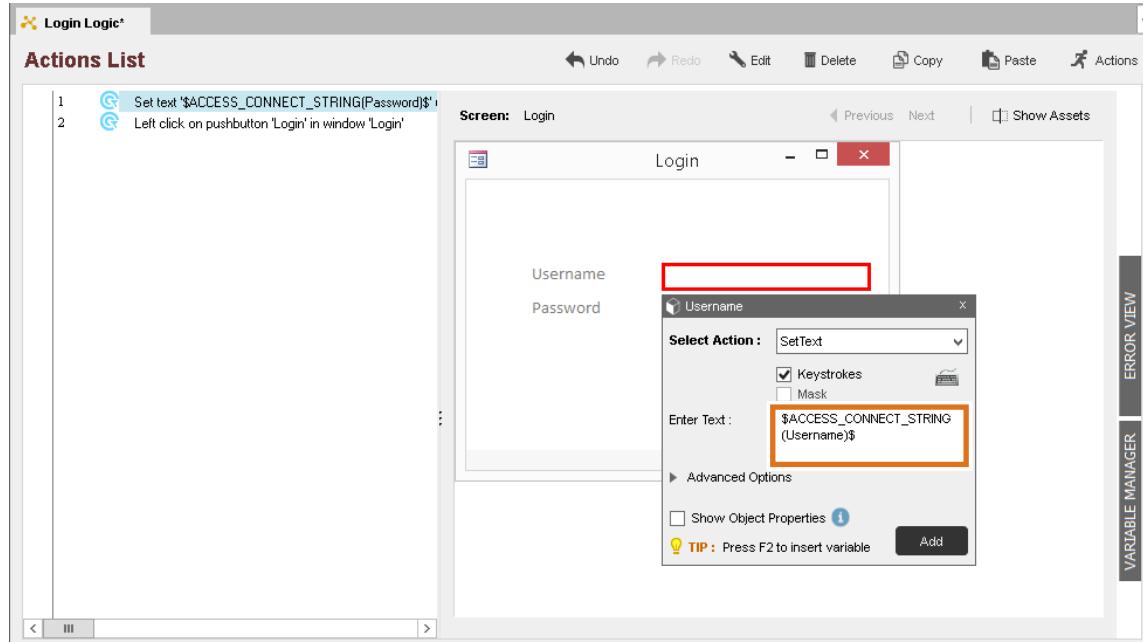
© Copyright IBM Corporation 2018, 2019

Figure 9-20. Accessing credentials in a MetaBot



Accessing credentials in a MetaBot (1 of 2)

- Credentials can be used instead of variables



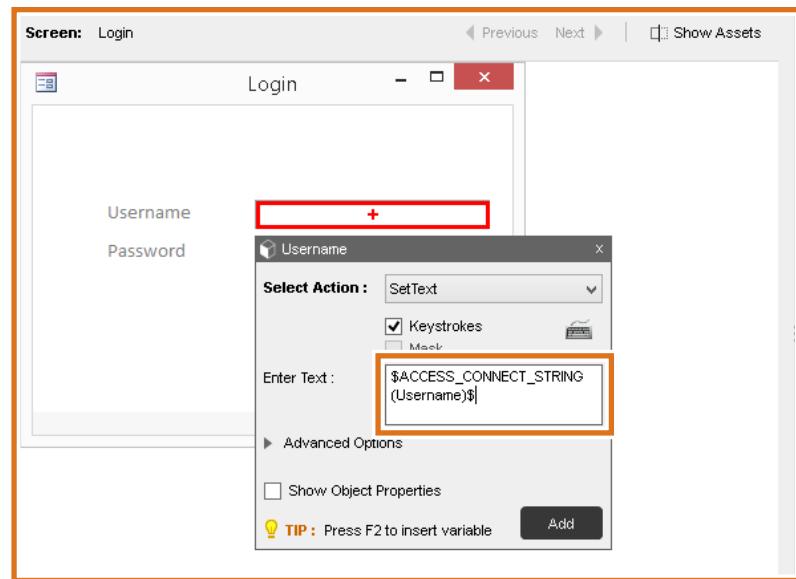
Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-21. Accessing credentials in a MetaBot (1 of 2)

Accessing credentials in a MetaBot (2 of 2)

- By using credentials, the user name and password for a Login screen are managed at the server
- Promotes reusability and maintenance by using the same MetaBot when accessing the same Login screen and credentials



Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-22. Accessing credentials in a MetaBot (2 of 2)

9.5. Running a MetaBot from a task bot

Running a MetaBot from a task bot

[Introducing MetaBots](#)

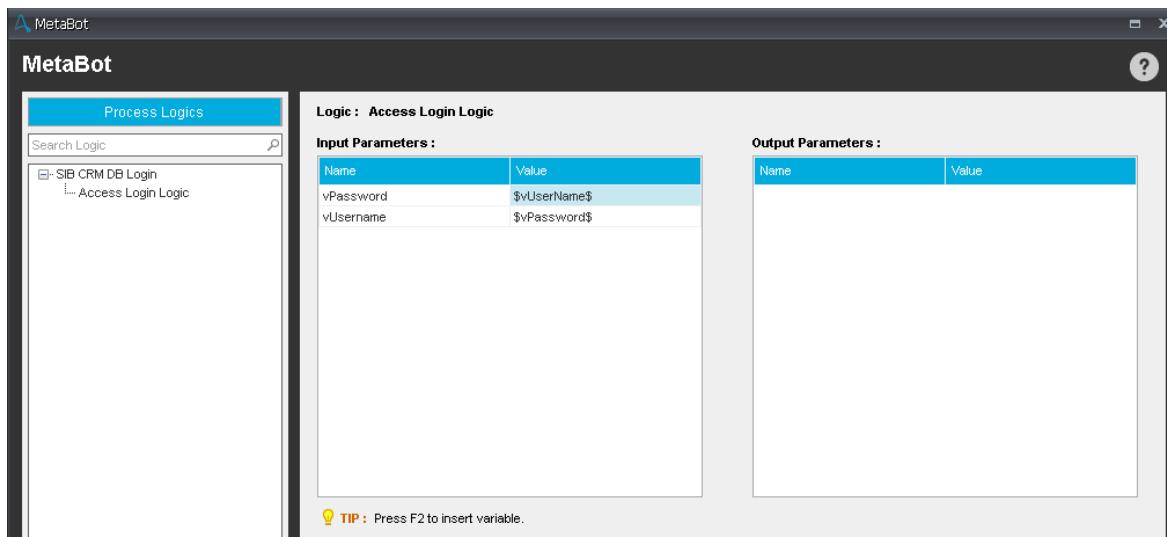
© Copyright IBM Corporation 2018, 2019

Figure 9-23. Running a MetaBot from a task bot



Running a MetaBot from a task bot (1 of 2)

- After you define your MetaBot, you can run the MetaBot from any task bot
- Drag the MetaBot from the MetaBots folder to the bot code in the Workbench
 - A MetaBot window opens with parameters that are defined in the MetaBot



Introducing MetaBots

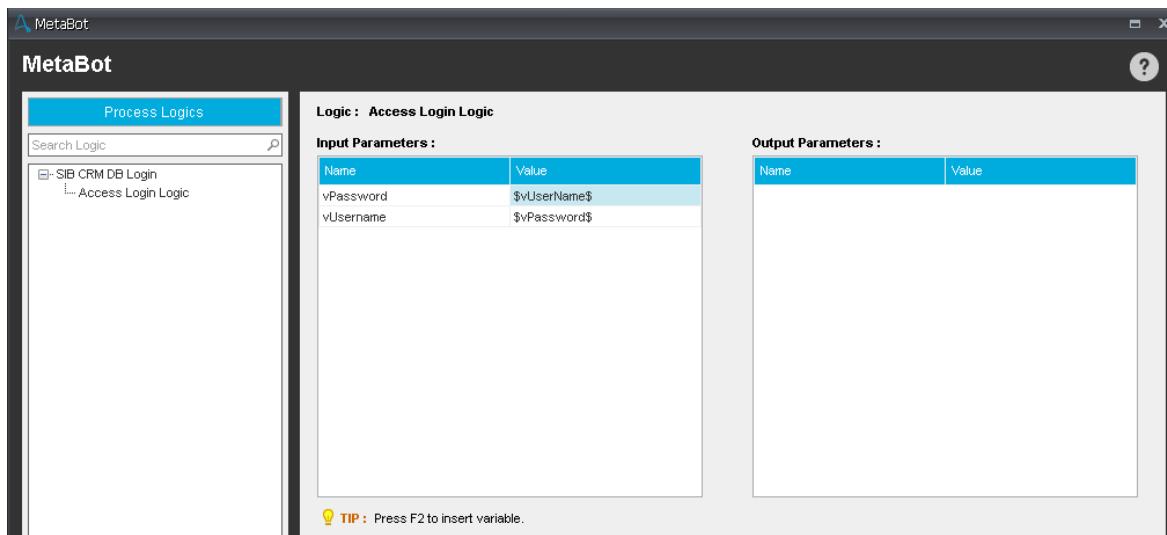
© Copyright IBM Corporation 2018, 2019

Figure 9-24. Running a MetaBot from a task bot (1 of 2)



Running a MetaBot from a task bot (2 of 2)

- In the MetaBot window, you can configure the source for the parameters in the MetaBot
 - These might be variables in the task bot or system credentials
- If the MetaBot was configured by using credentials instead of variables, no parameters appear in this window



Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-25. Running a MetaBot from a task bot (2 of 2)

Unit summary

- Describe how to use MetaBots
- Describe the features and components of the MetaBot Designer
- Explain how to create a MetaBot

[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-26. Unit summary

Review questions

1. True or False: MetaBots are highly reusable and can be used by any task bot.
2. MetaBots and task bots share the following functions **except:**
 - A. Both MetaBots and task bots have variables
 - B. Both MetaBots and task bots can use credentials
 - C. Both MetaBots and task bots are application resilient by using calibration
 - D. None of the above
3. The following statements about MetaBots are true **except:**
 - A. MetaBots are made up of assets and logic
 - B. Assets include screens and DLLs
 - C. Logic represents navigational flows that use screens and DLLs
 - D. MetaBots contain metadata about task bots



Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-27. Review questions

Review answers



1. True or False: MetaBots are highly reusable and can be used by any task bot.
The answer is True.

2. MetaBots and task bots share the following functions **except**:
 - A. Both MetaBots and task bots have variables
 - B. Both MetaBots and task bots can use credentials
 - C. Both MetaBots and task bots are application resilient by using calibration
 - D. None of the aboveThe answer is C.

3. The following statements about MetaBots are true **except**:
 - A. MetaBots are made up of assets and logic
 - B. Assets include screens and DLLs
 - C. Logic represents navigational flows that use screens and DLLs
 - D. MetaBots contain metadata about task botsThe answer is D.

Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-28. Review answers

Exercise 8: Creating a login MetaBot

Introducing MetaBots

© Copyright IBM Corporation 2018, 2019

Figure 9-29. Exercise 8: Creating a login MetaBot

Exercise objectives

- Create a MetaBot to handle application login
- Reuse the MetaBot in an existing task bot



[Introducing MetaBots](#)

© Copyright IBM Corporation 2018, 2019

Figure 9-30. Exercise objectives

Unit 10. Working with web services

Estimated time

01:00

Overview

This unit describes how to work with web services.

How you will check your progress

- Checkpoint

Unit objectives

- Call web services in a bot

Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-1. Unit objectives

Topics

- SOAP Web Service command
- REST Web Service command

Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-2. Topics

10.1. SOAP Web Service command

SOAP Web Service command

Working with web services

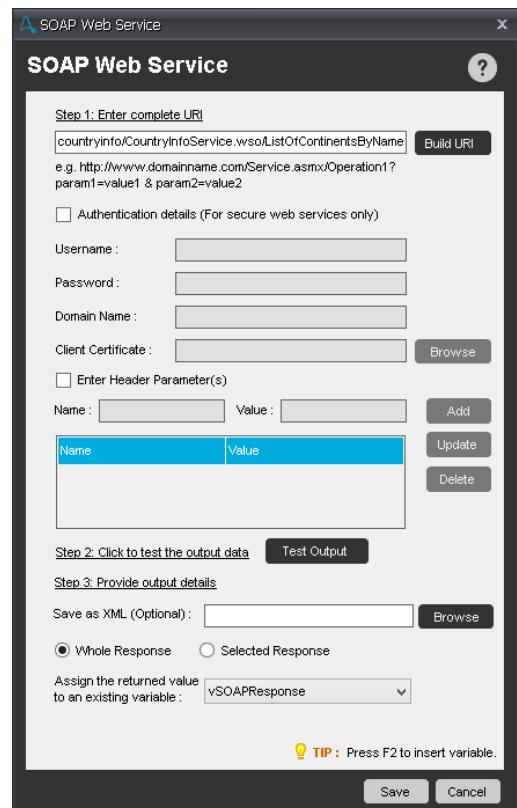
© Copyright IBM Corporation 2018, 2019

Figure 10-3. SOAP Web Service command



SOAP Web Service command

- Used to connect to applications and services over the internet
- Support exists for:
 - XML (Extensible Markup Language)
 - WSDL (Web Services Description Language)
 - SOAP (Simple Object Access Protocol)
 - UDDI (Universal Description Discovery and Integration)
- Supports:
 - Content Type: Text/XML
 - Encoding Type: UTF-8 Format
 - Authentication Type: Basic authentication



Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-4. SOAP Web Service command

10.2. REST Web Service command

REST Web Service command

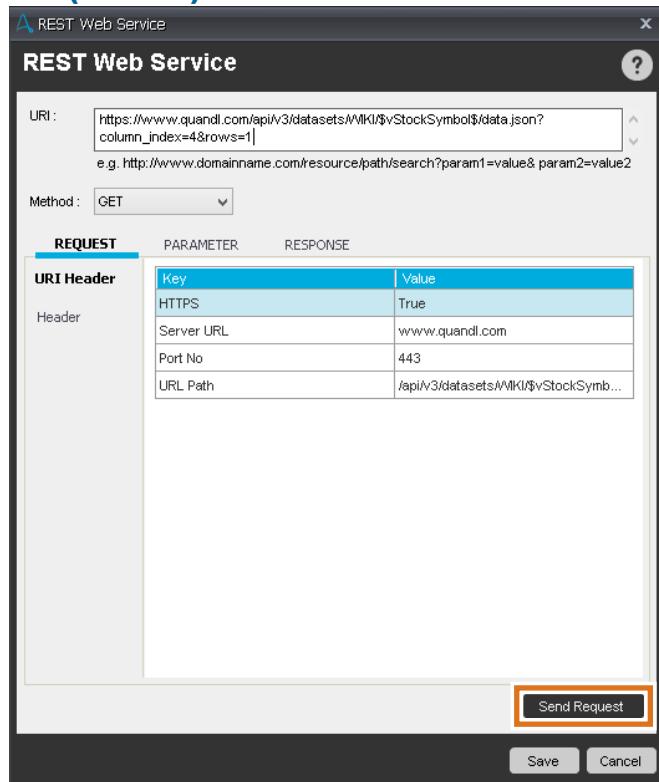
Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-5. REST Web Service command

REST Web Service command (1 of 3)

- The REST (Representational State Transfer) command supports the following methods:
 - GET
 - POST
 - PUT
 - DELETE
- To build a REST call, enter the **URI**, select the **GET** method, and click **Send Request** to verify the request
- If the request has a parameter, a prompt opens with the value of the parameter



Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-6. REST Web Service command (1 of 3)



REST Web Service command (2 of 3)

- Clicking **Get output** executes the call
- The **RESPONSE** tab opens with key/value pairs for the Header along with the Body of the response

The screenshot shows two windows. The main window is titled "REST Web Service" and contains a "REQUEST" tab with fields for "URI" (set to "https://www.quandl.com/api/v3/datasets/MKI/\$vStockSymbol\$/data.json?column_index=4&rows=1") and "Method" (set to "GET"). Below the REQUEST tab are sections for "URI Header" and "Header". A "RESPONSE" tab is visible but empty. The second window is titled "Variable(s) Value" and has a table with one row: "Name" \$vStockSymbol\$ and "Value" IBM. At the bottom of this window is a note: "NOTE: Credential variable(s) are excluded in the above list". There are "Get output", "Cancel", and "Save" buttons at the bottom. An orange box highlights the "Get output" button. The entire interface is set against a dark background.

Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-7. REST Web Service command (2 of 3)



REST Web Service command (3 of 3)

- If the request is valid, the results appear in the **RESPONSE** tab
- It's common to use the **XML** or **String Operation** commands with a web service to parse the results

The screenshot shows two overlapping windows. The top window is titled "REST Web Service" and displays a REST API call to Quandl. The bottom window is titled "String Operation" and shows a configuration for extracting a substring from a source string.

REST Web Service Window:

- URI:** https://www.quandl.com/api/v3/datasets/MKI/\$vStockSymbol\$/data.json?column_index=4&rows=1
- Method:** GET
- RESPONSE:** (Content shown in the screenshot)

String Operation Window:

- Select Operation:** Before-After
- Note:** Returns the sub string enclosed by 'Before' and 'After' strings.
- Source String:** \$vStockReturn\$
- Before (Optional):** "data":[]
- Occurrence (Optional):** 1
- Logical Operator:** AND
- After (Optional):**]],"collapse":null,"order":null}
- Occurrence (Optional):** 1
- No. of characters to be extracted:** 1
- If no match found, return:** Source String Empty (Null) String
- Assign the output to variable:** vStockReturn

Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-8. REST Web Service command (3 of 3)

Unit summary

- Call web services in a bot

Working with web services

© Copyright IBM Corporation 2018, 2019

Figure 10-9. Unit summary

Review questions



1. True or False: Bots can use both SOAP and REST web services.
2. True or False: Both SOAP and REST web service results can be saved to a variable.
3. To configure a REST Get command in a bot, the following information is required:
 - A. REST URI
 - B. REST Header Key \ Value pairs
 - C. WSDL document
 - D. Swagger document

Review answers



1. True or False: Bots can use both SOAP and REST web services.
The answer is True
2. True or False: Both SOAP and REST web service results can be saved to a variable.
The answer is True
3. To configure a REST Get command in a bot, the following information is required:
 - A. REST URI
 - B. REST Header Key \ Value pairs
 - C. WSDL document
 - D. Swagger documentThe answer is A.

Unit 11. Exceptions and error handling

Estimated time

01:00

Overview

This unit provides an overview of some of the basic error handling features that you can use in a bot.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Add error handling to a bot
- Evaluate error codes that are received in a web service
- Display a variable as currency

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-1. Unit objectives

Topics

- Error handling concepts
- Error Handling command
- Error View
- Evaluate error codes that are received in a web service call
- Display a variable as currency

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-2. Topics

11.1. Error handling concepts

Error handling concepts

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-3. Error handling concepts

Error handling concepts

- Error handling is a common programming construct that involves the detection and resolution of errors that occur during normal processing
- It is important that any bot that goes into production has an error handling component
- Global error handling is used to detect all errors that occur in the bot during normal processing
- If potential errors are known ahead of time, the bot can detect and handle them
 - For example, if the bot is unable to reach a web service, the bot can email the support desk to inform them that the service is unavailable

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-4. Error handling concepts

11.2. Error Handling command

Error Handling command

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

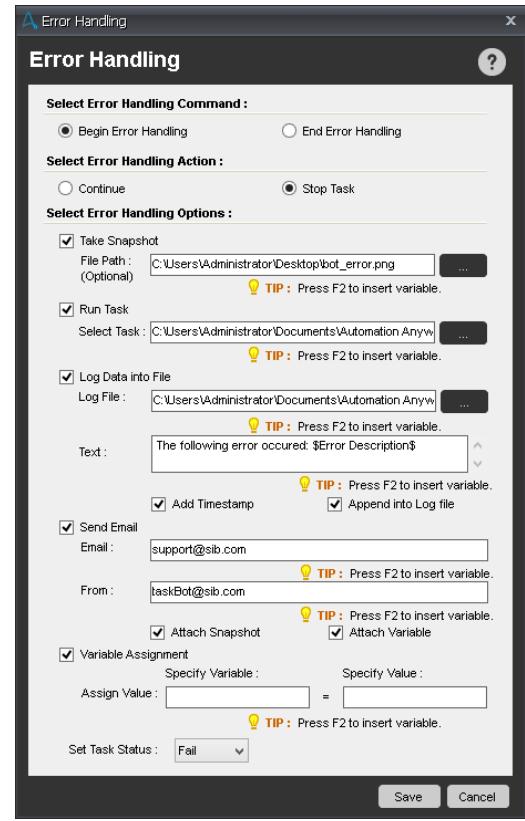
Figure 11-5. Error Handling command



Error Handling command

- The **Error Handling** command is used to intercept and respond to errors (global error handling)
- When an error occurs, you can choose how the bot should respond to the error:
 - Error handling action:
 - Continue
 - Stop Task
 - Error handling options:
 - Take snapshot
 - Run Task
 - Log Data into File
 - Send Email
 - Variable Assignment
- When choosing to send an email, you can also choose to attach the following to the email:
 - Snapshot – screen capture of the bot
 - Variable – a text file with all variable values (system and user defined)

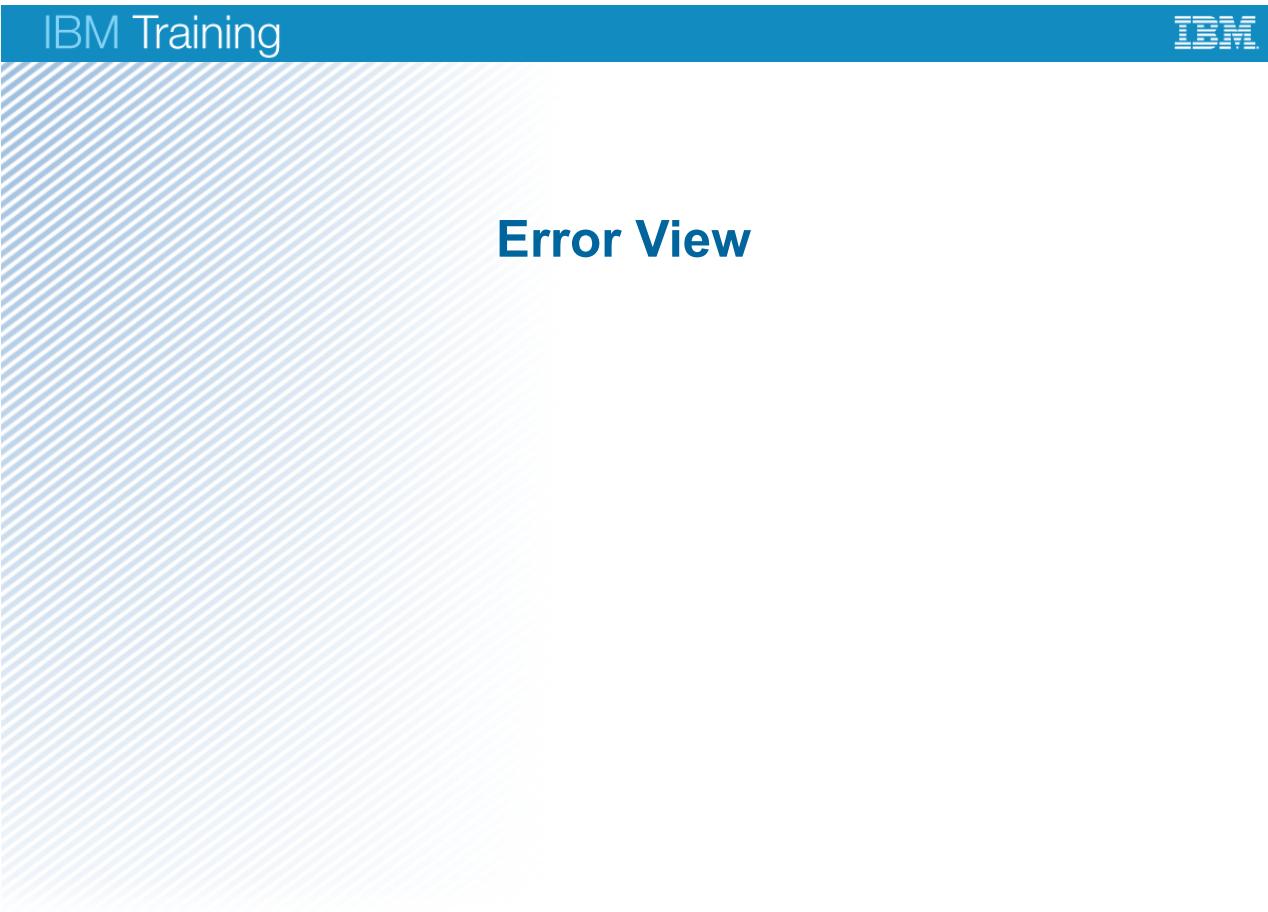
Exceptions and error handling



© Copyright IBM Corporation 2018, 2019

Figure 11-6. Error Handling command

11.3. Error View



Exceptions and error handling

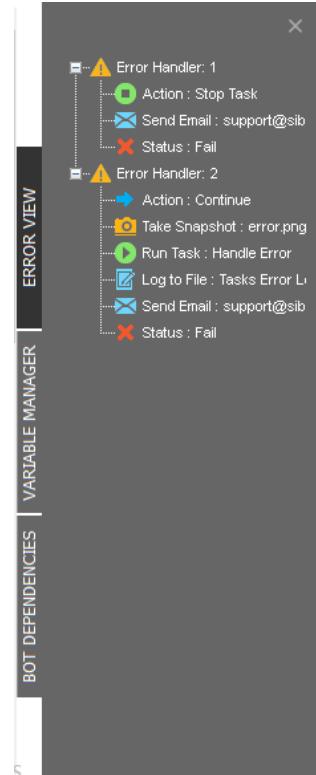
© Copyright IBM Corporation 2018, 2019

Figure 11-7. Error View



Error View

- You can use any or all of these options in any combination
 - For example, you can stop the bot and then send an email with the snapshot attached
- As you add error handling to your bot, you can use the Error View tab to see a breakdown of all errors configured
- The Error View is especially helpful for bots that have multiple error handling commands
- This view is automatically updated when **Error Handling** commands are added to the bot



Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-8. Error View

11.4. Evaluate error codes that are received in a web service call

Evaluate error codes that are received in a web service call

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-9. Evaluate error codes that are received in a web service call

Evaluate error codes that are received in a web service call

- The **XML** and **String Operation** commands are commonly used with a web service to parse the results is common and with error handling
- Consider the following scenario:
 - Quandl provides a free REST web service that provides after-trade day prices for stock symbols
 - Quandl responds with an error code: QECx02, if the stock symbol is bad or not found
 - If the bot is aware of the error, it can parse the error results (by using the **String Operation** command), evaluate the error code, and respond as needed

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-10. Evaluate error codes that are received in a web service call

11.5. Display a variable as currency

Display a variable as currency

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-11. Display a variable as currency

How to display a variable as currency (1 of 2)

- It is important to be aware that while using the **Variable Operation** command in the Workbench, values do not contain zeros
- Example: If the value “04.50” is assigned to a variable, the value that is held in memory is 4.5
 - Works fine for calculations but does not work well for display
- To display a variable as currency with decimal places for example, the bot needs to:
 - Evaluate the value
 - Determine whether a decimal needs to be added or if zeros need to be added

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-12. How to display a variable as currency (1 of 2)

How to display a variable as currency (2 of 2)

- Evaluate the value by extracting values to evaluate decimal length
 - Assign values after decimal to a variable
 - Assign the length of that variable to another variable to evaluate length
- Evaluate need for decimal and concatenate as necessary

```
If variable decimal length = 1  
    Concatenate 0 to variable  
Else If variable length = 0 or > 2  
    Concatenate .00 to variable  
End If
```

Figure 11-13. How to display a variable as currency (2 of 2)

Unit summary

- Add error handling to a bot
- Evaluate error codes that are received in a web service
- Display a variable as currency

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-14. Unit summary

Review questions



1. True or False: Bots need to be aware of what the error is to be able to respond.
2. True or False: Not all bots that go to production need to have error handling.
3. Which command is used to extract a substring of a web service response to evaluate an error code?
 - A. Variable Operation
 - B. String Operation
 - C. REST Web Service
 - D. SOAP Web Service

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-15. Review questions

Review answers

1. True or False: Bots need to be aware of what the error is to be able to respond.

The answer is False



2. True or False: Not all bots that go to production need to have error handling.

The answer is False

3. Which command is used to extract a substring of a web service response to evaluate an error code?

- A. Variable Operation
- B. String Operation
- C. REST Web Service
- D. SOAP Web Service

The answer is B.

Exercise 9: Working with web services and error handling

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-17. Exercise 9: Working with web services and error handling

Exercise objectives

- Implement basic error handling in a bot
- Work with a REST service to incorporate web service data in a bot
- Implement more complicated conditional (If/Else) logic in a bot

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-18. Exercise objectives

Exercise introduction (1 of 2)



- In this exercise, you implement error handling at various levels, including detecting all bot errors and detecting two specific REST service errors
- Two levels of error handling are introduced:
 - Detecting a specific error code that is returned from a web service call
 - Detecting “all” other errors
- This exercise builds on the Trade Booking process bot and includes an extra scenario in which the data for the stock symbol is incorrect

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-19. Exercise introduction (1 of 2)

Exercise introduction (2 of 2)

The following commands are introduced in this exercise:



Command	Purpose
Error Handling	Handle any errors that occur during normal processing.
REST Web Service	Execute a GET method to obtain a stock price from a web service

Exceptions and error handling

© Copyright IBM Corporation 2018, 2019

Figure 11-20. Exercise introduction (2 of 2)

Unit 12. Hardening bots against exceptions

Estimated time

01:00

Overview

In this unit, you learn good practices for defensively coding bots against exceptions.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe good practices for hardening bots against exceptions

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-1. Unit objectives

Topics

- Bot hardening concepts
- Bot hardening scenarios and suggested practices

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-2. Topics

12.1. Bot hardening concepts

Bot hardening concepts

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-3. Bot hardening concepts

Bot hardening concepts

- There are many places a bot can run into problems during normal processing, especially for browser-based automations
- Following good practices can help prevent issues when the bot runs but many things can't be planned for
- Never assume the design-time environment mirrors exactly the runtime environment
- The key to hardening a bot is predicting and handling both expected and unexpected events
- The Error Handling command is used to detect unexpected errors throughout the bot
- Any one particular hardening scenario to plan for can be dealt with multiple ways

12.2. Bot hardening scenarios and suggested practices

Bot hardening scenarios and suggested practices

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-5. Bot hardening scenarios and suggested practices

Bot hardening example scenarios (1 of 7)

- In browser-based automations, any combination of the following issues might happen:
 - Web page title changes
 - Web page layout changes
 - Web page doesn't appear when the bot expects it
- Possible approaches to address these issues are described next, based on the browser-based automation example

Hardening bots against exceptions

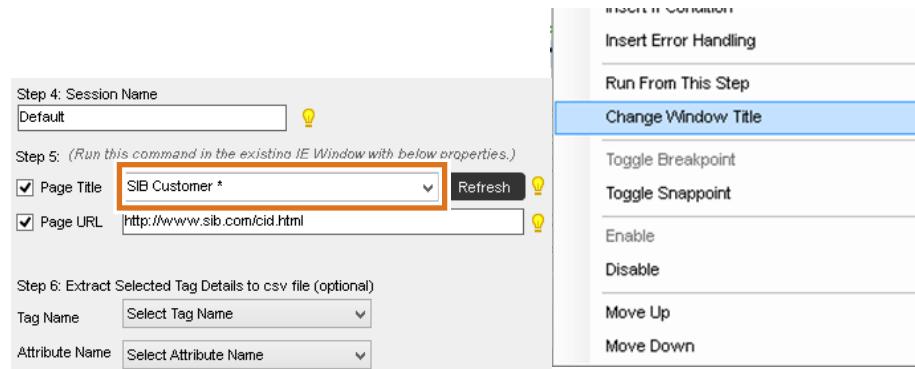
© Copyright IBM Corporation 2018, 2019

Figure 12-6. Bot hardening example scenarios (1 of 7)



Bot hardening example scenarios (2 of 7)

- Web page title changes
 - At design-time, right-click multiple Object Cloning commands and select “Change Window Title”
 - At run time, use wildcards (*) with Window names



Hardening bots against exceptions

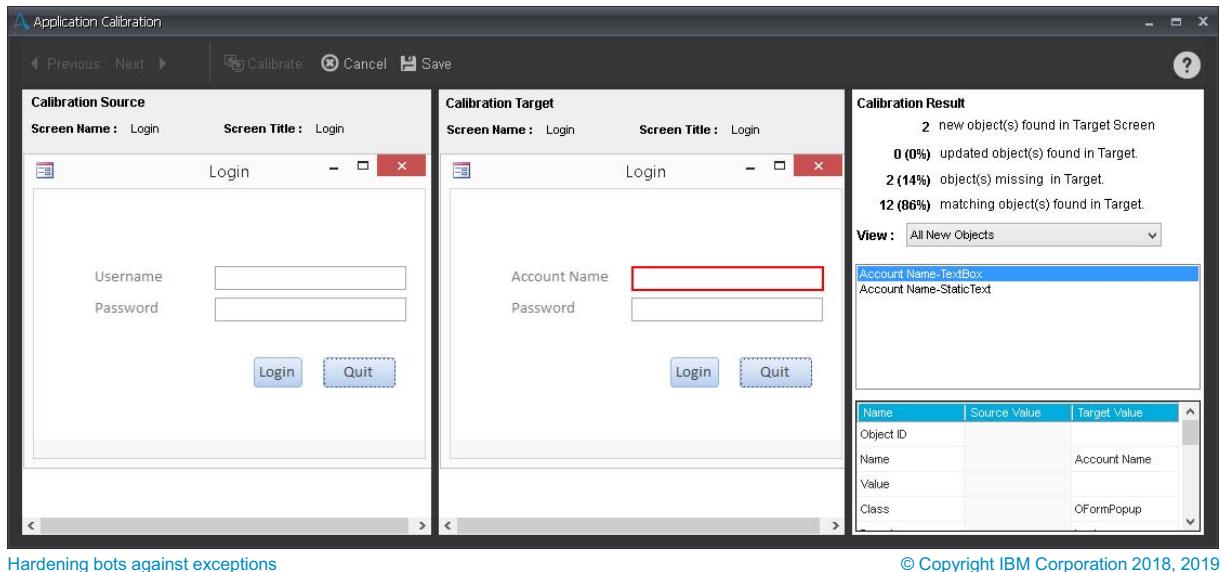
© Copyright IBM Corporation 2018, 2019

Figure 12-7. Bot hardening example scenarios (2 of 7)



Bot hardening example scenarios (3 of 7)

- Web page layout changes
 - If the bot knows ahead of time where to find the information, it can evaluate the layout first then determine how to find the information
 - A MetaBot could be built that would calibrate new user interface changes



Hardening bots against exceptions

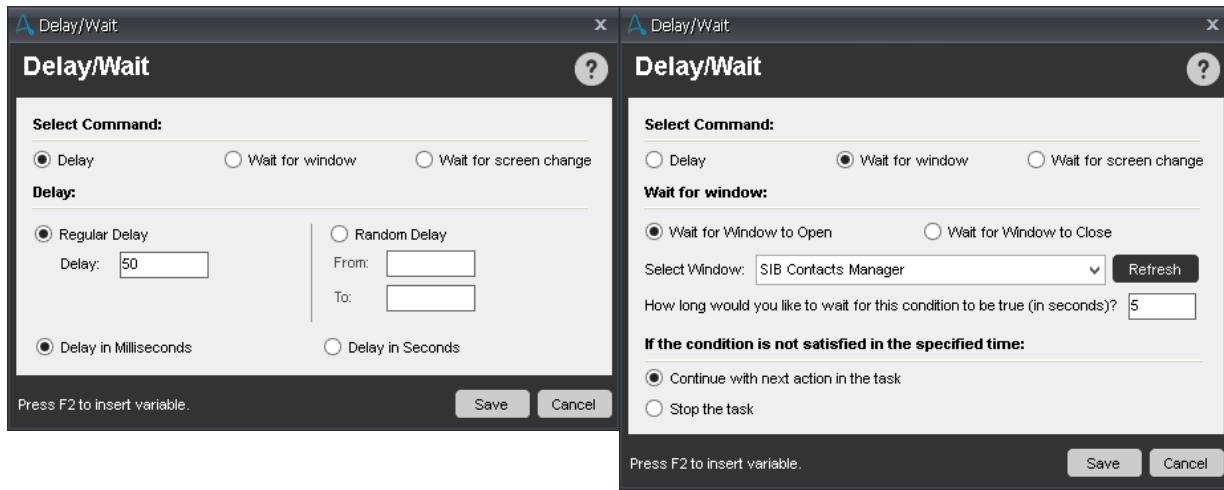
© Copyright IBM Corporation 2018, 2019

Figure 12-8. Bot hardening example scenarios (3 of 7)



Bot hardening example scenarios (4 of 7)

- Web page doesn't appear when the bot expects it
 - The **Delay/Wait** command might be used to create a delay before attempting to open the Window
 - The **Delay/Wait** command might be used to wait for the Window to open



Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-9. Bot hardening example scenarios (4 of 7)

Bot hardening example scenarios (5 of 7)

- How to handle expected errors varies depending on the type of error
- Example: Excel spreadsheet scenarios
 - Spreadsheet has incorrect extension (XLS instead of XLSX)
 - Location of source column changes
- If this information is known ahead of time, the bot can respond
- Example:
 - Spreadsheet has incorrect extension
 - Detect file extension of spreadsheet
 - IF file extension equals XLS THEN
 - rename the file with an XSLX file extension
 - Continue normal processing
 - Location of source column changes
 - Determine whether the source column is in the correct location
 - IF source the column is not in the correct location THEN
 - Locate source column and move to correct location
 - Continue normal processing
- Next: How to implement this in a bot



Bot hardening example scenarios (6 of 7)

- Spreadsheet has incorrect extension
 - Use the **If/Else** command to determine whether the file with the incorrect extension exists
 - If it does, use the **File/Folders Rename Files** subcommand to rename the file

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-11. Bot hardening example scenarios (6 of 7)



Bot hardening example scenarios (7 of 7)

- Location of source column changes
 - Use the **Excel Find/Replace** subcommand to find the source column
 - Use the **If/Else** command to determine whether it is in the correct location
 - If the column is not in the correct location, use the Smart Recorder to move the column to the correct location

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-12. Bot hardening example scenarios (7 of 7)

Run a bot from another bot

- Modularization is important for complex bots
- Create small reusable bots that complete discreet functions instead of large bots that perform many functions
 - This is called “nesting tasks”
- When you run a bot from another bot, a bot dependency is created

Good practices for running a bot from another bot

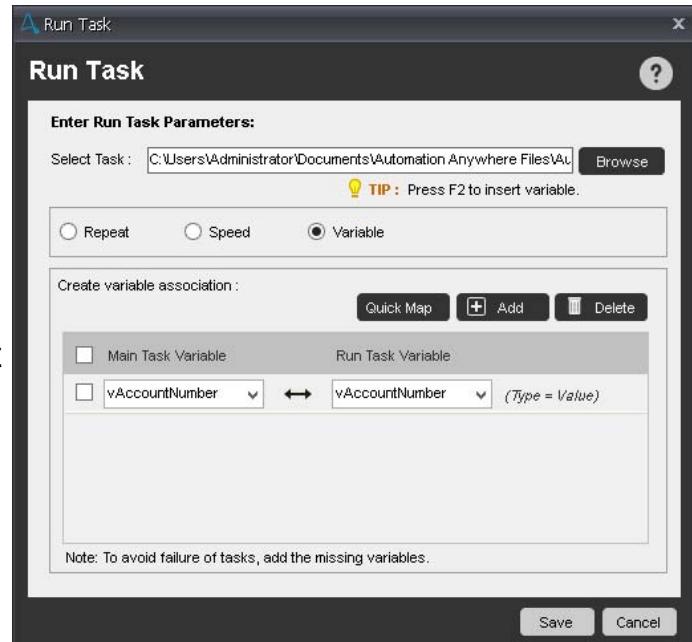
- “Don’t Repeat Yourself” principle
 - The point of this practice is to reduce the repetition of code
 - In bot development, this refers to creating sets of logic that are repeated throughout a single bot or across multiple bots
 - Running sub bots from a parent bot promotes reusability by sharing logic across multiple bots
 - Subbots are often referred to as a “Help Task” or “Utility Task” since their purpose is supporting the parent bot
- Decoupling and loose coupling
 - Task dependency needs to be minimized as much as possible
 - When building bots, this is often unavoidable
 - Developing bots with this in mind can improve bot performance and aid in bot management
- Avoid bidirectional dependencies
 - As much as possible, a calling bot should not have a dependency on the calling bot
 - This can create a recursive dependency that might lead to problems in both execution and management

Figure 12-14. Good practices for running a bot from another bot



Run Task command

- To run a bot from another bot, you use the **Run Task** subcommand of the **Task** command
- You can add variables as parameters to pass to the bot
- If the variables have the same names in both bots, you can click **Quick Map** to map variables
- It is important to manage bot dependencies when running one bot from another bot
 - Covered in more detail later in this course



Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-15. Run Task command

Challenge: Create an Error Handler bot (1 of 4)

- In this scenario, a customer would like to use a reusable error handling bot that could be shared across other bots to standardize error handling
- As a bot Developer, you must build a reusable bot that is decoupled from the calling bots and handles errors from the calling bot
- Based on the good practices that were described earlier, how could this challenge be implemented?

Challenge: Create an Error Handler bot (2 of 4)

- Consider the following two scenarios and whether good practices are being applied
- Create a bot that receives the error from the calling bot
 - The Don't Repeat Yourself Principle: YES. Code is not repeated
 - Decoupling and Loose Coupling: PARTIAL. A bot dependency is created
 - Avoid Bidirectional Dependencies: YES. No bidirectional dependency is created
- Create a bot that is triggered when a log file is created from the calling bot
 - The Don't Repeat Yourself Principle: YES. Code is not repeated
 - Decoupling and Loose Coupling: YES. A bot dependency is not created
 - Avoid Bidirectional Dependencies: YES. No bidirectional dependency is created



Challenge: Create an Error Handler bot (3 of 4)

- The second scenario creates less coupling between the two bots
- How this would look in code:
 - You would still need an **Error Handling** command to detect the error and write the log
 - Then, you would create a Handle Errors bot that is triggered when the file is modified

The screenshot shows two windows side-by-side. On the left is the 'Trigger' configuration window under the 'PROPERTIES' tab. It has three tabs: 'PROPERTIES', 'SCHEDULE', and 'TRIGGER'. The 'TRIGGER' tab is selected. It contains fields for 'Trigger Type' (set to 'File'), 'File Name' (set to 'C:\Users\Administrator\Documents\Automation Anywhere'), and 'Action' (set to 'When file is modified'). Below these are 'Save' and 'Cancel' buttons. On the right is the 'Error Handling' dialog box. It has three main sections: 'Select Error Handling Command' (radio button for 'Begin Error Handling' is selected), 'Select Error Handling Action' (radio button for 'Stop Task' is selected), and 'Select Error Handling Options'. Under 'Options', there are several checkboxes: 'Take Snapshot' (unchecked), 'Run Task' (unchecked), 'Log Data into File' (checked), 'Text' (containing 'Error: \$Error Description\$ Task: \$AA_TaskName\$'), 'Add Timestamp' (unchecked), 'Send Email' (unchecked), and 'Variable Assignment' (unchecked). There are also tips for using F2 to insert variables. At the bottom are 'Save' and 'Cancel' buttons.

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-18. Challenge: Create an Error Handler bot (3 of 4)

Challenge: Create an Error Handler bot (4 of 4)

- System variables that are useful when errors are detected include **\$Error Description\$** and **\$AATaskName\$**
- Although you might perform other subcommands such as running a task or sending an email, these actions would be performed for every error
- To enable the bot to respond selectively, the Handle Errors bot determines how to respond to the errors by:
 - Reading the log file
 - Parsing the error description to evaluate what error was received
- Example: The following error description logged:

```
Error: [DB2/NT64] SQL0433N Value ")_(*&(*&^&$^#%@" is
too long. SQLSTATE=22001 Task:
C:\Users\Administrator\Documents \Automation Anywhere
Files\Automation Anywhere\My Tasks\Account Opening with
Hardening.atmx
▪ The bot should evaluate this part of the error description:
SQLSTATE=22001
```

Figure 12-19. Challenge: Create an Error Handler bot (4 of 4)

Unit summary

- Describe good practices for hardening bots against exceptions

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-20. Unit summary

Review questions



1. True or False: If a file arrives as expected but does not have the expected name, a bot has no choice but to stop processing.
2. Which command is used to run another bot?
 - A. Run Bot
 - B. Run Task
 - C. Error Handler
 - D. Open Program/File
3. True or False: The key to hardening a bot is in predicting and handling both expected and unexpected events.

Review answers

1. True or False: If a file arrives as expected but does not have the expected name, a bot has no choice but to stop processing.
The answer is False
2. Which command is used to run another bot?
 - A. Run Bot
 - B. Run Task
 - C. Error Handler
 - D. Open Program/FileThe answer is B.
3. True or False: The key to hardening a bot is in predicting and handling both expected and unexpected events.
The answer is True

Exercise 10: Hardening the Account Opening bot

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-23. Exercise 10: Hardening the Account Opening bot

Exercise objectives

- Implement bot commands that harden the bot against anticipated failure points and known issues that can affect successful task automation
- Implement error handling commands that address specific error situations
- Create a reusable error handling bot

Hardening bots against exceptions

© Copyright IBM Corporation 2018, 2019

Figure 12-24. Exercise objectives

Exercise introduction



- In this exercise, you implement error handling at various levels, including detecting all bot errors and detecting two specific SQL errors
- Because the SQL errors are experienced across multiple bots, a Handle Errors bot is built that receives errors from other bots
- This lab builds on the Account Opening bot. You also harden the Account Opening bot to respond to the following scenarios:
 - Source spreadsheet file has wrong extension
 - Source spreadsheet layout is different

Unit 13. Bot management and reporting

Estimated time

01:00

Overview

This unit describes bot management features in the Enterprise Client.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Deploy bots with dependencies
- Manage bots in the Enterprise client
- Monitor bot performance in the Enterprise client
- Use the Report Designer to create performance reports

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-1. Unit objectives

Topics

- Deploying bot dependencies
- Bot management tools
- Bot performance monitoring tools

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-2. Topics

13.1. Deploying bot dependencies

Deploying bot dependencies

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-3. Deploying bot dependencies



Deploying bot dependencies (1 of 2)

- If a bot includes subbots or other files, it is understood to have a dependency
- When a file dependency is created, the file is automatically uploaded to the Control Room when the bot is deployed
 - Files can include: DOC, DOCX, XLS, XLSX, CSV, MDB, PKX, ATMX, EXE, PDF, TXT, JPG, PNG, BMP, XML
- The Bot Dependencies tab in the Workbench is used to manage dependent files
- Dependent files need to be located in one of the following folders:
 - My Tasks
 - My Docs
 - My Scripts



© Copyright IBM Corporation 2018, 2019

Bot management and reporting

Figure 13-4. Deploying bot dependencies (1 of 2)



Deploying bot dependencies (2 of 2)

- When the bot is uploaded to the Control Room, dependencies are highlighted
- After a bot is uploaded to the Control Room, the bot Developer can perform code comparisons between the currently deployed bot and the one being developed in the Workbench

Upload Task(s) and Dependencies

Select the Task(s) and Dependencies you want to Upload.

Name	Date Modified	Client Status	Type	Mode	Remarks
<input checked="" type="checkbox"/> Data Consistency Client	3/7/2018 3:43:1...	New	TaskBot		
<input checked="" type="checkbox"/> Data Consistency without Prompt	3/7/2018 11:26:55...	New	TaskBot	Scanned	

Export to CSV Upload Cancel

BOT DEPENDENCIES

Note: All the above listed dependencies must be referenced in the bot with the \$AAAApplicationPath\$ variable.

Copy Copy All Paste
Add Edit Delete

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-5. Deploying bot dependencies (2 of 2)



Performing a code comparison

- Code comparisons between the currently deployed bot and the one being developed in the Workbench can be performed after the bot is uploaded
- Code comparisons can be performed only between two task bots
- MetaBots cannot be compared

Automation Anywhere Enterprise Client - Compare

Compare file: Automation Anywhere\My Tasks\Account Opening with Hardening 4.athx

No. of differences in file : 6

Task Actions List of Server

```

1 Begin Error Handling; Action: Stop Task; Options: Log to File, Task S
2 Comment: Evaluate and prepare file for processing
3 If File Exists ("C:\Vabfiles\account_opening\new_customers.xls")
4 Comment: Rename XLS spreadsheet to XLSX extension.
5 Rename Files "C:\Vabfiles\account_opening\new_customers.xls"
6 Else If File Exists ("C:\Vabfiles\account_opening\new_customers.xls")
7 Comment: Get location of Customer ID column.
8 Excel: Open Spreadsheet "C:\Vabfiles\account_opening\new_customers.xls"
9 Excel: Find "Customer ID"; Range : "All Cells"; Assign to : "$VC
10 If !$CustomerIDCell$ Not Equal To (<=>) "A1" Then
11 Comment: Move Customer ID column.
12 Maximize Window: "Microsoft Excel - new_customers.xlsx"
13 Object Cloning: Click On Client "new_customers.xlsx" in wi
14 Object Cloning: Set Text of Window "Microsoft Excel - new_
15 Object Cloning: Click On Client "new_customers.xlsx" in wi
16 Object Cloning: Set Text of Window "Microsoft Excel - new_
17 End If
18 End If
19 Comment: Open the Excel spreadsheet
20 Excel: Open Spreadsheet "C:\Vabfiles\account_opening\new_cus
21 Maximize Window: "Microsoft Excel - new_customers.xlsx"
22 Excel: Get All Cells Session: NewCustomers
23 Connect to "$DB_CONNECT_STRING(DB_CONNECT_STRING)$"
24 Start Loop "Each row in an Excel dataset of Session: NewCusto
25 Comment: For each row, create a customer
  
```

Task Actions List of Client

```

1 Begin Error Handling; Action: Continue; Options: Take Snapshot, Log
2 Comment: Evaluate and prepare file for processing
3 If File Exists ("C:\Vabfiles\account_opening\new_customers.xls")
4 Comment: Rename XLS spreadsheet to XLSX extension.
5 Rename Files "C:\Vabfiles\account_opening\new_customers.xls"
6 Else If File Exists ("C:\Vabfiles\account_opening\new_customers.xls")
7 Comment: Get location of Customer ID column.
8 Excel: Open Spreadsheet "C:\Vabfiles\account_opening\new_customers.xls"
9 Excel: Find "Customer ID"; Range : "All Cells"; Assign to : "$VC
10 If !$CustomerIDCell$ Not Equal To (<=>) "A1" Then
11 Comment: Move Customer ID column.
12 Object Cloning: Click On Client "new_customers.xlsx" in wi
13 Object Cloning: Set Text of Window "Microsoft Excel - new_
14 Object Cloning: Click On Client "new_customers.xlsx" in wi
15 Object Cloning: Set Text of Window "Microsoft Excel - new_
16 End If
17 End If
18 Comment: Open the Excel spreadsheet
19 Excel: Open Spreadsheet "C:\Vabfiles\account_opening\new_cus
20 Maximize Window: "Microsoft Excel - new_customers.xlsx"
21 Excel: Get All Cells Session: NewCustomers
22 Connect to "$DB_CONNECT_STRING(DB_CONNECT_STRING)$"
23 Start Loop "Each row in an Excel dataset of Session: NewCusto
24 Comment: For each row, create a customer
  
```

OK

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-6. Performing a code comparison

13.2. Bot management tools

Bot management tools

Bot management and reporting

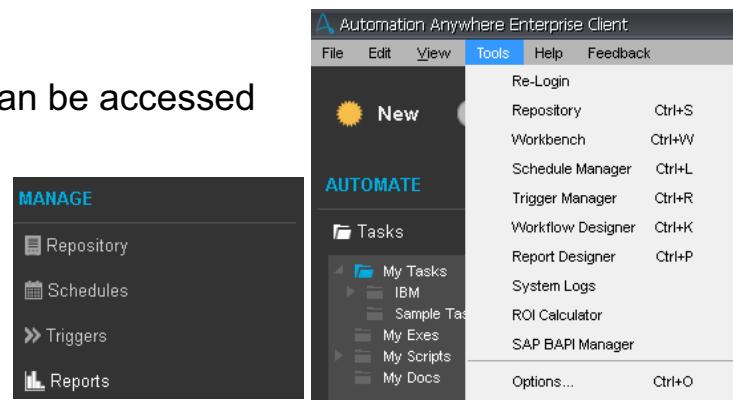
© Copyright IBM Corporation 2018, 2019

Figure 13-7. Bot management tools



Bot management tools

- Bot management tools exist in both the Control Room and in the Enterprise client
- The Control Room is supported by a server administrator role. The Enterprise client is used with the Workbench by bot Developers to build and manage their bots
- Utilities that can be used by the bot Developer to manage their bots in the Enterprise client include:
 - Repository
 - Schedule Manager
 - Trigger Manager
- Bot management utilities can be accessed from the Tools menu or by clicking **Manage** in the Features panel



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-8. Bot management tools

Bot management tools: Repository (1 of 2)

- The Repository view in the Enterprise client allows the Developer to compare which bots are uploaded to the server and which are in the client repository
- The client repository contains all bots that are being developed on the Developers local workstation
 - The server repository contains bots ready to be deployed to other environments
- The bot Developer can see all artifacts in both repositories, including bots, MetaBots, dependent artifacts, reports, and others
- From the Repository view, the bot Developer can upload and download bots and artifacts
 - The bot Developer can also delete items from either repository
- Code comparisons can be made from the Repository screen
 - By making a code comparison, the Developer can determine whether updates were made to the local copy since it's last deployment to the server repository

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-9. Bot management tools: Repository (1 of 2)



Bot management tools: Repository (2 of 2)

Repository

Server Repository

Name	Type	Date Modified	Status
Account Opening 1.atmx	Task File	2/27/2018 5:10:48 PM	Same
Account Opening with Hardening...	Task File	3/5/2018 12:18:42 PM	Old
Data Consistency Client.atmx	Task File	3/7/2018 3:43:15 PM	Old
Data Consistency solution.atmx	Task File	3/7/2018 11:26:00 AM	Old
Data Consistency without Prom...	Task File	3/7/2018 11:26:55 AM	New
Handle Errors OLD.atmx	Task File	3/1/2018 11:27:04 AM	Same
Run Data Consistency Bot.atmx	Task File	3/6/2018 5:35:13 PM	New
test0.atmx	Task File	3/6/2018 4:12:01 PM	Same
test1.atmx	Task File	3/6/2018 4:12:15 PM	Old

Client Repository

Name	Type	Date Modified	Status
Account Opening 1.atmx	Task File	2/27/2018 5:10:48 PM	Same
Account Opening 2.atmx	Task File	2/23/2018 4:51:13 PM	New
Account Opening solution June.atmx	Task File	2/27/2018 2:42:33 PM	New
Account Opening solution OLD.atmx	Task File	2/15/2018 12:19:28 PM	New
Account Opening solution.atmx	Task File	2/27/2018 2:16:01 PM	New
Account Opening with Hardening 2...	Task File	2/26/2018 5:38:35 PM	New
Account Opening with Hardening 3...	Task File	3/2/2018 2:46:22 PM	New
Account Opening with Hardening 4...	Task File	3/6/2018 4:26:05 PM	Updated
Account Opening with Hardening O...	Task File	3/1/2018 11:04:01 AM	New

Buttons: Upload, Download, Compare, Delete

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-10. Bot management tools: Repository (2 of 2)

Bot management tools: Schedule Manager (1 of 4)

- The Schedule Manager displays all bot schedules created by the bot Developer
- The bot Developer can view schedules by day, week, or month
- Using the Schedule Manager, the bot Developer can add, edit, and delete schedules
- Bots can be scheduled to run on particular days, weeks, or months of the year
- Bot creators can schedule only by using the “One Time Only” option
- The bot runner role is required to create repeating schedules
- To create a schedule:
 - Select Schedule Manager from the **Tools** menu
 - On the Schedule Manager screen, click **Add**
 - Configure the start date and time for the schedule
 - Click **Save**

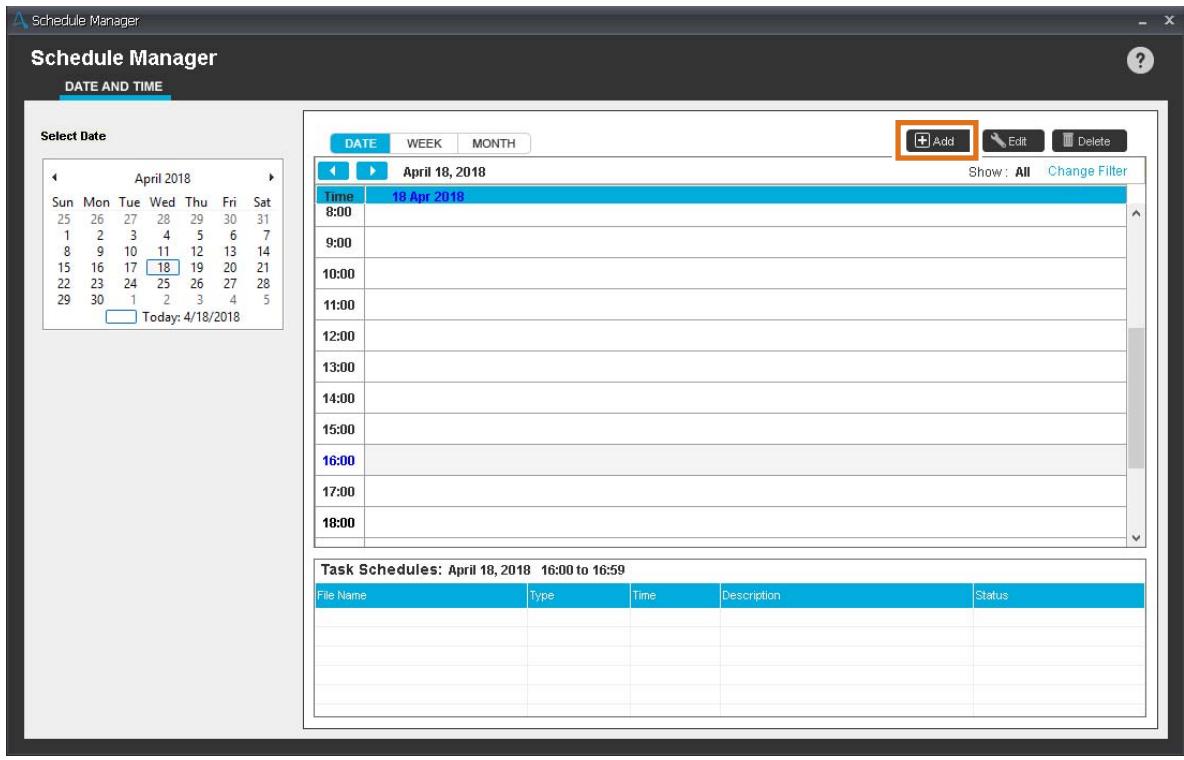
Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-11. Bot management tools: Schedule Manager (1 of 4)



Bot management tools: Schedule Manager (2 of 4)



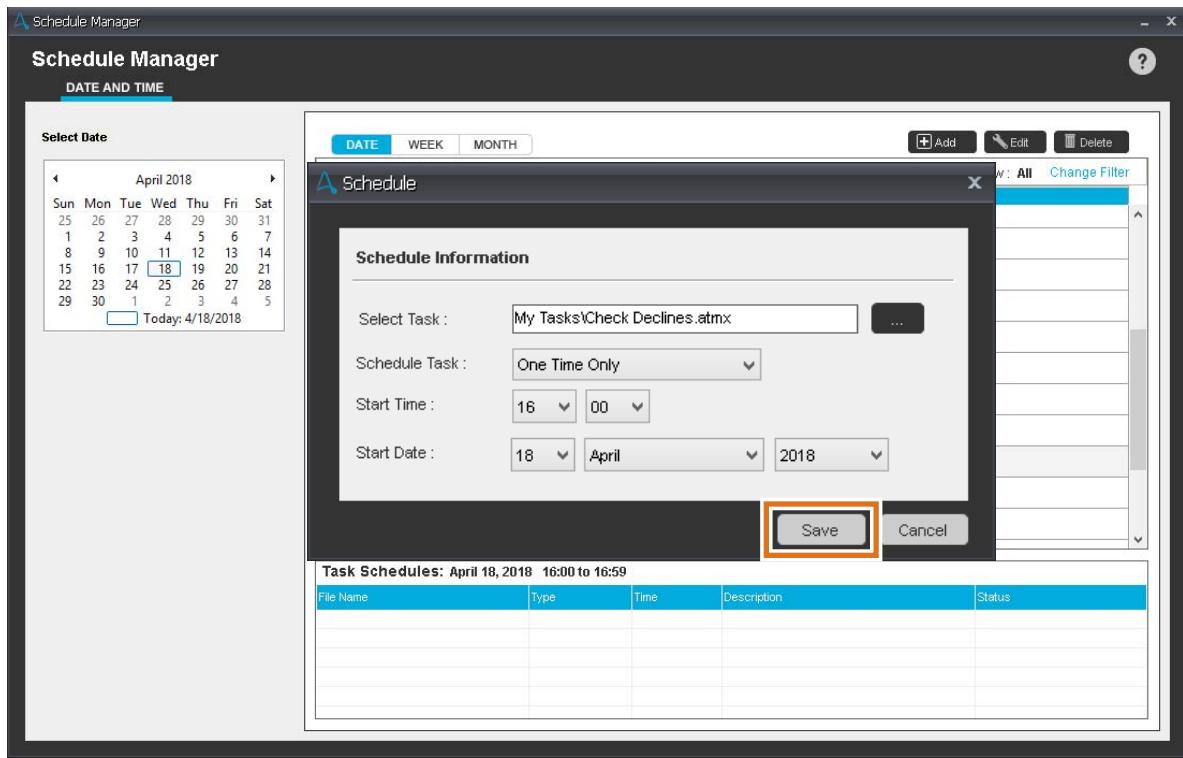
Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-12. Bot management tools: Schedule Manager (2 of 4)



Bot management tools: Schedule Manager (3 of 4)



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-13. Bot management tools: Schedule Manager (3 of 4)



Bot management tools: Schedule Manager (4 of 4)

The screenshot shows the IBM Schedule Manager application interface. On the left, there is a calendar for April 2018 with the 18th highlighted. On the right, a detailed schedule for April 18, 2018, is displayed from 8:00 to 18:00. A specific task at 16:00 is selected and highlighted with an orange border. Below the schedule, a table titled "Task Schedules: April 18, 2018" lists the task details.

File Name	Type	Time	Description	Status
My Tasks\Check Declines.atmx	One Time Only	4:00 PM	At 16:00 on 4/18/2018	The task has not yet run.

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-14. Bot management tools: Schedule Manager (4 of 4)

Bot management tools: Trigger Manager (1 of 4)

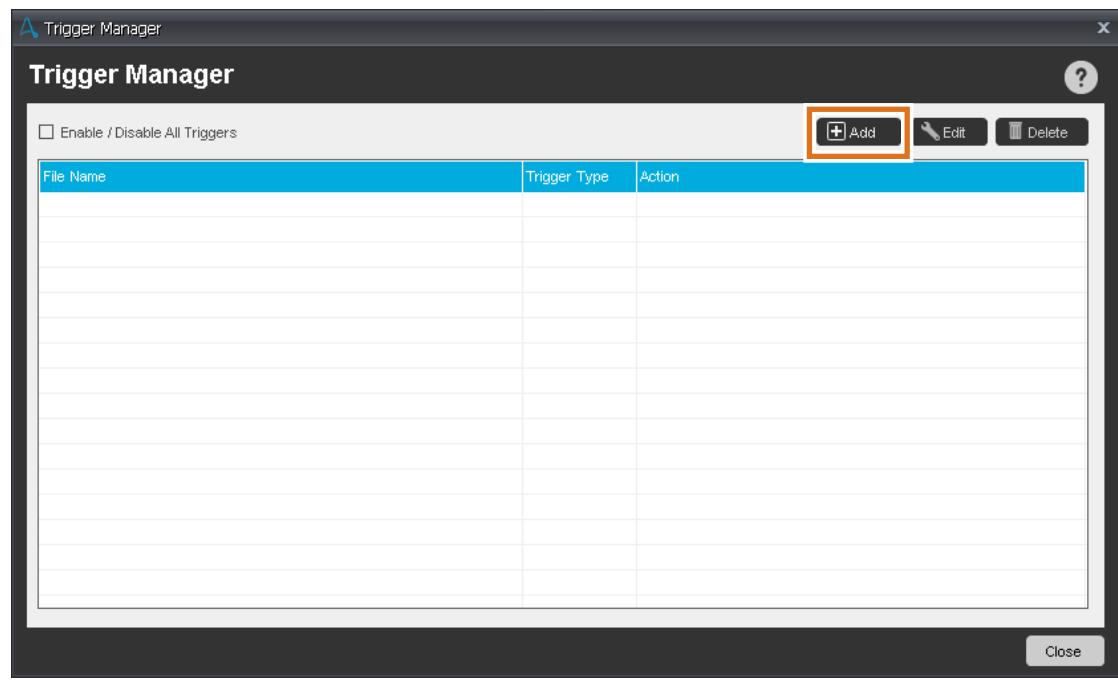
- In addition to scheduling tasks to run based on a date and time, the bot Developer can also use trigger events to automatically start a bot
- The Trigger Manager displays all bot triggers created by the bot Developer
- Using the Trigger Manager, the bot Developer can add, modify, delete, enable, or disable triggers
- There's a many-to-many relationship between triggers and bots:
 - One trigger can be used by multiple bots
 - One bot can be used by multiple triggers
- To create a schedule:
 - Select Trigger Manager from the **Tools** menu
 - On the Trigger Manager screen, click **Add**
 - Configure the trigger type and action to be performed
 - Click **Save**

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-15. Bot management tools: Trigger Manager (1 of 4)

Bot management tools: Trigger Manager (2 of 4)

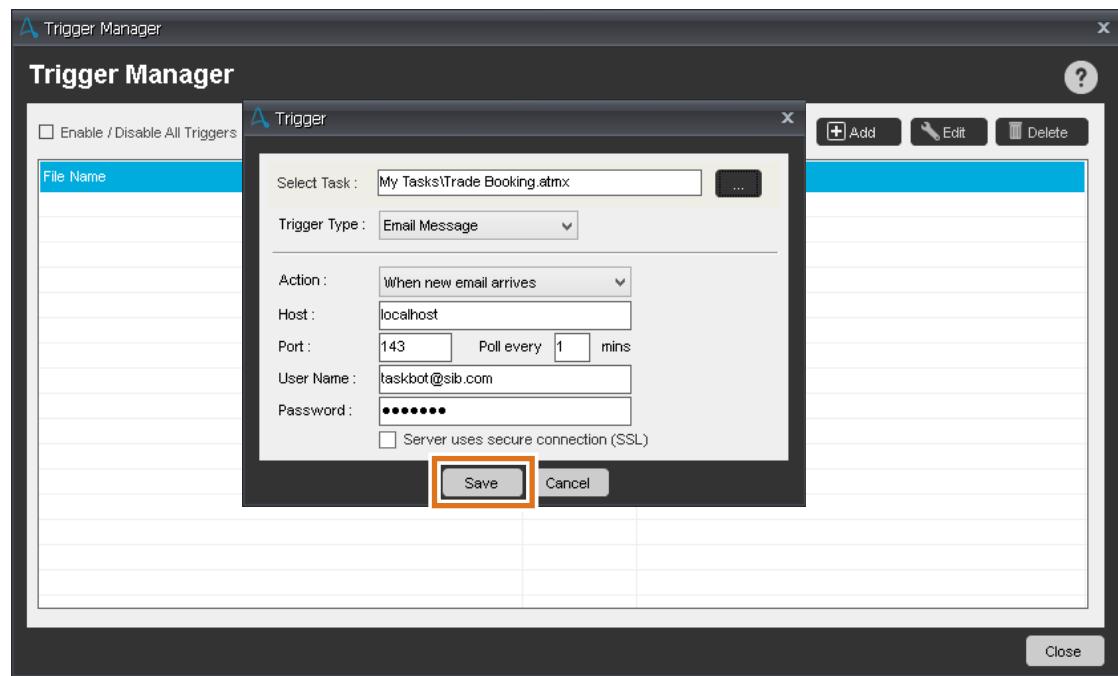


Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-16. Bot management tools: Trigger Manager (2 of 4)

Bot management tools: Trigger Manager (3 of 4)

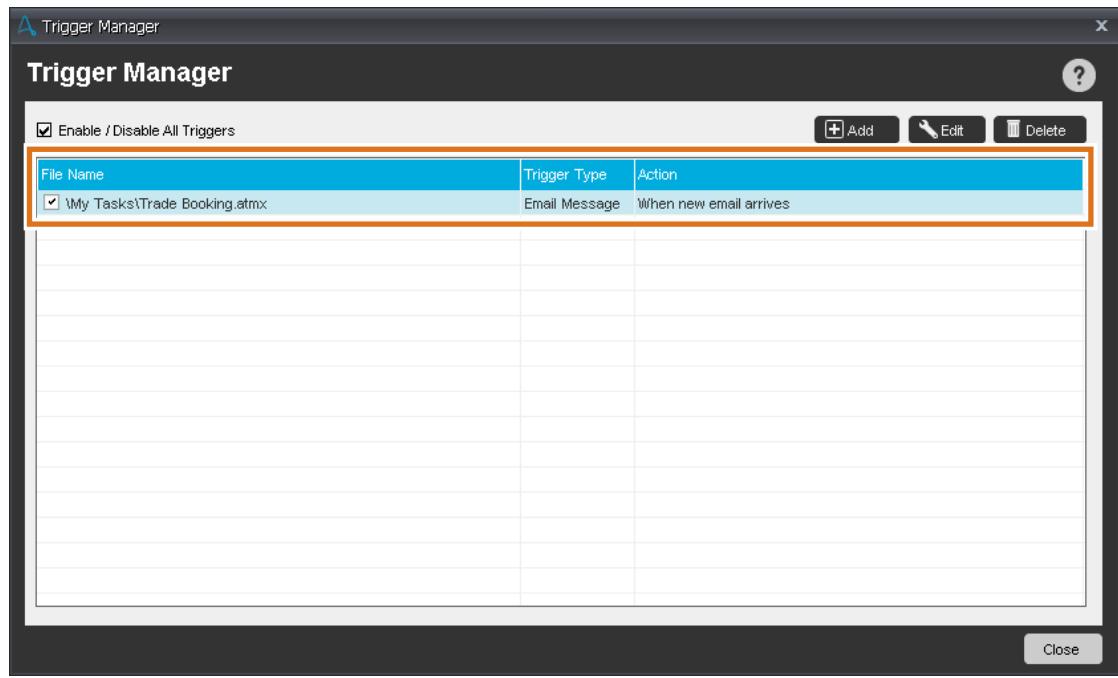


Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-17. Bot management tools: Trigger Manager (3 of 4)

Bot management tools: Trigger Manager (4 of 4)



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-18. Bot management tools: Trigger Manager (4 of 4)

13.3. Bot performance monitoring tools

Bot performance monitoring tools

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-19. Bot performance monitoring tools

Bot performance monitoring tools

- Bot performance can also be monitored from the Enterprise client
- The following tools in the Enterprise client are useful in evaluating bot performance:
 - Notifications
 - System Logs
 - ROI Calculator
 - Report Designer
 - Bot Insights

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-20. Bot performance monitoring tools

Bot performance monitoring tools: Notifications (1 of 3)

- Auto-generated notifications are sent out for one of three types of events:
 - Trial Notification
 - Notifications are sent out regarding the trial license when it is about to expire
 - Task Failure Notification
 - Notifications of bot failures that are scheduled or triggered
 - External Notification
 - External notifications occur when there are new enhancements or features in the product
 - You need to be connected to the internet to receive external notifications
- Notifications can be viewed from the Enterprise client by clicking the bell in the upper right of the Help bar
- The bell icon varies depending on what notifications are available



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-21. Bot performance monitoring tools: Notifications (1 of 3)

Bot performance monitoring tools: Notifications (2 of 3)

- Notifications legend

Icon	Indicates
	All notifications were read
	No notifications are available
	Number of unread notifications
	External notification
	Trial notification
	Task failure notification

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

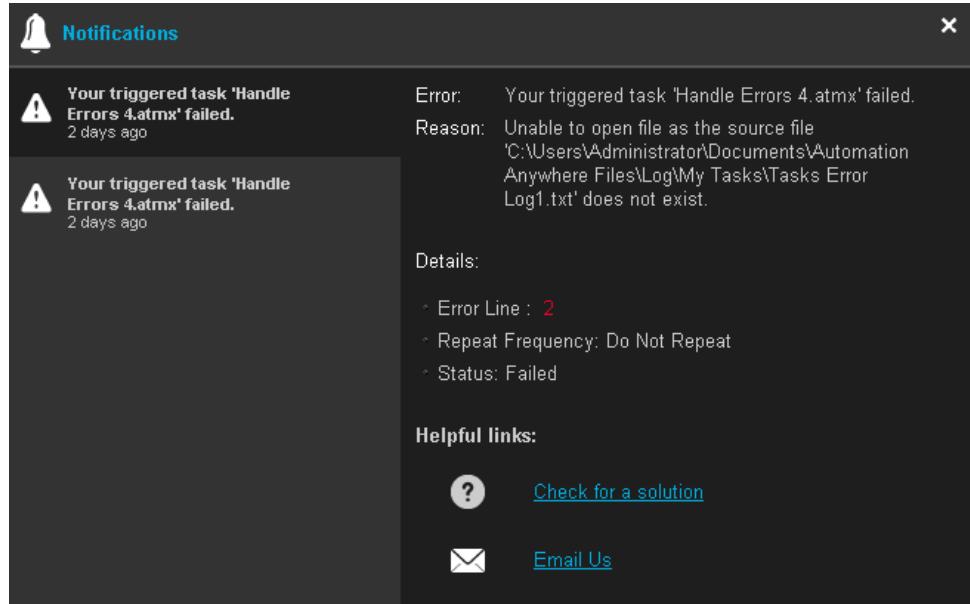
Figure 13-22. Bot performance monitoring tools: Notifications (2 of 3)



Bot performance monitoring tools: Notifications (3 of 3)

- A primary purpose of the auto-generated notifications is to prevent bottlenecks from occurring for queued tasks
- For a bot failure notification, the following is displayed:

- Error
- Reason
- Details



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-23. Bot performance monitoring tools: Notifications (3 of 3)

Bot performance monitoring tools: System logs (1 of 4)

- All events that occur are logged, including major events such as a task run, task creation, and changes to task properties
- To view system logs:
 - Select System Logs from the Tools menu
 - Select Log Type to display
 - Configure date range
 - Click Generate Logs

Log types

- Task Creation
- Task Run
- Task Modification
- Task Deleted
- Report Creation
- Report Run
- Report Modification
- Report Deleted
- Task To Exe
- Schedule
- Trigger
- Task Properties
- App Configuration
- File
- Folder
- Script
- Other

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-24. Bot performance monitoring tools: System logs (1 of 4)



Bot performance monitoring tools: System logs (2 of 4)

System Logs

Select Log Type : Task Run Start Date : 04/17/2018 End Date : 04/18/2018 Generate Logs

Task Run Logs

Sr. No.	Task/Variable Name	Date	Time	Description
1	Data Consistency Client.atmx	04/17/2018	11:16:21	Task Run, Started, C:\Users\Administrator\Documents\Automation...
2	Data Consistency without Prompt ...	04/17/2018	11:16:21	Task Run, Started, C:\Users\Administrator\Documents\Automation...
3	Data Consistency without Prompt ...	04/17/2018	11:16:27	Task Run, Completed, C:\Users\Administrator\Documents\Automation...
4	Data Consistency Client.atmx	04/17/2018	11:16:28	Task Run, Completed, C:\Users\Administrator\Documents\Automation...
5	Data Consistency Client.atmx	04/17/2018	11:16:51	Task Run, Started, C:\Users\Administrator\Documents\Automation...
6	Data Consistency without Prompt ...	04/17/2018	11:16:51	Task Run, Started, C:\Users\Administrator\Documents\Automation...
7	Data Consistency without Prompt ...	04/17/2018	11:17:23	Task Run, Completed, C:\Users\Administrator\Documents\Automation...
8	Data Consistency Client.atmx	04/17/2018	11:17:23	Task Run, Completed, C:\Users\Administrator\Documents\Automation...
9	Check Declines.atmx	04/18/2018	16:00:10	Task Run, Started, C:\Users\Administrator\Documents\Automation...
10	Check Declines.atmx	04/18/2018	16:00:44	Task Run, Failed, C:\Users\Administrator\Documents\Automation...

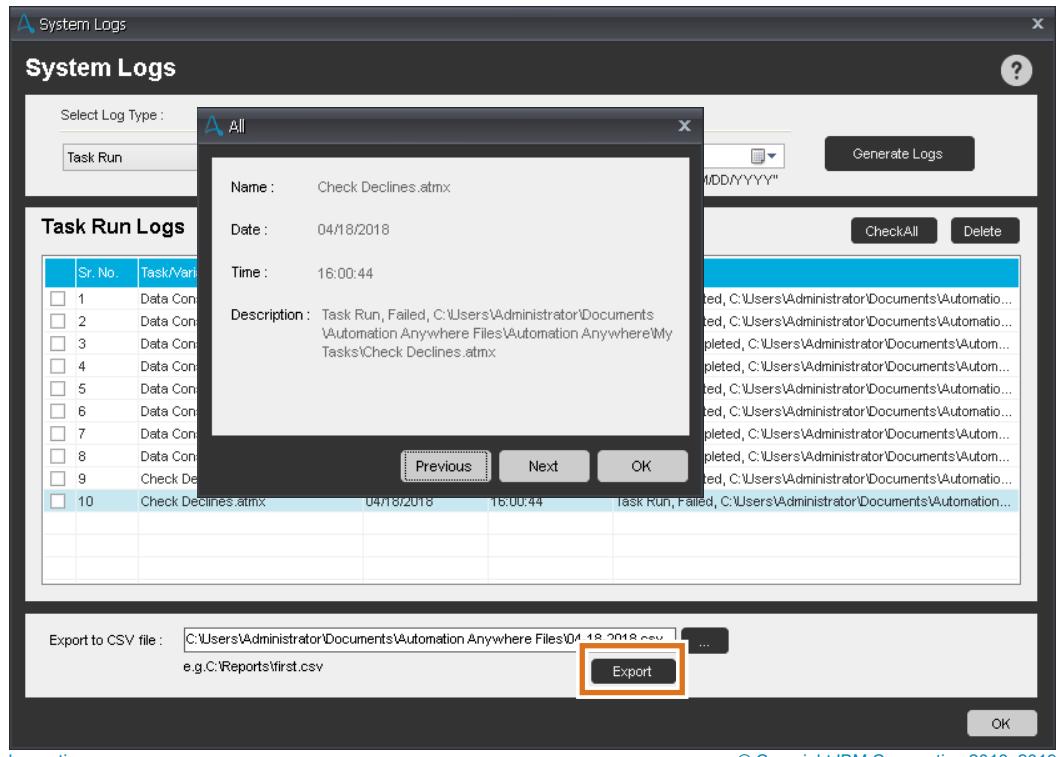
Export to CSV file : C:\Users\Administrator\Documents\Automation Anywhere Files\04-18-2018.csv ... e.g.C:\Reports\first.csv Export OK

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-25. Bot performance monitoring tools: System logs (2 of 4)

Bot performance monitoring tools: System logs (3 of 4)



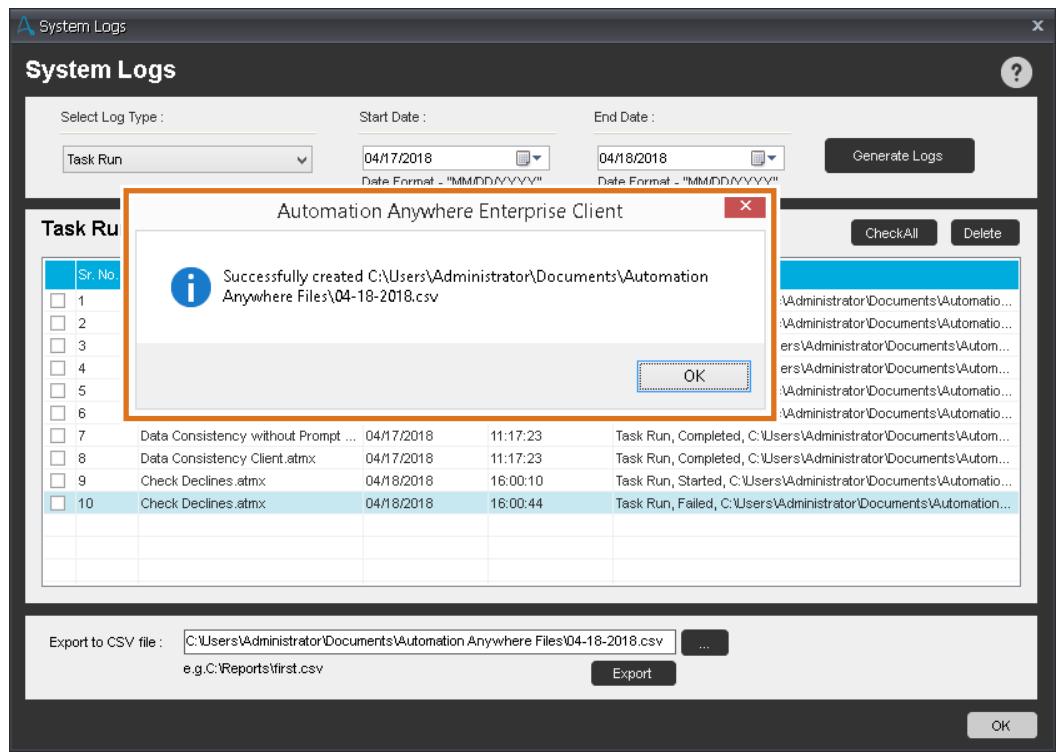
Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-26. Bot performance monitoring tools: System logs (3 of 4)



Bot performance monitoring tools: System logs (4 of 4)



Bot management and reporting

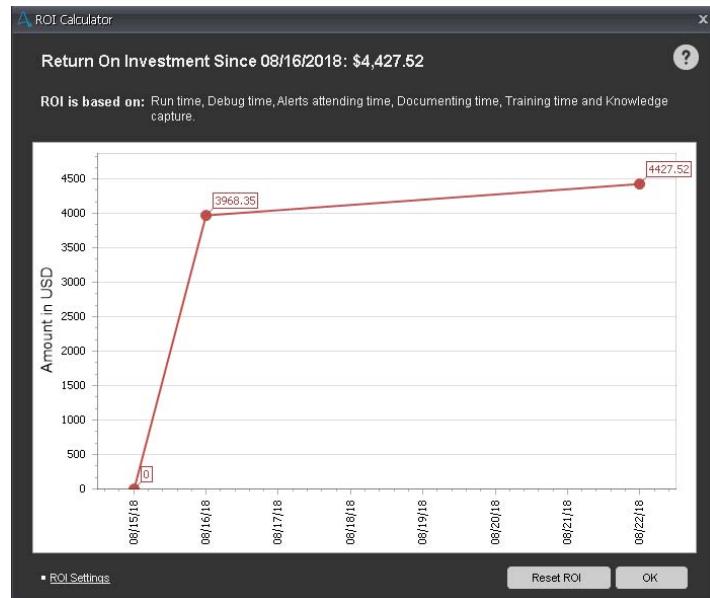
© Copyright IBM Corporation 2018, 2019

Figure 13-27. Bot performance monitoring tools: System logs (4 of 4)



Bot performance monitoring tools: ROI Calculator

- The ROI (Return On Investment) Calculator calculates a cumulative dollar amount based on the results of automating tasks
 - Calculated based on the ROI hourly personnel rate
- The ROI Calculator makes the following calculations:
 - Resource costs for time saved
 - Reduced error rates
 - Automated troubleshooting
 - Automated documentation
 - Resource realignment
 - Training benefits
 - Knowledge capture



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-28. Bot performance monitoring tools: ROI Calculator

Bot performance monitoring tools: Report Designer (1 of 4)

- The Report Designer produces graphical reports that display the status of tasks, and ROI, over time
- The following reports can be generated:
 - Task Run
 - Workflow Run
 - ROI
 - Task Timeline
 - Workflow Timeline
 - Visual Logs



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-29. Bot performance monitoring tools: Report Designer (1 of 4)

Bot performance monitoring tools: Report Designer (2 of 4)

- Task Run
 - Displays how many times a bot ran successfully and how many times it failed during the period
- Workflow Run
 - Displays how many times a workflow ran successfully and how many times it failed during the period
- Task Timeline
 - Displays task history in a graphical timeline
 - Includes number of times a task was run, modified, deleted, or created during that time
- Workflow Timeline
 - Displays workflow history in a graphical timeline, including number of times the workflow was run, modified, deleted, or created during that time
- ROI
 - Displays how much USD/Dollar Value was saved over a period of time
- Visual Logs
 - Displays bots that ran by providing step-by-step SnapPoints for each action in a task

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

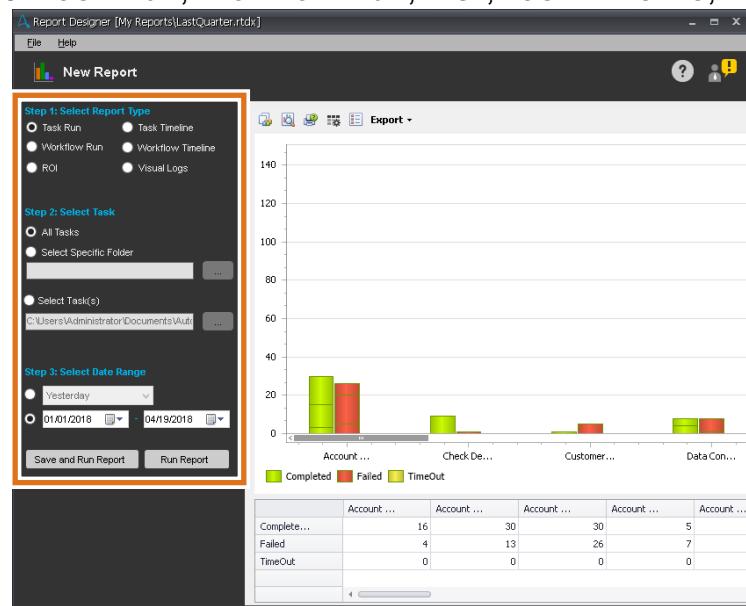
Figure 13-30. Bot performance monitoring tools: Report Designer (2 of 4)

IBM Training



Bot performance monitoring tools: Report Designer (3 of 4)

- To generate a report by using the Report Designer:
 - Select Report Designer from the Tools menu or click Manage in the Features pane, then select Reports
 - Step 1: Select Report type:** Task Run, Workflow Run, ROI, Task Timeline, Workflow Timeline, or Visual Logs
 - Step 2: Select Task:** You can select all tasks, a specific folder, or one task
 - Step 3: Select Date Range** – You can select one of the default ranges or you can create your own date range
 - Click **Run Report** or **Save and Run Report**



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-31. Bot performance monitoring tools: Report Designer (3 of 4)

IBM Training

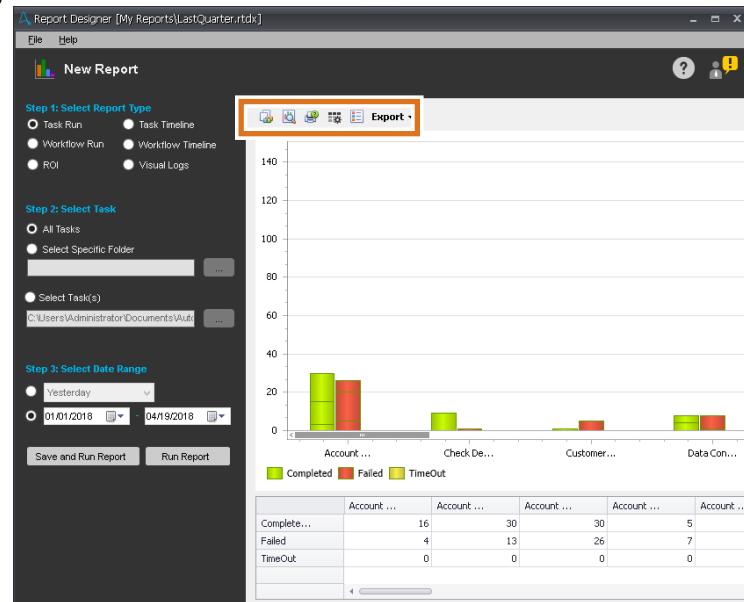


Bot performance monitoring tools: Report Designer (4 of 4)

- After you generate the report, you can change how the report is displayed, print the report, or export the report to: PDF, HTML, MHT, XLS, XLSX, RTF, or image

- Report Designer icons

Icon	Indicates
	Copy to Clipboard
	Print Preview
	Print
	Display Data Grid
	Display Legend Box
	Export the report



Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-32. Bot performance monitoring tools: Report Designer (4 of 4)

Unit summary

- Deploy bots with dependencies
- Manage bots in the Enterprise client
- Monitor bot performance in the Enterprise client
- Use the Report Designer to create performance reports

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-33. Unit summary

Review questions



1. True or False: There is a one-to-one relationship between bots and bot triggers.
2. True or False: A version control system is required to perform a code comparison between the bot server copy and the Enterprise client copy.
3. The following are bot management tools available in the Enterprise client **except**:
 - A. Repository
 - B. Trigger Manager
 - C. Schedule Manager
 - D. Operations Room

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-34. Review questions

Review answers

1. True or False: There is a one-to-one relationship between bots and bot triggers.

The answer is False



2. True or False: A version control system is required to perform a code comparison between the bot server copy and the Enterprise client copy.

The answer is False

3. The following are bot management tools available in the Enterprise client **except**:

- A. Repository
- B. Trigger Manager
- C. Schedule Manager
- D. Operations Room

The answer is D.

Exercise 11: Managing bots

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-36. Exercise 11: Managing bots

Exercise objectives

- Create dependencies between bots
- Pass parameters from one bot to another
- Upload bots to the Server Repository and compare bot code
- Use bot management features in the Enterprise Client

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-37. Exercise objectives

Exercise introduction



- In this exercise, you become familiar with task dependencies, bot deployment, bot management, and reporting
 - Build a bot (Client) that passes the Account Number to the Data Consistency bot (Server)
 - Create a dependency between the bots
 - Deploy both bots to the Server repository
 - Make a change in the Data Consistency bot in the client repository to remove the code to prompt for an account number
 - Perform a code comparison between the two bots in the Enterprise Client
 - Run the bot
 - View the system logs and Notifications available in the Enterprise Client
 - Design a simple report by using the Report Designer
- This lab builds on the Data Consistency bot

Bot management and reporting

© Copyright IBM Corporation 2018, 2019

Figure 13-38. Exercise introduction

Unit 14.

Administering bots through the Control Room

Estimated time

01:00

Overview

In this unit, you learn about the bot administration features in the Control Room. You learn how administrators can use the Repository Manager, Operations Room, and Audit Trail to manage bots and view operational data about bot performance and Control Room events.

How you will check your progress

- Checkpoint
- Machine exercises

Unit objectives

- Describe how to manage bot workload using bot runners, device pools and queues.
- Explain how to run and schedule bots from the Control Room
- Describe how to monitor bot performance using Bot Insights
- Understand bot lifecycle management

[Administering bots through the Control Room](#)

© Copyright IBM Corporation 2018, 2019

Figure 14-1. Unit objectives

Topics

- Workload management
- Running and scheduling bots from Control Room
- Bot Insights Overview
- Bot lifecycle management

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-2. Topics

14.1. Workload management

Workload management

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-3. Workload management

Queues and device pools



Device pools

- Device pools provide a logical grouping of similar bot runners on which you can run bots with the work item from a queue.



Queues

- Queues are used to manage work items in the Control Room.
- An administrator user can create, update, and control the way work items are included in queues and distributed to bot runners that are part of device pools.
- Queues can be prioritized to achieve Service Level Agreements.

Keep in mind:

- Managing workload by dividing work into small logical units helps to ensure SLAs are met and resource usage is optimized.
- Auto-distribution of workload to bot runners achieves best optimization of resources.

[Administering bots through the Control Room](#)

© Copyright IBM Corporation 2018, 2019

Figure 14-4. Queues and device pools



Create a device pool

- Go to **Devices > My device pools** and click **Create device pool**
- Enter name, associate bot runner device, add pool owners and consumers, and click **Create device pool**.

Create device pool

[Cancel](#) [Create device pool](#)

BOT RUNNERS (1)

- Name Test Bot Runner
- Description Bot runner used for ...
- Devices selected (1)

DEVICE POOL OWNERS (1)

- User selected (1)

DEVICE POOL CONSUMERS (1) optional

- Device name ▾
- Search device name 🔍

Bot runners

Name this device pool (description is optional) and select one or more bot runners for the pool.

Name <input style="width: 100%; border: 1px solid #ccc; padding: 2px;" type="text" value="Test Bot Runner"/> <small>Max characters = 50</small>	Description (optional) <input style="width: 100%; border: 1px solid #ccc; padding: 2px;" type="text" value="Bot runner used for testing."/> <small>Max characters = 255</small>
--	--

Choose one or more bot runners. Disabled bot runners are already in a device pool and cannot be used in this one.

Available devices (0 of 1)	Selected (1)
STATUS	DEVICE NAME ↑
<input type="checkbox"/>	Connected VCLASSBASE

[Next >](#)

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-5. Create a device pool



Create a queue

- Go to **Workload > queues** and click **Create queue...**
- Enter required information. If not all information is filled out, you can create a draft. Otherwise, if all information is filled out, click **Create queue**.

Create queue

Cancel
Create draft of queue

GENERAL

Name
• Test Queue

Description
• Queue used for test...

Threshold
• 1 work item(s)

Time it takes
• --

OWNERS
• 1 user selected

PARTICIPANTS
(optional)

CONSUMERS
(optional)

DEFINE WORK ITEM ST...
Get column headers from:
• Excel file

WORK ITEM STRUCTURE
• 0 columns

ADD WORK ITEMS
• Add items

General settings

In this wizard, you will be able to create a queue of work items and assign permissions. You can save a draft at any time and come back later to complete the queue.

Queue Name	<input type="text" value="Test Queue"/> Description (optional)
Max characters = 50	
Queue used for testing bot execution.	
Max characters = 255	
Reactivation Threshold	
<input type="text" value="1"/> work item(s)	
Minimum number in queue to resume processing.	
Time it takes for a person to complete 1 work item (optional)	
<input type="text"/>	seconds

Next >

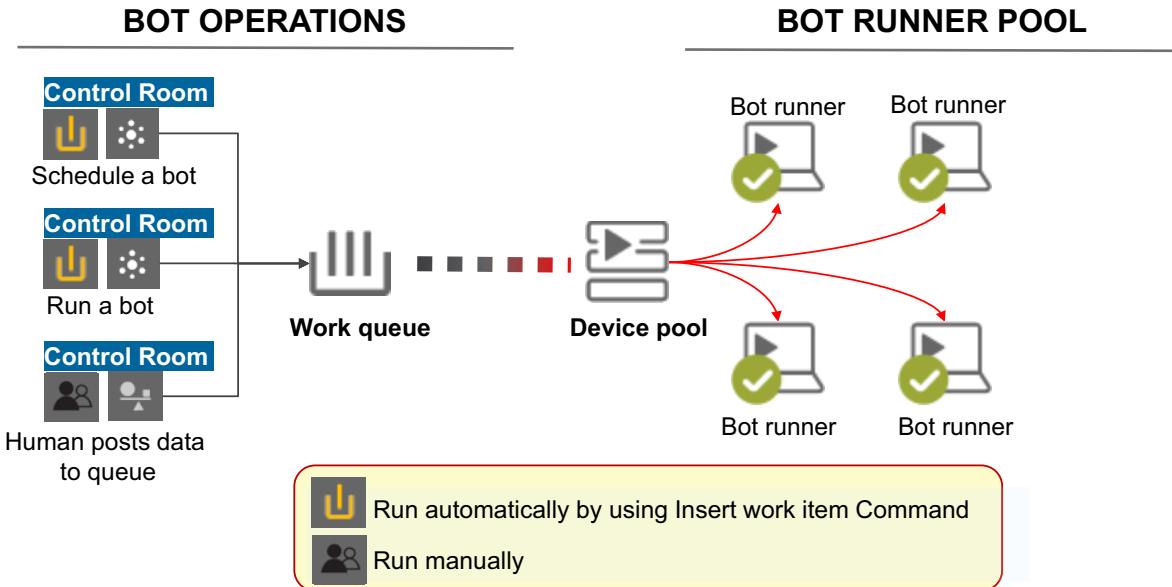
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-6. Create a queue

Queue-based workload management

An administrator can run bots immediately through scheduling or by using a queue or device pool. Developers can run bots by inserting work items directly into the queue.



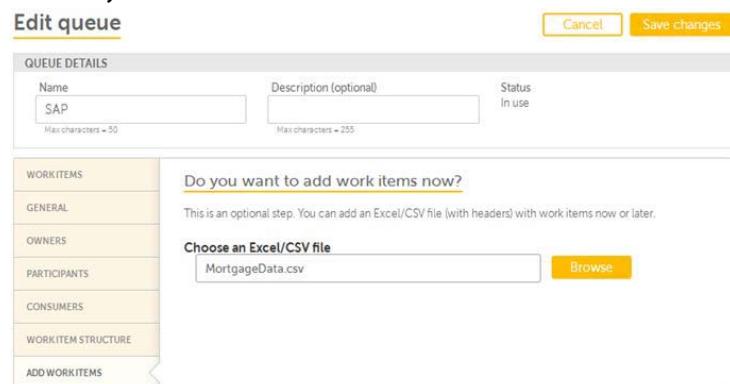
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-7. Queue-based workload management

Managing work items

- As an administrative user with queue management privileges, you can view, edit, and delete work items in queues of which you are an owner, consumer, or participant.
- This allows you to fix discrepancies before queue processing and reduces automation-related errors and failures.
- You can view a work item when its status is **Successful**, **Unsuccessful**, **On hold**, **Active**, or **Data Error**.
- Every queue that is created in the Control Room is associated with a queue category. Hence all work items in a queue have the same set of attributes that are specified in that queue category.



The screenshot shows the 'Edit queue' interface. At the top right are 'Cancel' and 'Save changes' buttons. Below is a 'QUEUE DETAILS' section with fields for 'Name' (SAP), 'Description (optional)', and 'Status' (In use). A note says 'Max characters = 50' for Name and 'Max characters = 255' for Description. To the left is a vertical sidebar with tabs: WORKITEMS (selected), GENERAL, OWNERS, PARTICIPANTS, CONSUMERS, WORKITEM STRUCTURE, and ADD WORKITEMS. On the right, under 'WORKITEMS', there's a section titled 'Do you want to add work items now?'. It includes a note: 'This is an optional step. You can add an Excel/CSV file (with headers) with work items now or later.' Below is a 'Choose an Excel/CSV file' input field containing 'MortgageData.csv' and a 'Browse' button.

[Administering bots through the Control Room](#)

© Copyright IBM Corporation 2018, 2019

Figure 14-8. Managing work items

Queue orchestration

- A Developer can configure a bot to process a work item from one queue and push the outcome as a work item into another queue by using the **Insert work item** command.
- To do this, you must add the queue category in the bot. The bot can then consume work items from queues of the selected category.
- Within the Workbench, a queue category can be selected at the bottom of the variable manager.



- To access the work item attributes from the queue category, use the system defined variable **\$WorkItem\$** in the command.

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-9. Queue orchestration

Using work queues

To use a work queue



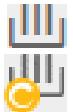
1. Enable bot runner device

	STATUS	DEVICE NAME	USERNAME	DEVICE POOL	TYPE
<input type="checkbox"/>	Disconnected	VCLASSBASE	devuser1	N/A	Bot creator
<input type="checkbox"/>	Connected	VCLASSBASE	botuser1	Transactions	Bot runner



2. Create a device pool and attach bot runner device(s)

	STATUS	DEVICE POOL NAME	DETAILED STATUS	# OF AUTOMATIONS	# OF DEVICES	OWNERS
<input type="checkbox"/>	Connected	Transactions	All connected	1	1	devuser2 3 more...



3. Create a queue with a queue category



4. Associate queue with a work item and device pool

	STATUS	QUEUE NAME	MY ACCESS	AUTOMATION NAME	AUTOMATION STATUS	BOT NAME	DEVICE POOL
<input type="checkbox"/>	In use	Transactions	Queue owner	Transactions Solution.	Active	Transactions Solution.atmx	Transactions



5. Post work to queues

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-10. Using work queues

14.2. Running and scheduling bots from the Control Room

Running and scheduling bots from the Control Room

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-11. Running and scheduling bots from the Control Room



Run bot now (1 of 5)

- You can run a bot immediately, schedule it for execution later, or run it with a queue.
- You can run a bot immediately from the **In progress**, **Scheduled**, and **My Bots** pages. The procedure is the same in all pages.
- When you choose **Run bot now...**, there is no need for a device pool or queue. However, a bot runner device must be connected.

My bots

[Import bots...](#) [Export bots...](#) [Run bot now...](#) [Run bot with queue...](#) [Schedule bot...](#)

To see files here, upload them from your Bot creator or your Bot runner. You will only see files that you have permission to see.

Files and folders {2 of 2}						
	TYPE	NAME	SIZE	CLIENT LAST MODIFIED	LAST MODIFIED	MODIFIED BY
<input type="checkbox"/>	Task Bot	Rest Test.atmx	10.92 KB	15:21:14 EDT 2018-08-14	12:12:54 EDT 2018-08-24	de
<input type="checkbox"/>	Task Bot	Transactions Solution.atmx	15.79 KB	12:09:08 EDT 2018-08-22	12:12:54 EDT 2018-08-24	de

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-12. Run bot now (1 of 5)

IBM Training

Run bot now (2 of 5)

Select a bot and review any dependencies

Run bot now

BOT + DEPENDENCIES	DEVICES	NAME + DESCRIPTION
Bot Transactions Solution	Devices --	Name Transactions Solution.18.08.29.1711.41.devuse...
Dependencies Rest Test.atmx		Description --

Select a Task Bot

Folders

- My Docs
- My MetaBots
- My Reports
- My Scripts
- My Tasks

Search name

TYPE	NAME
Task Bot	Rest Test.atmx

Transactions Soluti...

Review dependencies for Transactions Solution

Automation Anywhere\My Tasks\Transactions Solution.atmx

Next >

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-13. Run bot now (2 of 5)

IBM Training

Run bot now (3 of 5)

Select a bot runner device

Run bot now

BOT + DEPENDENCIES	DEVICES	NAME + DESCRIPTION
Bot Transactions Solution	Devices VCLASSBASE	Name Transactions Solution.18.08.29.17.17.23.rpaad...
Dependencies Rest Test.atmx		Description --

Devices

Run bot runner session on control room

Search name

Available devices (0 of 1)

<input type="checkbox"/> STATUS	NAME
<input type="checkbox"/>	VCLASSBASE

Selected (1)

<input type="checkbox"/> STATUS	NAME
<input type="checkbox"/> Connected	VCLASSBASE

Upcoming schedules for selected device: VCLASSBASE

Scheduled activity (0)

TYPE	ACTIVITY NAME ↑	NEXT OCCU...	BOT	SCHEDULE
< / >	III			>

< Back **Next >**

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-14. Run bot now (3 of 5)

Run bot now (4 of 5)

Click Run now

Run bot now

BOT + DEPENDENCIES	DEVICES	NAME + DESCRIPTION
Bot Transactions Solution	Devices VCLASSBASE	Name Transactions Solution.18.08.29.17.17.23.rpaad...
Dependencies Rest Test.atmx		Description --

General

Name: Transactions Solution.18.08.29.17.17.23.rpaadmin

Description: (Optional)

[Back](#)

Run now (highlighted)

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-15. Run bot now (4 of 5)



Run bot now (5 of 5)

After you run the bot, you can view the status of the run under **Activity > Historical**.

Historical activity

Status	Choose status			
Activity (2 of 2)				
STATUS	DEVICE NAME	AUTOMATION NAME	BOT NAME	USER
<input type="checkbox"/>	Completed	VCLASSBASE	Transactions Solution...	Transactions Solution... botuser1
<input type="checkbox"/>	Completed	VCLASSBASE	Rest Test.18.08.24.1...	Rest Test.atmx botuser1

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-16. Run bot now (5 of 5)

Schedule a bot for execution (1 of 5)

- Scheduling bots allows a Control Room administrator to run bots on a one-time basis or on a regular schedule.
- When you choose **Schedule bot...**, there is no need for a device pool or queue. However, a bot runner device must be connected.

The screenshot shows the 'My bots' section of the IBM Control Room. On the left, there's a sidebar with 'Folders' containing 'My Docs', 'My MetaBots', 'My Reports', and 'My Scripts'. Below that is a 'My Tasks' section. On the right, there's a search bar labeled 'Search name' and a table titled 'Files and folders {2 of 2}' with two entries:

	TYPE	NAME	SIZE	CLIENT LAST MODIFIED	LAST MODIFIED	MODIFIED BY
<input type="checkbox"/>	Task Bot	Rest Test.atmx	10.92 KB	15:21:14 EDT 2018-08-14	12:12:54 EDT 2018-08-24	de
<input type="checkbox"/>	Task Bot	Transactions Solution.atmx	15.79 KB	12:09:08 EDT 2018-08-22	12:12:54 EDT 2018-08-24	de

At the top right, there are buttons for 'Import bots...', 'Export bots...', 'Run bot now...', 'Run bot with queue...', and 'Schedule bot...'. The 'Schedule bot...' button is highlighted with a red box.

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-17. Schedule a bot for execution (1 of 5)



Schedule a bot for execution (2 of 5)

Select a bot and review any dependencies

Schedule bot

BOT + DEPENDENCIES	SCHEDULE + DEVICES	NAME + DESCRIPTION
Bot Transactions Solution Dependencies Rest Test.atmx	Schedule On 2018-08-30 at 12:00 PM EDT Devices --	Name Transactions Solution.18.08.30.11.32.03.rpa... Description --

Select a Task Bot

Folders	Search name				
<ul style="list-style-type: none"> ▶ <input type="checkbox"/> My Docs ▶ <input type="checkbox"/> My Exes ▶ <input type="checkbox"/> My MetaBots ▶ <input type="checkbox"/> My Reports ▶ <input type="checkbox"/> My Scripts ▶ <input checked="" type="checkbox"/> My Tasks ▶ <input type="checkbox"/> My Workflow 	<table border="1"> <thead> <tr> <th>TYPE</th> <th>NAME</th> </tr> </thead> <tbody> <tr> <td><input type="radio"/></td> <td>Task Bot Rest Test.atmx</td> </tr> </tbody> </table>	TYPE	NAME	<input type="radio"/>	Task Bot Rest Test.atmx
TYPE	NAME				
<input type="radio"/>	Task Bot Rest Test.atmx				

Review dependencies for Transactions Solution

▶ Automation Anywhere\My Tasks\Transactions Solution.atmx

Next >

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-18. Schedule a bot for execution (2 of 5)



Schedule a bot for execution (3 of 5)

- Select bot runner device
- Build schedule

Schedule bot

BOT + DEPENDENCIES		SCHEDULE + DEVICES		NAME + DESCRIPTION											
<input type="radio"/> Run once <input checked="" type="radio"/> Run repeatedly		Repeats <input checked="" type="radio"/> Daily <input type="radio"/> Weekly <input type="radio"/> Monthly	Every <input type="text" value="1"/> day(s)	Start date <input type="text" value="2018-08-30"/>	End date <input type="text" value="2018-09-28"/>										
				Start time <input type="text" value="4:00 PM"/> EDT	End time <input type="text" value="11:59 PM"/> EDT										
				<input checked="" type="checkbox"/> Repeat every <input type="text" value="1"/> hour											
				<input type="button" value="Schedule bot"/>											
<p>Upcoming schedules for selected device: VCLASSBASE</p> <p>Scheduled activity (0)</p> <table border="1"> <thead> <tr> <th>TYPE</th> <th>ACTIVITY NAME</th> <th>NEXT OCCU...</th> <th>BOT</th> <th>SCHEDULE</th> </tr> </thead> <tbody> <tr> <td>< </td> <td></td> <td> </td> <td></td> <td> ></td> </tr> </tbody> </table>						TYPE	ACTIVITY NAME	NEXT OCCU...	BOT	SCHEDULE	<				>
TYPE	ACTIVITY NAME	NEXT OCCU...	BOT	SCHEDULE											
<				>											

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-19. Schedule a bot for execution (3 of 5)



Schedule a bot for execution (4 of 5)

Click Schedule bot

Schedule bot

BOT + DEPENDENCIES	SCHEDULE + DEVICES	NAME + DESCRIPTION
Bot Transactions Solution	Schedule Every day at 4:00 PM EDT	Name Transactions Solution.18.08.30.11.32.03.rpaad...
Dependencies Rest Test.atmx	Devices VCLASSBASE	Description --

General

Name Transactions Solution.18.08.30.11.32.03.rpaadmin	Description <small>Optional</small>
--	--

[**< Back**](#)

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-20. Schedule a bot for execution (4 of 5)

Schedule a bot for execution (5 of 5)

After you schedule the bot, you can view the schedule along with any other bot schedules under **Activity > Scheduled**.

Scheduled activity

Run bot now... Run bot with queue... Schedule bot...

You can put a scheduled bot on hold and it will not run for any future occurrence. If it is currently running, you must go to the In progress tab to pause it. If a scheduled bot is on hold, you can remove the hold and it will go back to the 'Ready to run' state.

The screenshot shows the IBM Control Room interface with the 'Scheduled' activity list. A specific bot entry, 'Transactions Solution.18.08.30.12.01.36.rpaadmin', is selected and highlighted with a red box. The 'View activity...' button in the toolbar is also highlighted with a red box. The interface includes a search bar, a toolbar with various icons, and a detailed view of the selected bot's configuration.

Type	Next Occurrence	Activity Name	Bot Name	Schedule	Devices
Recurring	16:00:00 EDT 2018-08-30	Transactions Solution.1...	Transactions Solution.at...	Every day at 4:00 PM	VCLASSBASE
Transactions Solution.18.08.30.12.01.36.rpaadmin <div style="border: 1px solid #ccc; padding: 5px; margin-top: 5px;"> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> BOT + DEPENDENCIES <p>Bot: Transactions Solution.atmx</p> <p>Dependencies: Rest Test.atmx</p> </div> <div style="flex: 1;"> SCHEDULE + DEVICES <p>Schedule: Every day at 4:00 PM EDT</p> <p>Devices: VCLASSBASE</p> </div> <div style="flex: 1;"> NAME + DESCRIPTION <p>Name: Transactions Solution.18.08.30.12.01.36.rpaad...</p> <p>Description: --</p> </div> </div> <hr/> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;"> RUN DETAILS <p>The run details for the schedule will appear here when the schedule has ran at least once.</p> </div> <div style="flex: 1;"> SCHEDULE DETAILS <p>Schedule type: Recurring</p> <p>Next occurrence: 16:00:00 EDT 2018-08-30</p> </div> </div> </div>					

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-21. Schedule a bot for execution (5 of 5)



Run bot with queue (1 of 5)

- As a Control Room user with the appropriate privileges, you can choose to run a bot with the work items present in queues. These work items are collectively processed across all the bot runner devices present in one or more device pools.
- You can **Run a bot with queue** from the **Activity > Scheduled, Bots > My Bots**, and **Workload > Queues** pages.

My bots

[Import bots...](#) [Export bots...](#) [Run bot now...](#) [Run bot with queue...](#) [Schedule bot...](#)

To see files here, upload them from your Bot creator or your Bot runner. You will only see files that you have permission to see.

Files and folders {2 of 2}						
	TYPE	NAME	SIZE	CLIENT LAST MODIFIED	LAST MODIFIED	MODIFIED BY
<input type="checkbox"/>	Task Bot	Rest Test.atmx	10.92 KB	15:21:14 EDT 2018-08-14	12:12:54 EDT 2018-08-24	de
<input type="checkbox"/>	Task Bot	Transactions Solution.atmx	15.79 KB	12:09:08 EDT 2018-08-22	12:12:54 EDT 2018-08-24	de

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-22. Run bot with queue (1 of 5)



Run bot with queue (2 of 5)

Select a bot and review any dependencies

Run bot with queue

BOT + DEPENDENCIES	QUEUE + DEVICE POOL	NAME + DESCRIPTION
Bot Transactions Solution	Queue Transactions	Name Transactions Solution.18.08.30.12.19.21.devuser1
Dependencies Rest Test.atmx	Device pool Transactions	Description --

Select a Task Bot

Folders

- My Docs
- My MetaBots
- My Reports
- My Scripts
- My Tasks

Search name

TYPE	NAME
Task Bot	Rest Test.atmx

Transactions Solut...

Review dependencies for Transactions Solution

Automation Anywhere\My Tasks\Transactions Solution.atmx

Next >

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-23. Run bot with queue (2 of 5)

IBM Training

Run bot with queue (3 of 5)

Select a queue and device pool

Run bot with queue

BOT + DEPENDENCIES	QUEUE + DEVICE POOL	NAME + DESCRIPTION
Bot Transactions Solution	Queue Transactions	Name Transactions Solution.18.08.30.12.19.21.devus...
Dependencies Rest Test.atmx	Device pool Transactions	Description --

Queue

Select a queue

Queue name▼	Search queue name

Available queues (0 of 1)

STATUS	QUEUE NAME ↑	MY ACCESS

Device pool

Select a device pool

Search device pool name

Available device pools (0 of 1)

STATUS	DEVICE POOL NAME ↑	DETAILED STATUS

Back Next

Administering bots through the Control Room © Copyright IBM Corporation 2018, 2019

Figure 14-24. Run bot with queue (3 of 5)



Run bot with queue (4 of 5)

Click Run now

Run bot with queue

BOT + DEPENDENCIES	QUEUE + DEVICE POOL	NAME + DESCRIPTION
Bot Transactions Solution	Queue Transactions	Name Transactions Solution.18.08.30.12.19.21.devuser1
Dependencies Rest Test.atmx	Device pool Transactions	Description --

General

Name Transactions Solution.18.08.30.12.19.21.devuser1	Description Optional
--	-------------------------

Cancel **Run now**

◀ Back

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-25. Run bot with queue (4 of 5)



Run bot with queue (5 of 5)

After you queue the bot for execution, you can view the status of the queue under **Workload > Queues**.

Queues

The screenshot shows the 'Queues' page in the IBM Control Room. A single queue named 'Transactions' is listed. The 'Work items' section is expanded, showing 15 work items. One work item is selected, showing details like ID 1, STATUS Active, and DATE Fri Jan 26 00:00:00 ES... 87431. The 'View activity...' button is highlighted with an orange box. The bottom right corner of the screenshot contains the copyright notice: © Copyright IBM Corporation 2018, 2019.

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-26. Run bot with queue (5 of 5)

14.3. Bot Insights Overview

Bot Insights overview

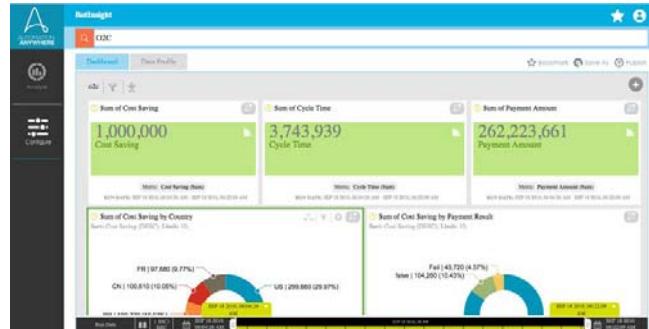
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-27. Bot Insights Overview

Bot Insights overview (1 of 4)

- Bot Insights provides the means to interactively analyze large amounts of real-time data that bots generate.
- Customers need to buy appropriate subscriptions for the Bot Insight capability and for Bot Insight users before they are entitled to use variable tagging, Bot Insight APIs, and Bot Insight dashboards for business operations



- 1 **QUICK TAGGING:** Tag any variable of your choice. The bot logs all the variables that are tagged.
- 2 **INSTANT DASHBOARDS:** Jump-start bot data analysis and visualizations without IT help.
- 3 **TAILORED DASHBOARDS:** Customize auto-generated dashboards.
- 4 **INTERACTIVE ANALYSIS:** Publish the customized dashboards for business consumption.
- 5 **CENTRAL MANAGEMENT:** Monitor bot performance, troubleshoot bots and bot runners.

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-28. Bot Insights Overview (1 of 4)

Bot Insights overview (2 of 4)

- Business Analytic Features
 - Autogenerated business dashboards and KPI
 - Customizable and actionable
 - Business analysts can jump-start bot data analysis
- Benefits
 - Quantify business value added by bots
 - Identify new opportunities from business insights
 - Improve business process (not) efficiency
 - Low-code development environment and intelligent dashboard generation



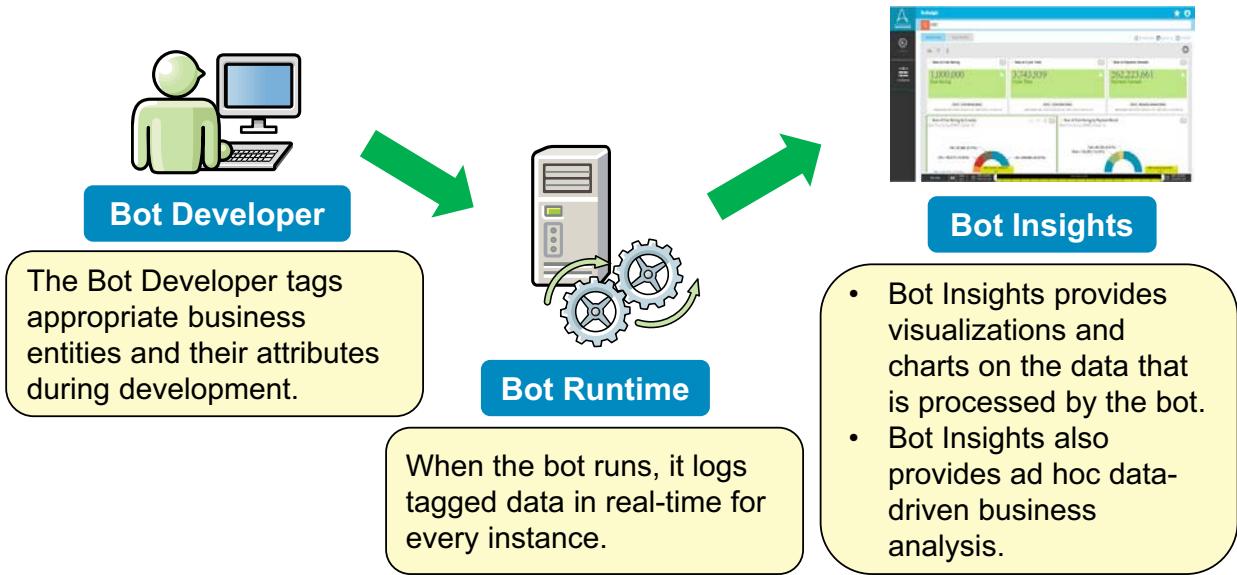
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-29. Bot Insights Overview (2 of 4)

Bot Insights overview (3 of 4)

- The following diagram represents the steps necessary to retrieve Analytics data from an automation task when using Bot Insights.



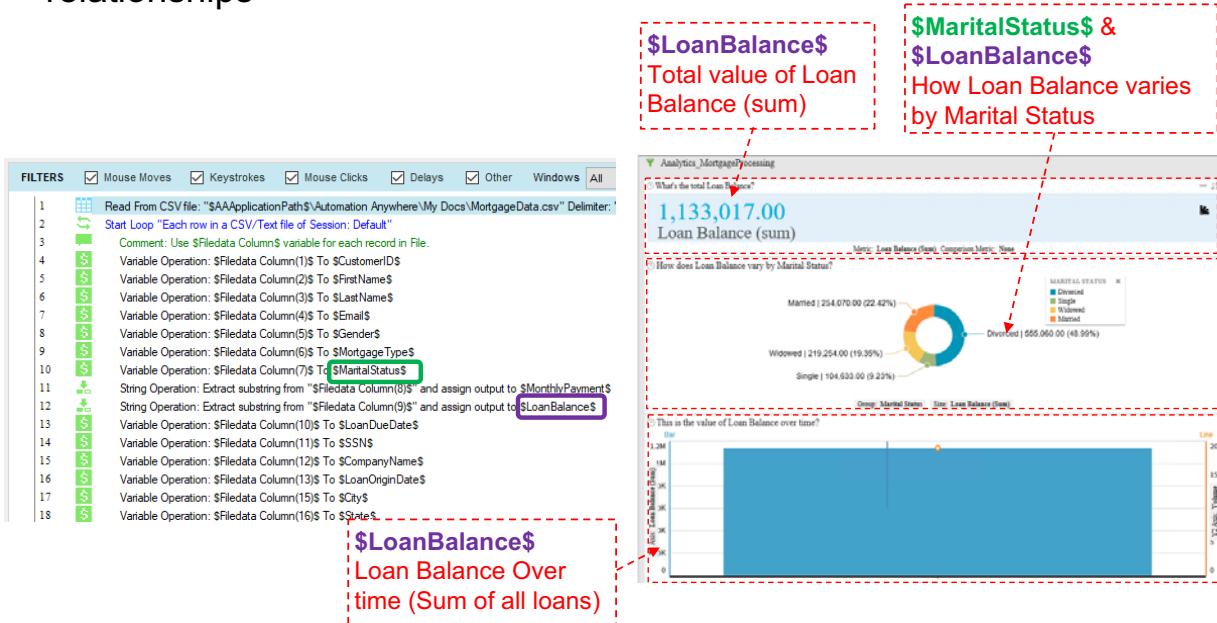
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-30. Bot Insights Overview (3 of 4)

Bot Insights overview (4 of 4)

- From the Task variables, besides simple metric visualizations, the generated Dashboard contains KPIs and visualizations that involve relationships



Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-31. Bot Insights Overview (4 of 4)

14.4. Bot lifecycle management

Bot lifecycle management

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-32. Bot lifecycle management

Bot lifecycle management overview (1 of 2)

- As a Control Room user with appropriate privileges, you can move your bots from one environment to another. For example, you can move bots that are verified as production ready from staging to production.
- The process is performed in two stages:
 - Export Bots from a source Control Room
 - Import Bots to a destination Control Room
- Bot Lifecycle Management (BLM) enables organizations to have separate automation environments like: Development, Testing, Acceptance, Production

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-33. Bot lifecycle management overview (1 of 2)

Bot lifecycle management overview (2 of 2)

- Allows bots to graduate through its lifecycle stages
- Helps organizations meet compliance mandates such as Capability Maturity Model Integration (CMMI) Level 5 certification and Sarbanes-Oxley (SOX)
- Key Features & Business Benefits:
 - Move multiple bots, their dependent bots, and files between Control Rooms.
 - Role-based granular access to import and export bot packages.
 - API support for BLM automation

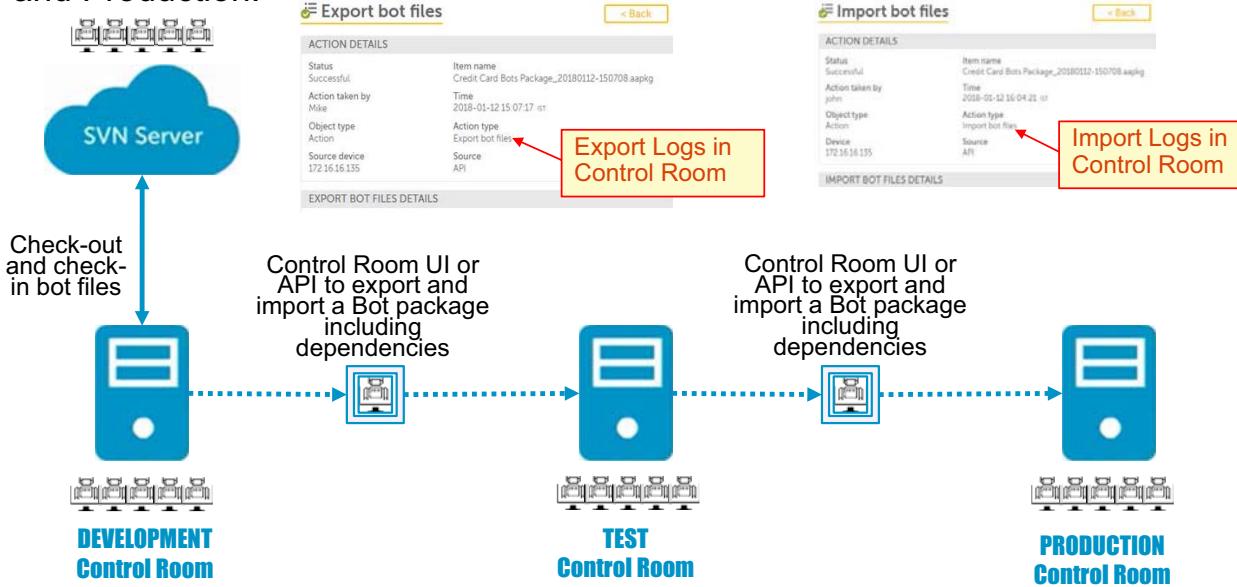
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-34. Bot lifecycle management overview (2 of 2)

Bot lifecycle management workflow

This slide displays a graphical view of bot lifecycle management in action. Using an SVN server, a Developer can check out their bot files. These files can then be imported into other environments including Development, Test, and Production.



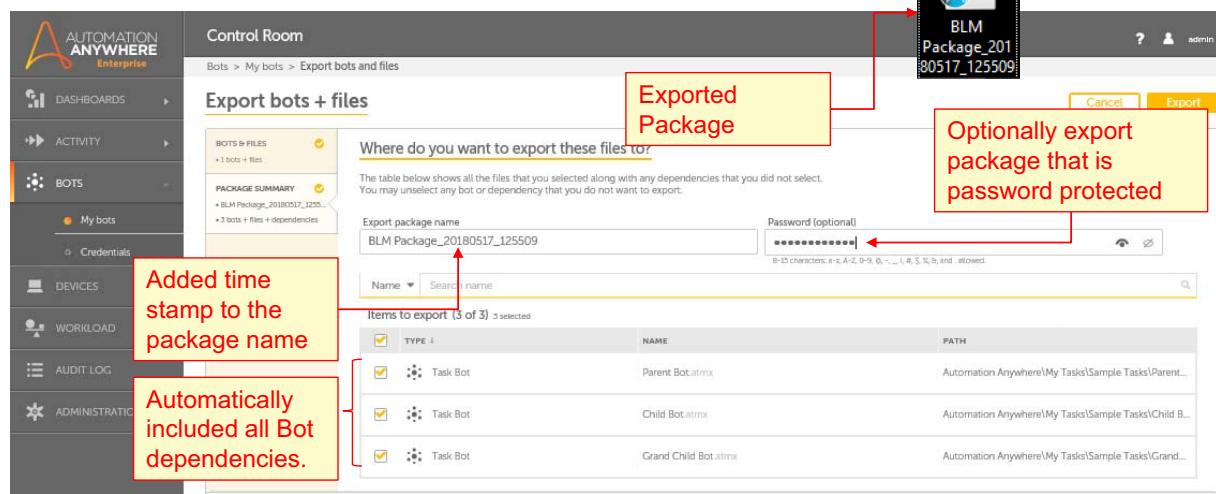
Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-35. Bot lifecycle management workflow

Exporting bot deployment package

- As a Control Room user with the appropriate privileges, you can export bots and dependent files in different automation environments to help manage your organization's Bot Lifecycle Management (BLM).
- The exported package can then be imported in another Control Room environment.



Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-36. Exporting bot deployment package



Importing bot deployment package

- As a Control Room user with the appropriate privileges, you can import bots and dependent files that were exported by another Control Room user in different automation environments to help manage your organization's Bot Lifecycle Management (BLM).

The screenshot shows the Automation Anywhere Control Room interface. On the left, the navigation menu includes DASHBOARDS, ACTIVITY, BOTS (selected), DEVICES, WORKLOAD, AUDIT LOG, and ADMINISTRATION. Under BOTS, 'My bots' is selected. In the center, the 'Control Room' page displays the 'Bots > My bots > Import bot files' section. A red box highlights an error message: 'Unable to import the bots.' It states: 'Package is not imported because following bots already existed in Control Room:' followed by a bulleted list: 'Automation Anywhere\My Tasks\Sample Tasks\Child Bot.atmx', 'Automation Anywhere\My Tasks\Sample Tasks\Grand Child Bot.atmx', and 'Automation Anywhere\My Tasks\Sample Tasks\Parent Bot.atmx'. Below the message is a note: 'Please select Skip or Overwrite option and try again.' To the right, a file icon labeled 'BLM Package_20180517_125509.aapkg' is shown with a red arrow pointing to it. At the bottom right of the import dialog are 'Cancel' and 'Import' buttons. A red box highlights the 'Cancel import of the Bot or its dependencies' radio button under 'During import, if a file already exists' options.

© Copyright IBM Corporation 2018, 2019

Figure 14-37. Importing bot deployment package

Unit summary

- Describe how to manage bot workload using bot runners, device pools and queues.
- Explain how to run and schedule bots from the Control Room
- Describe how to monitor bot performance using Bot Insights
- Understand bot lifecycle management

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-38. Unit summary

Review questions



1. True or false. A connected bot runner device is necessary to run a bot from the Control Room.
2. True or false. A queue must be created before a device pool can be configured.
3. _____ provide a logical grouping of similar bot runners on which you can run bots with the work item from a queue.
 - A. Device pools
 - B. Queues
 - C. Work Items
 - D. Bot Lifecycle Management

[Administering bots through the Control Room](#)

© Copyright IBM Corporation 2018, 2019

Figure 14-39. Review questions

Review answers

1. True or false. A connected bot runner device is necessary to run a bot from the Control Room.

The answer is True.



2. True or false. A queue must be created before a device pool can be configured.

The answer is False.

3. _____ provide a logical grouping of similar bot runners on which you can run bots with the work item from a queue.

- A. Device pools
- B. Queues
- C. Work Items
- D. Bot Lifecycle Management

The answer is A.

Exercise 12: Administering bots in the Control Room

Administering bots through the Control Room

© Copyright IBM Corporation 2018, 2019

Figure 14-41. Exercise 12: Administering bots in the Control Room

Exercise objectives

- Build device pools, queues, queue categories, and work items.
- Run and schedule bots on bot runner devices from the Control Room.
- View in progress, scheduled, and historical bot activity.



[Administering bots through the Control Room](#)

© Copyright IBM Corporation 2018, 2019

Figure 14-42. Exercise objectives

Unit 15. Course summary

Estimated time

00:30

Overview

This unit summarizes the course and provides information for future study.

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-1. Unit objectives

Course objectives

- Describe the IBM Robotic Process Automation with Automation Anywhere Control Room and Enterprise Client
- Work with the various Enterprise Client Recorders and commands to create and manage bots

[Course summary](#)

© Copyright IBM Corporation 2018, 2019

Figure 15-2. Course objectives

Command matrix: Commands covered

- This command matrix displays all commands that are used in this course during bot development
- Exercises 3-10 cover bot development
 - Most of this course focuses on bot development
 - The ratio in which the commands are used follow typical development patterns
 - For example, Looping and the If/Else commands are used extensively, which are also used extensively in real-world situations
- Exercises 4-5 use the Smart Recorder
- Exercise 7 uses the Web Recorder
- Exercise 8 is a MetaBot exercise
- Exercises 11 and 12 deal with bot management and administrative topics

COMMAND	EXERCISE LAB USAGE									
	3	4	5	6	7	8	9	10		
Clipboard			X							X
Comment	X	X	X	X	X				X	X
Database			X	X						
Email Automation					X					
Error handling									X	X
Excel	X	X				X			X	
Files/folders					X				X	
If/else			X	X	X	X			X	X
Insert keystrokes				X						X
Loop	X	X	X	X	X					
Message Box							X			
Object cloning	X	X				X				X
Open program file	X	X				X				
PDF integration					X					
Prompt							X			
Read from CSV/Text	X		X							
REST web service									X	
Send email			X	X					X	X
String operation	X	X			X				X	X
Task							X			
Variable operation	X	X	X			X			X	
Web recorder							X			
Window actions	X	X				X				X

© Copyright IBM Corporation 2018, 2019

Course summary

Figure 15-3. Command matrix: Commands covered

Command matrix: Commands not covered

- This command matrix displays commands that are NOT used in this course during bot development

COMMAND	COMMAND
Active Directory	Manage windows controls
App Integration	OCR
Citrix Automation	PGP
Delay	Play sound
FTP/SFTP	Printers
Image recognition	Run script
Insert mouse click	SAP integration
Insert mouse move	Screen capture
Insert mouse scroll	Services
Insert Work Item	SNMP
Internet connection	SOAP web service
IQ Bot	System
Launch website	Terminal Emulator
Log to file	Wait
	XML

Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-4. Command matrix: Commands not covered

To learn more on the subject

- IBM Training website:
www.ibm.com/training
- IBM Robotic Process Automation with Automation Anywhere V11 documentation
<https://support.automationanywhere.com/hc/en-us/articles/230199268-Welcome>
- IBM Business Process Manager Developer Center
<https://developer.ibm.com/bpm/docs/ibm-robotic-process-automation-automation-anywhere>
- Other IBM websites:
 - www.ibm.com/middleware
 - www.redbooks.ibm.com
 - www.ibm.com/developerworks

Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-5. To learn more on the subject

Additional resources

- IBM Cloud RPA AAI Practitioners Slack Channel:
<https://ibm-cloud.slack.com/messages/C7D3PNWN8>
- Automation Anywhere Bot Store
 - Public Marketplace for Partners & Developers
 - 100+ Bots that are divided into categories with IP Protection
 - After you register, you can download any Bot
<https://botstore.automationanywhere.com/>
- IBM RPA with Automation Anywhere V11.0 documentation
https://www.ibm.com/support/knowledgecenter/en/SSMGNY_11.0/com.ibm.wbpm.rpa.main.doc/kc-homepage-rpa.html

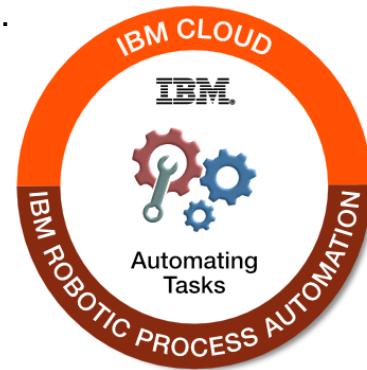
Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-6. Additional resources

Earn an IBM Badge

- After completing this course, you might be ready to take an IBM Badge test.
- Use IBM Badges to share verified proof of your IBM credentials.
- Find your Badge test on this site:
 - <https://www.ibm.com/services/learning/ites.wss/zz-en?pageType=badgesearch>
- After completing this course, take the Badge test.
- The earner of this badge understands how to use IBM Robotic Process Automation with Automation Anywhere software and is able to work with the various Enterprise Client Recorders and commands to create and manage bots.



Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-7. Earn an IBM Badge

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
 - Product information
 - Training and certification
 - Documentation
 - Support
 - Technical information



- To learn more, see the IBM Cloud Education Resource Guide:
 - www.ibm.biz/CloudEduResources

Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-8. Enhance your learning with IBM resources

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Course summary

© Copyright IBM Corporation 2018, 2019

Figure 15-9. Unit summary

IBM Training

IBM

Course completion

You have completed this course:

Automating Tasks Using IBM Robotic Process Automation with
Automation Anywhere

Any questions?



[Course summary](#)

© Copyright IBM Corporation 2018, 2019

Figure 15-10. Course completion



IBM Training



© Copyright International Business Machines Corporation 2018, 2019.