

IBM Runbook Automation

IBM Runbook Automation Guide



Contents

List of Figures.....	vii
List of Tables.....	ix
Chapter 1. Flags used in the Runbook Automation Knowledge Center.....	1
Chapter 2. Getting started.....	3
What is IBM Runbook Automation?.....	3
Runbook types.....	3
Automations.....	3
Runbook example.....	4
What's new.....	4
Differences between an RBA Cloud and an RBA Private Deployment.....	7
System requirements.....	8
Lifecycle of a runbook.....	8
Access IBM Runbook Automation.....	9
Installing and configuring software to access your backend systems.....	9
Configure browser settings for Internet Explorer 11.....	9
Create users and assign roles.....	10
Using keyboard shortcuts.....	11
RBA Private Deployment considerations for GDPR readiness.....	12
Chapter 3. Installing, configuring, and troubleshooting a Private Deployment.....	17
Overview.....	17
Planning for RBA installation.....	18
System prerequisites.....	18
Software prerequisites	19
Docker considerations.....	19
Planning for RBA DASH integration.....	21
System prerequisites.....	21
Software prerequisites	21
Create users and groups in the external user repository with read-only access.....	21
Planning for IBM BigFix.....	22
Firewall requirements.....	22
Planning for single sign-on.....	24
Planning for Backup/Restore.....	24
Planning for Database Encryption.....	25
Installing.....	25
Installing the RBA service.....	26
Installing the RBA DASH integration.....	27
Post-installation steps for the RBA DASH integration.....	27
Getting started.....	28
Updating the installation.....	29
Updating the RBA service.....	29
Updating the RBA DASH integration.....	29
Configuring.....	29
Configuring the RBA service.....	30
Configuring the RBA DASH integration.....	31
Configuring a setup with multiple Netcool/OMNIbus Web GUI servers.....	32
Configuring an IBM Netcool/OMNIbus event tool for launch-in-context to IBM Runbook Automation.....	32

Configuring certificate validation.....	36
Pre-installation configuration for WebSphere Application Server single sign-on (SSO).....	37
Initial configuration and manual secure configuration.....	38
Level of security after initial installation.....	38
Generate and install signed RBA server certificate.....	39
Generate root CA key and certificate.....	39
Import root CA certificate into the browser.....	40
Generate RBA server key and certificate request.....	40
Generate the RBA server certificate.....	41
Install the RBA server certificate on the RBA server.....	42
Replace the RBA server certificate in the WAS trust store.....	42
Replace the RBA server certificate in Netcool/Impact.....	43
Test the signed RBA server key and certificate.....	43
Renewing the signed RBA server certificate.....	43
Enable secure communications within the RBA service.....	44
User groups and roles.....	45
Creating, adding, and removing WebSphere users.....	47
Operating the RBA service.....	47
Cleaning up old docker images.....	48
Uninstalling.....	49
Uninstalling the RBA service.....	49
Uninstalling the RBA DASH integration.....	49
Troubleshooting.....	49
Collecting problem determination information for the RBA service.....	49
Writing RBA service traces to log file instead of syslog.....	50
Enabling and collecting problem determination information for the RBA DASH integration.....	50
Messages in /var/log/messages.....	51
RBA service error scenarios.....	51
RBA DASH integration error scenarios.....	52
Verifying the DASH console integration to the RBA service.....	53
RBA service directories.....	55
Chapter 4. Users and their daily tasks.....	57
Who works with runbooks?.....	57
Access control.....	57
Assigning runbooks to groups.....	58
Run and monitor runbooks.....	59
Create and edit a runbook.....	59
Run steps automatically.....	60
Map Netcool/OMNIbus events to runbooks.....	60
Monitor runbook status in Netcool/OMNIbus events.....	60
Configure the Netcool/OMNIbus Event Viewer to include the RunbookStatus column.....	60
Values of the RunbookStatus field.....	61
State transitions of the RunbookStatus field.....	61
Improving runbook execution.....	63
Export and import runbooks.....	64
Example: complete export and import.....	65
Migrating data from RBA Cloud to RBA Private Deployment and vice versa.....	65
Chapter 5. Library.....	67
Run your first runbook.....	67
Load and reset example runbooks.....	68
Create a runbook.....	69
Run a runbook.....	70
Edit a runbook.....	70
Editing actions.....	71
Adding automations.....	72

Adding parameters.....	74
Adding a GOTO element.....	75
Configuring runbook creation.....	76
Configuring runbook approval.....	76
Runbook approval process.....	76
Runbook versions.....	78
Save and publish a runbook.....	78
Filtering runbooks.....	79
Delete a runbook.....	79
Chapter 6. Executions.....	81
Monitor runbook history.....	81
Chapter 7. Automations.....	83
Create an automation.....	83
Create an automation of type script.....	83
Create an automation of type BigFix.....	86
Create an automation of type HTTP.....	87
Manage existing automations.....	88
Chapter 8. Connections.....	89
Create a connection.....	89
Event Trigger (Netcool/Impact).....	89
IBM BigFix automation provider	90
SSH script automation provider.....	90
Install the IBM Secure Gateway client.....	92
Configure the IBM Secure Gateway client.....	93
Installing Netcool/Impact to run the trigger service.....	94
Postinstallation of Netcool/Impact V7.1.0.13 or higher.....	94
Import certificate.....	96
Configure user access rights.....	97
Update the Netcool/Impact configuration.....	98
Enable the launch-in-context menu.....	100
Configure a connection to the Netcool/Impact server.....	101
Upgrading Netcool/Impact to a higher fix pack level when the RBA add-on is already installed...	102
Migrating from IBM Workload Automation Agent to SSH.....	103
Chapter 9. Triggers.....	105
Create a trigger.....	105
Triggers.....	106
Chapter 10. Runbook Automation API.....	109
Creating or modifying a runbook with the API.....	110
API usage scenario.....	112
API Keys.....	113
Chapter 11. IBM Runbook Automation FAQ.....	115

List of Figures

- 1. Moving from manual to fully automated runbooks.....8
- 2. Server topology for IBM Runbook Automation..... 17
- 3. Firewall considerations..... 23
- 4. State transitions of the RunbookStatus field.....62
- 5. Working areas of the runbook editor..... 71
- 6. Runbook approval process.....77

List of Tables

- 1. Differences between a Cloud and Private Deployment.....7
- 2. RBA User roles and tasks..... 10
- 3. RBA Author roles and tasks..... 10
- 4. RBA Approver roles and tasks..... 11
- 5. RBA Manager roles and tasks..... 11
- 6. Common user interface button shortcuts..... 11
- 7. Keyboard shortcut for runbook elements12
- 8. Runbook Automation components..... 18
- 9. Free disk space requirements.....19
- 10. Free disk space requirements..... 21
- 11. RBA DASH integration user groups22
- 12. Firewall requirements..... 23
- 13. RBA installation sequence..... 26
- 14. Configuration parameters..... 30
- 15. Installation topology and tool type33
- 16. RBA DASH integration users and groups.....46
- 17. RBA service error scenarios..... 51
- 18. RBA DASH integration error scenarios..... 53
- 19. RBA service directories..... 55
- 20. Values of the RunbookStatus field..... 61
- 21. Editing actions..... 71
- 22. Comparison of Single and Multi Target Automations..... 86
- 23. Default values of Netcool/Impact derby database..... 98

Chapter 1. Flags used in the Runbook Automation Knowledge Center

The information contained in the IBM Runbook Automation Knowledge Center applies to all editions of Runbook Automation unless otherwise specified. Where content relates only to a specific edition of Runbook Automation, the flags described below are used to indicate which edition.

Flags used in the Runbook Automation Knowledge Center
Cloud : Information applies to the Runbook Automation managed service in the cloud.
Private Deployment : Information applies to a Runbook Automation Private Deployment.
Cloud Event Management : Information applies to the Runbook Automation component of IBM Cloud Event Management.

Note: By default, any content marked with the **Cloud Event Management** flag also applies to Cloud Event Management on IBM Cloud Private.

Chapter 2. Getting started

Find out how IBM Runbook Automation can help you to improve your daily operational tasks. Learn what a runbook is used for and how to access IBM Runbook Automation.

What is IBM Runbook Automation?

Use IBM Runbook Automation to build and execute runbooks that can help IT staff to solve common operational problems. IBM Runbook Automation can automate procedures that do not require human interaction, thereby increasing the efficiency of IT operations processes. Operators can spend more time innovating and are freed from performing time-consuming manual tasks.

Which problem does IBM Runbook Automation address?

IT systems are growing. The number of events are increasing, and the pressure to move from finding problems to fixing them is increasing. IBM Runbook Automation supports operational and expert teams in developing consistent and reliable procedures for daily operational tasks.

How can I simplify daily operational tasks?

Using IBM Runbook Automation, you can record standard manual activities, so that they are run consistently across the organization. The next step is to replace manual steps with script-based tasks.

What is a runbook?

A runbook is a controlled set of automated and manual steps that support system and network operational processes. A runbook orchestrates all types of infrastructure elements, like applications, network components, or servers.

IBM Runbook Automation helps users to define, build, orchestrate, and manage Runbooks and provides metrics about runbooks. A knowledge base is built up over time and the collaboration is fostered between Subject Matter Experts (SMEs) and operational staff.

[Video: Why IBM Runbook Automation Makes IT Ops Teams Efficient](#)

Runbook types

Transfer your procedure based documentation to manual runbooks. Then add automation and continue with semi-automated runbooks or fully automated runbooks.

IBM Runbook Automation supports three different types of runbooks:

Manual runbooks

Steps describe the exact procedure that an operator must follow. The operator uses standard tools, which can be accessed from their working environment.

Semi-automated runbooks

Each step describes exactly what an operator must do. Additionally, the operator can run an automated task on a target system.

Fully automated runbooks

The runbook runs automatically without operator interaction.

Automations

Automation is a task that runs autonomously on a target system. For example, a shell script that identifies large files in a file system.

IBM Runbook Automation supports the following technologies to connect to a backend system:

IBM BigFix

IBM BigFix is an automation provider. Use tasks or fixlets to run on your target systems.

Script Automation Provider

Use scripts to run on your target system.

IBM Netcool/OMNIBus

A trigger puts a runbook into context by associating it with a particular problem that is represented by an event. A runbook that provides instructions to remediate a particular event might be triggered by such an event. Ideally, information from the event is provided as input to the runbook, so that the runbook itself is contextualized.

Runbook example

The following example outlines a procedure for finding large files in a file system on Linux.

1. Open a PuTTY session to host `ibm1balx49`.
2. Enter the following command to navigate to the file system that is full:

```
cd
```

3. Enter the following command to identify core files in the file system:

```
find . -name core.*
```

4. Enter the following command to identify large files in the file system:

```
du -hsm . | sort -nr
```

What's new

Find out about features that are new or updated in IBM Runbook Automation.

24 August 2018

- Significantly enhanced Runbook Automation User Interface:
 - Enhanced search capabilities for runbooks and runbook executions.
 - New **Library** page replaces the Runbooks page. The Library page lists all runbooks and provides enhanced filtering capabilities including drill down to a specific set of runbooks, combining several filters such as groups, runbook status, tags, and name.
 - New **Execution** page lists the runbook execution history. The Execution page provides enhanced filtering capabilities including drill down by user, runbook name, runbook type, or runbook execution status.
 - New **History** page. The History page allows you to see runbook executions for a particular runbook, including further details such as runbook rating and statistics.
 - Enhanced user experience with multi selection on Library and Execution pages.
- Enhanced **Connections** page:
 - Simplified and more intuitive configuration and editing of connections.
 - The user flow of the script connection configuration has been improved significantly.
- Performance improvements:
 - Significantly improved performance in the execution of fully automated runbooks, in particular under high load conditions.
 - Introduced pagination for tables to achieve a faster and smoother user experience on the pages of **Library**, **Execution**, **Automations**, and **Triggers**.
- Enhanced API:
 - New API to query runbook instances.
- Cloud Event Management:
 - API keys configured in Cloud Event Management can also be used for Runbook Automation.

- Special characters can now also be used for Launch-in-Context scenarios using B64 encoding. This allows you to pass parameters to runbooks that contain special characters such as backslash, slash, and so on.

Note: IBM Netcool/Impact fixpack 7.1.0.13 is required for this functionality.

- Installation Manager for the Runbook Automation DASH integration now supports group mode.

17 May 2018

- Multi-Target Capability for SSH automations
 - It is now possible to specify multiple targets to execute an SSH automation on several systems at once. This significantly reduces the effort and time required for similar tasks executed on multiple systems and allows you to control execution targets based on the output of previous automation results.
 - Learn more about the capabilities of this feature in “Single and Multi Target Automations” on page 84 and on the RBA Center of Excellence at <https://developer.ibm.com/rba/currel/>.
- Conditional Automations
 - By specifying an empty list of targets, an automation can be skipped during runbook execution. This can be used in fully automated runbooks to control the workflow of the runbook and skip automations if they are not necessary.
 - Learn more about the capabilities of this feature in “Single and Multi Target Automations” on page 84 and on the RBA Center of Excellence at <https://developer.ibm.com/rba/currel/>.
- Automated JSON formatting of automation output
 - If an automation produces JSON output, the JSON document is automatically formatted for better readability.
- IBM Workload Automation agent discontinued
 - Script automation via the TWA Agent has been discontinued and is no longer available.
 - For information about how to migrate script automations from using the TWA infrastructure to direct SSH, see “Migrating from IBM Workload Automation Agent to SSH” on page 103.
- Enhanced API
 - GET Status: A new API call to query and monitor the status of the Runbook Automation application is available. For more information, see <https://rba.mybluemix.net/swagger?url=https://rba.mybluemix.net:443/doc-v1>.
 - GET Runbook Instances: Retrieve multiple runbook instances from your subscription at once. This call supports multiple options to filter the results, for example based on time frame. This allows you to retrieve precisely the data required.

8 March 2018

- Improved user experience when running runbooks:
 - Runbook elements such as parameter, command, goto, and automation are always visible on a collapsible sidebar with links to where the elements are used in the runbook.
 - Improved visualization of completed, current, and future steps.
 - You can provide feedback immediately after the last step of a runbook.
 - Easily move to the next runbook step after the completion of a step.
 - For a brief overview of user interface changes, go to <https://developer.ibm.com/rba/currel/>.
- SSH provider:
 - SSH script automations now contain a "user" parameter to define which user should connect to the endpoint.
 - You can use the user name of the currently logged in user as the UNIX user name to execute scripts.

- API enhancements:
 - Added a new call to retrieve the status of the Runbook Automation application.
 - For more information, see the API documentation at <https://rba.mybluemix.net/swagger?url=https://rba.mybluemix.net:443/doc-v1>.
- Security fixes:
 - Upgraded to CouchDB Version 2.1.1 due to security issues in CouchDB 2.0.

7 December 2017

- Adapted User Interface to Carbon Design.
- Improved browser support for Internet Explorer.
- Improved browser support for Safari 11.
- Configure a jump server with the SSH automation provider.
- Runbook Automation Badges are available at Acclaim:
 - IBM Runbook Automation Explorer. For more information, see <https://www.youracclaim.com/org/ibm/badge/ibm-runbook-automation-explorer>.
 - IBM Runbook Automation Practitioner. For more information, see <https://www.youracclaim.com/org/ibm/badge/ibm-runbook-automation-practitioner>.
- New functions for Cloud Event Management:
 - Preview a runbook before executing the runbook.
 - Execute fully automated runbooks with a Cloud Event Management event policy.
- New API to retrieve usage statistics for runbook instances.

29 September 2017

- Multiple automation steps are permitted in a fully automated runbook.
- API documentation on Swagger. See <https://rba.mybluemix.net/swagger?url=https://rba.mybluemix.net:443/doc-v1>.
- The result of an automation is passed as a parameter to the next step(s).
- New and improved import and export of runbooks.
- SSH Provider is available as an automation provider for script automation, as an alternative to the deprecated TWA provider.
- SSH Provider has been enhanced - scripts no longer require root access to run on the target machine.
- For Private Deployment installations, it is recommended to backup the RBA service data before updating. For more information, see [“Planning for Backup/Restore”](#) on page 24.
- Enhanced documentation.

30 June 2017

- Improved user interface following Bluemix design guidelines.
- New page for creating, editing, running, and previewing runbooks.
- New page for creating connections and automations.
- New feature in approval process: approved runbooks expire after a configured time period.
- Updates required for including Runbook Automation in the new Bluemix BETA offering of **Cloud Event Management**.
- The deprecated gateway using the Hybrid Cloud Connector is no longer supported. Use IBM Secure Gateway instead.

- Execute Automations: new SSH automation provider eliminates the requirement of a Workload Automation agent client to execute scripts. This new automation provider is only available on request to IBM.

19 May 2017

- HTTP Automation Provider allows automations of type "http".
- Fix for Japanese locale used in Automation target endpoints.
- Fix for IBM Workload Automation agent not working with latest version of IBM Secure Gateway.

28 March 2017

- Initial version of the Runbook Automation Private Deployment option. RBA can be downloaded and installed in a network isolated customer environment. It is installed, operated, and used by the customer on their own system.

20 March 2017

- Improved user interface following Bluemix design guidelines.
- New approval process for runbooks.
- Various bug fixes.

16 December 2016

- Replaced gateway technology with IBM Secure Gateway (<https://console.ng.bluemix.net/catalog/services/secure-gateway>).
- Various bug fixes.

15 July 2016

- Various improvements and fixes for stability and usability.

29 April 2016

- Initial version goes live as a service on IBM Marketplace.
- Create and execute Runbooks.
- Execute Automations - supporting script and IBM BigFix automations.
- Runbooks can be triggered by Netcool/OMNIBus events.

Differences between an RBA Cloud and an RBA Private Deployment

Be aware of the differences between IBM Runbook Automation running in the Cloud and an IBM Runbook Automation Private Deployment.

The following table outlines the main differences between the two deployments. Some of the variances are due to differences in architecture. For example, RBA Private Deployment uses IBM WebSphere Application Server authorization and authentication mechanisms instead of advanced user management services that are available in the Cloud.

<i>Table 1: Differences between a Cloud and Private Deployment</i>		
Function	RBA Cloud	RBA Private Deployment
User Management Managing users, authentication, and authorization.	User Management is performed within the web application.	User Management, authentication, and authorization is performed within IBM WebSphere Application Server.

Table 1: Differences between a Cloud and Private Deployment (continued)

Function	RBA Cloud	RBA Private Deployment
Approval workflow This function enables an approval chain for new or changed runbooks.	Supported	Not supported in the current release.
Group Management This function allows you to assign runbooks to groups. Only the users who belong to a group can see the runbooks assigned to that group.	Supported	Not supported in the current release.
Connections Connectivity to automation providers and Netcool/Impact in the data center.	Requires IBM Secure Gateway client.	Does not require the IBM Secure Gateway client.

System requirements

IBM Runbook Automation requires a supported browser.

Cloud and **Private Deployment** The following browsers are currently supported:

- Chrome 57 or later
- Firefox 45.8 ESR
- Firefox 52.0.1 ESR
- Internet Explorer 11

Always check the support life cycle of your preferred browser.

Private Deployment

- For RBA server, see “Planning for RBA installation” on page 18.
- For RBA DASH integration, see “Planning for RBA DASH integration” on page 21.

Lifecycle of a runbook

Runbooks start as documented procedures on a piece of paper that can become fully automated procedures. Find out how can you move your documented procedures to fully automated runbooks.

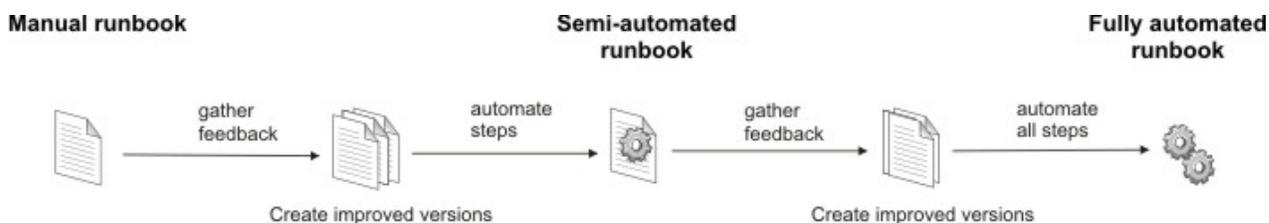


Figure 1: Moving from manual to fully automated runbooks

The following steps outline the process of moving from documented procedures to fully automated runbooks:

1. Transfer your documented procedures into a runbook.
For more information, see [“Create a runbook” on page 69](#).
2. Assess the runbook and gather feedback. You can get information about the quality and the performance of the runbook by analyzing available metrics. For example, ratings, comments, success rate, and more.
For more information, see [“Improving runbook execution” on page 63](#).
3. Create improved versions based on the feedback. With each new version, improve the success rate.
For more information, see [“Runbook versions” on page 78](#).
4. Investigate which runbooks are suitable for automations. What steps can be put into an automated procedure? Not every runbook moves from a manual to a fully automated runbook. Carefully consider which steps you want to automate, depending on the comments received, the effort it takes to automate, the frequency the runbook is used, and the potential impact automated steps could have.
For more information, see [“Create an automation” on page 83](#).
5. Provide a new version with automated steps and see how it runs.
For more information, see [“Manage existing automations” on page 88](#).
6. Continue to provide more automations until all steps are automated. Run runbooks that are started by a trigger.
For more information, see [“Create a trigger” on page 105](#).

Access IBM Runbook Automation

You access IBM Runbook Automation through your internet browser with the URL you received from your purchase or trial request.

A welcome page is presented for getting started with IBM Runbook Automation. Follow the guidance given in the welcome page and load the sample runbooks to get started.

For more information about how to load the sample runbooks, see [“Load and reset example runbooks” on page 68](#).

Installing and configuring software to access your backend systems

If you want to run runbooks in the cloud which contain automations of type script or type BigFix, you must install additional software on the backend hosts so that IBM Runbook Automation can automatically access your backend system. To trigger runbooks from OMNIBus/Impact you must install Netcool/Impact.

The following software must be installed:

- **Cloud** [“Install the IBM Secure Gateway client” on page 92](#)
- [“Installing Netcool/Impact to run the trigger service” on page 94](#)

Configure browser settings for Internet Explorer 11

You must configure browser settings for Internet Explorer 11 to set up IBM Runbook Automation.

Complete the following steps to add the **IBM Agile Operations** URL to your Internet Explorer 11 settings:

1. Open your Internet Explorer browser.
2. Navigate to **Tools > Internet options > Privacy**.
3. Select **Sites**.

4. Add the IBM Agile Operations URL.
5. Click **Allow**.
6. Click **OK**.

Create users and assign roles

Add the users that are working with IBM Runbook Automation and assign the appropriate roles for their job responsibilities.

For more information about adding users, see [Managing users and groups](#).

Note: A user with the RBA Manager role requires an additional role of either RBA Approver or RBA Author.

The following tables describe the roles of IBM Runbook Automation and the tasks that are assigned to the role:

Task	Documentation
Preview and run a runbook.	“Run your first runbook” on page 67
Monitor all runbooks, which are run by the user that is logged on.	Chapter 6, “Executions,” on page 81
Add comments about the quality of runbooks, after the runbook is completed.	“Improving runbook execution” on page 63
Export runbooks.	“API Keys” on page 113

Tasks	Documentation
Preview and run a runbook.	“Run your first runbook” on page 67
Add and edit the runbook.	“Create a runbook” on page 69 “Edit a runbook” on page 70
Monitor all runbooks, which are run by the user that is logged on.	Chapter 6, “Executions,” on page 81
Monitor all runbooks that are run by all users.	“Monitor runbook history” on page 81
Edit and create a trigger.	“Create a trigger” on page 105 “Triggers” on page 106
Add comments about the quality of runbooks, after the runbook is completed.	“Improving runbook execution” on page 63
Run a draft version of a runbook.	“Runbook versions” on page 78
Load runbook examples.	“Load and reset example runbooks” on page 68

Tasks	Documentation
Preview and run a runbook.	“Run your first runbook” on page 67
Add and edit the runbook.	“Create a runbook” on page 69 “Edit a runbook” on page 70
Monitor all runbooks that are run by the current user.	Chapter 6, “Executions,” on page 81
Monitor all runbooks that are run by all users.	“Monitor runbook history” on page 81
Edit and create a trigger.	“Create a trigger” on page 105 “Triggers” on page 106
Add comments about the quality of runbooks, after the runbook is completed.	“Improving runbook execution” on page 63
Run a draft version of a runbook.	“Runbook versions” on page 78
Publish a runbook.	“Save and publish a runbook” on page 78

Tasks	Documentation
Import runbooks.	“Export and import runbooks” on page 64
Delete a runbook.	“Delete a runbook” on page 79
Delete a trigger.	“Triggers” on page 106
Create and edit a connection.	“Create a connection” on page 89
Delete a connection.	Chapter 8, “Connections,” on page 89
Preview, create, and delete API Keys.	“API Keys” on page 113

Using keyboard shortcuts

IBM Runbook Automation uses standard Microsoft Windows navigation keys.

The following tables provide additional information about keyboard shortcut in certain parts of the user interface.

Action	Keyboard shortcut
To move up and down the Library, Execution, Execution history, Triggers, Automations, Connections, and API Keys tables.	Press Tab until the table is selected. Then use the cursor key to select a runbook.
Select an entry in the table.	Select a runbook using the cursor keys. Move with the Tab key to the check box and press space.
Collapse and expand a runbook section in the Runbooks table.	Select a runbook using the cursor keys. Move with the Tab key to the arrow in front of the runbook name. Press Enter to open the section, press enter again to close the section.

<i>Table 6: Common user interface button shortcuts (continued)</i>	
Action	Keyboard shortcut
Start an action within a table row, for example preview, run, edit, delete, show history.	Select a runbook using the cursor keys. Press the Tab key until the action items are displayed. Move with the Tab key to the desired action. Press Enter.
Select an action button. For example, Cancel, Apply and Run, Save as Draft. Move with the Tab key to the target button.	Press Enter

There are additional keyboard shortcuts for the runbook editor. To work with the editor, open the **Library** page. You need to have the role RBA Author or RBA Approver to edit a runbook. Click **New runbook** or **Edit** to open the editor.

The following keyboard shortcuts are used to work with runbook elements.

<i>Table 7: Keyboard shortcut for runbook elements</i>	
Runbook element	Keyboard shortcut
Add step	Ctrl + 3
Add command	Ctrl + 4
Add GOTO	Ctrl + 5
Add automation	Ctrl + 7
Add parameter	Ctrl + 8

RBA Private Deployment considerations for GDPR readiness

This topic is intended to help you in your preparations for GDPR readiness. It provides information about features of IBM Runbook Automation Private Deployment that you can configure, and aspects of the product's use, that you should consider to help your organization with GDPR readiness. This information is not an exhaustive list, due to the many ways that clients can choose and configure features, and the large variety of ways that the product can be used in itself and with third-party applications and systems.

Clients are responsible for ensuring their own compliance with various laws and regulations, including the European Union General Data Protection Regulation. Clients are solely responsible for obtaining advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulations that may affect the clients' business and any actions the clients may need to take to comply with such laws and regulations.

The products, services, and other capabilities described herein are not suitable for all client situations and may have restricted availability. IBM does not provide legal, accounting, or auditing advice or represent or warrant that its services or products will ensure that clients are in compliance with any law or regulation.

GDPR

General Data Protection Regulation (GDPR) has been adopted by the European Union ("EU") and applies from May 25, 2018.

Why is GDPR important? GDPR establishes a stronger data protection regulatory framework for processing of personal data of individuals. GDPR brings:

- New and enhanced rights for individuals
- Widened definition of personal data
- New obligations for processors
- Potential for significant financial penalties for non-compliance

- Compulsory data breach notification

Read more about GDPR:

- EU GDPR Information Portal <https://www.eugdpr.org/>
- ibm.com/GDPR website <https://ibm.com/GDPR>

Product Configuration - Considerations for GDPR Readiness

The following sections provide considerations for configuring Runbook Automation to help your organization with GDPR readiness.

IBM Runbook Automation is a service to build and execute runbooks that can help IT staff to solve common operational problems. IBM Runbook Automation can automate procedures that do not require human interaction, thereby increasing the efficiency of IT operations processes.

The most common scenario is that Runbook Automation Private Deployment is installed together with an existing IBM Tivoli Netcool/OMNIBus system and integrates into the existing Dashboard Application Services Hub (DASH) which uses WebSphere Application Server for authentication. Runbook Automation Private Deployment does not provide its own authentication mechanism, therefore a DASH installation is always a prerequisite.

If events always correspond to the same runbook then a trigger can be created and linked to the event with the runbook. Triggers can run with manual and automated runbooks. IBM Tivoli Netcool/Impact is required to run the trigger service.

Once the runbook is either launched by a trigger or manually, scripts or automations are executed on a target system to solve a problem or perform an administrative task. Runbooks use secure ssh connections to communicate with the target systems. Execution history and success statistics are logged for audit purposes and analysis into a Couch DB system.

Data Life Cycle

What is the end-to-end process through which personal data go through when using our offering?

Runbook Automation processes the types of Personal Data listed below:

- Authentication Credentials (username only).
- Basic Personal Information (such as Contact Names and e-mail address to identify an owner of a runbook).
- Technically Identifiable Personal Information (such as device IDs, usage based identifiers, IP address, etc. - when linked to an individual).

This offering is not designed to process any special categories of personal data. Where the runbook has been used to add custom data, this should also be reviewed for potential inclusion of personal data.

The processing activities with regard to personal data within this offering include:

- Receipt of data from Data Subjects and/or third parties.
- Computer processing of data, including data transmission, data retrieval, data access, and network access to allow data transfer if required.
- Storage and associated deletion of data.

This offering may integrate with the following IBM offerings, which may process personal data content:

- IBM Tivoli Netcool/Impact
- IBM Tivoli Netcool/OMNIBus
- IBM WebSphere Application Server
- IBM BigFix

Runbook Automation Private Deployment (PD) clients can submit online comments/feedback/requests to contact IBM about Runbook Automation PD subjects in a variety of ways, primarily:

- Public comments area on pages of Runbook Automation documentation in the IBM Knowledge Center.
- Public comments in the Runbook Automation space of dWAnswers.

Typically, only the client name and email address are used, to enable personal replies for the subject of the contact, and the use of personal data conforms to the IBM Online Privacy Statement (<https://www.ibm.com/privacy/us/en/>).

Data Collection

This offering collects the following types of personal data:

Runbook Automation Private Deployment does not collect any personal data, it just processes user ID, name and e-mail addresses that are passed on from WebSphere Application Server/DASH.

Data Storage

Storage of account data:

Runbook Automation relies on WebSphere Application Server/DASH for the management of user names and passwords. Runbook Automation does not provide its own management of user names and passwords.

For details on how to configure the WebSphere Application Server/DASH integration, see [“Planning for RBA DASH integration”](#) on page 21.

Storage of client Data:

The primary data processed by Runbook Automation relates to runbook content (for example scripts) which depending on the deployment environment could be data that belongs to you or originates from your end clients environment. Runbook Automation provides group based access controls. For more information, see: [“Create users and groups in the external user repository with read-only access”](#) on page 21.

Storage in backups:

All Runbook Automation data is stored in Couch DB. For more information, see [“Planning for Backup/Restore”](#) on page 24.

Storage in archives:

Historical data might be archived using a database backup. Runbook Automation does not offer an archival function to archive data for a specific period of time.

Data Access

Runbook Automation provides a number of groups and roles for controlling access to runbooks. The groups and roles enable differentiation between normal users and those with extra privileges. For more information, see [“Create users and groups in the external user repository with read-only access”](#) on page 21.

Logging is maintained for diagnostic and support purposes. For details about the default logs and how to configure them, see: [“Troubleshooting”](#) on page 49.

Data Processing

Encryption in motion:

Runbook Automation can be configured to use SSL communications for data transferred between components. For more information, see [“Configuring certificate validation”](#) on page 36.

Encryption at rest:

By default, the Runbook Automation service database data is not encrypted. If it is required that data in the database is encrypted, it is recommended to use Linux Unified Key Setup (LUKS) disk encryption to encrypt the file system that stores the data. For more information, see [“Planning for Database Encryption” on page 25](#).

Data Deletion

Removal of users from WebSphere Application Server/DASH will prevent the user from logging into Runbook Automation. It will not remove the users data (for example, name) from active or historical runbooks as there is an ongoing need from an operational/audit perspective to maintain this data. However, as part of your deployment you should review the period for which data is archived, backups are stored, and logs are maintained, to determine if they are reasonable based on your operational needs.

Account Data deletion:

Where Runbook Automation is being used in a managed service environment, and a single deployment is being used to manage runbooks from multiple end customers (tenants), consideration should be given to the processes for onboarding and offboarding and what mechanisms need to be in place to remove a tenants data, for example use of distinct databases per tenant.

Data Monitoring

Personal data in Runbook Automation is limited to Basic Personal Information (for example usernames for authentication) and Technical Personal Information (for example IP addresses/hostnames from systems used by the user to access the solution and potentially captured in debug/trace logs). Runbook executions are logged in the runbook history. Runbook Automation is enabled to audit access to the database which stores all runbook activity.

Responding to Data Subject Rights

The Personal Data stored and processed by Runbook Automation falls under the categories of Basic Personal Data (for example usernames used for authentication and Name/ID to show ownership of an runbook) and Technically Identifiable Personal Information (such as IP addresses and hostnames to which user activity could potentially be linked). This data is intrinsic to the operation of an effective runbook automation workflow. Removal of data, modification of historical data and sharing of this data is likely to be counter to your enterprises policies.

However, consideration may need to be given to the following:

- Data is only retained for a reasonable period based on operational, compliance and industry audit requirements that pertain.
- Data is secured appropriately when in archive format.
- When Runbook Automation is used in a managed services context with multiple tenants in a single instance, having mechanisms in place to clearly identify the data belonging to a specific tenant.
- When Runbook Automation is used for managing your enterprises own IT/network environment and the users of the solution are employees/contractually engaged staff, that the contract terms are GDPR compatible.
- When the runbooks have been augmented with additional data sourced from other data sources available in your environment, whether these customizations add Personal Data and what implications there are on doing this from a GDPR compliance perspective.

Chapter 3. Installing, configuring, and troubleshooting a Private Deployment

Private Deployment

The topics under this section are relevant for administrators who are installing and configuring a Private Deployment of IBM Runbook Automation.

IBM Runbook Automation Private Deployment allows you to install and use IBM Runbook Automation in a network environment that is independent of the Internet.

Overview

Private Deployment

IBM Runbook Automation Private Deployment can be installed into an existing Netcool environment. You can work with runbooks and automations using the existing Netcool interfaces and means of authentication.

The following figure shows a typical server topology when using IBM Runbook Automation.

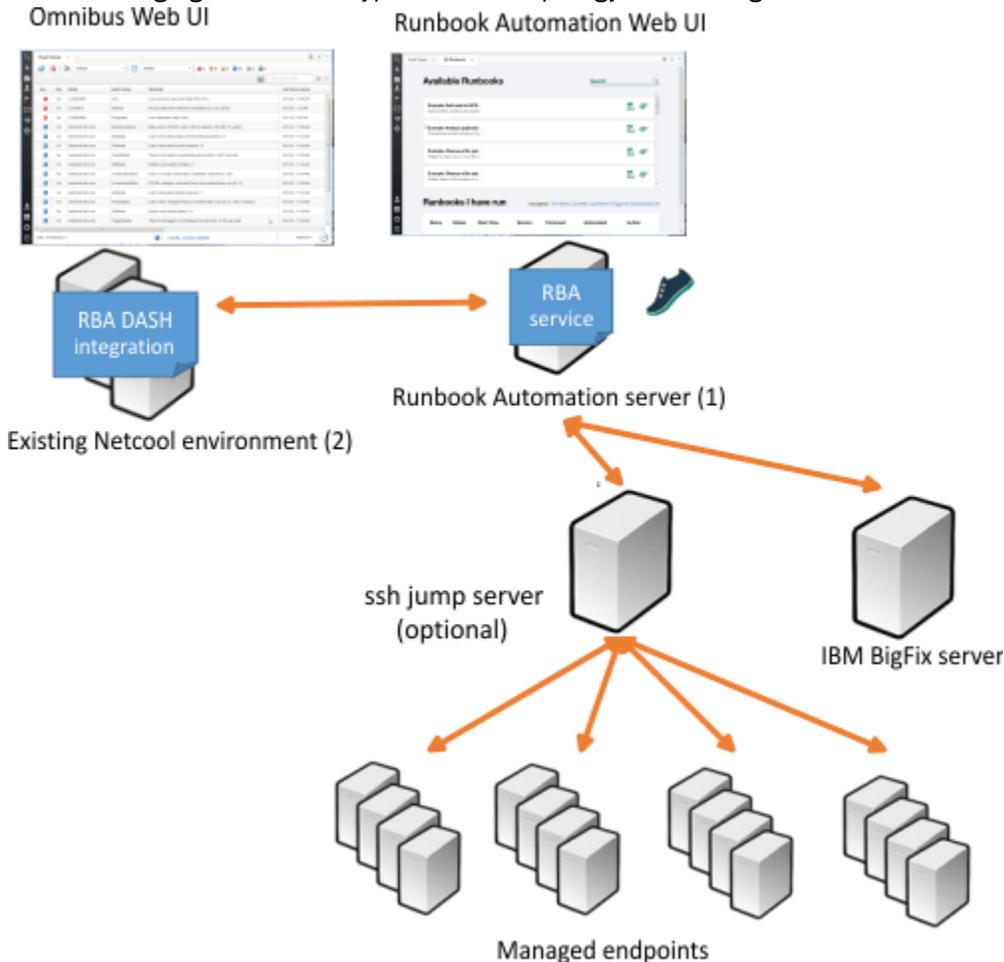


Figure 2: Server topology for IBM Runbook Automation

IBM Runbook Automation Private Deployment contains two components that must be installed. The two components are installed on the servers marked with (1) and (2) in Figure 1:

(1) Runbook Automation server (RBA server) - The RBA server hosts the RBA service which consists of several micro services that are required to run IBM Runbook Automation. The RBA service is delivered as an rpm package and is installed using yum. While it is recommended to have a dedicated server for the RBA service, this is not a requirement and any system in the existing Netcool environment which meets the prerequisites can be used.

(2) Existing Netcool Environment – The existing Netcool Environment consists of, for example, a data base server, Netcool/OMNIbus and Netcool/Impact core components, and a Web GUI component for Netcool/OMNIbus and Netcool/Impact. In order to seamlessly integrate into the Netcool/OMNIbus Web GUI, IBM Runbook Automation Private Deployment provides a plugin called **RBA DASH integration** that you can install into the same WAS/DASH server that also hosts the Netcool/OMNIbus Web GUI.

You can install one or multiple instances of the Netcool/OMNIbus Web GUI server. If you install multiple instances, each of the instances can have a different purpose. The following installation topologies are supported:

- In a setup with one Netcool/OMNIbus Web GUI server instance, you use the event console and the RBA DASH integration in the same environment.
- In a setup with multiple Netcool/OMNIbus Web GUI server instances, you can have one instance of the Netcool/OMNIbus Web GUI server that you are using as the event console, and another instance of the Netcool/OMNIbus Web GUI server where you install the RBA DASH integration. For more information, see [“Configuring a setup with multiple Netcool/OMNIbus Web GUI servers”](#) on page 32.

Component Name	Description	Installation Location
rba-<version>-<timestamp>.x86_64.rpm For example: rba-1.4.0-1606071446.x86_64.rpm	RBA service	(1) Recommended: Dedicated server. Can also be installed on any existing Netcool Operations Insight server if it meets the system prerequisites.
rba-im-<version>-repository.zip	RBA DASH integration: Installation Manager Package which contains the WAS/DASH extensions for IBM Runbook Automation.	(2) On the system where the Netcool/OMNIbus Web GUI is installed on a WebSphere Application Server.

Planning for RBA installation

Private Deployment

Before installing the product, read about the system and software requirements for IBM Runbook Automation.

System prerequisites

Private Deployment

The IBM Runbook Automation service has the following system prerequisites.

- Architecture: x86_64
- CPU: minimum 1 core
- Memory: minimum 4GB
- Free disk space:

File system	Description	Required free space
/opt/ibm/arb	Product code	5GB
/var/lib/docker	Used by docker to store docker images and containers.	Minimum 150GB recommended 300GB
/tmp	Temporary directory for downloading the rpm package.	5GB
/var/log/ibm/arb	RBA logs	20GB
/var/lib/ibm/arb	RBA data	Minimum 50GB recommended 100GB This file system grows. The free space requirements depend on the data stored in runbooks and the frequency of runbook execution.

Software prerequisites

Private Deployment

The IBM Runbook Automation service has the following software prerequisites.

- Red Hat Enterprise Linux (RHEL) Server 7.2 to 7.4.
- Red Hat Satellite Servers must be configured on the system.
 - The Docker version included with each RHEL release is the required Docker version. For example, for RHEL 7.2 the docker version is 1.9.1. For RHEL 7.4 the docker version is 1.12.6.
 - To install Docker, you must subscribe to the Red Hat Satellite Server Channel **Server Extras**.
- The following packages are marked as dependencies in the rpm package for the RBA service. They are prerequisites and will install automatically from Red Hat Satellite Servers when the rpm package for the RBA service is installed.
 - docker 1.9.1 (for RHEL 7.2)
 - lvm2
 - openssl 1.0 (or later)
 - bash v4 (or later)
 - curl
- Security-Enhanced Linux (SELinux) must be disabled or run in permissive mode.

Docker considerations

Private Deployment

If Docker is not already installed, it will be installed as a dependency when the RBA service is installed.

IBM Runbook Automation Private Deployment uses docker for running the RBA service. In order to install the dependencies correctly, the Red Hat Satellite server must be configured on the RBA server system and the channel **Server Extras** must be subscribed to. This channel contains the required docker packages.

Adding RedHat Satellite Server Extras channel

Private Deployment

It is a prerequisite to add the subscription to the **Server Extras** channel (rhel-7-server-extras-rpms).

About this task

The commands for adding the subscription to the Server Extras channel depend on the version of Red Hat Enterprise Linux (RHEL) Server you have installed. For the commands that apply to your installed version of RHEL, see the RedHat documentation at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/.

Firewall considerations

Private Deployment

It is recommended to disable the firewall on the RBA server.

In the current version of Docker in RHEL7.2, the docker daemon in combination with the firewall daemon might be unreliable. Instead, configure the firewall in routers and switches as described in [“Firewall requirements”](#) on page 22.

```
systemctl stop firewalld
systemctl disable firewalld
```

If the firewall is enabled, the Docker interface must be added to the trusted zone. For more information, see [“Firewall requirements”](#) on page 22.

Note: If the firewall is reconfigured, you must restart the docker service using the command `systemctl restart docker`.

Docker storage driver

Private Deployment

Docker has a pluggable storage driver architecture. By default, Red Hat uses Device Mapper as a storage driver using a loopback device. This default setting can be used for test purposes. However, it is strongly discouraged by Red Hat to use this default for production environments. For production systems, the Red Hat recommendation is to use direct LVM devices instead of loopback devices.

About this task

Complete the following steps to configure Docker to use direct LVM devices. These steps must be completed before the IBM Runbook Automation installation.

Procedure

1. The LVM2 package is a prerequisite and must be installed.
2. Create an LVM device:

```
pvcreeate <physical device name>
vgcreate vg_docker <physical device name>
```

For example:

```
pvcreeate /dev/sdd
vgcreate vg_docker /dev/sdd
```

3. Add the following lines to `/etc/sysconfig/docker-storage-setup`:

```
SETUP_LVM_THIN_POOL=yes
VG="vg_docker"
DATA_SIZE=10G
```

4. Stop the docker service:

```
systemctl stop docker
```

5. Remove `/var/lib/docker`. **Note:** this will remove all docker containers and images. For example, if you already installed IBM Runbook Automation and do the reconfiguration after, IBM Runbook Automation will need to be reinstalled.

6. Setup docker storage using the RedHat tool **docker-storage-setup**:

```
docker-storage-setup
```

7. Start the docker service:

```
systemctl start docker
```

What to do next

To check if the changes are effective, use the command `docker info`. Ensure that no loop devices are used for Data file and Metadata file.

For more information, see the Docker documentation:

<https://docs.docker.com/engine/userguide/storagedriver/selectadriver/>

<https://docs.docker.com/engine/userguide/storagedriver/device-mapper-driver/>

Planning for RBA DASH integration

Private Deployment

The IBM Runbook Automation and Dashboard Application Services Hub (DASH) integration is installed into the existing Netcool environment on the WebSphere Application server that hosts the Netcool/OMNIBus Web GUI.

This section contains information about the system, software, and user group requirements for RBA and DASH integration.

System prerequisites

Private Deployment

The IBM Runbook Automation DASH integration has the following system prerequisites.

- Free disk space:

File system	Description	Required free space
/tmp	Temporary directory for downloading the rpm package.	1GB
JazzSM profile directory	Product code.	1GB

- The operating system limit for open files must be at least 4096. You can check this by using the command `ulimit -n`.

Software prerequisites

Private Deployment

The IBM Runbook Automation DASH integration has the following software prerequisites.

- Operating System: Linux.
- WebSphere Application Server V8.5.5.4 or later.
- Jazz for Service Management V1.1.2.1 or later (which includes DASH V3.1.2.1).
- DASH V3.1.2.1 cumulative patch 6. Patches can be downloaded from [IBM Fix Central](#).

Create users and groups in the external user repository with read-only access

Private Deployment

You can skip this section if the WebSphere Application Server where the RBA DASH integration is installed allows for the active management of users and groups using the WebSphere Application Server

administrative interfaces. In this case, the user groups that are required by the RBA DASH integration are created during installation.

If the WebSphere Application Server is set up with read-only access to an external user repository, then you must create several user groups within the external user repository before you start installing the RBA DASH integration.

Use the user interface of the external user repository to create the default groups listed in [Table 11](#) on [page 22](#). You can choose to use non-default groups. Non-default group names must be mapped later during installation of the RBA DASH integration.

Default Group Name	Description
ARBUserGroup	IBM Runbook Automation users with the authority to perform operational activities.
ARBAuthorGroup	IBM Runbook Automation users with the additional authority to create and manage runbook drafts, triggers, and automations.
ARBApproverGroup	IBM Runbook Automation users with the additional authority to approve runbooks.
ARBManagerGroup	IBM Runbook Automation users with the additional authority to Connections and API Keys.

For more information about the level of authority associated with these groups, see [“User groups and roles”](#) on [page 45](#).

After you have created the groups you should also assign users to the appropriate groups. It is recommended to have at least one member in the group with the default name "ARBManagerGroup".

Planning for IBM BigFix

You can optionally use automations of type BigFix.

Prerequisites

You must use a supported BigFix version 9.2 or later. It is recommended to apply the latest fix pack.

See the IBM BigFix Knowledge Center for planning and installation instructions: <https://www.ibm.com/support/knowledgecenter/SSQL82>.

The BigFix REST API must be enabled. For information about prerequisites and how to enable the BigFix REST API, see <https://developer.bigfix.com/rest-api/prerequisites.html>. The BigFix REST API must be set up to use HTTPS.

Firewall requirements

Private Deployment

Firewall rules must be configured for the firewalls on the systems and routers that communicate with the RBA server.

The following figure shows the possible connections between the different systems and their default ports.

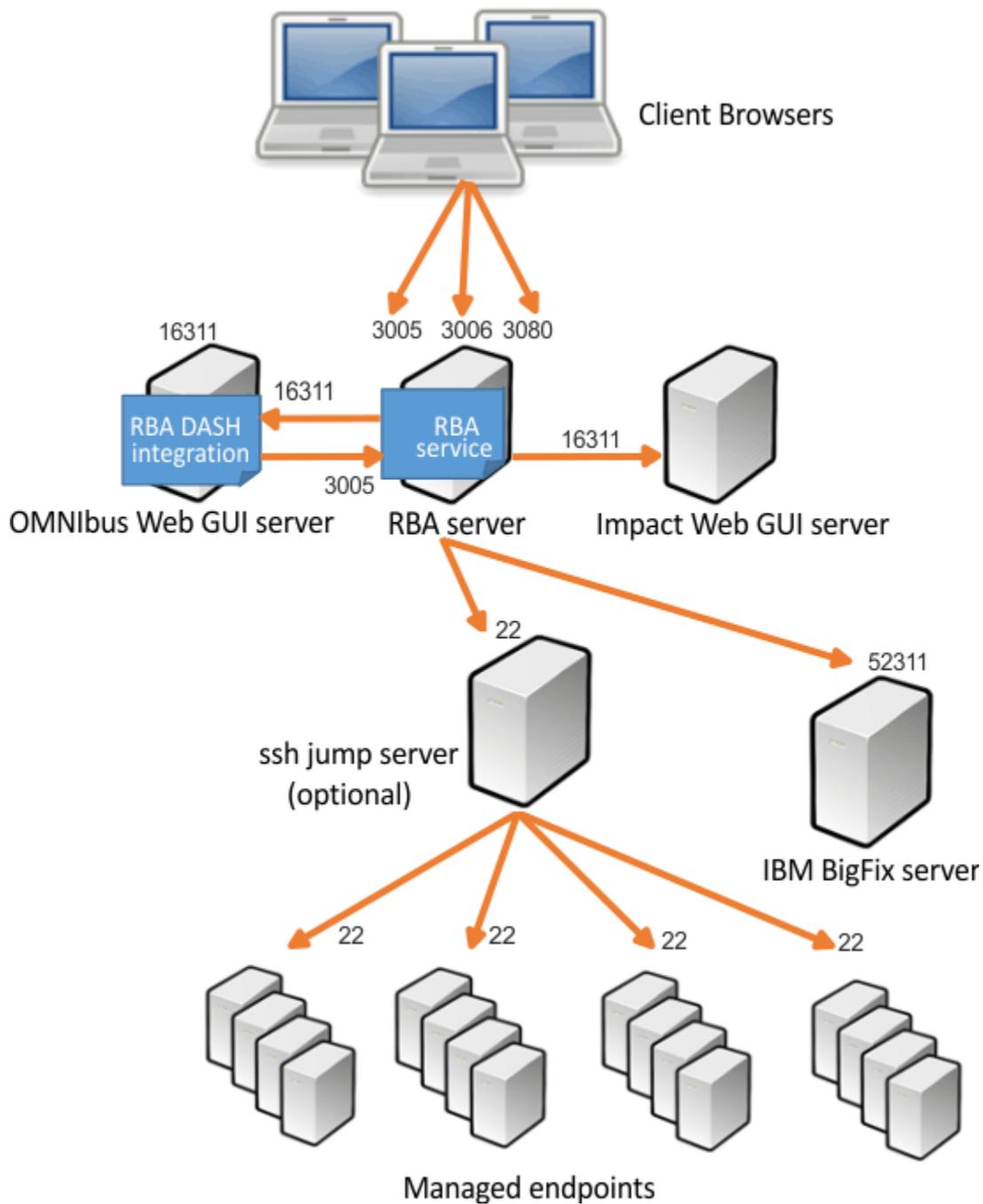


Figure 3: Firewall considerations

Table 12: Firewall requirements				
Client System	Server System	Service	Default Port on Server	Protocol
Client browser	RBA server	Runbook service	3005	TCP
Client browser	RBA server	Trigger service	3006	TCP
Client browser	RBA server	Automation service	3080	TCP
Client browser	OMNibus Web GUI	DASH	16311	TCP

Table 12: Firewall requirements (continued)

Client System	Server System	Service	Default Port on Server	Protocol
OMNIbus Web GUI	RBA server	Runbook service	3005	TCP
RBA server	OMNIbus Web GUI	DASH	16311	TCP
RBA server	Impact Web GUI	DASH	16311	TCP
RBA server	ssh jump Server	sshd	22	TCP
RBA server	IBM BigFix Server	IBM BigFix	52311	TCP

Note: The ports shown in the table are default ports. If you reconfigured the ports, they need to be adapted accordingly.

It is recommended to disable the firewall on the RBA server. However, if the firewall is enabled on the RBA server, additional firewall configuration for docker is required. The docker0 interface must be added to the trusted zone, otherwise the docker containers cannot communicate with each other. The RBA service code contains a tool `rbaAddFirewallRules.sh` that configures the required firewall rules automatically.

Planning for single sign-on

Private Deployment

IBM Runbook Automation Private Deployment uses WebSphere Application Server for authentication and authorization.

Single sign-on has the following requirements:

- The RBA server and the WebSphere Application Server/DASH server hosting the Netcool/OMNIbus Web GUI must be in the same network domain.
- The WebSphere Application Server hosting the Netcool/OMNIbus Web GUI must be configured for single sign-on (SSO).

For more information, see [“Pre-installation configuration for WebSphere Application Server single sign-on \(SSO\)”](#) on page 37.

Planning for Backup/Restore

Private Deployment

Create or restore a backup of the RBA service data.

Before you begin

The RBA service data is stored in a database located in the directory `/var/lib/ibm/arb/couchdb/` data. The configuration for the database is stored in the directory `/etc/ibm/arb/couchdb`.

About this task

The RBA service must be stopped before a backup of the database data is created to ensure that the data is consistent. In addition to creating manual backups, it is recommended to replicate the data using a software-mirror or hardware-mirror.

Procedure

- **Perform the following steps to create a backup of the RBA service data:**
 - a) Login to the RBA server.

b) Stop the RBA service:

```
rba stop
```

c) Make a backup copy of the RBA service data:

```
cp -ar /var/lib/ibm/arb/* <backup dir>/data  
cp -ar /etc/ibm/arb/* <backup dir>/etc
```

d) Start the RBA service:

```
rba start
```

- **Perform the following steps to restore a backup from RBA service data:**

a) Login to the RBA server.

b) Stop the RBA service:

```
rba stop
```

c) Delete the old data:

```
rm -rf /var/lib/ibm/arb/  
rm -rf /etc/ibm/arb/
```

d) Recreate the deleted directories:

```
mkdir -p /var/lib/ibm/arb/  
mkdir -p /etc/ibm/arb/
```

e) Restore the backup copy of the RBA service data:

```
cp -ar <backup dir>/data/* /var/lib/ibm/arb/  
cp -ar <backup dir>/etc/* /etc/ibm/arb/
```

f) Start the RBA service:

```
rba start
```

Planning for Database Encryption

By default, the Runbook Automation service database data is not encrypted. If it is required that data in the database is encrypted, it is recommended to use Linux Unified Key Setup (LUKS) disk encryption to encrypt the file system that stores the data.

Procedure

1. Create a file system that is encrypted using LUKS. For more information, go to https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Security_Guide/sec-Encryption.html.
2. Mount the encrypted file system to `/var/lib/ibm/arb/couchdb`.

Installing

Private Deployment

This section outlines the recommended installation sequence and where to find more information.

Table 13: RBA installation sequence

Step	More information
1. Extract the package components.	The deliverable contains the following components: <ul style="list-style-type: none"> • rpm package for RBA service • Installation Manager Repository for RBA DASH integration
2. Ensure that Websphere Application Server Single Sign On is configured.	“Pre-installation configuration for WebSphere Application Server single sign-on (SSO)” on page 37
3. Install the RBA service	“Installing the RBA service” on page 26
4. Install RBA DASH integration	“Installing the RBA DASH integration” on page 27
5. Perform post installation tasks	“Post-installation steps for the RBA DASH integration” on page 27
6. Install Netcool/Impact (optional)	“Installing Netcool/Impact to run the trigger service” on page 94
7. Configure Firewalls (optional)	“Firewall requirements” on page 22
8. Configure SSL Communication (optional)	“Enable secure communications within the RBA service” on page 44

Installing the RBA service

Private Deployment

Use these steps to set up and configure your environment.

Procedure

1. Ensure that the RHEL Satellite Server is set up correctly and the channel **Server Extras** is subscribed to. For more information, see [“Adding RedHat Satellite Server Extras channel” on page 19](#).
2. Ensure that your system has latest RHEL updates.
 - a) Update the RHEL packages:

```
yum update
```

- b) Reboot to make updates effective.
3. Upload the rpm package to the RBA server to a temporary directory, for example `/tmp/rba`.
 4. Change to the following directory:

```
cd /tmp/rba
```

5. Install the rpm package:

```
yum localinstall rba-x.x.x-.....x86_64.rpm
```

6. Ensure that SELinux is turned off or permissive mode is enabled:

```
sestatus
SELinux status: disabled
```

7. Configure the firewall:

- a) Turn off the firewall:

```
service firewalld stop
service firewalld disable
```

- b) Or add the required firewall rules:

- 1) Review the file `/opt/ibm/arb/bin/rbaAddFirewallRules.sh`.
- 2) Execute the script `/opt/ibm/arb/bin/rbaAddFirewallRules.sh` or define the firewall rules manually.

8. Start the RBA service:

```
rba start
```

Installing the RBA DASH integration

Private Deployment

Complete these steps to install the RBA DASH integration.

Procedure

1. Upload the IBM Installation Manager repository for RBA DASH integration to the WAS Server where Netcool/OMNIBus Web GUI is hosted.
2. Start the IBM Installation Manager, for example: `/opt/IBM/InstallationManager/eclipse/IBMIM`.
3. Add the repository to the Installation Manager. Click **File > Preferences > Repositories > Add Repository** and select the uploaded repository zip file.
4. On the Installation Manager home screen, click **Install**.
5. Select the **RBA DASH integration** package and click **Next**.
6. Accept the license agreement and follow the on-screen instructions to complete the installation.

Post-installation steps for the RBA DASH integration

After installing RBA DASH integration, you must perform a number of post-installation tasks

Retrieve the RBA server certificate into the WAS trust store

IBM Runbook Automation private deployment uses the **DASH console integration** component in the WebSphere Application Server to integrate the Runbook Automation specific menu items and the related pages into the DASH console. The console integration component needs to communicate with the RBA server using the HTTPS protocol. Therefore you need to import the certificate from the RBA server into the trust store of the WebSphere Application Server.

Complete the following steps to retrieve the RBA server certificate and store it in the WebSphere Application Server trust store:

1. Make sure the RBA service has been installed and started.
2. Login to the WAS administrative console as WAS administrator.
3. Navigate to **Security > SSL certificate and key management**.
4. Click **Key stores and certificates**.
5. Click **NodeDefaultTrustStore**.
6. Click **Signer certificates**.
7. Click **Retrieve from port**.
8. On the Retrieve from port panel, enter the following data:
 - a. **Field Host:** Enter the fully qualified domain name of the RBA server, for example: `myrbaserver.mycompany.com`.
 - b. **Field Port:** Enter the port number of the RBS service (3005).

- c. **Field SSL configuration for outbound connection:** Use the default value `NodeDefaultSSLSettings`.
 - d. **Field Alias:** Enter `rba.server.certificate`.
9. Click **Retrieve signer information**.
 10. Review the data that is displayed in the Retrieved signer information area.
 11. After you have successfully validated the signer information, click **Apply**.
 12. Save the WAS configuration.

Retrieve the RBA server certificate in the browser

If the self-signed certificate that is created during the first start of the RBA server is not replaced with a certificate that is issued by a trusted certificate authority, each user needs to import the RBA server self-signed certificate from the following URL prior to accessing the Runbook Automation user interface:

```
https://<rbaServerFQDN>:3005
```

where the `rbaServerFQDN` is the fully qualified domain name of the RBA server.

Getting started

Private Deployment

The IBM Runbook Automation Web GUI is accessed using the WebSphere Application Server/DASH interface of your existing Netcool/OMNIbus Web GUI server.

About this task

Procedure

- **Logging in**
 - a) Enter the DASH login URL in your browser. Include the fully qualified domain name of the WebSphere Application Server, for example: `https://omnigui.mylab.mycompany.com:16311/ibm/console/logon.jsp`.
 - b) Enter the user ID and password for Runbook Automation.

- **Selecting a task for IBM Runbook Automation**

- a) Click  **Console Integration**.
- b) Select an RBA task from the list of displayed options.

- **Checking the About page**

If you are using IBM Runbook Automation for the first time, you can check basic communication functions on the **About** page. On this page you can confirm that all required SSL certificates are being used.

- a) Click  **Console Integration > About**.
- b) Ensure that all services have a status of **Checked**.
- c) If you have not accepted the SSL certificate for IBM Runbook Automation services in the client browser, you might get an error message saying the connection is not secure. To resolve this issue, see [“Retrieve the RBA server certificate in the browser” on page 28](#).
- d) If you only accepted part of the SSL certificates for IBM Runbook Automation services in the client browser, you might see a failed status for some services on the **About** page. To resolve this issue, see [“Retrieve the RBA server certificate in the browser” on page 28](#).

- [“Load and reset example runbooks” on page 68](#)

Updating the installation

Private Deployment

Complete the steps in this section to update the Runbook Automation service.

Updating the RBA service

Private Deployment

Complete these steps to update the RBA service.

Before you begin

Before updating the RBA service you should backup the RBA service data. For more information, see [“Planning for Backup/Restore” on page 24](#).

Procedure

1. Upload the rpm package to the RBA server to a temporary directory, for example /tmp/rba.
2. Change to the following directory:

```
cd /tmp/rba
```

3. Install the rpm package:

```
yum localinstall rba-x.x.x-.....x86_64.rpm
```

4. Restart the RBA service:

```
rba restart
```

Updating the RBA DASH integration

Private Deployment

Complete these steps to update the RBA DASH integration.

Procedure

1. Upload the IBM Installation Manager repository for RBA DASH integration to the WAS Server where Netcool/OMNIBus Web GUI is hosted.
2. Start the IBM Installation Manager, for example: /opt/IBM/InstallationManager/eclipse/IBMIM.
3. Add the repository to the Installation Manager. Click **File > Preferences > Repositories > Add Repository** and select the uploaded repository zip file.
4. On the Installation Manager home screen, click **Update**.
5. Select the **RBA DASH integration** package and click **Next**.
6. Accept the license agreement and follow the on-screen instructions to complete the installation.

Note: The RBA service is delivered as docker images. When updating the RBA service, old images are not removed to facilitate roll back to older version using the old images. For more information about removing old images, see “Cleaning up old docker images”

Configuring

Private Deployment

Use the information in this section to help you understand how to configure the RBA service, RBA DASH integration, and the Netcool/OMNIbus event tool for launch-in-context.

For information about configuring automations or configuring Netcool/OMNIbus and Netcool/Impact, see the related links at the end of this page.

Configuring the RBA service

Private Deployment

The configuration of the Runbook Automation service is done via the configuration file `/etc/ibm/arb/rba.config`.

An example config file is available in `/opt/ibm/arb/etc/rba.config.template`. The following table contains the Runbook Automation service configuration parameters:

<i>Table 14: Configuration parameters</i>		
Parameter name	Description	Example value
<code>couchdb_user</code>	Required parameter: RBA Database user name.	<code>dbuser</code>
<code>couchdb_pw</code>	Optional parameter: RBA Database password for couchdb user. If this parameter is not specified, it will be asked interactively when starting the server.	<code>Rbasecret</code>
<code>dash_host</code>	Required parameter: Fully qualified domain name of WAS/DASH server which hosts the OMNIbus Web GUI.	<code>DASH.mycompany.com</code>
<code>dash_port</code>	Required parameter: Port of DASH which hosts the OMNIbus Web GUI.	<code>16311</code>
<code>secure_mode</code>	Required parameter: Enforces SSL authentication between RBA service and DASH, Impact, Automation Agents if set to true.	<code>true / false</code>
<code>local_connection_domain</code>	Fully qualified domain name of the RBA server that is used to create a certificate.	<code>\$(hostname -f)</code>
<code>docker_run_options_rbs</code>	Optional parameter: Additional parameters that are passed to the docker run command when starting the container for the runbook service. For example, it can be used to add entries in <code>/etc/host</code> of the docker container in test environments where no DNS is set up. Note: Only use this parameter when advised to do so by IBM support.	

Table 14: Configuration parameters (continued)

Parameter name	Description	Example value
docker_run_options_as	Optional parameter: Additional parameters that are passed to the docker run command when starting the container for the automation service. For example, it can be used to add entries in /etc/host of the docker container in test environments where no DNS is set up. Note: Only use this parameter when advised to do so by IBM support.	
docker_run_options_ts	Optional parameter: Additional parameters that are passed to the docker run command when starting the container for the trigger service. For example, it can be used to add entries in /etc/host of the docker container in test environments where no DNS is set up. Note: Only use this parameter when advised to do so by IBM support.	
docker_run_options_couchdb	Optional parameter: Additional parameters that are passed to the docker run command when starting the container for the database service. For example, it can be used to add entries in /etc/host of the docker container in test environments where no DNS is set up. Note: Only use this parameter when advised to do so by IBM support.	

If you start the server without a configuration file, the required parameters are queried interactively and the corresponding config file is created.

You must restart the RBA service after the configuration file has been edited in order to effect the changes. Use the command `iba restart` to restart the service.

Configuring the RBA DASH integration

Private Deployment

The RBA DASH integration is completely configured after the installation (including post-installation) has been completed.

If you choose to install the RBA service on a system other than the one you used during the initial installation of the RBA DASH integration, you must import the new self-signed certificate and reconfigure the RBA server host name in the DASH console integration definition.

Importing the new self-signed RBA server certificate

Follow the steps outlined in [“Post-installation steps for the RBA DASH integration”](#) on page 27.

Modifying the DASH console integration definition

1. Login to the DASH console as with a WebSphere Application Server user id that is in the “iscadmins” role. Note that after installation, the ARBManager group is associated with the “iscadmins” role by default, so all members of the ARBManager group inherit the “iscadmins” role.
2. Navigate to **Console Settings > Console Integrations**.
3. In the table of console integrations, click **Runbook Automation**.
4. Edit the Console Integration URL and replace the previous RBA server FQDN with the new RBA server FQDN.

Configuring a setup with multiple Netcool/OMNIBus Web GUI servers

Private Deployment

If you have one instance of the Netcool/OMNIBus Web GUI server that you are using as the event console and another instance of the Netcool/OMNIBus Web GUI server where you installed the RBA DASH integration, you must ensure that you can be logged into both Netcool/OMNIBus Web GUI servers at the same time.

The related configuration updates are described in the sections that follow.

Configure WebSphere LTPA tokens for multiple servers

User can only be logged into both of the Netcool/OMNIBus WebSphere (WAS) servers simultaneously if one of the following conditions is met:

- Both of these WAS servers participate in a WAS single sign-on setup, so the LTPA token created when logging in to one of the WAS servers can also be validated in the other WAS server.
- The two WAS servers issue LTPA tokens with different names (“LTPA V2 cookie name”), so the user's browser can maintain two separate LTPA tokens with different names, based on two separate login actions.

For more information on LTPA tokens, see [“Pre-installation configuration for WebSphere Application Server single sign-on \(SSO\)”](#) on page 37.

Configure a special launch-in-context tool

The event tool for launch-in-context from Netcool/OMNIBus events to the associated runbooks must be configured as described in [“Configuring an IBM Netcool/OMNIBus event tool for launch-in-context to IBM Runbook Automation”](#) on page 32 (see *Create a launch-in-context tool configuration for a setup with two Netcool/OMNIBus installations*).

Allow pop-ups in the browser

To open a new browser tab when clicking the xLaunch tool, the browser must be set up to allow pop-ups. If pop-ups are not generally allowed, they must at least be allowed from the URL `https://<OMNIBus_FQDN>:<OMNIBus_port>`, where `<OMNIBus_FQDN>` is the fully qualified domain name of the Netcool/OMNIBus Web GUI server that is used for displaying the event viewer, and `<OMNIBus_port>` is the related DASH port number.

Configuring an IBM Netcool/OMNIBus event tool for launch-in-context to IBM Runbook Automation

Private Deployment

To start manual or semi-automated runbooks from events that are listed in the Netcool/OMNIBus Event Viewer, you must enable the launch-in-context menu.

To use the launch-in-context tool, you must add it to the list of available Netcool/OMNIBus tools using the Tool Creation option within Netcool/OMNIBus Web GUI. You must then add the tool to the list of tools that appear on the pop-up menu in the Netcool/OMNIBus Event List using the Menu Configuration option within Netcool/OMNIBus Web GUI.

Launch-in-context is only available if you create a matching trigger. For more information about how to create a trigger, see [“Create a trigger” on page 105](#).

You must define different OMNIBus event tools depending on the installation topology, that is if you have a single Netcool/OMNIBus installation with both the event viewer and the RBA DASH integration installed, or two separate Netcool/OMNIBus installations. The following table outlines the differences between the tools. The sections below the table guide you through the tool creation.

Installation topology	RBA DASH integration installed into the same Netcool/OMNIBus Web GUI server that is used as the OMNIBus event console	OMNIBus event console and RBA DASH integration installed into different Netcool/OMNIBus Web GUI servers
Tool type	Script	CGI/URL
Browser tabs	One single browser-level tab for DASH, containing separate DASH-level tabs for the OMNIBus Event Viewer and for RBA.	Two browser-level tabs for two separate DASH instances, with the following characteristics: <ul style="list-style-type: none"> - One tab for the OMNIBus Event Viewer. - When you launch RBA from an event for the first time, a second DASH tab is added at the browser level displaying the login for DASH where the RBA DASH integration is installed. After login, you go directly to the Run runbook page. - When you launch RBA again from the DASH tab with the OMNIBus Event Viewer, the related runbook opens in the second RBA DASH tab.
Session life time	One single session timeout for both OMNIBus and RBA tabs. You might be prompted to re-authenticate for the single session. If re-authentication is not done, unfinished or unsaved work might be lost.	You might get prompted to re-authenticate for both DASH tabs independently. If re-authentication is not done, unfinished or unsaved work might be lost.

For information about supported characters in runbook parameters, see [Supported characters for launch-in-context](#).

Create a launch-in-context tool configuration for a single Netcool/OMNIBus installation

To create a launch-in-context tool configuration, use the following steps:

1. Log on to Netcool/OMNIBus Web GUI as an administrator.

2. Navigate to **Administration > Event Management Tools > Tool Configuration**
3. Click **Create Tool**.
4. Set the **Name** field to LaunchRunbookPD.
5. From the **Type** drop-down field, select **Script**.
6. Make sure the data source, for example Netcool/OMNIbus, is selected in the data source configuration.
7. In the **Script Commands** field, enter one of the following JavaScript snippets:

Up to Netcool/Impact 7.1.0.12:

```
var runbookId = '${selected_rows.RunbookID}';

if (runbookId == '' || runbookId == ' ' || runbookId == ',' || runbookId == '0' ||
runbookId == 0) {
    alert("This event is not linked to a runbook");
}
else {
    var launchEvent = {
        name: 'http://ibm.com/isclite#launchPage',
        NavigationNode: 'ARBTaskLIC',
        switchPage: true,
        payload: {
            rbid: runbookId,
            bulk_params: '${selected_rows.RunbookParameters}'
        }
    };
    ${param.portletNamespace}sendPortletEvent(launchEvent);
}
```

Starting with Netcool/Impact 7.1.0.13:

```
var runbookId = '${selected_rows.RunbookID}';

if (runbookId == '' || runbookId == ' ' || runbookId == ',' || runbookId == '0' ||
runbookId == 0) {
    alert("This event is not linked to a runbook");
}
else {
    var launchEvent = {
        name: 'http://ibm.com/isclite#launchPage',
        NavigationNode: 'ARBTaskLIC',
        switchPage: true,
        payload: {
            rbid: runbookId,
            bulk_params: '${selected_rows.RunbookParametersB64}'
        }
    };
    ${param.portletNamespace}sendPortletEvent(launchEvent);
}
```

Note:

- Do not add any comments to this script. All lines after a comment line will be ignored.
 - You can modify the alert message text "This event is not linked to a runbook" as required. For example, you can translate the message text to your preferred language.
8. Ensure the **Execute for each selected row** check box is not selected, since only single events can be selected as the context for launching a runbook.
 9. Select the appropriate group roles in the **Access Criteria** field. For example, Netcool_OMNIbus_Admin and Netcool_OMNIbus_User.
 10. Click **Save**.

Create a launch-in-context tool configuration for an environment with two Netcool/OMNIbus installations

To create a launch-in-context tool configuration for a setup with two Netcool/OMNIbus installations, complete the following steps:

1. Log on to the Netcool/OMNIBus Web GUI where the Event Viewer is configured.
2. Select **Administration > Event Management Tools > Tool Configuration**.
3. Click **Create Tool** icon to create a new tool.
4. Set the **Name** field to **LaunchRunbookPD**.
5. From the **Type** drop-down field, select **CGI/URL**.
6. Make sure the data source, for example **OMNIBus**, is selected in the data source configuration.
7. In the **URL** field, enter one of the following URLs:

Up to Netcool/Impact 7.1.0.12:

```
https://<RBA_DASH_HOSTNAME>:<RBA_DASH_PORT>/ibm/action/launch?
pageID=ARBTaskLIC&rbid={$$selected_rows.RunbookID}&bulk_params={$$selected_rows.RunbookParameters}&ISC.NEWINSTANCE=false
```

Starting with Netcool/Impact 7.1.0.13:

```
https://<RBA_DASH_HOSTNAME>:<RBA_DASH_PORT>/ibm/action/launch?
pageID=ARBTaskLIC&rbid={$$selected_rows.RunbookID}&bulk_params={$$selected_rows.RunbookParametersB64}&ISC.NEWINSTANCE=false
```

Where <RBA_DASH_HOSTNAME> is the fully qualified domain name of the Netcool/OMNIBus Web GUI server where the RBA DASH integration is installed.

Where <RBA_DASH_PORT> is the DASH port number of the Netcool/OMNIBus Web GUI server where the RBA DASH integration is installed (WebSphere Application Server port **WC_defaulthost_secure**, default port number: 16311).

8. Make sure **Method: GET** is selected.
9. Select **Open in: Specific window**. Enter a name for the window, for example: "IBM Dashboard Application Services Hub". When the launch tool is started for the first time a new DASH browser tab opens with the login screen for the Netcool/OMNIBus Web GUI where the RBA DASH integration is configured. For all subsequent invocations the same browser tab is reused.
10. 10. Ensure the **Execute for each selected row** and **Window for each selected row** check boxes are not selected, since only single events can be selected as the context for launching a runbook.
11. Select the appropriate group roles in the **Access Criteria** section. For example, **Netcool_OMNIBus_Admin** and **Netcool_OMNIBus_User**.
12. Click **Save**.

Create a menu entry in the launch-in-context menu of the Netcool/OMNIBus event list

To add the tool to the list that appears on the pop-up menu in the Netcool/OMNIBus event list, use the following steps:

1. Log on to Netcool/OMNIBus Web GUI as an administrator.
2. Navigate to **Administration > Event Management Tools > Menu Configuration**.
3. Select **Alerts > Modify**
Netcool/OMNIBus Web GUI displays the **Menus Editor** dialog box.
4. Select **LaunchRunbookPD** from the available items.
5. Select the right-arrow button to add LaunchRunbookPD to the **Current Items** list. You can optionally rename the menu entry name. You can also rearrange the order of the menu items.
6. Click **Save**.

For more information, see [“Enable the launch-in-context menu” on page 100](#).

Add Netcool roles to Runbook Automation user groups

Add some Netcool roles to the Runbook Automation user group ARBUserGroup to grant the users in that group access to the Netcool/OMNIBus Event Viewer and to the launch-in-context menu item for Runbook Automation:

1. Log on to Netcool/OMNIBus Web GUI as an administrator.
2. Navigate to **Console Settings > Group roles**.
3. Search for group **ARBUserGroup** and click the group name link.
4. On the Group Roles page, select the roles `ncw_user` and `netcool1_rw`.
5. Click **Save**.

Repeat this procedure for each Runbook Automation user group that should have access to the Netcool/OMNIBus Event Viewer and the launch-in-context menu item for Runbook Automation.

Add the Runbook Automation console integration to your customized Netcool/OMNIBus DASH views

Allow users who are working with customized DASH views to use the Runbook Automation launch-in-context feature.

1. Log on to Netcool/OMNIBus Web GUI as an administrator.
2. Navigate to **Console Settings > Views**.
3. Select the view that needs to be configured for launch-in-context to Runbook Automation.
4. Expand the **Console Integrations in This View** section and click **Add...** to open the table of available console integrations. Search and select the console integration Runbook Automation and add it to the view.

Note: In addition use the WebSphere Application Server administrative console to ensure that all users who need to use the Runbook Automation user interface are members of the appropriate Runbook Automation user groups, as described in [“User groups and roles” on page 45](#).

Configuring certificate validation

Private Deployment

This section covers configuring certificate validation within the RBA server and between the RBA server and the WebSphere Application Server. This first part is a description of the level of security for internal and external communications that is available after installation. The subsequent topics explain how to create and manage certificates in order to improve the level of security of the communication between the RBA services themselves, and between the RBA services and the WebSphere Application Server. It leverages OpenSSL's certificate authority capabilities and demonstrates how you could use them to build an own internal root CA within your company. If you are considering building this kind of infrastructure ensure that you adhere to your corporate security policies and instructions.

The following sequence is an overview of the communication paths in IBM Runbook Automation for private deployment.

1. The user's browser contacts the WebSphere Application Server/DASH user interface and the user logs into DASH.
2. When the user logs into WebSphere Application Server/DASH, the DASH console integration component contacts the RBA service component to retrieve the menu items that need to be displayed in the DASH navigation bar for the particular DASH user.
3. When the user selects an RBA menu item, the browser gets redirected to the RBA service and retrieves and displays the page that is related to the selected menu item.
4. The browser might need to retrieve some of the page content from the other RBA services.
5. The RBA services (runbook service (RBS), automation service (AS), trigger service (TS)) contact each other to retrieve and manage data. All of them run on the same Docker host.

6. The RBA services use the CouchDB database which runs on the same Docker host to retrieve and manage data.
7. The RBA services contact an RBA component in WebSphere Application Server (“ARBUserAuthServlet”) for single sign-on between WebSphere Application Server and the RBA services. Single sign-on uses the WebSphere Application Server LTPA token mechanism.
8. The trigger service (TS) contacts the Impact UI server to create Triggers. Impact triggers for IBM Runbook Automation are used to establish the link between OMNIbus events that match some filter criteria and a runbook.
9. Impact core starts a runbook in the runbook service if an Impact trigger fires based on an OMNIbus event and has a fully automated runbook associated with it, and the trigger has been configured for automatic execution.
10. When the user selects an OMNIbus event in the OMNIbus event viewer and this event has a runbook id associated with it (based on an Impact trigger), the user can open the context menu of the event and launch directly to the runbook execution page.

Pre-installation configuration for WebSphere Application Server single sign-on (SSO)

Private Deployment

WebSphere Application Server must be configured for SSO. This configuration is typically already done when you install and configure IBM Netcool Operations Insight. When the user logs into WebSphere Application Server/DASH, the WebSphere Application Server issues an encrypted and signed cookie (LTPA token) for that user. This LTPA token gets associated with the browser session.

About this task

The WebSphere Application Server/DASH host and the Docker host (where the RBS services are deployed) must be located in the same domain. So that the LTPA token is included in each browser request to an RBA service. To validate the request, the RBA service sends a request to an RBA servlet in the WebSphere Application Server. This request includes the LTPA token again. The WebSphere Application Server validates the LTPA token and invokes the RBA servlet, which returns authorization information about the user to the RBS service.

Procedure

- Navigate to **Security > Global security**.
- Under **Authentication**, click **Web and SIP security > Single sign-on (SSO)**.
- Review the following properties under **General Properties**:
 - The **Enabled** check box must be enabled.
 - The **Requires SSL** check box for should be enabled.
 - The **Domain name** field contains `UseDomainFromURL ;mycompany.com`, where *mycompany.com* must be replaced with the common domain name that is used both for the WebSphere Application Server and for the Docker host.
 - The **Interoperability mode** check box does not need to be enabled.
 - The **LTPA V2 cookie name** field may remain empty.
 - The **Web inbound security attribute propagation** check box must be enabled.
 - The **Set security cookies to HTTPOnly to help prevent cross-site scripting attacks** may be enabled.
 - A WebSphere Application Server restart is required to effect changes.

For more information about WebSphere Application Server SSO, see the WebSphere Application Server documentation at https://www.ibm.com/support/knowledgecenter/SSEQTP_8.5.5/com.ibm.websphere.base.doc/ae/tsec_msso.html.

Initial configuration and manual secure configuration

Private Deployment

When RBA Private Deployment is deployed, the installer creates an initial configuration that allows the product to be used quickly. This facilitates use in proof of concept and test environments. Further post-installation configuration is strongly recommended in production environments.

The initial installation creates a self-signed certificate for the RBA services that must be accepted by each user. For more information, see *Retrieve the RBA server certificate in the browser* in [“Post-installation steps for the RBA DASH integration”](#) on page 27.

Although all communication paths between the various components are encrypted, they are still vulnerable to a "man-in-the-middle" attack. For example, the LTPA token that is used for single sign-on might get stolen and misused. As a result, it is strongly recommended to perform the manual configuration steps for all production environments. This includes importing the (WAS) root CA certificate into the RBA runbook service and replacing the self-signed certificate that was created during installation with a certificate that is issued by your preferred (company-wide) certificate authority. A certificate that is automatically trusted by the browsers in your company.

Level of security after initial installation

Private Deployment

This section summarizes the level of security after the initial Runbook Automation Private Deployment installation.

Creation of self-signed certificate during installation of the RBA service

The self-signed certificate is created without user supplied parameters. The deployment script uses openssl to create the certificate. The only dynamic parameter value is the host name of the Docker host.

Communication between the RBA services

The RBA services and the CouchDB are running on the same Docker host. They share the same private key and self-signed certificate that are created during installation on the Docker host.

At runtime, all communication paths between these services use HTTPS and are encrypted, but the clients do not validate the certificate.

WebSphere Application Server/DASH console integration component → RBA service

At installation time, the self-signed certificate that is provided by the RBS HTTPS port needs to be imported into the WebSphere Application Server trust store. For more information, see [“Post-installation steps for the RBA DASH integration”](#) on page 27.

At runtime, the console integration component in WebSphere Application Server establishes a secure connection to the RBA service based on that certificate.

Browser → RBA services

When the user logs into DASH and selects an RBA related menu item for the first time, his browser notifies him that some content is about to be retrieved from an untrusted source. The user must review and import the self-signed RBA service certificate into the browser.

RBA services → WebSphere Application Server

The RBA services are configured to connect to the secure WebSphere Application Server/DASH port (name in the WAS administrative console: `WC_defaultHost_secure`, default value in the JazzSM profile: 16311), so all data gets encrypted.

The RBA services are initially configured to accept all certificates, so this connection is vulnerable to a man-in-the-middle attack.

Netcool/OMNIBus event → associated runbook (Launch-in-context)

Once the browser has imported the self-signed RBA service certificate, no additional actions are needed to be able to launch the runbook that is associated with an Netcool/OMNIBus event. **Note:** there is no direct communication from Netcool/OMNIBus to any of the RBA services.

Netcool/OMNIBus event → associated fully automated runbook

If Netcool/Impact detects an Netcool/OMNIBus event that matches the trigger criteria for a fully automated runbook, then Netcool/Impact invokes the fully automated runbook. For more information about how to import the RBA server certificate to Netcool/Impact, see [“Import certificate” on page 96](#). **Note:** the documentation is based on the SaaS environment.

For example, if your RBA service is located at the host `myrbasever.mycompany.com`, retrieve the certificate that needs to be imported to Netcool/Impact from `https://myrbasever.mycompany.com:3005`.

Generate and install signed RBA server certificate

Private Deployment

For production environments you must generate and install a signed RBA server certificate. You should plan for a maintenance window to perform the steps in this section because you must install the new certificate into several key stores and enable its use in the RBA service. During that time some communication paths will be temporarily unavailable. The procedures that follow attempt to minimize this disruption. However, depending on the external services you are using (for example IBM Netcool Impact) some disruption is unavoidable.

Generate root CA key and certificate

Private Deployment

Before you begin

If your company has a root certificate authority (CA) certificate available already, and if the root CA certificate has already been imported into your browser, you can skip this procedure and the next section and go straight to [“Generate RBA server key and certificate request” on page 40](#).

Login to the system that you want to use for certificate management. If you intend to use the internal certificate authority (CA) for more than just testing purposes, make sure to establish proper security measurements. Ideally, the system used should have no internet access and possibly not even any network access.

Procedure

1. Create the root CA directory:

```
mkdir -p /root/internalca
cd /root/internalca
```

2. Generate the private key of the root CA:

```
openssl genrsa -out rootCAKey.pem 2048
```

3. Generate the self-signed root CA certificate:

```
openssl req -x509 -sha256 -new -nodes -key rootCAKey.pem -days 3650 -out rootCACert.pem
```

In this example, the validity period is 3650 days. Set the appropriate number of days for your company. Make a reminder to renew the certificate before it expires.

4. Review the certificate:

```
openssl x509 -in rootCACert.pem -text
```

Import root CA certificate into the browser

Private Deployment

You must import the root CA certificate into each browser that will be used to login to IBM Runbook Automation.

Before you begin

If your company has a root certificate authority (CA) certificate available already, and if the root CA certificate has already been imported into your browser, you can skip this procedure and go straight to [“Generate RBA server key and certificate request”](#) on page 40.

About this task

If the certificate is not imported you will be notified by the browser about a potentially insecure connection when you login to the Dashboard Application Services Hub. This is not necessary if you import the root CA certificate.

Complete the following steps to import the root CA certificate.

Note: These instructions apply to Mozilla Firefox, other browsers follow similar procedures.

Procedure

1. Copy the root CA certificate file `rootCACert.pem` to your workstation.
2. Open the browser and click **Options**.
3. Click the **Advanced** tab.
4. Click the **Certificates** tab.
5. Click **View Certificates** to open the Certificate Manager.
6. Click the **Authorities** tab.
7. Click **Import**.
8. Select the root CA certificate file `rootCACert.pem`.
9. Select the check box **Trust this CA to identify websites**.
10. Once you have examined the certificate, click **OK** to add the certificate.
11. Close the **Certificate Manager**.

Generate RBA server key and certificate request

Private Deployment

Follow these steps to generate the RBA server key and certificate request.

Procedure

1. Login to the RBA server.
2. Create the RBA server key file in a temporary directory. You can use another path if `/tmp` is not secure enough to temporarily store the new private key file:

```
mkdir -p /tmp/ibakeys  
cd /tmp/ibakeys  
openssl genrsa -out rbaServerKey.pem 2048
```

Note: The server key file `rbaServerKey.pem` must not be encrypted using a passphrase.

3. Create an OpenSSL request configuration file called `rbaServerCertReq.config` with the following content:

```
[req]
req_extensions = v3_req
distinguished_name = dn
prompt = no

[dn]
CN = <FQDN of your RBA server>
C = <Country Name (2 letter code)>
L = <Locality Name (eg, city)>
O = <Organization Name (eg, company)>
OU = <Organizational Unit Name (eg, section)>

[v3_req]
subjectAltName = DNS:<FQDN of your RBA server>
```

You can remove items that are unnecessary for your organization from the list of informational items below the `[dn]` entry. However, you must include the `CN` item. You can also add additional informational items (such as email address) according to the OpenSSL specification. Replace the remaining placeholders `<...>` with the appropriate information. For the **CN** field, enter the fully qualified domain name of the RBA server, for example, `myrbaServer.mycompany.com`. In the other fields, enter data that describes your location and organization, in accordance with your company's security guidelines.

4. Create the RBA server certificate signing request (CSR) file:

```
openssl req -new -key rbaServerKey.pem -sha256 -out rbaServerCert.csr -config
rbaServerCertReq.config
```

5. Enter the following command to review the CSR file:

```
openssl req -in rbaServerCert.csr -noout -text
```

6. Verify that the CSR file has a section such as the following:

```
Requested Extensions:
    X509v3 Subject Alternative Name:
        DNS:<FQDN of your RBA server>
```

In addition, make sure the common name field is set to the FQDN of the RBA server.

Generate the RBA server certificate

Private Deployment

Complete the following steps to generate the RBA server certificate.

Procedure

1. Switch to the system where you created the root CA directory, and create a new subdirectory:

```
mkdir /root/internalca/certs
cd /root/internalca/certs
```

2. Copy the CSR file `rbaServerCert.csr` into that directory.
3. Create an OpenSSL request signing configuration file `v3.ext` that contains the following line:

```
subjectAltName = DNS:<FQDN of your RBA server>
```

4. Create the signed certificate for the RBA server from the CSR:

```
openssl x509 -req -sha256 -in rbaServerCert.csr -CA ../rootCACert.pem -CAkey ../
rootCAKey.pem -CAcreateserial -out rbaServerCert.pem -days 365 -extfile v3.ext
```

In this example, the validity period is 365 days. Set the appropriate number of days for your company. Make a reminder to renew the certificate before it expires.

5. Enter the following command to review the certificate:

```
openssl x509 -in rbaServerCert.pem -text
```

6. Verify that the certificate has a section such as the following:

```
X509v3 extensions:  
    X509v3 Subject Alternative Name:  
        DNS:<FQDN of your RBA server>
```

Install the RBA server certificate on the RBA server

Private Deployment

Complete the following steps to install the RBA server certificate on the RBA server.

Procedure

1. Copy the generated certificate file `rbaServerCert.pem` from the system where you created the root CA directory to the RBA server directory `/tmp/rbakeys`.
2. On the RBA server, replace the self-signed key and certificate that were created during installation with the new files, and remove the temporary directory:

```
cd /etc/ibm/arb/keys  
cp rbaServerKey.pem rbaServerKey.pem.bakFromInstallation  
/bin/cp -f /tmp/rbakeys/rbaServerKey.pem .  
cp rbaServerCert.pem rbaServerCert.pem.bakFromInstallation  
/bin/cp -f /tmp/rbakeys/rbaServerCert.pem .  
rm -rf /tmp/rbakeys
```

3. Use the following command to restart the RBA service: `iba restart`.

Replace the RBA server certificate in the WAS trust store

Private Deployment

You need to replace the original self-signed RBA server certificate that is stored in the WebSphere Application Server trust store with the new signed RBA server certificate.

About this task

Perform the following steps to replace the RBA server certificate:

Procedure

1. Make sure the RBA service has been started.
2. Login to the WAS administrative console as WAS administrator.
3. Click **Security > SSL certificate and key management**.
4. Click **Key stores and certificates**.
5. Click **NodeDefaultTrustStore**.
6. Click **Signer certificates**.
7. On the **Signer certificates** page select the checkbox in the table row for the alias **rba.server.certificate**.
8. Click **Delete**.
9. Click **Save** to save the changed configuration.
10. Click **Retrieve from port**.
11. On the **Retrieve from port** panel, enter the following data:
 - a) **Host**: enter the fully qualified domain name of the RBA server, for example: `myrbaserver.mycompany.com`.
 - b) **Port**: enter the port number of the RBS service (3005).
 - c) **SSL configuration for outbound connection**: use the default value `NodeDefaultSSLSettings`.

- d) **Alias:** enter `rba.server.certificate`.
12. Click **Retrieve signer information**.
13. Review the data displayed in **Retrieve signer information**.
14. After you have validated the signer information, click **OK**.
15. Click **Save**.

Replace the RBA server certificate in Netcool/Impact

Private Deployment

If you have already set up the connection between the RBA server and Netcool/Impact, you must replace the RBA server certificate in the Netcool/Impact key store with the new certificate.

Follow the instructions in [“Import certificate” on page 96](#) to delete the old certificate and to import and install the new certificate.

Test the signed RBA server key and certificate

Private Deployment

You can test if the new signed RBA server key and certificate files can be loaded and used. Check if you can still log in, browse runbooks, and perform other tasks in the Runbook Automation user interface.

Note: At this point, although the browser considers the connection now as secure and does no longer ask you to verify a self-signed certificate when you access the Runbook Automation user interface, the actual level of security for the internal communication within the RBA service has not been changed. For more details, see [“Level of security after initial installation” on page 38](#). To increase the level of security continue with [“Enable secure communications within the RBA service” on page 44](#).

Renewing the signed RBA server certificate

Private Deployment

The signed RBA server certificate must be replaced with a new certificate before it expires. The replacement procedure is similar to the procedure for creating and installing the initial certificate.

Before you begin

Because you need to install the new certificate into several key stores and enable its use in the RBA service, you should plan for a maintenance window to perform the required steps. During that time some communication paths will be temporarily unavailable. The following procedure attempts to minimize this disruption. However, depending on the external services you are using (for example IBM Netcool Impact) some disruption is unavoidable.

Procedure

1. Find the certificate request file created in [“Generate RBA server key and certificate request” on page 40](#). You can also create a new certificate request. You do not need to create a new RBA server key.
2. Generate the RBA server certificate as described in [“Generate the RBA server certificate” on page 41](#).
3. Import the new RBA server certificate to the WebSphere Application Server trust store.

Note: you cannot use the **Retrieve from port** feature as the new certificate cannot be obtained from the RBA server port yet. You must import the RBA server certificate file directly.

4. **If you are using Netcool/Impact:** Import the new RBA server certificate into the Netcool/Impact server key store. If you are using a procedure to dynamically retrieve the RBA server certificate, you must postpone this step until the RBA service has been restarted.
5. Install the new RBA server certificate as described in [“Install the RBA server certificate on the RBA server” on page 42](#).
6. Restart the RBA service: `rba restart`.
7. If you have not yet imported the certificate to the WebSphere Application Server trust store, then you should now update the WebSphere Application Server trust store as described in [“Replace the RBA server certificate in the WAS trust store” on page 42](#). You can use the **Retrieve from port** feature now.

Enable secure communications within the RBA service

Private Deployment

Complete the steps in this section to enable secure communications within the RBA service for production environments.

Before you begin

Make sure you have completed the tasks outlined in [“Generate and install signed RBA server certificate”](#) on page 39.

Procedure

- **Extract WAS root certificate**

Perform the following steps to export the WAS root CA certificate using the WAS administrative console:

- a) Login to the WAS administrative console as WAS administrator.
- b) Click **Security > SSL certificate and key management**.
- c) Click **Key stores and certificates**.
- d) Click **NodeDefaultTrustStore**.
- e) Click **Signer certificates**.
- f) On the **Signer certificates** page select the checkbox in the table row for the alias **root** and click **Extract**.

Note: If you have replaced the self-signed WAS root certificate that was created during the initial installation of the WebSphere Application Server with a signed certificate, then use the appropriate signed certificate instead.

- g) On the **Extract signer certificate** page:
 - a. Enter a fully qualified file name for the WAS root CA certificate, for example `/tmp/wasRootCACert.pem`.
 - b. In the **Data type** field, keep the default selection (Base64-encoded ASCII data).
 - c. Click **OK**.

- **Install the WAS root CA certificate on the RBA server**

- a) Copy the extracted WAS root CA certificate file (for example, `wasRootCACert.pem`) from the WAS server to the RBA server directory `/etc/ibm/arb/keys`. Make sure the file is readable for all users.
- b) Edit the configuration file `/etc/ibm/arb/rba.config` and set the value of the property `was_root_ca_cert_filename` to the WAS root CA certificate file name (for example, `wasRootCACert.pem`).

- **Install the root CA certificate on the RBA server**

- a) Copy the root CA certificate file of the internal CA authority (for example, `rootCACert.pem`) from the system you use for certificate management to the RBA server directory `/etc/ibm/arb/keys`. Make sure the file is readable for all users.
- b) Edit the configuration file `/etc/ibm/arb/rba.config` and set the value of the property `root_ca_cert_filename` to the root CA certificate file name (for example, `rootCACert.pem`).

- **Store the target server certificates for Impact or Automation connections**

For **BigFix** and **Event Trigger** connections, the target server certificate must be specified in the connection properties. Each of those certificates must be valid and have a value for the **common name** (CN) property that matches the host name of the target server.

Click **Connections** and for each connection click **Edit**. If the target system certificate is not yet stored, complete the following steps:

- a) On Linux systems, enter the following command to receive the correct certificate:

```
echo -n | openssl s_client -connect <TARGET_SERVER_URL> | sed -ne '/-BEGIN
CERTIFICATE-/,/-END CERTIFICATE-/p' > serverCert.pem
```

Replace the <TARGET_SERVER_URL> with your target server URL. For example:

```
echo -n | openssl s_client -connect myimpact.mycorp.com:16311 | sed -ne '/-BEGIN
CERTIFICATE-/,/-END CERTIFICATE-/p' > serverCert.pem
```

If errors occur, make sure your exported certificate that is stored in file .cert contains a full and valid certificate in PEM format. Errors such as "verify error:num=20:unable to get local issuer certificate" occur due to a missing CA root certificate for the DigiCert CA.

- b) Review the certificate and ensure that the target system hostname (which you use in the URL) matches the hostname defined in the CN or in the certificate altnames (for example, myimpact.mycorp.com):

```
openssl x509 -in serverCert.pem -text
```

- c) Display the content of serverCert.pem:

```
cat serverCert.pem
```

The certificate begins and ends as follows:

```
-----BEGIN CERTIFICATE-----
....
-----END CERTIFICATE-----
```

- d) Copy the target system certificate (including the lines with BEGIN CERTIFICATE and END CERTIFICATE) into the connection's **server certificate** field.

- e) Click **Save**.

- **Enable the secure mode to allow certificate checking**

- a) Edit the configuration file /etc/ibm/arb/rba.config and set the value of the property secure_comm_mode_enabled to true.
- b) Restart the RBA service for the changes to take effect: rba restart.
- c) Verify that RBA UI functionality works, and that RBA service logs show that the secure comm mode is enabled.

User groups and roles

Private Deployment

The level of authority that is associated with a user depends on their assigned user group and role.

User groups in IBM Runbook Automation private deployment are handled in one of two ways:

WebSphere Application Server is set up to actively manage users and groups

Runbook Automation creates some product-specific user groups when installing the RBA DASH integration.

WebSphere Application Server is set up with read-only access to an external user repository

Runbook Automation expects that user groups have already been created before the RBA DASH integration is installed. For more information, see [“Create users and groups in the external user repository with read-only access”](#) on page 21.

The RBA DASH integration installer associates these Runbook Automation related groups with the Runbook Automation user roles. As a result, if you add a WebSphere user to one of the groups, the user receives the authority to execute all tasks that are associated with the group's roles.

The following table lists the Runbook Automation private deployment groups and the related roles and menu items. The level of authority that is associated with each group increases as you go down the table.

<i>Table 16: RBA DASH integration users and groups</i>			
Default Group Name	Description	Roles	Menu Items
ARBUserGroup	IBM Runbook Automation users with the authority to perform operational activities.	ARBUser	Library (without delete, edit and approver functionality, only published runbooks are shown), Execution
ARBAuthorGroup	IBM Runbook Automation users with the additional authority to create and manage runbook drafts, triggers, and automations.	ARBAuthor ARBUser	Library (without approver functionality published and draft runbooks are shown), Triggers, Automations and all of the preceding menu items.
ARBAApproverGroup	IBM Runbook Automation users with the additional authority to approve runbooks.	ARBAApprover ARBAuthor ARBUser	Library (with approver functionality) and all of the preceding menu items.
ARBManagerGroup	IBM Runbook Automation users with the additional authority to Connections and manage API keys.	ARBManager ARBAApprover ARBAuthor ARBUser	Connections, API Keys , and all of the preceding menu items.

In addition to the levels of authority that are listed in the table, all authenticated WebSphere users can use the **Getting Started** and **About** menu items.

Changing the mapping between RBA user roles and the associated WebSphere user groups

The names of Runbook Automation roles cannot be changed, but you can change the mapping between these roles and the associated WebSphere user groups. Complete the following steps to change the mapping in the WebSphere administrative console:

1. Login to the WebSphere administrative console.
2. Navigate to **Applications > Application Types > WebSphere enterprise applications**.
3. Click the **isc** application link.
4. Under Detail properties, click **Security role to user/group mapping**. A list of all the roles that belong to this application is displayed.
5. Select the Runbook Automation user role that you want to change and click **Map Groups...**
6. Search and select all user groups that you want to assign the role's authority to.
7. Click **OK** to close the mapping editor.
8. Repeat steps 5-7 for each role that needs to be adjusted.
9. Click **OK** to close the **Security role to user/group mapping** editor.
10. Save the configuration.

Creating, adding, and removing WebSphere users

Private Deployment

All user administration for IBM Runbook Automation private deployment is performed in the same way as the user administration for Netcool OMNIbus, either directly within the WebSphere Application Server administrative console or within an external user repository that the WebSphere Application Server uses to retrieve user related information.

Adding a WebSphere user to an IBM Runbook Automation user group

If the WebSphere administrative console is configured to directly manage users, use the WAS administrative console to add the WebSphere user to the appropriate Runbook Automation user group. Otherwise use the administrative interface of the external user repository to add the WebSphere user to the appropriate Runbook Automation user group.

If the user should be able to use the Netcool OMNIbus event viewer and launch runbooks from there, then also add the user to one of the groups `Netcool_OMNIbus_Admin` or `Netcool_OMNIbus_User`.

Removing a WebSphere user from an IBM Runbook Automation user group

If the WebSphere administrative console is configured to directly manage users, use the WAS administrative console to remove the WebSphere user from the IBM Runbook Automation user group. Otherwise use the administrative interface of the external user repository to remove the WebSphere user from the Runbook Automation user group.

Operating the RBA service

Private Deployment

The Runbook Automation service is operated using the `rba` command.

Procedure

- To start the server:

```
rba start
```

Note:

Note:

- This command starts the Runbook Automation service only. It does not start the RBA DASH integration. Before you can use Runbook Automation Private Deployment, both the Runbook Automation service and the RBA DASH integration must be started.
 - When `rba start` is invoked for the first time, it will request configuration parameters and store them in the configuration file. If no password for the database is specified in the configuration file, this command will request a new password which will be valid as long as the docker container for the database is running. The password will not be stored permanently. If you do not want to specify a database password at each start, you can configure the password in the RBA configuration file permanently. For more information, see [“Configuring the RBA service” on page 30](#).
- To stop the server:

```
rba stop
```

- To restart the server:

```
rba restart
```

Restarting the RBA service is required if the RBA service is updated or the RBA service configuration has changed. It might also be required as part of problem determination.

Note: If no password for the database is specified in the configuration file, this command will request a new password which will be valid as long as the docker container for the database is running. The password will not be stored permanently. If you do not want to specify a database password at each start, you can configure the password in the RBA configuration file permanently. For more information, see [“Configuring the RBA service”](#) on page 30.

- To retrieve status information such as running containers, available images, and current RBA version:

```

rba status

Example output
status of rba containers:
24ddc2144d50      ibm/rba-rbs:1.4.0-1606141703      "/entrypoint.sh start"      43 hours
ago              Up 17 hours      0.0.0.0:3005->3005/tcp      rba-rbs
a2425ff7ba9e      ibm/rba-ts:1.4.0-1606141703      "/entrypoint.sh start"      43 hours
ago              Up 17 hours      0.0.0.0:3006->3006/tcp      rba-ts
3f91e98bfe12      ibm/rba-as:1.4.0-1606141703      "/entrypoint.sh start"      43 hours
ago              Up 17 hours      0.0.0.0:3080->3080/tcp      rba-as
7a0c23e1f034      ibm/rba-couchdb:1.4.0-1606141703  "/entrypoint.sh couch"      45 hours
ago              Up 17 hours      6984/tcp                    rba-couchdb

available rba images:
ibm/rba-rbs      1.4.0-1606141703      a6d30e0512f5      3 days ago      398.4 MB
ibm/rba-rbs      1.4.0-1606161642      a6d30e0512f5      3 days ago      398.4 MB
ibm/rba-as       1.4.0-1606141703      192e0eebf1f3      3 days ago      311.3 MB
ibm/rba-as       1.4.0-1606161642      192e0eebf1f3      3 days ago      311.3 MB
ibm/rba-ts       1.4.0-1606141703      bc452f0796d3      7 days ago      321.4 MB
ibm/rba-ts       1.4.0-1606161642      bc452f0796d3      7 days ago      321.4 MB
ibm/rba-couchdb 1.4.0-1606141703      9fa4e863fff6      8 days ago      537.9 MB
ibm/rba-couchdb 1.4.0-1606161642      9fa4e863fff6      8 days ago      537.9 MB

available rba rpm packages:
rba-1.4.0-1606161642.x86_64

```

- To get the current version of the RBA service:

```
rba version
```

Cleaning up old docker images

Private Deployment

Each update of an RBA service is shipped via docker images. The old docker images are not automatically removed so that you can roll back to a previous version if required. Old images that are no longer required can be removed through docker commands.

Procedure

- To list all docker images:

```

docker images

Example output
REPOSITORY          TAG          IMAGE ID          CREATED          VIRTUAL SIZE
ibm/rba-rbs         1.3.0-1606221124  8a001a4e233d     4 weeks ago     321.4 MB
ibm/rba-as          1.3.0-1606221124  2e2e0eebf1f3     4 weeks ago     321.4 MB
ibm/rba-ts          1.3.0-1606221124  ad452f0796d3     4 weeks ago     321.4 MB
ibm/rba-couchdb    1.3.0-1606221124  bba4e863fff6     4 weeks ago     537.9 MB
ibm/rba-rbs         1.4.0-1606221124  88001a4e233d     26 hours ago    380.3 MB
ibm/rba-as          1.4.0-1606221124  192e0eebf1f3     9 days ago      311.3 MB
ibm/rba-ts          1.4.0-1606221124  bc452f0796d3     13 days ago     321.4 MB
ibm/rba-couchdb    1.4.0-1606221124  9fa4e863fff6     2 weeks ago     537.9 MB

```

- To remove old docker images:

```
docker rmi <image id> [image id ...]
```

For example, use the following command to delete the old images for version 1.3.0-*:

```
docker rmi 8a001a4e233d 2e2e0eebf1f3 ad452f0796d3 bba4e863fff6
```

The RBA commands start and stop the RBA service without creating orphaned docker volumes. However, due to abnormal conditions, it might be possible that orphaned docker volumes exist.

- To check if orphaned docker volumes exist:

```
docker volume ls -qf dangling=true
```

- To delete orphaned docker volumes:

```
docker volume rm $(docker volume ls -qf dangling=true)
```

Uninstalling

Private Deployment

There are two parts to uninstalling an IBM Runbook Automation private deployment.

Uninstalling the RBA service

About this task

Procedure

1. Use the following command to uninstall the RBA service:

```
yum remove rba
```

The following data is kept after uninstallation and available again after a new installation:

- Configuration data
 - Log files
 - Data in **couchdb** is kept and not deleted in case of a new installation.
2. To clean up all data, remove the following directories:
 - /etc/ibm/arb
 - /var/log/ibm/arb
 - /var/lib/ibm/arb

Uninstalling the RBA DASH integration

Procedure

1. Start the IBM Installation Manager, for example:

```
/opt/IBM/InstallationManager/eclipse/IBMIM
```

2. Click **Uninstall**.
3. Select the package **RBA DASH integration** and follow the on-screen instructions.

Troubleshooting

Private Deployment

Review this information for help in resolving some of the issues you might encounter.

Collecting problem determination information for the RBA service

Private Deployment

Before you begin

The RBA service writes traces by default. The tracing level cannot be reconfigured.

Procedure

1. Log in to the RBA service.
2. Run the following command:

```
/opt/ibm/arb/bin/rbagetlogs.sh
```

Example output:

```
IBM Runbook Automation data collection finished  
Support data bundle can be found at /tmp/rba-pddata/201606170932.tar.gz
```

3. Provide the tar.gz file to IBM support.

Writing RBA service traces to log file instead of syslog

Private Deployment

If the RBA service trace entries are flooding the syslog (/var/log/messages) then you should reconfigure the docker logging option **log-driver**.

Procedure

1. Logon to the RBA service.
2. Issue the command `docker info | grep Logging`.
3. If the result shows `Logging Driver: journald`, complete the following steps:
 - a) Edit the file `/etc/sysconfig/docker` and set the `OPTIONS` property to `OPTIONS='--log-driver=json-file'`.
 - b) Restart docker: `service docker restart`.

Enabling and collecting problem determination information for the RBA DASH integration

Private Deployment

About this task

Perform the following steps in the WebSphere Application Server administrative console to enable and collect diagnostic traces for the RBA DASH integration component.

Procedure

- Enable diagnostic traces for the RBA DASH integration component:
 - a) 1. Navigate to **Troubleshooting > Logs and trace**.
 - b) In the table, click **server1** server resource.
 - c) Click **Change log detail levels**.
 - d) Click the **Runtime** tab.
 - e) Select the **Save runtime changes to configuration as well** check box.
 - f) Enter the string `*=info: com.ibm.arb.*=all` in the text entry field. Alternatively, expand the **Components and Groups** area and select **com.ibm.arb.* > All Messages and Traces**.
 - g) Click **OK**.
This enables writing the traces for the RBA DASH integration component immediately.
- To collect the problem determination information, log in to the server where the RBA DASH integration is installed and issue the following command:

```
/opt/IBM/arb/was/install/tools/rbagetdilog.sh
```

Messages in /var/log/messages

Private Deployment

The RBA service does not log messages to /var/log/messages. However, docker logs messages to /var/log/messages and those messages can be used to debug docker issues.

For example, the following messages are shown if a docker container required a restart:

```
Jun 15 16:00:01 rbaserver systemd: Stopped docker container
24ddc2144d5013683149f7108c6afa023628d9b98f20ff023d66d0ea23f1eb77.
Jun 15 16:00:01 rbaserver systemd: Stopping docker container
24ddc2144d5013683149f7108c6afa023628d9b98f20ff023d66d0ea23f1eb77.
Jun 15 16:00:01 rbaserver systemd: Started docker container
24ddc2144d5013683149f7108c6afa023628d9b98f20ff023d66d0ea23f1eb77.
Jun 15 16:00:01 rbaserver systemd: Starting docker container
24ddc2144d5013683149f7108c6afa023628d9b98f20ff023d66d0ea23f1eb77.
```

Note: Messages shown are for docker version 1.9 and may be different on other docker versions.

RBA service error scenarios

Private Deployment

Use this information to troubleshoot RBA service issues.

Symptom	Explanation	Resolution
During installation of the RBA service (yum localinstall rba-...rpm, the following error message is shown: <pre> rba Requires: docker >= 1.9 Error: Package: rba-...) Requires: docker >= 1.9</pre>	yum cannot automatically resolve the dependency to docker. The reason might be because the RedHat Satellite Channel Server Extras is not enabled.	Add the subscription for RHEL Server Extras . For example: <pre>RHEL Server Extras (v.7 for 64-bit x86_64)</pre>

Table 17: RBA service error scenarios (continued)

Symptom	Explanation	Resolution
<p>During installation of the RBA service (yum localinstall rba-... rpm, transaction check errors are shown. For example:</p> <pre>file /usr/lib/systemd/system/blk-availability.service from install of device-mapper-7:1.02.107-5.el7_2.2.x86_64 conflicts with file from package lvm2-7:2.02.105-14.el7.x86_64 file /usr/sbin/blkdeactivate from install of device-mapper-7:1.02.107-5.el7_2.2.x86_64 conflicts with file from package lvm2-7:2.02.105-14.el7.x86_64 file /usr/share/man/man8/blkdeactivate.8.gz from install of device-mapper-7:1.02.107-5.el7_2.2.x86_64 conflicts with file from package lvm2-7:2.02.105-14.el7.x86_64yu</pre>	<p>This problem is caused by incorrect package dependency descriptions in the docker rpm packages or related rpm packages.</p>	<p>It is recommended to update all modules to the latest version:</p> <pre>yum update</pre> <p>If you do not want to update all installed modules, you can attempt to remove the conflicting component and retry, for example:</p> <pre>yum remove lvm2</pre>
<p>During startup of the RBA service, the following warning is shown:</p> <pre>Usage of loopback devices is strongly discouraged for production use. Either use `--storage-opt dm.thinpooldev` or use `--storage-opt dm.no_warn_on_loop_devices=true` to suppress this warning.</pre>	<p>You are using the default docker storage driver setup which is not recommended for production use.</p>	<p>Configure the docker storage driver as described in “<u>Docker storage driver</u>” on page 20.</p>

RBA DASH integration error scenarios

Private Deployment

Use this information to troubleshoot RBA DASH integration issues.

Table 18: RBA DASH integration error scenarios

Symptom	Explanation	Resolution
After login to DASH, you do not see any IBM Runbook Automation related menu items, or the WebSphere Application Server/DASH page is empty.	The DASH console integration, the browser, or both, cannot connect to the RBA service, or the single sign-on is failing.	Ensure that you are using a fully qualified domain name to access. For example, the URL must be <code>https://mydash.mycompany.com:16311/ibm/console</code> and not just <code>https://mydash:16311/ibm/console</code> . Also ensure that the firewall configuration is correct. If the firewall is enabled on the RBA server, try temporarily disabling the firewall on the RBA server and restart the docker service to activate the firewall change.
After login to DASH, you see the IBM Runbook Automation related menu items, but as soon as you click a menu item, the DASH login page is displayed again.	The single sign-on is failing.	Ensure that the WebSphere Application Server configuration for single sign-on is properly defined, as described in “Pre-installation configuration for WebSphere Application Server single sign-on (SSO)” on page 37.
The About page shows error icons for one or more components.	<ol style="list-style-type: none"> The component’s self-signed certificate has not been imported into the browser yet. The component is not running. 	<ol style="list-style-type: none"> See Retrieve the RBA server certificate in the browser in “Post-installation steps for the RBA DASH integration” on page 27. Restart the RBA service.
Some Runbook Automation pages do not show any data although other users can see the data.	The self-signed certificate for some RBA components have not been imported into the browser.	Verify that the About page displays the correct data version for all components.
The user expects to have access to more Runbook Automation related menu items than the ones that are offered.	The user’s WebSphere Application Server user id has not been added to the appropriate user group.	See “User groups and roles” on page 45 for information about the authority related to Runbook Automation user groups and how to administer users.

Verifying the DASH console integration to the RBA service

Private Deployment

Before you begin

Before you continue with this section, verify first that all actions described in [“Post-installation steps for the RBA DASH integration”](#) on page 27 have been completed.

About this task

- Login to the DASH console with a WebSphere Application Server user id that is in the **iscadmins** role.

- Navigate to **Console Settings > Console Integrations**. In the table of console integrations, click **Runbook Automation**. Verify that the Console Integration URL looks like the following example: `https://myrbahost.mycompany.com:3005/ci/rest`.
- Click the **Test** button and check that the status field indicates that the connection has been verified successfully, and the table of available tasks contains **Getting Started, About, Library**, and other tasks.

If the test fails to verify the connection to the RBA service, perform the following steps.

Procedure

1. Ensure that the RBA service is running. For more information, see [“Operating the RBA service” on page 47](#).
2. Ensure that the RBA service can be reached from the RBA DASH integration server.
 - a) Open an operating system shell on the server where the RBA DASH integration has been installed.
 - b) Issue `ping <rbaServerFQDN>`.
Where `<rbaServerFQDN>` is the fully qualified domain name of the RBA server host. Make sure the host can be reached successfully.
 - c) Issue `curl -k https://<rbaServerFQDN>:3005/` and verify that the result looks like the following:

```
Found. Redirecting to https://<rbaDashFQDN>:16311/ibm/console/logon.jsp
```

Where `<rbaServerFQDN>` is the fully qualified domain name of the RBA DASH integration host, for example: `mydash.mycompany.com`. If this does not work, check the firewall setup and adjust as required.

3. Ensure that the RBA DASH integration server can be reached from the RBA server.
 - a) Open an operating system shell on the server where the RBA DASH integration has been installed.
 - b) Issue `ping <rbaDashFQDN>`.
Where `<rbaDashFQDN>` is the fully qualified domain name of the RBA DASH integration host. Make sure the host can be reached successfully.
 - c) Issue `curl -k https://<rbaDashFQDN>:16311/` and verify that the result looks like the following:

```
H1>SRVE0255E: A WebGroup/Virtual Host to handle / has not been defined.</H1><BR><H3>SRVE0255E: A WebGroup/Virtual Host to handle <rbaDashFQDN>:16311 has not been defined.</H3><BR><I>IBM WebSphere Application Server</I>
```

If this does not work, check the firewall setup and adjust as required.

4. Ensure that the RBA DASH integration server can be reached from within the RBS service Docker container.
 - a) Open an operating system shell on the server where the RBA server has been installed.
 - b) Issue `docker ps`.
 - c) Issue `docker exec -it rba-rbs /bin/bash`.
 - d) Issue `ping <rbaDashFQDN>`.
Where `<rbaDashFQDN>` is the fully qualified domain name of the RBA DASH integration host. Make sure the host can be reached successfully.
 - e) Issue `curl -k https://<rbaDashFQDN>:16311/` and verify that the result looks like the following:

```
<H1>SRVE0255E: A WebGroup/Virtual Host to handle / has not been defined.</H1><BR><H3>SRVE0255E: A WebGroup/Virtual Host to handle <rbaDashFQDN>:16311 has not been defined.</H3><BR><I>IBM WebSphere Application Server</I>
```

If this does not work, check the firewall setup and adjust as required.

RBA service directories

Private Deployment

Table 19: RBA service directories

Directory	Description
/opt/ibm/arb	Product code
/opt/ibm/arb/bin	RBA commands
/opt/ibm/arb/etc	Config file example
/etc/ibm/arb/keys	Server Certificates. During the first startup of the RBA service, self signed server certificates are created automatically.
/etc/ibm/arb	RBA configuration
/var/log/ibm/arb	Logs

Chapter 4. Users and their daily tasks

Create and run runbooks and automated tasks for faster, more repeatable, and consistent problem resolution. Transition from documented procedures, written in a wiki or text document, to consistent and reusable runbooks.

These are some of the benefits of using IBM Runbook Automation:

Investigate and delegate problems faster and more efficiently.

Events can be mapped to runbooks so that operators can react quickly and efficiently to the problem that is communicated by the event. If the operator can immediately access the right runbook, the solution to the problem is already on its way.

Diagnose and fix problems faster and build operational knowledge.

Operators can document their knowledge in reusable runbooks. All operators will benefit from the knowledge and the experience that is documented in a runbook.

Easily create, publish, and manage Runbooks and automations.

Create, publish, and Runbooks with a few clicks.

Track achievements and identify opportunities for improvement.

Metrics are provided to give you visibility on how successfully a runbook was executed. Operators can provide feedback on their level of satisfaction with a runbook and any ideas they have for improvement.

Who works with runbooks?

IBM Runbook Automation can help you solve day to day problems faced by service delivery engineers and operations analysts.

IBM Runbook Automation is used by operators, whose job it is to solve IT problems that are indicated by events. Solving these problems requires two different user types:

Operations Analyst

The operations analyst runs runbooks. On the **Library** page, the operation analysts can find their working environment. The **Available Runbooks** table provides all runbooks that an operations analyst can execute. On the **Execution** page, you can see the executed runbooks and check your previous work.

Service Delivery Engineer

Service delivery engineers create and edit runbooks. They monitor each runbook's metrics to evaluate how they are performing. It is up to the service delivery engineers to improve runbooks by introducing automation, and react to any feedback that is provided.

Access control

This section explains the rules around access control and runbooks.

By default, all operations analysts can access all runbooks within a subscription. In some cases it might be preferable to restrict access to certain runbooks. For example:

- Operations analysts are specialists in certain areas, so runbooks from other areas might not be relevant.
- There might be security reasons for restricting access to certain runbooks.

For more information about introducing access control, see [“Create users and assign roles” on page 10](#).

What are the rules for accessing a runbook?

The following simple rules apply to access control for runbooks:

- Access control affects only operations analysts who are assigned the **user** role in IBM Runbook Automation. Users assigned any other role can access all runbooks, regardless of their group assignment.
- Operations analysts who are members of any groups can access only runbooks that are assigned to at least one of those groups.
- Operations analysts who are not members of any groups can access all runbooks.

Note: Since the default behavior is to allow access to all runbooks, any new user added to the subscription can access all runbooks, even if existing users are already restricted by their group membership. If you are using access control, remember to always add new users to some group(s) after they are added to the subscription.

What exactly is affected by the access control?

Users who do not have access to a runbook are not able to:

- See the runbook on the **Library** page.
- Execute the runbook from a direct Launch in Context URL.

They can however:

- See a runbook execution in the **Execution** page, even if access to the runbook has been removed since a user ran it.
- See the details of any runbook execution (this includes seeing the runbook).
- Resume the runbook execution.

Assigning runbooks to groups

Cloud

Each runbook can be assigned to one or more groups, or left unassigned.

Before you begin

If you only want to check what groups a runbook is assigned to, complete steps 1 - 3 below but do not make any changes.

About this task

Assigning runbooks to groups allows you to organize runbooks in a more efficient manner. There are two main reasons for assigning runbooks to groups:

- Groups determine the visibility of runbooks for operations analysts.
- It is easier to find relevant runbooks in the **Library** page when [“Filtering runbooks” on page 79](#) by groups.

Complete the following steps to assign runbooks to groups.

Procedure

1. Go to **Library**.
2. Use the check boxes to select one or more runbooks.
3. An action bar is displayed. The bar shows available actions and how many runbooks are selected.
4. On the action bar, click **Grant permission**.
5. Use check boxes to choose which groups the selected runbooks should belong to. You can also select all or unselect all groups by using the checkbox in the header row.

Note: Some checkboxes might appear selected, but dimmed. This is an intermediate state that occurs when some (but not all) of the selected runbooks belong to the group.

6. Click **Assign** to confirm the selection or **Close** to discard any changes.

See runbooks assigned by group

Use the filtering capability to check which runbooks are assigned to a group.

Procedure

1. Go to **Library**.
2. Click the Group filter.
3. Use the check boxes to select one or more group.

Results

Now only runbooks assigned to the selected groups are displayed.

Run and monitor runbooks

Find a matching runbook for any problem that you need to resolve. Issues might be reported by an event, a ticket, or an alert.

An Operations Analyst runs and monitors runbooks. Log on as RBA User to run and monitor runbooks:

Run a runbook

If you found the runbook that can resolve your issue, run the runbook.

For more information, see [“Run your first runbook” on page 67](#).

Monitor runbooks

If you want to review what you done to date you can view the runbooks that you ran.

For more information, see [Chapter 6, “Executions,” on page 81](#).

Create and edit a runbook

Move from procedures that are documented in wikis and other sources to consistent and reusable runbooks.

As an RBA Author and RBA Approver, you can create and edit a runbook:

Create a runbook

Create your procedure steps, use commands, parameters, automation scripts, or GOTO elements.

For more information, see [“Create a runbook” on page 69](#).

Edit a runbook

Incrementally improve your runbook by incorporating comments from other operators.

For more information, see [“Edit a runbook” on page 70](#).

Save and publish the runbook

Save your work and share it with the rest of the team.

For more information, see [“Save and publish a runbook” on page 78](#).

Create different versions

Work together and refine the runbook by creating different versions.

For more information, see [“Runbook versions” on page 78](#).

Run steps automatically

Operators will often need to execute the same steps over and over again. Automations allow you to run repeatable steps automatically with one click.

As a Service Delivery Engineer, you must be logged on as an RBA Author to create automations. The following topics outline the tasks involved.

“Create a runbook” on page 69

Create a runbook and run the runbook.

“Improving runbook execution” on page 63

Gather feedback and monitor runbook statistics to gain knowledge about the runbook.

“Create an automation” on page 83

Create an automation to summarize a couple of steps in one single step.

“Create a connection” on page 89

Runbooks need to be able to connect to system backends, so that you can run automations.

Map Netcool/OMNIBus events to runbooks

If you have events that always correspond to the same runbook, create a trigger and link the events with the runbook.

As a Service Delivery Engineer, you must be logged on as an RBA Author or RBA Approver to create a trigger.

Install the trigger service

For more information, see [“Installing Netcool/Impact to run the trigger service” on page 94](#).

Create a trigger

Triggers map events from Netcool/Impact with runbooks that solve the problems reported by the event. Triggers can run with manual and automated runbooks. If the runbook is a manual or a semi-automated runbook, the operator must complete the parameter values. If the runbook is fully automated, the trigger runs with pre-defined values. The operator will not even notice that the runbook has been executed.

For more information, see [“Create a trigger” on page 105](#).

Monitor runbook status in Netcool/OMNIBus events

The Netcool/OMNIBus events that are mapped to runbooks get updated by Netcool/Impact to indicate this relationship. If the runbook is fully automated, then Netcool/Impact updates the event with the execution status of the fully automated runbook.

Before you begin

Perform the actions listed in [“Map Netcool/OMNIBus events to runbooks” on page 60](#).

Configure the Netcool/OMNIBus Event Viewer to include the RunbookStatus column

Procedure

1. Open the Netcool/OMNIBus Event Viewer.
2. Click the **Edit Views** button.
3. In the Netcool/OMNIBus View Builder, select the view that should include the **RunbookStatus** column.
4. On the **Display Columns** tab, move the **RunbookStatus** item from the **Available fields** list to the **Event list view** list.
5. Adjust the ordering of the columns according to your preferences.

6. Save the changes and close the Netcool/OMNIbus View Builder.
7. Verify that the **RunbookStatus** column is displayed in the Event Viewer.

Values of the RunbookStatus field

The following table defines the values of the RunbookStatus field.

<i>Table 20: Values of the RunbookStatus field</i>	
Value	Condition
<empty value>	Initial state of the event, or no trigger exists that maps this event to a runbook.
Runbook set for manual execution	The event is mapped to a manual runbook, or the related trigger's Run this runbook automatically check box is disabled.
Runbook set for automatic execution	The event is mapped to a fully automated runbook, but no runbook instance has been started yet for this event.
in progress	The event is mapped to a fully automated runbook. A runbook instance has been started.
complete	The event is mapped to a fully automated runbook. A runbook instance has completed successfully.
canceled	The event is mapped to a fully automated runbook. The runbook instance has been created but could not invoke the nested automation.
failed	The event is mapped to a fully automated runbook. The runbook instance has been created but the nested automation has reported a failure.
No status	The event is mapped to a fully automated runbook. The runbook instance has been created but Netcool/Impact has reached its timeout and stopped polling for the status of the runbook instance. Check within the Runbook Automation user interface for details about the runbook instance.

State transitions of the RunbookStatus field

The following flowchart describes in more detail how the values of the RunbookStatus field can change over time.

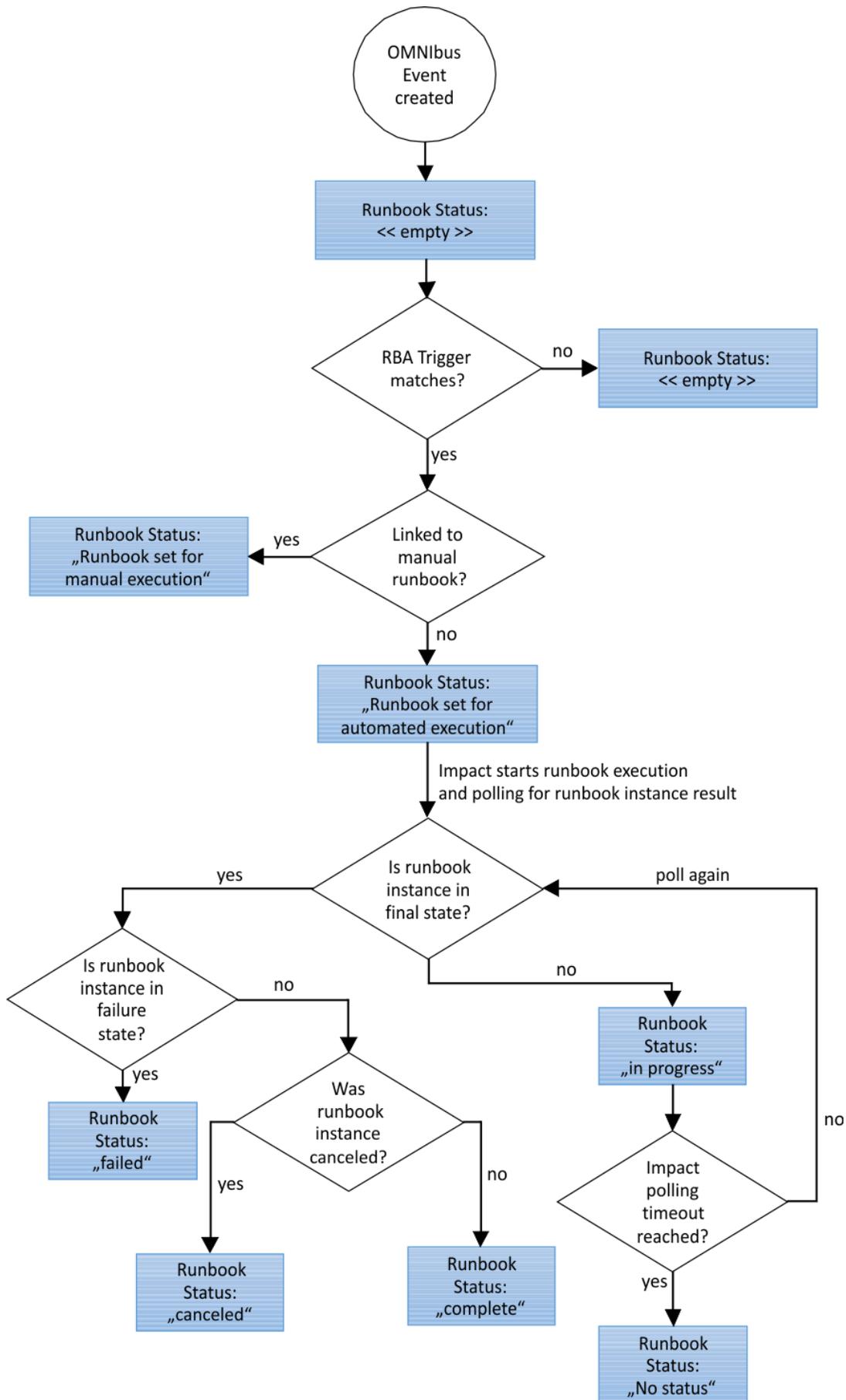


Figure 4: State transitions of the RunbookStatus field

Improving runbook execution

Runbooks can be continually refined and improved. Operators can provide feedback on the quality of runbooks through comments and ratings. More information is provided by success rate, failures, and cancelation of runbooks.

To continuously improve the quality of a runbook, IBM Runbook Automation provides feedback mechanisms for the operations analyst to communicate what needs to be improved. The service delivery engineer monitors the execution of the runbooks and makes improvements where necessary.

Providing feedback

An operations analyst can provide feedback about the quality of the runbooks.

Providing comments

An operations analyst can provide direct feedback after the execution of a runbook. The operations analyst is prompted to provide a rating (five stars for the best rating) and comments. Comments can explain what needs to be improved. For example: "step 3-5 can be automated", "step 4 was too complex", or "could not follow the instructions in step 6", and so on.

Cancellation reason

If an operations analyst cancels a runbook, a dialog opens to select a cancellation reason. This feedback helps to improve the runbooks for the next time they are used. The following cancellation reasons are displayed:

- **Selected wrong runbook:** the wrong runbook was accidentally selected and the user does not want to run the runbook.
- **Runbook seems to have errors:** the runbook cannot be executed due to errors.
- **The runbook solved my problems before I completed all steps:** The problem was resolved by running only some of the steps. It was not necessary to complete all steps.
- **Found the solution and no longer needed the runbook:** the runbook was started but the issue was then resolved.
- **Other**

Monitoring runbook execution

Run History

Select the execution dashboard and you see all recent runbook executions. You can drill down to a specific user, runbook name, automation type, or executions status by using the filters **User**, **Name**, **Type** and **Status**. The Subject Matter Expert (SME) can monitor the ratings of a runbook, who ran the runbook, what the status is, and any comments that were provided. Based on this feedback the SME might contact users of the runbook for further investigation. The SME can then open a new draft and refine and improve the runbook.

Runbook metrics

On the **Library** page, you can find all available runbooks. Select a runbook and click  **History**. On the **History** page, you will find more data about the selected runbook, that is: Description, Type, Rating, Success Rate, Successes, Failures, Cancels, In Progress, Total executions and Average execution time. Additionally the list of all executions of this runbook is shown in a table. This table can be filtered by a time period, for example the last seven days. Click the column status to sort the table by runbook execution status. Now the SME can analyze, for example, why a runbook was canceled so often and how it might be improved.

Improving a runbook

The SME observes the execution of a runbook and gathers information from comments, cancellation reasons, and ratings. Based on this information, a new version is created. The SME now solves the reported issues and adds, for example, automations or corrects steps that did not run correctly.

You can provide feedback if you are an RBA User, RBA Author, or RBA Approver. For more information about roles, see [“Create users and assign roles” on page 10](#).

Export and import runbooks

Export and import runbooks by sending requests to the HTTP API. You can use existing tools to send requests, for example cURL, Postman for Chrome, or RESTClient for Firefox.

Export Runbooks

The following two calls create output, which can be imported again.

Use the following API to export all runbooks:

```
GET api/v1/rba/runbooks?exportFormat=true
```

When enabling the option `exportFormat` via the query parameter, the usual output of the command is replaced by an array of JSON objects. These JSON objects can be used to create the runbooks by using the import API endpoint. The following call is an example:

```
curl -u <APIKeyUsername>:<APIKeyPassword> https://rba.mybluemix.net/api/v1/rba/runbooks?exportFormat=true
```

Replace `<APIKeyUsername>` with your user name and `<APIKeyPassword>` with your password.

Use the following API to export a single runbook:

```
GET api/v1/rba/runbooks/<RunbookId>?exportFormat=true
```

Replace `<RunbookId>` with the ID of the runbook. You can retrieve this ID, for example, from the **Edit Runbook** window. The following call is an example:

```
curl -u <APIKeyUsername>:<APIKeyPassword> https://rba.mybluemix.net/api/v1/rba/runbooks/<RunbookId>?exportFormat=true > exportedRunbook.json
```

Replace `<APIKeyUsername>` with your user name and `<APIKeyPassword>` with your password. Replace `<RunbookId>` with the ID of the runbook.

Import Runbooks

Use the following API to import runbooks:

```
POST api/v1/rba/runbooks/import
```

The following call is an example:

```
curl -H "Content-Type: application/json" -u <APIKeyUsername>:<APIKeyPassword> -d./exportedRunbooks.json https://rba.mybluemix.net/api/v1/rba/runbooks/import
```

Replace `<APIKeyUsername>` with your user name and `<APIKeyPassword>` with your password. Note: `./exportedRunbooks.json` is the output file created from the previous example.

General Notes on export and import

- Exporting runbooks returns the data of referenced automations inline. These automations are created new when the runbook is imported.
- Import actions do not overwrite existing runbooks, automations, or both.
- Exporting multiple runbooks at once will consolidate automations referenced by more than one runbook and only create the minimum necessary amount of new automations.
- Imported runbooks are in draft state and do not retain any kind of version, execution, or approval history. The user performing the import action will be visible as the runbook creator.

For more information about creating an API key, see “API Keys” on page 113.

For more information about the API, see <https://rba.mybluemix.net/swagger/?url=https://rba.mybluemix.net:443/doc-v1>.

Example: complete export and import

The following actions are a sample workflow of a complete export and import cycle. The process outlined here is not the only way to achieve an export and import.

1. The user with MANAGER_ROLE creates an API key. The API key username is `thisismyname`. The API key password is: `thesecretpassword`.
2. The user creates a runbook. The runbook ID is `*0123456789abcdef*`.
3. The user exports the runbook using the following command:

```
curl -u thisismyname:thesecretpassword https://rba.mybluemix.net/api/v1/rba/runbooks?exportFormat=true > exportedRunbook.json
```

Note, the user has exported the output to a file.

4. The user imports the runbook using the following command:

```
curl -H "Content-Type: application/json" -u thisismyname:thesecretpassword -d @./exportedRunbook.json https://rba.mybluemix.net/api/v1/rba/runbooks/import
```

5. The output of the command shows a success and the content of the created runbooks.

Migrating data from RBA Cloud to RBA Private Deployment and vice versa

The import and export runbook function can be used to transfer runbooks that were created on Cloud to a local environment, and vice versa. These functions import and export runbooks and all referenced script automations.

About this task

- If you import more than once, runbooks are duplicated and not replaced.
- If several runbooks refer to the same script automation, the automation will be duplicated during import.
- The described procedures will not migrate other configuration, such as the following:
 - BigFix automations (because fixlets are stored in BigFix)
 - Triggers (because triggers are stored in Impact)
 - Gateway setup (because a gateway is not required on a Private Deployment)
 - Connections to automation providers
 - Connection to Impact
 - Users and user role mappings

Procedure

- **Migrating from IBM Runbook Automation running on Cloud to Private Deployment**
 - a) In IBM Runbook Automation running on cloud, export all runbooks as described in [“Export and import runbooks”](#) on page 64 → *Export runbooks*.
 - b) In IBM Runbook Automation Private Deployment, import the exported runbooks as described in [“Export and import runbooks”](#) on page 64 → *Import runbooks*.
- **Migrating from IBM Runbook Automation Private Deployment to Cloud**
 - a) In IBM Runbook Automation Private Deployment, export all runbooks as described in [“Export and import runbooks”](#) on page 64 → *Export runbooks*.
 - b) In IBM Runbook Automation on Cloud, import the exported runbooks as described in [“Export and import runbooks”](#) on page 64 → *Import runbooks*.

Chapter 5. Library

View the available runbooks, find matching runbook for the events that you need to solve, and run the runbooks. You can also review the runbooks that you have used to date.

Run your first runbook

Note: If you are using IBM Runbook Automation for the first time, it is recommended to load example runbooks and then edit and publish a runbook. For more information, see [“Load and reset example runbooks”](#) on page 68 and [“Edit a runbook”](#) on page 70.

1. Click **Library**.

2. Select a runbook and click  **Preview** to find out what a runbook consists of:

Details

The details section provides descriptive details of the runbook. You can collapse and expand this section. By default, the section is collapsed. Expand the section in order to see the following fields:

- **Description:** shows a description of the runbook.
- **Runbook ID:** ID of the runbook which can be used to identify the runbook using the Runbook Automation HTTP REST Interface.
- **Frequency:** shows the average frequency of running the runbook, for example twice a day.
- **Rating:** shows the average rating that is provided by the user of the runbook. 1 star is the lowest and 5 the highest.
- **Success rate:** indicates how often the runbook completed successfully.
- **Parameter:** Parameter values put runbooks into the context of a particular problem - both in manual and automated steps.

Procedure

The Procedure section displays the runbook steps. For more information about the runbook elements, see [“Edit a runbook”](#) on page 70.

Close the Preview. Open more example runbooks in preview mode to find out how runbooks are described.

3. Select one of the runbooks and click  **Run**.

4. The **Run runbook** page opens, which consists of a sidebar on the left side of the window and a procedure description on the right side of the window. The sidebar can be collapsed and expanded using the  **Details** icon beside the Runbook title. The sidebar contains the following collapsible sections:

- **parameters** – displays a list of the parameters used.
- **commands** – shows the number of used commands and a description of how to use the commands.
- **goto** – shows the number of goto's used within the runbook and a description of how to use the goto elements.
- **automations** – displays a list of the automations used.
- **Details** – displays details such as description, Runbook ID, frequency, rating, and success rate.

5. The **parameters** section on the left of the window is expanded and you are requested to enter parameter values. Click **Read more** to see a description and a reference to where this parameter is used within the runbook. Enter the parameter values, for example host1 for the parameter HOSTNAME and click **Apply and Run**.

Note:

- In some cases, you might not know the parameter value when starting the Runbook. For example, the parameter could be determined by a runbook step during the runbook execution. In this scenario, as soon as the parameter value is known, go back to the parameter section and enter the parameter value. To apply the value click **Update**. The **Update** button is only displayed if the parameter value has been entered.
 - In any step of a runbook, you can select the parameter section of the runbook in the sidebar on the left of the window. You might need to first expand the section if it is collapsed. The parameter can then be changed to a new value. To apply the new value click **Update** (the **Update** button is only displayed if the parameter value has been changed). As soon as a parameter has been used by a Runbook step, it cannot be changed.
6. Follow the instructions in the runbook and proceed to the next step by clicking **Next step**.
 7. If the step contains a command, use the **Copy** button within a command to copy the command to your clipboard. Open a command-line window and paste the command.
 8. If the step contains an automation, click on the automation and select **Run** to run the automation. The output will be displayed in a text box. If an automation produces JSON output, the JSON document is automatically formatted for better readability. Select **Show information** to see description, prerequisites and parameter values.
 9. If you want to document your results of the runbook steps, expand the details section in the sidebar. Enter your comments into the **My personal notes** field.
 10. You can pause or cancel the runbook by clicking **Pause** or **Cancel**. Pause temporarily suspends the execution of the runbook. You can pick up later from where you left off. You can use **Pause** to handle long running automations, for example automations that take longer than 10 minutes. The automations that are in progress continue while you pause the runbook. If you cancel the runbook you are prompted to select a reason. This feedback is helpful to the author of the runbook.
 11. After you completed all steps, click **Complete** at the end of the runbook. Provide a rating, a comment (both optional), and a final assessment of the runbook. You are taken back to the **Library** page.
 12. The **Execution** page provides the **Runbook executions** table. Here you can find all the runbooks that were executed with information such as Status, Start time, Version, Comments, if the runbook is automated, and actions. Filters such as user, runbook status, runbook type and runbook name can be applied. For more information, see [Chapter 6, “Executions,” on page 81](#).

Load and reset example runbooks

IBM Runbook Automation provides ready-to-go sample runbooks. Load the examples to explore what you can do with runbooks.

Before you begin

You can load example runbooks if you have the RBA Author or RBA Manager role. For more information about roles, see [“Create users and assign roles” on page 10](#).

Load examples

Load all pre-defined examples. You can also edit the example runbooks and create new versions.

1. Open the **Library** page.

2. Click  **Load examples** on the upper right of the window (below the **Search** field).
3. The sample runbooks are shown in the **Library** page. The runbooks will show up for Operators if they are published. The samples are marked by the text "Example:"
4. Click **Preview** to view the runbook and see what it does.
5. Click  **Run the steps of this runbook** and run your first runbook. For more information about how to run a runbook, see [“Run your first runbook” on page 67](#).

Reset examples

Follow these steps if you edited the examples and would like to reload the original version:

1. Open the **Library** page.
2. Click **Load Examples** to reload the original version of the example runbook. Any changes that were made will be overwritten.

Convert a sample runbook to a regular runbook

To save a sample runbook as a regular runbook:

1. Open the **Library** page.
2. On the runbook that you want to convert, click  **Edit**.
3. Click **Save as draft > Save As**.
4. Change the runbook name by adding additional text or removing "Example:" from the name. You can optionally update the description.
5. Click **Save**.

After saving, the runbook will no longer be tagged as an example.

For more information about using the editor, see [“Edit a runbook” on page 70](#).

Create a runbook

Document your existing operational procedures using the runbook editor. Share the knowledge and experience that you documented in a runbook with your team members.

Before you begin

You must have user access as an RBA Author or RBA Approver to access the **Library** page.

About this task

Runbooks are created for one of the following reasons:

- Operators document their daily tasks in documents. Runbooks allow you to document and store these procedures in an ordered manner. When a runbook is published the knowledge can be shared between all operators.
- A new set of events has occurred and a couple of skilled operators work out a new procedure to solve the issue that caused these events.

Procedure

1. Click **Library**.
2. Click **New runbook**. The runbook editor opens. For more information about how to use the editor, see [“Edit a runbook” on page 70](#).
3. Provide data to describe the runbook. Make sure that you use a **Name** for the runbook that describes what problem this runbook solves.

4. Provide a **Description**. The description contains information about specific situations that this runbook solves. Operators use the description to decide if a runbook meets their requirements.
5. Add **Tags**. These help to filter a large amount of runbooks, as well as group runbooks with the same content.
6. Start **Inserting step separators**. Click the  icon in the icon palette of the editor. A horizontal line is added to the editor, marked as **Step 1**.
7. Click below the step separator and describe the first step of your procedure. Use  **commands**,  **parameter**,  **automations**, or  **GOTO** elements to distinguish between different types of elements within a runbook. Remember to always create a new step if you start to describe a new task that the operator can run.
8. When all steps have been documented, click **Save draft > Save draft & close**.
9. Test the runbook. Search for your runbook in the **Library** dashboard. Click **Run the steps of this runbook** to test the execution of the new runbook. Modify the runbook if necessary.
10. Publish your runbook.
 - a) Click **Edit** to open the runbook editor.
 - b) Click **Publish**. The runbook is now available to all operators.

Run a runbook

Run all steps that are described in a runbook. When you have completed the runbook you can provide any comments you might have.

To run a runbook you must be assigned the role RBA User, RBA Author, or RBA Approver.

Follow the steps outlined in [“Run your first runbook”](#) on page 67.

Edit a runbook

The runbook editor provides editing capabilities as well as runbook elements that help to document operational procedures. Runbook elements are designed to avoid mistakes.

Open the runbook editor and click **Library > New runbook** to create a new runbook or scroll through the list of runbooks and click **Edit** to open an existing runbook in the editor.

The editor consists of the following areas:

1. Enter descriptive data about the runbook: Name, Description, Runbook ID and Tags.	
2. Editor: Provides text edit actions and runbook elements to describe operational tasks.	3. Parameters: Create parameters that are used in this runbook.
	4. Automations: Add pre-defined automations to the runbook.
	5. Legend for elements: Runbook elements are editing actions that describe operational tasks.
6. Save and Publish the runbook.	

Figure 5: Working areas of the runbook editor

Name

The name identifies the runbook. Try to be as precise as possible and describe the problem that this runbook solves.

Description

Describe the problems that this runbook solves in more detail and sketch out the solution.

Tags

Provide tags to filter runbooks.

Runbook ID

This field is generated by Runbook Automation. The Runbook ID can be used to identify the runbook using the Runbook Automation HTTP REST Interface. Use the copy button to copy the ID to the clipboard.

You can edit a runbook if you are an RBA Author or RBA Approver. For more information about roles, see [“Create users and assign roles”](#) on page 10.

Editing actions

Use the editor to describe your daily tasks step by step. The editor provides editing actions and runbook specific elements to help you write the instructions.

Use common text editing actions to edit and format your operator instructions.

Icon	Editing action
	Undo. You can undo your last 20 editing actions.
	Redo. You can redo your last 20 editing actions.
	Find and Replace. Search for a text string within the runbook editor and replace it with another one. For example: Search for db2 and replace with DB2.

Table 21: Editing actions (continued)

Icon	Editing action
	Size. Change the size of your font. You can start with font size 8 and go up to font size 72.
	Bold. Select a text and make it bold by selecting b .
	Italic. Select a text and make it italic by selecting <i>i</i> .
	Remove Format. If you paste text, the text gets pasted with the original format. If you want to remove the format, select the text with the unwanted format and click Remove Format.
	Left align. Select text that is spread over a couple of rows and click align left. All text rows are aligned left. You can also use this action to place text within one row.
	Align center. Select text that is spread over a couple of rows and click align center. All text rows are aligned center. You can also use this action to place text within one row.
	Right align. Select text that is spread over a couple of rows and click right align. All text rows are aligned right. You can also use this action to place text within one row.
	Align justified. Select text that is spread over a couple of rows and click align justified. All text rows are aligned justified. You can also use this action to place text within one row.
	Numbered list. Select text that is spread over a couple of rows and click numbered list. All lines are indented and numbered.
	Bulleted list. Select text that is spread over a couple of rows and click bulleted list. All lines are indented and bulleted.
	Increase indent. Select text that is spread over a couple of rows and click increase indent. The whole text section is moved to the right.
	Decrease indent. Select text that is spread over a couple of rows and click decrease indent. The whole text section is moved to the left.
	Insert table. If you want to insert a table, click Insert Table . Enter the number of rows and columns that you need and specify the column with and width unit.
	Insert image. If you want to add a screen capture for example, select Insert Image .
	Insert Special Character. If you want to add special characters, select a character from the palette provided.

For a description of the Runbook elements, see [Runbook elements](#).

Adding automations

Automations are used to run several steps automatically. Use the **Automations** page to create automations.

About this task

The job of an operator can involve repetitive tasks. For example, "logon to DB2" or "start server dagobert45". Automations are scripts or IBM BigFix fixlets that document these steps. If you use automations within your runbook then the operator does not have to manually run these steps. They can run all steps with one click.

For more information about how to create automations, see [“Create an automation”](#) on page 83.

The runbook editor displays any created automations in the **Automations** pane.

Only one automation can be added to a runbook step.

It is common practice for automations to contain parameters. When adding automations to a runbook you must decide how the values of the automation parameters are filled by creating a parameter mapping. Select from the following options:

Use a runbook parameter

A runbook parameter, which can be filled via a Trigger from an event or manually by an operator executing the runbook, is used to fill the specific automation parameter. When adding an automation to a runbook, you can select existing runbook parameters or directly create a new runbook parameter.

Define a fixed value

The automation added to the step of a runbook will always be launched with a fixed value for the parameter. (If the same automation is used in a different runbook – another fixed value could be defined there.)

Use the default from the automation

If the automation defines a default value for this parameter, it is possible to use that value.

Use the output of a previous automation

The output of an automation that was added to a previous step can be used as input for the automation of the current step.

Use the logged in user

The parameter value will be filled with the username of the user, who is logged in to Runbook Automation at the time when the runbook is run.

Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it will be removed from the username.

Procedure

1. Search for the automation. Type in the search term that describes the automation that you need.
2. If your search results in several hits that look similar, click  to view the automation in more detail.
3. Add the automation to your runbook. Add a step to your runbook and click  next to the automation name in the Automation pane. You can also preview the details of an automation and then click **Add to runbook** within the preview dialog. The automation is added at your cursor position. If the cursor focus is not within the editor window, a new step is created at the beginning of the runbook with the automation as a step description.
4. Select the parameter mapping type. A window opens with automation details and the parameter definitions. In the **Mapping** column, select how automation parameters are to be filled with values. You can choose from the following options:

New runbook parameter

Add a parameter. If default values are provided, update the default values as required. Enter a new parameter name and description.

Fixed value

An entry field is provided to enter the parameter value.

(Existing) runbook parameter

Select the parameter from the list of existing parameters.

Use default

Use the default value set by the automation.

Automation output

Choose the automation from a previous runbook step. The output value of the automation will be used as the parameter value for the current automation. This option is only available if the runbook contains a previous step with an automation.

Use logged in user

Select this option to fill the parameter value with the username of the user who is logged in to Runbook Automation at the time when the runbook is run.

Note: If the username contains the @ symbol, for example because the username is an email address, the @ symbol and all characters that follow it will be removed from the username.

What to do next

Edit parameter configuration

If you want to edit the parameter configuration of your automation, select the automation and click **Configure**. Change your settings and click **Apply**.

Automation with errors

If the automation is decorated with an error symbol, then the parameter settings are not correct. Select the automation and click **Configure** to correct the parameter settings.

Delete an automation

If you want to remove the automation, select the automation and press **delete** on your keypad or select the automation and click **Remove**.

Adding parameters

Parameters are used as general placeholders for values that are used on a frequent basis. Examples of parameters are DATABASE_NAME, DIRECTORY, HOSTNAME, URL and so on.

Before you begin

You can use runbook parameters in the following scenarios:

- Parameters can be used as variables that get substituted by a value in the text of a runbook step, for example:

```
STEP 2
In this step you determine the longest running process of system $SYSTEM
```

- As input to a command defined in a runbook, for example:

```
STEP 5
Now issue the command: cd $DIRECTORY
```

- Parameters of an automation can be filled with the value of a parameter defined for the runbook.
- Automations of type BigFix and Script have a system parameter `target` which defines the target system where the BigFix Fixlet or script is run.
- Automations of type Script have a system parameter `user` that specifies the UNIX username which is used to run the script on the target system. This parameter can be mapped automatically to the user who is currently logged in to Runbook Automation. For more information, see [“Adding automations” on page 72](#).

About this task

Operators enter values such as `username`, `password`, `hostname`, `database name` or `URL` on a frequent basis. If you have to deal with many different computer systems, you must remember numerous user names, passwords, host names or database names. In order to avoid typos or having to remember values, you can define parameters. For example, you can define a parameter `DATABASE_NAME` and either provide a default value or enter the value when you run the runbook. Default values are the best fit if you do not want operators to remember and enter the value.

Parameters are local to the runbook. They are not available for other runbooks.

Parameters are used within runbooks and for automations. Within runbooks, you can use parameters either standalone, for example `URL`, or within a command, for example `connect to HOSTNAME`.

Parameters must be filled with values when executing a runbook. This can be done in the following ways:

- When a runbook is launched by clicking the execute button from the Runbook Automation UI, an operator must first enter the parameter values. The operator is prompted with the name and description of all parameters defined for the runbook, as well as an optional default value.
- When a runbook is launched from an event, a Trigger defines which values of the event data are mapped to which parameter of the runbook. Values that are entered in this way cannot be changed.

Runbook parameter values can be changed during the execution of a runbook. In any step of a runbook, you can select the parameter section of the runbook in the sidebar on the left. You might need to expand the section first, if it is collapsed. The parameter can then be changed to a new value. To apply the new value click **Update** (the Update button is only displayed if the parameter value has been changed). Steps of the runbook that are not yet executed will reflect the new value.

Runbook parameters can be used to define parameter values for automations within the same runbook. Parameter values for automations can be filled using the result of a previously executed automation.

In this way, the parameter Automation output of a previous automation `Find large file systems` is an automatic parameter available for automations in the current step. It is possible to use the output of any automation that ran before in this runbook.

Procedure

1. Create a parameter. Click  **Add Parameter**. A dialog box opens where you can enter the parameter name, the description, and the default value. If you select **Not mandatory on start** you can start the runbook execution without defining a value, and set the parameter value during runbook execution. This setting is useful if the parameter value is not available at the beginning of the execution, but during execution.

When you enter the runbook parameter name you can use alphanumeric characters, characters from national languages, and some special characters. But do not use the special characters ampersand (&), backslash (\), space (), greater than (>) or less than (<) when you define runbook parameter names.

As a result, a parameter of the runbook is created and can be used as a variable in the text of a step, as a variable value of a command, or as input of an automation added to a runbook step.

2. Use the parameter. Drag and drop a parameter from the parameter panel to the canvas. For more information about how to use parameters, see [Runbook elements](#).

Runbook parameters can be dragged and dropped to a location in the text of a step, or to replace a part of a command that must be executed by an operator.

Parameters of a runbook that are used by an automation must be changed.

3. Edit a parameter. Click the edit icon next to the parameter that you want to edit.
4. To edit a parameter used by an automation, click on the automation you have added to a runbook step and then click **Configure**.

Adding a GOTO element

The GOTO element is used to skip steps or jump to the end of the runbook.

Use the GOTO element in the following scenarios:

The execution was successful and you can complete the runbook

For example:

1. Restart the Application Server.
2. If this was successful, then go to the **END**.
3. Check your log files.

Run optional steps as result of a check

For example:

1. Log in to the server where you want to start the application server.

2. Check the current CPU consumption. If it is below 80% GOTO **Step 5**.
3. Find the process consuming the most CPU.
4. Determine the process.
5. Start the application server.
6. Check the log file for confirmation that the application server is started.

Configuring runbook creation

Use these procedures to define the runbook creation process and control the publication of runbooks.

Configuring runbook approval

Define whether to use the approval workflow or the direct publish workflow to make runbooks available for execution by operators.

Before you begin

You must be assigned the manager role to configure the runbook creation process.

Procedure

1. Navigate to **Library**.
2. Expand the drop-down menu .
3. Click  **Settings** next to the search field.
4. Select **Approve runbooks before publish** if you want runbooks to be approved before they are published.
5. If you want to enforce that an approval must be refreshed at a regular interval, select **Set expiration date after xxx days** and choose a time interval. If this option is enabled, the runbook approval will expire after the specified number of days and can no longer be executed until the runbook is reviewed, submitted for approval, and approved again.

The deactivated approval process is described in [“Create a runbook” on page 69](#).

The activated approval process is described in [“Runbook approval process” on page 76](#).

Note: If the approval process is deactivated, all pending approvals are deleted.

Runbook approval process

The Runbook approval process offers control over the publication of runbooks. This is beneficial if a mandatory review process is necessary, if prematurely released runbooks could be harmful, or when auditable information on the publication is required.

About this task

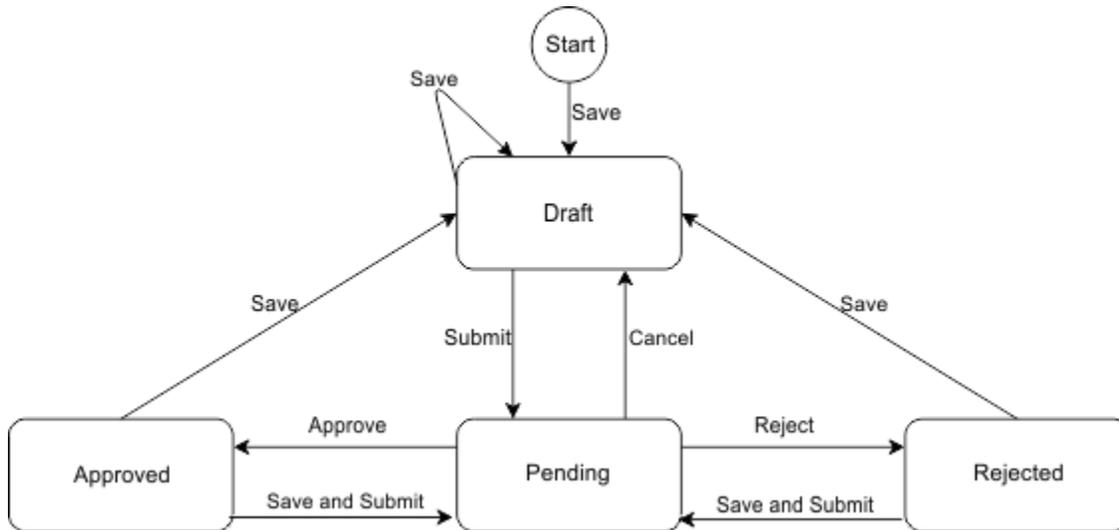


Figure 6: Runbook approval process

Figure 1 illustrates the states and transitions of the runbook approval process. If the approval process is enabled, it is not possible to directly publish a runbook. Instead an option to submit the runbook is displayed. The following options are then available:

Procedure

- **List all runbooks that must be approved**

- Open the **Library** page.
- From the **Status** filter drop-down menu, select **Pending approval**.
- All runbooks for which approval is pending are shown.

- **List all runbooks that are approved**

- Open the **Library** page.
- From the **Status** filter drop-down menu, select **Approved**.
- All approved runbooks are shown.

- **Note:** you can also filter for a status of **Approval expired** and **Approval rejected**.

- **Submit a runbook draft for approval**

The roles that can perform this action are Author, Approver, and Manager.

- Create or edit a runbook and click **Submit for approval**.
- Enter the name of the approver in the field provided and click **Send**.

The runbook is now in a pending state. It can no longer be edited, except for changes to its state.

- **Cancel a pending runbook approval**

The roles that can perform this action are Author, Approver, and Manager.

- Locate the runbook on the **Library** page.
- Click **Approve or reject this runbook**.
- Click **Remove approval**.

The runbook is now in normal draft mode and can be edited. The assignee can no longer approve or reject the runbook.

- **Approve a runbook**

Only the assigned person from the submission can approve a runbook. The roles that can perform this action are Approver and Manager.

- a) Locate the runbook on the **Library** page.
- b) Click **Approve or reject this runbook**.
- c) Click **Approve**.

The runbook now has a status of approved, which is the default for execution. New drafts can be created. Information pertaining to the approval assignee and timestamp is stored within the runbook.

- **Reject a runbook**

Only the assigned person from the submission can reject a runbook. The roles that can perform this action are Approver and Manager.

- a) Locate the runbook on the **Library** page.
- b) Click **Approve or reject this runbook**.
- c) Click **Reject**.

The runbook now has a status of rejected and can be edited again. Information pertaining to the approval assignee and timestamp is stored within the runbook until a new draft version is created and the old information is deleted.

Runbook versions

You can use versioning to create incremental improvements to runbooks and monitor the success of changes. Subject Matter Experts can continuously improve the quality of runbooks by using the comments provided and available run metrics. As a result different versions of a runbook will be created.

All available runbooks are listed on the **Library** page. Click the arrow at the beginning of the row to display all the versions of a runbook.

Draft version

The Draft version is a work-in-progress runbook. It is marked with a draft flag. A draft runbook is not yet published and does not run in production. It is not visible to any operator. You can run and test the current draft.

Latest published version

The latest published version is the version that is running and used by all operators. This is the version that the operations analyst can preview and work with.

Archived version

Archived versions are previously used versions. They are not in production and cannot be used by any operations analyst. The metrics of the previous versions help the Subject Matter Expert to monitor continuous improvements of the runbook. You can run archived version to learn why this runbook did not run successfully.

You can create different versions of a runbook if you are an RBA Author or RBA Approver. For more information about roles, see [“Create users and assign roles” on page 10](#).

Save and publish a runbook

Creating a runbook involves documenting complex steps. These steps must be reviewed and tested before you can publish the runbook for use in production.

After a runbook is published and used in production, you might improve it based on the feedback that you receive from comments, success rate, and ratings. The runbook editor provides the following actions:

Cancel

Cancel closes the runbook editor without saving any changes.

Save as

Use this action to save a copy of the runbook using a different runbook name.

Save draft

Save draft saves the updates that are added. The editor remains open. **Save draft** is useful if you are editing a runbook and you want to take a break and continue later. Or you just want to save your changes as you go.

Save draft & close

Save draft & close saves your changes and closes the editor. You use this option if you need to work on a different item.

Publish

This button is only available if the approval process has been deactivated. You can publish a runbook if you are an RBA Approver. For more information about roles, see [“Create users and assign roles” on page 10](#). After a runbook has been published it can be used by all operators in a production environment.

Submit for Approval

This button is only available if the approval process is enabled.

If the approval process has been activated, see [“Runbook approval process” on page 76](#) for more information about approval before publication.

Filtering runbooks

Filter by status, groups or tags to easily locate runbooks.

Procedure

- Click **Library**.
- To filter for runbook status, select one or more items in the **Status** filter. The filter matches any selected status (or condition).

Note: Filtering for status **Draft** will display runbooks that have never been published. Draft versions of already published runbooks won't be displayed.

- To filter for groups, select one or more items in the **Group** filter. The filter matches any selected group (or condition).
- To filter for tags, select one or more items in the **Tag** filter. All tags must match to apply the filter (and condition).
- If a combination of several filters are used, all filters must match to apply the filter (and condition).
- Additionally, you can filter by runbook name using the **Search** field.

Results

Only runbooks that match the specified filter are displayed.

Delete a runbook

Users assigned the RBA Manager role can delete a runbook.

To delete a runbook:

1. Open the **Library** page.
2. Use the check boxes to select one or more runbooks.
3. An action bar is displayed showing the available actions and the number of runbooks selected.

4. On the action bar, click  **Delete**. If you delete a runbook, all versions and the execution history of the runbook are deleted as well.

For more information about roles, see [“Create users and assign roles” on page 10](#).

Chapter 6. Executions

See recently executed runbooks, including runbooks in progress.

On the **Execution** page, you can find all runbooks that have been started. You can filter your list by the following criteria:

User who has executed the runbook

Select the **User** filter and select one or more users. All runbook executions for the selected users are shown. For example, if you want to see only the runbooks that you have executed, select your user name.

Runbook name

Select the **Name** filter and select one or more runbook names. All runbook executions for the selected runbook names are shown.

Runbook execution status

Select the **Status** filter and choose a status such as Success, Failed, Cancelled, In progress, or Completed. For example, you can find all runbook executions that are paused by selecting **In progress**. When you are ready to return to the task you can resume the runbook. Select status **Canceled** if you want to see all canceled runbooks and the reason why they have been canceled.

Runbook type

Select the filter **Type** and select type **Manual** or **Triggered automatically**. Fully automated runbooks are runbooks in which each step contains an automation. An operator does not need to interact with the runbook. Automated runbooks can be mapped with events by creating a trigger. To see the automated runbooks that are started by a trigger, click **Triggered Automatically**.

For more information about how to create automations, see [Chapter 7, “Automations,”](#) on page 83.

For more information about how to create a trigger, see [Chapter 9, “Triggers,”](#) on page 105.

Monitor runbook history

You can monitor the runbook history if you are an RBA Author or RBA Approver.

1. Click **Library**.
2. Select a runbook.
3. Click  **History**.

The execution history table and runbook details such as description, type, rating, success rate, and execution statistics are displayed.

4. Use the filter box on the upper right corner to display only specific runbooks, for example:
 - a. To see runbook executions of the last seven days only, select **Last 7 days**.
 - b. To see runbook executions that are still in progress, select **In progress**. If you want to resume a runbook execution, click **Resume**.
- 5.

For more information about roles, see [“Create users and assign roles”](#) on page 10.

Chapter 7. Automations

Create an automation to summarize and automate several steps into a single step.

In runbooks, an automation is the collection of several manual actions into a single automated entity. Automations use the parameters of the runbook. Customize the parameters for the execution of the runbook. Parameters lower the time needed for the execution of a runbook. Automations eliminate the risk of manual errors that you get repeating the same steps many times.

The Service Delivery Engineer can provide an automation to replace frequently used steps with a single click. As the automation needs a target system where it is processed, an existing connection to the local environment is required before automations can be added. For more information about how to set up a connection, see [“Create a connection” on page 89](#).

Create an automation

Create an automation by defining the automation type and configuring the parameters and fields.

Automations are routines that run a task automatically. There are different types of automations depending on how tasks are run. Runbook Automation supports automations of type script, BigFix and HTTP. For automations of type script and BigFix a connection needs to be configured. To learn how to configure a connection, see [“Create a connection” on page 89](#).

You must be logged on as an RBA Author to create automations.

Create an automation of type script

You can create an automation of type script by using the **SSH provider**. The SSH provider can execute a script on an endpoint system in any scripting language the target system can interpret. The SSH provider does not require an additional agent as it uses a direct remote execution connection via ssh.

You must install the IBM Secure Gateway if you are using Runbook Automation in the cloud. The IBM Secure Gateway is not required for a Private Deployment of Runbook Automation.

Complete the following steps to enable script automations:

1. [“Install the IBM Secure Gateway client” on page 92](#).

Private Deployment You do not need to install the IBM Secure Gateway client in a private deployment.

2. Create a connection, see [Chapter 8, “Connections,” on page 89](#).
3. Create an automation of type script, see [“Create an automation” on page 83](#).
4. Add the automation to the runbook, see [“Adding automations” on page 72](#).
5. [“Run a runbook” on page 70](#).

Click **Automations > New automation**. Complete the following fields:

Script

If you want to run a script as an automation, you must configure a Script Automation Provider on the **Connections** page.

Name

Provide a name that describes what this automation does. For example, `Find large file systems`.

Description

Provide any helpful additional information so that the user can immediately understand which problem this automation solves.

Prerequisites

If this automation requires prerequisites, add this information. For example, DB2 Version 10.5 or higher.

Script

You can either select **Import** to select the script from your file system, or you can directly enter the script into the editor. The script is run by using the default shell or the interpreter, which is specified after the shebang `#!`.

Parameters

Add input parameters to run the automation script. Those input parameters are available to the script as environment variables. For example, a parameter `filesystem` is used in a Linux script as `$filesystem`. The `$` sign is automatically added to distinguish user-defined parameters from system parameters such as `target` and `user`. The parameter is exported without the dollar sign. **Note:** input parameter names must consist of alphanumeric characters and the underscore character only. The following system parameters exist:

target

Mandatory parameter. Applies to automations of type BigFix and Script. The `target` parameter is created to define the target machine where the Bigfix Fixlet or script is running.

user

Mandatory parameter. Applies to automations of type Script. The `user` parameter defines the UNIX username which is used to run the script on the target machine.

Note: If the username contains the `@` symbol, for example because the username is an email address, the `@` symbol and all characters that follow it will be removed from the username.

Single and Multi Target Automations

You can use single target or multi target automations to execute script automations.

Single Target

The parameter with the name **target** has special meaning to script automations as it defines the system on which the automation will run. The system can be defined by a short hostname, a FQDN, or an IP address. When executed a script automation will connect to the system identified by the **target** and execute the script there.

The execution result of the script will be reflected in the status of the automation. Possible automation states are **executing**, **successful**, **unsuccessful**, **failed**, and **unknown**. In the case of a fully automated runbook execution, this status will also be used to decide whether the runbook will be canceled or continued.

Examples:

- An automation which receives **prod-server1**, as the content of the target variable will execute this automation on the system **prod-server1**. The script exited with return code of zero and the state of the automation instance is set to **successful**.
- An automation which receives **192.168.55.56**, as the content of the target variable will execute this automation on the system with the matching IPv4 address. The script exited with a non-zero return code and the state of the automation instance is set to **unsuccessful**.
- An automation which ran in step 1 of a fully automated runbook failed to reach the endpoint (for example because the server was not available) and the automation could not be executed at all. The status of the automation instance is **failed**. This status is reflected in the status of the execution. The execution will stop after the first step and report a failure.

Multi Target Automations

The other mode of operation of an automation is the **Multi Target Automation** (MTA). In order to execute an automation as an MTA, specify the target string in the following format: **[\$target1, \$target2, ..., \$targetN]**.

The following rules apply to MTAs:

- The target string must begin with a left square bracket ([) and must end with a right square bracket (]).
- Between the square bracket define a list of systems separated by comma (,).
- Duplicated entries are detected and ignored. This means specifying the same target multiple times will have no effect.
- An empty list is also allowed.

Multi Target Automations allow you to execute the same automation on any number of targets in parallel, removing the necessity to execute the same automation with different targets sequentially. If the target is specified in such a way, how the automation is executed and how the results are treated will be different. The following criteria apply to MTAs:

1. For every entry in the comma-separated list, the script will be executed on the specified target. **Note:** you cannot specify the same target more than once.
2. All actions and information are summarized in one log. This log follows a specific format, see [“Output format of Multi Target Automations”](#) on page 85.
3. The status of the execution will be **executing** as long as at least one execution is still ongoing and **successful** once all script processes are finished. The result will be **successful** even if some or even all automations reported **unsuccessful, failed, or unknown**.
4. Specifying an empty array is allowed. This special case is called a **Conditional Automation**. No actual execution occurs, but an execution record is still created. The record will always report **successful** as its state. If the runbook is executed fully automated, the runbook instance will proceed to the next step.

Examples:

- An automation receives **[prod-server1]** as the target. It will execute the script on the system **prod-server1**. The script exited with a non-zero return code. The status of the execution is **successful**. The output of the automation instance contains information that this script execution was **unsuccessful on prod-server1**.
- An automation receives **[prod-server1, prod-server2, prod-server3]** as the target. It will execute the script on three systems in parallel. All scripts exit with zero as the return code. The output and status of each of the three script executions is written to the output. The status of the overall execution is **successful**.
- An automation receives **[]** as the target. This triggers the conditional automation workflow. No execution will take place. The execution record will have an almost empty output and the status will be **successful**.

Output format of Multi Target Automations

Despite being executed on any number of endpoints, the results of Multi Target Automations (MTA) are consolidated into one execution record. This record combines the information from all executions.

The output is formatted as follows:

- If the script produces output the output is displayed after two prefixes. The first item is the name of the target. The second item is either **out** or **err** depending on whether the output was written to `stdout` or `stderr`.
- The log will group the output by target. So the complete output of the first target will be displayed before the output of the second target is displayed and so on. This means the output is not ordered chronologically, although the output by target is.
- At the end of the output a summary is always present. The summary consists of three additional lines. They follow the pattern **\$status : [\$listOfTargetsWithThisState]**. Where **\$status** is one of the following: **unsuccessful, successful, or failed**.
- If the target array is an empty array, the output will only contain the summary.

Example

An automation has been executed with the target **[prod-server1, prod-server2]**. The execution on **prod-server1** was successful, while the execution on **prod-server2** was unsuccessful. The output will look as follows:

```
prod-server1 out: Some output from the script to stdout happening at 08:34
prod-server1 out: More output from the script to stdout happening at 08:36
prod-server1 out: Script prints more to stdout
prod-server1 status: successful

prod-server2 out: Some output from the script happening at 08:35
prod-server2 out: More output from the script happening at 08:37
prod-server2 err: Script prints output concerning a failure to err
prod-server2 status: unsuccessful

unsuccessful: [prod-server2]
successful: [prod-server1]
failed: []
```

Comparison of Single and Multi Target Automations

The following table outlines the difference between the operation modes:

Aspect	Single Target	Multi Target Automation
Number of parallel executions	One	Number of entries in target array
Final execution status	Equal to status of automation	Always successful
Failure during fully automated runbook (FARB) execution	Aborts the FARB	Proceeds the FARB
Specifying an empty target	Failure to execute automation	Enters special case for empty array

Create an automation of type BigFix

If you want to use an automation of type BigFix, install the required components to run the automation.

You must complete the following steps before you can use an automation of type BigFix:

1. [“Install the IBM Secure Gateway client” on page 92.](#)
Private Deployment You do not need to install the IBM Secure Gateway client in a private deployment.
2. Create a connection: [Chapter 8, “Connections,” on page 89.](#)
3. Create an automation of type BigFix: [“Create an automation” on page 83.](#) You need to know site ID, fixlet/task ID, and action ID.
4. Add the automation to the runbook: [“Adding automations” on page 72](#)
5. [“Run a runbook” on page 70](#)

Select **Automations** and click **New automation**. A new page is displayed. Complete the following fields:

IBM BigFix

If you want to run an IBM BigFix fixlet or task as an automation, you must configure an IBM BigFix Automation Provider on the **Connections** page.

Name

Provide a name that describes what this automation does. For example, `Find large file systems`.

Description

Provide any helpful additional information so that the user can immediately understand which problem this automation solves.

Prerequisites

If this automation requires prerequisites, add this information. For example, DB2 Version 10.5 or higher.

Site Name

The **Site Name** is the location where the fixlet or task runs. For example, BES Support.

Fixlet ID

Provide the numerical ID of the fixlet or task. For example, 168.

Action

Provide the action that you want to run within your fixlet or task. For Example, Action1.

Target machine

Mandatory field. The **Target machine** parameter is created to define the target machine where the fixlet is running. Each automation must define a target machine.

Parameters

Add input parameters to run the IBM BigFix action. Those input parameters are available to the fixlet as environment variables. For example, a parameter `filesystem` is used in a Linux fixlet as `$filesystem`. The \$ sign is automatically added to distinguish a user-defined parameter from a system parameters such as TARGET. The parameter is exported without the dollar sign. For Windows fixlets, parameters are available in the fixlet with the percentage sign, for example `%filesystem%`.

Create an automation of type HTTP

The HTTP automation provider allows to send HTTP requests to a specified web service.

Cloud The specified web service must be located in the public internet. It is not possible to connect to your private data center.

Private Deployment The specified web service can be in your private data center or in the public internet provided the RBA server can reach the destination. Ensure that the network setup, for example firewalls can reach the API endpoint of the web service.

Click **Automations** > **New automation**. Complete the following fields:

Type

Select HTTP.

Name

Provide a name that describes what this automation does. For example, IBM Watson Translate.

Description

Provide any helpful additional information so that the user can immediately understand which problem this automation solves.

Prerequisites

If this automation requires prerequisites, add this information. For example, Watson service credentials are required.

API endpoint

Specify the http API endpoint of the web service to which the HTTP request will be sent to. For example `https://gateway.watsonplatform.net/qagw/service/v1/question/1?TestIt`.

METHOD

Choose the HTTP method. For example POST.

Username

If basic authentication is required to use the web service, specify the API user name for basic authentication.

Password

If basic authentication is required to use the web service, specify the API password for basic authentication.

Accept

Specify the accept request header. The accept header is used to specify the media types which are acceptable for the response of the request. For example `text/html`.

Accept-Language

Specify the accept-language request header which is used to restrict the set of languages that are preferred for the response of the request. For example `en-US`.

Additional headers

Optionally, specify any additional request headers that are needed for the request. For example `accept-charset: utf-8`.

Ignore certificate errors

Select this check box to ignore certificate errors. Use this option only for test purposes. In production environments, ensure that the correct certificates are installed on the target web service.

Parameters

Add input parameters to run the automation. Those input parameters can be referred in the entry fields. For example, a parameter `text` can be referred as `$text`.

Manage existing automations

As Service Delivery Engineer, you are interested in how the automations run and how to keep them running smoothly. Use the **Automations** page to view statistics and administer the available automations.

The following types of runbook contain automations:

Semi-automated runbooks

Runbooks that contain one or more automations. These runbooks consist of manual and automated steps.

Fully automated runbooks

Runbooks that contain an automation in each step. If fully automated runbooks are started by a trigger, you can find their status on the **Execution page**. Select the filter **Type** and select **Triggered automatically**.

On the **Automations** page, you can find all available automations.

Name / Description

Name and description of the automation.

Automation type

Automations are either of the type script or IBM BigFix, depending on the connection type used.

Success rate

How reliably did this automation run so far? The success rate indicates how well an automation ran.

Actions

As a Subject Matter Expert you can  **Edit**,  **Preview**, and  **Delete** an automation.

Automation metrics

As a Subject Matter Expert, you are interested how the automations run. What is their success rate, how often are they used and for how long on average did they run? Click the arrow at the end of the row to see more metrics about the automation. The number of invocations and the average, minimum, and maximum execution time are displayed.

Chapter 8. Connections

For semi-automated and fully-automated runbooks, which use automations of type script or type BigFix, a connection must be set up to connect to your target endpoints. To trigger runbook executions from incoming events, set up a trigger connection. You can add, edit, or delete the connection for each automation provider or the event trigger provider.

You can perform the following actions on each connection tile in the **Connections** page:

Configure

If you setting up your environment for the first time, or you want to use another automation provider, you can create a new connection. For more information about how to create a connection, see [“Create a connection”](#) on page 89.

Edit

If your settings changed, for example your username or password, you can edit your connection.

Select  **Edit** to open the configuration dialog and apply your changes. You can edit a connection if you are an RBA Author or RBA Approver.

Check connection status

Navigate to the **Connections** page to check if the connection is working or if it has failed.

Delete

If you no longer need a connection, click  **Delete** to delete the connection. You can delete a connection if you are an RBA Manager.

For more information about RBA roles, see [“Create users and assign roles”](#) on page 10.

Create a connection

IBM Runbook Automation connects to your target endpoints, for example your on-premise back-end system. If you use automations of type script or BigFix, you must create a Script/BigFix connection to access your target endpoints. If you want to use a trigger, connect to Netcool/Impact also referred to as the event trigger provider.

You can configure a connection on the **Connections** page. To create a new connection, click **Configure** on

the corresponding connection tile. To edit an existing connection, click  **Edit** on the corresponding connection tile. You can configure one connection of each type. For example, you can configure one Script automation provider connection. Follow the on-screen instructions in the **New connection** window.

You can create a new connection if you are an RBA Manager. For more information about roles, see [“Create users and assign roles”](#) on page 10.

Event Trigger (Netcool/Impact)

Create an Event Trigger connection to map runbooks with Netcool/Impact events.

Cloud Event Management Cloud Event Management does not support the Event Trigger Integration described in this section. Refer to the Cloud Event Management documentation for integration options.

You can configure a connection on the **Connections** page. Click **Configure** on the Event Trigger tile to open the configuration window. You can configure one Event Trigger connection. Follow the on-screen instructions in the **New Connection** window.

1. Install IBM Netcool/Impact fix pack V7.1.0.13 or higher

Check that the fix pack level of your current Netcool/Impact installation is V7.1.0.13 or higher. A Netcool/Impact license is required.

2. Configure IBM Netcool/Impact

For more information about how to configure IBM Netcool/Impact, see [“Installing Netcool/Impact to run the trigger service”](#) on page 94.

Cloud 3. If you are setting up your first connection, download and install a gateway client

To establish a connection to your local environment a gateway client must be installed. Click **Connections > Configure/Edit Event Trigger > Download gateway client** to open the IBM Secure Gateway management interface. For more information, see [“Install the IBM Secure Gateway client”](#) on page 92.

4. Open the whitelist of your gateway client and add the Netcool/Impact URL

For more information, see [“Configure the IBM Secure Gateway client”](#) on page 93.

5. Enter IBM Netcool/Impact access information

Enter the URL and credentials for the IBM Netcool/Impact connection. You can specify the certificate of the IBM Netcool/Impact server for additional security. For more details on the connection dialog and the connection parameters, see [“Configure a connection to the Netcool/Impact server”](#) on page 101.

IBM BigFix automation provider

IBM BigFix automation provider connects to your back-end system through your local IBM BigFix setup.

You can configure a connection on the **Connections** page. Click  **Edit** on the BigFix tile to open the configuration window.

You can configure only one IBM BigFix connection. Follow the step-by-step instructions in the **New connection** window.

1. Enter the IBM BigFix access information

It is a requirement that you already have IBM BigFix and Web Reports installed. Check with your system administrator to obtain access information.

Enter the URL in the following format:

```
https://<DomainnameOfBigFixServer>:52311/api
```

You can specify the certificate of the IBM BigFix server for additional security.

Cloud 2. If you are setting up your first connection, download and install a gateway client

To establish a connection to your local environment a gateway client must be installed. Click

Connections >  Configure/Edit BigFix > Download gateway client to open the IBM Secure Gateway management interface. For more information, see [“Install the IBM Secure Gateway client”](#) on page 92.

3. Open the whitelist of your gateway client and add the Netcool/Impact URL

For more information, see [“Configure the IBM Secure Gateway client”](#) on page 93.

SSH script automation provider

Use a script automation provider to connect to your back-end system (targets). The SSH Provider is agentless and connects directly to the target machine. It authenticates using public key-based authentication in SSH.

The script automation provider works for back-end systems (targets) and provides the following standard UNIX features. The user executing this automation must have sufficient rights to execute these features.

- `bash` – a shell that is used to wrap and execute the specified commands / script.
- `mktemp` – A utility that is used to create a temporary file, which is required for the script execution with this automation provider to work.
- `openssl` – A utility that is used on the target system to decrypt the transferred commands / script.

Defining which RBA user is allowed to run an automation

The current public key must be added to all target machines that you plan to execute scripts on via the SSH Provider. Make sure that you add the public key in the correct repository, so that the script can be executed:

- By the root user.
- Or by any user on this target.
- By specific user(s) on this target. For example, by putting the key in the `authorized_keys` file of home directory of this/these specific user(s).

Cloud Depending on the public key used, any RBA user or only members of specific RBA groups will be able to access the given target system. See step 5 in the procedure below for more information about creating public keys for specific groups.

Defining which UNIX user is used to run an automation

By default, scripts are executed on the target machine using the UNIX root user name. It is possible to run the script with a different UNIX user. The user name can either be fixed or depend on the RBA user that is currently logged in. For more information, see [“Create an automation of type script” on page 83](#).

Defining an SSH jumpserver

An optional SSH jumpserver can be specified. If chosen, any connections to target systems will be routed through this jumpserver. See step 3 for more information about using a jumpserver.

Before you begin

Private Deployment The IBM Secure Gateway client is not required for a Private Deployment installation.

About this task

You can configure a connection on the **Connections** page. Click  **Edit** on the Script tile to open the configuration window and follow the on-screen instructions.

Procedure

1. **Cloud** If this is your first connection, download and install a gateway client from the Secure Gateway Dashboard.

A gateway client must be installed to establish a connection to your local environment. Click **Secure Gateway Dashboard** to open the IBM Secure Gateway management interface. For more information, see [“Install the IBM Secure Gateway client” on page 92](#).

2. **Cloud** Open the access control list of your gateway client and enter the target endpoints.

The target machines that you intend to execute scripts on via the SSH Provider must be added to the gateway clients access control list. For more information, see [“Configure the IBM Secure Gateway client” on page 93](#).

3. If you are using a jumpserver you must configure it.

Depending on your environment, you might require a jumpserver to access your target endpoints. A jumpserver is an SSH endpoint that is used to connect to the nested SSH endpoints. This is a common approach used to communicate between different network zones. To use a jumpserver with RBA it must have an SSH server running and the `nc` command must be available. This is used to connect to nested SSH target endpoints.

Click **Use a jumpserver** and specify the following jumpserver credentials:

Jumpserver address

The hostname or IP address of the jumpserver.

Jumpserver port

The SSH port of the jumpserver.

Jumpserver username

The username for authentication on the jumpserver.

Jumpserver password

The password for authentication on the jumpserver.

Any connections to SSH target endpoints will use the specified jumpserver.

Cloud If you are using the secure gateway client when a jumpserver is specified, the connection between the secure gateway client and the target endpoint will use the jumpserver.

4. On your target machine, register the default public key to enable access to the target endpoints via SSH for all users.

Configuring SSH public key authentication for the UNIX root user

The displayed public key must be added to all target machines that you plan to execute scripts on via the SSH Provider. This key enables any RBA user to run script automations on the given target endpoint. The key must be added to the `authorized_keys` file which is usually found in the `/root/.ssh/authorized_keys` folder.

Configuring SSH public key authentication for a specific UNIX user

If you want to enforce that only a specific UNIX user can execute the script on this target endpoint you should copy the key to the `authorized_keys` file in the home directory of the specific user, for example `/home/john/.ssh/authorized_keys`.

You can choose to regenerate the public key by clicking the refresh button in the upper right corner of the public key.

Note: Regenerating the public key will delete the old key pair. If you choose to regenerate the key pair you must exchange the public key in each target machine that you plan to access via the SSH Provider.

For more information about how to configure which UNIX user is used to run the script, see [“Create an automation of type script”](#) on page 83.

5. **Cloud** Optionally, you can generate group-specific keys. Use these if you only want users from a specific group to have access to a machine.

In this scenario, the default public key can act as a fallback in the event that none of the other keys work.

- a. Click **New public key for groups**.
- b. Select a group, then use the refresh button to create a public key for the selected group.
- c. The table lists all existing group-specific keys. Use the action buttons on the right to change, delete, or copy the public keys.

Note: Runbook Automation will try every eligible public key for a given RBA user to access a target endpoint until it finds an authorized public key. Some target endpoints might have security policies in place that ban further connection after a certain number of unauthorized connections. Therefore, it is good practice to either avoid having too many group-specific public keys or avoid having RBA users in too many different groups.

Install the IBM Secure Gateway client

For any connection that you want to set up to use automation, you must install the IBM Secure Gateway client.

Before you begin

Private Deployment The IBM Secure Gateway client is not required for a Private Deployment installation.

About this task

You can download the IBM Secure Gateway client from the IBM Secure Gateway management interface.

Procedure

1. Click **Connections**.
2. Click  **Edit** on the Script connection tile.
3. Follow the on-screen instructions on installing the gateway client.
4. In the IBM Secure Gateway management interface, select **RBA Gateway**. Switch to the **Clients** tab and click **Connect Client**. A dialog box is displayed with download options and instructions.

Note: The default security token that is used for securing the connection from your local gateway client to the IBM cloud expires after 90 days. To change the expiration period or to regenerate the security token, in the IBM Secure Gateway management interface click **Settings** (represented by a cogwheel) and then click **Edit**.

5. Verify the connection. The IBM Secure Gateway client is connected when the two rings in the upper right corner are green.

What to do next

Once the IBM Secure Gateway client is installed and connected you can start configuring the IBM Secure Gateway client.

- For more information about the download and installation options, see https://console.ng.bluemix.net/docs/services/SecureGateway/sg_021.html.
- For specific information about the docker installation, see https://console.ng.bluemix.net/docs/services/SecureGateway/sg_003.html.
- For specific information about the native Linux installation, see https://console.ng.bluemix.net/docs/services/SecureGateway/sg_025.html.

Configure the IBM Secure Gateway client

The IBM Secure Gateway client initiates a direct outbound connection to the IBM Cloud Infrastructure.

Before you begin

Private Deployment The IBM Secure Gateway client is not required for a Private Deployment installation.

About this task

To enable connections from IBM Runbook Automation to your local environment you must add the hostname and port of the desired connection to the access control list (ACL) of the Secure Gateway Client. This can be done by using one of the following two methods:

- Client CLI (for more information on how to access the client CLI, see [Using the Secure Gateway client CLI](#)).
- Client Web UI (for further information on the client web UI, see [Setting up a client](#)).

The following procedure describes how to add the hostnames to the ACL using the client CLI.

Procedure

- Depending on your integration goals you must add the following hostnames to the ACL:
 - IBM Netcool/Impact (for using Triggers)
 - IBM BigFix (for using BigFix automations)
 - Any SSH endpoints (for using script automations via the SSH Provider)

- Use the following commands to add these hostnames to the ACL via the Secure Gateway client CLI:

```
acl allow <impact-hostname>:16311
acl allow <bigfix-hostname>:33180
acl allow <ssh-endpoint>:22
```

Note: You can also disable the ACL by using the following command:

```
acl allow :
```

To allow connections to any SSH endpoints using port 22 enter the following command:

```
acl allow :22
```

Finally, to see if your additions were accepted enter the following command:

```
show acl
```

This displays the complete ACL.

Installing Netcool/Impact to run the trigger service

Triggers map events to runbooks. Using triggers, an operations analyst can immediately execute the runbook that matches to an event. To use this mechanism you must set up Netcool/Impact to run the trigger service.

A trigger specifies which event maps to which runbooks.

All information about a trigger, including the event filters, is stored in Netcool/Impact. Netcool/Impact monitors the incoming events from Netcool/OMNIBus. If an event is found in Netcool/OMNIBus that is stored in Netcool/Impact, the following actions are run:

Manual or semi-automated runbooks

Netcool/Impact stores the ID of the runbook and the matching parameters in the OMNIBus alerts.status table. The operational analyst can then use the right-click context menu to launch the matching runbook directly from the event.

Fully automated runbooks

Netcool/Impact runs the runbook via an API call to IBM Runbook Automation. The result is stored in the event journal.

Postinstallation of Netcool/Impact V7.1.0.13 or higher

Configure IBM Tivoli Netcool/Impact V7.1.0.13 or higher to integrate IBM Runbook Automation and map runbooks to events.

Run the following steps on the Impact server. If there are any secondary Impact server(s) installed, ensure that these servers are stopped. The immediate steps below must be completed on the primary Netcool/Impact server. For secondary Netcool/Impact servers, see [“Configuring a secondary ObjectServer”](#) on page 95 and [“Configuring a secondary Netcool/Impact server”](#) on page 96.

Set up IBM Tivoli Netcool/Impact V7.1.0.13 or higher:

Import the database schema and Runbook Automation project

Run the following command under the same user account that you used to install Netcool/Impact..

On Linux systems:

```
$IMPACT_HOME/install/cfg_addons/rba/install_rba.sh <derby_password>
```

On Windows systems:

```
<IMPACT_INSTALL_LOCATION>/install/cfg_addons/rba/install_rba.bat <derby_password>
```

Update the ObjectServer

IBM Runbook Automation requires additional fields in the `alert.status` events table. If you have a high availability ObjectServer setup you must run the command for both the primary and backup server. Copy the following file to the ObjectServer:

```
$IMPACT_HOME/add-ons/rba/db/rba_objectserver_fields_update.sql
```

Then run the following command:

```
$OMNIHOME/bin/nco_sql -username <username> -password <password> <  
rba_objectserver_fields_update.sql
```

<username> is a placeholder for ObjectServer user name and <password> for ObjectServer password.

Note: The < operator is used to pipe the file to the ObjectServer to create the new fields.

Configure and test RBA_ObjectServer data source connection

1. Log in to Netcool/Impact.
2. Switch to the **RunbookAutomation** project from the drop-down menu in the top right corner.
3. Click the **Data Model** tab.
4. Right-click **RBA_ObjectServer datasource** and click **Edit**.
5. Enter the access information for your Netcool/OMNIBus ObjectServer.
6. Test the connection.
7. Save the changes and close the editor.
8. Expand the **RBA_ObjectServer** data source in the tree view.
9. Right-click **RBA_Status** data type and click **Edit**.
10. In the **Table Description** area, click **Refresh Fields** for the **alerts.status** base table.
11. Ensure that the table row for the **Serial** item has a value of **Yes** in the **Key Field** column. If the value displayed is **No**, double-click this table entry to enter edit mode, and select the check box that appears in edit mode.
12. Ensure that **Identifier** is selected in the drop-down box for the **Display Name Field**.
13. Save the changes and close the editor.
14. Right-click **RBA_Status** data type and click **View Data Items**.
15. Verify that some events from Netcool/OMNIBus are listed.
16. Close the **Data Items: RBA_Status** view.

Configure and test RBA_Derby data source connection

1. Log in to Netcool/Impact.
2. Switch to the **RunbookAutomation** project from the drop-down menu in the top right corner.
3. Click the **Data Model** tab.
4. Right-click **RBA_Derby datasource** and click **Edit**.
5. Test the connection to ensure you can communicate successfully with your Derby database.

Note: If you cannot communicate with the Derby database, ensure that the username, password, host, and port are correctly setup in the datasource.

6. Save the changes and close the editor.

Configuring a secondary ObjectServer

If there are multiple ObjectServers, perform the following step on each object server:

Update the ObjectServer

- IBM Runbook Automation requires additional fields in the alert.status events table. If you have a high availability ObjectServer setup you must run the command for both the primary and backup server. Copy the following file to the ObjectServer:

```
$IMPACT_HOME/add-ons/rba/db/rba_objectserver_fields_update.sql
```

Then run the following command:

```
$OMNIHOME/bin/nco_sql -username <username> -password <password> <
rba_objectserver_fields_update.sql
```

<username> is a placeholder for ObjectServer user name and <password> for ObjectServer password.

Note: The < operator is used to pipe the file to the ObjectServer to create the new fields.

Configuring a secondary Netcool/Impact server

If there are any secondary Impact server(s) installed, perform the following steps on the secondary Impact server(s):

- The Derby Database Failure Policy must be set to Failover/Failback.
- Ensure that the following property is set in the primary: NCI1_server.props:
 - Add the following property to the \$IMPACT_HOME/etc/<ServerName>_server.props file on the primary impact server:

```
impact.server.preferredprimary=true
```

When these steps have been completed, start the secondary Impact server(s). The replication in derby will copy across the Runbook Automation configuration to the secondary(s). The impact server logs can be monitored to confirm that the configuration has been copied to the secondary(s).

Note: The following steps are executed on the primary server only:

1. Import the database schema and Runbook Automation project.
2. [“Configure and test RBA_ObjectServer data source connection” on page 95.](#)
3. [“Configure and test RBA_Derby data source connection” on page 95.](#)

Import certificate

The Netcool/Impact servers use SSL connections to communicate with IBM Runbook Automation. Therefore the server certificate for Runbook Automation must be imported into Netcool/Impact's truststore.

Note: Run the following steps on the Impact server, Impact secondary server(s) if you have them installed, and Impact GUI server(s).

Import the signed certificate:

1. On Linux systems, enter the following command to receive the correct certificate:

```
echo -n | openssl s_client -connect <RBA_ACCESS_URL> | sed -ne '/-BEGIN CERTIFICATE-/,/-END
CERTIFICATE-/p' > file.cert
```

Cloud Replace the <RBA_ACCESS_URL> with your Runbook Automation API endpoint, for example with rba-prod.mybluemix.net.

Private Deployment Replace the <RBA_ACCESS_URL> with your Runbook Automation local deployment's runbook service FQDN and port, for example with myrbaerver.mycompany.com:3005.

For example:

```
echo -n | openssl s_client -connect rba.mybluemix.net:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-
END CERTIFICATE-/p' > file.cert
```

If the command does not work in your environment, use the following variant of the command:

```
ex +' /BEGIN CERTIFICATE/, /END CERTIFICATE/p' <(echo | openssl s_client -showcerts -connect <RBA_ACCESS_URL>) -scq > file.cert
```

If errors occur, make sure your exported certificate that is stored in `file.cert` contains a full and valid certificate. Errors like `verify error:num=20:unable to get local issuer certificate` occur due to a missing CA root certificate for the DigiCert CA.

The certificate begins and ends as follows:

```
-----BEGIN CERTIFICATE-----  
...  
-----END CERTIFICATE-----
```

On Windows systems, use your preferred browser to export the certificate.

2. Use the following command to import the certificate:



Warning:

Note:

- The `import_cert` script does not only import the certificate but it also restarts the Netcool/Impact server and the Netcool/Impact GUI server. If this is a production environment you should run this script during planned maintenance only.
- If you have Netcool/Impact running under process control, stop all Netcool/Impact processes in the process control and restart them manually using the stop and start scripts found in `$IMPACT_HOME/bin`. This is necessary because the `import_cert.sh` script will start and stop the Netcool/Impact processes. Once the `import_cert.sh` script completes, stop all Netcool/Impact processes and restart them using the process control.
- If you need to change your RBA certificate, you must delete the old certificates before running the import script again. Use the following command for the Impact and Impact GUI server to delete certificates from the keystore:

```
$IMPACT_HOME/sdk/bin/keytool -delete -alias rba_certificate -keystore $IMPACT_HOME/wlp/usr/servers/<instance>/resources/security/trust.jks -storepass <KEY_STORE_PASSWORD>
```

On Linux systems:

```
$IMPACT_HOME/install/cfg_addons/rba/import_cert.sh <KEY_STORE_PASSWORD>  
<CERTIFICATE_FILE_FULL_PATH>
```

On Windows systems:

```
<IMPACT_INSTALL_LOCATION>/install/cfg_addons/rba/import_cert.bat <KEY_STORE_PASSWORD>  
<CERTIFICATE_FILE_FULL_PATH>
```

where `<KEY_STORE_PASSWORD>` is your Netcool/Impact admin password.

Configure user access rights

IBM Runbook Automation must be connected to Netcool/Impact for trigger management. This connection is secured with a user name and a password.

If you want to create a new user, run step 1 and map the role in step 2. You can also add the role to an existing user and run step 2 only.

1. To create a user, enter:

```
cd $IMPACT_HOME/install/security  
./configUsersGroups.sh -add -user <username> -password <password>
```

2. Map the impactRBAUser role to an existing user or to the user that you created.

```
cd $IMPACT_HOME/install/security
./mapRoles.sh -add -user <username> -roles impactRBAUser
```

Update the Netcool/Impact configuration

Edit the default values of the Netcool/Impact derby database to configure launch-in-context support and integration with IBM Runbook Automation.

Netcool/Impact configuration for fully automated runbooks is stored in the Netcool/Impact derby database in the rbaconfig.defaults table with the exception of the RBAAPIKeyPassword.

The table contains the following default values:

<i>Table 23: Default values of Netcool/Impact derby database</i>	
Default value	Description
RBAHost= ' '	<p>Cloud Change the default value to your runbook automation API endpoint. For example, rba.mybluemix.net.</p> <p>Private Deployment Change the default value to the DNS name or IP address of the server where you installed the runbook service.</p>
RBAManualExecHost= ' '	<p>Cloud Change the default to the URL you were given as part of your subscription.</p> <p>If you work with the following URL https://ibmopsmanagement.mybluemix.net/index?subscriptionId=EF0E868D1965A2D1E6F64EDFF51BA985778C690A2, specify <code>ibmopsmanagement.mybluemix.net</code> as default value.</p> <p>Private Deployment This property is not used.</p>
RBAPort= '443'	<p>Cloud No changes are required for RBA Application Port.</p> <p>Private Deployment Change the default value to the port of the runbook service. The default value of the runbook service port is 3005.</p>
RBAProtocol= 'https'	No changes are required for RBA Application Protocol.
RBARESTPath= '/api/rba/'	No changes are required for RBA REST path.
RBARESTPathToView= 'api path'	No changes are required for RBA API path.

Connect to your database with the same connection string as identified in [“Postinstallation of Netcool/Impact V7.1.0.13 or higher”](#) on page 94. For example:

```
ij> connect 'jdbc:derby://localhost:1527/ImpactDB;user=impact;password=derbypass;';
ij>
```

Use the sample commands to update the fields:

```
update rbaconfig.defaults set FieldValue = 'RBAdevelopmentTeam/hytmamvirtex' WHERE
FieldName='RBAAPIKeyName';
update rbaconfig.defaults set FieldValue = 'rba.mybluemix.net' WHERE FieldName='RBAHost';
update rbaconfig.defaults set FieldValue = 'ibmopsmanagement.mybluemix.net' WHERE
FieldName='RBAManualExecHost'; ;
update rbaconfig.defaults set FieldValue = '600' WHERE
FieldName='MaxNumTrialsToUpdateRBAStatus';
update rbaconfig.defaults set FieldValue = 'EF0E868D1965A2D1E6F64EDFF51BA985778C690A2' WHERE
FieldName='RBASubscription';
```

Storing the API Key Password

The API key password that Netcool/Impact uses to access Runbook Automation is based on the configuration property `RBAAPIKeyPassword` and must be encrypted by using `$IMPACT_HOME/bin/nci_crypt <RBAAPIKeyPassword value >`.

For example, an encrypted value can look as follows:

```
{aes}BE0A8FAB4084D460CBD664FECDE3674A8B728BE7986F56F8B2F542C2056B4A2D1E6F64EDFF51BA985778C690A2E
95
```

This encrypted value must be stored in a text file within the Netcool/Impact install location. Create the file `$IMPACT_HOME/etc/RBAAPIKeyPassword.txt` and copy the encrypted value into this file.

Note: If there are any additional secondary Netcool/Impact server(s) installed, repeat all of the steps on each of the secondary Netcool/Impact server(s). This is necessary because the encrypted value of the API key password depends on the individual Netcool/Impact system and cannot be shared by the primary and secondary Netcool/Impact server(s).

Enable the launch-in-context menu

You must enable the launch-in-context menu to start manual or semi-automated runbooks from events.

Launch-in-context is only available if you created a matching trigger. For more information about how to create a trigger, see [“Create a trigger”](#) on page 105.

Private Deployment For an IBM Runbook Automation Private Deployment, follow the procedure in [“Configuring an IBM Netcool/OMNIBus event tool for launch-in-context to IBM Runbook Automation”](#) on page 32.

Cloud Use the Netcool/OMNIBus Web GUI to create the launch-in-context menu.

1. Create a launch-in-context menu and a corresponding menu entry.
 - a. Log on to Netcool/OMNIBus Web GUI as an administrator.
 - b. Select **Administration > Event Management Tools > Tool Configuration**.
 - c. Click the **Create Tool** icon to create a new tool.
 - d. Set the **Name** field to **LaunchRunbook**.
 - e. From the **Type** drop-down field, select **CGI/URL**.
 - f. Make sure the data source, for example **OMNIBus**, is selected in the data source configuration.
 - g. In the **URL** field, enter one of the following URLs:

Up to Netcool/Impact 7.1.0.12:

```
https://<URL>/index?
subscriptionId=<YOUR_SUBSCRIPTION_ID>&dashboard=rba.dashboard.executerunbook&context=%7B
%22rbid%22%3A%22%7D%22%2C%22bulk_params%22%3A
%22%7D%22%7D
```

Starting with Netcool/Impact 7.1.0.13:

```
https://<URL>/index?
subscriptionId=<YOUR_SUBSCRIPTION_ID>&dashboard=rba.dashboard.executerunbook&context=%7B
%22rbid%22%3A%22%7D%22%2C%22bulk_params%22%3A
%22%7D%22%7D
```

Where: <URL> is your URL to IBM Runbook Automation.

<YOUR_SUBSCRIPTION_ID> is your subscription ID.

Up to Netcool/Impact 7.1.0.12 example:

```
https://pano-prod.mybluemix.net/index?
subscriptionId=RBADevelopmentTeam&dashboard=rba.dashboard.executerunbook&context=%7B
%22rbid%22%3A%22%7D%22%2C%22bulk_params%22%3A
%22%7D%22%7D
```

Starting with Netcool/Impact 7.1.0.13 example:

```
https://pano-prod.mybluemix.net/index?
subscriptionId=RBADevelopmentTeam&dashboard=rba.dashboard.executerunbook&context=%7B
%22rbid%22%3A%22%7D%22%2C%22bulk_params%22%3A
%22%7D%22%7D
```

- h. Make sure **Method: GET** is selected.
 - i. Select **Open in: Specific window**. Enter a name for the window, for example: "Runbook Automation". When the launch tool is started for the first time a new browser tab opens. For all subsequent invocations the same browser tab is reused.
 - j. Select the **Execute for each selected row** check box.
 - k. Select the appropriate group roles in the **Access Criteria** section. For example, Netcool_OMNIBus_Admin and Netcool_OMNIBus_User.
 - l. Click **Save**.

For more information about configuring launch-in-context support, see [Configuring launch-in-context integrations with Tivoli products](#).
2. Complete the following steps to add the Runbook Automation entry to the launch-in-context menu of the Netcool/OMNIBus event list:
 - a. Log on to Netcool/OMNIBus Web GUI as an administrator.
 - b. Select **Administration > Event Management Tools > Menu Configuration**.
 - c. Select **Alerts > Modify**. Netcool/OMNIBus Web GUI displays the **Menus Editor** dialog box.
 - d. Select **LaunchRunbook** from the available items.
 - e. Log on to Netcool/OMNIBus Web GUI as an administrator.
 - f. Select the arrow to add **LaunchRunbook** to the **Current Items** list. You can optionally rename the menu entry name or add a space between Launch and Runbook.
 - g. Click **Save**.

Configure a connection to the Netcool/Impact server

This section outlines the steps for adding a connection to Netcool/Impact in IBM Runbook Automation.

Before you begin

Verify that all prerequisites are met, as described in [“Event Trigger \(Netcool/Impact\)”](#) on page 89.

Cloud A gateway client must be installed and running.

Procedure

Complete the following steps to configure a new connection:

1. Logon to Bluemix.
2. You can configure a connection on the **Connections** page. Click  **Edit** on the Event Trigger tile to open the configuration window.
3. Select **I want an Event Trigger (OMNIbus/Impact)**.
4. For the IBM Netcool/Impact REST service URL, enter the IP address and port of your local Netcool/Impact GUI server installation, followed by `ibm/custom/RBARestUIServlet`. For example, `https://9.168.48.28:16311/ibm/custom/RBARestUIServlet`.

Note: **Cloud** The URL can be a private IP address. The gateway infrastructure routes your request to the gateway connector, from which you can then use IP addresses local to your environment. For the user and password, add the username and password that you configured in [“Configure user access rights”](#) on page 97.

Private Deployment The URL can be a private IP address only if you do not plan to [“Enable secure communications within the RBA service”](#) on page 44. In general, you should reference the host name of the Impact server as it is included in the "Common Name (CN)" field of the SSL certificate for the Impact server. For the user and password, add the username and password that you configured in [“Configure user access rights”](#) on page 97.

Upgrading Netcool/Impact to a higher fix pack level when the RBA add-on is already installed

Migrate the RBA add-on from a previous Netcool/Impact level to a new Netcool/Impact level.

Procedure

- If the RBA add-on is installed and configured already within a Netcool/Impact installation, and you upgrade the Netcool/Impact installation to a higher level, the upgrade also refreshes the RBA add-on. So typically no additional action is required.
- It is recommended to install Netcool/Impact V7.1.0.13 or higher.
- If you upgrade to Netcool/Impact V7.1.0.9, you must perform some additional steps to refresh the RBA add-on:
 - a) Locate the following files that have been updated for Netcool/Impact V7.1.0.9. After the fix pack installation the files are located in `/opt/IBM/tivoli/impact/add-ons/rba/importData/Policy/com.micromuse.response.dblayer.PolicyOrgNode`:
 - RBA_Constants.js
 - RBA_EventsProcessing.js
 - RBA_ExecuteRunbook.js
 - RBA_ExecutionStatusTracker.js
 - RBA_TrackProgress.js
 - b) Compare the content of each file with the "active" content in the **Policies** view of the Netcool/Impact UI. For each file that is not identical, create a backup copy of the active file, and replace it with the content from the directory in step 1.
- If you upgrade to Netcool/Impact V7.1.0.10, you must perform some additional steps to refresh the RBA add-on:

- a) Locate the following file that has been updated for Netcool/Impact V7.1.0.10. After the fix pack installation the files are located in /opt/IBM/tivoli/impact/add-ons/rba/importData/Policy/com.micromuse.response.dblayer.PolicyOrgNode:
 - RBA_ManageTriggers.js
- b) Compare the content of this file with the "active" content in the **Policies** view of the Netcool/Impact UI. If the file is not identical, create a backup copy of the active file, and replace it with the content from the directory in step 1.
- c) Refresh the **RBA_Status** data type within the **RBA_ObjectServer** data source connection as described in "Configure and test RBA_ObjectServer data source connection" in ["Postinstallation of Netcool/Impact V7.1.0.13 or higher"](#) on page 94.
- Verify the connection to RBA_Derby:
 - a) Open the Netcool/Impact UI and click **Data Model > RBA_Derby > Test Connection**.
 - b) If the message says "Connection could not be made", complete these additional steps:
 - a. Edit **RBA_Derby**.
 - b. In **Primary Source / Host Name** change **localhost** to the actual host name.
 - c. Click **Test Connection**. The message should say **Connection OK**.
 - d. Save RBA_Derby configuration and close the editor.
 - e. Repeat **RBA_Derby > Test Connection**. The message should say "Connection OK".

Migrating from IBM Workload Automation Agent to SSH

Before you begin

1. **Private Deployment** Update the RBA installation. For more information, see ["Updating the RBA service"](#) on page 29.
2. Decide if you need a jump server to access endpoints using SSH.

Private Deployment By default, the SSH calls are executed from the RBA server. If the RBA server has no connection to the endpoints, a jump server is required. For example, use the system where the TWA agent was installed as a jump server.

Cloud By default, the SSH calls are executed from the system where the IBM Secure Gateway client is installed. If this system has no connection to the endpoints, a jump server is required. For example, use the system where the TWA agent was installed as a jump server.

3. Plan a maintenance window for the migration. Note that after step 1 of the Migration procedure, you cannot execute automations until step 3 is completed.

Note: IBM Workload Automation Agent stores the credentials to access endpoints in a credential vault. Using the SSH provider, this is no longer required. Instead SSH public key authentication is used. This can be done for specific users or the root user. For more information, see ["SSH script automation provider"](#) on page 90.

Procedure

1. Delete the existing TWA script connection.
2. Create a new SSH script connection.
3. Follow the steps described in the on-screen dialog. For more information, see ["SSH script automation provider"](#) on page 90.
4. If an SSH script connection has been set up and script automations are running successfully on the target endpoints, IBM Workload Automation Agent is no longer required and can be uninstalled.

Chapter 9. Triggers

If you have events that always correspond to the same runbook you can create a trigger and link the event with the runbook. Triggers can run with manual and automated runbooks. If the runbook is a manual or semi-automated runbook, the operator must complete the parameter values. If the runbook is fully automated, the trigger runs with pre-defined values. The operator does not even notice that the runbook was executed.

You must install Netcool/Impact to run the trigger service. For more information, see [“Installing Netcool/Impact to run the trigger service”](#) on page 94.

Create a trigger

Map existing runbooks to Netcool/Impact events. Triggers can run manually or automatically, depending on the type of runbook that is mapped to the event.

You can create a new trigger if you are an RBA Author or RBA Approver. For more information about roles, see [“Create users and assign roles”](#) on page 10.

Before you begin

You must create an Event Trigger connection before you can work with triggers. For more information, see [“Event Trigger \(Netcool/Impact\)”](#) on page 89.

Procedure

1. Open the **Triggers** page and click **New Trigger**. The Add Trigger dialog opens.
2. Enter the **Name** and **Description** of the Trigger. Name is a required field. Describe what kind of events this trigger is used for. What problem that is reported by an event does the mapped runbook solve?
3. Click **Add Filter** to create a filter for the events. You can click Add Filter after you have entered the Name.
4. Create at least one filter or combine many filters with each other. The following filter types are provided:

Event Summary

A simple way to filter events. Define a filter that runs over the Event Summary field. Select an operator from the drop-down list, which indicates whether the entered text is contained, not contained, equal, or not equal. Then, enter a text string or numbers.

For example, Event Summary contains 'DB2'. This query returns all events that contain DB2 in the Event Summary field.

Event Filter

Use existing filters. Select all filters that you want to use from the list. You can select several items. Use search if you want to filter the list of filters. Hover over the event filter to understand what the query looks like. For example, hover over the **CPU-High** filter. The hover displays Summary LIKE 'cpu-high'.

For example, select Critical Alarms and Errors Alarms. The query returns all events where severity is higher than 5 and the **Summary** field contains the string error.

Attribute Filter

Create your own filter. Enter a description for the new filter that you want to create. Select an attribute, operator, and enter a value. Examples for each attribute are displayed below the **Value** field. Click **Add** filter to add the new filter to the list.

Click **Save**. All filters that are selected or created are displayed. You can edit **Event Summary** and **Attribute Filter**. Choose how you want to match filter criteria to events.

Choose whether events and filters are mapped with AND or OR. For example, if you selected AND, then the mapping is processed as follows: Event1 is mapped with filter1 AND filter 2 AND filter 3. If you selected OR, then the mapping is processed as: Event 1 is mapped with filter 1 OR filter 2 OR filter 3.

You can click **View Sample Events** if you want to know the outcome of your selected filters. You can close the **Sample Events** window by selecting the cross icon in the upper right corner.

5. Click **Select Runbook** to map a runbook to your event filter.
6. Select **Runbook Type** and choose if you want to list automatic or manual runbooks. The difference is that automatic runbooks do not require any manual interaction from the operator. If the event occurs, the runbook runs automatically. For manual runbooks, the runbook is added to the event, so that the operator can launch to the runbook from the event viewer. You can search the list of runbooks to filter the result list. Select the runbook and click **Save**.
7. The parameters of the selected runbook are displayed. Select how you want to enter the parameter value:

From event

The parameter value is contained in the event. If the runbook is started, the parameter value is used from the event.

Manual

If the runbook is started, the operator manually enters the parameter values.

You can also enter a pattern to extract a subset of the parameter value by entering a regular expression. This is optional. The regular expression syntax is based on the Perl regular expression syntax. The syntax is the same as that used in Netcool/Impact.

Note: If you change the corresponding runbook and, for example add or remove a parameter, the parameter mapping will no longer work. Ensure that you edit the trigger again and update the parameter mapping.

8. Select **Enable this trigger to run** if you want this trigger to be used. Then, the trigger is added to the **Triggers** table with the flag **Enabled**. If you do not want to use this trigger, clear the check box. The trigger is added with the flag **Disabled**.
9. Select **Run this runbook automatically**. This option is only available for automatic runbooks.

Triggers

Add, edit, delete, or check if your trigger is disabled or enabled.

The following functions are available on the **Triggers** page:

New Trigger

Click Add Trigger to open the Add Trigger dialog and configure your trigger. For more information, see [“Create a trigger” on page 105](#). You can add a trigger if you are an RBA Author or RBA Approver.

Edit Trigger

In the Triggers table, click  **Edit** to change your current settings.

Delete Trigger

If this trigger is no longer valid, click  **Delete**. You can delete a trigger if you are an RBA Manager.

Display statistics

Click the arrow at the end of the row to see how often the runbook that is used for this trigger ran successfully or failed.

Check if the trigger is enabled or disabled

The Manage Trigger table displays the current status of a trigger (disabled or enabled). To enable a trigger, click **Edit** and select **Enable this trigger to run** at the bottom of the dialog.

Check if an automatic or manual runbook is used

You can find out if the trigger is mapped to an automatic or manual runbook (Automated Runbook **Yes** or Automatic Runbook **No**).

Check if the runbook that is assigned to the trigger still exists

The Manage Trigger table displays any triggers that are broken because the assigned runbook has been deleted.

For more information about roles, see [“Create users and assign roles” on page 10](#).

Chapter 10. Runbook Automation API

IBM Runbook Automation supports integration with other software and tools via the external Runbook Automation HTTP REST Interface.

The full interface with the complete list of available API endpoints can be accessed at: <https://rba.mybluemix.net/apidocs-v1>.

The interface is documented with Swagger (<http://swagger.io>) and the Swagger UI (<https://swagger.io/swagger-ui/>).

Working with the Swagger UI

The Swagger UI allows you to try out the API calls immediately once a user is logged in and authorized. Complete the following steps to authorize at the Swagger UI:

1. Create an API key as described in “API Keys” on page 113.
2. Make a note of the generated API name and password.
3. From the Swagger header section, click **Authorize** and enter the API key name and API key password in the login dialog and then click **Authorize**.

Private Deployment Using the Swagger UI for API calls does not work for an RBA Private Deployment as it connects to <https://rba.mybluemix.net/>. For a Private Deployment use a HTTP REST client or a tool such as cURL to test your calls.

Working with cURL

If you have a Runbook Automation Private Deployment, do not want to use the Swagger UI, or want to script from a shell script, cURL is a good (but not the only) option to access the Runbook Automation API.

The following items need to be considered when using cURL:

- Runbook Automation does not support charsets other than UTF-8. If a header is specified defining another charset, the call will fail. This is an example of a failing cURL command:

```
curl (...) --header 'Content-Type: application/json; charset=usascii'
```

- Runbook Automation does not support the HTTP header "Expect: 100-continue". If the input you provide is larger than 1024 Bytes, some versions of cURL send this header by default and you must turn it off. To do this, add the following header to your cURL command: `--header 'Expect: '`.

Obtaining the base URL of the RBA API

RBA Marketplace (standalone or as part of Cloud Event Management)

The base URL is <https://rba.mybluemix.net/api/v1/rba>.

RBA Private Deployment

The base URL is https://<RBA_SERVER_HOST_NAME>/api/v1/rba.

Note: The RBA server hostname is not necessarily equal to the DASH server host name. Typically, the RBA server is installed on a different system than the DASH server where the Netcool/OMNIBus UI is running.

RBA as part of Cloud Event Management on IBM Cloud Private

The base URL is https://<CEM_HOST_NAME>/api/v1/rba.

Note: In Cloud Event Management deployments, use the API keys for Cloud Event Management to access the RBA API.

RBA as part of Cloud Event Management on Bluemix

The base URL depends on the region used:

Sydney: <https://rba-bm.au-syd.mybluemix.net/api/v1/rba> and <https://rba-au-syd.opsmgmt.bluemix.net/api/v1/rba>.

London: <https://rba-bm.eu-gb.mybluemix.net/api/v1/rba> and <https://rba-eu-gb.opsmgmt.bluemix.net/api/v1/rba>.

Dallas: <https://rba-bm.mybluemix.net/api/v1/rba> and <https://rba-us-south.opsmgmt.bluemix.net/api/v1/rba>.

Creating or modifying a runbook with the API

This section contains instructions for creating and modifying runbooks with special runbook elements, such as parameters, automations, commands, and goto elements.

About this task

You must send a JSON document to the API endpoint using the POST API: `POST /api/v1/rba/runbooks`. The document contains the definition of the runbook, as per the input model defined in the Swagger API documentation. For more information, see [Chapter 10, “Runbook Automation API,”](#) on page 109.

Use the following points to create the runbook elements within the description of a step.

Procedure

- **Create a parameter**

To create a parameter, include it in the **parameters** section of your JSON input. Use the following syntax to use it inside the runbook description:

```
<span class=\"span-grp rba-param\">${PARAMETERNAME}</span>
```

Where `${PARAMETERNAME}` is the name of the parameter chosen in the **parameters** section of the input JSON. The following JSON shows an example of a runbook defining one step with a mandatory parameter:

```
{
  "name": "An example runbook",
  "steps": [
    {
      "number": 1,
      "description": "Solve the task to solve the problem in data center <span class=\"span-grp rba-param\">dataCentre</span>."
    }
  ],
  "parameters": [
    {
      "name": "dataCentre",
      "description": "The data centre in which the operation needs to be done"
    }
  ]
}
```

- **Insert an automation**

To insert an automation, define the parameter mapping in the **automationMappings** section of your JSON input. Use the following syntax to apply it inside the runbook description:

```
<span class=\"span-grp rba-autom\" data-automationid=\"${AUTOMATIONID}\" data-mappingid=\"${MAPPINGID}\">${AUTOMATIONNAME}</span>
```

Where `${AUTOMATIONID}` is the ID of the Automation Definition, `${MAPPINGID}` is the number of the mapping defined in the **automationMappings** section of the JSON input, and `${AUTOMATIONNAME}` is the name of the automation.

The following JSON shows an example of a runbook defining one step containing an automation and a matching automation mapping. The automation mapping object maps the runbook parameter

hostname to the automation parameter TARGET and the runbook parameter serviceName to the automation parameter SERVICE:

```
{
  "name": "An example runbook with an automation",
  "steps": [
    {
      "number": 1,
      "description": "To restart the service, execute the following automation: <span class=
\"span-grp rba-autom\" data-automationid=\"abcdef0123456789abcdef0123456789\" data-mappingid=
\"1\">Restart Service</span>."
    }
  ],
  "automationMappings": [
    {
      "mappingId": "1",
      "parameterMappings": [
        {
          "automationParameterName": "TARGET",
          "parameterMappingType": "parameter",
          "parameterValue": "hostname"
        },
        {
          "automationParameterName": "SERVICE",
          "parameterMappingType": "parameter",
          "parameterValue": "serviceName"
        }
      ]
    }
  ]
}
```

- **Insert a command**

Use the following syntax to mark a section of your step description as a command:

```
<span class=\"span-grp rba-command\">${COMMAND}</span>
```

Where \$COMMAND is the text you want to be displayed and executed as a command. The following JSON shows an example of a runbook defining one step with a command:

```
{
  "name": "An example runbook with a command",
  "steps": [
    {
      "number": 1,
      "description": "Log in to the system and display the running processes with the
following command: <span class=\"span-grp rba-command\">ps -ef</span>."
    }
  ]
}
```

- **Insert a GoTo element**

Use the following syntax to insert a GoTo element into a runbook:

```
<span class=\"span-grp rba-goto\" data-stepid=\"${NUMBER}\">${TEXT}</span>
```

Where \$NUMBER is the step number that the GoTo element is directing the user to. Use -1 to go to the end of the runbook. And \$TEXT is the text to appear inside the element. The following JSON shows an example of a runbook defining multiple steps, with a GoTo to step 3 within the first step:

```
{
  "name": "An example runbook with a GoTo",
  "steps": [
    {
      "number": 1,
      "description": "If you are running on a system with SELinux disabled, directly <span
class=\"span-grp rba-goto\" data-stepid=\"3\">Step 3</span>."
    },
    {
      "number": 2,
      "description": "Disable SELinux."
    }
  ]
}
```

```

    },
    {
      "number": 3,
      "description": "Issue a command that should prevented by SELinux and make sure it
executed okay."
    }
  ]
}

```

- **Insert a collapsible section**

Use the following syntax to insert a collapsible section with a title into a runbook:

```

<div class=\"rba-collapse\"><p class=\"rba-collapse-title\">${TITLE}</p>\n\n<div class=\"rba-
collapse-content\">\n<p>${COLLAPSIBLECONTENT}</p>\n</div>

```

Where `$(TITLE)` is the text to always be displayed and `$(COLLAPSIBLECONTENT)` is the content only visible when expanded. The following JSON shows an example of a runbook with a collapsible section in step 1. The title contains the text **Show Details** and the text contains additional information:

```

{
  "name": "An example runbook with a collapsible section",
  "steps": [
    {
      "number": 1,
      "description": "<p>The system name is server01.mydomain.com</p><div class=\"rba-
collapse\"><p class=\"rba-collapse-title\">Show details:</p>\n\n<div class=\"rba-collapse-
content\">\n<p>The system is has a 4 core CPU and 8 GiB RAM.</p>\n</div>"
    }
  ]
}

```

- **Changing a runbook**

To change an existing runbook instead of creating a new one, use the API endpoint `PATCH /api/v1/rba/runbooks/$(RUNBOOKID)`. Where `$(RUNBOOKID)` is the runbook ID of the runbook that you want to change. The JSON sent will update the sections present in the JSON and leave the sections not present untouched. The following JSON is sent to `/api/v1/rba/runbooks/abcdefabcdefabcdefabcdefabcdefab`:

```

{
  "description": "This runbook has been updated with a new description: Use this runbook to
resolve issues around the error code \"ARBR108E\"."
}

```

This action will only change the description of the runbook and leave all other properties, for example name, steps, parameters, and tags unchanged.

API usage scenario

The following example demonstrates how the Runbook Automation API can be applied in a more complex workflow scenario. In this scenario, a user wants to integrate an external program with IBM Runbook Automation. The external program should scan the runbook repository and add a new step to all runbooks where the runbook contains a tag with the content `addstep`. Additionally, the external program should only consider runbooks that have been published.

1. Get a list of all available runbooks

The application calls the API endpoint `GET /api/v1/rba/runbooks` to retrieve a list of all runbooks. Looking at the complete response model of the runbook, it becomes clear that the program must only evaluate the fields **runbookId**, **steps** and **tags**. Steps and tags are on the root level of the document, but `runbookId` is part of the **readOnly** section. As only published versions are relevant, the query parameter `version` is set to `approved`. The complete call, as provided by the Swagger UI, is:

```

curl -X GET --header 'Accept: application/json' 'https://rba.mybluemix.net/api/v1/rba/
runbooks?version=approved&fields=readOnly%2Csteps%2Ctags'

```

The response is a JSON array with each entry containing the properties **readOnly**, **steps** and **tags**.

2. Output processing by external program

The user writes logic to the external program in order to filter the output. All entries of the JSON array where the tag array does not contain an entry with `addStep` are discarded. Of the remaining entries, the `runbookIds` are stored together with the steps in a data structure, as per the following JSON format:

```
[
  {
    "runbookId" : "$runbookId",
    "steps" : [ ... ]
  },
  (...)
]
```

Within the **steps** section, the external program changes the content by adding a new step at the end that contains the static content required by the user. For example, an existing runbook with the following steps:

```
steps : [
  {
    "number" : 1,
    "description" : "Log in to the system and install the latest updates provided by the OS vendor."
  }
]
```

Would be extended to:

```
steps : [
  {
    "number" : 1,
    "description" : "Log in to the system and install the latest updates provided by the OS vendor."
  },
  {
    "number" : 2,
    "description" : "To conclude this runbook, reboot the system."
  }
]
```

3. Change the runbooks

The user creates a for-each loop in the external program that iterates over the data structure created in step 2. For each entry, the external program calls the API endpoint `PATCH /api/v1/rba/runbooks/:runbookId` (where `:runbookId` is the ID of the runbook from the data structure shown above). The **steps** section of the prepared data is sent to the body. An example call for a runbook with the ID `7ef63332-5a3e-40f7-a923-af3ffb6795d3` and a steps section as described above would look as follows:

```
curl -X PATCH --header 'Content-Type: application/json' --header 'Accept: application/json' -d '\ steps : [ \ { \ "number" : 1, \ "description" : "Log in to the system and install the latest updates provided by the OS vendor." \ }, \ { \ "number" : 2, \ "description" : "To conclude this runbook, reboot the system." \ } \ ] \ ' 'https://rba.mybluemix.net/api/v1/rba/runbooks/7ef63332-5a3e-40f7-a923-af3ffb6795d3'
```

Summary

By using the Runbook Automation API in an external program, you can automatically and systematically change a large number of runbooks based on the data already present in the system.

API Keys

Cloud Private Deployment

Use API keys to import or export runbooks by using the HTTP API of IBM Runbook Automation. API keys are used to authenticate your HTTP API requests.

Cloud Event Management The procedure for generating API Keys in Cloud Event Management is documented at <https://www.ibm.com/support/knowledgecenter/en/SSURRN/com.ibm.cem.doc/index.html>.

Each API key is associated with a particular user. Requests that are authenticated with an API key are authorized with the same permissions as the owner of that API key. The owner of an API key is the user who created it.

You can export a runbook by using existing APIs. Complete the following steps to export a runbook:

Create an API key

Create an API key to authenticate to the Runbook Automation API:

1. Log in to IBM Runbook Automation as an RBA Manager. The RBA Manager role provides the access rights to create API keys.
2. Open the **API keys** page and click **New API key**.
3. Enter a **Description**. You can create multiple API keys. The description helps you to distinguish between them.
4. Click **New API key**. The **API key name** and **API key password** are generated and displayed in the dialog.
5. Copy the API key name and password and save them for your reference. For security reasons, the API key password is not displayed in **API Keys**. If you lose your API key password you must create a new API key. Delete any unused API key.

Delete an API Key

On the API Keys page, click  **Delete** to delete the API Key.

You can create and delete API keys if you are an RBA Manager. For more information about roles, see [“Create users and assign roles” on page 10](#).

Chapter 11. IBM Runbook Automation FAQ

Various questions arise as users work with IBM Runbook Automation. Answers to these questions are provided for your reference.

The following frequently asked questions are answered in this section:

- [“Can I create an unlimited number of runbooks?” on page 115](#)
- [“Can I integrate with existing automation tools?” on page 115](#)
- [“Can I run runbooks from products other than OMNIBus?” on page 115](#)
- [“How many instances of the IBM Secure Gateway client and IBM BigFix do I need to install/integrate?” on page 116](#)
- [“In which scenario might I need another instance of a local component?” on page 116](#)
- [“Can I customize the credentials for execution of script automation?” on page 116](#)
- [“How can I avoid an unexpected newline character at the end of a script automation's output that gets passed as input to another automation?” on page 116](#)
- [“How does user management for the SaaS solution work?” on page 116](#)
- [“What happens if I deactivate the approval workflow?” on page 116](#)
- [“Where can I find a trial version of IBM Runbook Automation Private Deployment?” on page 116](#)
- [“How can I update the event journal from a runbook?” on page 117](#)
- [“Why is Internet Explorer not displaying RBA application texts in the language I have set in the language configuration of the browser?” on page 117](#)
- [“Can I change a runbook parameter during execution?” on page 117](#)
- [“Why are special characters incorrectly encoded if I create content via the RBA API?” on page 117](#)

Can I create an unlimited number of runbooks?

Yes, however, as per the terms and conditions you are only allowed to execute up to 4000 automatic or semi-automatic runbooks per month.

Can I integrate with existing automation tools?

Yes. RBA supports direct integration with IBM BigFix, and comes with an agent to execute scripts in your environment. Via scripts, you could write integrations with any other product that supports script based access.

Can I run runbooks from products other than OMNIBus?

Yes, however, you must build the integration yourself.

- Integration is possible via launch-in-context to the RBA UIs, as described in [Launching the Run runbook browser page](#).
- You can also automatically execute a runbook via the API. For more information about this feature, go to <https://rba.mybluemix.net/swagger/?url=https://rba.mybluemix.net:443/doc-v1>.

How many instances of the IBM Secure Gateway client and IBM BigFix do I need to install/integrate?

Per SaaS subscription, you will need exactly one installation of IBM Secure Gateway client. Depending on which kind of automations you want to leverage you need one TWA and/or one BigFix server. RBA only supports one instance of TWA and one instance of BigFix.

Private Deployment You do not need to install the IBM Secure Gateway client in a private deployment.

In which scenario might I need another instance of a local component?

If you have more than one subscription. If you want to automate in two (or more) distinct (logical or physical) networks that cannot communicate with each other, you must have two (or more) subscriptions and therefore require the local setup for each subscription.

Can I customize the credentials for execution of script automation?

Yes. For every target server you can define which credentials should be used, when the request from a specific RBA user comes in. You can also define generic credentials for a machine if that level of customizing is not necessary or as a fallback.

How can I avoid an unexpected newline character at the end of a script automation's output that gets passed as input to another automation?

Make sure that the output from the first script does not end with a newline character. For example, use the **echo** command's **-n** option: **echo -n "<some text>"** as the last statement in the first script.

How does user management for the SaaS solution work?

The general workflow for the Bluemix Service and Marketplace offerings is the same:

1. Log in to IBM Marketplace with your IBM ID (this is the same ID used for Developerworks).
2. Create a subscription to Runbook Automation and access the application. For more information, see [“Create users and assign roles” on page 10](#).
3. Grant access to this subscription to other IBM IDs. For more information, see [“Create users and assign roles” on page 10](#).

For Bluemix, any user that is part of the organization where the IBM Runbook Automation service is added, and accesses RBA through the launch page, is added automatically as a user.

What happens if I deactivate the approval workflow?

If the approval process is deactivated, all pending approvals are deleted.

Where can I find a trial version of IBM Runbook Automation Private Deployment?

For a trial of IBM Runbook Automation use the trial version of the cloud service (<https://www.ibm.com/us-en/marketplace/runbook-automation>). The private deployment version has very similar features. For more information, see [“Differences between an RBA Cloud and an RBA Private Deployment” on page 7](#).

Note: If you want to migrate runbooks created during the trial phase to your private deployment, ensure that **before** the trial period expires, you migrate the created runbooks as described in [“Migrating data from RBA Cloud to RBA Private Deployment and vice versa”](#) on page 65.

How can I update the event journal from a runbook?

You can update the event journal using Automations in the runbook. The Automation calls the ObjectServer REST API. For more information, see [Updating Netcool/OMNIbus journal from manually executed runbooks](#).

Why is Internet Explorer not displaying RBA application texts in the language I have set in the language configuration of the browser?

This is a known bug in Internet Explorer. The browser takes the locale information from the system locale settings instead of the browser locale settings. If you want to change the display language of RBA in your browser, change the Windows system locale settings.

Can I change a runbook parameter during execution?

Yes, you can change parameter values after you click **Apply and Run**, as long as the parameter has not yet been used by a step in the runbook. To change a parameter, scroll to the top of the steps section and define a value in the details section of the runbook. Note, the details section is an expandable section beside the runbook title. After specifying a new parameter value you must click **Update**.

Why are special characters incorrectly encoded if I create content via the RBA API?

Make sure your input is always encoded with an UTF-8 character set. If the input is encoded differently, special characters will not be displayed correctly.



Part Number:

SC27-8766-05



(1P) P/N: