IBM

# IBM® Rational® Test Automation Server

## 10.1.2 Documentation

## December 2020

# Special notice

Before using this information and the product it supports, read the information in Notices on page cccxxxii.

# Contents

**Chapter 1. Release Notes....6**
Description....6
What's new....6
Installing Rational® Test Automation Server....8
Known issues....8
Contacting IBM Rational Software Support....9
**Chapter 2. System Requirements....10**
Hardware....11
Operating systems and Containers....11
Host prerequisites....13
Supported software....14
**Chapter 3. Getting Started Guide....16**
Server overview....16
Supported versions of assets and resources....17
Tests supported on Rational Test Automation Server....18
Task flow: Test runs and results....20
Task flow: Server access from desktop clients....22
Accessibility features....24
**Chapter 4. Administrator Guide....25**
Installation of the server software....25
Installation of the server software on Red Hat OpenShift....26
Installation of the server software on Ubuntu....35
Server software upgrade methods....47
Server software upgrade methods on Red Hat OpenShift....48
Server software upgrade on Ubuntu....48
Backing up and restoring the user data....49
Backing up and restoring the user data on Red Hat OpenShift....50
Backing up and restoring the user data on Ubuntu....51
Uninstallation of the server software....56
Uninstalling the server software....56
Uninstalling the server software....57
Configuration of the server software....57
User administration....58
Certificate authority: Importing and extending lists....71
Copying third-party application Jars to Kubernetes....74
Changing the password seed....76
License management....78
Floating licenses....79
Configuring floating licenses....80
**Chapter 5. Test Author Guide....83**
Datasets overview....83
Creating a dataset....84
Editing a dataset....85
Publishing a dataset....93
Viewing a dataset....94

Deleting a dataset....96
**Chapter 6. Test Execution Specialist Guide....98**
Prerequisites to running tests....98
Test run considerations for AFT Suites....98
Test run considerations for API Suites....99
Test run considerations for schedules....103
Considerations for using Jaeger traces in reports....104
Agents overview....104
Test run considerations for running tests on remote agents....105
Adding an agent....106
Managing Docker hosts....108
Test run considerations for running tests on remote Docker hosts....108
Setting up a remote Docker host....110
Setting up a secure remote Docker host....111
Copying third-party application Jars to a remote Docker host....113
Registering a remote Docker....115
Adding a remote Docker host....117
Editing configurations....118
Deleting a remote Docker....120
Unregistering a remote Docker....121
Test run configurations....121
Configuring a test for a quick run....122
Configuring an AFT Suite run....123
Configuring an API Suite run....130
Configuring a run of a Compound Test that contains traditional HTML tests....138
Configuring a run of a Compound Test that contains Web UI tests....145
Configuring a run of a Compound Test that contains mobile tests....153
Configuring a run of a Compound Test that contains performance tests....160
Configuring a run of a Rate Schedule or VU Schedule....168
Management of running tests....176
Viewing the progress of running test assets....176
Checking logs....178
Resetting the configuration settings for a test run....179
Stopping a test run....180
Canceling a scheduled test run....182
Management of virtualized services....183
Prerequisites for running stubs....184
Starting a stub....188
Working with HTTP stubs without using proxies....191
Viewing routing rules....193
Viewing stub usage statistics....195
Stopping a stub....199
Test results....200
Test results and reports overview....200
Resource Monitoring service....210
Resource monitoring capabilities....213
Resource Monitoring agents....227

Integration with Jira as a defect management system....237
Task flow for integrating Jira....237
Configuring the Jira server....238
Generating public and consumer keys....240
Creating Jira defects....241
Integrating with other applications....243
Integration plugin compatibility matrix....243
Integration with Azure DevOps....243
Integration with Jenkins....250
Integration with UrbanCode Deploy....257
Managing access to the server....277
**Chapter 7. Test Manager Guide....280**
Test assets and a server project....280
Repository considerations for a server project....280
Adding details to a server project....282
Adding repositories to a server project....282
Secrets configuration....284
Adding users to a server project....284
Becoming a project member....285
Managing access to server projects....286
Archiving or unarchiving server projects....287
Deleting server projects....288
Project overview....289
Working with Git repositories....290
Adding repositories to a server project....293
Protecting API test assets by using secrets....295
Managing secrets collections....296
Granting access to members or member roles....299
Creating a secret in a secrets collection....298
Managing secrets....300
Managing an encrypted dataset....302
Creating a classification....302
Editing or deleting a classification....303
Moving an encrypted dataset to another classification....305
Removing a dataset from the classification....305
Granting classification access to members or members roles....306
Managing notifications....307
Managing in-app notifications....308
Managing email notifications....314
**Chapter 8. Troubleshooting Guide....320**
Troubleshooting issues....320
Security Considerations....cccxxx
Notices....cccxxxii
Index....336

# Chapter 1. Release Notes for IBM® Rational® Test Automation Server

This document includes information about What's new, installation and upgrade instructions, known issues, and contact information of IBM Customer Support.

### Contents


- Description on page 6
- What's new on page 6
- Installing Rational Test Automation Server on page 8
- Known issues on page 8
- Contacting IBM Rational Software Support on page 9


## Description

Rational® Test Automation Server is a server that includes capabilities such as project and role-based security, Docker-based distribution and installation, and running of test cases. For more information about the server, see IBM Rational Test Automation Server overview on page 16.

## What's new

The following sections list the new features, enhancements or other changes made in this release.

- **Documentation**
  - You can now quickly add your comments to provide your feedback about the documentation for Rational® Test Automation Server by posting to a product forum. To add your comments as feedback, you must first register at IBM Support. Your login information along with your comments are governed by the privacy policy of IBM. By commenting, you accept the terms of use of IBM.
- **Installation**
  - You can now install Rational® Test Automation Server on Ubuntu (using k3s) platform. See Installing the server software on Ubuntu by using k3s on page 42.
- **Server licensing**
  - **Enhanced license configuration settings**

    If you are an administrator, the **Configure license** option enables you to change the configuration values such as the value of Rational License Key Server. An administrator can also add, remove, or replace prioritized users. See Configuring floating licenses on page 80.
- **Repository**

- **Refreshing repositories**

  The repositories are refreshed automatically at regular intervals to ensure that each repository is in synchronization with the remote repository. You can also trigger the refresh activity manually for a repository to fetch the changes instantly. See Refreshing repositories manually on page 291.

- **Creating webhooks**

  You can now create a webhook for a repository in Git server. For each push event in the Git server, the webhook notifies Rational® Test Automation Server to trigger the refresh activity to pull the latest changes to the server repository. See Creating webhooks on page 292.

- **Agents**

  - **Viewing capabilities of registered agents**

    Previously, you were able to view the details about the agents. You can now view and ascertain the system and application capabilities of all agents that are registered with Rational® Test Automation Server. With the information presented, you can make informed decisions about selecting an agent for running tests from Rational® Test Automation Server. See Adding an agent to the project for running tests on page 106.

- **Projects**

  - **Managing notifications**

    You can now choose whether to receive in-app, email, or both notifications for each notification channel. By default in-app notifications are enabled. The notifications are about tasks that need completion from users or for information purposes. See Managing notifications.

  - **Configuring notifications**

    As an administrator, you can now set up an SMTP email server to send email notifications. The notifications are about tasks that need completion from the users or for information purposes. See Configuring email notifications.

- **Integrations**

  - **Enhancement of integration plugins**

    - **Azure**

      You can now use the *IBM Rational Test Workbench* extension to establish a trusted and secure connection between Azure DevOps Server and Rational® Test Automation Server to run the Rational® Test Automation Server tests from the Azure DevOps Server. See Integration with Azure DevOps on page 243.

    - **Jenkins**

You can now use the Rational® Test Automation Server Jenkins plugin to establish a trusted and secure connection between the Jenkins server and Rational® Test Automation Server to run the Rational® Test Automation Server tests from the Jenkins server. See Integration with Jenkins on page 250.

- **UrbanCode Deploy**

  You can now use the IBM® Rational® Test Automation Server UCD plugin to establish a trusted and secure connection between the UCD server and Rational® Test Automation Server to run the Rational® Test Automation Server tests from the UCD server. See Integration with UrbanCode Deploy on page 257.

- **Jira software**

  Rational® Test Automation Server is now integrated with Jira software, as a result, you can create a defect in Jira from the Rational® Test Automation Server UI. See Integration with Jira as a defect management system on page 237.

- **Test authoring**

  - **Importing a CSV file with a separator character**

    You can now import a CSV file that has a separator character into the dataset and configure the required separator values in the CSV Editor. See Editing a dataset on page 85.

# Installing Rational® Test Automation Server

- For instructions about installing the software, see Installation of the server software on page 25.
- For instructions about upgrading the software, see Server software upgrade methods on page 47.

# Known issues

You can find information about the known issues identified in this release of Rational® Test Automation Server.

Known problems are documented in the download document for each product and in the form of individual technotes in the Support Knowledge Base:

**Table 1. Download documents and technotes**

| Product | Download document | Knowledge base |
|---|---|---|
| Rational® Test Automation Server | Release document | Knowledge articles |

The knowledge base is continually updated as problems are discovered and resolved. By searching the knowledge base, you can quickly find workarounds or solutions to problems.

# Contacting IBM Rational Software Support

- For contact information and guidelines or reference materials that you might need when you require support, read the IBM Support Guide.
- For personalized support that includes notifications of significant upgrades, subscribe to Product notification.
- Before you contact IBM Rational Software Support, you must gather the background information that you might need to describe your problem. When you describe a problem to an IBM software support specialist, be as specific as possible and include all relevant background information so that the specialist can help you solve the problem efficiently. To save time, know the answers to these questions:
    - What software versions were you running when the problem occurred?
    - Do you have logs, traces, or messages that are related to the problem?
    - Can you reproduce the problem? If so, what steps do you take to reproduce it?
    - Is there a workaround for the problem? If so, be prepared to describe the workaround.

# Chapter 2. System Requirements

This document includes information about hardware and software requirements for IBM® Rational® Test Automation Server.

### Contents


- Hardware on page 11
- Operating systems and Containers on page 11
    - Operating systems on page 12
    - Containers on page 12
- Host prerequisites on page 13
    - Licensing on page 13
    - Web browsers on page 13
- Supported software on page 14
    - Development tools on page 14
    - Runtime environment on page 14
- Disclaimers on page 10


### Disclaimers

This report is subject to the Terms of Use (https://www.ibm.com/legal/us/en/) and the following disclaimers:

The information contained in this report is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied, including but not limited to the implied warranties of merchantability, non-infringement, and fitness for a particular purpose. In addition, this information is based on IBM's current product plans and strategy, which are subject to change by IBM without notice. IBM shall not be responsible for any direct, indirect, incidental, consequential, special or other damages arising out of the use of, or otherwise related to, this report or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

References in this report to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. Product release dates and/or capabilities referenced in this presentation may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. The underlying database used to support these reports is refreshed on a weekly basis. Discrepancies found between reports generated using this web tool and other IBM documentation sources may or may not be attributed to different publish and refresh cycles for this tool and other sources. Nothing contained in this report is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results. You assume sole responsibility for any results you obtain or decisions you make as a result of this report.

Notwithstanding the Terms of Use (https://www.ibm.com/legal/us/en/), users of this site are permitted to copy and save the reports generated from this tool for such users own internal business purpose. No other use shall be permitted.

# Hardware

You can find information about the hardware requirements for IBM® Rational® Test Automation Server.

| Hardware | Platform | Requirement | Notes |
|---|---|---|---|
| Disk space | Container | 256 GB | Additional resources are required by the cluster for the execution of test assets. |
| Memory | Container | 32 GB | Additional resources are required by the cluster for the execution of test assets. |
| Processor | Container | 8 CPUs | Additional resources are required by the cluster for the execution of test assets. |
| Network | Container | Gigabit (1000) Ethernet | |
| Other hardware | Container | | Operating system support is specified by the supported platforms: Kubernetes and OpenShift. Refer to their associated documentation for a list of supported operating systems. |

Related information

System Requirements on page 10

# Operating systems and Containers

You can find details about the supported operating systems and containers.

**Contents**


- Operating systems on page 12
- Containers on page 12

### Bit version support

Different parts of a product might run on the same operating system but support different application bitness.

For example, one part of the product might run only in 32-bit mode, whereas another might support 64-bit tolerate mode.

| Bitness | Description |
|---|---|
| 64-Tolerate | The product or part of the product runs as a 32-bit application in the 64-bit platforms listed as supported. |
| 64-Exploit | The product or part of the product runs as a 64-bit application in the 64-bit platforms listed as supported. |

### Operating systems

| Operating system | Hardware | Bitness | IBM® Rational® Test Automation Server | Notes |
|---|---|---|---|---|
| Ubuntu Server 18.04 LTS | x86-64 | 64-Exploit | ✔ | Running K3s v1.18 |
| Ubuntu Server 20.04 LTS | x86-64 | 64-Exploit | ✔ | Running K3s v1.18 |

### Containers

You can find details about the supported containers.

| Container | Runtime | Version | Notes |
|---|---|---|---|
| IBM® Rational® Test Automation Server | OpenShift Container Platform | 4.4 | The Container Platform that is supported by OpenShift V4.4.3. |

Related information

System Requirements on page 10

# Host prerequisites

You can find the prerequisites that support the operating capabilities for IBM® Rational® Test Automation Server.

### Contents


- Licensing on page 13
- Web browsers on page 13


### Licensing

| License Server | Version | IBM® Rational® Test Automation Server | Notes |
|---|---|---|---|
| IBM Rational License Key Server | 8.1.5 | ✔ | Optional.<br>Required for high-performance load tests. |
| | 8.1.6 | ✔ | Optional.<br>Required for high-performance load tests. |

### Web browsers

| Browser | Version | IBM® Rational® Test Automation Server |
|---|---|---|
| Apple Safari | 12 or later | ✔ |
| Google Chrome | 78 or later | ✔ |
| Microsoft Edge | 80 or later | ✔ |
| Microsoft Internet Explorer | 11 | ✔ |
| Mozilla Firefox (including ESR versions) | 68 or later | ✔ |

Related information

System Requirements on page 10

# Supported software

You can find details about the additional software that are supported.

## Contents


- Development tools on page 14
- Runtime environment on page 14


## Development tools

| Supported software | Version | IBM® Rational® Test Automation Server |
|---|---|---|
| Git | 2.19.0 | ✔ |
| | 2.22.0 | ✔ |

## Runtime environment

| Supported software | Version | IBM® Rational® Test Automation Server | Notes |
|---|---|---|---|
| Helm CLI | 3.3.4 | ✔ | Required locally for Helm. |
| Jaeger Operator | 1.20 | ✔ | Required if tests contribute trace information. The version depends on the target platform. |
| OCP CLI (oc) | 4.4 | ✔ | Required locally for OCP. |
| OpenShift Container Platform (OCP) | 4.4 | ✔ | The Container Platform that is supported by OpenShift V4.4.3. |

| Supported software | Version | IBM® Rational® Test Automation Server | Notes |
|---|---|---|---|
| OpenShift Container Storage (OCS) | 4.4 | ✔ | The storage class must support ReadWriteMany (RWX) so that executions can be performed on all nodes. |

Related information

System Requirements on page 10

# Chapter 3. Getting Started Guide

This guide provides an overview and describes the task flows to get you started with Rational® Test Automation Server. This guide is intended for new users.

## IBM® Rational® Test Automation Server overview

IBM® Rational® Test Automation Server brings together test data, test environments, and test runs and reports into a single, web-based browser for testers and non-testers.

IBM® Rational® Test Automation Server provides the following capabilities:

**Web-based continuous testing platform**

Rational® Test Automation Server is a web-based continuous testing platform built on modern, cloud native technologies that enables test teams to run a breadth of tests that includes API, functional, and performance tests and to benefit from a holistic view of test progress. Rational® Test Automation Server is built on Docker for easy deployment. You can scale your testing on the cloud or remote on premise systems with native Docker, Kubernetes, and IBM Red Hat OpenShift support.

**Role-based access and security**

Security is a key concern for IBM clients and therefore, Rational® Test Automation Server brings a comprehensive, role-based access control scheme to the server with project owners assigning specific member's key permissions (by using roles), for example managing test data or working with secrets such as passwords.

**Running of tests from the server by using Docker containers**

Server-based running of tests is the starting point for Rational® Test Automation Server. For members of a project with the appropriate role, Rational® Test Automation Server enables direct running of tests from the browser by using transient Docker containers.

**Connected agents for existing performance agents**

Agent owners can connect existing performance agents to the server and add them to a project for running schedules and Accelerated Functional Testing (AFT) Suites on the current infrastructure.

**Project overview statistics**

The Overview page for Rational® Test Automation Server offers you a quick, simple view on the state of testing for your projects.

**Project home page**

The home page lists your projects and other projects, which makes it easy to manage different projects within an organization.

**Reporting and the Resource Monitoring Service**

Rational® Test Automation Server provides the home for capabilities that previously were hosted on Rational® Test Control Panel. Reporting and the Resource Monitoring Service are in Rational® Test

Automation Server and provide a more direct relationship with their related projects. These capabilities also benefit from the project level, role-based access controls. You can view unified test results to help you make informed business decisions.

**Test data authoring**

Beyond the concept of a project held in a Git repository for a simple location of tests and related assets, you can do full concurrent editing of test data sets directly from Rational® Test Automation Server. This true multiuser capability enables team members to collaborate more easily as well as try out data changes without impacting the rest of the team. When satisfied with the results, team members can push their changes.

**Integration with DevOps tools**

You can integrate Rational® Test Automation Server with various popular DevOps tools like Jenkins, UrbanCode Deploy, and Microsoft Azure DevOps to get more value from your DevOps pipelines. You can use UrbanCode Deploy to define a deployment process that automatically triggers test cases and have those test insights available directly within UrbanCode Velocity. With the Microsoft Azure DevOps integration, you can also define a DevOps pipeline that includes direct execution of tests by using the Rational® Test Automation Server scalable infrastructure. Rational® Test Automation Server also integrates with Jira for defect tracking and GitHub for software source management and version control.

## Supported versions of assets and resources

You can find information about the versions of assets, resources, agents and Dockers that are supported in Rational® Test Automation Server. You can also find information about the versions of the desktop clients that are supported.

The following table lists the versions of assets or resources created in the desktop clients along with the agents, Dockers, or proxies that are supported in Rational® Test Automation Server:

| Resource | Supported | Not supported |
|---|---|---|
| Assets or resources created in IBM® Rational® Functional Tester | Assets or resources created in Rational® Functional Tester which are of versions earlier or the same as Rational® Test Automation Server. | Assets or resources created in any later version of Rational® Functional Tester that is later than the version of Rational® Test Automation Server. |
| Assets or resources created in IBM® Rational® Integration Tester | Assets or resources created in Rational® Integration Tester which are of versions earlier or the same as Rational® Test Automation Server | Assets or resources created in any later version of Rational® Integration Tester that is later than the version of Rational® Test Automation Server. |
| Assets or resources created in IBM® Rational® Performance Tester | Assets or resources created in Rational® Performance Tester which are | Assets or resources created in any later version of Rational® Performance Tester that is later than the |

| Resource | Supported | Not supported |
|---|---|---|
| | of versions earlier or the same as Rational® Test Automation Server | version of Rational® Test Automation Server. |
| IBM® Rational® Performance Tester Agents | Rational® Performance TesterAgents which is the same version as Rational® Test Automation Server | Before you install the agents that are of the current version, see Agents overview on page 104. |
| Dockers on a remote host computer | Image of Rational® Test Automation Server which is the same version as Rational® Test Automation Server that you want to use to run tests, must be installed on the Docker on a remote host computer. | Before you set up Dockers remotely, see Managing Docker hosts on page 108. |
| HTTP proxies | Installation of the proxy component from Rational® Test Control Panel which is the same version as Rational® Test Automation Server. | Before you use HTTP proxies, see Setting up the HTTP proxy on page 187. |

If there is a mismatch in the versions of the assets, resources, agents, or images used in Dockers with the version of Rational® Test Automation Server that you are using, any of the following events can occur:

- Warnings or Errors are displayed when you add a repository to a project in your team space.
- Warnings or Errors are displayed when you open a project that you migrated from a previous version of Rational® Test Automation Server.
- Warnings or Errors are displayed when you add or manually refresh a repository to which you added assets or resources that were created on a desktop client with a version previous to or later than the version of Rational® Test Automation Server that you want to use.
- Test runs that you start or schedule on an agent or Docker fail to run.
- You cannot view the proxies that are registered with Rational® Test Automation Server on the **Agents and Intercepts** page.

## Tests supported on Rational® Test Automation Server

You can find information about the types of tests that you can configure for a test run in a project on Rational® Test Automation Server. You can also find information about the desktop clients in which you can create the test.

The following table lists the test types that are supported and those tests that are not supported for running on Rational® Test Automation Server:

| Product | Supported test runs | Not supported test runs |
|---|---|---|
| IBM® Rational® Functional Tester | • Accelerated Functional Testing (AFT) Suites<br>• Compound Tests that contain any of the following types of tests:<br>◦ Web UI tests<br>◦ Traditional HTML tests<br>◦ Mobile tests | Traditional non-HTML tests.<br>An independent Web UI test and a mobile test outside of a Compound Test or an AFT Suite. |
| IBM® Rational® Integration Tester | API Suites that include the following Test Suites:<br>• Test Suites that contain scenarios with references to local stubs.<br>• Test Suites that contain stubs that virtualize services run in any Kubernetes cluster or in the Kubernetes cluster that hosts Rational® Test Automation Server.<br>Stubs | Test Suites that have tests with subscribe actions that operate in the watch mode.<br>A stand-alone API test case cannot be run independently. An API test case must be part of a Test Suite. |
| IBM® Rational® Performance Tester | • Compound Tests that contain performance tests<br>• Rate Schedules<br>• VU Schedules | A single test script of any test extension.<br>Tests that belong to 32-bit test extensions and SOA Quality included in VU Schedule, Rate Schedule, and Compound Test. |

**Note:** You must create compound tests for the same type of tests. For example, you can create a compound test that has different mobile tests that use different mobile testing platforms.

**Restriction:** You cannot merge folders that contain different test projects in the same project directory. For example, you cannot create a project directory that contains a Performance test project and an API Suite project. You must create separate project directories for each test type. For example, you must create the

Performance test in the project directory as `/MyPerfProject/` and the API Suite in the project directory as `/MyAPIProject`, before you commit the projects to the remote repository.

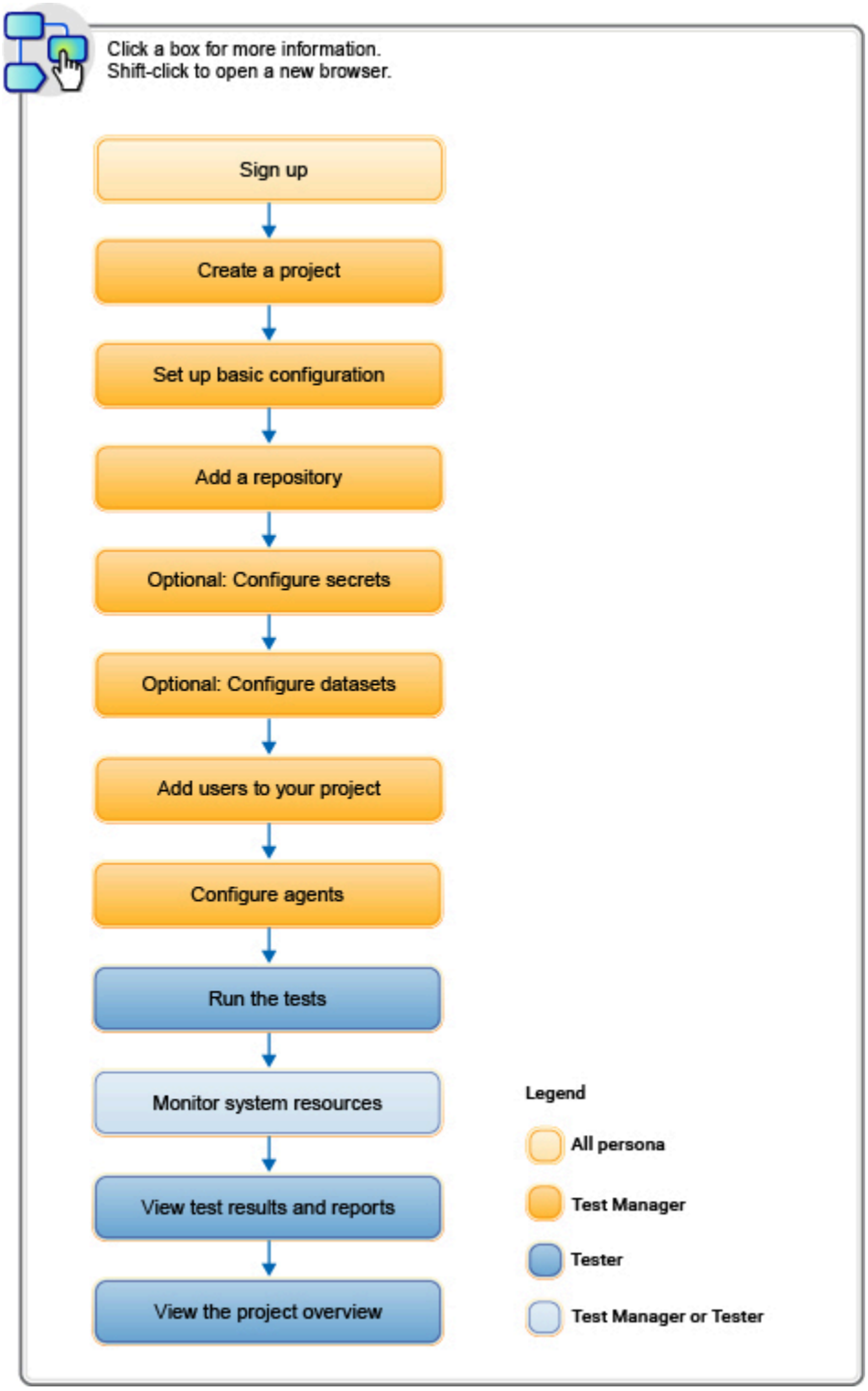
Related information

Tests configurations and test runs

# Task flow: Test runs and results in IBM® Rational® Test Automation Server

You can use the task flow diagram to get started with IBM® Rational® Test Automation Server. After you install the software, you can complete the tasks in sequence to run test assets in Rational® Test Automation Server and view and analyze the test results.

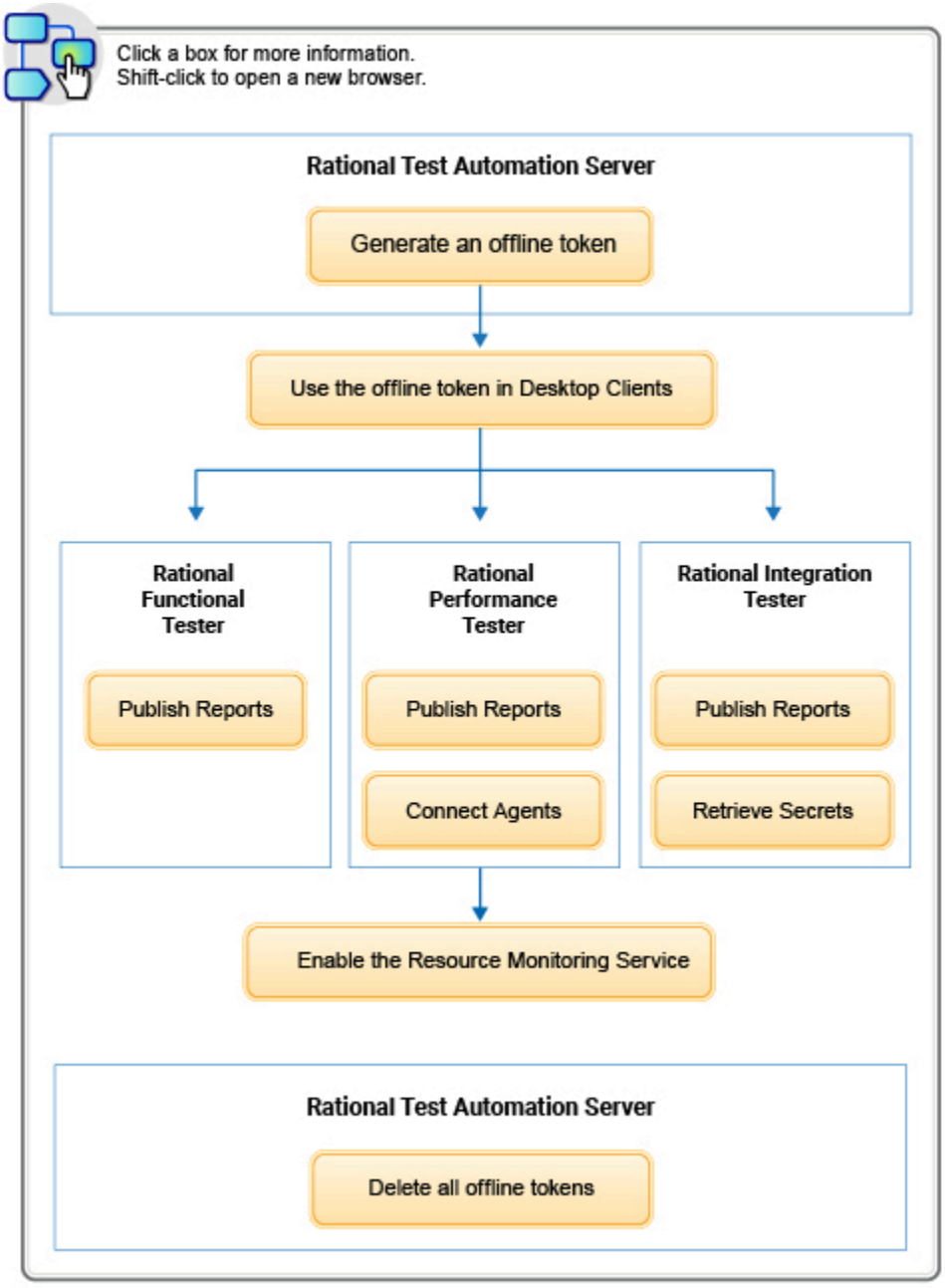
You can click the links to get more information about the tasks. The diagram also provides the tasks to be performed by personas. Rational® Test Automation Server supports user roles.




1. Default user administration on page 58
2. Test assets and a server project on page 280
3. Adding details to a server project on page 282
4. Adding repositories to a server project on page 282
5. Protecting API test assets by using secrets on page 295
6. Managing an encrypted dataset on page 302
7. Adding users to a server project on page 284
8. Adding an agent to the project for running tests on page 106
9. Tests configurations and test runs

# Task flow: IBM® Rational® Test Automation Server access from desktop clients

You can use the task flow diagram to help you access the server from the different desktop clients to retrieve secrets, publish reports, or enable resource monitoring agents. You can perform these tasks after you install and set up the server.




1. Managing access to IBM Rational Test Automation Server on page 277
2. Test results and reports overview on page 200
3. Test results and reports overview on page 200
4. Test results and reports overview on page 200
5. Retrieving secrets
6. Managing access to IBM Rational Test Automation Server on page 277
7. Managing access to IBM Rational Test Automation Server on page 277
8. Installing Rational Performance Tester Agent
9. Enablement of Resource Monitoring services for a schedule

Related information

Managing access to server projects on page 286

# Accessibility features

Accessibility features help users who have physical disabilities, such as visual and, hearing impairment, or limited mobility, to use the software products successfully.

To understand the accessibility compliance status for IBM® Rational® Test Automation Server, refer to Accessibility Conformance Reports.

For more information about IBM and accessibility, refer to IBM Accessibility.

The product documentation is published by using Oxygen XML WebHelp Responsive. To understand the accessibility compliance status for Oxygen XML WebHelp Responsive, refer to WebHelp Responsive VPAT Accessibility Conformance Report.

# Chapter 4. Administrator Guide

This guide describes how to install IBM® Rational® Test Automation Server software. After you install the software, you can perform administration tasks such as license configuration, user management, security, memory, and disk usage management, back up and restore user data. This guide is intended for administrators.

## Installation of the server software

To get started with IBM® Rational® Test Automation Server, you must first install the server software.

You can install Rational® Test Automation Server on the following platforms:

- Red Hat OpenShift V4.4 or later
- Ubuntu (using k3s)

**Note:** As part of the installation process, you can restore user data from a backup file. The backup file might be either from a previous version or the current version of the server software. Depending on the backup file that you have, the procedures for restoring the backups are different.

### Task flow for installing the sever software on Red Hat OpenShift

The following steps describes tasks to be performed in brief to install Rational® Test Automation Server:

1. Set up a Red Hat OpenShift Container Platform.
2. Install the Jaeger operator to trace test logs and Jaeger-based reports when you run tests.
3. Install the OpenShift Container Platform command-line interface (CLI) and the Helm software.
4. Back up the user data from the previous version.
5. Install Rational® Test Automation Server.
6. Configure and manage the license.

### Task flow for installing the sever software on Ubuntu

The following steps describes tasks to be performed in brief to install Rational® Test Automation Server:

1. Set up a Ubuntu server.
2. Install the OpenSSH server and the Helm software.
3. Set up the k3s Kubernetes environment along with other optional configurations such as firewall, Jaeger, and Prometheus server.
4. Back up the user data from the previous version.

5. Install Rational® Test Automation Server.
6. Configure and manage the license.

## Installation of the server software on Red Hat OpenShift

You can find information about the tasks that you can perform to install Rational® Test Automation Server software on the Red Hat OpenShift platform.

## Prerequisites for installing the server software on Red Hat OpenShift

Before you install Rational® Test Automation Server on the Red Hat OpenShift platform, you must first complete certain tasks.

The following section describes each prerequisite in detail:


- Internet access on page 26
- Entitlement key on page 26
- Red Hat OpenShift platform on page 26
- Backup user data on page 27
- OpenShift Kubernetes Engine on page 27
- Mandatory software on page 27
- Persistent storage on page 28
- Service virtualization through Istio on page 29


### Internet access

You must have access to the internet to install Rational® Test Automation Server.

### Entitlement key

You must copy an **Entitlement key** from Container software library to pull the images used by the server software. For more information, refer to Creating an API key.

### Red Hat OpenShift platform

You must set up a Red Hat OpenShift platform.

For more information about hardware and software requirements, see System Requirements on page 10.

### Backup user data

If you want to migrate user data from a previous version of the product during the installation of the server software, then you must perform the following tasks:

- Back up the user data from V10.0.2, Fix Pack 1 or earlier. See Backing up the user data from a previous release.
- Back up the user data from V10.1.0 or later. See Backing up and restoring the user data on Red Hat OpenShift on page 50.

### OpenShift Kubernetes Engine

OpenShift Kubernetes Engine is a product that permits you to use an enterprise-class Kubernetes environment as a production platform to launch containerized applications.

- You must have set up OpenShift Kubernetes Engine and installed the following components in your IT infrastructure:
  - Red Hat OpenShift Container Platform V4.4 or later. For more information, refer to the Red Hat OpenShift documentation.
  - Dynamic Volume Provisioning or manually created Persistent Volumes (PV) of an appropriate size that are already available. For more information, refer to the Dynamic provisioning section in the Red Hat OpenShift documentation.
  - Jaeger operator to trace test logs and Jaeger-based reports when you run tests. For more information, refer to the Installing Jaeger.
  - **Optional**: Red Hat Service Mesh V1.1 or later, when additional service virtualization features are required. For more information, refer to the Red Hat OpenShift Service Mesh overview section in the Red Hat OpenShift documentation.
- You must configure the OpenShift software-defined networking (SDN) to provide a unified cluster network that enables communication between pods across the OpenShift Container Platform cluster. The default installation includes the NetworkPolicy resources and these resources are acted upon only if you configure the SDN appropriately. For more information, refer to the OpenShift SDN default CNI network provider section in the RedHat OpenShift documentation.
- You must have access to the OpenShift cluster with required privileges. You can run the following command on the OpenShift command-line interface to gain access:

```
oc login -u kubeadmin -p {password} https://api.{openshift-cluster-dns-name}:6443
```

### Mandatory software

You must install the following software on Red Hat OpenShift:

- Helm V3.3.4 or later. For more information, refer to the Installing Helm section in the Helm documentation.
- OpenShift CLI. For more information, refer to the Getting started with the CLI section in the Red Hat OpenShift documentation.

### Persistent storage

Rational® Test Automation Server requires persistent storage in addition to the minimum resources required by the cluster. Depending on the workload, you might require additional persistent storage.

For more information about persistent storage, refer to the Persistent Volumes section in the Kubernetes documentation.

You can set the pod `securityContext.fsGroup` for each of the following PersistentVolume by using Helm parameters if your PersistentVolume provisioner requires `securityContext.fsGroup`. The default configuration creates the PersistentVolumeClaim that dynamically provision the PersistentVolume. The following table lists the PersistentVolumeClaim, its size, and access mode during the configuration:

| PersistentVolumeClaim | Size | Access Mode | Content |
|---|---|---|---|
| data-{my-rtas}-datasets-postgresql-0 | 2 GB | ReadWriteOnce<br><br>**CLI Abbreviation:** RWO | Edits to Datasets |
| data-{my-rtas}-execution-post-gresql-0 | 2 GB | RWO | Infrastructure details |
| data-{my-rtas}-gateway-postgresql-0 | 2 GB | RWO | Project details |
| data-{my-rtas}-rabbitmq-0 | 8 GB | RWO | Notifications between services |
| data-{my-rtas}-keycloak-postgresql-0 | 2 GB | RWO | Users and resource ownership |
| data-{my-rtas}-results-0 | 8 GB | RWO | Reports |
| data-{my-rtas}-results-postgresql-0 | 2 GB | RWO | Execution result metadata |
| data-{my-rtas}-rm-postgresql-0 | 2 GB | RWO | Source details |
| data-{my-rtas}-tam-0 | 32 GB | RWO | Cloned git repositories |
| data-{my-rtas}-tam-postgresql-0 | 8 GB | RWO | Test asset metadata |
| data-{my-rtas}-userlibs-0 | 1 GB | ReadWriteMany<br><br>**CLI Abbreviation:** RWX | User provided third party li-braries |

**Note:** `{my-rtas}` is the release name that you provide during the installation of the server software.

### Service virtualization through Istio

The default configuration does not enable service virtualization through Istio. To use the service virtualization feature, you must configure it appropriately and it must be installed by a cluster-admin. For more information about how to enable service virtualization through Istio, see Installing the server software on Red Hat OpenShift on page 31.

**Disclaimer:** This release contains access to the virtualized services feature in Rational® Test Automation Server as a Tech Preview. The Tech Preview is intended for you to view the capabilities of virtualized services offered by Rational® Test Automation Server V10.1.2, and to provide your feedback to the product team. You are permitted to use the information only for evaluation purposes and not for use in a production environment. IBM provides the information without obligation of support and "as is" without warranty of any kind.

## Installing the server software on Red Hat OpenShift

You can install Rational® Test Automation Server on the Red Hat OpenShift server that has the Kubernetes Engine environment to run functional, integration, and performance tests. Rational® Test Automation Server combines test data, test environments, and test runs and reports into a single web-based browser for testers and non-testers.

**Before you begin**

You must have performed the following tasks:

- Completed tasks provided in the Prerequisites section. See Prerequisites for installing the server software on Red Hat OpenShift on page 26.
- Ensured that you have installed the following software:
  - OpenShift CLI
  - Helm
- Copied the **Entitlement key** from the Container software library.

**About this task**

As part of the installation process, you might optionally migrate user data from a previous version (V10.0.2, Fix Pack 1 or earlier) the product, or restore a backup of user data from the current version (V10.1.0 or later) of the product. You must perform different steps, depending on which version of the user data backup file you are applying during the installation process. For more details, see the following table:

| When you backed up user data from... | Then to restore... |
|---|---|
| Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier | Perform step 6 on page 30. |
| Rational® Test Automation Server V10.1.0 or later | Perform step 9 on page 35. |

1. Open and log in to the terminal.
2. Create a namespace to install the server software by running the following command:

```
oc new-project test-system
```

3. Add the repository to Helm to access the server install charts by running the following command:

```
helm repo add ibm-helm https://raw.githubusercontent.com/IBM/charts/master/repo/ibm-helm
```

4. Run the following command to get the latest updates from the repository:

```
helm repo update
```

5. Create a Secret to pull images that are used by Rational® Test Automation Server by running the following commands:

**Notes:**

- You must substitute `{your_entitlement_key}` with the key that you copied from Container software library.
- You can replace `example@abc.com` with the administrator email address, if required.

```
oc create secret docker-registry cp.icr.io \
  -n test-system \
  --docker-server=cp.icr.io \
  --docker-username=cp \
  --docker-password={your_entitlement_key} \
  --docker-email=example@abc.com
```

6. **Optional:** Perform the following steps to migrate user data from the previous version (V10.0.2, Fix Pack 1 or earlier):

   a. Run the following commands to start a pod and to migrate the user data:

```
#Retrieve the backup yaml file
helm repo update
helm pull --untar ibm-helm/ibm-rtas-prod --version 3.1012.0

#Modify the backup yaml file with the release name for your server by replacing {my-rtas}
sed -i 's/{{ \.Release\.Name }}/{my-rtas}/g' ibm-rtas-prod/files/import-prek8s-backup.yaml

#Update the runAsUser to match scc policy
sed -i -e "s/1001/$(oc get project test-system -oyaml \
  | sed -r -n 's# *openshift.io/sa.scc.uid-range: *([0-9]*)/.*#\1#p')/g"
 ibm-rtas-prod/files/import-prek8s-backup.yaml
```

```
#Add the fsGroup settings
sed -i 's/\(.*\)\(runAsUser:\)\(.*\)/\1\2\3\n\1fsGroup:\3/'
 ibm-rtas-prod/files/import-prek8s-backup.yaml

#To fix slow network issues
sed -i 's/sleep 5/sleep 30/' ibm-rtas-prod/files/import-prek8s-backup.yaml

#Apply the backup yaml file to start the import-prek8s-backup pod
oc apply -f ibm-rtas-prod/files/import-prek8s-backup.yaml -n test-system
```

b. Run the following command to display the import-prek8s-backup pod log.

```
oc logs import-prek8s-backup -n test-system
```

c. Follow the instructions in the log file to copy your backup file to a location where the pod is waiting to process it.

**Result**

When the user data backup is migrated, the import-prek8s-backup pod changes from `Running` to `Completed` state.

**Note:** You can run the `oc get pod import-prek8s-backup -n test-system` command to view the pod state.

d. Add the additional Helm parameters that the log file displays when the state of the pod is `Completed`, to step 8 on page 33.

The following is an example that shows the additional Helm parameters:

```
--set keycloak.keycloak.username=admin \
 --set execution.existingPostgresqlPassword=SwztX9JRO/yL9jtgioUiKMGjlsWfM7Ok \
 --set existingKeycloakPassword=jqdV7aveCa3WShkdSF/Z8hAasBupsvp6 \
 --set existingKeycloakPostgresqlPassword=pKVFQUQWRU7BJC6sKeOZIm8zCzLkbXM7 \
 --set existingOauthClientSecret=EMRpHBRurFIMB9X+nfwApnLBBpkvn8+z \
 --set existingPostgresqlPassword=8HY3nADjTZBC2lLj0i7Y7v6wHeEji9ds \
 --set existingSecretsMasterPassword=ibIPh6wGAv0IjsveOrYX1TRwo3ChvxWx \
 --set results.existingPostgresqlPassword=7X3aaAVeHxdzJ+BlFTG7G7rypzsbhd6D \
 --set rm.existingPostgresqlPassword=3uaEx5wTja+PSzvG64O1+cpO3glGAaVt \
 --set tam.existingPostgresqlPassword=7JO95keyZ/F9w5rBcXj2YZFKFf/aAK1F \
```

7. **Optional:** Perform the following actions described in the following table to enable the additional features on Rational® Test Automation Server:

| Features | Descriptions | Action to be perform |
|---|---|---|
| Service virtualization | To enable the running of stubs that virtualize services in the RedHat OpenShift Service Mesh | Perform the following steps: |

| Features | Descriptions | Action to be perform |
|---|---|---|
| | | a. Add the following parameter to the `helm install` command:<br>`--set execution.istio.enabled=true`<br>b. Run the following command to create the role bindings required for Istio:<br>`oc create rolebinding istio-virtualization-enabled -n {namespace} \ --clusterrole={my-rtas}-execution-istio-test-system --serviceaccount=test-system:{my-rtas}-execution`<br>**Note:** When you uninstall the chart, the manually created role bindings are not deleted from the namespace. |
| | To enable the running of stubs that virtualize services in namespaces | Add the following parameter to the `helm install` command:<br>`--set execution.istio.namespaces='{namespaceA,namespaceB}' \`<br>Alternatively, you can add the parameter to the `helm install` command by using an array index notation as follows:<br>`--set execution.istio.namespaces[0]=namespaceA \` |

| Features | Descriptions | Action to be perform |
|---|---|---|
| | | `--set execution.istio.namespaces[1]=namespaceB` |
| Jaeger | To use Jaeger for performance and Web UI tests logs | Add the following parameter to the `helm install` command:<br><br>**Note:** You must substitute `{my-jaeger-dashboard-url}` with the URL of the Jaeger server.<br><br>`--set-string execution.annotations.sidecar\\.jaegertracing\\.io/inject=true --set global.jaegerAgent.enabled=true --set global.jaegerAgent.internalHostName=localhost --set global.jaegerDashboard.enabled=true --set global.jaegerDashboard.externalURL={my-jaeger-dashboard-url}` |

8. Run the following commands to install the server software:

   You must substitute the value of the following variables with the actual value in the command:

   - `{my-rtas}` with the release name of your choice.

     **Note:** If you migrated user data from a previous release (V10.0.2, Fix Pack 1), then you must use the same `{my-rtas}` name that you used in step 7a on page 30.

   - `{openshift-cluster-dns-name}` with the ingress DNS name that you selected for the server.

     **Note:** You can run the following command to obtain the default value of `openshift-cluster-dns-name`:

```
oc get --namespace=openshift-ingress-operator ingresscontroller/default
 -ojsonpath='{.status.domain}'
```

**Remember:** You must provide the value that consists of lowercase alphanumeric characters, "-"(hyphen) or "."(period). Also, the value must start and end with an alphanumeric character.

- **Optional**: `{rlks-ip-address}` with the IP address of Rational License Key Server.

  Alternatively, you can configure the **Rational License Key Server** value from the **Configure License** window when the installation of the server is complete.

- `{my-super-secret}` with a value of your choice.

  **Important:** This password seed is used to create several default passwords for the server. You must store the password seed securely. When you install the server software by using the backup of the user data, you can reuse the password seed. You can use this seed to restore backup files either on the current or later versions of the server software.

**Note:** The default storage class must support both the ReadWriteOnce (RWO) and ReadWriteMany (RWX) access modes. If the default storage class does not support ReadWriteMany (RWX), you must add the following parameter to the helm install command:

```
--set global.persistence.rwxStorageClass=ibmc-file-gold
```

```
#Make sure the repository is current and retrieve the charts required to install the server
#You can skip the following two commands if you have migrated from the previous version (V10.0.2,
 Fix Pack 1 or earlier)
helm repo update
helm pull --untar ibm-helm/ibm-rtas-prod --version 3.1012.0

#Update the runAsUser and fsGroup to match scc policy
sed -i -e "s/runAsUser: 1001/runAsUser: $(oc get project test-system -oyaml \
  | sed -r -n 's# *openshift.io/sa.scc.uid-range: *([0-9]*)/.*#\1#p')/g;
          s/fsGroup: 1001/fsGroup: $(oc get project test-system -oyaml \
  | sed -r -n 's# *openshift.io/sa.scc.supplemental-groups: *([0-9]*)/.*#\1#p')/g"
 ibm-rtas-prod/values-openshift.yaml

#Install the server
helm install {my-rtas} ./ibm-rtas-prod -n test-system \
--set license=true \
-f ibm-rtas-prod/values-openshift.yaml \
--set global.ibmRtasIngressDomain=rtas.{openshift-cluster-dns-name} \
--set global.rationalLicenseKeyServer=@{rlks-ip-address} \
--set global.ibmRtasPasswordAutoGenSeed={my-super-secret} \
--set global.ibmRtasRegistryPullSecret=cp.icr.io
```

**Tip:** After you installed the server software, anytime if you want to enable or disable any specific features on Rational® Test Automation Server you can do so by using the `helm upgrade` command.

For example, if you want to enable service virtualization feature to run stubs that virtualize services in namespaces, you can run the following commands:

```
helm upgrade {my-rtas} -n test-system ./ibm-rtas-prod \
  --reuse-values \
--set execution.istio.namespaces='{namespaceA,namespaceB}'
```

9. **Optional:** Run the following command to restore the backed up user data from the current version:

```
velero restore create --from-backup=<backup-file-name> --restore-volumes
```

**Note:** You must replace `<backup-file-name>` with the name of the back up file that you saved.

10. Run the following command to verify and test the installed server software:

```
$ helm test {my-rtas} -n {namespace}
```

where:

- `{my-rtas}` is the release name that you provide during the installation of the server software.
- `{namespace}` is the name of the namespace that you created during the server installation process.

**Results**

On successful installation of Rational® Test Automation Server, the output displays the URL to access the Rational® Test Automation Server UI.

**What to do next**

- You can back up the user data that are saved in the Kubernetes clusters to secure your data. See Backing up and restoring the user data on Red Hat OpenShift on page 50.
- You can configure the license to use Rational® Test Automation Server. See Configuring floating licenses on page 80.

## Installation of the server software on Ubuntu

You can find information about the tasks that you can perform to install Rational® Test Automation Server software on the Ubuntu platform.

## Prerequisites for installing the server software on Ubuntu

Before you install Rational® Test Automation Server on the Ubuntu platform, you must first complete certain tasks.

The following section describes each prerequisite in detail:

- Internet access on page 36
- Entitlement key on page 36
- Ubuntu server on page 36
- Backup user data on page 36
- Mandatory software on page 37
- Optional software on page 37
- Non-root user on page 37
- k3s environment on page 38
- Persistent storage on page 38
- Service virtualization through Istio on page 39


### Internet access

You must have access to the internet to install Rational® Test Automation Server.

### Entitlement key

You must copy an **Entitlement key** from Container software library to pull the images used by the server software. For more information, refer to Creating an API key.

### Ubuntu server

You must set up an Ubuntu server.

For more information about hardware and software requirements, see System Requirements on page 10.

**Note:** Depending on your testing workload, Rational® Test Automation Server might require more resources. You must use the entire disk space and set up Logical Volume Manager (LVM) by using the `ext4` file system. If your organization requires application data to be stored in a separate partition, then you can create a mount point at `/var/lib/rancher/k3s/storage/` with at least 128 GB capacity.

### Backup user data

If you want to migrate user data from a previous version of the product during the installation of the server software, then you must perform the following tasks:

- Back up the user data from V10.0.2, Fix Pack 1 or earlier. See Backing up the user data from a previous release.
- Back up the user data from V10.1.0 or later. See Backing up and restoring the user data on Ubuntu on page 51.

### Mandatory software

You must install the following mandatory software:

- Helm V3.3.4 or later. For more information, refer to the Installing Helm section in the Helm documentation.

  **Note:** You must ensure that you have access to the Kubernetes cluster with required privileges before installing Helm. You can contact your cluster administrator to gain access to the cluster. For more information, refer to the Accessing Clusters section in the Kubernetes documentation.

- OpenSSH Server. For more information, refer to the Installation section in the OpenSSH Server documentation.

### Optional software

You can install the following optional software:

- Jaeger operator to trace test logs and Jaeger-based reports when you run tests. For more information, refer to the Jaeger operator documentation.
- Prometheus server to monitor your system resources by using metrics data that are collected by a Prometheus server. For more information, refer to the Installation section in the Prometheus documentation.

**Tip:** Alternatively, you can also install Jaeger and Prometheus server as part of the k3s environment setup. See Setting up a Kubernetes environment on Ubuntu on page 39.

### Non-root user

You can use the `sudo` command to execute a single command as the root user when needed. If you do not want to enter a password each time when you run the `sudo` command, you can enable the passwordless `sudo` environment by performing the following steps:

1. Run the `sudo visudo` command to modify the `sudo` settings.
2. Locate the `%sudo ALL=(ALL:ALL) ALL` and modify it to `%sudo ALL=(ALL) NOPASSWD:ALL` in the file.
3. Save your changes, and then close the editor.

### k3s environment

You must set up the k3s Kubernetes environment along with other optional configurations such as firewall, Jaeger, and Prometheus server. See Setting up a Kubernetes environment on Ubuntu on page 39.

### Persistent storage

Rational® Test Automation Server requires persistent storage in addition to the minimum resources required by the cluster. Depending on the workload, you might require additional persistent storage.

You can set the pod `securityContext.fsGroup` for each of the following PersistentVolume by using Helm parameters if your PersistentVolume provisioner requires `securityContext.fsGroup`. The default configuration creates the PersistentVolumeClaim that dynamically provision the PersistentVolume. The following table lists the PersistentVolumeClaim, its size, and access mode during the configuration:

| PersistentVolumeClaim | Size | Access Mode | Content |
|---|---|---|---|
| data-{my-rtas}-datasets-postgresql-0 | 2 GB | ReadWriteOnce<br><br>**CLI Abbreviation:** RWO | Edits to Datasets |
| data-{my-rtas}-execution-post-gresql-0 | 2 GB | RWO | Infrastructure details |
| data-{my-rtas}-gateway-postgresql-0 | 2 GB | RWO | Project details |
| data-{my-rtas}-rabbitmq-0 | 8 GB | RWO | Notifications between services |
| data-{my-rtas}-keycloak-postgresql-0 | 2 GB | RWO | Users and resource ownership |
| data-{my-rtas}-results-0 | 8 GB | RWO | Reports |
| data-{my-rtas}-results-postgresql-0 | 2 GB | RWO | Execution result metadata |
| data-{my-rtas}-rm-postgresql-0 | 2 GB | RWO | Source details |
| data-{my-rtas}-tam-0 | 32 GB | RWO | Cloned git repositories |
| data-{my-rtas}-tam-postgresql-0 | 8 GB | RWO | Test asset metadata |
| data-{my-rtas}-userlibs-0 | 1 GB | ReadWriteMany<br><br>**CLI Abbreviation:** RWX | User provided third party li-braries |

**Note:** `{my-rtas}` is the release name that you provide during the installation of the server software.

### Service virtualization through Istio

The default configuration does not enable service virtualization through Istio. To use the service virtualization feature, you must configure it appropriately and it must be installed by a cluster-admin. For more information about how to enable service virtualization through Istio, see Installing the server software on Ubuntu by using k3s on page 44.

**Disclaimer:** This release contains access to the virtualized services feature in Rational® Test Automation Server as a Tech Preview. The Tech Preview is intended for you to view the capabilities of virtualized services offered by Rational® Test Automation Server V10.1.2, and to provide your feedback to the product team. You are permitted to use the information only for evaluation purposes and not for use in a production environment. IBM provides the information without obligation of support and "as is" without warranty of any kind.

## Setting up a Kubernetes environment on Ubuntu

You can use the readily available scripts to set up a k3s Kubernetes environment along with other configurations to install Rational® Test Automation Server software in the Ubuntu server.

**Before you begin**

- You must ensured that you have installed the following software:
  - OpenSSH server
  - Helm
- You must ensure that your computer has a static IP address or Domain Name Server (DNS) resolvable host name to resolve host name into a machine-readable IP address.

1. Log in to the Ubuntu server as a non-root user.
2. Run the following command to retrieve the case from IBM Cloud Pack repository:

```
curl -o ibm-rtas-case-3.1012.0.tgz
 https://raw.githubusercontent.com/IBM/cloud-pak/master/repo/case/ibm-rtas-case-3.1012.0.tgz
```

3. Run the following command to extract the case:

```
tar xvf ibm-rtas-case-3.1012.0.tgz
```

4. Run any of the following commands to install the Kubernetes environment:

   **Note:** The second option allows you to override the name of the Kubernetes domain that is created.

   **Choose from:**

- Run the following commands to install with the default name that is either based on IP address or fully qualified host name:

```
#Run the following commands if you are on Ubuntu 18.04
$ cd ibm-rtas-case/inventory/ibmRtasBase/files
$ sudo ./ubuntu-init.sh

#Run the following commands if you are on Ubuntu 20.04
$ cd ibm-rtas-case/inventory/ibmRtasBase/files
$ sudo HOME=$HOME ./ubuntu-init.sh
```

- Run the following commands to install by overriding the default name:

```
$ cd ibm-rtas-case/inventory/ibmRtasBase/files
$ sudo INGRESS_DOMAIN={rtas.}myorg.com ./ubuntu-init.sh
```

where:

- *{rtas.}* is a sub-domain name that you specified for the server. For example, `testenv.`

  **Note:** The sub-domain must consist of lowercase alphanumeric characters, "-"(hyphen) or "."(period). Also, the value must start and end with an alphanumeric character.

- *myorg.com* is a domain name of your organization. For example, `ibm.com`

You can access the product through a web browser by using any of the following ways:

- **Fully Qualified Hostname**: When the server is configured, you can use `hostname -f` command to get the fully qualified hostname defined in the DNS to access Rational® Test Automation Server. For example, `rtas.myorg.com`
- **IP address**: You can use the IP address to access Rational® Test Automation Server when you cannot create a specific DNS record for the server. For example, `ip-address.nip.io`

**Result**

On completion of the scripts, the output displays the following information on the command-line interface:

- The INGRESS_DOMAIN that is in use. This is the URL from where you can access Rational® Test Automation Server. You must use this value for the **global.ibmRtasIngressDomain** parameter in the step 9 on page 45 in the server installation topic.
- The DNS information that the Kubernetes cluster uses to resolve names.
- Certificate Authority (CA) certificate that you must be import into the browser to prevent certificate errors.
- Instructions to confirm whether the Kubernetes environment has started.

5. Perform one of the following to configure a firewall:

**Choose from:**

- Run the following script to configure the firewall that allows traffic on *cni0* and port *443*:

```
#Run the following command if you are on Ubuntu 18.04
$ sudo ./ubuntu-firewall.sh

#Run the following command if you are on Ubuntu 20.04
$ sudo HOME=$HOME ./ubuntu-firewall.sh
```

- You must update the firewall that allows traffic on *cni0* and port *443*, if your Ubuntu server is already configured with the firewall.

6. **Optional:** Run the following command to enable the Jaeger traces for performance and Web UI tests:

```
$ ./service.sh expose jaeger
```

**Note:** If you do not enable Jaeger, Rational® Test Automation Server produces text output in a microservice log file instead of Jaeger traces when you run performance and Web UI test assets.

**Important:** The Jaeger traces are not protected, thus, any information logged into the Jaeger server might be easily accessible by anyone who has or discovers the *<server-url>/jaeger* URL.

7. **Optional:** Run the following command to enable the Prometheus server to monitor your system resources by using metrics data:

```
$ ./service.sh expose prometheus
```

**Important:** The Prometheus metrics are not protected, thus, any information logged into the Prometheus server might be easily accessible by anyone who has or discovers the *<server-url>/prometheus* URL.

**Results**

You have set up the Kubernetes environment on Ubuntu.

**What to do next**

- You must log in again to the server host system after the installation process completes so that changes to the group membership are applied.
- You can run the `kubectl get pods -A` command to verify that the Kubernetes environment is working. After a while, the status of the pods must be `Running` state.
- You can install the server software. See Installing the server software on Ubuntu by using k3s on page 42.

Related information

Jaeger Operator

Prometheus server

## Installing the server software on Ubuntu by using k3s

You can install Rational® Test Automation Server on the Ubuntu server that has a Kubernetes environment to run functional, integration, and performance tests. Rational® Test Automation Server combines test data, test environments, and test runs and reports into a single, web-based browser for testers and non-testers.

**Before you begin**

- Completed tasks provided in the Prerequisites section. See Prerequisites for installing the server software on Ubuntu on page 35.
- Ensured that you have installed the following software:
  - OpenSSH server
  - Helm
- Know the following values to be used during the installation process:
  - The **Entitlement key** from the Container software library.
  - The ingress DNS name that displayed on completion of the `ubuntu-init.sh` script.

**About this task**

As part of the installation process, you might optionally migrate user data from a previous version (V10.0.2, Fix Pack 1 or earlier) the product, or restore a backup of user data from the current version (V10.1.0 or later) of the product. You must perform different steps, depending on which version of the user data backup file you are applying during the installation process. For more details, see the following table:

| When you backed up user data from... | Then to restore... |
|---|---|
| Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier | Perform step 7 on page 43. |
| Rational® Test Automation Server V10.1.0 or later | Perform step 10 on page 46. |

1. Log in to the Ubuntu server as a non-root user.
2. Create a namespace to install the server software by running the following command:

   ```
   kubectl create namespace test-system
   ```

3. Add the entitlement registry to Helm to access the server install charts by running the following command:

```
helm repo add ibm-helm https://raw.githubusercontent.com/IBM/charts/master/repo/ibm-helm
```

4. Run the following command to get the latest updates from the repository:

```
helm repo update
```

5. Create a Secret to pull images that are used by Rational® Test Automation Server by running the following command:

**Notes:**

- You must substitute `{your_entitlement_key}` with the key that you copied from Container software library.
- You can replace `example@abc.com` with the administrator email address, if required.

```
kubectl create secret docker-registry cp.icr.io \
  -n test-system \
  --docker-server=cp.icr.io \
  --docker-username=cp \
  --docker-password={your_entitlement_key} \
  --docker-email=example@abc.com
```

6. Run the following command to add the Certificate Authority (CA) certificate into a Secret:

```
kubectl create secret generic -n test-system ingress \
"--from-literal=ca.crt=$(kubectl get -n istio-system secret istio-ingressgateway-certs
 -ojsonpath='{.data.ca\.crt}' | base64 --decode)"
```

When some components such as static agents or Docker agents want to communicate with Rational® Test Automation Server, the component presents its certificate to the server to verify its identity. Rational® Test Automation Server trusts the component only if it is signed by a recognized and trusted CA. To enable certificates as trusted certificates, you must add the signed CA into a trust by placing it in a Secret.

7. **Optional:** Perform the following steps to migrate user data from the previous version (V10.0.2, Fix Pack 1 or earlier):

   a. Run the following commands to start a pod and to migrate user data:

```
#Retrieve the backup yaml file
helm repo update
helm pull --untar ibm-helm/ibm-rtas-prod --version 3.1012.0

#Modify the backup yaml file with the release name for your server by replacing {my-rtas}
sed -i 's/{{ \.Release\.Name }}/{my-rtas}/g' ibm-rtas-prod/files/import-prek8s-backup.yaml

#Apply the backup yaml file to start the import-prek8s-backup pod
kubectl apply -f ibm-rtas-prod/files/import-prek8s-backup.yaml -n test-system
```

   b. Run the following command to display the import-prek8s-backup pod log.

```
kubectl logs import-prek8s-backup -n test-system
```

c. Follow the instructions in the log file to copy your backup file to a location where the pod is waiting to process it.
**Result**

When the user data backup is migrated, the import-prek8s-backup pod changes from `Running` to `Completed` state.

**Note:** You can run the `kubectl get pod import-prek8s-backup -n test-system` command to view the pod state.

d. Add the additional Helm parameters that the log file displays when the state of the pod is `Completed`, to the install step 9 on page 45.

The following is an example that shows the additional Helm parameters:

```
--set keycloak.keycloak.username=admin \
 --set execution.existingPostgresqlPassword=SwztX9JRO/yL9jtgioUiKMGjlsWfM7Ok \
 --set existingKeycloakPassword=jqdV7aveCa3WShkdSF/Z8hAasBupsvp6 \
 --set existingKeycloakPostgresqlPassword=pKVFQUQWRU7BJC6sKeOZIm8zCzLkbXM7 \
 --set existingOauthClientSecret=EMRpHBRurFIMB9X+nfwApnLBBpkvn8+z \
 --set existingPostgresqlPassword=8HY3nADjTZBC2lLj0i7Y7v6wHeEji9ds \
 --set existingSecretsMasterPassword=ibIPh6wGAv0IjsveOrYX1TRwo3ChvxWx \
 --set results.existingPostgresqlPassword=7X3aaAVeHxdzJ+BlFTG7G7rypzsbhd6D \
 --set rm.existingPostgresqlPassword=3uaEx5wTja+PSzvG64O1+cpO3glGAaVt \
 --set tam.existingPostgresqlPassword=7JO95keyZ/F9w5rBcXj2YZFKFf/aAK1F \
```

8. **Optional:** Perform the following actions described in the following table to enable the additional features on Rational® Test Automation Server:

| Features | Descriptions | Action to be perform |
|---|---|---|
| Service virtualization | To enable services in the Istio Service Mesh to be virtualized as stubs | Add the following parameter to the `helm install` command:<br>`--set execution.istio.enabled=true \`<br>`--set execution.istio.clusterRoleBinding.create=true` |
| | To enable the running of stubs that virtualize services in namespaces | Add the following parameter to the `helm install` command:<br>`--set execution.istio.namespaces='{namespaceA,namespaceB}' \`<br>Alternatively, you can add the parameter to the `helm install` command by using an array index notation as follows: |

| Features | Descriptions | Action to be perform |
|---|---|---|
| | | `--set execution.istio.namespaces[0]=namespaceA \`<br>`--set execution.istio.namespaces[1]=namespaceB` |
| Jaeger | To use Jaeger for performance and Web UI tests logs | Add the following parameter to the `helm install` command:<br><br>**Note:** You must substitute `{my-jaeger-dashboard-url}` with the URL of the Jaeger server.<br><br>`--set-string`<br>`execution.annotations.sidecar\\.jaegertracing\`<br>`\.io/inject=true`<br>`--set global.jaegerAgent.enabled=true`<br>`--set global.jaegerAgent.internalHostName=localhost`<br>`--set global.jaegerDashboard.enabled=true`<br>`--set`<br>`global.jaegerDashboard.externalURL={my-jaeger-dashboard-url}` |
| NodePort | To run performance tests on the Static agents | Add the following parameter to the `helm install` command:<br><br>`--set networkPolicy.enabled=false` |

9. Run the following commands to install the server software on your computer:

   You must substitute the value of the following variables with the actual value in the command:

   - `{my-rtas}` with the release name of your choice.

     **Note:** If you migrated user data from a previous release (V10.0.2, Fix Pack 1), then you must use the same `{my-rtas}` name that you used in step 7a on page 43.

   - `{my-ingress-dns-name}` with the same ingress DNS name that displayed on completion of the `ubuntu-init.sh` script.

   - `{my-super-secret}` with a value of your choice.

     **Important:** This password seed is used to create several default passwords for the server. You must store the password seed securely. When you install the server software by using the

backup of the user data, you can reuse the password seed. You can use this seed to restore backup files either on the current or later versions of the server software.

- **Optional**: `{rlks-ip-address}` with the IP address of Rational License Key Server.

  Alternatively, you can configure the **Rational License Key Server** value from the **Configure License** window when the installation of the server is complete.

```
#Make sure the repository is current and retrieve the charts required to install the server
#You can skip the following two commands if you have migrated from the previous version (V10.0.2,
 Fix Pack 1 or earlier)
helm repo update
helm pull --untar ibm-helm/ibm-rtas-prod --version 3.1012.0

#Install the server
helm install {my-rtas} ./ibm-rtas-prod -n test-system \
--set license=true \
-f ibm-rtas-prod/values-k3s.yaml \
--set global.ibmRtasIngressDomain={my-ingress-dns-name} \
--set global.rationalLicenseKeyServer=@{rlks-ip-address} \
--set global.ibmRtasPasswordAutoGenSeed={my-super-secret} \
--set global.ibmRtasRegistryPullSecret=cp.icr.io
```

**Tip:** After you installed the server software, anytime if you want to enable or disable any specific features on Rational® Test Automation Server you can do so by using the `helm upgrade` command.

For example, if you want to enable service virtualization feature to run stubs that virtualize services in namespaces, you can run the following commands:

```
helm upgrade {my-rtas} -n test-system ./ibm-rtas-prod \
  --reuse-values \
--set execution.istio.namespaces='{namespaceA,namespaceB}'
```

10. **Optional:** Perform the following steps to restore the backed up user data from the current version:

    a. Stop Kubernetes by running the `k3s-killall.sh` command.

    b. Restore the backed-up user data by running the following commands:

```
cd ibm-rtas-case/inventory/ibmRtasBase/files
sudo ./backup.sh restore [options] <backup-file-name>
```

    **Note:** You must replace `<backup-file-name>` with the name of the back up file that you saved.

    c. Restart Kubernetes by running the `systemctl start k3s` command to start Rational® Test Automation Server.

11. Run the following command to verify and test the installed server software:

```
$ helm test {my-rtas} -n {namespace}
```

where:

- `{my-rtas}` is the release name that you provide during the installation of the server software.
- `{namespace}` is the name of the namespace that you created during the server installation process.

**Results**

On completion of the installation of server software, the output displays the following information on the command-line interface:

- Instructions to access Keycloak to manage and authenticate users.

  You can use the username as *keycloak* and password can be retrieved by running the following command:

  ```
  kubectl get secret -n namespace rtas-keycloak-postgresql -o jsonpath="{.data.password}" | base64
   --decode; echo
  ```

  where:

  - *rtas* is a sub-domain name that you select for the server.
  - *namespace* is the name of the namespace that you created.

- The URL to access the Rational® Test Automation Server UI.

**What to do next**

- You can back up the user data that are saved in the Kubernetes clusters to secure your data. See Backing up and restoring the user data on Ubuntu on page 51.
- You can configure the license to use Rational® Test Automation Server. See Configuring floating licenses on page 80.

---

Related information

Helm documentation

# Server software upgrade methods

When you want to use the enhanced functionalities of Rational® Test Automation Server, you must upgrade to the latest version of the server software.

You can upgrade the server software by using two methods:

- **In-place upgrade:** In this method, you can upgrade the server software in the same location that contained the V10.1.0 or later version of server software by using the `helm upgrade` command.
- **Upgrade (by new install):** In this method, you can upgrade the server software by following one of the two methods, depending upon the existing version of the server software:

- For V10.0.2, Fix Pack 1 or earlier: Back up the user data, and restore it as part of the installation process.
- For V10.1.0 or later: Back up the user data, install the server software, and then restore it on the latest version of server software.

## Server software upgrade methods on Red Hat OpenShift

When you want to use the latest version of the server software on Red Hat OpenShift, you must upgrade Rational® Test Automation Server from the previous version of the software.

**Important:** On the Red Hat OpenShift platform, you can either use **in-place upgrade** or **upgrade (by new install)** methods depending on your requirements.

## Server software upgrade on Ubuntu

When you want to use the latest version of the server software on Ubuntu, you must upgrade Rational® Test Automation Server from the previous version of the software.

**Important:** On the Ubuntu platform, the **in-place upgrade** is not available. You must use the **upgrade (by new install)** method to install the latest version of the server software.

Before you upgrade the server software, you must ensure that you have completed the following tasks:

- Installed the following software:
  - Rational® Test Automation Server V10.1.0 or V10.1.1.
  - Helm V3.3.4 or later. For more information, refer to the Installing Helm section in the Helm documentation.
- Informed users that Rational® Test Automation Server is offline temporarily during the upgrade process.
- Completed all test executions that are running on the existing Rational® Test Automation Server V10.1.0 or V10.1.1.
- Stopped all stub executions that are running on the existing Rational® Test Automation Server V10.1.0 or V10.1.1.
- Cancelled any scheduled test runs that have a future date or time.

In upgrade (by new install) method, depending on the existing version of the software, you can upgrade the server software by using any of the following options. The following table describes the options that are available in the this method:

**Table 2. Upgrade options**

| Existing version | Upgrade methods |
|---|---|
| Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier | Table 3: Option 1 on page 49 |
| Rational® Test Automation Server V10.1.0 or V10.1.1 | Table 4: Option 2 on page 49 |

**Table 3. Option 1**

| Tasks | Task description | More information |
|---|---|---|
| 1 | Back up the user data, and then uninstall the server software. | Uninstalling the server software |
| 2 | Install the server software, and then restore the user data on the existing version of the server software. | Installing the server software on Ubuntu by using k3s on page 42 |

If your existing version of server software is 10.0.2, Fix Pack 1, or earlier, then you cannot directly upgrade to the latest version of the server software due to the steps involved in the migration process. Therefore, you must perform the tasks in sequence as listed in the following table:

**Table 4. Option 2**

| Tasks | Task description | More information |
|---|---|---|
| 1 | Back up the user data from V10.1.0 or V10.1.1. | Backing up the user data on page 52 |
| 2 | Uninstall the V10.1.0 or V10.1.1 of the server software. | Uninstalling the server software on Ubuntu on page 57 |
| 3 | Install the latest version of the server software, and then restore the user data on the existing version of the server software. | Installing the server software on Ubuntu by using k3s on page 42 |

# Backing up and restoring the user data

You might want to restore an earlier environment state if you install a new version of Rational® Test Automation Server, move environments to different systems (both real and virtual), or set up a test environment ready for testing. To address any of these needs, you can back up and restore Rational® Test Automation Server user data installed either on Ubuntu or on Red Hat OpenShift platform.

## Backing up and restoring the user data on Red Hat OpenShift

To secure the user data in IBM® Rational® Test Automation Server that is installed on the OpenShift server, you can back up the user data. At any point in time, after you install IBM® Rational® Test Automation Server, you can restore the backed-up user data.

**Before you begin**

You must have been granted access to the cluster.

**About this task**

This topic provides you with an example of backing up and restoring the user data by using Velero. Velero is one of the methods for backing up and restoring the user data. You can also use other methods for backing up and restoring the user data. If you use a different method to back up, then you must include the Persistent Volumes in the cluster.

## Preparing your cluster to backup and restore the user data

You must prepare your cluster before you back up or restore the user data in Rational® Test Automation Server.

1. Log in to the cluster by using oc login.
2. Install and configure Velero with the Restic Integration.
3. Set the name of the namespace in which Rational® Test Automation Server is installed by running the following command:

   ```
   NS=<namespace>
   ```

4. Update stateful sets to apply the annotations required by Velero to back up the PVs used by the pods by running the following commands:

   ```
   for sts in $(oc get sts -n "$NS" -o name); do \
     if ! oc -n "$NS" patch --type=json "$sts" -p \
       '[{"op":"add","path":"/spec/template/metadata/annotations/backup.velero.io~1backup-volumes",
    "value":"data"}]' 2>/dev/null; then
         oc -n "$NS" patch --type=json "$sts" -p \
          '[{"op":"add","path":"/spec/template/metadata/annotations",
    "value":{"backup.velero.io/backup-volumes":"data"}}]'
     fi \
   done
   ```

5. Change the security restriction of the pods by running the following commands:

   ```
   for sts in $(oc get sts -n "$NS" -oname); do \
     if oc get -n "$NS" "$sts" -ojsonpath='{.spec.template.spec.securityContext}' | grep -q
    "runAsNonRoot:true"; then \
       echo $sts
       oc patch -n "$NS" "$sts" --type json \
         -p '[{"op":"replace", "path":"/spec/template/spec/securityContext", "value":
    {"runAsNonRoot": false}}]'
     fi \
   done
   ```

## Backing up the user data

You can back up the user data by using Velero.

**Before you begin**

You must have completed the steps in the Preparing your cluster to backup and restore the user data on page 50.

**About this task**

This procedure is an example of backing up the user data by using Velero.

1. Log in to the cluster by using oc login.
2. Start the backup process by running the following commands:

```
velero backup create <backup-name> --include-namespaces=<ibm-rtas-namespace>
```

## Restoring the user data

If you want to restore the Velero backup on to a cluster in which Rational® Test Automation Server is not installed, you can restore the user databy using Velero.

**Before you begin**

You must have backed up the user data. See Backing up the user data on page 51.

1. Log in to the cluster by using oc login.
2. Start the restore process by entering the following command:.

```
velero restore create --from-backup=<backup-name> --restore-volumes
```

**Results**

You have backed up or restored the user data for Rational® Test Automation Server.

---

Related information

Persistent storage on page 28

Velero Documentation

## Backing up and restoring the user data on Ubuntu

To secure the user data in Rational® Test Automation Server that is installed on the Ubuntu server, you can back up the user data. At any point in time, after you install Rational® Test Automation Server, you can restore the backed-up user data.

**Before you begin**

You must have completed the following tasks:

- Been granted with the same permissions that are required for installing and uninstalling the software.
- Been granted with the *sudo* access.
- Communicated to the users that Rational® Test Automation Server might be unavailable for some time until the process is complete.

**About this task**

You may need to back up and restore the user data in the following scenarios:

- Move the existing environment to a new system.
- Change the release name or the namespace.
- Back up the user data at regular intervals and restore the backed-up data when required.

## Backing up the user data

You can back up the user data in Rational® Test Automation Server and use the backed-up data to restore at any point in time.

1. Log in to the Ubuntu server and open a terminal.
2. Run the following command to retrieve the case from IBM Cloud Pack repository:

   ```
   curl -o ibm-rtas-case-3.1012.0.tgz
    https://raw.githubusercontent.com/IBM/cloud-pak/master/repo/case/ibm-rtas-case-3.1012.0.tgz
   ```

3. Run the following command to extract the case:

   ```
   tar xvf ibm-rtas-case-3.1012.0.tgz
   ```

4. Change to the `ibm-rtas-case/inventory/ibmRtasBase/files` directory.

   This directory contains the `backup.sh` script which is required to complete the backup operation.

5. Create a directory that contains metadata related to the Persistent Volume Claims and their Persistent Volumes by running the following command:

   ```
   sudo ./backup.sh create-pvc-links
   ```

   You can use the following optional parameter along with the **create-pvc-links** command:

   --volumes or -v: Use this parameter to specify the directory path of the Volumes.

   For example, `backup.sh create-pvc-links -v ibm-rtas-case/inventory/ibmRtasBase/files/my-pvc-links` creates a directory called `my-pvc-links` that stores the metadata required for backup.

   **Remember:**

- If you did not provide the volume parameter, by default, the command creates a sub-directory in the same directory where you have the backup script.
- You must ensure that the directory you create must be empty or not exist to avoid script failure.
- If you specify a directory by using the -v parameter, then you must use the same parameter value when you restore the data.

6. Run the following command to stop the cluster and Rational® Test Automation Server:

```
k3s-killall.sh
```

7. Run the following command to create a backup of the existing user data:

```
sudo ./backup.sh create [options] <backup-file-name>
```

After you run this command, a backup of the local Persistent Volumes is created. The backup is created as tar archives that is compressed by using gzip (.tar.gz). The **create** command archives the Persistent Volumes into the *<backup-file-name>*.

You can use the following parameters along with the **create** command:

- --namespace or -n: Use this parameter to back up the Persistent Volumes in the specified namespace. If you do not mention the namespace, then all the Volumes from all the namespaces are included in the backup. The syntax is:

```
–namespace <name of the namespace>
```

- --volumes or -v: Use this parameter to specify the directory path of the Volumes. The syntax is:

```
–volumes <path-of-the-directory>
```

**Remember:**

- If you did not provide the volume parameter, by default, the command creates a sub-directory in the same directory where you have the backup script.
- If you specify a directory by using the -v parameter, then you must use the same parameter value when you restore the data.

For example,

```
sudo ./backup.sh create --namespace test-system my-backup.tar.gz
```

This command creates a backup file named *my-backup.tar.gz* that contains all of the Persistent Volumes associated with pods available in the *test-system* namespace.

8. Run the following command to restart the cluster and Rational® Test Automation Server:

```
sudo systemctl start k3s
```

## Restoring the user data

You can restore the user data that is backed up in Rational® Test Automation Server at any point in time after you install latest version of server software.

**About this task**

The server data from Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier, can be migrated to Rational® Test Automation Server during the installation of latest version of the server software. For more information, see the **Related information** section.

**Note:** While restoring backed up user data from V10.1.1 to V10.1.2, on the terminal you see information about Kafka messages. You can ignore those messages as they do not affect the restoration process.

1. Log in to the Ubuntu server and open a terminal.
2. Run the following command to retrieve the case from IBM Cloud Pack repository:

   ```
   curl -o ibm-rtas-case-3.1012.0.tgz
    https://raw.githubusercontent.com/IBM/cloud-pak/master/repo/case/ibm-rtas-case-3.1012.0.tgz
   ```

3. Run the following command to extract the case:

   ```
   tar xvf ibm-rtas-case-3.1012.0.tgz
   ```

4. Change to the `ibm-rtas-case/inventory/ibmRtasBase/files` directory.
   This directory contains the `backup.sh` script which is required to complete the restore operation.
5. If you have not already done so, create a directory that contains metadata related to the Persistent Volume Claims and their Persistent Volumes by running the following command:

   ```
   sudo ./backup.sh create-pvc-links
   ```

   You can use the following optional parameter along with the **create-pvc-links** command:

   --volumes or -v: Use this parameter to specify the directory path of the Volumes.

   For example, `backup.sh create-pvc-links -v ibm-rtas-case/inventory/ibmRtasBase/files/my-pvc-links` creates a directory called `my-pvc-links` that stores the metadata required to restore your user data.

   **Remember:**

   - If you did not provide the volume parameter, by default, the command creates a sub-directory in the same directory where you have the backup script.
   - You must ensure that the directory you create must be empty or not exist to avoid script failure.

6. Run the following command to stop the cluster and Rational® Test Automation Server:

   ```
   k3s-killall.sh
   ```

7. Run the following command to restore the backed-up user data:

```
sudo ./backup.sh restore [options] <backup-file-name>
```

The **restore** command overwrites the existing Persistent Volumes with data from the *<backup-file-name>*.

You can use the following parameters along with the **restore** command:

- --namespace or -n: Use this parameter to restore a specific namespace from the backup file. If you do not mention the namespace, then volumes from all the namespaces in the backup file are restored. You can map one namespace to another namespace by using colon (:). The syntax is:

```
--namespace <name of the namespace> [:<target-namespace>]
```

- --release or -r: Use this parameter if the helm release name of the server to which the backup is being restored is different than the helm release name of the server where the backup was taken. The syntax is:

```
--release <backup-release>:<target-release>
```

- --volumes or -v: Use this parameter to specify the directory path of the Volumes. The syntax is:

```
--volumes <path-of-the-directory>
```

- -k or --confirm: Use this parameter to skip the confirmation step.

For example,

```
sudo ./backup.sh restore --namespace test-system:new-test-system -r rel-1:rel-2 -k
 my-backup.tar.gz
```

This command restores the volumes that are backed up from the *rel-1* release in the *test-system* namespace to the *rel-2* release in the *new-test-system* namespace and skips the confirmation step.

8. Run the following command to restart the cluster and Rational® Test Automation Server:

```
sudo systemctl start k3s
```

**Note:**

- The cluster and Rational® Test Automation Server might take some time to restart and until that time, you cannot access the Rational® Test Automation Server URL.
- You can run the `kubectl get pods -A` command to verify that the Kubernetes environment is working. After a while, the status of the pods must be `Running` state.

### Results

You have backed up or restored the user data for Rational® Test Automation Server.

Related information

Installing the server software on Ubuntu by using k3s on page 42

# Uninstallation of the server software

When you no longer require Rational® Test Automation Server, you can uninstall the server software. You can uninstall the server software depending on the platform on which you installed the server software.

## Uninstalling the server software

When you want to install a new version of server software or to reinstall when an installation fails, you can uninstall the server software and its components from the Red Hat OpenShift platform.

**Before you begin**

You must have completed the following tasks:

- Installed Rational® Test Automation Server software.
- Closed Rational® Test Automation Server, any open web browsers, and all other applications that are enabled by Rational® Test Automation Server.

1. Open and log in to the terminal.
2. Run the following command to stop the workload that is running:

   ```
   oc delete all,cm,secret -lexecution-marker -n test-system
   ```

3. Run the following command to uninstall the server software:

   ```
   helm uninstall {my-rtas} -n test-system
   ```

   The PersistentVolumeClaims and PersistentVolumes that were created during the installation are not deleted automatically. If you reinstall the server software, the user data is reused unless you specifically delete those volumes.

   **Note:** You must substitute `{my-rtas}` with the same release name that you used during the installation of the server software.

4. **Optional:** Run the following command to delete everything including user data contained in PersistentVolumeClaims and PersistentVolumes:

   ```
   oc delete project test-system
   ```

**Results**

You have uninstalled the server software from the Red Hat OpenShift platform.

## Uninstalling the server software

When you want to install a new version of server software or to reinstall when an installation fails, you can uninstall the server software and its components from the Ubuntu platform.

**Before you begin**

You must have completed the following tasks:

- Installed Rational® Test Automation Server software.
- Closed Rational® Test Automation Server, any open web browsers, and all other applications that are enabled by Rational® Test Automation Server.

1. Log in to the Ubuntu server as a non-root user.
2. Run the following command to stop the workload that is running:

   ```
   kubectl delete all,cm,secret -lexecution-marker -n test-system
   ```

3. Run the following command to uninstall the server software:

   ```
   helm uninstall {my-rtas} -n test-system
   ```

   The PersistentVolumeClaims and PersistentVolumes that were created during the installation are not deleted automatically. If you reinstall the server software, the user data is reused unless you specifically delete those volumes.

   **Note:** You must substitute `{my-rtas}` with the same release name that you used during the installation of the server software.

4. **Optional:** Run the following commands to delete the entire Kubernetes environment, including all user data:

   ```
   #Run the following command if you are on Ubuntu 18.04
   cd ibm-rtas-case/inventory/ibmRtasBase/files
   sudo ./ubuntu-wipe.sh --confirm

   #Run the following command if you are on Ubuntu 20.04
   cd ibm-rtas-case/inventory/ibmRtasBase/files
   sudo HOME=$HOME ./ubuntu-wipe.sh --confirm
   ```

   After you run this command, the system returns to the same state as if `ubuntu-init.sh` was never run.

**Results**

You have uninstalled the server software from the Ubuntu platform.

# Configuration of the server software

You can find information about the tasks that you must perform after you installed the server software to configure Rational® Test Automation Server.

The following table describes the tasks that you can perform as an administrator after you installed the server software:

| Require-ment | Tasks | More information |
|---|---|---|
| Required | Review the default user management to know how the server software manages the users and their authentication. | User administration on page 58 |
| Required | Import Certificate Authority (CA) into a browser to prevent certificate errors when you access the Rational® Test Automation Server UI. | Importing Certificate Authority into a browser |
| Optional | Copy the third-party application Jars to Kubernetes to run API Suites that uses a transport. | Copying third-party application Jars to Kubernetes on page 74 |
| Optional | Change the password seed to provide enhanced security to Rational® Test Automation Server. | Changing the password seed on page 76 |

## User administration

After you install IBM® Rational® Test Automation Server, you must consider how the software manages users and authentication.

You can review the default user management provided by the server and decide what additional controls you might want to add. If you manage users and authentication through an existing LDAP/AD server, you can review how to use that server to manage Rational® Test Automation Server users.

### Default user administration

You installed IBM® Rational® Test Automation Server and you want to know about how the software manages user access.

Rational® Test Automation Server uses Keycloak V9.0.2 (https://www.keycloak.org/) to manage and authenticate users.

If you manage and authenticate users by using an LDAP and Active Directory server, you can configure Keycloak to connect to that server. For more information, see LDAP user administration on page 60.

Keycloak uses the concept of a realm to manage and authenticate users. When you install the server software, a realm called testserver is created for you in Keycloak. All server users belong to this realm and when they log in to the server, they log into that realm.

As an administrator, it is important to consider the following points about the server administration:

- By default, there is no administrator for Rational® Test Automation Server.

  Such an administrator is required for accessing additional functions, which includes claiming ownership of server projects and unarchiving them. But you can assign administrative privileges to any user. You must do this by adding the admin role to the user in Keycloak.

- You must sign up a user that you want to be the administrator. You must go to the Login page at `https://<fully-qualified-dns-name>:443` and sign up.

  **Note:** Do not use that admin user to perform non-administration tasks. Instead, sign up another user.

- After you sign up the user that you want to be the administrator for Rational® Test Automation Server, you must log in to the Keycloak Admin Console at `https://<fully-qualified-dns-name>:443/auth/admin/` to make that user the server administrator.

  The default user name for the Keycloak administrator is `keycloak`. The password is randomly generated when the software is installed. You can see the password by using the following kubectl command:

  ```
  kubectl get secret -n test-system {my-rtas}-keycloak-postgresql -o jsonpath="{.data.password}" |
   base64 --decode; echo
  ```

  **Note:** You must substitute `{my-rtas}` with the release name of your choice.

  After you log in to the Keycloak Admin Console, from the Users page, you can search and select the user that you want to make an administrator. From the **Groups** tab, you can join the user to the **Admins** group.

  For more information about assigning user roles, see Groups in the Keycloak documentation.

Now that you are the server administrator, it is important to consider the following points about the default user management and authentication:

- Minimum password length defaults to 8 characters
- Email verification of new users is turned off
- The Forgot Password feature is turned on by default but no instructions are sent to the user to reset their password
- Forgotten user passwords are changed by you if you do not enable Keycloak to send instructions to reset a password

You can review the following sections about changing the default authentication controls.

### Email settings

By default, the testserver realm sets the Forgot Password switch on. However, as an administrator, you must enable Keycloak to send an email to the user with instructions to reset their password. If you want to verify an email, you must also enable Keycloak to send an email to the user to verify their email address.

You must provide SMTP server settings for Keycloak to send an email. After you log in to the Keycloak Admin Console, see Email Settings in the Keycloak documentation.

Then, to set up the email verification, see Forgot Password in the Keycloak documentation.

### Password policy

By default, the testserver realm has a password policy where the minimum length of a password is 8. As an administrator, you can update password policies in Keycloak.

After you log in to the Keycloak Admin Console, see Password Policies in the Keycloak documentation.

### User password

If you did not enable Keycloak to send instructions to a user about how to reset a password, you must use the Keycloak Admin Console to change their password for them.

After you log in to the Keycloak Admin Console, see User Credentials in the Keycloak documentation.

### User deletion

When a user is inactive or no longer needs to access the server, you can delete that user.

After you log in to the Keycloak Admin Console, see Deleting Users in the Keycloak documentation.

## LDAP user administration

You installed IBM® Rational® Test Automation Server and you are using a Lightweight Directory Access Protocol and Active Directory HTTP server to manage users and authentication. You want to know the best practices to use that LDAP/AD server to manage user access to Rational® Test Automation Server.

Rational® Test Automation Server uses Keycloak V9.0.2 (https://www.keycloak.org/) to manage and authenticate users.

Existing user databases hold user credentials. Keycloak federates these existing external user databases through the concept of storage providers. By default, Keycloak supports an LDAP and Active Directory storage provider. By adding a storage provider, you can map LDAP user attributes into Keycloak. You can also configure more mappings.

Before you configure Keycloak to use an existing LDAP/AD provider, you must consider the following best practices:

- Set up your LDAP/AD provider as a read-only repository so that Rational® Test Automation Server cannot change it.
- Add and remove users in LDAP/AD and not the Keycloak local user database.
- Import and synchronize your LDAP/AD users to your Keycloak local database.
    - An import for an LDAP/AD user can fail if the LDAP/AD field chosen for the username mapping in Keycloak is not filled in for that user in LDAP/AD.
    - Filter LDAP/AD users by using the Custom User LDAP Filter, so you can import a subset of all your LDAP users. For example, you can set up a Server user group in LDAP and only import those users to Keycloak.
- Map a login style name, for example, `user1@server.com`, by using the `UserPrincipalName` attribute in LDAP/AD to a `username` in Keycloak. If you want the full name of the user as your login style, use the `cn` attribute in LDAP/AD.

**Note:** The LDAP/AD user name attribute must match the LDAP/AD provider user name attribute (**Username LDAP attribute**) in Keycloak for the LDAP/AD provider to connect with Keycloak.

The following sections use these best practices to guide you to set up Keycloak to connect to your LDAP/AD HTTP server.

### LDAP provider selection in Keycloak

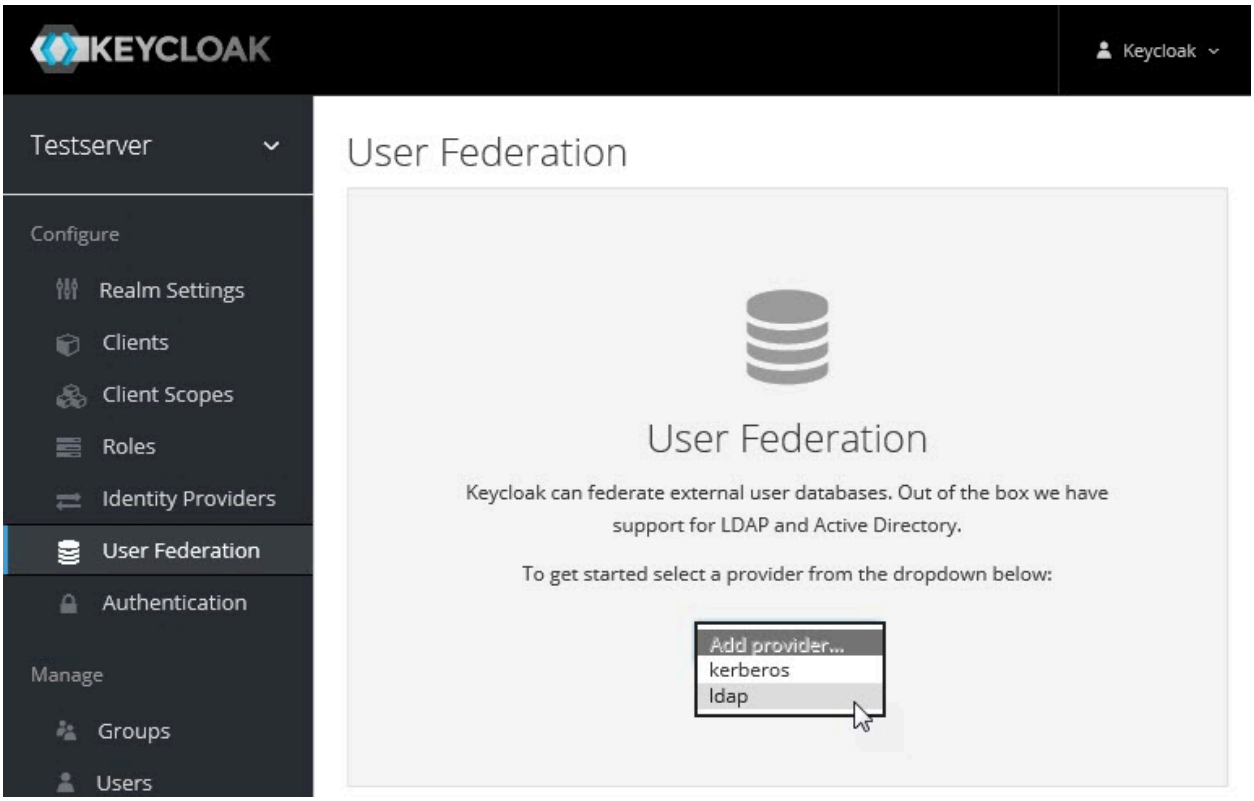
You can use the Keycloak Admin Console to add an LDAP/AD provider.

You can log in to the Keycloak Admin Console at `https://<fully-qualified-dns-nam>/auth/admin`. The default user name for the Keycloak administrator is `keycloak`. The password is randomly generated when you installed the IBM® Rational® Test Automation Server software. You can see the password by using the following kubectl command:

```
kubectl get secret -n test-system {my-rtas}-keycloak-postgresql -o jsonpath="{.data.password}" | base64
 --decode; echo
```

**Note:** You must substitute `{my-rtas}` with the release name of your choice.

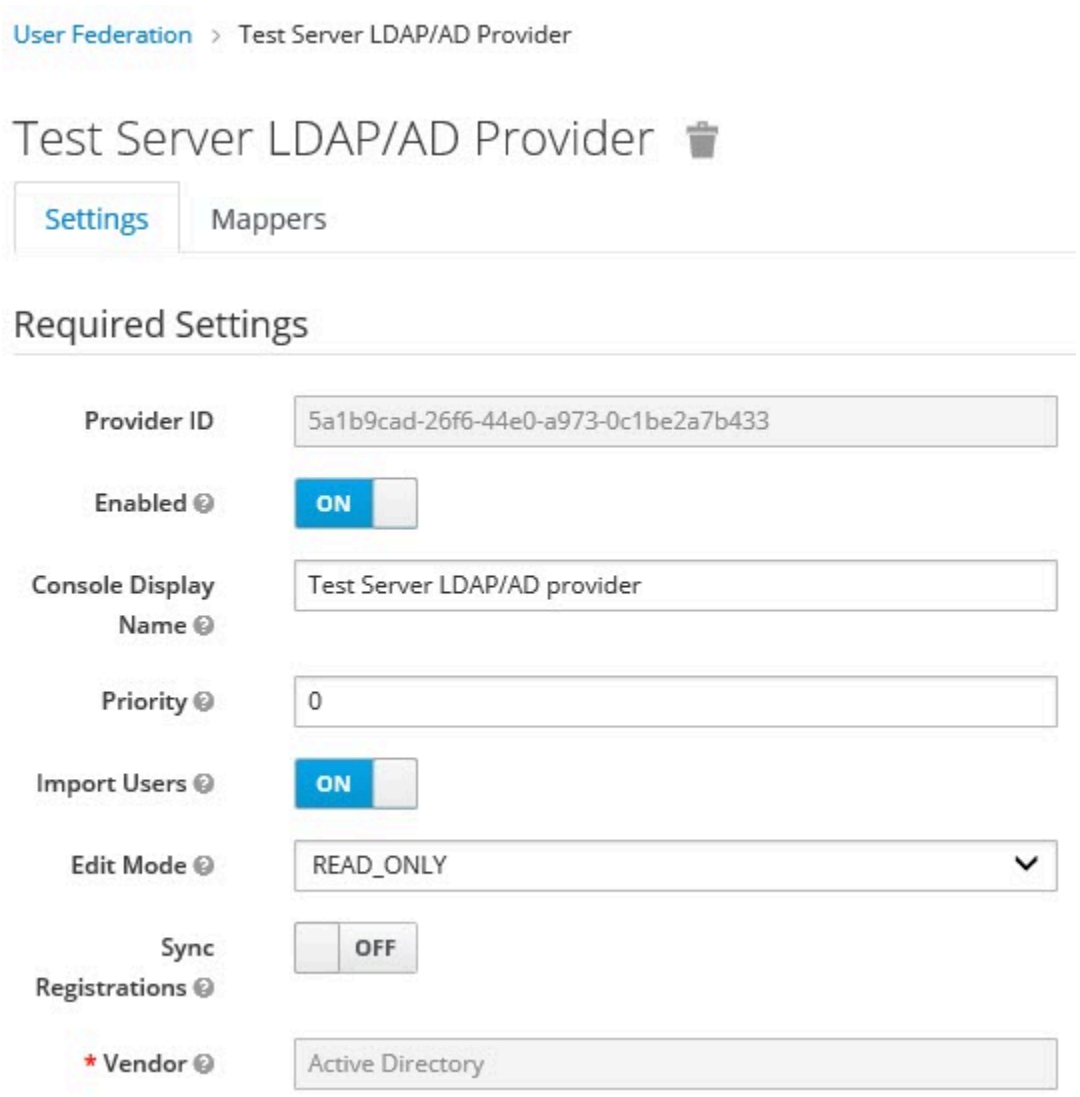
After you log in, you can go to the User Federation page to add your provider.




You want to select the provider named **ldap** from the list. From the Add User Federation Provider page, you can use a form to complete your LDAP/AD connection parameters.

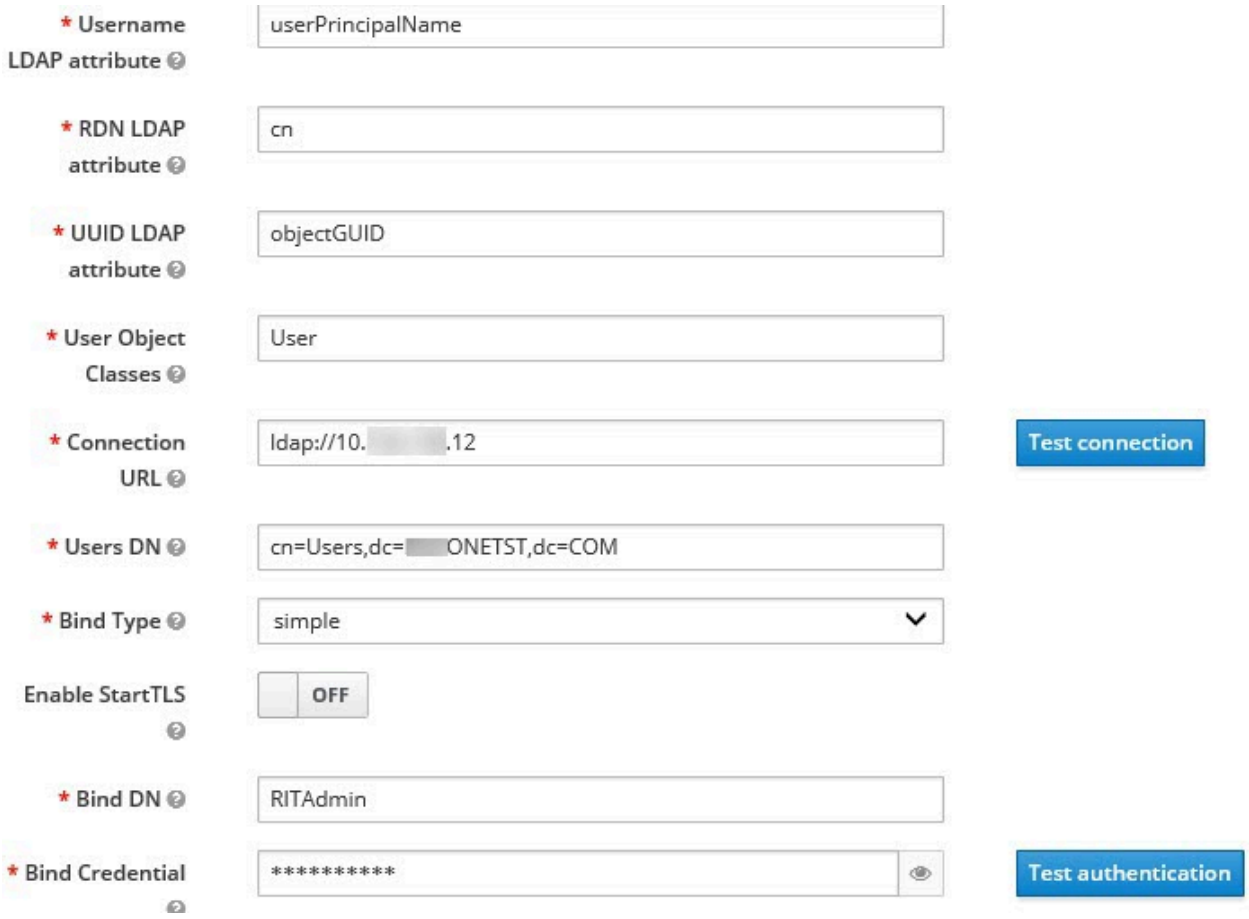
### Required settings for a successful connection to your LDAP/AD provider

The form includes many fields and several fields include default values. Tables describe the important fields that you must complete to ensure a successful HTTP connection to your LDAP/AD provider.



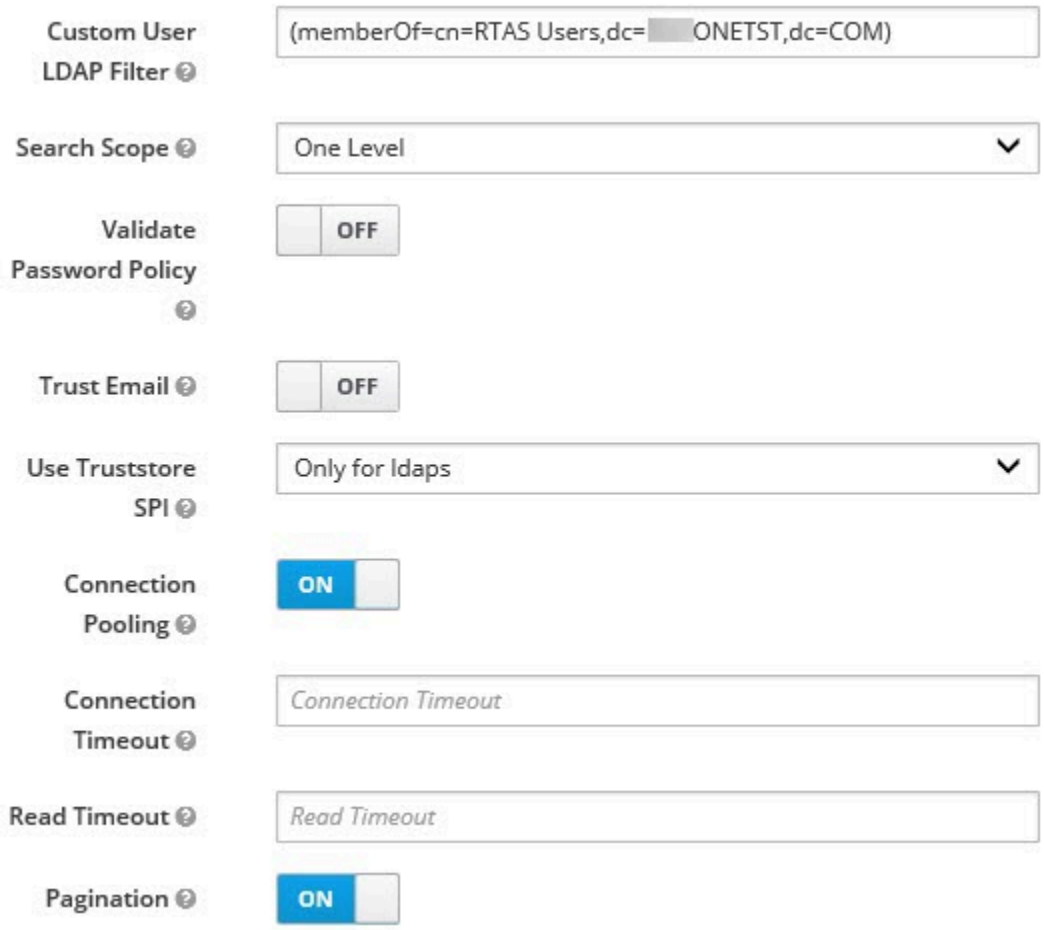

| Field | Description |
|---|---|
| Console Display Name | You want to make your console name recognizable, for example, Test Server LDAP/AD Provider. This name is shown in the Keycloak Admin Console. |
| Priority | The priority must be set to 0 so that when a user logs in to Rational® Test Automation Server, Keycloak looks up the user first in the list of users managed in the LDAP/AD user database. If the user is not found in your LDAP/AD user database, Keycloak then looks up the user in the local Keycloak user database. |
| Import Users | You want to leave import users **ON** so that users in your LDAP/AD user database are synchronized (imported) automatically into the Keycloak local user database. The import uses the settings that you are defining now.<br><br>After the initial import, when you add a user to your LDAP/AD user database and that user logs in to Rational® Test Automation Server, the LDAP/AD provider imports the |

| Field | Description |
|---|---|
| | LDAP/AD user into the Keycloak user database and authenticates the user against the LDAP/AD password.<br><br>Similarly, after the initial import, when you remove a user from your LDAP/AD user database that user cannot log in to Rational® Test Automation Server.<br><br>**Note:** For consistent user management, continue to manage users in your LDAP/AD user database. |
| Edit Mode | Make sure that edit mode is set to **READ_ONLY**.<br><br>This setting means that your LDAP/AD user database is read only. No user data defined through the mapping of attributes in Keycloak such as the `username` is written back from the Keycloak user database to your LDAP/AD user database.<br><br>This read-only setting also means that a new user cannot sign up from Rational® Test Automation Server as no user data can be written to your LDAP/AD user database. |
| Vendor | You want to select **Active Directory** from the list of vendors. Many fields complete with default values based on this selection. |

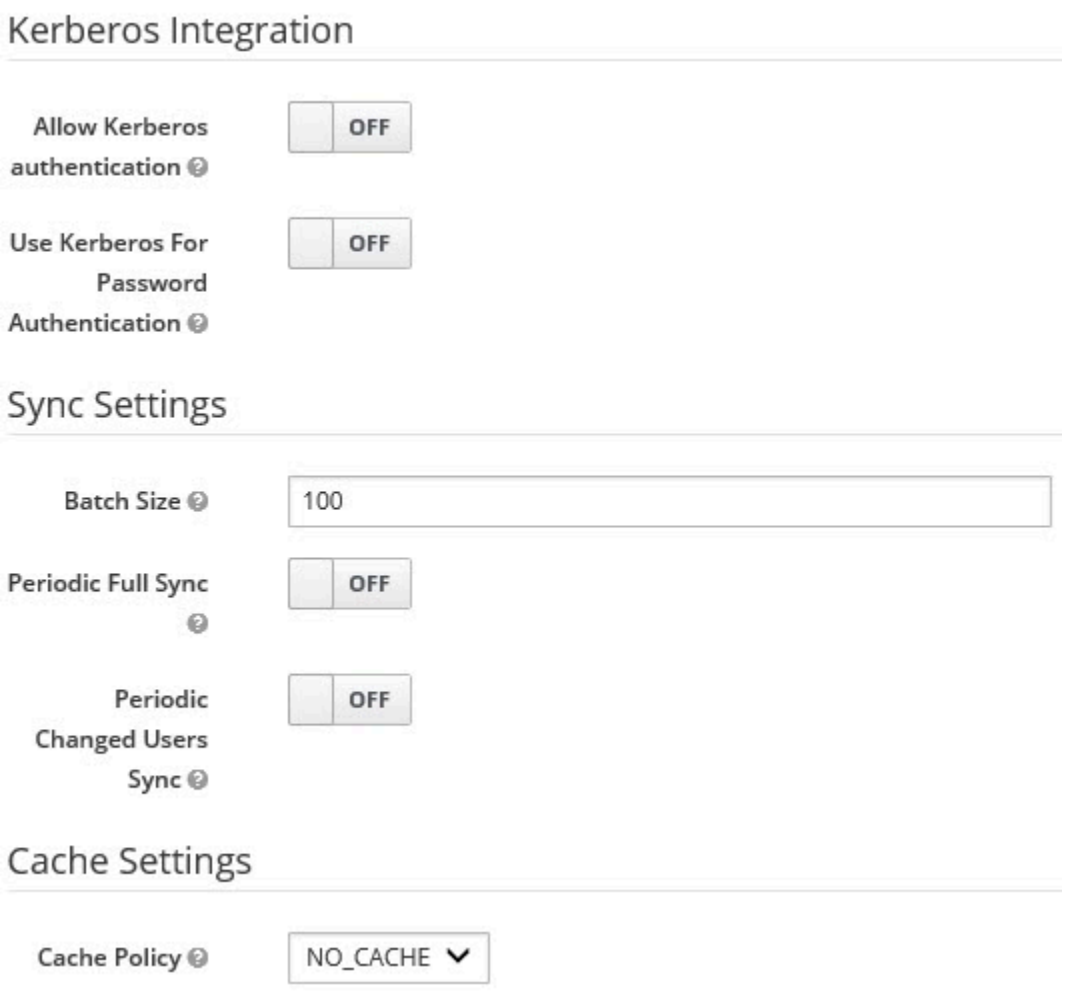



| Field | Description |
|---|---|
| Username LDAP attribute | You can use the default value `cn` for the `username`, which is first name, last name, or you can use `userPrincipalName`, which is username@domain.<br><br>You might want a login that matches your company style. For example, you might prefer joetester@mycompany.com instead of Joe Tester.<br><br>If you do make this change the `username` that you specify now must match the `username` in the Mapper. Keycloak makes this change for you. |
| User Object Classes | Because LDAP user records are found based on a user object class, you must set the User Object Classes to `User`. |
| Connection URL | So that Keycloak can connect to your LDAP/AD user database, you must enter your LDAP/AD URL, for example, `ldap://<hostname>.<domain>`<br><br>Make sure that you test the connection and confirm that the connection is successful. |
| Users DN | You must provide the directory where the LDAP users are listed, for example, `cn=User-s,dc=MYCOMPANY,dc=COM`. |

| Field | Description |
|---|---|
| Bind DN<br><br>Bind Credential | You must also provide the LDAP/AD user database administrator user ID for **BIND DN** and password for the **BIND Credential**. These credentials are used by Keycloak to access the LDAP/AD user database.<br><br>Make sure that you test the authentication and confirm that the authentication is successful. |




| Field | Description |
|---|---|
| Custom User LDAP Filter | You can filter your LDAP/AD users to import a subset of all your LDAP users.<br><br>For example, you can set up a testers user group for your LDAP/AD user database such that only those users are imported to Keycloak.<br>`(memberOf=cn=TESTERS,dc=MYCOMPANY,dc=COM)` |

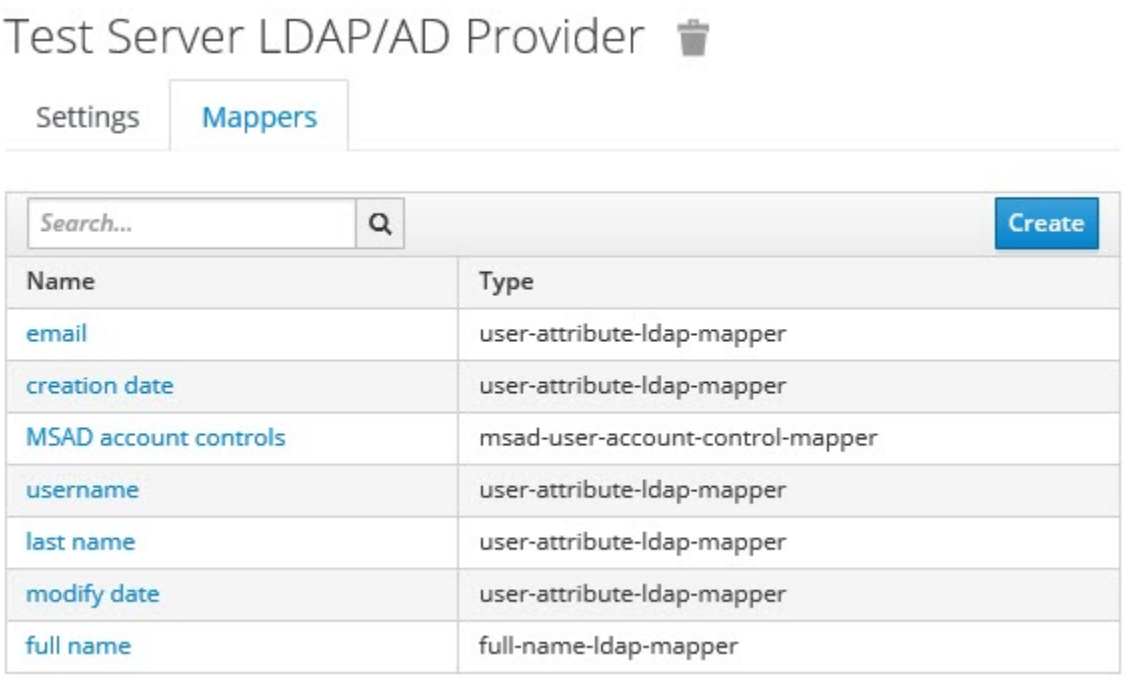



| Field | Description |
|---|---|
| Batch size | You can change the number of users to import in a single transaction by using the batch size. |
| Cache Policy | If you have many users or performance concerns, you can change the cache policy to no cache. |

You can save the settings, which creates your LDAP/AD provider.

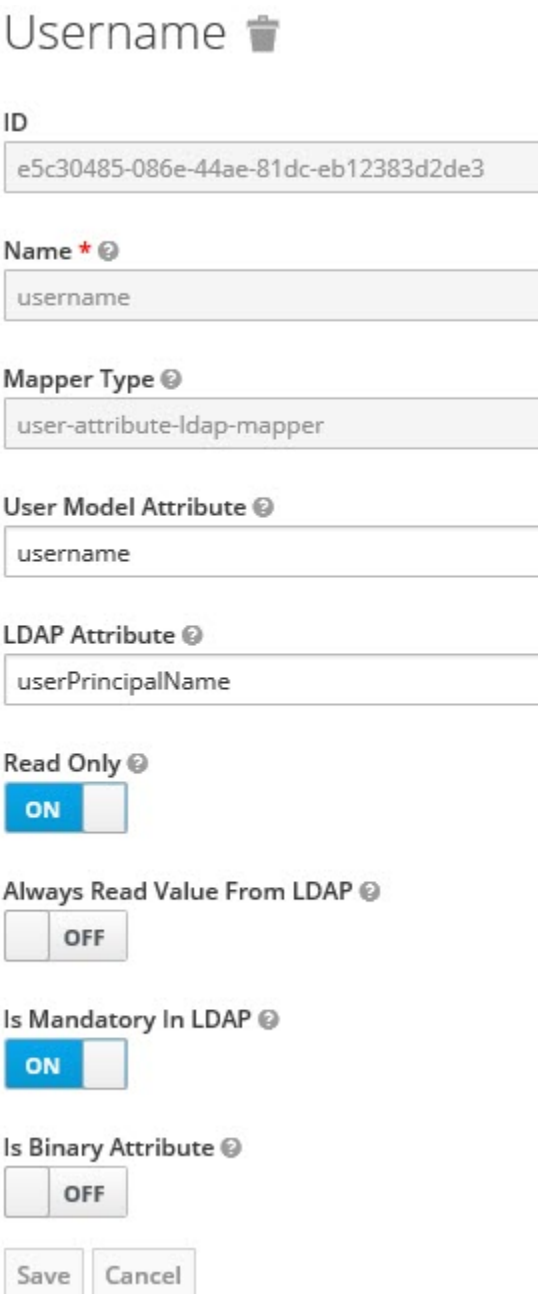
### Mappers

Keycloak uses mappers to map the user attributes defined in the Keycloak user model such as **username** and **email** to the corresponding user attributes in the LDAP/AD user database. By default, when you saved your settings and created your LDAP/AD provider, the following mappers were created.



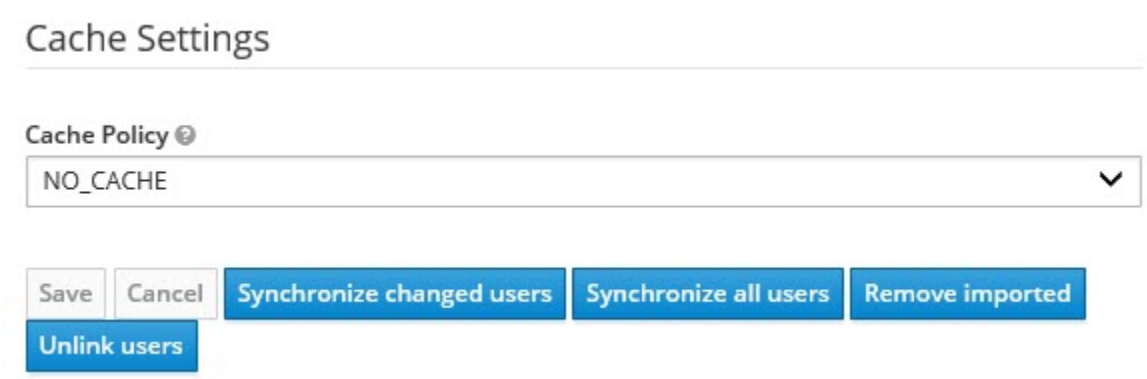

The username attribute that you specified in the **Username LDAP attribute** must match the **username** t attribute defined in the Keycloak mapper for the LDAP/AD user database to connect with Keycloak.

Because you changed the **Username LDAP attribute** from the default value `cn` to `userPrincipalName`, Keycloak made the same change in the mapper called **username** to match.



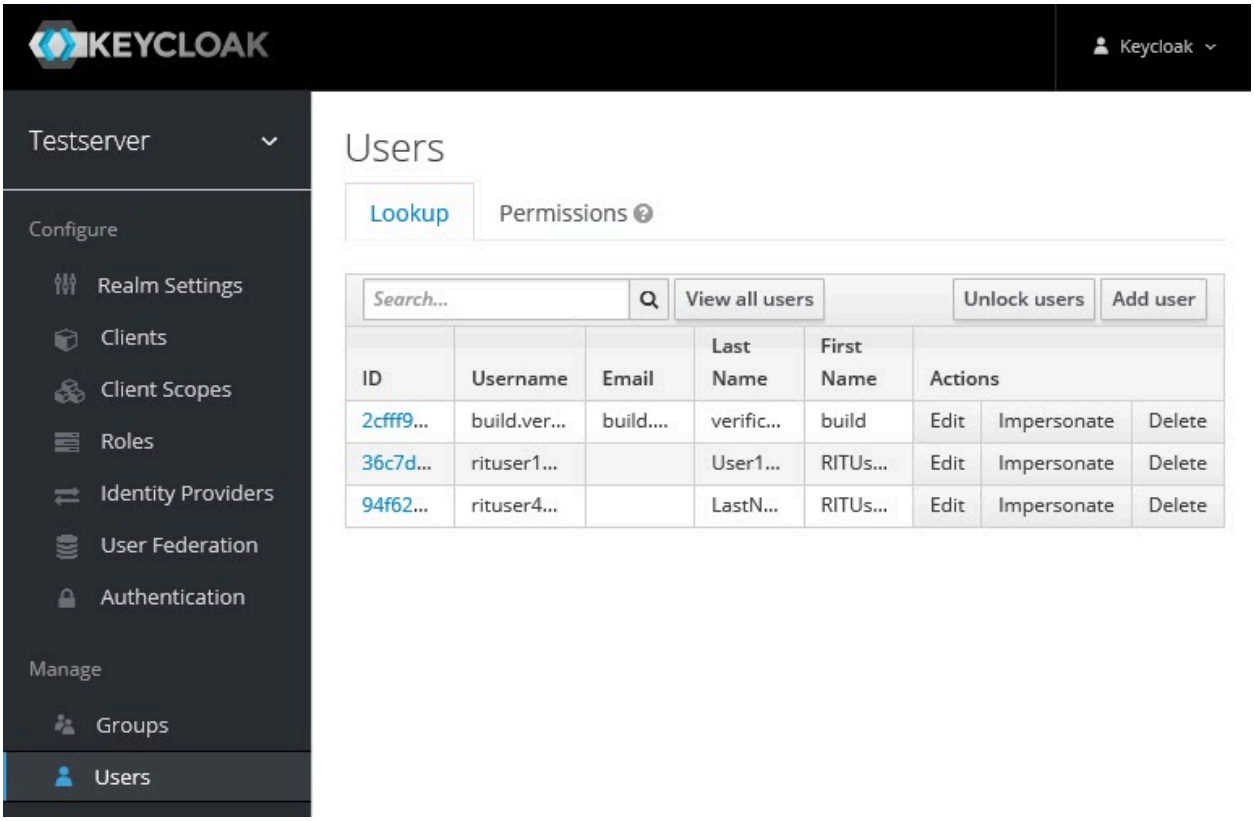

### User synchronization

You must import all users from your LDAP/AD user database by using the option to **Synchronize all users**. Users are imported based on your saved settings when you set up your LDAP/AD provider.




A successful import is followed by a success message with the number of users imported. A failed import typically results when there is a mismatch between user attributes in the Keycloak user database and the LDAP/AD user database.

You can view all the LDAP/AD database users that were imported and authenticated from the Users page in the Keycloak Admin Console.




Users are listed with ID, Username, Email, Last Name, and First Name. The ID is generated by Keycloak. The value of the other attributes is fetched from the LDAP/AD user database by using mappers.

## Managing account settings

You might want to change your password or other settings for your account.

**About this task**

As an account owner, you can view your account settings, which include your user name, email, and first and last name. You can also change all of these settings except the user name. Changing your password is also possible from your account settings.

**Note:** Account settings cannot be changed if IBM® Rational® Test Automation Server was configured to use an LDAP/AD provider. For more information about server security, see the related links.

- Edit your account details by following these steps:
  1. Click the **User** icon from the menu bar, and select **Account Settings**.

     The Edit account page is displayed.
  2. Edit your user details and save the changes. You can also reset your changes. If you cancel your changes, you return to the Home page.
- Change your password by following these steps:
  1. Click the **User** icon from the menu bar, and select **Account Settings**.

     The Edit account page is displayed.
  2. Click **Change password**.

     The Change password page is displayed.
  3. Type your current password followed by your new password
  4. Confirm the new password by typing it again and save the changes. You can also reset your changes. If you cancel your changes, you return to the Home page.

Related information

Default user administration on page 58

## Certificate authority: Importing and extending lists

When you want browsers and other applications to access Rational® Test Automation Server, you must enable browsers to trust the certificate authority (CA) of Rational® Test Automation Server. For some applications that integrate with Rational® Test Automation Server, you must also extend the CA lists that are trusted by the applications.

You can find the following tasks to import the certificate authority in the browser that you want to use:


- Importing Certificate Authority into the Google Chrome browser on page 72
- Importing Certificate Authority into the Microsoft Edge browser on page 72
- Importing Certificate Authority into the Mozilla Firefox browser on page 73


## Importing Certificate Authority into the Google Chrome browser

You must import certificate authority into the Google Chrome browser to prevent certificate errors when accessing the Rational® Test Automation Server UI.

**Before you begin**

You must have completed the following tasks:

- Saved a certificate authority (CA) during the installation of Rational® Test Automation Server.

  You can run the following command to get the certificate from the system:

  ```
  kubectl get secret ingress -n test-system -o jsonpath={.data.ca\\.crt} | base64 -d
  ```

  **Remember:** *test-system* is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the *test-system*.

- Installed the browser that you want to use to access Rational® Test Automation Server.

1. Open a Google Chrome browser.
2. Navigate to **Customize and control Google Chrome > Settings**.
3. Click **Security** from the **Privacy and security** pane.
4. Click **Manage certificates**, and then **Import**.
5. Click **Next** in the **Certificate Import Wizard** window.
6. Click **Browse** and select the CA that you want to import, and then click **Next**.
7. Select the **Place all certificates in the following store** option to store the CA securely.
8. Click **Browse**, and then select **Trust Root Certification Authorities** as certificate store.
9. Click **OK**, and then **Finish**.

**Results**

You have imported the CA into the Google Chrome browser.

## Importing Certificate Authority into the Microsoft Edge browser

You must import Certificate Authority (CA) into the Microsoft Edge browser to prevent certificate errors when accessing the Rational® Test Automation Server UI.

**Before you begin**

You must have completed the following tasks:

- Saved a certificate authority (CA) during the installation of Rational® Test Automation Server.

  You can run the following command to get the certificate from the system:

  ```
  kubectl get secret ingress -n test-system -o jsonpath={.data.ca\\.crt} | base64 -d
  ```

  **Remember:** *test-system* is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the *test-system*.

- Installed the browser that you want to use to access Rational® Test Automation Server.

1. Type `certmgr.msc` in the **Start** menu **Search** filed, and then press **Enter**.
2. Expand **Trusted Root Certification Authorities**, and then select **Certificate**.
3. Right-click on the empty space, and then select **All tasks > Import**.
4. Click **Next** in the **Certificate Import Wizard** window.
5. Click **Browse** and select the CA that you want to import, and then click **Next**.
6. Select the **Place all certificates in the following store** option to store the CA securely.
7. Click **Browse**, and then select **Trust Root Certification Authorities** as certificate store.
8. Click **Next**, and then **Finish** to import the certificate.

**Results**

You have imported the CA into the Microsoft Edge browser.

## Importing Certificate Authority into the Mozilla Firefox browser

You must import Certificate Authority (CA) into the Mozilla Firefox browser to prevent certificate errors when accessing the Rational® Test Automation Server UI.

**Before you begin**

You must have completed the following tasks:

- Saved a certificate authority (CA) during the installation of Rational® Test Automation Server.

  You can run the following command to get the certificate from the system:

  ```
  kubectl get secret ingress -n test-system -o jsonpath={.data.ca\\.crt} | base64 -d
  ```

  **Remember:** *test-system* is the name of the namespace. If you created a namespace by using a different value, then you must use that value in place of the *test-system*.

- Installed the browser that you want to use to access Rational® Test Automation Server.

1. Open a Mozilla Firefox browser.
2. Enter `“about:preferences#privacy”` in the address bar, and then press **Enter**.
3. Locate and click the **View Certificates** option.

4. Select **Authorities** tab, and then click **Import**.
5. Select the CA that you want to import, and then click **Open**.
6. Select **Trust this CA to identify websites** in the **Downloading Certificate** window.
7. Click **OK** to import the CA into the browser, and then close the window.
8. Restart Firefox.

**Results**

You have imported the CA into the Mozilla Firefox browser.

## Copying third-party application Jars to Kubernetes

You can run API Suites in a project on Rational® Test Automation Server. If the API Suite uses a transport and the transport requires third-party application `Jar` files for a successful run, you must ensure that the third-party application `Jar` files are available at the test run time. To achieve this, you must copy the third-party application `Jar` files to the computer where Rational® Test Automation Server is installed on Kubernetes.

**Before you begin**

If you want to copy the third-party application `Jar` files to the remote Docker host, see Copying third-party application Jars to a remote Docker host on page 113.

You must have server administrator or cluster administrator privileges.

You must have completed the following tasks:

- Identified the third-party application `Jar` files that are required and copied the files. See Test run considerations for API Suites on page 99.
- Copied the third-party application `Jar` files from Rational® Integration Tester to the directory or folder in Kubernetes from where you can run the kubectl commands.

**About this task**

You can copy the third-party application `Jar` files to the folder that is specific for the application under the `/data/<application_name>` folder. You need not extract the files.

You can perform this task any time after you have installed Rational® Test Automation Server and you plan to run an API Suite that meets any of the following conditions:

- The API Suite uses transports and the transports require third-party application `Jar` files to be available at the test run time.
- The API Suite in the project has a results database configured.

You can get help on the kubectl commands by running the command: kubectl cp --help from the `/kube` directory. For more information, refer to the kubectl documentation.

1. Use the following table to find the name of the folder that corresponds to the specific third-party application for the transport used in the API Suite.

   **Table 5. Name of the folder for the application**

| Application | Name of the folder to use |
|---|---|
| Camel | Camel |
| CentraSite | CentraSite |
| CICS | CICS |
| Coherence | Coherence |
| Database | JDBC |
| IMS | IMS |
| Integra | Integra |
| JMS | JMS |
| SAP RFC | SAP |
| Software AG Universal Messaging | SoftwareAGUM |
| TIBCO EMS | TIBCO |
| TIBCO Rendezvous | |
| TIBCO SmartSockets | |
| WebSphere Application Server Service Integration Bus (SiBus) | WAS |
| WebLogic | WebLogicJMX |
| Software AG webMethods | webMethods |
| WebSphere MQ | WMQ |

2. Open the command prompt from the `/kube` directory and copy the `Jar` files to the folder that corresponds to the application by using the following command:

   ```
   kubectl cp <compressed_files> test-system/{my-rtas}-userlibs-0:/data/<application_name>/
   ```

   For example, if you are copying the Database `Jar` files, then the name of the folder is *JDBC* and the command to use is:

   ```
   kubectl cp mysql-connector-java.jar test-system/{my-rtas}-userlibs-0:/data/JDBC/
   ```

**Note:** You must substitute `{my-rtas}` with the release name of your choice.

3. Verify whether the `Jar` files are copied by running the following command:

```
kubectl exec {my-rtas}-userlibs-0 -n test-system -- ls /data/<folder_created>
```

For example, if you copied the Database `Jar` files, `mysql-connector-java.jar` to the folder *JDBC*, then the command to verify is:

```
kubectl exec {my-rtas}-userlibs-0 -n test-system -- ls /data/JDBC
```

The following information is displayed for the `JDBC` folder:

```
mysql-connector-java.jar
```

**Note:** You must substitute `{my-rtas}` with the release name of your choice.

**Results**

You have successfully copied the third-party application `Jar` files to the folder in Kubernetes.

**What to do next**

You can configure the API Suite run. See Configuring an API Suite run on page 130.

## Changing the password seed

As an administrator, you can change the password seed that is used when you install the server software to provide enhanced security to Rational® Test Automation Server.

**Before you begin**

You must have completed the following tasks:

- Installed Rational® Test Automation Server. See Installation of the server software on page 25.
- Installed the JSON command-line tool, jq, and ensured that the jq is in your environment PATH. For information about jq, refer to jq documentation.
- Installed the Curl command line tool. For more information refer to curl documentation.

**About this task**

When you install Rational® Test Automation Server, you supply a password seed when you run the `helm install` command. This password seed is used to generate several Kubernetes secrets. Kubernetes Secrets can contain the following information:

- The authentication credentials for micro-services.
- An encryption key for the user-created secrets collection or other secrets.

When you change the password seed for Rational® Test Automation Server, you must consider the following scenarios:

- Rational® Test Automation Server cannot communicate until you reconcile the passwords which are in Kubernetes Secrets by using the old and a new password seed.
- Users cannot read secret collections or other secrets that they have created in Rational® Test Automation Server until you re-encrypt them using a new password seed.

**Important:** You must provide an offline token and old password seed that you used during the installation of server software to re-encrypt user secrets.

1. Run the following command to change the password seed for Rational® Test Automation Server:

**Notes:** You must substitute the value of the following variables with the actual value in the command:

   - `{my-rtas}` with the release name that you used during the installation of the server software.
   - `{my-new-super-secret}` with a new value of your choice as the password seed.
   - You must run the following `helm upgrade` command from the same directory where the `helm install` command was run during the installation of the server software. Because the upgrade is dependent on the helm charts and .yaml file values used during the run time of the `helm install` command.

```
helm upgrade {my-rtas} ./ibm-rtas-prod -n test-system \
--reuse-values \
--set global.ibmRtasPasswordAutoGenSeed={my-new-super-secret}
```

2. Run the following script to generate new server secrets from the updated password seed and to save them to the persistent storage:

```
./ibm-rtas-prod/files/reconcile-secrets.sh -n test-system {my-rtas}
```

3. Run the following command to restart all the pods:

```
kubectl delete pods -n test-system \
  -lapp.kubernetes.io/instance={my-rtas} \
  -lapp.kubernetes.io/managed-by=Helm
```

4. Run the following commands to re-encrypt the user-created secrets collection or other secrets by providing the old password seed:

**Note:** You must substitute the value of the following variables with the actual value in the following commands:

- `{$SERVER_URL}` with the URL of your Rational® Test Automation Server UI.
- `{$OFFLINE_TOKEN}` with the offline token that belongs to a user with the administrator role.
- `{$OLD_SEED}` with the previous password seed that you used during the installation of the server software.

```
export ACCESS_TOKEN=$(curl -k -X POST {$SERVER_URL}/rest/tokens/ \
-H "Content-Type: application/x-www-form-urlencoded" \
-H "accept: application/json" \
-d "refresh_token={$OFFLINE_TOKEN}" | jq -r '.access_token')

curl -k -X POST {$SERVER_URL}/rest/secrets/re-encrypt/ \
     -H "Authorization: Bearer $ACCESS_TOKEN" \
     -H "Content-Type: application/json" \
     -d "{\"type\":\"helm\",\"password_auto_gen_seed\":\"{$OLD_SEED}\"}"
```

5. Run the following command to display the log file of the gateway pod:

**Note:** You must substitute the value of the following variables with the actual value in the command:

- `{my-rtas}` with the release name that you used during the installation of the server software.
- `abcdefghij-abcde` with an identifier that is assigned to the name of the gateway pod.

  You can run the `kubectl get pods -n test-system` command to obtain the identifier that is assigned to the gateway pod.

```
kubectl logs {my-rtas}-gateway-abcdefghij-abcde -n test-system
```

**Result**

The following message is displayed when re-encryption is completed:

```
reEncrypt complete. StringyReEncryptor [total=100, fixed=100, broken=0, noop=0]
```

**Results**

You have successfully changed the password seed for Rational® Test Automation Server.

# License management

You can manage and administer licensing for your installed server software through the **Configure Licenses** window in Rational® Test Automation Server.

## Floating licenses

You can purchase floating licenses, and then install the licenses on Rational License Key Server to access Rational® Test Automation Server.

When you log in to Rational® Test Automation Server as a new user, you must consider the following conditions:

- The **No licenses configured** dialog box is displayed.
- The **New Project** button is disabled.
- The **No licenses configured** dialog box is displayed if you click the **User** icon , and then **Configure Licenses**.

Users can create projects in Rational® Test Automation Server after the administrator configures the licenses. If the users access the **Configure Licenses** option, they can view only the following information in the **Available licenses** dialog box:

- The number of licenses that are currently in use.
- The total number of floating licenses that are configured by the administrator.
- The value of Rational License Key Server that is set during the installation of Rational® Test Automation Server.
- The name and email ID of the users that are configured as prioritized floating users by the administrator.

The addition of the **Prioritized Floating Users** option in the **Configure License** window enables the administrator to add users to the prioritized users list. The prioritized floating users do not return their floating licenses to the license pool when they log out from Rational® Test Automation Server. Thereby, Rational License Key Server ensures that the licenses are always available for the prioritized users. This feature is useful for those users who generate offline tokens that are used for long-run tests even though when the members are logged out of Rational® Test Automation Server.

A single floating license is used when the user performs any of the following tasks for the first time on Rational® Test Automation Server:

- Create a project as your first action after you log in.
- Log in subsequently and open a project that you created as your first action.
- Open a project of which you are already a member as your first action after you log in.
- Run tests on any of the third-party applications that are integrated with Rational® Test Automation Server.

The floating license that was assigned to you is returned to the license pool when you perform any of the following actions:

- Log out from Rational® Test Automation Server.

  **Remember:** If the users are added as prioritized users, they do not return license though they have logged out of Rational® Test Automation Server.

- End session on Rational® Test Automation Server or session is timed out.
- Complete the test runs on any of the third-party applications that are integrated with Rational® Test Automation Server.

**Important:**

Processor Value Unit (PVU) licenses are not supported for running Rational® Performance Tester schedules in Rational® Test Automation Server. You must configure floating licenses to run Rational® Performance Tester schedules in Rational® Test Automation Server.

## Configuring floating licenses

As an administrator, you must configure the number of purchased floating licenses in Rational® Test Automation Server so that users can create a project and perform other tasks in the project.

**Before you begin**

- You must have installed the product and signed up as a user on Rational® Test Automation Server.
- You must be an administrator of Rational® Test Automation Server.

**About this task**

As an administrator, you can assign the role of an administrator to any user. For information about how to change the role, see the related link.

**Note:** When your role from a user is changed to an administrator, you must log out and log in again so that Rational® Test Automation Server can apply the change in your role.

1. Log in to Rational® Test Automation Server.

   The **Configure licensing** dialog box is displayed for the first time you log in or if the licenses are not configured.

   Alternatively, you can access the **Configure Licenses** option by clicking the **User** icon .
2. Enter the number of floating user licenses in the **No. of floating users (Max)** field.

   Alternatively, you can use the up-down control buttons to change the number of licenses in the field.

**Remember:** You must ensure that the number of floating licenses that you configure does not exceed the number of licenses you have purchased.

3. Enter the value in the **Rational License Key Server** field, if you want to provide the value that is different from the value configured during the installation of Rational® Test Automation Server.
4. Click **Configure** to verify that you have entered the permissible number of licenses that you want to use on this server.
   The **Licensing compliance** dialog box is displayed.

   **Important:** You can allocate the purchased licenses across multiple servers only if the total number of floating users that you configure across the servers does not exceed the total number of purchased licenses.

5. Click **Confirm** to apply the license configuration.
6. Enter the name or the email ID of the users in the **Add prioritized floating users** field, and then select the name from the list that is displayed to add users to the **Prioritized Floating Users** list.

   **Note:** You can press the Ctrl key to select multiple users simultaneously.

7. Click **Add** to make selected users as the prioritized floating users.
8. Perform the following actions described in the following table to remove or replace users from the **Prioritized Floating Users** list:

| Options | Actions |
|---|---|
| **Remove** | Perform the following steps:<br>a. Locate the user from the **Prioritized Floating Users** list that you want to remove.<br>b. Click the **Menu** icon next to the name of the user, and then click **Remove**.<br>c. Click **Remove** on the **Remove prioritized user** dialog box to complete the action.<br>**Remember:** When you remove the user from the **Prioritized Floating Users** list, the test runs scheduled by the user fails to run if sufficient floating licenses are unavailable. If the user is currently on Rational® Test Automation Server, then the license can be used until the current session remains active. |
| **Replace** | Perform the following steps: |

| Options | Actions |
| --- | --- |
| | a. Locate the user from the **Prioritized Floating Users** list that you want to replace with another user.<br>b. Click the **Menu** icon next to the name of the user, and then click **Replace**.<br>c. Enter the email ID or name of another user that you want to add to the **Prioritized Floating Users** list.<br>d. Click **Replace** to complete the action. |

**Results**

The licenses are configured and the message `Licensing is successfully configured` is displayed. The **License usage** section displays the number of floating licenses and prioritized floating user licenses that are currently in use.

---

Related information

Default user administration on page 58

# Chapter 5. Test Author Guide

This guide describes how to create test assets in Rational® Test Automation Server and publish test assets to the Git repository that you have configured. This guide is intended for testers and test managers.

## Datasets overview

A dataset provides tests with variable data during a run. The test that uses a dataset at run time replaces a value in the recorded test with variable test data that is stored in the dataset.

In Rational® Test Automation Server, you can create a dataset and use it to replace the dataset values with original values during run time, when you want to run test assets that contain the dataset.

From the **Datasets** page, you can perform the following actions:

**Table 6. Options available in Datasets page**

| **Actions** | **Descriptions** |
|---|---|
| View | You can view the contents of a dataset. See Viewing a dataset on page 94. |
| Create | You can create a dataset to use it during a test or schedule run. See Creating a dataset on page 84. |
| Edit | You can edit a dataset when you want to run a test asset with different dataset values. See Editing a dataset on page 85. |
| Configure | You can configure the string values in a dataset that contains variable data for tests to use when they run. See Editing a dataset on page 85. |
| Delete | You can delete the dataset when it is not required in your test environment. See Deleting a dataset on page 96. |
| Save As | You can save a copy of a dataset in a different folder or make a copy with a different name by clicking the **Menu** icon and then by selecting the **Save As** option. |
| Branch selection | You can view the dataset listed in the other branches of the repository by selecting the name of the branch from the **Branch** list. When you access the **Datasets** page for the first time after adding the repository, the default branch that it displayed is **master**.<br><br>When multiple repositories are added to the same project, the following events occur:<br><br>• The datasets stored in the **master** branch of all the repositories are displayed.<br><br>• All the branches in all the repositories are listed in the drop-down list. |

**Table 6. Options available in Datasets page (continued)**

| Actions | Descriptions |
|---|---|
| | To differentiate a common branch across multiple repositories added to the project, the tooltip is not displayed in front of the common branch name in the list. |

**Notes:**

- Only a project **Owner** or a project member whose role is a **Tester** can view and edit the dataset.
- To view the content of an encrypted dataset, you must provide an encryption key that you set while encrypting the dataset column.
- As a **Viewer**, when you try to access the **Datasets** page, an error message is displayed because you do not have the permission to view this page.

## Creating a dataset

You can create datasets in Rational® Test Automation Server to replace the existing dataset values with new dataset values during a test or schedule run.

**Before you begin**

You must be a member of the project with the **Owner** or **Tester** role and must have completed the following tasks:

- Created a project in Rational® Test Automation Server. See Adding details to a server project on page 282.
- Configured the Git repository in your project. See Adding repositories to a server project on page 282.

**About this task**

When you create a dataset in Rational® Test Automation Server, it always creates 1 Row, 2 Column (1R X 2C) dataset. Later, you can edit the dataset by adding some rows and columns that you want, to add the data in it.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the Rational® Test Automation Server UI.
4. Go to the **Datasets** page, and then click **Create dataset**.
5. Enter a name for the dataset in the **Asset name** field, and then select the place where you want to save the dataset in the **Location** drop-down list.
6. Click **Create**.

   **Result**

   The new dataset opens in a CSV Editor in a web browser. The dataset created is listed on the **Datasets** page.

**Results**

You have created a dataset in your project.

**What to do next**

- You can add, modify, or remove data in the dataset. See Editing a dataset on page 85.
- You can publish the dataset to the Git repository so that other members of the project can use the dataset. See Publishing a dataset on page 93.

## Editing a dataset

You can add, modify, remove, import, or export data from a dataset by using the CSV Editor. The working principle of the CSV Editor is similar to that of a spreadsheet.

**Before you begin**

You must have created a dataset or configured a repository that contains the dataset.

**About this task**

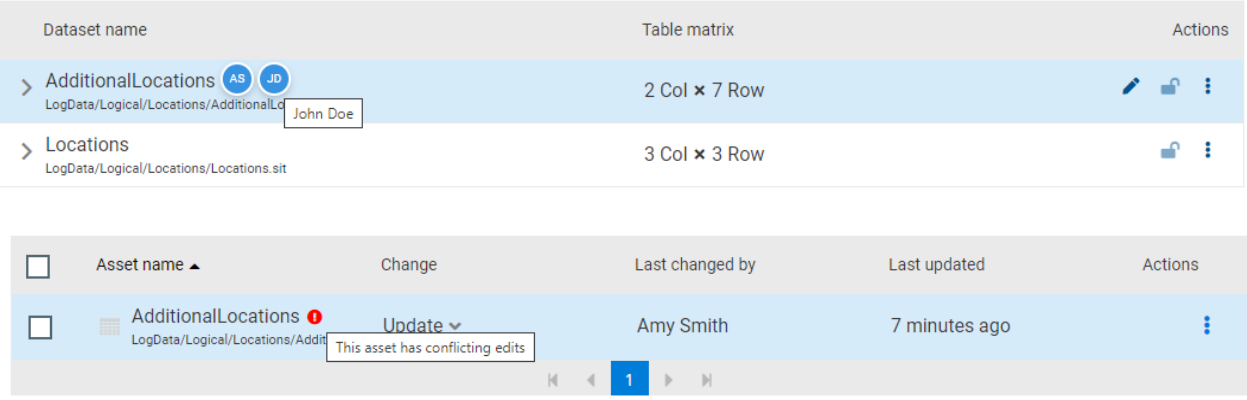
If you are a project **Owner** or **Tester** in Rational® Test Automation Server V10.1.0 or later, you can perform basic tasks in the CSV Editor by right-clicking any row, column, or cell in the dataset to organize your data in a better way. For example, you can perform tasks such as updating data in a cell, inserting or deleting rows and columns, or renaming column names.

When you edit the dataset in the CSV Editor, you can use the following keyboard shortcuts to control the cursor selection in the CSV Editor:

- **Tab** - To move the cursor control to the next available option.
- **Shift-Tab** – To move the cursor control to the previous option.
- **Shift+F10** – To open the context menu from the dataset cell.

After you edit the dataset, you can save the changes made to the dataset, and then you can publish the dataset to the Git repository. If you save and close the edited dataset, the **Changes** page lists the edited dataset and later you can publish to the Git repository for other members to use.

When you or other members of the project edit the same dataset, you can see an icon with the initials of the member next to the name of the dataset on the **Datasets** and **Changes** pages. If you do not see the icon, you must refresh the **Datasets** page to view the icons. The **Changes** page denotes conflicting edits when another member publishes their edited dataset first.




When a member with conflicting edits tries to publish the edited dataset, an error about conflicting changes is displayed. However, the member can use the **Save As** option to save and publish a copy of the dataset edits under a new asset name. The member must discard the edits that were made to the original dataset.

For example, consider a scenario when Amy Smith and John Doe edit the same dataset, and Amy Smith edits and publishes the dataset. On the **Changes** page, a message that indicates a conflict is displayed when John Doe tries to publish the same dataset.

**Note:** You cannot resize the width of rows in the CSV Editor. When you have a large amount of data in a cell, you can right-click the cell and select **Copy** (or Ctrl+C), and then paste it into a text-editing program to view the content. Alternatively, you can hover the mouse over the cell to view the content.

When you have a CSV file that has data separated from a character, then you can import that CSV file into the dataset. You can select any of the following separator characters from the **Configure Dataset** window, and the selection can be the separator character that you used in the CSV file:

- Comma
- Semicolon
- Space
- Tab
- Other

Consider that you have the data in the CSV file in the following format:

| Name;CCNum |
| --- |
| John;1234 5678 1234 5678 |
| Bob;1122 3344 5566 7788 |
| Amy;2233 4455 6677 8899 |

When you import the CSV file in the dataset, and then select the separator value as **Semicolon**, the data in the dataset is displayed as follows:

| | Name | CCNum |
|---|---|---|
| 1 | John | 1234 5678 1234 5678 |
| 2 | Bob | 1122 3344 5566 7788 |
| 3 | Amy | 2233 4455 6677 8899 |

If you want the data in its original format, that is, a semicolon (;) character to separate the data, then you can choose any other separator value from the **Configure Dataset** window.

**Note:** The default separator value is **Comma**.

1. Go to the **Datasets** page and find the dataset which you want to edit.
2. Click the **Edit** icon from the **Actions** column of the dataset.
   **Result**
   The dataset opens in the CSV Editor in a web browser.
3. Perform the following actions to use the options available in the CSV Editor:

| Options | Actions |
|---|---|
| Find and Replace | To find:<br>a. Click the **Find and Replace** icon .<br>b. Enter the content that you want to search in the **Find** field.<br>c. Select any or all the following options to find the search content more effectively:<br>▪ Select the **Case sensitive** checkbox to search the content that is the exact letter case of the content entered in the **Find** field.<br>▪ Select the **Match entire cell contents** checkbox to search for cells that contain only the characters that you have entered in the **Find** field.<br>▪ Select the **Search using regular expression** checkbox to search the pattern that matches strings.<br>For example, to search a cell that contains any number between 0 to 9, do the following:<br>i. Enter \d in the **Find** filed.<br>ii. Select the **Search using regular expression** checkbox.<br>iii. Click **Find**.<br>d. Click **Find**. If the text is found, the cell containing that text is selected.<br>e. Click **Find** again to find further instances of the search text.<br>To find and replace: |

| Options | Actions |
|---|---|
| | a. Click the **Find and Replace** icon .<br>b. Enter the content that you want to search in the **Find** field.<br>c. Enter the content that you want to replace in the **Replace** field.<br>d. Select any or all the following options to find and replace the content more effectively:<br>▪ Select the **Case sensitive** checkbox to find the content that is the exact letter case of the content entered in the **Find** field.<br>▪ Select the **Match entire cell contents** checkbox to find and replace for cells that contain only the characters that you have entered in the **Find** and **Replace** fields.<br>▪ Select the **Search using regular expression** checkbox to find and replace the pattern that matches strings.<br>e. Click **Replace** to replace the individual instances.<br>f. Click **Replace All** to replace every instance of the content throughout the dataset. |
| Undo | a. Click the **Undo** icon .<br>b. Select the recent changes from the list that you want to undo, and then click the list.<br>The **Undo** option undoes anything you do in the dataset. The CSV Editor saves the unlimited undo-able action. You can perform the undo action even after you save your changes made to the dataset. |
| Redo | a. Click the **Redo** icon .<br>b. Select the recent changes from the list that you want to redo, and then click the list.<br>The CSV Editor saves the unlimited redo action. |
| Import | When you have a large amount of data stored in a CSV file, you can import that into a dataset instead of creating a new dataset. You must have a .csv file that contains variable data to import into a dataset.<br>a. Click the **Import** icon .<br>b. Click **Choose File** and select the CSV file that you want to import in the **Import File** dialog box.<br>c. Select one of the following options to append or overwrite data in the dataset:<br>▪ Click **Overwrite** to add the rows and columns from the selected CSV file from the beginning of the dataset.<br>▪ Click **Append** to add rows and columns from the selected CSV file to the end of the dataset.<br>d. Select the **First row contains headers** checkbox if your CSV file contains the header. |
| Export | You can export variable data from the dataset into a CSV file to reuse in future tests when required. You must have a dataset that you want to export. |

| Options | Actions |
|---|---|
| | Click the **Export** icon to download the dataset as a CSV file. |
| Set as current row | During the test run, if you want variable data to be selected from a current row instead of the first row in a dataset, right-click any cell in a row and select **Set as current row**.<br><br>Also, you can set the current row from the **Datasets** page by clicking **Menu**, and then the **Configure** option.<br><br>**When rows are deleted:**<br><br>If you delete any row between row 1 to current row, the current row data is taken from the next row. For example, when you set the current row as 6, and then you delete any row between row 1 to row 6, the current row remains at row 6, but the content of row 7 is moved to row 6.<br><br>**When rows are inserted:**<br><br>If you insert any new row between row 1 to the current row, the current row data is taken from the previous row. For example, when you set the current row as 6, and then you insert any row between row 1 to row 6, the current row remains at row 6, but the content of row 5 is moved to row 6. |
| Dataset configuration settings | In the **Configure Dataset** window, you can change the row and column settings and configure the string values in the dataset that contains variable data for tests to use when they run.<br><br>a. Click the **Menu** icon  , and then select the **Configure** option.<br><br>b. Select any of the separator values that you used in the CSV file.<br><br>The available options are **Comma**, **Semicolon**, **Space**, **Tab**, and **Other**. In the CSV file, if you have any other separator characters other than the available options, then you can select the **Other** option, and then can specify a value.<br><br>For example, if the data in the CSV file is separated by a character #, then select the **Other** option and enter # in the field.<br><br>c. Configure the following options to change the row and column settings:<br><br>▪ **Column header** - Use an up-down control button to increment or decrement the value of the column header.<br><br>▪ **Data start point** - Use an up-down control button to increment or decrement the value of the data starting pointer.<br><br>▪ **Current row** - Use an up-down control button to increment or decrement the value of the current row. |

| Options | Actions |
|---|---|
| | d. Configure the following options to change the string values in the dataset:<br>▪ **Treat as null** - Enter a string value that is to be treated as null when running the test.<br>▪ **Treat as empty** - Enter a string value that is to be treated as empty when running the test.<br>For example, when you run the test and the data *123* in the dataset to be treated as empty, then you can specify *123* in the **Treat as empty** field.<br>▪ **Treat empty text as null** - Select this field when you want the dataset that contains any blank cells, and the value of those blank cells to be interpreted as null.<br>e. Click **Update** to apply the changes. |
| Discard | Click the **Menu** icon and select **Discard** to discard the changes made to the dataset. |

4. Click the **Save** icon to save the changes made to the dataset.
5. Click the **Publish** icon to publish the dataset to the Git repository, and then close the CSV editor.

**Results**

You have edited the dataset.

## Dataset encryption

Encrypted datasets are useful when you want to run tests that contain confidential information such as a set of passwords or account numbers.

When you run a test that uses an encrypted dataset, then you must provide an encryption key for decrypting the encrypted data in columns so that the data can be used in the test. If the test uses data from the multiple encrypted dataset columns, you must enter the same encryption key for every encrypted dataset column that the test uses.

When you run the test that uses the dataset with the encrypted column, the value of the column is decrypted at a run time. The data in the column is sent as a clear-text string in requests to the server. The actual values of the encrypted dataset variables are not displayed in the test log. The test log displays asterisks for the encrypted dataset variables.

You can use only one encryption key to encrypt data in the columns in any dataset.

**Important:**

The encryption keys that you use to encrypt data in a dataset are not stored on the server nor can be retrieved from the server. Therefore, you must remember to store the encryption keys in a secure location. You must use the same encryption keys to perform the following operations:

- View the encrypted values
- Decrypt data
- Enable the use of the encrypted dataset during test runs

## Encrypting a dataset column

To secure test data, you must encrypt datasets. You can encrypt data in the columns of a dataset by using an encryption key. When you run a test that utilizes a dataset with encrypted variables, you must enter the encryption key for the encrypted column that the test uses.

**Before you begin**

You must have created a dataset. See Creating a dataset on page 84.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the Rational® Test Automation Server UI.
4. Go to the **Datasets** page and find the dataset that you want to encrypt.
5. Click the **Edit** icon from the **Actions** column of the dataset.

   **Result**

   The dataset opens in the CSV Editor in a web browser.
6. Right-click any cell in a column that you want to encrypt and select **Encrypt column data**.

   **Result**

   The **Encrypt Column** window is displayed.
7. Enter an encryption key in the **Encryption Key** field to encrypt the data in the column.

   **Remember:** When you have already encrypted other columns in the dataset, you must enter the same encryption key that you used previously. You can use only one encryption key to encrypt columns in a dataset.

   **Important:**

   The encryption keys that you use to encrypt data in a dataset are not stored on the server nor can be retrieved from the server. Therefore, you must remember to store the encryption keys in a secure location. You must use the same encryption keys to perform the following operations:

- View the encrypted values
- Decrypt data
- Enable the use of the encrypted dataset during test runs

8. Click **Encrypt Column**.
   **Result**
   Asterisks are displayed instead of actual data for the encrypted column.

**Results**

You have encrypted the dataset column in your project.

**What to do next**

You can publish the dataset to the Git repository so that other members of the project can use the dataset. See Publishing a dataset on page 93.

## Decrypting a dataset column

To view the content of an encrypted dataset, you can decrypt the dataset column. Removing encryption from a dataset revokes the protection offered to the test data.

**Before you begin**

You must have created at least one dataset and encrypted the dataset with an encryption key. See Creating a dataset on page 84 and Encrypting a dataset column on page 91.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the Rational® Test Automation Server UI.
4. Go to the **Datasets** page and find the dataset that you want to decrypt.
5. Click the **Edit** icon from the **Actions** column of the dataset.
   **Result**
   The dataset opens in the CSV Editor in a web browser.
6. Right-click encrypted cells that display the contents with asterisks, and then select **Decrypt column data**.
   **Result**
   The **Decrypt Column** window is displayed.
7. Enter the encryption key that you used to encrypt the data in the column in the **Encryption Key** field.
8. Click **Decrypt Column**.
   **Result**
   Asterisks are replaced with the actual data in the decrypted column.

**Results**

The encryption is removed from the selected column in the dataset. When you run a test that uses a dataset that contains decrypted data, the variable data is substituted for the original data in the recorded test without prompting for the encryption key.

**What to do next**

You can publish the dataset to the Git repository so that other members of the project can use the dataset. See Publishing a dataset on page 93.

## Publishing a dataset

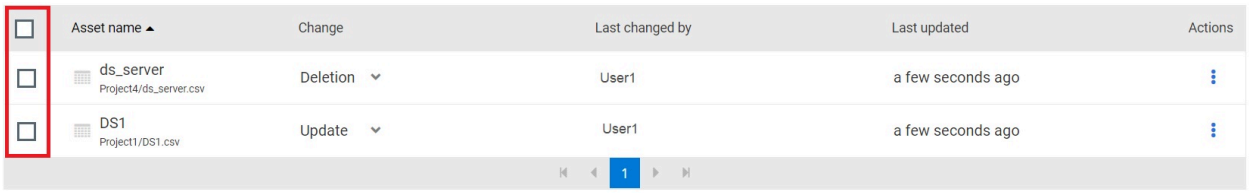
When you create or edit any datasets in Rational® Test Automation Server, you can publish your changes to the Git repository. Therefore, when you publish a dataset, other members in the project can use your dataset in their test assets run, if required.

**Before you begin**

You must have created, edited, or deleted dataset assets in Rational® Test Automation Server.

**About this task**

The **Changes** page lists dataset assets that are modified. You can publish all datasets or a single dataset listed in the **Changes** page to the Git repository by selecting the appropriate checkboxes.




Rational® Test Automation Server processes one publish request at a time. Therefore, when multiple users attempt to publish the same dataset, the request that reaches first from the users is processed. The other users receive an error message, and they are unable to publish.

When you modify the dataset and publish it to the Git repository, the other members who have access to that dataset can view the updated dataset.

You can also perform the following actions from the Changes page by clicking the **Menu** icon .

- **Edit** - If you want to make any last-minute updates to the dataset before publishing.
- **Discard** - To remove the changes that you made to the dataset asset.
- **Save As** - To save a copy of dataset.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the Rational® Test Automation Server UI.

4. Go to the **Changes** page, find the dataset that you want to publish.
5. Select the checkbox that precedes the **Dataset name** to publish to the Git repository, and then click **Publish**.
6. Enter a description for the changes made to the dataset, and then click **Publish**.

   **Result**

   A message is displayed for successful pushing of changes to the Git repository.

**Results**

You have published the modified dataset into the Git repository.

## Viewing a dataset

When test assets includes datasets, you can view the contents of a dataset from Rational® Test Automation Server. The datasets residing in the Git repository are listed in the **Datasets** page.

**Before you begin**

You must be a member of the project with the **Owner** or **Tester** role and must have completed the following tasks:

- Created a project in Rational® Test Automation Server. See Adding details to a server project on page 282.
- Configured the Git repository in your project. See Adding repositories to a server project on page 282.
- Created at least one dataset and encrypted the dataset with an encryption key. See Creating a dataset on page 84 and Encrypting a dataset column on page 91.
- Created a classification for an encrypted dataset. See Creating a classification on page 302.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the Rational® Test Automation Server UI.
4. Go to the **Datasets** page, find the dataset that you are interested in, and then expand the dataset by clicking the **Expand** icon ›.

   **Note:** You can click the **Dataset Name** field to sort the datasets by name in alphabetical order. Alternatively, you can use the **Search** field to search the dataset by name.

**Results**

You have viewed the contents of the dataset.

## Viewing an encrypted dataset

You can use the **Dataset** page to view the contents of an encrypted dataset from Rational® Test Automation Server.

1. Go to the **Datasets** page, find the encrypted dataset that you are interested in, and then expand the dataset by clicking the **Expand** icon .
2. Perform the following steps, if you have created a classification but yet to add the encrypted dataset to it:
    a. Find the encrypted dataset that you are interested in, and then click the **Lock** icon from the **Actions** column of the dataset.
    b. In the **Change the classification for the dataset** window, select a classification from the list and enter the encryption key set for the dataset.
    c. Click **Save** to save the classification details.
3. Right-click the encrypted column and click **Show encrypted data**.
   You can right-click the decrypted column and select **Hide encrypted data** to encrypt the data again.

**Important:** The **Show encrypted data** option is available only when you have added the encrypted dataset to the respective classification.

**Results**

You have viewed the contents of an encrypted dataset.

## Changing classification for an encrypted dataset

To move the encrypted dataset from one classification to another, you can change the classification for an encrypted dataset from the **Dataset** page.

**Before you begin**

You must have created at least two or more classifications. See Creating a classification on page 302.

1. Go to the **Datasets** page and find an encrypted dataset for which you want to change the classification.
2. Click the **Lock** icon from the **Actions** column of a dataset.
3. In the **Change the classification for the Dataset** window:
    a. Choose the classification from the list.
    b. Enter the encryption key for the dataset that was set while encrypting the dataset column.
    c. **Save** the classification details.

**Remember:** You can perform this task only if you know the encryption key set for the dataset while encrypting the dataset column.

**Results**

You have changed the classification for an encrypted dataset.

Related information

Managing an encrypted dataset on page 302

Creating a classification on page 302

## Deleting a dataset

You can delete the dataset when it is not required in your test environment.

**Before you begin**

You must have at least one dataset asset in your Git repository that you have configured.

**About this task**

You can delete datasets in Rational® Test Automation Server in the following scenarios:

- Datasets that you created in the desktop clients and that are cloned to the repository in your project on Rational® Test Automation Server.
- Datasets that you created in Rational® Test Automation Server and that are published to the repository in your project.
- Datasets that are in the `.csv` file format.

You cannot delete the following datasets:

- Datasets that you created or edited in Rational® Test Automation Server and that are not published to the repository in your project.
- Datasets that are in the `.sit` file format.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the Rational® Test Automation Server UI.
4. Go to the **Datasets** page, find the dataset that you want to delete.
5. Click the **Menu** icon ⋮ from the **Actions** column of a dataset, and then click **Delete**.
6. Clear the **Publish delete to *<branch name of the configured Git repository>*** checkbox to delete the dataset from your project.

   When you clear this option, the deletion of the dataset does not reflect in the configured Git repository. Therefore, the other members of the project can still use the dataset that you have deleted. If you want, later you can publish the dataset to the Git repository from the **Changes** page.

**Note:** By default, **Publish delete to *<branch name of the configured Git repository>*** field is selected.

7. Enter a description for deleting the dataset in the **Description of change** field.
8. Perform the following action:

| Option | When | Action |
|---|---|---|
| **Publish delete to *<branch name of the configured Git repository>*** | Selected ☑ | Click **Delete and Publish** |
| | Cleared ☐ | Click **Delete** |

**Result**

A message is displayed for the successful deletion of the dataset.

# Chapter 6. Test Execution Specialist Guide

This guide describes tasks that you must complete before you can configure and run tests in Rational® Test Automation Server. You can find information about other tasks that you can perform on the Resource Monitoring Service, Virtualization, and Integrations with third-party applications. This guide is intended for testers or test execution specialists.

## Prerequisites to running tests

Before you can configure and run a test in a project on Rational® Test Automation Server, you must read the information about the different tests. You might want to add a remote static agent or a remote Docker host to the project as an alternate location to run the tests.

You can find the following information:

---

Related information

Tests configurations and test runs

### Test run considerations for AFT Suites

Before you configure an AFT Suite run, you must read the considerations you must take into account.

When you want to run an AFT Suite from a project on Rational® Test Automation Server, you can check if the AFT Suite meets any of the following conditions:

| If... | Then... |
|---|---|
| The AFT Suite uses the settings configured in an AFT XML file for a remote agent | You must ensure that you have configured the location element in the AFT XML to point to a remote agent.<br><br>For more information about setting the location element, refer to Using an XML file to run multiple Web UI tests and compound tests simultaneously. |
| You want to run an AFT Suite on a remote agent | You must run the test on the remote agent before you commit the test asset to the remote repository.<br><br>When you run the test on the remote agent, the following events occur: |

| If... | Then... |
|---|---|
| | • The agent is added as the location in the AFT XML file, on which the test is to be run.<br>• The agent is displayed as the agent on which the test is to be run, under the **Host** column within the **Location** tab in the **Execute test asset** dialog box. |
| You commit the test asset to the remote repository without running the test on the remote agent | The agent is not displayed under the **Host** column within the **Location** tab in the **Execute test asset** dialog box. |

Before you configure the run for an AFT Suite, you must complete the following tasks, if they are applicable:

- Add the remote agents, on which the test is to be run, to the project on Rational® Test Automation Server. See Adding an agent to the project for running tests on page 106.
- After you add the agent, you can select the agent configured in the AFT XML file as the location for the test run or select the remote agents that you add to the project, when you configure a run for an AFT Suite.

  The remote agents are shown as the agents available for selection under the **Override** column within the **Location** tab in the **Execute test asset** dialog box.

  You can then select the remote agent as the location to run the AFT Suite when you are configuring the test run. Alternatively, you can enter the argument `-swaplocation <configured_agent_location>:<overriding_agent_location>` in the **Program Arguments** field in the **Advanced Setting** dialog box when you are configuring the test run.

For running an AFT Suite, see Configuring an AFT Suite run on page 123.

---

Related information

Tests configurations and test runs

## Test run considerations for API Suites

Before you configure an API Suite run, you must read the considerations you must take into account.

You can find the following information about API Suites:

- Important information about API Suites on page 100
- Important information about report configuration in API Suites on page 102

### Important information about API Suites

When you want to run an API Suite from a project on Rational® Test Automation Server and the test suite in Rational® Integration Tester meets any of the following conditions:

- The test suite refers to local stubs that use a transport other than HTTP or MQ.
- The test suite refers to local stubs that use the HTTP transport.
- The transport used in the tests in the test suite is configured with the host name set to *localhost* for the HTTP/TCP proxy.
- The test suite has tests that use a transport and the transport requires third-party application `Jar` files for a successful run.

You must then refer to the following table for the next steps:

| If... | Then... |
|---|---|
| The tests in an API Suite refer to local stubs that use a transport other than HTTP or MQ. | You must perform any of the following actions before you commit the test asset to the remote repository:<br>• Remove the reference to the local stub from the test suite.<br>• Publish the stubs to Rational® Test Control Panel.<br>To publish and edit stubs, see Publishing stubs. |
| The tests in an API Suite refer to local stubs that use the HTTP transport. | You must perform the following actions before you commit the test asset to the remote repository:<br>1. Create a test suite that contains the stub and tests that run on the stub.<br>2. Configure the physical HTTP transport with the following parameters in the **Client** tab in the **Physical View**:<br>◦ Set the `Host name or IP address of the HTTP Proxy server` in the **Proxy Host** field.<br>◦ Set *3128* as the proxy server port in the **Proxy Port** field.<br>3. Set the reference to the stub for the tests in the test suite scenario from the **Scenario Editor** dialog box from the **Test Factory** view by selecting the *Live system* option in the **Satisfied by** column |

| If... | Then... |
| --- | --- |
| | to run the stubs on the HTTP proxy registered with Rational® Test Automation Server.<br>4. Save the project. |
| The transport used in the tests in an API Suite is configured with the host name set to *localhost* for the HTTP/TCP proxy. | You must replace the host name with *fully-qualified-domain-name* or *IPAddress* of the proxy host in the tests before you commit the test asset to the remote repository. |
| You plan to run the API Suite on the computer where Rational® Test Automation Server is installed and the API Suite contains tests that use a transport from any of the following third-party applications:<br>• Camel<br>• CentraSite<br>• CICS<br>• Coherence<br>• Database<br>• IMS<br>• Integra<br>• JMS<br>• SAP RFC<br>• Software AG Universal Messaging<br>• TIBCO EMS<br>• TIBCO Rendezvous<br>• TIBCO SmartSockets<br>• WebSphere Application Server Service Integration Bus (SiBus)<br>• WebLogic<br>• Software AG webMethods<br>• WebSphere MQ | You must complete the following tasks:<br>1. Identify the location of the third-party application JAR files. You can use Library Manager to know the location where the third-party application JAR files are saved.<br>2. Copy the third-party application JAR files to the computer where Rational® Test Automation Server is installed.<br>See Copying third-party application Jars to Kubernetes on page 74. |
| You plan to run the API Suite on a remote Docker host and the API Suite contains tests that use a transport from any of the following third-party applications: | You must complete the following tasks: |

| If... | Then... |
|---|---|
| • Camel<br>• CentraSite<br>• CICS<br>• Coherence<br>• Database<br>• IMS<br>• Integra<br>• JMS<br>• SAP RFC<br>• Software AG Universal Messaging<br>• TIBCO EMS<br>• TIBCO Rendezvous<br>• TIBCO SmartSockets<br>• WebSphere Application Server Service Integration Bus (SiBus)<br>• WebLogic<br>• Software AG webMethods<br>• WebSphere MQ | 1. Identify the location of the third-party application JAR files. You can use Library Manager to know the location where the third-party application JAR files are saved.<br>2. Copy the third-party application JAR files to the computer where the remote Docker host is installed.<br><br>See Copying third-party application Jars to a remote Docker host on page 113. |

### Important information about report configuration in API Suites

You must configure test suites in Rational® Integration Tester to use a results database so that the details of the test results can be captured and displayed in the Rational® Test Automation Server API results reports. See Configuring the project results database.

If your server projects use either a Microsoft SQL Server, MYSQL, or both as a project results database, you must copy the Microsoft SQL Server and MySQL database JAR files to the JDBC folder in the path `/data/JDBC` on IBM® Rational® Test Automation Server. See Copying third-party application Jars to Kubernetes on page 74.

The database JAR files that you need are based on the version of Microsoft SQL Server or MQSQL that you use with Rational® Integration Tester. See Adding Microsoft SQL Server and MySQL drivers.

To run an API Suite, see Configuring an API Suite run on page 130.

---

Related information

Scenario reference setting

Tests configurations and test runs

## Test run considerations for schedules

Before you configure a Rate Schedule or VU Schedule run, you must read the considerations that you must take into account.

### Important information about JMeter tests in schedules

When you want to run JMeter tests as part of the VU Schedule or Rate Schedule on Rational® Test Automation Server, you must have completed the following tasks:

- Installed the IBM® Rational® Performance Tester Agent on a computer.
- Installed the JMeter application on the computer where you have installed the IBM® Rational® Performance Tester Agent.
- Set the environment variable **JMETER_HOME** that points to the JMeter installation directory.

### Important information about Resource Monitoring sources in schedules

When you want to run schedules on Rational® Test Automation Server, you must have completed the following tasks in Rational® Performance Tester:

- Created a performance schedule of **Rate** or **VU** type.
- Added Resource Monitoring sources, or the labels of Resource Monitoring sources to a schedule. Refer to Adding resource monitoring sources to a performance schedule by using labels .

Before you configure a run of a Schedule, you must complete the following tasks in Rational® Test Automation Server:

- Add Resource Monitoring sources to your project in Rational® Test Automation Server. See Monitoring host resources on page 215.
- Enable Rational® Test Automation Server as Resource Monitoring service only if you want to replace Resource Monitoring labels set in Rational® Performance Tester with the labels that you created in Rational® Test Automation Server. See Controlling Resource Monitoring sources in a schedule.
- Perform the following steps when you want to override the Resource Monitoring labels in the test asset with the Resource Monitoring labels that you create in Rational® Test Automation Server:
  1. Click the **Resource Monitoring** tab.

     **Note:**

     The **Resource Monitoring** tab displays only if you enabled the **Resource Monitoring from Service** option in Rational® Performance Tester when you created the test asset.

  2. Click , and press the `Ctrl` + `Space bar` keys, or enter the initial letter of a label to select a label in the list.

**Note:**

You can select or add labels only if you added the labels in the **Resource Monitoring Sources** page in Rational® Test Automation Server.

For running a Rate Schedule or VU Schedule, see Configuring a run of a Rate Schedule or VU Schedule on page 168.

Related information

Tests configurations and test runs

## Considerations for using Jaeger traces in reports

Before you can use Jaeger to report the test results in the supported tests, you must read the considerations that you must take into account.

When you want to enable Suites, Tests, or Schedules that are run from Rational® Test Automation Server to use Jaeger traces for reporting the test results, you must have completed the following tasks:

- Installed Jaeger at the time of installation of the server software. See Prerequisites for installing the server software on Red Hat OpenShift on page 26 or Prerequisites for installing the server software on Ubuntu on page 35.
- Verified that Jeager is enabled and working.

After you install and verify that Jaeger is enabled, and when you want Jaeger to report the test results for the tests that you configure for a run, you must perform the following steps:

1. Click **Advanced**.
2. Enter `-history jaeger` in the **Program Arguments** field.

Related information

Tests configurations and test runs

# Agents overview

You can find information about the agents that you must install on remote computers and configure the agents with the offline user token so that the agents register with IBM® Rational® Test Automation Server. After the agents are registered, you can add them to your projects and then select them as locations to run tests.

When you plan to run any of the tests on a remote agent from a project on Rational® Test Automation Server, you must have completed the following tasks:

- Installed the agent of the same version as the version of Rational® Test Automation Server.
- Configured the agent with your offline user token so that the agent can connect and also register with Rational® Test Automation Server.

  **Note:** When you configure the agent to connect to the server by using the offline user token that you generated on Rational® Test Automation Server, you become the owner of the agent.

- Verified that you have write permissions to the `majordomo.config` file so that certain transactions can be written by Rational® Test Automation Server to the `majordomo.config` file.

  **Note:** You can verify if you have the *Read* and *Write* permissions by checking the properties of the `majordomo.config` file.

- Added the remote agents that are registered with Rational® Test Automation Server to your project on Rational® Test Automation Server. See Adding an agent to the project for running tests on page 106.

  **Note:** After you add the remote agents to your project, Rational® Test Automation Server determines the capabilities that are provided by the remote agent and displays the details and capabilities of the agent on the **Infrastructure** page.

---

Related information

Agent installation

## Test run considerations for running tests on remote agents

Before you configure a test to run on a remote agent, you must read the considerations you must take into account.

You can run the following tests on a remote agent:

- AFT Suite
- Compound Tests that contains performance tests
- Compound Tests that contains Web UI tests
- Compound Tests that contains traditional HTML tests
- Rate Schedule
- VU Schedule

When you plan to run any of the tests on a remote agent from a project on Rational® Test Automation Server, you must complete the following tasks:

- Install the agent V10.0.2 or later.
- Start the agent on which you want to run an AFT Suite as a non-admin user. Ensure that you have write permissions for the `majordomo.config` file.
- Configure the agent with your offline user token to connect to IBM® Rational® Test Automation Server. The agent can be connected to both the server and desktop clients.

Before you add the tests to the remote repository, ensure that you have completed the following tasks:

- Associated the agents with the performance schedule in Rational® Performance Tester.
- Specified the host name of the remote workstation in the AFT XML file.

You must add the remote agents to your project on IBM® Rational® Test Automation Server. See Adding an agent to the project for running tests on page 106.

---

Related information

Tests configurations and test runs

Agent installation

Running user group at a remote location

Running multiple tests on multiple browsers and devices simultaneously

## Adding an agent to the project for running tests

If you have performance tests that distribute workload across different workstations or Accelerated Functional Testing (AFT) Suites that need to run on different platforms and browsers, you can add IBM® Rational® Performance Tester Agents to your projects to run tests on remote workstations.

**Before you begin**

You must have completed the following tasks:

- Read about the considerations that you must take into account before you configure a test run to run on a remote agent. See Test run considerations for running tests on remote agents on page 105.
- Been assigned the *Owner* or *Tester* role in the project to which you want to add the agent.
- Owned at least one agent that is not added to the project.

**About this task**

When you configure the agent to connect to the server, you must specify an offline user token. This token defines your ownership over the agent. Only you can add or remove this agent from the project. However, any member of your project with the permission to run the tests can initiate the run of the performance schedule and AFT Suites that are associated with the agent.

1. Log in to Rational® Test Automation Server, open your project, and go to the **Infrastructure** page.
2. Click **Add Agents**.

   The agents that you configured with your user token are displayed.

3. Select the agents that you want to use and click **Add**.

   The agents that you added are displayed.

4. Perform the following steps to view details and capabilities of the agents:

   a. Expand the agent to view the details panel.

      You can find the following details of the agent displayed:

| **Parameter** | **Description** |
|---|---|
| Version | The version of the installed agent. |
| Address | The IP address of the computer on which the agent is installed. |
| License type | The license type that is enabled for you to use the agent. |
| Load generation service | The status of the agent whether it can be used for load generation service. |
| Last contact | The time of the last contact with the agent by Rational® Test Automation Server. |

   b. Expand the capabilities panel to view the system and application capabilities.

      You can find the following capabilities of the agent displayed:

| | **Capability** | **Description** |
|---|---|---|
| System capabilities | Architecture | The architecture of the CPU processor of the computer on which the agent is installed. |
| | Operating system | The operating system of the computer on which the agent is installed. |
| | RAM | The memory that is allotted to RAM in the computer on which the agent is installed. |
| Application capabilities | Browser | The browsers along with their version that are installed on the computer on which the agent is installed. |

c. Check for the status of the agent from the **Status** column.

**Note:** You can add agents that are in any state to the project. However, the agent must be in the `Ready` state for you to select the agent as the location for running the test.

**Results**

You have added remote agents to your project.

**What to do next**

You can run the tests on the agent that is added to the project from the **Execution** page. See Test run configurations on page 121.

Related information

Agent installation

Running user group at a remote location

Running multiple tests on multiple browsers and devices simultaneously

# Managing Docker hosts

You can set up dedicated external Docker hosts to run large tests or run tests remotely to free up the dependency of running tests on the default cluster of IBM® Rational® Test Automation Server.

You can set up the remote computer to connect in a secure or non-secure mode with IBM® Rational® Test Automation Server. You can register the Docker host with IBM® Rational® Test Automation Server to communicate in either a secure mode or a non-secure mode. You must add the registered remote Docker host to your project before you can run tests on the remote Docker host location. You can also, remove the Docker hosts from a project or unregister it from IBM® Rational® Test Automation Server.

You must copy the third-party application `Jar` files to the remote Docker host if you want to run tests on the remote Docker host.

## Test run considerations for running tests on remote Docker hosts

Before you can configure a test to run on a remote Docker host, you must read the considerations that you must take into account.

You can run all the tests that are supported on Rational® Test Automation Server on remote Docker hosts.

**Important:** You can run stubs only in the **Default Cluster** location of Rational® Test Automation Server. You cannot run stubs on a remote Docker host.

When you plan to run any of the tests on a remote Docker host from a project on Rational® Test Automation Server, you must complete the following tasks:

| | Tasks | For information, go to... |
|---|---|---|
| 1.1 | Set up non-secure remote Docker hosts. | Setting up a remote Docker host computer on page 110 |
| 1.2 | Set up secure remote Docker hosts. | Setting up a secure remote Docker host computer on page 111 |
| 2 | Register the remote Docker host on Rational® Test Automation Server. | Registering a remote Docker host with IBM Rational Test Automation Server on page 115 |
| 3 | Add a remote Docker host to the Rational® Test Automation Server project. | Adding a remote Docker host to the project for running tests on page 117 |
| 4 | Configure a run for your test that you want to run on the remote Docker host by performing the following actions:<br>1. Go to Test run configurations on page 121.<br>2. Check for the prerequisite tasks that you must perform before you configure a test run for the test.<br>3. Select the task for the test that you want to configure for a run, and then follow the steps to run the test. | 1. Test run configurations on page 121<br>2. Prerequisites to running tests on page 98<br>3. Test run configurations on page 121 |

If you plan to run API Suites that use a transport and the transport requires third-party application `Jar` files for a successful run on a remote Docker host, then you must perform the following task:

- You must copy the third-party application `Jar` files to the third-party application folder on the computer where the remote Docker host is installed. See Copying third-party application Jars to a remote Docker host on page 113.

After you set up and register the remote Docker host with Rational® Test Automation Server, you can add the remote Docker hosts to your project on Rational® Test Automation Server. See Adding a remote Docker host to the project for running tests on page 117.

Related information

Tests configurations and test runs

Managing Docker hosts on page 108

## Setting up a remote Docker host computer

You must set up a computer to host the remote Docker host. You can set up the remote host with a non-secure mode of connection with IBM® Rational® Test Automation Server.

**Before you begin**

You must have identified a remote computer or VM that has a minimum of 16 GB RAM. You must have installed Ubuntu V18.04 or later, on the remote computer.

You must have installed Docker Enterprise Edition for a non-secure mode of connection with IBM® Rational® Test Automation Server on the remote Docker host computer. Refer to Get Docker Engine.

You must have the entitlement key that is required for you to connect to the IBM Docker images library to be able to pull in the software image to the remote Docker host.

**About this task**

You must set up a remote computer or a Virtual Machine (VM) on which you want to set up the remote Docker host. You must be a system administrator or get the remote Docker host set up by a system administrator. You must ensure that the system administrator completes the following actions:

- Enables SSH.
- Creates user credentials for you with root permissions.
- Provides the name of the Docker container.
- Provides the IP address of the remote Docker host computer.

1. As a system administrator of the remote system, specify a port to access the Docker daemon in the `docker.service` file located in the directory `/lib/systemd/system`. You must add the configured port at the line in the file:

   ```
   ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:PORT
   ```

   For example, if the port configured is *4342*, then the line must read as:

   ```
   ExecStart=/usr/bin/dockerd -H fd:// -H tcp://0.0.0.0:4342
   ```

2. Restart the Docker service by using the following commands:

   $ sudo systemctl daemon-reload

   $ sudo systemctl restart docker

3. Perform the following steps as a user with root permissions or ask the system administrator to complete the following steps:

    a. Open a terminal and connect to the remote Docker host by using an SSH tool by running the following command:

    ssh root@<IP_address_of_remote_host>

    b. Log in to IBM Docker images library by running the following command:

    sudo docker login cp.icr.io -u <username> -p <password>

    **Note:** You must provide the following username and password in the command to log in to the registry:
        - Username as cp
        - Password as <entitlement_key>

    c. Run the following commands to pull the product images from the registry on to the remote Docker host computer:

    **Note:** You must pull in the images with the version which is the same as the version of IBM® Rational® Test Automation Server that you use to run the tests.

    - sudo docker pull cp.icr.io/cp/<ibm-rtw_docker_image>
    - sudo docker pull cp.icr.io/cp/<ibm-rtvs_docker_image>

**Results**

You have set up the remote Docker host computer that can be accessed in a non-secure mode.

**What to do next**

You can set up a secure remote Docker host computer optionally, that enables a secure connection with Rational® Test Automation Server. See Setting up a secure remote Docker host computer on page 111.

You can register the remote Docker host on Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.

---

## Setting up a secure remote Docker host computer

You must set up a computer to host the remote Docker host. You can set up the remote host with a secure mode of connection with IBM® Rational® Test Automation Server by using certificates issued by the remote host computer that authenticate the connection.

**Before you begin**

You must have identified a remote computer or VM that has a minimum of 16 GB RAM. You must have installed Ubuntu V18.04 or later, on the remote computer.

You must have installed Docker Community Edition V18.09 or later. For instructions, refer to Get Docker Engine.

**Note:** For the remote Docker host to be reached in a safe manner through the network, you can enable *TLS* by specifying the **tlsverify** flag and by pointing the **tlscacert** flag of the remote Docker host to a trusted CA certificate. Refer to Protect the Docker daemon socket.

You must have the entitlement key that is required for you to connect to the IBM Docker images library to be able to pull in the software image to the remote Docker host.

**About this task**

You must set up a remote computer or a Virtual Machine (VM) on which you want to set up the remote Docker host. You must be a system administrator or get the remote Docker host set up by a system administrator. You must ensure that the system administrator completes the following actions:

- Enables SSH.
- Creates user credentials for you with root permissions.
- Provides the name of the Docker container.
- Provides the IP address of the remote Docker host computer.

Perform the following steps as a user with root permissions or ask the system administrator to complete the following steps:

a. Open a terminal and connect to the remote Docker host by using an SSH tool by running the following command:

   ssh root@<IP_address_of_remote_host>

b. Log in to IBM Docker images library by running the following command:

   sudo docker login cp.icr.io -u <username> -p <password>

   **Note:** You must provide the following username and password in the command to log in to the registry:
   - Username as cp
   - Password as <entitlement_key>

c. Run the following commands to pull the product images from the registry on to the remote Docker host computer:

**Note:** You must pull in the images with the version which is the same as the version of IBM® Rational® Test Automation Server that you use to run the tests.

- sudo docker pull cp.icr.io/cp/<ibm-rtw_docker_image>
- sudo docker pull cp.icr.io/cp/<ibm-rtvs_docker_image>

**Results**

You have set up the remote Docker host computer that can be accessed in a secure mode.

**What to do next**

You can optionally, set up a non-secure remote Docker host. See Setting up a remote Docker host computer on page 110.

You can register the remote Docker host on Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.

---

Related reference

Task flow for working with remote Docker hosts

## Copying third-party application Jars to a remote Docker host

You can run API Suites in a project on Rational® Test Automation Server on a remote Docker host. If the API Suites use a transport and the transport requires third-party `Jar` files for a successful run, you must ensure that the third-party application `Jar` files are available at the test run time. To achieve this, you must copy the third-party application `Jar` files to the computer where you have set up the remote Docker host.

**Before you begin**

If you want to copy the third-party application `Jar` files to the to the computer where Rational® Test Automation Server is installed on Kubernetes, see Copying third-party application Jars to Kubernetes on page 74.

You must have server administrator privileges.

You must have completed the following tasks:

- Set up and configured the remote Docker host. See Managing Docker hosts on page 108.
- Identified the third-party application `Jar` files that are required and copied the `Jar` files. See Test run considerations for API Suites on page 99.
- Copied the `Jar` files of the third-party application jars to the directory or folder on the remote Docker host from where you can run the docker commands.

**About this task**

You can copy the required third-party application `Jar` files to the folder that is specific for the application under the `/myFiles/Userlibs/<application_name>` folder. You need not extract the files.

You can perform this task any time after you have configured the remote Docker host and you plan to run an API Suite on the remote Docker host. The API Suite uses transports and the transports require third-party application `JAR` files to be available at the test run time.

You can get help on the docker commands by running the command: $ docker help from the docker command line. For more information about the docker commands, refer to the Docker command line documentation.

1. Use the following table to find the name of the folder that corresponds to the specific third-party application for the transport used in the API Suite.

   **Note:** You must provide the name of the folder listed for the third-party application as the `<application_name>` in the docker command.

   **Table 7. Name of the folder for the application**

| Application | Name of the folder to use |
|---|---|
| Camel | Camel |
| CentraSite | CentraSite |
| CICS | CICS |
| Coherence | Coherence |
| Database | JDBC |
| IMS | IMS |
| Integra | Integra |
| JMS | JMS |
| SAP RFC | SAP |
| Software AG Universal Messaging | SoftwareAGUM |
| TIBCO EMS | TIBCO |
| TIBCO Rendezvous | |
| TIBCO SmartSockets | |
| WebSphere Application Server Service Integration Bus (SiBus) | WAS |
| WebLogic | WebLogicJMX |
| Software AG webMethods | webMethods |
| WebSphere MQ | WMQ |

2. Run the following docker command to create or update the volume that contains the application `Jar` files:

   docker run --rm -v /<anyFolder>/UserLibs/<application_name>:/ulsrc -v userlibs/<application_name>:/uldest alpine:latest cp -r /ulsrc/. /uldest/UserLibs/<application_name>

   **Attention:** You must run the docker command when no other tests are running in the Docker host to prevent concurrent access problems.

**Results**

You have successfully copied the third-party application `Jar` files to the folder in the remote Docker host.

**What to do next**

- You can register the remote Docker host, if you have not already done so. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.
- You can configure the API Suite run. See Configuring an API Suite run on page 130.

## Registering a remote Docker host with IBM® Rational® Test Automation Server

You must register the remote Docker host with Rational® Test Automation Server before you add the remote Docker host to your project, which contains the test assets that you want to run on the remote Docker host.

**Before you begin**

Depending on the mode of connection with Rational® Test Automation Server, you must have set up the remote Docker host computer either for a secured mode or unsecured mode of connection. See Setting up a remote Docker host computer on page 110 or Setting up a secure remote Docker host computer on page 111.

1. Log in to Rational® Test Automation Server.
2. Click the **User** icon on the menu bar and select **Docker Hosts**.

   The **Manage docker hosts** dialog box is displayed.

3. Click **New docker host**.
4. Enter the *host name* and the *port* of the remote Docker host in the format *<host name or IP_address>:<port>* in the **Remote Docker Host** field.

   If you do not provide the port that you have configured for the remote Docker host, then the default port that is used is as follows:
   - Port *2375* for the unsecured mode
   - Port *2376* for the secured mode
5. Choose from any of the following authentication modes to establish the connection between Rational® Test Automation Server and the remote Docker host computer:

- For the secured mode, the **Secure mode** checkbox is selected by default.

  Perform the following steps:

  a. Select **Secure mode**, if it is not selected.
  b. Click the action labels to browse and select the `.pem` files that correspond to `ca_certificate.pem`, `client_certificate.pem`, and `client_key.pem`.

  **Note:** You must have generated the `.pem` files on the remote Docker host computer and copied the files to your local drive.

- For the unsecured mode, the **Secure mode** checkbox must not be selected.

  Perform the following step:

  a. Clear the **Secure mode** checkbox, if selected.

6. Click **Test connection** to test if a connection is established between Rational® Test Automation Server and the remote host computer.

   - On a successful connection, a message is displayed.
   - On a failure to connect to the remote host computer, an error message is displayed. You must resolve the error and reattempt to establish a successful connection.

   **Important:** The remote Docker host computer must be connected successfully before you can register the remote Docker host with Rational® Test Automation Server.

7. Click **Register**.

   The remote Docker host that you registered is displayed.

   **Important:** The remote Docker host must be registered successfully with Rational® Test Automation Server before you add the remote Docker host to your project.

8. Click **Close** to exit.

**Results**

You have registered the remote Docker host with Rational® Test Automation Server.

**What to do next**

You must add the registered remote Docker host to your project before you use the remote Docker host as a location to run tests in your project.

Related reference

Task flow for working with remote Docker hosts

## Adding a remote Docker host to the project for running tests

You can choose to run tests on Docker hosts that you have set up on remote host computers. You must register the Docker host with IBM® Rational® Test Automation Server and then add them to your project before you run tests on the remote Docker hosts.

**Before you begin**

You must have completed the following tasks:

- Read about the considerations you must take into account before you configure a test run to run on a remote Docker host. See Test run considerations for running tests on remote Docker hosts on page 108.
- Set up the remote Docker host system. See Setting up a remote Docker host computer on page 110 or Setting up a secure remote Docker host computer on page 111.
- Registered the remote Docker host with IBM® Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.

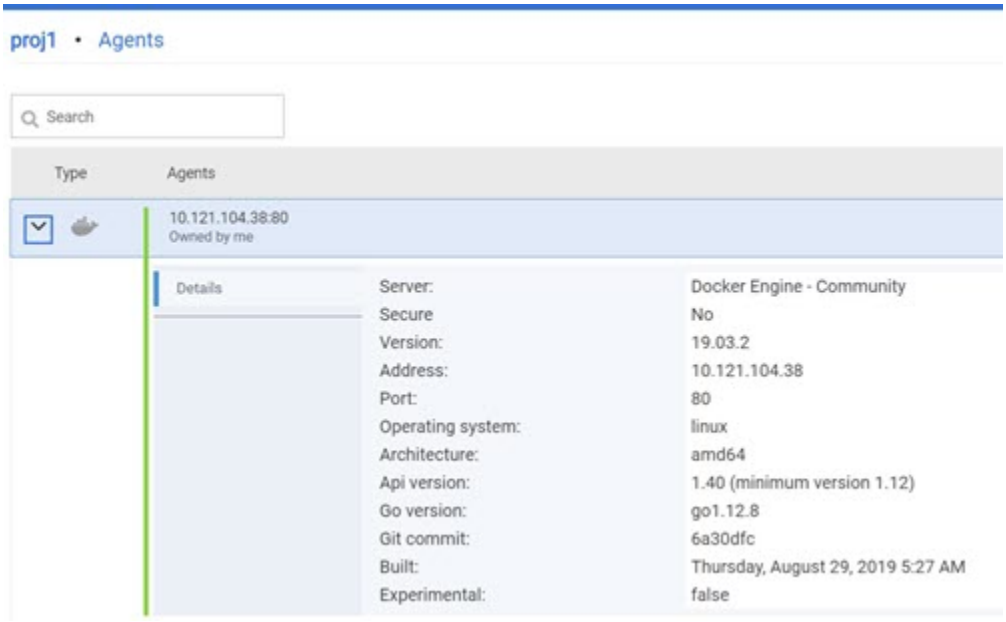
1. Log in to IBM® Rational® Test Automation Server and click **Infrastructure**.
2. Click **Add > Add docker**.

   The Docker hosts that are registered with IBM® Rational® Test Automation Server are displayed.
3. Select the Docker host that you want to add to the project from the list of Docker hosts, and then click **Add**.

   **Note:** You can add the Docker hosts that you have registered. You can add any number of Docker hosts to your project.

   The Docker hosts that you added to the project are displayed.

You can view the details of the Docker host by clicking the **Expand** icon >.




**What to do next**

You can select any of the remote Docker hosts as an alternate location to run the test asset in your project while configuring a test run from the **Execution** page.

---

Related reference

Task flow for working with remote Docker hosts

Related information

Tests configurations and test runs

## Editing configurations of a remote Docker host

You can edit the host name or the port of the registered Docker host computer instead of registering it as a new remote host if any of the parameters are changed on the remote host computer. You can also change the mode of authentication of a registered remote Docker host.

**Before you begin**

You must have completed the following tasks:

1. Set up the remote Docker host computer. See Setting up a remote Docker host computer on page 110 or Setting up a secure remote Docker host computer on page 111.
2. Registered the remote Docker host with IBM® Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.

1. Log in to Rational® Test Automation Server.
2. Click the **User** icon on the menu bar and select **Docker Hosts**.

   The **Manage docker hosts** dialog box displays the registered remote Docker hosts.
3. Modify the Docker host by clicking the **Edit docker host** icon.

   The **Update docker host** dialog box is displayed.
4. Enter the changed value of the host name or the port in **Remote hostname** text box.
5. Change the mode of authentication, if required:
   - To change from a non-secure mode to a secure mode, select the **Secure mode** option. Click the action labels to browse and select the `.pem` files that correspond to `ca_certificate.pem`, `client_certificate.pem`, and `client_key.pem`.

     **Note:** You must have generated the `.pem` files on the remote host computer and copied the files to your local drive.
   - To change from a secure mode to a non-secure mode, clear the **Secure mode** option.
6. Click **Test connection** to test whether a connection is established between Rational® Test Automation Server and the remote host computer:
   - On a successful connection, a message is displayed.
   - On a failure to connect to the remote host computer, an error message is displayed. You must resolve the error and reattempt to establish a successful connection.
7. Click **Update**.

   The remote Docker host with the updated details is displayed.

**Results**

You have updated the registered remote Docker host with the changed parameters.

**What to do next**

You can now add the updated Docker host to your project.

---

Related reference

Task flow for working with remote Docker hosts

Related information

Setting up a remote Docker host computer on page 110

Registering a remote Docker host with IBM Rational Test Automation Server on page 115

Adding a remote Docker host to the project for running tests on page 117

Deleting a remote Docker host from a project on page 120

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 121

## Deleting a remote Docker host from a project

You can delete a remote Docker host that you added to your project in IBM® Rational® Test Automation Server when you no longer need it to run tests.

**Before you begin**

You must have completed the following tasks:

1. Set up the remote Docker host computer. See Setting up a remote Docker host computer on page 110 or Setting up a secure remote Docker host computer on page 111.
2. Registered the remote Docker host with Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.
3. Added the registered Docker host to a project. See Adding a remote Docker host to the project for running tests on page 117.

1. Log in to Rational® Test Automation Server and click **Infrastructure** on the Navigation pane to display the **Infrastructure** page.

   The Docker hosts that you added to the project are displayed.

2. Delete the Docker host by clicking the **Remove docker host** icon .

   A message is displayed that the Docker host is deleted successfully.

**Results**

You have deleted the remote Docker host from your project. The remote Docker host is no longer available as a location while configuring a test run in your project.

**What to do next**

You might have to add a remote Docker host to your project to use the remote Docker host as a location to run tests.

---

Related reference

Task flow for working with remote Docker hosts

Related information

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 121

Editing configurations of a remote Docker host on page 118

## Unregistering a remote Docker host from IBM® Rational® Test Automation Server

You can unregister a remote Docker host that is registered with Rational® Test Automation Server when you no longer require it.

**Before you begin**

You must have completed the following tasks:

1. Set up the remote Docker host computer. See Setting up a remote Docker host computer on page 110 or Setting up a secure remote Docker host computer on page 111.
2. Registered the remote Docker host with Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 115.

1. Log in to Rational® Test Automation Server.
2. Click the **User** icon on the menu bar and select **Docker Hosts**.

   The **Manage docker hosts** dialog box displays the registered remote Docker hosts.
3. Unregister the Docker host by completing the following steps:
   a. Click the **Unregister docker host** icon.
   b. Click **Unregister** in the **Unregister docker host** dialog box.

   A message is displayed that the Docker host is unregistered successfully.

**Results**

You have unregistered the remote Docker host from Rational® Test Automation Server. You cannot run tests on this remote Docker host if it is added to your project. You might have to register it again with Rational® Test Automation Server, if you want to use the remote Docker host in your projects.

---

Related reference

Task flow for working with remote Docker hosts

Related information

Editing configurations of a remote Docker host on page 118

Adding a remote Docker host to the project for running tests on page 117

Deleting a remote Docker host from a project on page 120

# Test run configurations

You can configure and run tests in IBM® Rational® Test Automation Server after you add the test resources to your project.

Before you configure a test run, you must have completed the following tasks:

- Created a project on Rational® Test Automation Server or you must have been granted access to a project with the *Tester* role assigned. See Test assets and a server project on page 280.
- Created tests in the desktop clients and committed the test assets and test resources to a remote repository. You must have added the remote repository to the project.
- Read the considerations you must take into account for certain test types. See Prerequisites to running tests on page 98.

You must be a project *Owner* or a member with the *Tester* role assigned to configure and run a test.

You can find information about how to configure and run the following types of tests:

## Configuring a test for a quick run

You can configure any type of test to be run on IBM® Rational® Test Automation Server when you want to quickly ensure that the test runs correctly. You might not want to set the different options for the test nor want to schedule the run.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.
- You must have read and completed the tasks mentioned in Prerequisites to running tests on page 98 if they apply to the test you want to configure for a run.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify and select the test asset or resource that you want to run from the test assets listed.
4. Click the **Execute** icon in the row of the identified test asset.

   The **Execute test asset** dialog box is displayed.
5. Select the version of the test resources that you want to run, if you want to run a different version other than the latest version.
6. Select **Now** to initiate the test run immediately after you click **Execute**.
7. Enter a label for the test run that helps you to identify the test on the **Results** page.

   After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page.
8. Click **Execute**.
   The test run is started.

**Results**

You have started a test run.

**What to do next**

You can choose to perform any of the following tasks:

- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

---

Related information

Resetting the configuration settings for a test run on page 179

Viewing the progress of running test assets on page 176

Monitoring a test run

Stopping a test run on page 180

Canceling a scheduled test run on page 182

## Configuring an AFT Suite run

After you added the test resources that you created in the desktop client to the project, you can configure an AFT Suite to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.
- You must have read Test run considerations for AFT Suites on page 98 if you want to configure a run for an AFT Suite that has the agent location configured in the `AFT XML` file.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |

| Icon | Represents the test asset or resource |
|---|---|
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.
c. Create a filter query by using the **New filter** option by performing the following steps:
    i. Click **New filter**.
    ii. Add or enter the relevant parameters for the filter query.
    iii. Apply the filter query to filter the assets based on the query.
    iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

        1. Click **Save**.
        2. Enter a name for the filter query.
        3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:
    i. Click the **Open filters** icon .

    **Note:** To open the filter query, you must have created and saved a filter query.

    ii. Select the saved filter in the **Filters** dialog box.
    iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

   The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

**Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

   Use the following details about the version of the test resources that are displayed to identify the version that you want:
   - Commit message.
   - Tags labeled by the user for the version committed.
   - The user who committed the version to the repository.
   - Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

   The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

   The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**
- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:

   - Select **Now** to initiate the test run immediately after you click **Execute**.

     **Important:** Click **Execute** only after you have configured the other settings in this dialog box.

   - Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

The default time for scheduling a run is **Now**.

**Notes:**

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

   After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

   The default value for the **Label** field is null or an empty field.

   **Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

   If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

   If you are running an AFT Suite and the following conditions are true, then you must perform the next step:
   - You do not want to run the test on the agent configured in the `AFT XML` file.
   - You have not selected the agent from the **Override** column in the **Location** tab.

8. Click **Advanced** to make the following advanced configurations:

   a. Add the following setting in the **Program Arguments** field:

   ```
   -swaplocation <configured_agent_location>:<overriding_agent_location>
   ```

   For example, if the configured agent location is *91.2.352.24* and the remote agent where you want to run the test is *100.35.117.164*, then the entry in the **Program Arguments** field is as follows:

   ```
   -swaplocation 91.2.352.24:100.35.117.164
   ```

   If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |

| If... | Then... |
|---|---|
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

**Note:** The default report format is the *test log* format for the test reports.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you are running a test asset that contains datasets:
    a. Click the **DATA SOURCES** tab, if it is not already open.
    b. Consider the following information about datasets before you select a dataset:

        The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

        You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

        **Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

    c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

  **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.

  **Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

  **Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

    a. Click the **VARIABLES** tab, if it is not already open.

    b. Choose one of the following methods to add the variables:

    - To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.
    - To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

      **Note:** You must have created a file with the variables before you can select the file.

    The default value for the variables is null or an empty field.

    If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test that has static agents configured:

a. Click the **LOCATION** tab, if it is not already open.

The static agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the agent is displayed.

**Note:** You must have added agents to your project from the **Infrastructure** page for the agents to be displayed under the **Override** column.

b. Select the agent where you want to run the test asset.

You can select the same agent that is configured in the test asset. Alternatively, you can override the agent with any other agent added to the project by selecting it from the list in the **Override** column.

The default value for the agents is null or an empty field if no agents were configured in the test asset. If the test asset contains agents that are configured, then the default agent is the first item to be displayed on the list of agents listed in the increasing alphabetical order.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you want to change the location for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

**Important:** You must have added remote Docker hosts that are registered with Rational® Test Automation Server to your project from the **Infrastructure** page. The remote Docker hosts are then displayed under the **Override** column.

**Notes:**

- If remote Docker hosts are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Rational® Test Automation Server.
- If remote Docker hosts are added to your project, the added Docker hosts are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts are available in your project.
- Select the remote Docker host from the list when a remote Docker host is available in your project.
- Select **No override options**, if you selected any remote Docker host and want to revert to the **Default Cluster** to run the test asset.

If you want to run the test immediately or at the scheduled time, click **Execute**.

13. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of an AFT Suite.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can monitor the test from the **Progress** page. See Monitoring a test run.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

---

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

## Configuring an API Suite run

After you added the test resources that you created in the desktop client to the project, you can configure an API Suite to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.
- You must have completed the following tasks if you are running API Suites that use a transport and the transport requires third-party application `Jar` files for a successful run:

- Read Test run considerations for API Suites on page 99.
- Copied the third-party application `Jar` files to the third-party application folder on the computer where Rational® Test Automation Server is installed, if you are running the API Suite in Kubernetes. See Copying third-party application Jars to Kubernetes on page 74.
- Copied the third-party application `Jar` files to the third-party application folder on the computer where the remote Docker host is installed, if you are running the API Suite on a remote Docker host. See Copying third-party application Jars to a remote Docker host on page 113.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

   b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.
   c. Create a filter query by using the **New filter** option by performing the following steps:
      i. Click **New filter**.
      ii. Add or enter the relevant parameters for the filter query.

iii. Apply the filter query to filter the assets based on the query.
iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

1. Click **Save**.
2. Enter a name for the filter query.
3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:
i. Click the **Open filters** icon .

**Note:** To open the filter query, you must have created and saved a filter query.

ii. Select the saved filter in the **Filters** dialog box.
iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

**Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:
   - Select **Now** to initiate the test run immediately after you click **Execute**.

     **Important:** Click **Execute** only after you have configured the other settings in this dialog box.

   - Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

   The default time for scheduling a run is **Now**.

   **Notes:**

   - If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
   - If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

   After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

   The default value for the **Label** field is null or an empty field.

   **Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

   If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

8. Click **Advanced** to make the following advanced configurations:

a. Enter any Java arguments that must be passed to the test run at run time in the **Java Arguments** field, if applicable for the test.

   For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

   If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

   **Note:** The default report format is the *test log* format for the test reports.

c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

   For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if the API Suite has an environment or secrets configured:
   a. Click the **ENVIRONMENT** tab, if it is not already open.
   b. Select the API test environment from the list if there are multiple environments configured in the test asset.
   c. Select the secrets collection that contains the secrets to be used for the test run.

**Notes:**

- The test asset that was created in the desktop client and added to the Git repository must have the environments defined as part of the API test project.
- If the test asset contained secrets, then you must create those secrets in secrets collections in the project.

The default value for the environment is the environment configured in the test asset. The default value for secrets is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions if you are running a test asset that contains datasets:

    a. Click the **DATA SOURCES** tab, if it is not already open.

    b. Consider the following information about datasets before you select a dataset:

    The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

    You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

    **Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

    c. Select the dataset that you want to use in the test run from any of the following options:

    - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

      **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

    - Select the dataset from the list.

**Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

**Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

    a. Click the **VARIABLES** tab, if it is not already open.

    b. Choose one of the following methods to add the variables:

    - To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.
    - To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

    **Note:** You must have created a file with the variables before you can select the file.

    The default value for the variables is null or an empty field.

    If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you want to change the location for running the test:

    a. Click the **LOCATION** tab, if it is not already open.

    The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

    **Important:** You must have added remote Docker hosts that are registered with Rational® Test Automation Server to your project from the **Infrastructure** page. The remote Docker hosts are then displayed under the **Override** column.

    **Notes:**

- If remote Docker hosts are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Rational® Test Automation Server.
- If remote Docker hosts are added to your project, the added Docker hosts are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts are available in your project.
- Select the remote Docker host from the list when a remote Docker host is available in your project.
- Select **No override options**, if you selected any remote Docker host and want to revert to the **Default Cluster** to run the test asset.

If you want to run the test immediately or at the scheduled time, click **Execute**.

13. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of an API Suite.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

---

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

# Configuring a run of a Compound Test that contains traditional HTML tests

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains traditional HTML tests to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

   b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.
   c. Create a filter query by using the **New filter** option by performing the following steps:
      i. Click **New filter**.
      ii. Add or enter the relevant parameters for the filter query.

iii. Apply the filter query to filter the assets based on the query.
iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

1. Click **Save**.
2. Enter a name for the filter query.
3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:
i. Click the **Open filters** icon .

**Note:** To open the filter query, you must have created and saved a filter query.

ii. Select the saved filter in the **Filters** dialog box.
iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

**Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:
   - Select **Now** to initiate the test run immediately after you click **Execute**.

     **Important:** Click **Execute** only after you have configured the other settings in this dialog box.

   - Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

   The default time for scheduling a run is **Now**.

   **Notes:**

   - If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
   - If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

   After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

   The default value for the **Label** field is null or an empty field.

   **Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

   If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

8. Click **Advanced** to make the following advanced configurations:

a. Enter any Java arguments that must be passed to the test run at run time in the **Java Arguments** field, if applicable for the test.

   For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

   If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

   **Note:** The default report format is the *test log* format for the test reports.

c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

   For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you are running a test asset that contains datasets:

   a. Click the **DATA SOURCES** tab, if it is not already open.

   b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

**Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

  **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.

  **Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

  **Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions in this step if the test requires a variable that must be passed to the test at the test run time.

    You must configure the supported browser by using a variable if the test has a browser configured, which is different from the one that is supported by Rational® Test Automation Server.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.

  i. Enter `RTW_WebUI_Browser_Selection` as the name.
  ii. Enter `Firefox` as the value, if you want to use Firefox as the browser and overrides the browser specified in the HTML test.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

  **Note:** You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test that has static agents configured:

a. Click the **LOCATION** tab, if it is not already open.

The static agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the agent is displayed.

**Note:** You must have added agents to your project from the **Infrastructure** page for the agents to be displayed under the **Override** column.

b. Select the agent where you want to run the test asset.

You can select the same agent that is configured in the test asset. Alternatively, you can override the agent with any other agent added to the project by selecting it from the list in the **Override** column.

The default value for the agents is null or an empty field if no agents were configured in the test asset. If the test asset contains agents that are configured, then the default agent is the first item to be displayed on the list of agents listed in the increasing alphabetical order.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you want to change the location for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

**Important:** You must have added remote Docker hosts that are registered with Rational® Test Automation Server to your project from the **Infrastructure** page. The remote Docker hosts are then displayed under the **Override** column.

**Notes:**

- If remote Docker hosts are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Rational® Test Automation Server.
- If remote Docker hosts are added to your project, the added Docker hosts are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts are available in your project.
- Select the remote Docker host from the list when a remote Docker host is available in your project.
- Select **No override options**, if you selected any remote Docker host and want to revert to the **Default Cluster** to run the test asset.

If you want to run the test immediately or at the scheduled time, click **Execute**.

13. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of a Compound Test that contains traditional HTML tests.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.

- You can monitor the test from the **Progress** page. See Monitoring a test run.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

## Configuring a run of a Compound Test that contains Web UI tests

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains Web UI tests to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.
- You must have created the mobile tests in Rational® Functional Tester and added the test asset to the project repository on IBM® Rational® Test Automation Server.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |

| Icon | Represents the test asset or resource |
|---|---|
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.
c. Create a filter query by using the **New filter** option by performing the following steps:
    i. Click **New filter**.
    ii. Add or enter the relevant parameters for the filter query.
    iii. Apply the filter query to filter the assets based on the query.
    iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

        1. Click **Save**.
        2. Enter a name for the filter query.
        3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:
    i. Click the **Open filters** icon .

    **Note:** To open the filter query, you must have created and saved a filter query.

    ii. Select the saved filter in the **Filters** dialog box.
    iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

**Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.

- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:

- Select **Now** to initiate the test run immediately after you click **Execute**.

**Important:** Click **Execute** only after you have configured the other settings in this dialog box.

- Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

The default time for scheduling a run is **Now**.

**Notes:**

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

**Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

8. Click **Advanced** to make the following advanced configurations:

    a. Enter any Java arguments that must be passed to the test run at run time in the **Java Arguments** field, if applicable for the test.

    For example, you can set a maximum Java heap size.

    b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

    If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

**Note:** The default report format is the *test log* format for the test reports.

c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you are running a test asset that contains datasets:

a. Click the **DATA SOURCES** tab, if it is not already open.

b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

**Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

  **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.

**Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

**Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

    a. Click the **VARIABLES** tab, if it is not already open.

    b. Choose one of the following methods to add the variables:

    - To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.
    - To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

    **Note:** You must have created a file with the variables before you can select the file.

    The default value for the variables is null or an empty field.

    If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test that has static agents configured:

    a. Click the **LOCATION** tab, if it is not already open.

    The static agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the agent is displayed.

**Note:** You must have added agents to your project from the **Infrastructure** page for the agents to be displayed under the **Override** column.

b. Select the agent where you want to run the test asset.

You can select the same agent that is configured in the test asset. Alternatively, you can override the agent with any other agent added to the project by selecting it from the list in the **Override** column.

The default value for the agents is null or an empty field if no agents were configured in the test asset. If the test asset contains agents that are configured, then the default agent is the first item to be displayed on the list of agents listed in the increasing alphabetical order.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you are running a compound test that has static agents configured:

a. Click the **LOCATION** tab, if it is not already open.

The static agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the agent is displayed.

**Note:** You must have added agents to your project from the **Infrastructure** page for the agents to be displayed under the **Override** column.

b. Select the agent where you want to run the test asset.

You can select the same agent that is configured in the test asset. Alternatively, you can override the agent with any other agent that matches the capabilities specified in the `AFT XML` file. Agents that match the capabilities are filtered and displayed on the list in the **Override** column.

The default value for the agents is null or an empty field if no agents were configured in the test asset. If the test asset contains agents that are configured, then the default agent is the first item to be displayed on the list of agents listed in the increasing alphabetical order.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

13. Follow the instructions if you want to change the location for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

**Important:** You must have added remote Docker hosts that are registered with Rational® Test Automation Server to your project from the **Infrastructure** page. The remote Docker hosts are then displayed under the **Override** column.

**Notes:**

- If remote Docker hosts are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Rational® Test Automation Server.
- If remote Docker hosts are added to your project, the added Docker hosts are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts are available in your project.
- Select the remote Docker host from the list when a remote Docker host is available in your project.
- Select **No override options**, if you selected any remote Docker host and want to revert to the **Default Cluster** to run the test asset.

If you want to run the test immediately or at the scheduled time, click **Execute**.

14. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of a Compound Test that contains Web UI tests.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can monitor the test from the **Progress** page. See Monitoring a test run.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

## Configuring a run of a Compound Test that contains mobile tests

**Disclaimer:** This release contains access to run mobile tests on Rational® Test Automation Server as a Tech Preview. The Tech Preview is intended for you to view the capabilities offered by Rational® Test Automation Server V10.1.2, and to provide your feedback to the product team. You are permitted to use the information only for evaluation purposes and not for use in a production environment. IBM provides the information without obligation of support and "as is" without warranty of any kind.

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains mobile tests to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.
- You must have created the mobile tests in Rational® Functional Tester and added the test asset to the project repository on IBM® Rational® Test Automation Server.
- You must have created a variable file if you want to import the variables file. The file must contain the details of the Appium server that is connected to the remote agent or the device cloud to which the mobile device is connected.
- You must have read Considerations for using Jaeger traces in reports on page 104 when you configure a run and you want Jaeger to report the test results for the test.

**About this task**

You can run the mobile tests that are contained in a Compound Test after you create the tests in Rational® Functional Tester, and then add them to your project on IBM® Rational® Test Automation Server. You can then run the mobile tests on the mobile devices that are connected to any of the following agents or clouds:

- Remote agents on which the Appium server is installed.
- The BitBar Cloud.
- The Perfecto Mobile cloud.

You must provide the details of the server or the mobile cloud to which the mobile devices are connected as variables in Step 10 on page 159. You can either enter the variables or use the file in which you entered the variables. You must refer to the following tables for the variables that are required for a successful run:

**Table 8. Variables for the Appium server**

| Name of the Variable | Action for the Value field |
|---|---|
| Mobile_Device_Selection | Specify the name of the mobile device that is connected to the Appium server. |
| appium.server.host | Specify the host name or IP address of the Appium server. |
| appium.server.port | Specify the port on the Appium server that is configured to communicate with Rational® Test Automation Server. |

**Table 9. Variables for the BitBar cloud**

| Name of the Variable | Value |
|---|---|
| Mobile_Device_Selection | Specify the name of the mobile device that is connected to the BitBar Cloud. |
| bitbar.apikey | Specify the user token generated for your BitBar account to authenticate your connection with the BitBar Cloud. |
| bitbar.host | Specify the host name of the BitBar Cloud instance. |
| bitbar.project | Specify the name of the project that contains the recorded test. |
| bitbar.testrun | Specify a name for the test run that must be displayed in the BitBar dashboard for the test run. |

**Table 10. Variables for the Perfecto mobile cloud**

| Name of the Variable | Value |
|---|---|
| Mobile_Device_Selection | Specify the name of the mobile device that is connected to the Perfecto Mobile cloud. |

**Table 10. Variables for the Perfecto mobile cloud (continued)**

| Name of the Variable | Value |
|---|---|
| perfecto.securitytoken | Specify the user token generated for your Perfecto account to authenticate your connection with the Perfecto Mobile cloud. |
| perfecto.host | Specify the host name of the Perfecto Mobile cloud instance. |

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

   b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.

c. Create a filter query by using the **New filter** option by performing the following steps:
   i. Click **New filter**.
   ii. Add or enter the relevant parameters for the filter query.
   iii. Apply the filter query to filter the assets based on the query.
   iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

      1. Click **Save**.
      2. Enter a name for the filter query.
      3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:
   i. Click the **Open filters** icon .

      **Note:** To open the filter query, you must have created and saved a filter query.

   ii. Select the saved filter in the **Filters** dialog box.
   iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

   The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

   **Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

   a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

      Use the following details about the version of the test resources that are displayed to identify the version that you want:
      - Commit message.
      - Tags labeled by the user for the version committed.
      - The user who committed the version to the repository.
      - Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

      The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

   b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

      The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:

   - Select **Now** to initiate the test run immediately after you click **Execute**.

     **Important:** Click **Execute** only after you have configured the other settings in this dialog box.

   - Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

   The default time for scheduling a run is **Now**.

   **Notes:**

   - If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
   - If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

   After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

   The default value for the **Label** field is null or an empty field.

   **Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in

to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

8. Click **Advanced** to make the following advanced configurations:

   a. Enter any Java arguments that must be passed to the test run at run time in the **Java Arguments** field, if applicable for the test.

      For example, you can set a maximum Java heap size.

   b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

      If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

      **Note:** The default report format is the *test log* format for the test reports.

   c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

      For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you are running a test asset that contains datasets:

    a. Click the **DATA SOURCES** tab, if it is not already open.

    b. Consider the following information about datasets before you select a dataset:

    The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

    You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

    **Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

    c. Select the dataset that you want to use in the test run from any of the following options:

    - Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

      **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

    - Select the dataset from the list.

      **Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

    - Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

      **Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

    If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Perform the following steps to provide the variables that specify the server or cloud to which the mobile device is attached. You can either enter the variables that must be passed to the test at the test run time or import the file that contains the variables.

a. Click the **VARIABLES** tab, if it is not already open.

b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

**Note:** You must have created a file with the variables before you can select the file.

11. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of a Compound Test that contains mobile tests.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can monitor the test from the **Progress** page. See Monitoring a test run.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

---

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

## Configuring a run of a Compound Test that contains performance tests

After you added the test resources that you created in the desktop client to the project, you can configure a Compound Test that contains performance tests to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

   b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.
   c. Create a filter query by using the **New filter** option by performing the following steps:
      i. Click **New filter**.
      ii. Add or enter the relevant parameters for the filter query.
      iii. Apply the filter query to filter the assets based on the query.
      iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

1. Click **Save**.
2. Enter a name for the filter query.
3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:

i. Click the **Open filters** icon .

**Note:** To open the filter query, you must have created and saved a filter query.

ii. Select the saved filter in the **Filters** dialog box.

iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

**Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.
- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:
   - Select **Now** to initiate the test run immediately after you click **Execute**.

     **Important:** Click **Execute** only after you have configured the other settings in this dialog box.

   - Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

   The default time for scheduling a run is **Now**.

   **Notes:**

   - If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
   - If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

   After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

   The default value for the **Label** field is null or an empty field.

   **Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

   If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

8. Click **Advanced** to make the following advanced configurations:

a. Enter any Java arguments that must be passed to the test run at run time in the **Java Arguments** field, if applicable for the test.

   For example, you can set a maximum Java heap size.

b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

   If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

   **Note:** The default report format is the *test log* format for the test reports.

c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

   For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you are running a test asset that contains datasets:

   a. Click the **DATA SOURCES** tab, if it is not already open.

   b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

**Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

  **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.

  **Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

  **Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

    a. Click the **VARIABLES** tab, if it is not already open.

    b. Choose one of the following methods to add the variables:

- To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.
- To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

  **Note:** You must have created a file with the variables before you can select the file.

The default value for the variables is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test that has static agents configured:

    a. Click the **LOCATION** tab, if it is not already open.

    The static agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the agent is displayed.

    **Note:** You must have added agents to your project from the **Infrastructure** page for the agents to be displayed under the **Override** column.

    b. Select the agent where you want to run the test asset.

    You can select the same agent that is configured in the test asset. Alternatively, you can override the agent with any other agent added to the project by selecting it from the list in the **Override** column.

    The default value for the agents is null or an empty field if no agents were configured in the test asset. If the test asset contains agents that are configured, then the default agent is the first item to be displayed on the list of agents listed in the increasing alphabetical order.

    If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you want to change the location for running the test:

    a. Click the **LOCATION** tab, if it is not already open.

    The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

**Important:** You must have added remote Docker hosts that are registered with Rational® Test Automation Server to your project from the **Infrastructure** page. The remote Docker hosts are then displayed under the **Override** column.

**Notes:**

- If remote Docker hosts are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Rational® Test Automation Server.
- If remote Docker hosts are added to your project, the added Docker hosts are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts are available in your project.
- Select the remote Docker host from the list when a remote Docker host is available in your project.
- Select **No override options**, if you selected any remote Docker host and want to revert to the **Default Cluster** to run the test asset.

If you want to run the test immediately or at the scheduled time, click **Execute**.

13. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of a Compound Test that contains performance tests.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can monitor the test from the **Progress** page. See Monitoring a test run.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

## Configuring a run of a Rate Schedule or VU Schedule

After you added the test resources that you created in the desktop client to the project, you can configure a Rate Schedule or VU Schedule to be run on IBM® Rational® Test Automation Server.

**Before you begin**

- You must be a member of the project with the *Owner* or *Tester* role to configure and run tests.
- You must have read the following topics:
  - Test run considerations for schedules on page 103 if you want to configure a run for Rate Schedules or VU Schedules that have Resource Monitoring sources configured.
  - Considerations for using Jaeger traces in reports on page 104 if you want to configure a run and you want Jaeger to report the test results for the test.

1. Open the project that contains the test assets or resources that you added from the Git repository, and then click **Execution**.
2. Select the branch of the repository that contains the test assets or resources that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset or resource that you want to run by performing any of the following steps:

   a. Look for the test asset or resource by scrolling through the list.

   **Note:** You can also identify the type of the asset or resource by looking for the icon that represents the test type, as shown in the following table:

| Icon | Represents the test asset or resource |
|---|---|
| | AFT Suite |
| | API Suite |
| | Compound Test |
| | JMeter Test |
| | Postman resources |

| Icon | Represents the test asset or resource |
|---|---|
| | Rate Schedule |
| | Stubs |
| | VU Schedule |

b. Search for the test asset or resource by entering any text contained in the test asset or resource name in the **Search** text box.
c. Create a filter query by using the **New filter** option by performing the following steps:
    i. Click **New filter**.
    ii. Add or enter the relevant parameters for the filter query.
    iii. Apply the filter query to filter the assets based on the query.
    iv. Save the filter query, if you want to reuse the filter query later, by performing the following steps:

        1. Click **Save**.
        2. Enter a name for the filter query.
        3. Click **Save**.

d. Retrieve a saved filter, if you have saved filter queries earlier by performing the following steps:
    i. Click the **Open filters** icon .

    **Note:** To open the filter query, you must have created and saved a filter query.

    ii. Select the saved filter in the **Filters** dialog box.
    iii. Click **Apply** to apply the filter.

4. Click the **Execute** icon in the row of the identified test asset.

The **Execute test asset** dialog box is displayed.

5. Select the version of the test resources that you want to run by completing any of the following actions:

**Note:** The test resources in the version can contain the test assets, datasets, AFT XML files, API environment tags, and other resources specific to projects created from any of the desktop clients.

a. Expand the list in the **Version** field, find the version of the test resources, and then select the version.

Use the following details about the version of the test resources that are displayed to identify the version that you want:

- Commit message.
- Tags labeled by the user for the version committed.

- The user who committed the version to the repository.
- Relative time of the commit. For example, *2 hours ago* or *3 days ago*.

The list displays the versions of the test resources committed by all users to the branch in the repository. The versions are arranged with the latest version committed followed by the versions committed previously.

b. Expand the list in the **Version** field, and search for the version that you want to select by entering a partial or the complete commit message of that version.

The version that matches the search criteria is displayed and is selected for the test run.

The default value for the version selected for the run is the latest version in the selected branch in the repository. If you do not select any version, then the latest version is selected for the test run.

**Notes:**

- If you selected a version but you do not want to use that version in the test run, you can remove the selected version by clicking the ✖ icon, then the default version is selected for the test run.
- If you repeated a test or ran the test again from the **Results** page, then the version of the test resources that you had selected for the earlier run is shown as selected. You can either retain this version or select any other version from the list. You can also remove the previous version by clicking the ✖ icon.

6. Select the time for scheduling the test run from the following options:

- Select **Now** to initiate the test run immediately after you click **Execute**.

**Important:** Click **Execute** only after you have configured the other settings in this dialog box.

- Select **Later** and configure the date and time for scheduling a test to run at the scheduled date and time.

The default time for scheduling a run is **Now**.

**Notes:**

- If you have configured some or all of the settings for the current test run, and you do not want to continue with those settings, you can reset the settings by clicking **Reset**.
- If you are repeating a test run and do not want to use the saved settings from a previous run, you can reset all the saved settings to their default values by clicking **Reset**.

7. Enter a label for the test run that helps you to identify the test on the **Results** page.

After the test run completes, the label that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, any member of the project can use that label.

The default value for the **Label** field is null or an empty field.

**Important:** The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you run the same test again. Those changes are not visible when another user logs in to Rational® Test Automation Server. For example, if you created new variables on the server, those variables are available only for you when the same test is run again.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

8. Click **Advanced** to make the following advanced configurations:

    a. Enter any Java arguments that must be passed to the test run at run time in the **Java Arguments** field, if applicable for the test.

    For example, you can set a maximum Java heap size.

    b. Enter program arguments that must be passed to the test run at run time in the **Program Arguments**, if applicable for the test.

    If the test that you want to configure supports Jaeger tracing, then do the required tasks in the following scenarios:

| **If...** | **Then...** |
|---|---|
| You want to obtain the test results as a Jaeger trace. | Enter `-history jaeger` in the **Program Arguments** field. |
| You want to remove the Jaeger trace for your test, if Jaeger is already set as a program argument. | Delete the `-history jaeger` entry in the **Program Arguments** field. |
| You want the report as a test log and as a Jaeger trace. | Enter `-history jaeger,testlog` in the **Program Arguments** field. |

**Note:** The default report format is the *test log* format for the test reports.

c. Enter the environment variables that must be passed to the test run at run time in the **Environment Variables** field, if applicable for the test.

For example, enter the environment variables when the third-party libraries that are used in the test run refer to the environment variables for configuration.

**Note:** You must separate the arguments or variables with a white space when you enter them in the same line or start each argument or variable on a new line.

The default value for each of the fields for the advanced settings is null or an empty field.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

9. Follow the instructions if you are running a test asset that contains datasets:

a. Click the **DATA SOURCES** tab, if it is not already open.

b. Consider the following information about datasets before you select a dataset:

The default value for the datasets in the **DATA SOURCES** tab is null if the test asset did not have an associated dataset. If the asset had an associated dataset, the default value is the associated dataset.

You can utilize the dataset stored as an Excel or CSV file to override the original dataset associated with the Suite, test, or schedule. For example, you have associated a dataset in `.xlsx`, `.xls`, or `.csv` format with the test or schedule in desktop clients. If you have another set of data stored in an Excel or CSV file, you can select that dataset from the **Override** list.

**Remember:** You must have uploaded the dataset as an Excel or CSV file into the Git repository, and ensured that both the original dataset (from the test asset) and new datasets (added to the project) have the same column names.

c. Select the dataset that you want to use in the test run from any of the following options:

- Select the dataset that is displayed as the default dataset when the test asset contains a single dataset.

  **Note:** If there is only one dataset in the test asset, then that dataset is displayed as the default dataset.

- Select the dataset from the list.

**Note:** If there are multiple datasets in the test asset, the datasets are listed in their increasing alphabetical order.

- Select the dataset from the **Override** list to override the dataset that was associated with the test in the desktop client.

**Important:** If the test contains an encrypted dataset, the Project Owner must classify it in the **DATA SECURITY** tab on the Project page before you can select it. You must have added datasets to your project from the **Dataset** page for the datasets to be displayed in the **Override** list.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

10. Follow the instructions if the test requires a variable that must be passed to the test at the test run time.

    a. Click the **VARIABLES** tab, if it is not already open.

    b. Choose one of the following methods to add the variables:

    - To add new variables manually, click the **Add Variable** icon, enter the name, and value of the variable.
    - To add new variables from your local computer or from the Git repository that is associated with your server project, click the **Upload** icon and select the **Upload from local system** or **Browse from server** to select the variable file.

    **Note:** You must have created a file with the variables before you can select the file.

    The default value for the variables is null or an empty field.

    If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

11. Follow the instructions if you are running a test that has static agents configured:

    a. Click the **LOCATION** tab, if it is not already open.

    The static agents that are configured in the test asset are listed under the **Host** column. The information about the availability of the agent is displayed.

**Note:** You must have added agents to your project from the **Infrastructure** page for the agents to be displayed under the **Override** column.

b. Select the agent where you want to run the test asset.

You can select the same agent that is configured in the test asset. Alternatively, you can override the agent with any other agent added to the project by selecting it from the list in the **Override** column.

The default value for the agents is null or an empty field if no agents were configured in the test asset. If the test asset contains agents that are configured, then the default agent is the first item to be displayed on the list of agents listed in the increasing alphabetical order.

If you want to run the test immediately or at the scheduled time, click **Execute**, or continue with the next step.

12. Follow the instructions if you want to change the location for running the test:

a. Click the **LOCATION** tab, if it is not already open.

The **Default Cluster** is the default location where the test runs, and it is listed under the **Host** column. The information about the availability of the default location is displayed.

**Important:** You must have added remote Docker hosts that are registered with Rational® Test Automation Server to your project from the **Infrastructure** page. The remote Docker hosts are then displayed under the **Override** column.

**Notes:**

- If remote Docker hosts are not added to your project, the option **No override options** is displayed as the default value and the test runs in the Kubernetes cluster of Rational® Test Automation Server.
- If remote Docker hosts are added to your project, the added Docker hosts are displayed along with their availability status and ownership information.

b. Select the location where you want to run the test asset from the following options:

- Select the **Default Cluster** when no remote Docker hosts are available in your project.
- Select the remote Docker host from the list when a remote Docker host is available in your project.
- Select **No override options**, if you selected any remote Docker host and want to revert to the **Default Cluster** to run the test asset.

If you want to run the test immediately or at the scheduled time, click **Execute**.

13. Follow the instructions if you want to override the Resource Monitoring labels in the test asset with the Resource Monitoring labels that you created in Rational® Test Automation Server:

a. Click the **Resource Monitoring** tab.

**Note:**

The **Resource Monitoring** tab displays only if you enabled the **Resource Monitoring from Service** option in Rational® Performance Tester when you created the test asset.

b. Click , and press the `Ctrl` + `Space bar` keys, or enter the initial letter of a label to select a label in the list.

**Note:**

You can select or add labels only if you added the labels in the **Resource Monitoring Sources** page in Rational® Test Automation Server.

14. Click **Execute**.
    The test run is started.

**Results**

You have configured and either started or scheduled a test run of a Rate Schedule or VU Schedule.

**What to do next**

You can choose to perform any of the following tasks after you have initiated or scheduled a run:

- You can stop the test run at any point after the test run is initiated, from the **Execution** page. See Stopping a test run on page 180.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 182.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 176.
- You can monitor the test from the **Progress** page. See Monitoring a test run.
- You can view the results, reports, and logs of the test from the **Results** page after the test completes the run. See Test results on page 200.

---

Related information

Resetting the configuration settings for a test run on page 179

Tests configurations and test runs

Test run configurations on page 121

# Management of running tests

Find information about the tasks that you can perform on a test that you configured for a run either while it runs or after it completes the run.

## Viewing the progress of running test assets

After you initiate or schedule a test run on the Execution page, you can view the progress of the test assets that are running or scheduled to run, from the Progress page. You can also view the test assets that have completed their run, are stopped or canceled, during the past hour.

**Before you begin**

You must have initiated runs of the test assets in your project from the **Execution** page.

1. Open your project and click **Progress**.

   Any one of the following views is displayed:
   - No test assets are displayed in the Progress page if there are no test assets that are running, scheduled to run, or have completed their run in the past hour.
   - Test assets in your project that are in any of the following states are listed:
     - Running
     - Scheduled
     - Completed
     - Stopped
     - Canceled
   - No test assets are displayed but the options **Hide inactive** and **Show all** are present. Click **Show all** to display all the test assets on the Progress page. Test assets that were hidden from the display are displayed.

   **Note:** Test assets are automatically removed from the display after *60 minutes* of being added to the Progress page. Therefore, you can only view the test assets that were added to the Progress page in the past hour. You cannot use the Progress page to view the history of test assets that were run from the Execution page.

2. Identify the test asset for which you want to view the progress from the test assets listed. You can also identify the test asset by completing any of the following actions:
   - Search for the test asset by entering any text contained in the test asset name in the **Search** text box.
   - Sort the test assets by the user who started the test by clicking the **Started by** column header, and find the test asset listed against your name.
   - Sort the test assets by the Type of the test assets by clicking the **Started by** column header, and find the test asset type you ran or scheduled.
   - Click **Hide inactive** to only display the test assets that are running or scheduled to run. This action removes all completed, canceled, stopped, or failed test assets from the display.

You can view the following details of your test asset on the Progress page:

| Column header | Description |
|---|---|
| Type | Displays the icon for the type of the test asset. |
| Name | Displays the name of the test asset added to the project from the Git repository. |
| Started by | Displays the name of the user who started the run of the test asset. |
| Start Time | Displays the start time of the test asset run. |
| End Time | Displays the end time of the test asset run. |
| Status | Displays the state of the test asset progressively.<br>For example, after the run is in the `Initiated` state, it moves on to the `Running` state and ends in its final state depending on the verdict of the test run as either `Completed`, `Failed`, or `Inconclusive`.<br>If the test run is stopped or a scheduled run is canceled, the status displayed is `Stopped by User` for a stopped run or `Canceled` for a canceled test. |

From the **Actions** column, you can perform the following operations on your test asset:

- Stop a run.
- Cancel a scheduled run.
- Monitor a VU Schedule, Rate Schedule, Compound Test, or an AFT Suite.
- View the execution log.
- View the result of the test asset on the Results page.

**Results**

You have viewed the progress of the test assets that you ran from the Execution page.

**What to do next**

You can choose to perform any of the following tasks from the Progress page:

- You can stop a test asset that is running.
- You can cancel a scheduled run.
- You can monitor tests (such as the Schedules, Compound Test, and AFT Suites).

- You can see the results or the execution log of the test asset from the **Actions** column by clicking the **Open action menu** icon .
- You can view the results, reports, and logs of the completed test run from the Results page.

---

Related information

Test run configurations on page 121

Monitoring a test run

Checking logs on page 178

Canceling a scheduled test run on page 182

Stopping a test run on page 180

## Checking logs

To verify how the test ran or to debug test run failure, you can check the `Test Log` and `Execution log`.

**About this task**

The `Test Log` displays the interaction between Rational® Test Automation Server and the application or system under test. After the test run completes, the verdict of the run can be *Pass*, *Fail*, or *Inconclusive*. If the verdict of the run is Pass, the `Test Log` is available in the **Results** page.

The `Execution log` displays the console messages of the run time process that runs the test. This log is useful in determining the cause of the failure if the verdict of the run is *Fail* or *Inconclusive*. You can view the `Execution log` from the **Progress** page.

Viewing the `Test Log`

1. Go to the **Results** page and identify the test that you ran.
2. Click the test so that the **Reports** panel is displayed.
3. Click the **Test Log** in the **Reports** panel.

   The `Test Log` is displayed in a browser window.

Viewing the `Execution` log

4. Go to the **Progress** page and identify the test that is in the *Running* state.
5. Click the **Open action menu** icon.
6. Click **Execution log**.

   The `Execution log` is displayed in a browser window.

---

Related information

Test run configurations on page 121

Monitoring a test run

## Resetting the configuration settings for a test run

When you are configuring a test run either as a first-run or when repeating the run, and you do not want to proceed with the settings configured or saved for the test run, you can reset the configuration settings. Resetting the configuration reverts all the settings to their default values for the test run.

**Before you begin**

You must be a member of the project with the *Owner* or *Tester* role to run the tests.

You must ensure that you have the **Execute test asset** dialog box is displayed before you proceed with resetting the configuration settings for the test run.

**About this task**

You can reset the configuration settings when you are configuring a test run either for its first run or when you are repeating the test run.

1. Click **Reset** in the **Execute test asset** dialog box at any point when you want to reset the configuration settings for the test run.

   **Notes:**

   - If you have configured some or all of the settings for the current test run and you do not want to continue with those settings, clicking **Reset** resets the settings to their default values.
   - If you are repeating a test run and do not want to use the saved settings from a previous run, clicking **Reset** reverts all the saved settings to their default values.

   The settings in the **Execute test asset** dialog box are reset to their default values.
2. Use the following table to find the default values for each of the settings in the **Execute test asset** dialog box:

| Window or Tab | Field or Setting | Default value is |
|---|---|---|
| **Execute test asset** | Scheduling the test run | **Now** is selected. |
| **Execute test asset** | Label for settings | Null or empty field. |
| **Advanced** | **JVM Arguments** | Null or empty field. |
| | **Program Arguments** | Null or empty field. |
| | **Environment Variables** | Null or empty field. |
| **ENVIRONMENT** tab | API test environment | The environment configured in the test asset. |
| | Secrets collection | Null or empty field. |
| **DATA SOURCES** tab | **Override** list | Null or empty field if the asset has no dataset. |
| | | The dataset in the test asset if the asset has one dataset. |
| | | The first dataset on the list of datasets, which are listed in the increasing alphabetical order, if the asset has multiple datasets. |
| **VARIABLES** tab | | Null or empty field. |
| **LOCATION** tab | Agents | Null or empty field if the asset has no agents configured. |
| | | The first agent on the list of agents, which are listed in the increasing alphabetical order, if the asset has multiple agents configured. |
| | Docker host | Internal Docker host if no remote Docker hosts are added. |
| | | **No override options** if other Docker hosts are saved in previous runs. |

**What to do next**

You can complete configuring the settings that you want for the test run. See Test run configurations on page 121.

## Stopping a test run

You might want to stop a test run when you realized that you did not configure all the settings or you want to change a few settings for the test run.

**Before you begin**

You must have initiated a test run from the **Execution** page.

**About this task**

You can stop a running test from the Progress page by using the **Stop execution** icon. You cannot stop the test run if the test has already completed its run.

1. Go to the **Progress** page and identify the test that you want to stop.

   **Note:** You can stop a running test if it is in the `In Transition` or `Running` state.

2. Click the **Stop execution** icon in the **Actions** column of the selected test.
   **Result**

   The **Stop execution** dialog box is displayed.

3. Set the timeout period for stopping the test run. Enter a numeric value and select the unit from the options available as *Seconds*, *Minutes*, or *Hours*.

   The time out period is the time during which the test run is allowed to stop on its own and after the timeout period, the test is forced to stop abruptly.

   **Important:** If the timeout period is not set, the default of *30 seconds* is considered as the default timeout period.

4. Keep the option **Capture results** selected, if you want the results to be captured for the test. The captured results for the test are available to view from the Results page. Clear this option if you do not want to capture the results.

   **Note:** The option **Capture results** is selected by default.

   **Important:** The results are always captured for API suites even if you clear the **Capture results** option.

5. Keep the option **Execute 'finally-block', if present** selected, if you want the *finally-block* code in the test script, if present, to be run before the test run is stopped. Clear this option if you do not want to run the *finally-block* code in the test script.

   **Note:** The option **Execute 'finally-block', if present** is selected by default.

   **Restriction:** The *tear-down* steps equivalent to the **'finally-block'** code, if present in API suites are not controllable at runtime. The **Execute 'finally-block', if present** option has no impact on API suites.

6. Click **Stop execution**.

A notification message is displayed that the test run is stopped successfully.

**Note:** If the test run completes before you can configure the options for stopping the test run, a notification is displayed in the **Stop execution** dialog box that the test asset has completed its run.

**Results**

You have successfully stopped a running test. The stopped test runs are displayed on the Progress page with the status as *Stopped by User*. Stopped tests are not considered in the count of tests executed that are displayed on the Overview page.

**What to do next**

You can re-initiate the test run from the Execution page by completing the configurations that you want for the test.

You can see the results or the execution log of the stopped test from the **Actions** column on the Progress page by clicking the **Open action menu** icon.

Related information

Test run configurations on page 121

Viewing the progress of running test assets on page 176

Canceling a scheduled test run on page 182

## Canceling a scheduled test run

You might want to cancel a scheduled test run when you realized that you did not configure all the settings for a test, want to change a few settings for the test, or do not want that test to run.

**Before you begin**

You must have scheduled a test run from the **Execution** page.

**About this task**

You can cancel a scheduled test by using the **Cancel** icon available on the **Progress** page.

1. Go to the **Progress** page and identify the scheduled test that you want to cancel.
2. Click the **Cancel** icon in the **Actions** column of the selected test.
   **Result**

   The **Cancel scheduled execution** dialog box is displayed.
3. Click **Yes**.

**Result**

A notification is displayed that the scheduled test run is canceled.

**Results**

You have successfully canceled a scheduled test. The canceled test runs are displayed on the **Progress** page with the status as *Canceled*. The canceled test runs are not displayed on the **Results** page and are not considered in the count of tests executed that are displayed on the **Overview** page.

**What to do next**

You can re-initiate the test run from the **Execution** page by completing the configurations that you want for the test.

---

Related information

Test run configurations on page 121

Viewing the progress of running test assets on page 176

Stopping a test run on page 180

# Management of virtualized services

You can find information about the tasks that you can perform on and manage the virtualized services that run on Rational® Test Automation Server. You can start or stop the stubs connected to Rational® Test Automation Server. You can view the stub usage statistics and the routing rules configured for the stubs, agents, or proxies that are connected to Rational® Test Automation Server.

**Disclaimer:** This release contains access to the virtualized services feature in Rational® Test Automation Server as a Tech Preview. The Tech Preview is intended for you to view the capabilities of virtualized services offered by Rational® Test Automation Server V10.1.2, and to provide your feedback to the product team. You are permitted to use the information only for evaluation purposes and not for use in a production environment. IBM provides the information without obligation of support and "as is" without warranty of any kind.

You must create stubs in Rational® Integration Tester for the following types of services, and then add the stubs to your project on Rational® Test Automation Server before you can run the stubs on Rational® Test Automation Server:

- Stubs that utilize the WebSphere® MQ transport.
- Stubs that utilize the HTTP transport.
- Stubs that virtualize the Istio services.

  You can run stubs for the following types of requests received or sent by the Istio service mesh:

- Requests received by services in the Istio service mesh.
- Requests sent from namespaces in the Istio service mesh to external services that are not in the Istio service mesh.

**Important:** You can run stubs only in the **Default Cluster** location of Rational® Test Automation Server. You cannot run stubs on a remote Docker host.

After you run the stubs, you can view details of the stub usage from the **Stub Usage** page on Rational® Test Automation Server.

When stubs are created to run on agents or proxies, or have routing rules defined, and you run such stubs on Rational® Test Automation Server, you can view the agents, proxies or the routing rules from the **Routing Rules** page on the Rational® Test Automation Server User Interface.

Find information about the following tasks that you can perform on Rational® Test Automation Server:

## Prerequisites for running stubs

You can find information about the prerequisite tasks that you must complete before you can run stubs on Rational® Test Automation Server.

## Prerequisites for running HTTP stubs

You can find information about the tasks that you must complete before you run HTTP stubs on Rational® Test Automation Server.

### Conditions for running of HTTP stubs

Before you run the HTTP stubs from Rational® Test Automation Server, you must check for the following conditions and perform the actions indicated:

| If... | Then... |
|---|---|
| You want to route the HTTP traffic via the HTTP proxy either to the stubs or live systems, based on routing rules. | You must set up the HTTP proxy. See Setting up the HTTP proxy on page 187. |
| You are working with Rational® Test Automation Server V10.1.2 installed on Ubuntu. | HTTP stubs are exposed via host names as the default action and are of the following form:<br>`in-<unique_id>.<INGRESS_DOMAIN>`<br>**Note:** The DNS used by a computer that runs a client application or the HTTP proxy that routes |

| If... | Then... |
| --- | --- |
| | traffic to the stubs can resolve such host names to the IP address of the ingress domain.<br><br>You must perform the following actions that depend on the way you have set up the ingress domain:<br><br>• If you used **nip.io**, to form the ingress domain, then that stub host names are automatically resolved to the IP address of the ingress domain.<br><br>For example, if you used **nip.io** to form the ingress domain as follows:<br>`10.1.2.3.nip.io`<br>Then the stub host names are resolved by **nip.io** to the IP address as **10.1.2.3**.<br>• If you did not use **nip.io** to form the ingress domain, then you must ensure that the DNS in use by the client application or the HTTP proxy resolves the ingress domain to the IP address of the ingress domain.<br><br>For example, if the ingress domain is of the following form:<br>`*.<INGRESS_DOMAIN>`<br>Then the host name must resolve to the **<INGRESS_DOMAIN_IP_address>**. |

Related information

Setting up the HTTP proxy on page 187

Working with HTTP stubs without using proxies on page 191

## Prerequisites for running Istio stubs that virtualize services in Rational® Test Automation Server

You can find information about the tasks that you must complete before you run Istio stubs that virtualize services in Rational® Test Automation Server.

## Configuring Istio stubs that virtualize services in Rational® Test Automation Server

When you intend to virtualize services in the Istio mesh in Rational® Test Automation Server, you can create stubs in Rational® Integration Tester.

When you create the Istio stubs in Rational® Integration Tester, you must ensure that you are aware of the fully qualified domain name (FQDN) or the short name of the service that you want to virtualize in the Kubernetes cluster that hosts Rational® Test Automation Server.

The FQDN of the service is as follows:

```
<service-name>.<namespace>.<svc>.<cluster-name>
```

The short name of the service is as follows:

```
<service-name>
```

For example, if you want to virtualize the HTTP traffic in the service named as *reviews* that runs in a namespace named as *bookinfo*, and in the resource named as *svc* in the Kubernetes cluster named as *mycluster*, then the FQDN of the service is as follows:

```
reviews.bookinfo.svc.mycluster
```

The short name of the service is as follows:

```
reviews
```

You must enter the host of the service that you want to virtualize, in the **Host** field of the physical transport in Rational® Integration Tester. You can provide either the FQDN or the short name of the service.

**Note:** The pass-through action in the stubs is not supported when the short name of the service is entered in the **Host** field.

## Enabling the running of stubs that virtualize Istio services

You must have enabled service virtualization via Istio at the time of installing the server software. See the related links.

Before you run the stubs that virtualize services in namespaces contained in the Rational® Test Automation Server Kubernetes cluster, you must enable Rational® Test Automation Server to determine the namespace that contains the service that you want to virtualize.

Rational® Test Automation Server determines the namespace based on the following configurations of the **Host** of the physical transport:

- If you configured the FQDN in the host, Rational® Test Automation Server uses the namespace that is specified in the FQDN.
- If you configured the short name in the host, Rational® Test Automation Server uses the namespace from among the virtualization namespaces that were provided at the time of installation of Rational® Test Automation Server.

You must specify the namespaces that contain the services that you want to virtualize at the time of the installation of Rational® Test Automation Server.

---

Related reference

Installation of the server software on page 25

## Setting up the HTTP proxy

When you want to run stubs on Rational® Test Automation Server, you can set up an HTTP proxy to route requests to the stubs. You must configure the HTTP proxy with the Rational® Test Automation Server URL and offline user token generated from Rational® Test Automation Server so that the HTTP proxy can register with Rational® Test Automation Server.

**Before you begin**

Before you set up the HTTP proxy, you must ensure that you have installed the proxy component from Rational® Test Control Panel V10.1.1 or later.

**About this task**

When you create stubs in Rational® Integration Tester, you must configure the HTTP proxy in the **Client** tab for the HTTP transport. You must also edit the `registration.xml` file that is located in the `<RTCP_install_directory>\httptcp` directory for the following attributes:

- **server base-url**
- **security-token**

You must enter the Rational® Test Automation Server URL and offline user token as the values for the attributes. After you restart the proxy, the HTTP proxy registers with Rational® Test Automation Server. When you start the stubs from the **Execution** page, routing rules are applied to the HTTP proxy such that the requests that satisfy the rules are routed to the stubs.

1. Perform the following steps in Rational® Test Automation Server:
    a. Log in to Rational® Test Automation Server.
    b. Generate the offline user token.
    c. Copy the offline user token to a text file.
2. Perform the following steps to edit the `registration.xml` file that is located in the `<RTCP_install_directory>\httptcp` directory:

a. Use a text editor to open the `registration.xml` file.

b. Enter the Rational® Test Automation Server URL and offline user token generated from Rational® Test Automation Server in the following setting:

```
<server base-url="<Rational® Test Automation Server_URL>" security-token="<Offline_User_Token>" />
```

c. Save and close the file.

d. Restart the HTTP proxy.

3. Perform the following steps in Rational® Test Automation Server:
   a. Log in to Rational® Test Automation Server.
   b. Open your project.
   c. Click **Infrastructure**.

The HTTP proxies that you configured with your offline user token are displayed.

**Results**

You have successfully configured the HTTP proxy.

**What to do next**

You can check for the prerequisite tasks that you must complete before you can run the HTTP stubs. See Prerequisites for running HTTP stubs on page 184.

---

Related information

Running an API Suite that references local stubs by using proxies

## Starting a stub

When you have stubs in the API test assets, you can start the stubs contained in the API test assets from the **Execution** page in IBM® Rational® Test Automation Server.

**Before you begin**

You must have completed the following tasks:

- Been assigned the *Tester* or *Owner* role to run the tests.
- Created stubs that use a transport for tests in Rational® Integration Tester and committed the test resources to the remote repository.
- Selected the branch of the Git repository to view test resources on the **Execution** page.

- Enabled service virtualization through Istio at the time of installation of the server software, if you want to run the Istio stubs. See Prerequisites for running Istio stubs that virtualize services in Rational Test Automation Server on page 185.
- Completed the prerequisite tasks before you start HTTP stubs. See Prerequisites for running HTTP stubs on page 184.

**About this task**

You can start the following types of stubs contained in API Suites from the **Execution** page on Rational® Test Automation Server:

- Stubs that utilize the WebSphere® MQ transport.
- Stubs that utilize the HTTP transport.
- Stubs that virtualize the Istio services.

  You can run stubs for the following types of requests received or sent by the Istio service mesh:
  - Requests received by services in the Istio service mesh.
  - Requests sent from namespaces in the Istio service mesh to external services that are not in the Istio service mesh.

If you want to start the stub to run in a new container, you can specify the following JVM argument in the **Advanced** settings when you configure the run for the stub: **-Dexecution.stub.dedicated.container=true**.

1. Open the project that contains the stubs and click **Execution**.
2. Select the branch of the repository that contains the stubs that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the stubs that you want to run from the assets listed.

   **Tip:** You can identify the stubs by the **Stub** icon for the test asset under the **Type** column. Stubs are displayed with the extension `.stb` in their names under the **Name** column.

   You can also identify the stubs by completing any of the following steps:
   - Search for the stubs by entering any text contained in the test asset name in the **Search** field box.
   - Create a filter query by using the **New filter** option and complete the following steps:
     a. Enter the relevant parameters and apply the filter query.
     b. Save the filter query for retrieving it from the saved filters list.

        For example, you can create a rule, which has the **Type** attribute equal to *API Stub*, to display the stubs in the assets in the selected branch.
   - Retrieve a saved filter by using the **Open filters** icon by completing the following steps:

a. Select the saved filter.
b. Apply the filter.

**Note:** To open the filter query, you must have created and saved a filter query.

4. Click the **Execute** icon in the row of the identified stub.

   The **Execute test asset** dialog box is displayed.

5. Select the version of the stub that you want to start.
6. Select the environment that was used to bind the physical and logical resource in the API project, in the **ENVIRONMENT** tab.

   **Important:**
   - The results are not captured for the stubs started or run on Rational® Test Automation Server and the stubs are not displayed on the **Results** page.
   - The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you start stubs again. Those changes are not visible when another user logs in to Rational® Test Automation Server.

     For example, if you selected an environment, the same environment is selected when you start stubs again.

7. Click **Execute**.

**Results**

You have started a stub from the **Execution** page on Rational® Test Automation Server.

**What to do next**

You can perform any of the following tasks from the **Progress** page:

- You can stop the running stub at any point after you start the stub.
- You can view the progress of the stub.
- You can view the `Execution log` for the stub.

You can verify whether the stub is running correctly by running the test in the API test asset that utilizes the stub and checking the `Execution log` to see whether the test has passed on the stub.

You can view the routing rules of the stubs. See Viewing routing rules on page 193.

When you are done testing the stub, you must stop the running stub, see Stopping a stub on page 199.

## Working with HTTP stubs without using proxies

When you run the HTTP stubs on Rational® Test Automation Server, the endpoint on which the stub is exposed externally uses a host name that includes a generated ID. If a client application is required to call the stub directly via its external host name and not via the HTTP proxy, then you might want to determine the host name to configure the client first before you start the stub.

**Before you begin**

You must have completed the following tasks:

- Been assigned the *Tester* or *Owner* role to run the tests.
- Created a project on Rational® Test Automation Server and added the repository that contains the HTTP stubs.
- Completed the prerequisite tasks before you start HTTP stubs. See Prerequisites for running HTTP stubs on page 184.
- Selected the branch of the Git repository to view test resources on the **Execution** page.

**About this task**

When you have installed Rational® Test Automation Server on Ubuntu, the default HTTP stub endpoints are of the following form:

```
in-<unique_id>.<ingress_domain>
```

For example, with an ingress domain as **10.1.2.3.nip.io**, a stub endpoint might be as follows:

```
in-cd3a755635574c6fa7e264911301821a.10.1.2.3.nip.io
```

When you have installed Rational® Test Automation Server on OpenShift, the default HTTP stub endpoints are of the following form:

```
<ingress_domain_first_label>-<unique_id>.<remainder_of_ingress_domain>
```

For example, with an ingress domain as **rtas.mycluster-123456-0000.region.containers.appdomain.cloud**, a stub endpoint might be as follows:

```
rtas-cd3a755635574c6fa7e264911301821a.mycluster-123456-0000.region.containers.appdomain.cloud
```

The <unique_id> is generated when the stub is started. The full host name can be viewed in the **Routing to** field of the routing rule created for the stub on the **Routing Rules** page. When traffic is routed to the stubs via the HTTP proxy, a client application is unaware of this host name. If the client application is required to call the stubs directly, it must be aware of the stub host name. To allow a host name to be known before you start the stub, you can replace the generated ID with a value that you specify.

1. Open the project that contains the stubs and click **Execution**.
2. Select the branch of the repository that contains the stubs that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the stubs that you want to run from the assets listed.

**Tip:** You can identify the stubs by the **Stub** icon for the test asset under the **Type** column. Stubs are displayed with the extension `.stb` in their names under the **Name** column.

You can also identify the stubs by completing any of the following steps:

- Search for the stubs by entering any text contained in the test asset name in the **Search** field box.
- Create a filter query by using the **New filter** option and complete the following steps:
  a. Enter the relevant parameters and apply the filter query.
  b. Save the filter query for retrieving it from the saved filters list.

  For example, you can create a rule, which has the **Type** attribute equal to *API Stub*, to display the stubs in the assets in the selected branch.
- Retrieve a saved filter by using the **Open filters** icon by completing the following steps:
  a. Select the saved filter.
  b. Apply the filter.

**Note:** To open the filter query, you must have created and saved a filter query.

4. Click the **Execute** icon in the row of the identified stub.

   The **Execute test asset** dialog box is displayed.
5. Select the version of the stub that you want to start.
6. Select the environment that was used to bind the physical and logical resource in the API project, in the **ENVIRONMENT** tab.

**Important:**

- The results are not captured for the stubs started or run on Rational® Test Automation Server and the stubs are not displayed on the **Results** page.
- The configuration that you set for the test run in the **Execute test asset** dialog box is preserved when you start stubs again. Those changes are not visible when another user logs in to Rational® Test Automation Server.

  For example, if you selected an environment, the same environment is selected when you start stubs again.

7. Click **Advanced**, and then enter the following JVM argument in the **JVM Arguments** field:

```
-Dexecution.ingress.host.identifier=<custom_id>
```

**Notes:**

- The value of the **<custom_id>** that you specify is used as the **<unique_id>** for the stub.
- The **<unique_id>** must start and end with an alphanumeric character. The characters supported include: a-z, A-X, 0-9, and a hyphen. No other characters can be used in the **<unique_id>**.
- The **<unique_id>** must be unique to a stub that is running.

  **Restriction:** You must not assign the same **<unique_id>** to another stub that is running simultaneously. Doing so might result in an undefined behavior of the stub.

For example, on Ubuntu, if you specify the value of the *<custom_id>* for the stub as `stub123-test` and Rational® Test Automation Server has an ingress domain as **10.1.2.3.nip.io**. Then the host name of the stub endpoint is as follows:

```
in-stub123-test.10.1.2.3.nip.io
```

8. Click **Execute**.

**Results**

You have started an HTTP stub. The stub exposes the host name that contains a custom ID that you specified.

**What to do next**

You can perform any of the following tasks from the **Progress** page:

- You can stop the running stub at any point after you start the stub.
- You can view the progress of the stub.
- You can view the `Execution log` for the stub.

You can view the routing rules of the stubs. See Viewing routing rules on page 193.

## Viewing routing rules

When stubs are created to run on agents or proxies, or have routing rules defined, and you run such stubs on Rational® Test Automation Server, you can view the agents, proxies, or the routing rules from the **Routing Rules** page.

**Before you begin**

You must have completed the following tasks:

- Created stubs in Rational® Integration Tester and committed the test resources to the remote repository.
- Started the stubs on Rational® Test Automation Server. See Starting a stub on page 188.

**About this task**

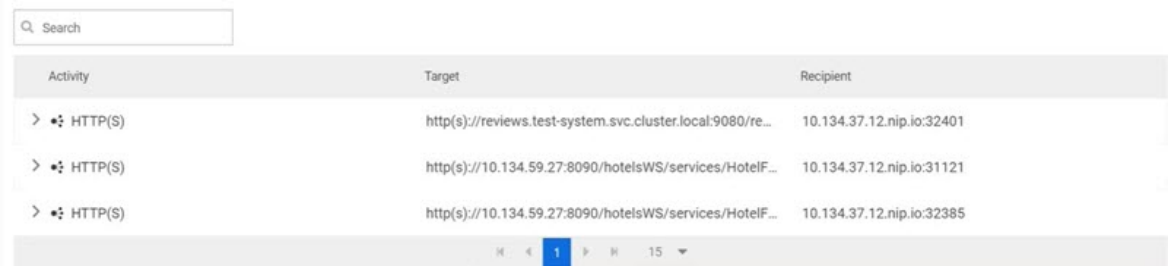
After you run the stubs that are connected to Rational® Test Automation Server, you can view details of the stubs that ran and the routing rules for the stubs, from the **Routing Rules** page.

1. Open your project on Rational® Test Automation Server.

   The **Overview** page is displayed as the default page after you created the project and logged in subsequently.

2. Click **Routing Rules**.

   The agents or proxies connected to Rational® Test Automation Server are displayed.

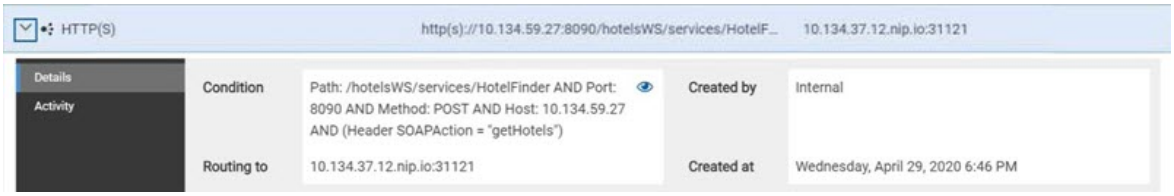



   You can find the following details about the agents or proxies:

| Column | Description |
|---|---|
| Activity | Lists the agent or proxy. |
| Target | Lists a summary view of the endpoint that was recorded or virtualized. |
| Recipient | Lists the destination where messages sent to are captured. |

3. Expand the agent or proxy.

   The details of the agent or proxy are displayed in the **Details** tab in the expanded panel.




4. Click each tab to find the following information:

| Tab | Description |
|---|---|
| Details | The **Details** panel provides the following information:<br><br>**Option** / **Description**<br>Condition — Lists the condition that causes the traffic to be recorded or routed to a virtual service.<br>Routing to — Lists the stub or agent to which traffic is routed.<br>Created by — Lists the user who created the rule.<br>Created at — Lists the date and time when the rule was created. |

| Tab | Description |
|---|---|
| Activity | The **Activity** panel lists the log of the activities performed by the agent or proxy. This panel also displays the details of the activity, date and time, and message for that activity. |

5. View the detailed condition of the rule defined for the agent or proxy by clicking the **View** icon.

   The detailed condition of the rule is displayed in the **Full condition** window.

   **Note:** You can copy the detailed condition if you intend to use it elsewhere.

**Results**

You have viewed the routing rules for the agents, proxies, or stubs that are running or connected to Rational® Test Automation Server.

**What to do next**

You can perform any of the following tasks:

- You can view the stub usage statistics. See Viewing stub usage statistics on page 195.
- You can stop the stubs that are running on Rational® Test Automation Server. See Stopping a stub on page 199.

---

Related information

Management of virtualized services on page 183

Stubs: Starting or stopping

## Viewing stub usage statistics

After you start stubs that are connected to Rational® Test Automation Server, and then run tests on the stub, you can view the stub usage statistics from the **Stub Usage** page on Rational® Test Automation Server.

**Before you begin**

You must have completed the following tasks:

- Created stubs in Rational® Integration Tester and committed the test resources to the remote repository.
- Started the stubs on Rational® Test Automation Server. See Starting a stub on page 188.

**About this task**

After you run the stubs, you can view details of the stub usage from the **Stub Usage** page on Rational® Test Automation Server.

**Important:** The settings that you configure (such as the **Period**, or **From** and **To**) to view the stub usage details are retained for you during your ongoing session and the settings are retained even when you perform the following activities:

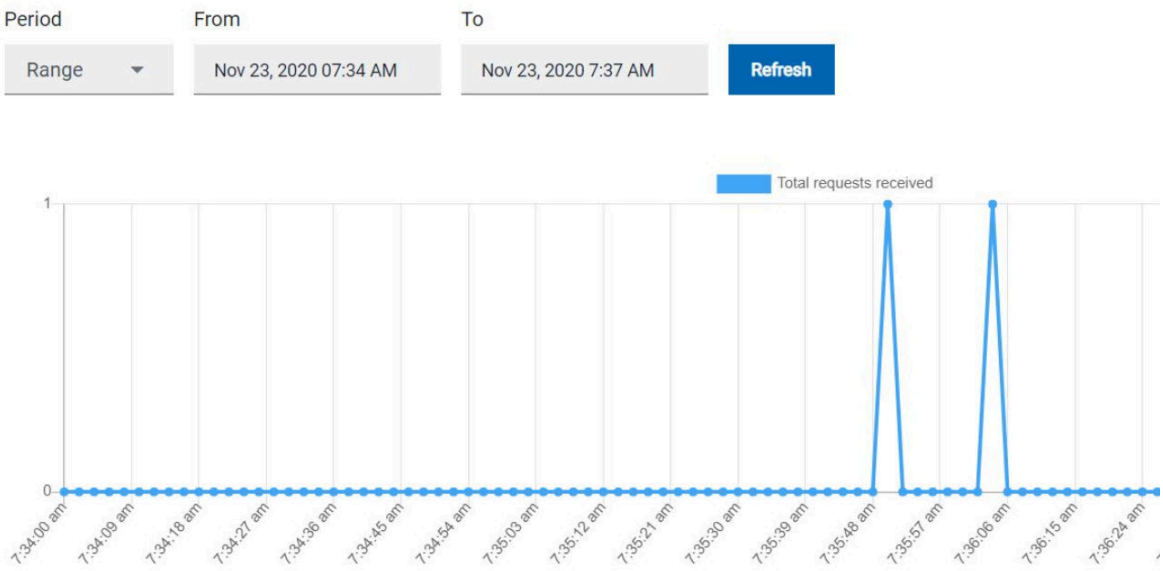
- You log out of the server and log in again.
- You log in after a server restart.

1. Open your project on Rational® Test Automation Server.

   The **Overview** page is displayed as the default page after you created the project and logged in subsequently.

2. Click **Stub Usage**.

   The stub details are displayed as a graph. The default view displays the stub usage data that is gathered during the previous week and grouped in an hourly interval. The total requests received from when the stub was started are displayed.




   The metric data that is reported is for the number of requests that are received by a stub. These can be requests, messages, or other calls that are received by the stub.

   You can change the default view by selecting an option from the following options:

| Option | Description |
|---|---|
| *1 Week* | Displays the stub usage data that is gathered during the previous week and grouped in an hourly interval. |

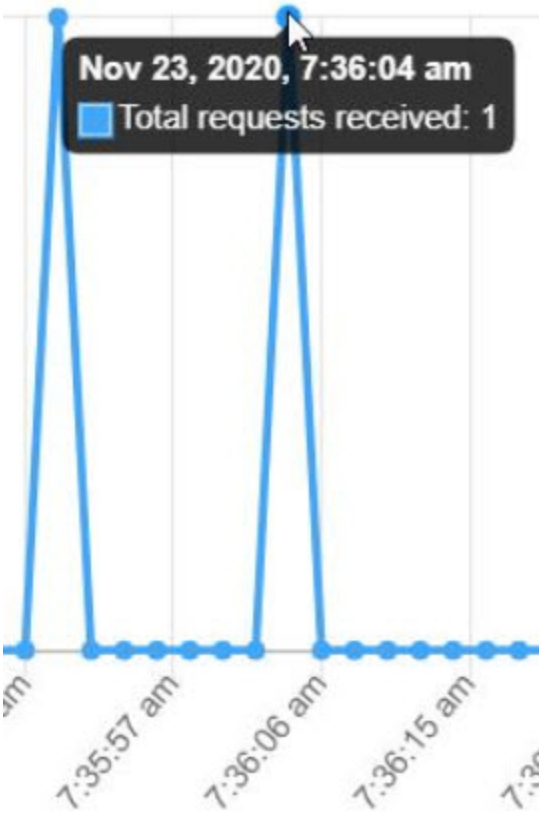
| Option | Description |
|---|---|
| | **Note:** This is the default view. |
| *1 Day* | Displays the stub usage data that is gathered during the previous day and grouped in an eight-minute interval. |
| *1 Hour* | Displays the stub usage data that is gathered during the previous hour and grouped in a 20-second interval. |
| *Range* | Displays the stub usage data that is gathered during the date range specified in the **From** and **To** fields. You can select the date and time for both the **From** and **To** fields. The data is dynamically grouped in intervals that depend on the range selected. |

**Note:** You can refresh the view at any time to present the graph for the latest data by clicking **Refresh**.

3. View details of the stub by hovering the mouse over any of the data-points to view the stub details at that point.

   The following details about the stub are displayed when you hover the mouse over the data-point:

- Date and time of the request received.
- Number of requests received.




**Results**

You have viewed the stub usage details for the stubs that ran on Rational® Test Automation Server.

**What to do next**

You can perform any of the following tasks:

- You can view the routing rules defined in the stub. See Viewing routing rules on page 193.
- You can stop the stubs that are running on Rational® Test Automation Server. See Stopping a stub on page 199.

---

Related information

Management of virtualized services on page 183

Stubs: Starting or stopping

## Stopping a stub

When you want to stop a stub that is running in IBM® Rational® Test Automation Server, you can stop the running stub from the **Progress** page. You must stop a running stub after you have completed running tests that are used by the stub.

**Before you begin**

You must have started the stub from the **Execution** page. See Starting a stub on page 188.

**About this task**

You can stop a running stub from the **Progress** page by using the **Stop execution** icon .

1. Go to the **Progress** page and identify the stub that you want to stop.

   **Note:** You can stop a running stub if it is in the `In Transition` or `Running` state.

2. Click the **Stop execution** icon in the **Actions** column of the selected stub.
   **Result**

   The **Stop execution** dialog box is displayed.

3. Set the timeout period for stopping the stub. Enter a numeric value and select the unit from the options available such as *Seconds*, *Minutes*, or *Hours*.

   The time out period is the time during which the stub is allowed to stop on its own and after the timeout period, the stub is forced to stop abruptly.

   **Important:** If the timeout period is not set, a time of *30 seconds* is considered as the default timeout period.

   Although the following options are selected by default, these options do not apply for stopping a running stub:
   - **Capture results**

     **Restriction:** Results are not captured for the stubs started or run on Rational® Test Automation Server and the stubs are not displayed on the **Results** page.

   - **Execute 'finally-block', if present**
4. Click **Stop execution**.

   A notification message is displayed that the running stub is stopped successfully.

**Results**

You have stopped a running stub. The stopped stub is displayed on the **Progress** page with the status as *Stopped by User*. Stopped stubs are not considered in the count of test assets run that is displayed on the **Overview** page.

**What to do next**

You can restart the stub by completing the configurations that you want from the **Execution** page.

You can open and view the `Execution log` of the stopped stub from the **Actions** column on the **Progress** page by clicking the **Open action menu** icon .

# Test results

After the tests or schedules are run and completed, you can view the results and reports in IBM® Rational® Test Automation Server to analyze the verdict, the performance, and statistics. You can also re-execute tests and schedules from the **Result** page with the same commit id.

From the Results page, you can perform the following tasks:

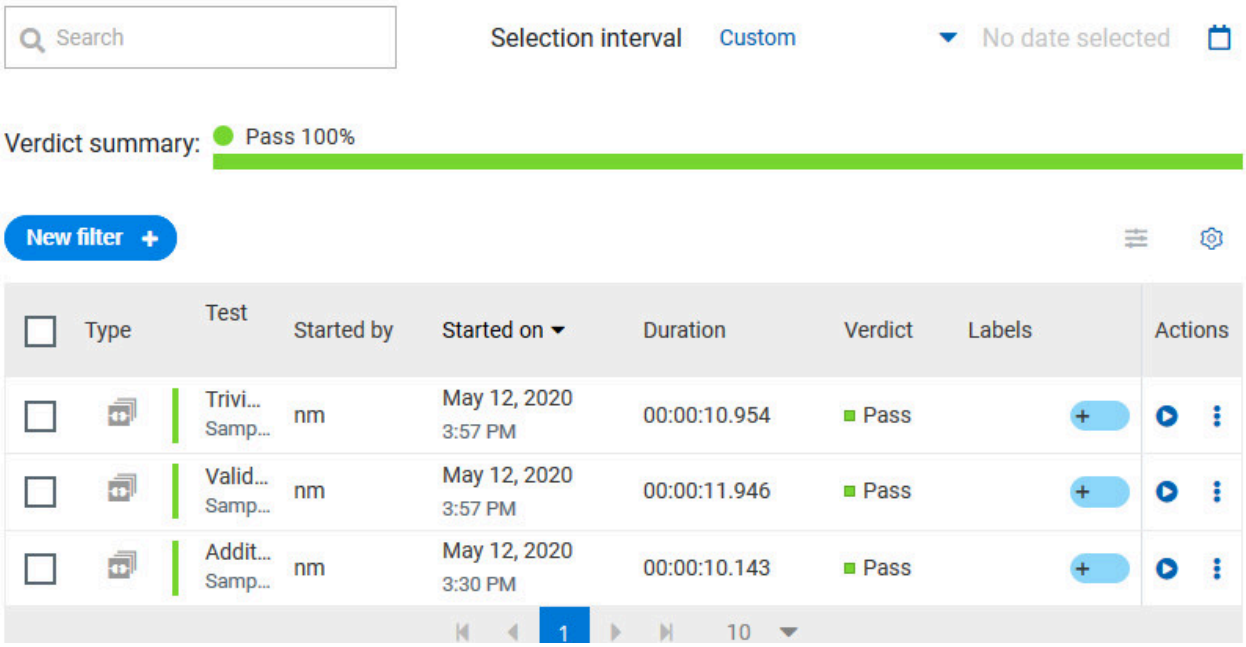
- Searching for test results. You can filter test results by using any of the following ways:
  - By using the **Search** field to search for results by name.
  - By selecting predefined time interval from the **Selection interval** drop-down list.
  - By selecting **To** and **From date** from the **Date Interval** option.
  - By clicking verdict from the **Verdict summary** slider (*Pass*, *Fail*, *Inconclusive*, or *Error*).
  - By creating filter queries.
- Adding labels if you have a tester or an owner role.
- Locking test results: Locked results can be unlocked by the project owner or the project member who locked the results.
- Deleting test results.
- Comparing performance reports.
- Viewing trending reports.
- Viewing multiple reports depending on the test types: Statistics reports, Mobile and Web UI reports, Functional reports, Unified Reports, or HTML reports generated from Postman or JMeter test runs.
- Re-executing tests with the same commit id.

  For more details, see the links in the next section of this page.

## Test results and reports overview

Rational® Test Automation Server is a single location for hosting the results and reports of all tests run on different desktop clients and for tests run from the server.

### Test results list




When you expand the results, the results details and reports cards are displayed.

Results, reports, and logs are generated for tests run from Rational® Test Automation Server, or from desktop clients such as Rational® Performance Tester, Rational® Functional Tester, or Rational® Integration Tester.

You must have configured the desktop clients to publish reports of tests that are run from the desktop client to the Rational® Test Automation Server. For more information about the publishing procedure, refer to the links at the end of this page.

### Test result details

By using the default settings, you can view the following details about the test assets used in the test run in the Details card:

- The status of the test results, the date and time it was executed and completed.
- The details about the Git repository that contains the test assets.
- The execution location that uses the test asset for a test run.

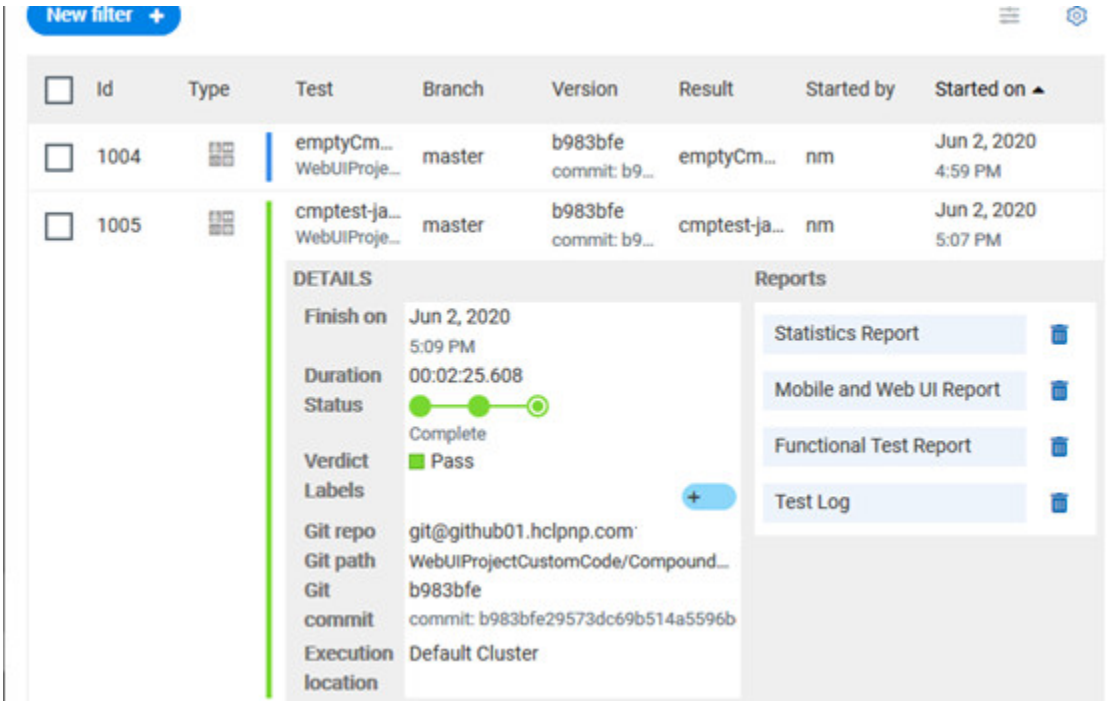
  **Note:** The execution location can be the default cluster or the location that you indicate to override a docker host or the agent location when you execute a test. For more information about test configuration, see the link at the end of this page.

- The Resource Monitoring labels (override) used in a performance schedule to control Resource Monitoring sources. For more information about controlling Resource Monitoring sources, see the link at the end of this page.

You can configure the information displayed in the **Details** card and the **Results** columns from the **View Settings** window. You access this window by clicking the **Settings** icon .

Depending on the attribute you cleared in the **View Settings** window, the Details card displays additional information about the test results.

The selected attributes are displayed in columns of the Result view.




## Reports

The Reports card contains the links to the test reports and the logs that are displayed in a web browser.

As a default configuration, test logs are delivered in a traditional format for the executed compound tests and schedules. You can still set the `-history jaeger` Program Argument to produce Jaeger traces when you run the tests.

The reports are generated for Compound Tests, Schedules, and Test Suites, or for other tests (Postman or JMeter tests for example). They can be run from a desktop client or from Rational® Test Automation Server.

The following table lists the reports that are generated for each type of test and the links to help pages.

**Table 11. Analyzing Reports**

| Report | Tests type | Product | More information |
|---|---|---|---|
| Functional Test Report | API Suite | IBM® Rational® Integration Tester | Viewing reports |

**Table 11. Analyzing Reports (continued)**

| Report | Tests type | Product | More information |
|---|---|---|---|
| | Compound Test<br>AFT Suite<br>Functional test (For Web UI tests) | IBM® Rational® Functional Tester | Unified Report |
| Mobile and Web UI Report | Compound Test | IBM® Rational® Functional Tester and IBM® Rational® Performance Tester | Mobile and web UI test result reports<br>Unified Report |
| | AFT Suite | IBM® Rational® Functional Tester | |
| Statistics Report | Compound Test | IBM® Rational® Functional Tester and IBM® Rational® Performance Tester | Reports and counters<br>Service Performance report<br>From the Statistic reports, you can add counters. Refer to Adding additional counters on a separate page.<br>You can also display the statistics of counters on graphs. Refer to Displaying counter data in tables or as graphs. |
| | AFT Suite | IBM® Rational® Functional Tester | |
| | VU Schedule<br>Rate Schedule | IBM® Rational® Performance Tester | |
| | Functional test (For Web UI tests) | | |
| Test Log | Compound Test | IBM® Rational® Functional Tester and IBM® Rational® Performance Tester | |
| | AFT Suite | IBM® Rational® Functional Tester | |
| | Functional Test (Web UI) | IBM® Rational® Functional Tester | |
| | VU Schedule<br>Rate Schedule | IBM® Rational® Performance Tester | |

Related information

Publishing test results to the server from Rational Integration Tester

Publishing test results to the server from Rational Performance Tester

Publishing test results to the server from Rational Functional Tester

Controlling Resource Monitoring sources in a schedule

Test run configurations on page 121

## Creating search queries to filter test results

To filter the test results list, you can create a search query that you apply as a filter in IBM® Rational® Test Automation Server.

**About this task**

You can use rules and group of rules in your query. This procedure explains how to create a query that contains two rules and a group of rules as an example, and how to select other filters.

1. Click **New filter** on the Result page.

Follow these steps to add a rule in your query:

2. Select **Add a rule** in your query, and enter or select the required attribute in the fields to define the rule:

    a. Select the first attribute that you want to query in the drop-down list.
    **Example**
    Select **Verdict**.

    b. Select the condition as *equal to (=)* or *not equal to (!=)*, *Contain* or *Not contain*.
    **Example**
    Select **=**.

    c. Select a value for the first attribute you selected.
    The content of this field depends on the option you selected as first attribute. Enter a value for the **id**, **TypeTest**, **Version**, **Result**, **Labels**, **Branch**, **Test path**, and **Text** attributes. For the other attributes, select a value in the drop-down list.
    **Example**
    Select **Fail**.

    d. Select either the **AND** or the **OR** operator in the first field to add a condition to compare the first rule with another rule.

    e. Select **Add rule** in the drop-down list.

    f. Select an attribute, a condition, and select or enter a value.

**Example**

Select **Test** as an attribute, select **=** as a condition operator, and enter *mytest* as a value.

Follow these steps to add a group in your query:

3. Select **Add group** in the drop-down list.

   **Note:** A group is a collection of rules and can also contain other groups within a query. You can add a rule or a group within this group.

4. Select a condition operator.

   **Example**

   Select **Or**.
5. Select **Add rule** in the drop-down list to add a rule in the group.
6. Select the required an attribute, an operator, and enter a value for the rule.

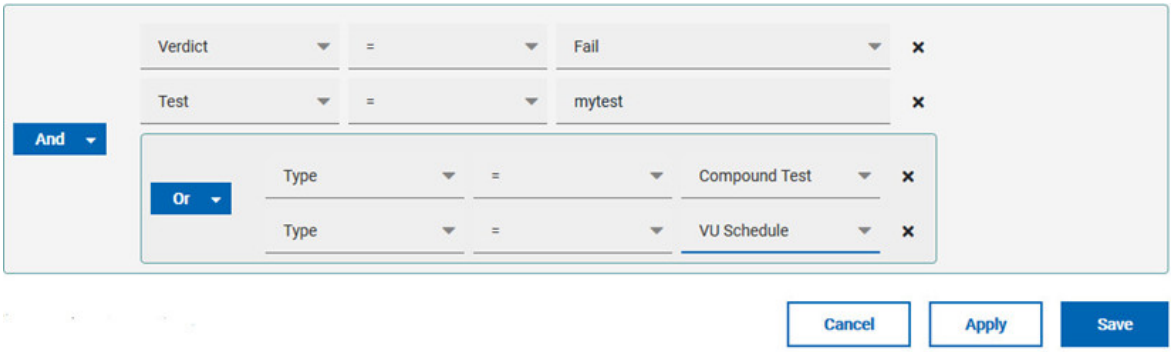
   **Example**

   Select the **Type** attribute, **=** as an operator, and **Compound Test** as a value.
7. Select **Add rule** to add another rule in the group.
8. Select and enter the appropriate parameters as you did for the other rules.

   **Example**

   Select **Type** as an attribute, **=** as a condition, and **VU Schedule** as a value.
9. Click **Apply** if you want to apply the filter now.

   You can cancel the query or save the query when the results are found.



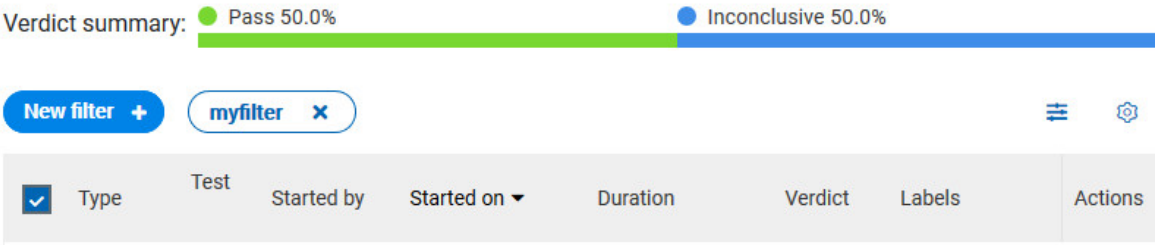

10. Click **Save**.

Proceed as follows to select your filters:

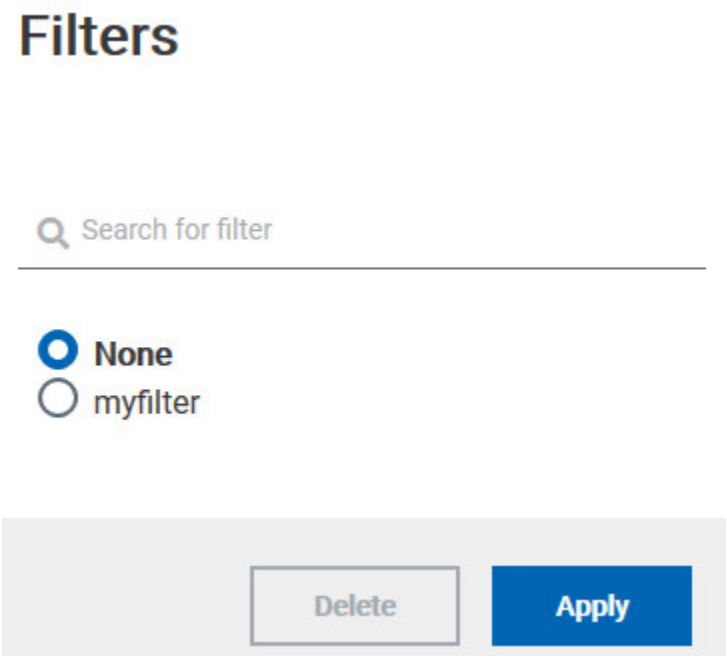
11. Enter a name for the filter and save in the dialog box that is displayed.

    **Result**

    The filter is displayed as a favorite filter.




12. Click the icon to select the filters you created from the list of saved filters.




13. Click a filter in the list of saved filters. If a filter does not display in the list, enter the name of a filter in the search field and select it.

    **Note:** You can delete a filter from the list or disable filters by selecting **none** in the list.

14. Apply the selected filter.

    **Result**

    The filter is displayed as a new favorite filter in the **Results** page.

## Comparing test reports

After you run performance tests, you can analyze the difference between two or multiple test results in IBM® Rational® Test Automation Server. For example, you can compare the performance of an application at different time slots or different milestone builds between two test results.

**About this task**

You can compare the test runs for performance tests that are in the same project or in different projects.

1. Select results in the **Results** page by checking the boxes in the first column.
2. Click the icon

**Note:** When comparing multiple runs, you cannot compare multiple time-ranges or stages.

**Result**

The two or multiple test results are displayed in a browser window in the same report.

## Viewing trending reports

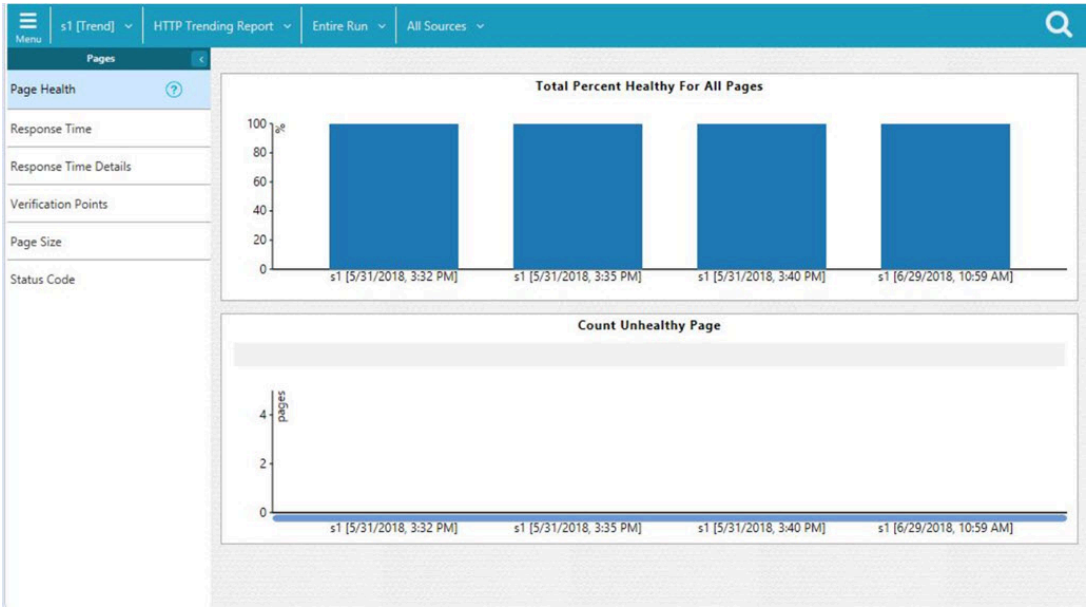
When the performance test runs are complete, you can view the trend of response time for an application over a period of time in IBM® Rational® Test Automation Server from trending reports. In addition to the response time, you can view the trend for the loops, transactions, and performance requirements for the application.

**About this task**

The trending reports are available for performance tests only.

1. Select results in the first column in the **Results** page.
2. Click the icon to view the trending reports.

**Note:** When comparing multiple runs, you cannot compare multiple time-ranges or stages.




**Results**

You can analyze the trend information for multiple test results in the same report from a browser window.

## Re-executing tests from results

You can re-execute a test, a schedule, a collection of tests or schedules from IBM® Rational® Test Automation Server if you want to keep the same commit id.

**Before you begin**

You must have executed a test or multiple tests from the **Execution** page in IBM® Rational® Test Automation Server.

Re-executing a test from the **Results** view requires a Tester or an Owner role.

**About this task**

The following procedure describes how to re-execute a single test or multiple tests at a time.

Perform all the following steps from the Results page.

- Proceed as follows to re-execute one test at a time:

  1. Identify the test that you want to re-execute and click the **Re-execute** button in the Actions column.
     **Result**
     The **Execute test asset** dialog box displays the same parameters as the ones that were set to execute the initial test from the **Execution** page.

2. Select another **Version** of the test.

   **Note:** If you select another version of the test, the settings configured in the dialog box and in the **Environment**, **Data Sources** and **Locations** tabs might change.

3. Modify the **Schedule** parameters.

   **Note:** If you don't want to execute the test now, schedule the time when you want the test to be re-executed.

4. Click the **Environment** tab and select another **API test environment** if multiple environments were initially set for a test that is running an API Suite.
5. Click the **Data Sources** tab and select an override option if multiple data sources were defined for the initial test run.
6. Click the **Variables** tab and add a new variable if you want to re-execute a test asset with a different variable from the one configured in the asset.
7. Click the **Location** tab to override the default cluster location if the test asset has a docker host or static agents configured.
8. Click **Advanced** to modify the advanced parameters that were set for the initial test run.
9. Enter a new **Program Argument** if applicable to the test.

   **Note:** If Jaeger is installed with IBM® Rational® Test Automation Server, you can enter the -history jaeger Program Argument to produce Jaeger traces when you run the tests.

10. Enter a new instance of **Java arguments** if applicable to the test.
11. Enter the environment variables that must be passed to the test run at runtime in the **Environment Variables** field.
12. In the **Resource Monitoring** tab, click the , and press the Ctrl + Space keys, or enter the initial letter of a label to select a label in the list and override the current sources in the schedule.

    **Note:** The **Resource Monitoring** tab is displayed for performance schedules and only if you have enabled the **Resource Monitoring from Service** option from Rational® Performance Tester.

    The label field is grayed if you don't have any label entered in your current project. To collect data during the test execution, you must first enter labels in the Resource Monitoring sources page from IBM® Rational® Test Automation Server.

For more details about controlling Resource Monitoring sources and labels, see the links at the end of this page.

13. Click the **Re-execute** icon to re-execute the test now or at a scheduled time.

- Proceed as follows to re-execute multiple tests at a time:

1. Click the checkboxes for the tests that you want to re-execute.
   You can select them one by one or select all of them.
   **Result**
   A toolbar is displayed.
2. Click the **Re-execute** icon in the toolbar.
   **Result**
   A message indicates the number of selected tests and the number of tests to be re-executed. The tests are re-executed with the same initial commit ids and parameters.
3. Click **Execute**.

**Result**
You can view the results, logs, and reports of the re-executed tests in the **Results** page.

---

Related information

Adding details to a server project on page 282

Test results and reports overview on page 200

Controlling Resource Monitoring sources in a schedule

Tests configurations and test runs

# Resource Monitoring service

When you apply load to a system under test, the amount of resources consumed by your system is increasing. If the capacity of the resources does not match the load, you can see performance issues in the results. The Resource Monitoring service in Rational® Test Automation Server helps you monitoring the resources of a system and establish the performance metrics of the system. Thus, you can observe the health of these resources while a schedule is running.

You can use the Resource Monitoring service while running a schedule to capture data, such as processor or memory usage, or to monitor the availability of hosts and services by using counters. The Resource Monitoring service can provide a comprehensive view of a system under test to help determine problems. Hosts and services can be servers, virtual machines, or any local host or network services. You can also monitor remote hosts and services with agents.

**Monitoring local host and service**

You can use the Resource Monitoring service to monitor any local host and network service. In this case, the resources of the hosts are locally monitored. This method is used for host targets that the service can reach directly. With Rational® Test Automation Server, you can also monitor performance metrics that are collected and stored by a monitoring system that is monitoring a host. You can use Prometheus server or a data collector through an exporter, OpenMetrics exporter, for example.

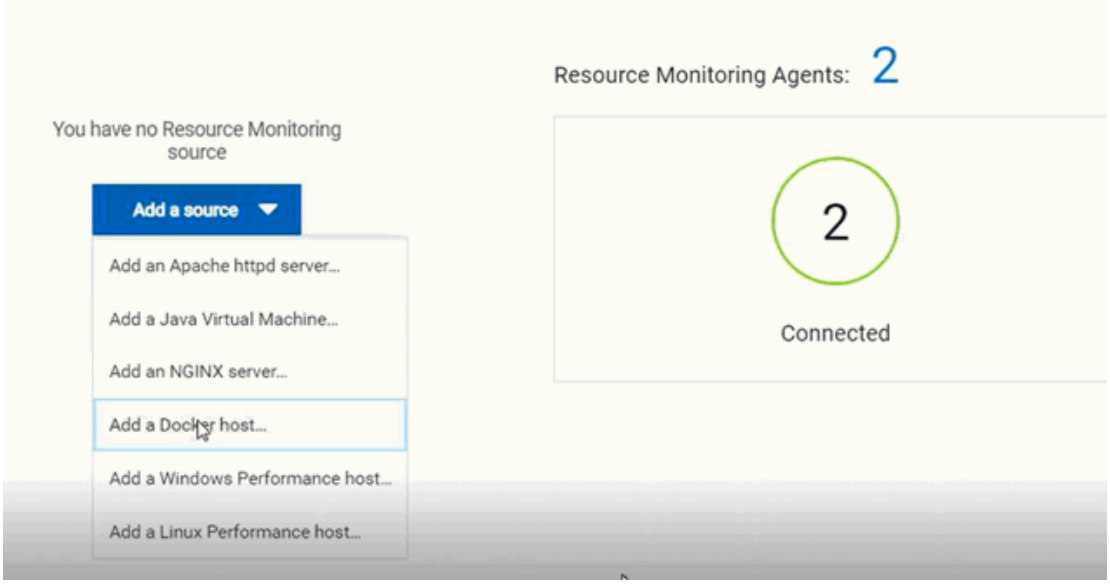
To monitor resources on a local system, you must add the source of the resource monitoring data from the Resource Monitoring page. Then you have to enter some connection settings and select the performance counters to monitor the sources. The metrics are exposed in a graph for each selected counter.

**Monitoring remote host with monitoring agents**

You can use the Resource Monitoring service to monitor remote hosts and services with agents through wider sets of data collectors, computers, and networks. The purpose is to capture CPU load, disk space, memory, and the running process for example.

Agent-based monitoring is useful when remote services are not directly accessible through the network. Agents are closer to the target that you want to monitor. You can set up the agent on an authorized host, for example, when access to an **Apache httpd** or **NGINX** server status page or to a **JVM** JMX port is restricted to one or few client hosts only. The configuration task is simplified, and no security changes are required.

To monitor a host across monitoring agents, you must first set up the Resource Monitoring agents on the target host for which you want to collect the performance statistics. The agents establish a connection with the Resource Monitoring service. When the connection is set, the agent is showing up in the list of sources in the Resource Monitoring main page. You can select it, choose performance counters, and view the metrics statistics in a graph for each selected counter.




**Resource Monitoring sources**

With Rational® Test Automation Server, you can monitor resources for the following sources:

- **Apache httpd server**
- **NGINX and NGINX Plus**
- **Java Virtual Machine**: You can monitor JVM resources from a local or from a remote system. Some parameters must be set in the command before running the Java Virtual Machine. For more details, see the link to the page about starting a Java Virtual Machine on this page.
- **Windows Performance host**: To monitor the performance of a Windows host, you must have installed an agent on the target Windows host.

  If IBM® Rational® Test Automation Server is installed on a Linux system, and you want to check the performance of the Windows host, you must have installed an agent on the target Windows host to start monitoring its performance.
- **Linux Performance host**: To monitor the performance of a Linux host, you must have installed an agent on the target Linux host.

  If IBM® Rational® Test Automation Server is installed on a Windows system and you want to check the performance of the Linux host, you must have installed the Resource Monitoring agent on the Linux system.
- **Docker host**: A Resource Monitoring agent is mandatory to get a Docker Data Collector. The Docker host source is added to the list of resource monitoring sources when the agent is installed.
- **Prometheus server**: You can monitor metrics of a host under test that are collected by a Prometheus server. Prometheus collects metrics from monitored targets by regularly requesting appropriate HTTP endpoints on these targets (called *scraping*).

The metrics data are collected through data collectors sets (groups of performance counters) that are tracking the system performance. The performance data results are stored under a Prometheus REST API so that they can be consumed by external systems.

With the Resource Monitoring service in Rational® Test Automation Server, you can query these performance metrics collected by Prometheus servers. 'PromQL' is the language for querying Prometheus metric data.

In the Resource Monitoring service, you can select default queries, or create additional ones.

A Prometheus server is required and it must be configured to scrap a set of exporters, every 15 seconds by default. Pushgateway, service discovery, and Alertmanager are optional.

- **OpenMetrics exporter**:

  You can monitor the metrics of a host under test that are collected and exposed through an OpenMetrics format or a Prometheus exporter format.

  The Resource Monitoring service in Rational® Test Automation Server is given the ability to scrap metrics exposed by an OpenMetrics or Prometheus exporter through metric counters. No Prometheus server installation is required, you only need to set up one or multiple OpenMetrics, or Prometheus exporters that fit your software or host target. Refer to the link at the end of this page to see the list of exporters.

Click the links in the next section of this page for more details on the Resource Monitoring tasks that you can perform in Rational® Test Automation Server.

---

Related information

Exporters and Integrations

Starting a Java Virtual Machine on page 236

## Resource monitoring capabilities

With Rational® Test Automation Server, you can monitor a host or a service that you add to a project as a source. You can select the statistics counters that are used to collect metrics and the resource usage data throughout the monitoring system. You can visualize metrics and data that are gathered during the monitoring process in Rational® Test Automation Server.

**Requirements:**

- You must be a *Team Space Owner* to create, modify and remove resource monitoring sources in a team space. Any team space *Member* (not owner) can only view resource monitoring sources.
- You must be a project *Owner* to create, modify and remove resource monitoring sources in a project. Any project *Member* (not the owner) can only view resource monitoring sources.

**Supported sources**

With IBM® Rational® Test Automation Server, you can monitor resources for the following sources:

- **Apache httpd server**: You can optionally use agents.
- **NGINX and NGINX Plus**: You can optionally use agents.
- **Java Virtual Machine**: You can optionally use agents.

  You can monitor JVM resources from a local or a remote system. Some parameters must be set in the command before running the Java Virtual Machine. For more details, see the link to the page about starting a Java Virtual Machine on this page.

- **Windows Performance host**:

  To monitor the performance of a Windows host, you must have installed an agent on the target Windows host

  or on a Windows host that is configured to access the performance data of the Windows host target.

- **Linux Performance host**:

  To monitor the performance of a Linux host, you must have installed an agent on the target Linux host.

- **Docker host**:

  A resource monitoring agent is mandatory to get a Docker data collector. The Docker host source is added to the list of resource monitoring data collectors when the agent is installed.

- **Prometheus server**:

  You can monitor metrics of a host under test that are collected by a Prometheus server. Prometheus collects metrics from monitored targets by regularly requesting appropriate HTTP endpoints on these targets (called *scraping*).

  The metrics data are collected through data collectors sets (groups of monitoring counters) that are tracking the system performance. The performance data results are stored in the Prometheus database so that they can be consumed by external systems through a REST API.

  In Rational® Test Automation Server, you can query these performance metrics collected by Prometheus servers. 'PromQL' is the language for querying Prometheus metric data. Refer to Prometheus PromQL documentation for information about this query language.

  You can select default queries, or create additional ones.

  A Prometheus server is required. It must be configured to scrap a set of exporters, every 15 seconds by default. Pushgateway, service discovery, and Alertmanager are optional.

If Rational® Test Automation Server cannot directly reach the Prometheus server, you can consider setting up an agent on an authorized host or on the same host that is running Prometheus.

- **OpenMetrics exporter**:

  You can monitor the metrics of a host under test that are collected and exposed through an OpenMetrics format or a Prometheus exporter format.

  Rational® Test Automation Server can scrap metrics exposed by an OpenMetrics or Prometheus exporter through metric counters. No Prometheus server installation is required, you only need to set up one or multiple OpenMetrics, or Prometheus exporters that fit your software or host target.

  If Rational® Test Automation Server cannot directly reach the Prometheus server, you can consider setting up an agent on an authorized host or on the same host that is running Prometheus.

## Monitoring host resources

To monitor host resources, you must add the monitoring sources to the Resource Monitoring service, enter connection settings and select performance counters that are used to capture the performance statistics.

**Before you begin**

You must be logged in Rational® Test Automation Server and be the owner of an existing new project.

If you monitor a remote host, it must be connected with the computer that you use to access Resource Monitoring service. See Installing Resource Monitoring agents on remote hosts on page 228.

**About this task**

This task applies to NGINX and NGINX Plus, Apache httpd server, Java Virtual Machine, Windows Performance host, Linux Performance host, and Docker host sources.

1. In the Resource Monitoring page, click **Add a source**, for example: **Add an NGINX server**.
2. In the **New server** dialog box, fill in the following connection settings:

   a. In **Target host**, enter the IP address or host name and port number of the host where the server to monitor is installed.

   b. In **Server edition**, select the appropriate server from the drop-down list.

   c. In **Path to the status page of the server**, enter the name of the page to view the status of the server.

      If you select NGINX Plus (with API version 3) as a source, you must specify the name of the path to view the API root of the NGINX Plus server.

d. In **Security**, select the following options:

**Secured with TLS/SSL**

If the application server is secured with TLS/SSL.

**Trust self-signed certificate**

To accept the server certificate.

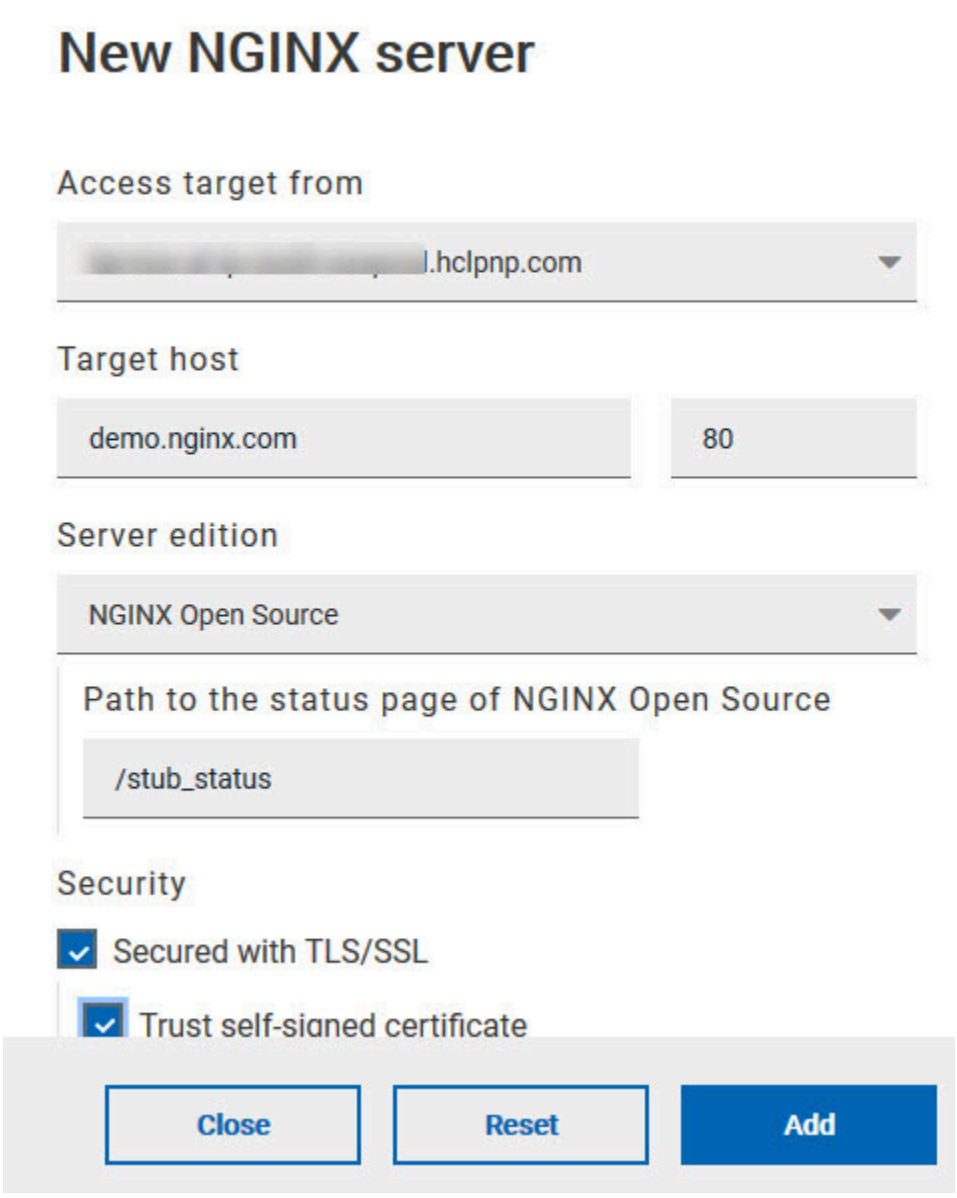
**Do not verify host name**

To ignore verification of the host name in the certificate.

**Require credentials**

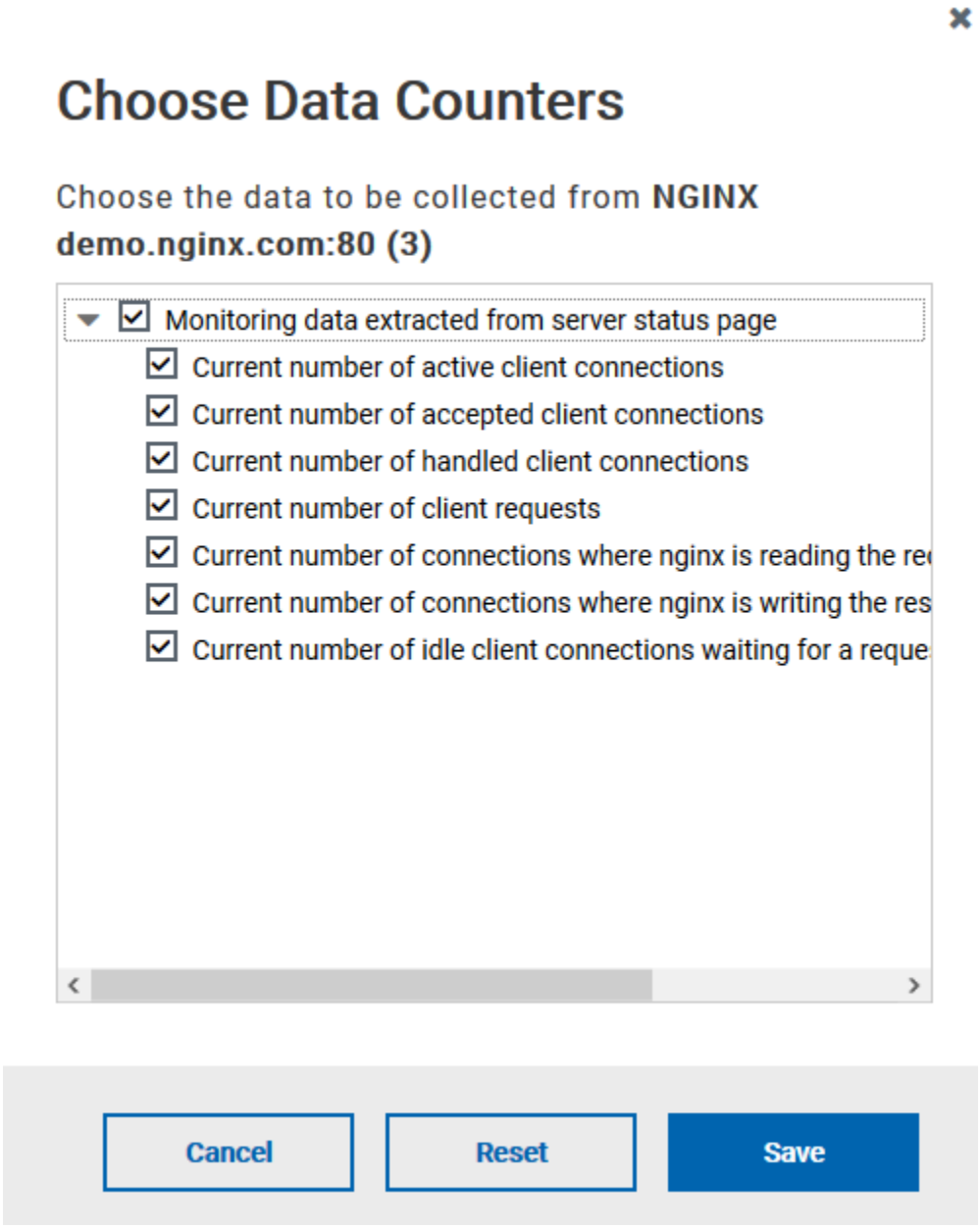
If the server requires log-in credentials, enter a **User name** and **Password**.

**Note:** For remote hosts that are already connected to Rational® Test Automation Server through an agent, you have only the **Access target from** field enabled.

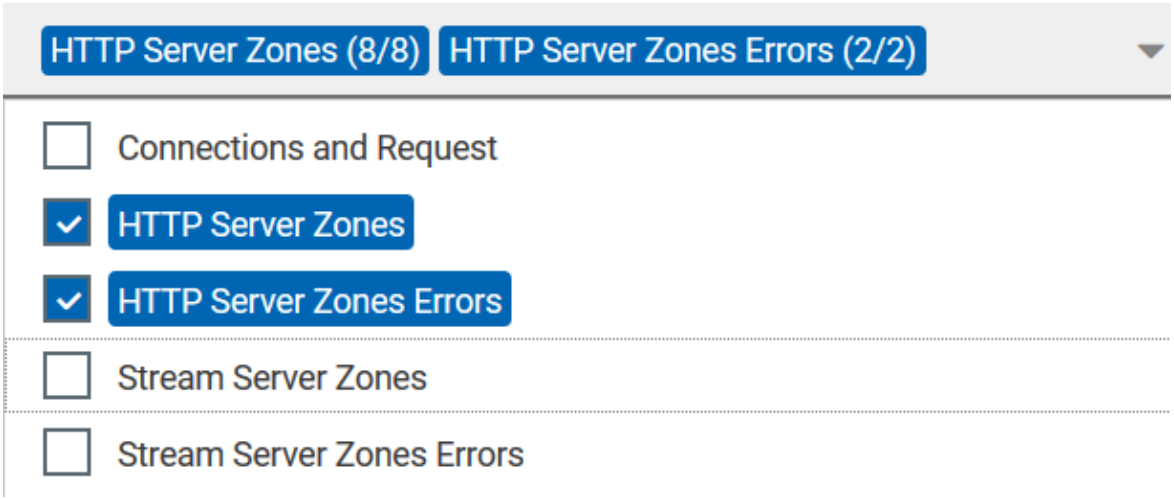



e. Click **Add**.

3. Select and save the resource counters to monitor. You can select them from the list where they follow the server logical organization.




For a faster selection, select the counters from the built-in sets drop-down list where they are organized by theme and save your selection.




**Note:**

For Docker host, you can select different sets of counters.

- The first level of counters: Generic counters that are related to a Docker Image and all its running Docker Containers (in an exit, running or paused, created, restarted, removing, dead or transitive state).
- The second level of counters: Specific counters that are related to the existing Docker Containers.

**Result**

When the selected counters are saved, two tables are displayed in the resource monitoring main page. They contain the total number of sources you have added and the number of sources ordered by type.

**What to do next**

You can click the links in the tables to view the performance metrics of your monitored system.

# Monitoring metrics collected by a Prometheus server

You can monitor your system resources by using metrics data that are collected by a Prometheus server while executing a schedule.

**Before you begin**

You must have created a project in Rational® Test Automation Server.

You must have installed and configured a Prometheus server.

Read the concept information about Resource Monitoring and monitoring Prometheus server source. See Resource Monitoring service on page 210

**About this task**

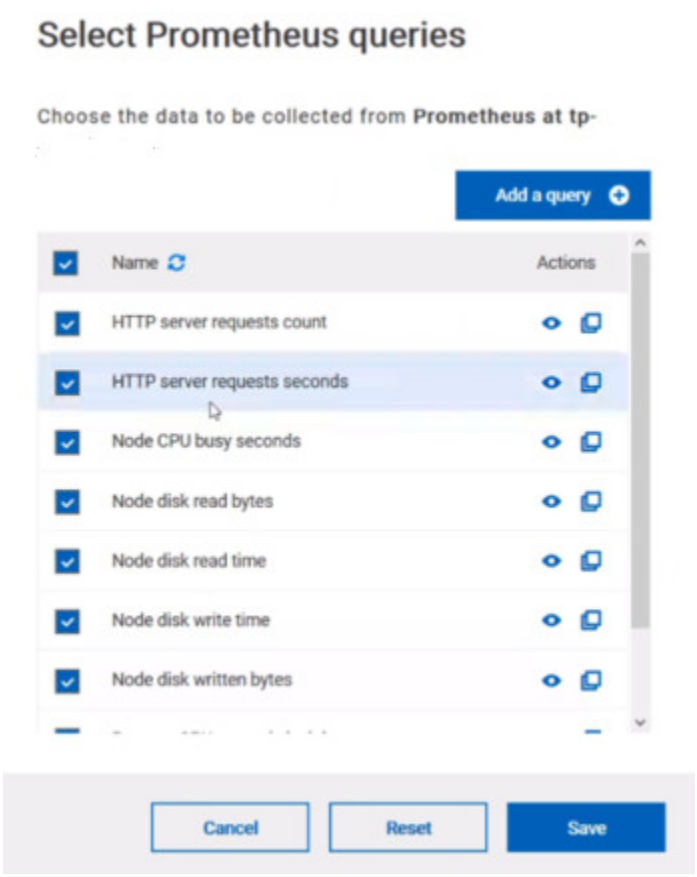
The following procedure describes how to select and create Prometheus queries in Rational® Test Automation Server. You must set a connection between Rational® Test Automation Server and the Prometheus server first to access to the Prometheus queries.

Enter the connection settings, then select queries.

1. Click **Add a source** and select **Add a Prometheus Server**.
2. Enter the connection settings in the **New server** dialog box,.
    a. In **Target host**, enter the IP address or host name and port number of the host where the Prometheus server is installed.
    b. Enter the **Path to the API root of Prometheus** to point to the Prometheus API where the monitoring data are saved.
    c. Enter the security parameters if required. For information about this section, see related links.
    d. Click **Add** to close the dialog box.

    **Result**

    The Prometheus server source is created successfully in the background and a dialog box opens to select the Prometheus data queries.
3. In the **Select Prometheus queries** dialog box, select queries that are available in the list to collect Prometheus metrics data by clicking the checkboxes.
    You can reset the selection.




4. Follow these steps to create a query:

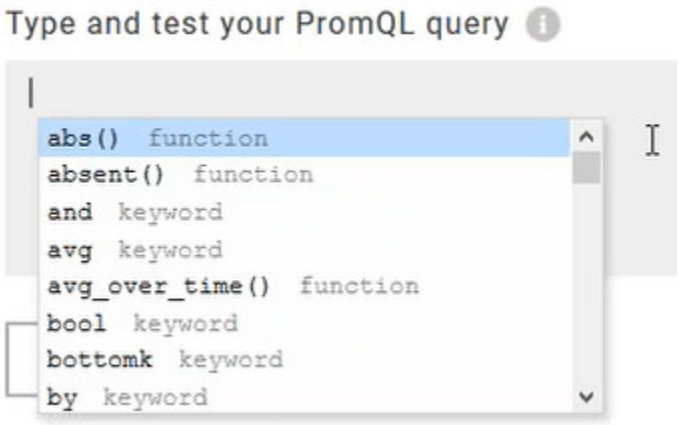
    a. Click **Add a query**.

    b. Enter a name for the new query in the **Create a Prometheus query** dialog box.

    c. Enter a PromQL query.any keyword and press Ctrl+Space in the **Type and test your PromQL query.**

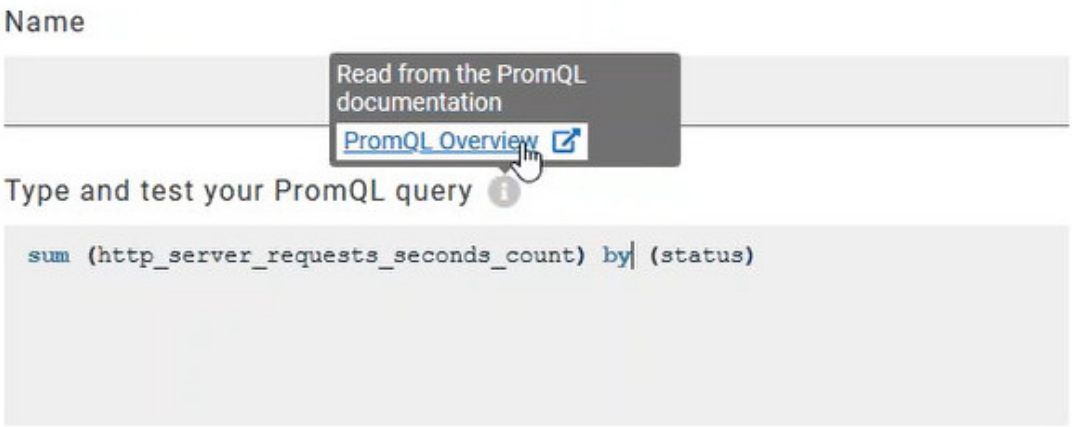
    **Result**

    All the possible combinations of keywords and metrics related to your searched keyword are listed.




d. Select a query in the list.

e. Click the link that points to the 'Querying Prometheus documentation' page in the information tool tip to find other examples of queries in a web page that you can use as samples.




f. Click **Test query.**
If the test is successful, a green check is displayed, otherwise you get a red check with an error message.

g. Click **Create**.

5. Proceed as follows to duplicate a query:

a. Click the **Duplicate** icon to duplicate a standard query.

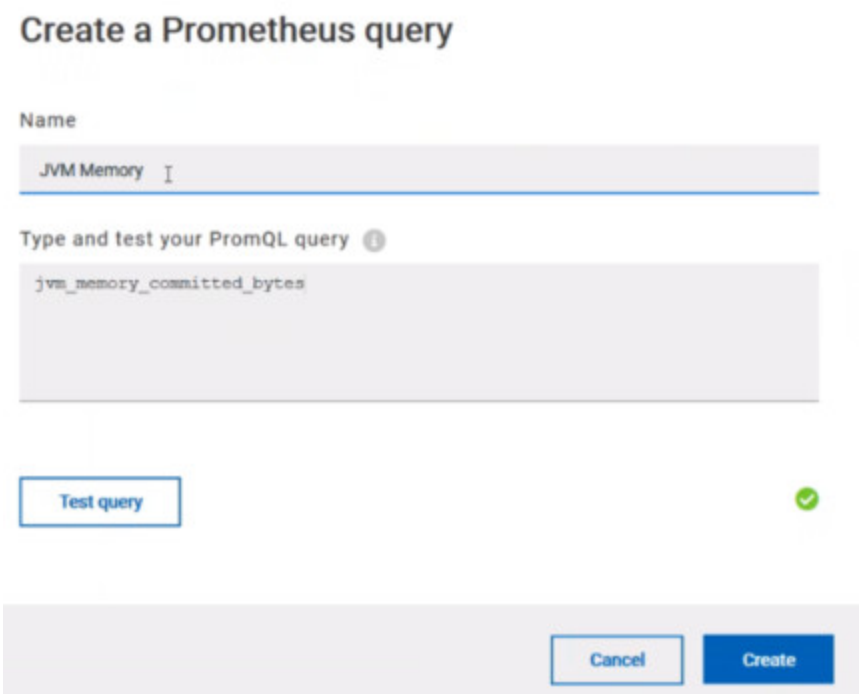
b. Rename the duplicated query in the **View the Prometheus query** dialog box.
Make sure that the name you give to the query is not already used.

c. Modify the query type.

d. Test the query .

e. Click **Create**.

**Note:** You can click Create or Duplicate only if the test of the query is successful and the name of the query is not already in use.




6. Proceed as follows to modify a query:
    a. Click the **Modify** icon .
    b. Modify the name and type of the query.
    c. Test the query.
    d. Click **Update**.
    e. Delete duplicated or created queries.
    f. Click **Save** to close the **Select Prometheus queries** dialog box.
7. Click Save and close the **Select Prometheus queries** dialog box.

**Results**

The Resource Monitoring page displays the total number of sources you have added to your project and the number of sources ordered by type in cards.

**What to do next**

Click the links in the cards to go to the sources page and see the performance metrics for each source you added to your project. For more details, see Viewing the performance metrics

Related information

Resource Monitoring service on page 210

Monitoring host resources on page 215

Viewing the performance metrics

## Monitoring metrics exposed by an OpenMetrics exporter

With Rational® Test Automation Server, you can monitor your system resources by using metrics data that are exposed by an OpenMetrics or a Prometheus exporter through metric counters while executing a schedule.

**Before you begin**

You must have at least one exporter or software to expose monitoring metrics with an OpenMetrics or Prometheus format.

You must have created a project in Rational® Test Automation Server.

**About this task**

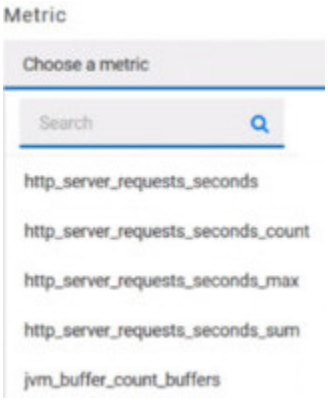
The following procedure describes how to select and create OpenMetrics counters in Rational® Test Automation Server. You must set a connection between Rational® Test Automation Server and the OpenMetrics exporter to select counters.

1. Click **Add a source** and select **Add an OpenMetrics exporter** .
2. In the **New OpenMetrics exporter** dialog box that opens, enter the OpenMetrics exporter details to access its exposition endpoint as follows:
    a. In **Target host**, enter the IP address or host name and port number of the host where the server is installed.
    b. Enter the **Path to the exposition endpoint** to point to the application where the data performance results must be stored. As the path depends on the type of exporter used, refer to the exporter's documentation and configuration to indicate the right path.
    c. In **Security**, enter the security parameters if required.
    d. Click **Add** to close the dialog box.
    **Result**
    The source is created successfully in the background. A dialog box opens to select the OpenMetrics counters.
3. Click the checkboxes to select counters in the **Select OpenMetrics counters** dialog box.
4. Follow these steps to create a counter:

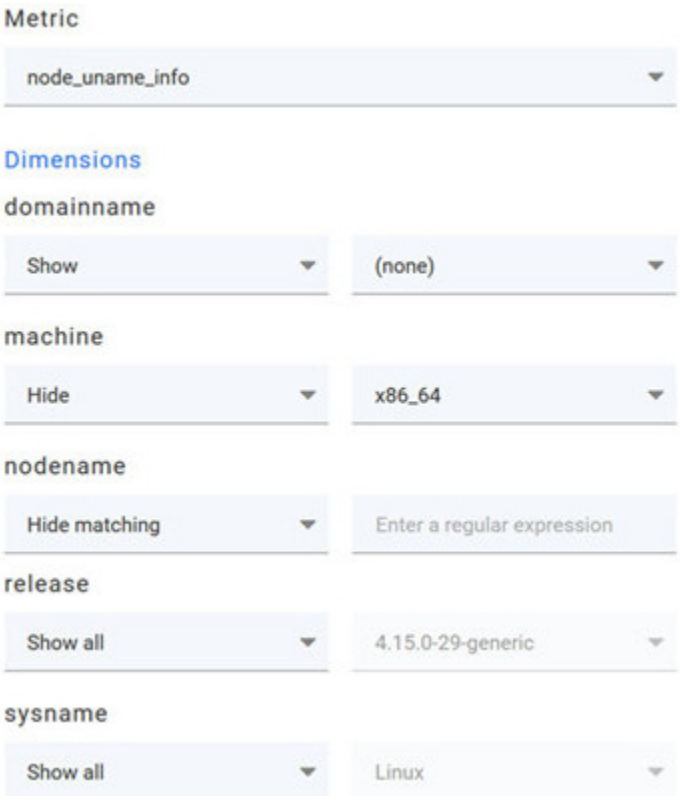
a. Click **Add a counter**.

b. Enter a name for the new counter, in the **OpenMetrics counter**.

c. Select metrics in the drop-down list or enter a keyword in the **Dimensions** field to enable the dynamic input help. You can also use the search field that appears in the list of metrics.

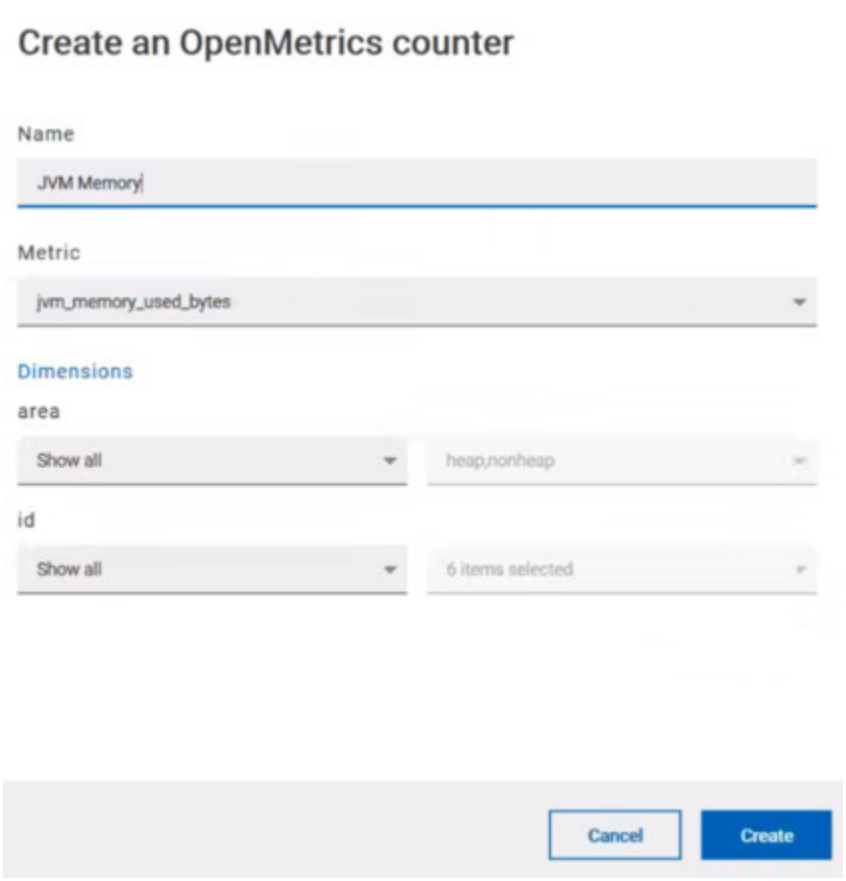



d. Select **Dimensions** from the dynamic fields when they are required.




e. Select the appropriate OpenMetrics counter that you are looking for.

f. Click **Create**.




5. Follow these steps to duplicate a counter:
    a. Select a counter in the list, and click the **Duplicate** icon .
    b. Modify the name of the counter and the Dimensions.
    c. Click **Create**.
6. Proceed as follows to modify a new counter:
    a. Click the **Modify** icon.
    b. Modify the name, Metrics, and Dimensions of the counter in the **Modify the OpenMetrics counter**.
    c. Click **Update**.

**Note:** You can create, duplicate, or update a counter only if the name of the counter is not already used.

7. Click **Save** and close the **Select OpenMetrics counters** dialog box.

**Note:** You can delete counters, but only the ones that you created or duplicated.

**Results**

The Resource Monitoring page displays the total number of sources you have added to your project and the number of sources ordered by type in cards.

**What to do next**

Click the links in the cards to go to the sources page and see the performance metrics for each source you added to your project. For more details, see Viewing the performance metrics.

---

Related information

Resource Monitoring service on page 210

## Resource Monitoring agents

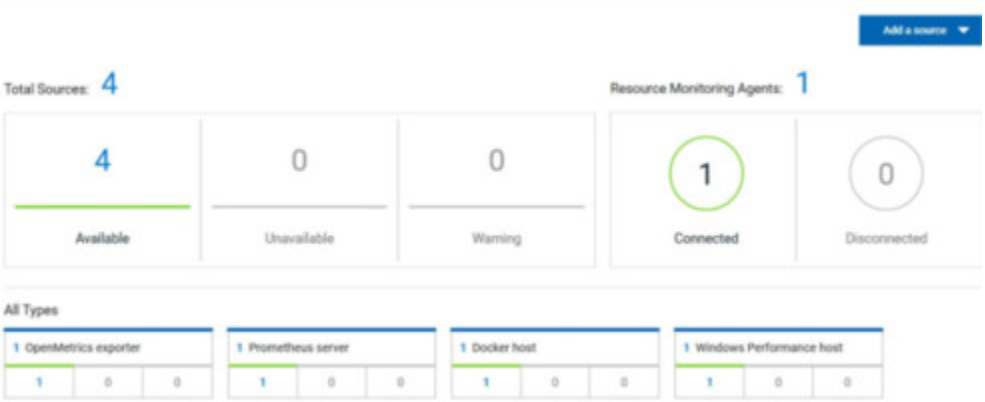
You can connect agents to the Resource Monitoring service from other hosts to monitor a larger set of data collectors, computers, servers, or systems, or if a server to monitor requires authorization access. To monitor these remote servers, you must configure and run the agents so that a connection is set between IBM® Rational® Test Automation Server Resource Monitoring service and the agents.

For testing, you would need many hosts. For example, you might have one host with the application server, another host with the database server, and some hosts to apply the user load. Due to network or firewall issues, sometimes, it becomes difficult for multiple hosts to connect to each other. Resource Monitoring agents are installed on the target hosts so that they can establish a connection with IBM® Rational® Test Automation Server to gather resource statistics of the target host.

The agents always try to connect with the server through the HTTPS protocol. You must install the agent and start it. From the Resource Monitoring service page in IBM® Rational® Test Automation Server, you can copy the command lines to download files and to run the agents. When you stop the agent, the monitoring sources that you have already added persist but the live data will not be available.

You can configure your Resource Monitoring agent as a service so that the agent automatically runs when the host is restarted. For details, see the documentation pages about starting Resource Monitoring agents on Windows or Linux.

When the agent is connected to IBM® Rational® Test Automation Server, it is added to the main page of the Resource Monitoring service.




To monitor your host across the agent, you must add the agent as a monitoring source and select performance counters to monitor as described in Monitoring host resources help page.

---

Related information

Resource Monitoring service on page 210

Monitoring host resources on page 215

Starting a Resource Monitoring agent as a service on Windows on page 230

Starting a Resource Monitoring agent as a service on Linux on page 233

## Installing Resource Monitoring agents on remote hosts

You must install the Resource Monitoring agents on the target host for which you want to monitor the performance statistics. You need to run the agents to establish a connection with the Resource Monitoring service.

**Before you begin**

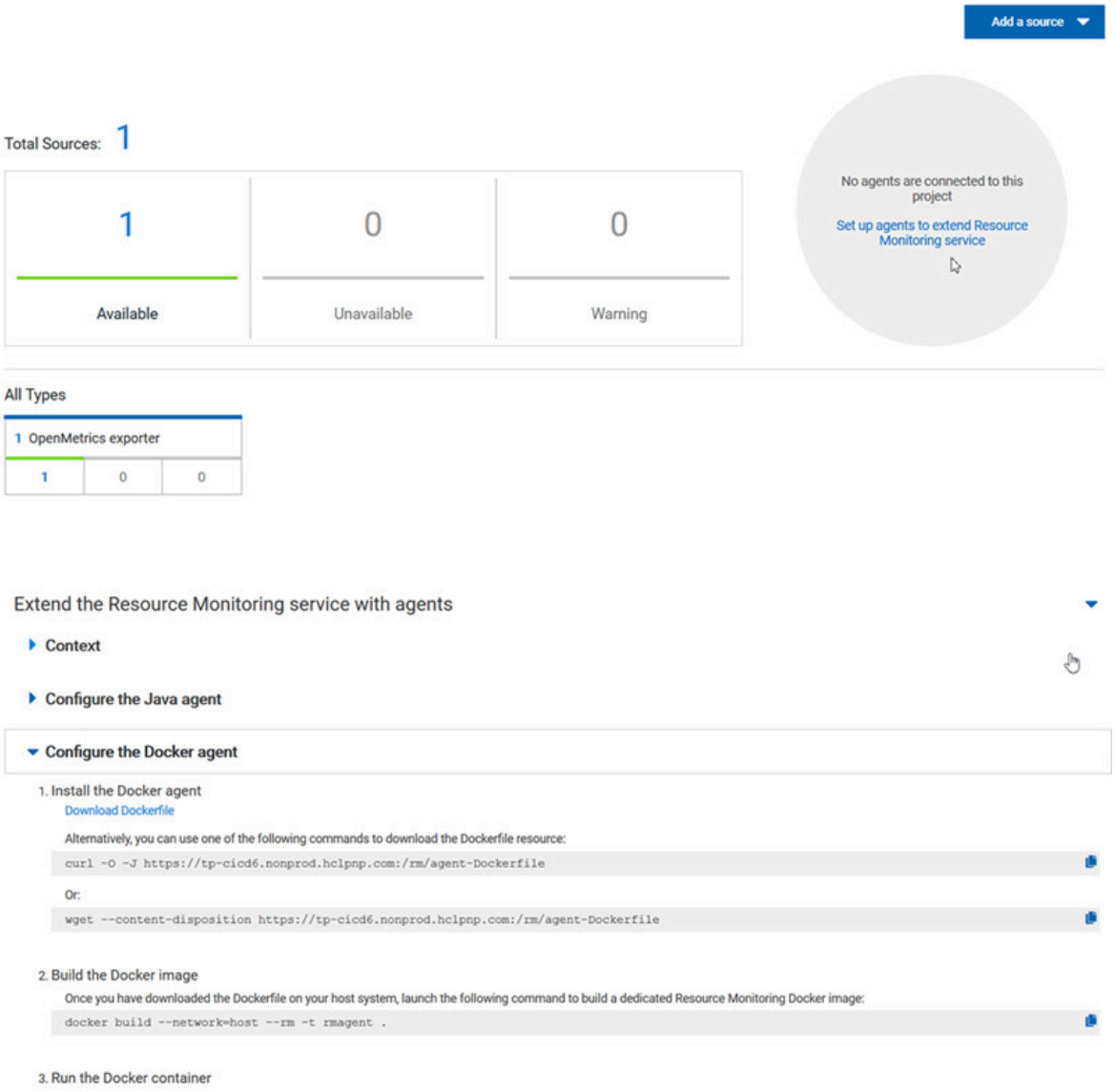
- The Resource Monitoring service does not require access to the agent host but the agent must have reached the service host over HTTPS.
- The Java agent must have been launched from a jar file and requires a Java 8 virtual machine.
- The Docker agent must have been launched in a Docker container and requires Docker 19.03.

**About this task**

This task applies to Java and Docker agents. However, the commands used to install agents are different from Java and Docker agents. You can find the commands and instructions in IBM® Rational® Test Automation Server in the Resource Monitoring agents page.

Before you start the Docker agents, you must build a Docker image.

1. Click **Set up Agents to extend Resource Monitoring service** link in IBM® Rational® Test Automation Server to access the agents page where you can find the instructions and commands that are to be used to install and run the agents.




2. Expand **Configure the Java agent** and click **Download jar file** to download the Java agent, or expand **Configure the Docker agent** and click **Download Docker file** to download a Docker agent.
   You can also use the *curl* and *wget* commands to download the agent without accessing the Resource Monitoring web UI. For more facility, use the code snippets to copy and fill in the commands with the valid offline token and the jar or docker file name.

```
curl -O -J https://hostName/rm/Agent-jar
wget --content-disposition https://hostName/rm/Agent-jar
```

3. Copy the build command that is under **Build the Docker image**, paste it in your console and run it to build a Resource Monitoring Docker image.

```
docker build --network=host --rm -t rmagent
```

**Note:** This step applies to Docker agents only.

4. Configure the command as follows to run your agent:
    a. Copy the appropriate command for Windows or Linux with the code snippets.
    b. Enter the path to the directory that contains the agent .jar file, and paste the command.

**Example**

On Windows:

```
set RTCP_OFFLINE_TOKEN=(Enter your offline token here)
java -jar (Enter the name of the downloaded jar file here) --ServiceUrl=https://hostName/rm
 --projectId=<project_id>
```

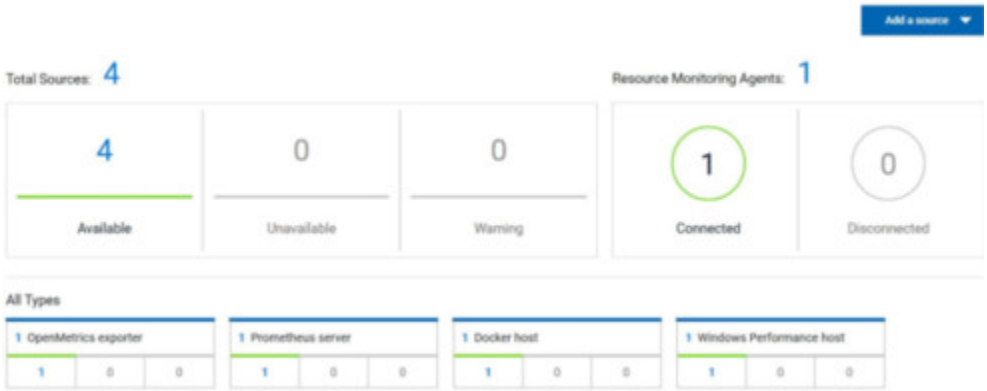
**Example**

On Linux:

```
sudo RTCP_OFFLINE_TOKEN=(Enter your offline token here) java -jar (Enter the name of the
 downloaded jar file here) --ServiceUrl=portNumber/rm --projectId=<project_id>
```

**Results**

When the agent is started, the agent is connected to IBM® Rational® Test Automation Server. The agent is displayed in the list of connected Resource Monitoring agents with the host name and status of the agent. The agent is also added to the main page of the Resource Monitoring service in IBM® Rational® Test Automation Server.




**What to do next**

You can add a new Resource Monitoring source. It can be collected from a service or from the named agents, depending on the capabilities supported by the environment.

## Starting a Resource Monitoring agent as a service on Windows

To ensure that the Resource Monitoring agent starts by itself when the host is running, you can set up the environment in such a way that the Resource Monitoring agent can be started as a service.

**Before you begin**

You must have completed the following tasks:

- Installed Java 8 on the host.
- Added IBM® Rational® Test Automation Server to the PATH environment variable.
- Created an offline token to connect the agent securely with appropriate permissions.

  **Note:** You can create an offline token from the User menu of the Resource Monitoring page or you can re-use your active offline token. The token expires if it is not used for a month.

  To allow remote access to the performance data on Windows 10, you must have considered the following information:

  - The user name and password are the same on the local and remote servers. Otherwise you must have provided remote server credentials.
  - The remote user must be a member of the Performance Monitor Users group (start `lusrmgr.msc` and add the user to this group).
  - The Remote Registry service must be running on the remote host (start `services.msc`) and must have verified the Remote Registry status).
  - File and printer sharing must be enabled on the Network Interface of the remote host that is implied in the communication with the local host.
  - You must have activated the following Windows Firewall rules so that the remote firewall does not block the access:
    - File and Printer Sharing (NB-Name-In)
    - File and Printer Sharing (NB-Session-In)
    - File and Printer Sharing (*)

1. Download the latest release of *winsw*.
   You can choose to download `WinSW.Net2.exe` or `WinSW.Net4.exe` depending on the version of .Net framework that you already have on the host Windows.
2. Create a folder on your local hard drive like `RMAgent-winservice`.
3. Copy the downloaded executable to this new folder and rename the file to `RMAgent-winservice.exe`.
4. Create a new text file in the same folder and name it `RMAgent-winservice.xml`.
5. Copy and adapt the following content to this new RMAgent-winservice.xml file to set up the offline token:

```
<service>
      <id>RMAgent-winservice</id>
      <name>Resource Monitoring Agent</name>
      <description>This service runs the Resource Monitoring Agent.</description>
      <executable>java</executable>
      <env name="RTCP_OFFLINE_TOKEN" value="(Enter your offline token here"/>
      <arguments>-jar %BASE%\RMAgent.jar --serviceUrl=https://<service-host>/rm
 --projectId=<projet-id>
 --autoUpgradeDownloadThen=execute:cmd,/c,start,%BASE%\auto-upgrade.bat</arguments>
      <onfailure action="restart" delay="10 sec"/>
      <logmode>rotate</logmode>
</service>
```

**Notes:**

- Replace *<service-host>* by the host name of the host that runs IBM® Rational® Test Automation Server.
- Replace *<project-id>* which is the number that you find after /projects/ in the browser's URL when you browse to this project.

6. Create a new text file in the same folder and name it `auto-upgrade.bat`.
7. Copy the following content to this new file to upgrade the agent .jar file automatically:

```
@echo off
for /f "tokens=*" %%a in ('dir /b /od %BASE%\com.hcl.test.rm.agent-*.jar') do set newest=%%a
%BASE%\RMAgent-winservice.exe stop
del %BASE%\RMAgent.jar
mklink %BASE%\RMAgent.jar %BASE%\%newest%
%BASE%\RMAgent-winservice.exe start
```

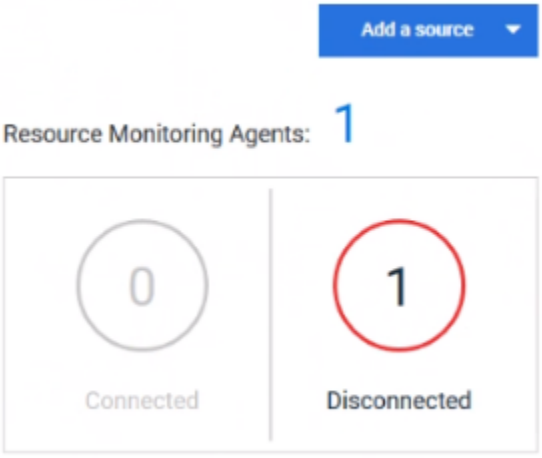
8. Start the command prompt as an administrator and change the directory to the newly created directory: `RMAgent-winservice`.
9. Download the agent .jar file that is available from the agents page in the Resource Monitoring service, under the **Extend the Resource Monitoring service with agents** section and save it to the same directory.
10. Enter the following command in the command prompt to create a symbolic link named `RMAgent.jar` to the agent jar file:

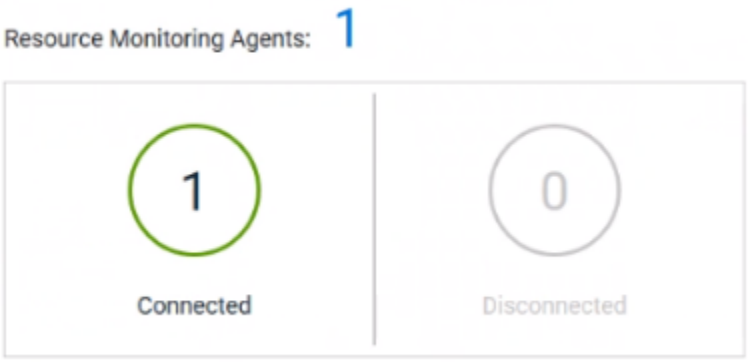
```
mklink RMAgent.jar com.hcl.test.rm.agent-<version-and-datetime>.jar
```

11. Enter the following commands:

```
RMAgent-winservice install
```

```
RMAgent-winservice start
```






**Notes:**

In the Services Windows application, you can check whether the service is up and running. You can see in the Resource Monitoring page whether the Resource Monitoring agent is disconnected or connected.

The Resource Monitoring agents write the logs to the same folder in files named `RMAgent-winservice.out.log`, `RMAgent-winservice.err.log` and `RMAgent-winservice.wrapper.log`. If the Windows service for the agent is not started or if it is not connected to the Resource Monitoring service, verify the log files.

**Results**

The Windows agent is added to the main page of the Resource Monitoring service in IBM® Rational® Test Automation Server.

**What to do next**

You can monitor the Windows agent by selecting counters. See Monitoring host resources on page 215.

Related information

Other commands

winsw file

## Starting a Resource Monitoring agent as a service on Linux

To ensure that the Resource Monitoring agent starts by itself when the host is restarted, you can set up the environment in such a way that the Resource Monitoring agent can be started as a service.

**Before you begin**

You must have completed the following tasks:

- Installed Java 8 on the host.
- Added IBM® Rational® Test Automation Server to the PATH environment variable.
- Created an offline token to connect the agent securely with appropriate permissions.

**Note:** You can create an offline token from the User menu of the Resource Monitoring page or you can re-use your active offline token. The token expires if it is not used for a month.

**About this task**

This topic relies on systemd services that are the default on most modern Linux distributions. Other ways mi require adaptations of the instructions but the provided script will be a good basis in most cases.

1. Create a folder on your local hard drive like `/opt/RMAgent-linuxservice`.
2. Download the agent .jar file that is available from the Agents page in the Resource Monitoring service, below the **Extend the Resource Monitoring service with agents** section and save it to the same directory.
3. Create a new file `/etc/systemd/system/RMAgent-linuxservice.service`.
4. Add the following content to this new file:

```
[Unit]
Description = Resource Monitoring Agent
After = network.target

[Service]
Type = forking
ExecStart = /opt/RMAgent-linuxservice/RMAgent-linuxservice.sh start
ExecStop = /opt/RMAgent-linuxservice/RMAgent-linuxservice.sh stop
Restart = on-failure
RestartSec = 10

[Install]
WantedBy = multi-user.target
```

5. Create another file `/opt/RMAgent-linuxservice/RMAgent-linuxservice.sh`.

```
#!/bin/sh

# Update the 3 following variables with the Server's host name, project id and offline token:
SERVICE_URL=https://<hostname>/rm
PROJECT_ID=<project-id>
export RTCP_OFFLINE_TOKEN=<offline-token>

ARGS="--serviceUrl=$SERVICE_URL --projectId=$PROJECT_ID --autoUpgradeDownloadThen=exitFailure"

SCRIPT=$(readlink -f "$0")
RMAGENT_HOME=$(dirname "$SCRIPT")

# Ensure we're using the latest downloaded jar file
PATH_TO_JAR=`ls -t $RMAGENT_HOME/com.hcl.test.rm.agent-*.jar | head -1`
if [ -z "$PATH_TO_JAR" ]
then
  cd $RMAGENT_HOME && { curl -k -O -J $SERVICE_URL/agent-jar; cd -; }
  PATH_TO_JAR=`ls -t $RMAGENT_HOME/com.hcl.test.rm.agent-*.jar | head -1`
```

```
  if [ -z "$PATH_TO_JAR" ]
  then
    echo "Start the server at $SERVICE_URL to allow download of the latest agent jar file"
    echo "Exiting..."
    exit 1
  fi
fi
SERVICE_NAME="Resource Monitoring Agent"
#Pid file will reside in this script's folder
PATH_TO_PID=$RMAGENT_HOME/RMAgent-pid
#Log file will reside in this script's folder
PATH_TO_LOG=$RMAGENT_HOME/RMAgent.log

case $1 in
    start)
        echo "Starting $SERVICE_NAME ..."
        if [ ! -f $PATH_TO_PID ]; then
            nohup java -jar $PATH_TO_JAR $ARGS >> $PATH_TO_LOG 2>&1 &
                        echo $! > $PATH_TO_PID
            echo "$SERVICE_NAME started ..."
        else
            echo "$SERVICE_NAME is already running ..."
        fi
    ;;
    stop)
        if [ -f $PATH_TO_PID ]; then
            PID=$(cat $PATH_TO_PID);
            echo "$SERVICE_NAME stopping ..."
            kill $PID;
            echo "$SERVICE_NAME stopped ..."
            rm $PATH_TO_PID
        else
            echo "$SERVICE_NAME is not running ..."
        fi
    ;;
    restart)
        if [ -f $PATH_TO_PID ]; then
            PID=$(cat $PATH_TO_PID);
            echo "$SERVICE_NAME stopping ...";
            kill $PID;
            echo "$SERVICE_NAME stopped ...";
            rm $PATH_TO_PID
            echo "$SERVICE_NAME starting ..."
            nohup java -jar $PATH_TO_JAR $ARGS >> $PATH_TO_LOG 2>&1 &
                        echo $! > $PATH_TO_PID
            echo "$SERVICE_NAME started ..."
        else
            echo "$SERVICE_NAME is not running ..."
        fi
    ;;
esac
```

**Notes:**

- Replace *<service-hostname>* by the host name of the host that runs IBM® Rational® Test Automation Server.
- Replace *<project-id>* which is the number you'll find after /projects/ in the browser's URL when browsing to this project.

**Results**

The Resource Monitoring agent starts automatically when the host restarts. The Linux agent is added to the main page of the Resource Monitoring service in IBM® Rational® Test Automation Server.

**What to do next**

You can monitor the Linux Performance host source by adding counters. See Monitoring host resources on page 215.

Related information

systemd services

## Starting a Java Virtual Machine

To monitor the Resource Monitoring data from a Java Virtual Machine (JVM), you must first start the JVM. You must enter the IP address of the JVM, the IP port, and the security parameters in the command before running the Java Virtual Machine.

1. Enter The IP address of the JVM (local or remote host) and the IP port in the command that is used to run the Java Virtual Machine.
2. **Optional:** You can use the authentication security data to start the virtual machine. In this case, you must enter the name of the password file. You can also enter the name of an access file that might be needed if user privileges are required.

**Exemple**

**Parameters used to launch a JVM without security:**

```
java
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=9010
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.authenticate=false
-Dcom.sun.management.jmxremote.ssl=false
-jar MyapplicationFile.jar
```

**Parameters used to launch a JVM with authentication security:**

```
java
-Dcom.sun.management.jmxremote
-Dcom.sun.management.jmxremote.port=9010
-Dcom.sun.management.jmxremote.local.only=false
-Dcom.sun.management.jmxremote.authenticate=true
-Dcom.sun.management.jmxremote.ssl=false
```

```
-Dcom.sun.management.jmxremote.password.file=jmxremote.password
-Dcom.sun.management.jmxremote.access.file=jmxremote.access (this command is optional)
-jar MyapplicationFile.jar
```

Related information

Resource Monitoring agents on page 227

Monitoring host resources on page 215

# Integration with Jira as a defect management system

You can integrate Jira as a defect management system with your project on IBM® Rational® Test Automation Server to create defects for the tests that are available in the project after the test run is complete.

### Prerequisites

You must have created the tests and committed the test resources to a remote repository. The remote repository must be added to a project to make the tests available on Rational® Test Automation Server.

### Overview

You can search for the tests and run the tests that are available in your project on Rational® Test Automation Server. After the test run completes, you can use Jira to create a defect for the tests on Rational® Test Automation Server.

**Note:** You can create multiple defects for a test in IBM® Rational® Test Automation Server, if required.

After you create a defect, you can view the Jira defect in the **Change Management System** card on the **Results** page.

You can now follow the tasks listed in the task flow table for integrating Jira with your project, and then create a defect for a test available in that project. See Task flow for integrating Jira on page 237.

## Task flow for integrating Jira

You must perform these tasks in sequence as listed in the following table. The table also provides you the links to the information about the tasks.

| | Tasks | More information |
|---|---|---|
| 1 | Install IBM® Rational® Test Automation Server. | Installation of the server software on page 25 |
| 2 | Create the type of tests that are supported for running on Rational® Test Automation Server. | Tests supported on Rational Test Automation Server on page 18 |

| | Tasks | More information |
|---|---|---|
| 3 | Create a project on Rational® Test Automation Server and add repositories to the project to access the test resources available in the respective repository. | Test assets and a server project on page 280 |
| 4 | Configure Jira to your project on Rational® Test Automation Server where the test resources are available.<br><br>**Note:** You must have the project *Owner* access to configure Jira. | Configuration of Jira |
| 5 | Create defects for the tests available in your project. | Creating Jira defects on page 241 |

Related information

Integration with Jira as a defect management system on page 237

## Configuring the Jira server

After the administrator of the Jira server sets up the application link in Jira by using both the public and consumer keys, you can configure the Jira server for your project on IBM® Rational® Test Automation Server where the tests are available.

**Before you begin**

You must have completed the following tasks:

- Generated a public key and a consumer key on Rational® Test Automation Server. See Generating public and consumer keys on page 240.
- Shared both the keys with the administrator of the Jira server to set up an application link in Jira.

**About this task**

When you share both the keys with the administrator of the Jira server, the administrator of the Jira server sets up the application link in Jira.

After the completion of setting up of the application link by the administrator of the Jira server, you can then open the **Change Management System** tab of your project and complete the authorization process to configure the Jira server for your project on Rational® Test Automation Server.

**Tasks performed by the administrator of the Jira server**

As an administrator of the Jira server, you must perform the following actions to set up an application link on the Jira server:

1. Log in to the Jira server and open the **Configure Application Links** page.
2. Enter the URL of the application, and then click **Create new link**.
3. Verify the URL on the **Configure Application URL** page, and then click **Continue** to open the **Link applications** page.
4. Enter the information for the fields specified as follows on the **Link applications** page, and then click **Continue**:
    - **Application Name**: Enter an application name. The application name must be unique. For example, `IBM`.
    - **Application Type**: Select the type of application from the **Application Type** drop-down list. The default selection is **General Application**.
    - Select the **Create incoming link** check box.

    **Note:** You need not set other fields when you set up an application link to establish a connection between the Jira server and Rational® Test Automation Server.

5. Enter the information on the **Link applications** page for the fields specified as follows, and then click **Continue** to set up the application link:
    - **Consumer Key**: Enter the consumer key shared by the administrator of Rational® Test Automation Server.
    - **Consumer Name**: Enter an application name. For example, `IBM_1`.
    - **Public Key**: Enter the public key shared by the administrator of Rational® Test Automation Server.

You have successfully set up the application link on the Jira server to establish a connection between the Jira server and Rational® Test Automation Server.

To know more about the application link, refer to the Configuring the client application as a consumer in Jira section.

1. Open your project.
2. Expand **Configure**, and then click **Project**.
3. Open the **Change Management System** tab.
4. Click the **Open action menu** button from **Jira Integration Details**, and then click **Configure**.
   **Result**
   The **Authorize User** page is displayed.
5. Complete the authorization process on the **Authorize User** page by performing the following actions:
    a. Click the authorization URL of the Jira server to open the **Jira authorization** page.
    b. Log in to your Jira server account with your valid credentials.
    c. Click **Allow** to authorize the Jira server account to generate an access token.
    **Result**

    The authorization is successful.

The **Access Approved** page is displayed with a verification code.

d. Copy the verification code, and then paste the verification code in the **Enter Verification Code** field on the **Authorize User** page.

e. Click **Save**.

**Result**

The **Configure Change Management System** page is displayed.

6. Select a Jira server project that is available from the **Project** drop-down list, and then click **Update**.

**Note:** The change management system name and the change management system link are already displayed on the **Configure Change Management System** page.

**Results**

You have successfully configured the Jira server for your project on Rational® Test Automation Server.

You can see a tick icon next to the URL of the Jira server on the successful configuration. You can also expand the **Jira Integration Details** section to see the project name of the Jira that you selected.

**Note:** You can delete the URL of the Jira server that you no longer require in your test environment.

**What to do next**

You can create defects for the tests available in your project and track the defects from the **Results** page on Rational® Test Automation Server after the test run is complete. See Creating Jira defects on page 241.

## Generating public and consumer keys

You must first generate a public key and a consumer key for your project that is available on IBM® Rational® Test Automation Server to configure Jira.

**Before you begin**

You must have completed the following tasks:

- You must have installed Jira on your computer. For more information about installing Jira, refer to Installing Jira software.
- You must be able to access the Jira server from Rational® Test Automation Server.

**About this task**

After you add the URL of the Jira server to your project, you can generate both the public key and the consumer key from Rational® Test Automation Server. You can then share both the keys with the administrator of Jira server.

1. Log in to Rational® Test Automation Server, if you are not already logged in.
2. Open your project.
3. Expand **Configure**, and then click **Project**.
4. Click the **Change Management System** tab, and then click **New System**.
   **Result**
   The **Configure Change Management System** page is displayed.
5. Enter the URL of Jira server in the **Change Management System** link field, and then click **Add**.
   **Result**
   A public key and a consumer key are generated and displayed on the **Configure Tracking System** page.

**Results**

You have successfully generated both the public and consumer keys.

**What to do next**

You can then click **Copy** to copy both the keys and save the keys in your local file. You can then share both the keys with the administrator of the Jira server to setup an application link in Jira.

The administrator of the Jira server then sets up the application link in Jira by using both the keys that you shared. After the application link set up is complete, you can then configure the Jira server for your project on IBM® Rational® Test Automation Server. See Configuring the Jira server on page 238.

## Creating Jira defects

You can create a Jira defect for a test available in your project and track the defect from IBM® Rational® Test Automation Server after the test run is complete.

**Before you begin**

You must have completed the following tasks:

- Been a member of the project with the *Owner* or *Tester* role to create a defect for a test.
- Configured Jira in Rational® Test Automation Server for your project. See Configuration of Jira.

  **Note:** You must have the project *Owner* access to configure Jira.

- Initiated a test run of a test available in your branch from the **Execution** page.

**About this task**

After you configure Jira for your project on Rational® Test Automation Server, you can create a Jira defect for a test available in your project and track the defect from the **Results** page after the test run is complete.

**Note:** You can create a Jira defect for a test result from the **Results** page only if you have executed the test on Rational® Test Automation Server.

1. Open your project on Rational® Test Automation Server.
2. Run a test from the **Execution** page.
3. Click **Results** from the left pane to open the **Results** page.
4. Select the test result from the **Results** page.
5. Click the **JIRA: Raise Defect** icon from **Actions**.

   **Important:** You can create a Jira defect for a test result only if Jira is configured to your server project. If Jira is not configured with your server project and you click the **Open action menu** button from **Actions** on the **Results** page for a test result, you can see a message is displayed on the **Results** page that states that there is no change management system configured for the project. You must then contact your project *Owner* to configure Jira for the project. You project *Owner* can then click the link provided in the message to open the project to configure Jira for your project.

   **Result**

   The **Raise defect in JIRA** page is displayed with the fields available for your project.
6. Enter the information required for all the fields, and then click **Create**.

   **Result**

   A Jira defect is created.
7. Expand the test result to see the Jira tracking ID details from the **Results** page.

   **Note:** You can raise multiple defects for a test, if required.

**Results**

You have created a defect for a test result from the **Results** page on Rational® Test Automation Server.

**What to do next**

You can now click the Jira tracking ID from the **Results** page to open the defect and view the details. The **Reporter** field on the Jira server page displays the name of the person who has created the defect. To know more about Jira server defects and how it works, refer to the documentation.

---

Related information

Integration with Jira as a defect management system on page 237

# Integrating with other applications

You can integrate certain applications with IBM® Rational® Test Automation Server to run tests, view the test results, and create defects.

## Integration plugin compatibility matrix

You can find information about the versions of the integration plugin that are compatible with IBM® Rational® Test Automation Server.

The following table lists the versions of the integration plugin that are required to integrate IBM® Engineering Test Management, Jenkins, and UrbanCode™ Deploy with Rational® Test Automation Server.

**Note:** You must download the required version of the integration plugin from the IBM WebSphere, Liberty & DevOps Community portal based on the existing version of Rational® Test Automation Server. You can then integrate Engineering Test Management, Jenkins, and UrbanCode™ Deploy with Rational® Test Automation Server.

| **Rational® Test Automation Server** | **10.1.0** | **10.1.1** | **10.1.2** |
|---|---|---|---|
| Jenkins plugin | RTAS-Jenkins-2.0 | RTAS-Jenkins-3.0 | RTAS-Jenkins-4.0 |
| UrbanCode™ Deploy plugin | RTAS-UCD-1.0 | RTAS-UCD-1.0 | RTAS-UCD-2.0 |

## Integration with Azure DevOps

When you use Azure DevOps for continuous integration and continuous development of your application, you can run tests created for your application and available in a project on IBM® Rational® Test Automation Server, in Azure DevOps pipelines by using the *IBM Rational Test Workbench* extension.

### Overview

You can use the *IBM Rational Test Workbench* extension to integrate Rational® Test Automation Server with Azure DevOps.

The *IBM Rational Test Workbench* extension enables you to select any type of test available for your project in Rational® Test Automation Server that you can add to your task for the job in the Azure DevOps pipelines.

You must have created the tests in the desktop clients and committed the test assets and test resources to a remote repository. The remote repository must be added to the project.

Depending on the type of tests you want to perform on your application, you must have created any or all of the following types of tests in the desktop clients for the application you are testing:

| Type of test | Desktop client |
|---|---|
| • Accelerated Functional Testing suites<br>• Compound tests | IBM® Rational® Functional Tester |
| • Test Suite | IBM® Rational® Integration Tester |
| • Compound tests<br>• VU Schedule<br>• Rate Schedule | IBM® Rational® Performance Tester |

You can now follow the tasks listed in the task flow table to integrate Rational® Test Automation Server with Azure DevOps. See Task flow for integrating Azure DevOps on page 244.

## Task flow for integrating IBM® Rational® Test Automation Server with Azure DevOps

The table shows the task flow for integrating Rational® Test Automation Server with Azure DevOps by using the *IBM Rational Test Workbench* extension. You must perform these tasks in sequence as listed in the following table. The table also provides you the links to the information about the tasks.

| | Tasks | More information |
|---|---|---|
| 1 | Install Rational® Test Automation Server. | Installation of the server software on page 25 |
| 2 | Create tests in the desktop clients for the application you are testing. | • Rational® Performance Tester documentation<br>• Rational® Functional Tester documentation<br>• Rational® Integration Tester documentation |
| 3 | Create a project on Rational® Test Automation Server and add the test assets that are created in the desktop clients to your project. | Test assets and a server project on page 280 |
| 4 | Create an organization and a project in Azure DevOps for running jobs in Azure DevOps pipelines. | Creating an organization |

| | Tasks | More information |
|---|---|---|
| 5 | Access the **Visual Studio Marketplace** portal and search for the latest version of the *IBM Rational Test Workbench* extension. | Visual Studio Marketplace |
| 6 | Install the latest version of the *IBM Rational Test Workbench* extension. | Installing the *IBM Rational Test Workbench* extension on page 245 |
| 7 | Extend the trust certificate if you have used an internal CA certificate. | Extending the trusted CA list |
| 8 | Run tests in an Azure DevOps pipeline. | Running tests in an Azure DevOps Pipeline on page 246 |

Related information

Integration with Azure DevOps on page 243

## Installing the *IBM Rational Test Workbench* extension

You must install the latest version of the *IBM Rational Test Workbench* extension in your Azure DevOps organization before you can use the extension for running your application tests in an Azure DevOps pipeline.

**Before you begin**

You must have access to the **Visual Studio Marketplace** portal to install the latest version of the *IBM Rational Test Workbench* extension.

**About this task**

After you install the latest version of the *IBM Rational Test Workbench* extension from the **Visual Studio Marketplace** portal in your Azure DevOps organization, you can select the tests that you want to run for your application in an Azure DevOps pipeline by using the *IBM Rational Test Workbench* extension.

1. Log in to the **Visual Studio Marketplace** portal, if you are not already logged in.
2. Click the **Azure DevOps** tab.
3. Search for the *IBM Rational Test Workbench* extension.
4. Click the *IBM Rational Test Workbench* extension.
5. Click **Get it free**.

   **Result**

   The **Visual Studio Marketplace** portal for the *IBM Rational Test Workbench* extension is displayed.
6. Select the organization where you want to run your test from the **Select an Azure DevOps Organization** list.
7. Click **Install**.

   **Result**

The installation is completed.

8. Click **Proceed to organization**.
   **Result**

   The **Organization** page in Azure DevOps is displayed.

9. Click **Organization settings > Extensions**.
   **Result**

   The *IBM Rational Test Workbench* extension is displayed as an installed extension.

**Results**

You have installed the *IBM Rational Test Workbench* extension in your Azure DevOps organization.

**What to do next**

You can run tests that are available in Rational® Test Automation Server as a job in an Azure DevOps pipeline. See Running tests in an Azure DevOps Pipeline on page 246.

## Running IBM® Rational® Test Automation Server tests in an Azure DevOps Pipeline

After you install the *IBM Rational Test Workbench* extension in your organization, you can run tests that are available in IBM® Rational® Test Automation Server as a job in Azure DevOps pipelines.

**Before you begin**

You must have completed the following tasks:

- Added the tests that you created in the desktop clients for your application to the project on Rational® Test Automation Server.
- Installed the latest version of the *IBM Rational Test Workbench* extension in your organization. See Installing the *IBM Rational Test Workbench* extension on page 245.
- Installed an agent in your pipeline. See Azure Pipelines agents.
- Generated an offline user token from Rational® Test Automation Server. See Generating an offline token on page 277.

**About this task**

After you add the *IBM Rational Test Workbench* extension in your Azure DevOps organization, you can use an existing pipeline or create a new one to add Rational® Test Automation Server test tasks. You can install an agent or use the one that you installed in your default agent pool. You can add Rational® Test Automation Server tests as tasks to your agent job, configure the task, and then run the task in the Azure DevOps pipeline.

1. Open your **Organization** page in Azure DevOps and perform the following steps:

   a. Click the project you want to use.

   b. Initialize the repository by performing the following steps:

i. Click **Repos** from the left pane.
ii. Click **Initialize** from the **Initialize with a README or gitignore** section.

**Note:** Select the **Add a README** checkbox if it is not selected.

c. Click **Pipelines** from the left pane.

d. Click **Create Pipeline**.

e. Click **Use the classic editor** to create a pipeline without YAML.

f. Verify the project, repository, and branch for manual and scheduled builds, and then click **Continue**.

g. Click **Empty job**.

2. Select **Pipeline** and complete the following steps:

a. Change the name for the build pipeline if required.

b. Select the **Agent pool** for your build pipeline.

You can use the agent from the default agent pool or use the one you have installed.

c. Select the **Agent Specification** for the agent if required.

3. Add a task to the agent job by completing the following steps:

a. Click the **Add Task** icon for the agent job.
**Result**

The **Add tasks** pane is displayed.

b. Search for the IBM tasks defined in the *IBM Rational Test Workbench* extension.
**Result**

The tasks that you can select are displayed.

c. Select the **IBM® Rational® Test Automation Server Task**, and then click **Add** to add the task to the agent job.

**Result**

The selected task is added to the agent job and it is displayed with a warning that some settings require attention. You must configure the settings mentioned in step 4 on page 248.

You can also remove the tasks that are not required in your job. Select the tasks in the list that you want to remove. You can then right-click the tasks, and click **Remove selected task(s)** to remove them.

4. Configure the settings by performing the following steps:

   a. Select the task version from the list if required.

   b. Follow the action for the task by referring to the following table:

   **Note:** All mandatory fields are marked with an asterisk (*) in the UI.

| Field | Action |
|---|---|
| Display name | Enter a name of the task. |
| IBM® Rational® Test Automation Server service connection | Select the service connection from the drop-down list.<br><br>If you are selecting the IBM® Rational® Test Automation Server service connection for the first time, you must click **New** to add the following details to add an IBM® Rational® Test Automation Server service connection, and then save the connection details:<br>▪ **Server URL** - Enter the URL of Rational® Test Automation Server. The format of the URL is as follows: `https://hostname`.<br>▪ **Offline Token** - Enter the offline token that you generated from Rational® Test Automation Server.<br>▪ **Service connection name** - Enter a service connection name.<br>▪ **Description (optional)** - Enter the details of the service connection if required.<br>▪ Optionally, select the **Grant access permission to all pipelines** checkbox if required.<br>You can save the service connection details. The service connection is available for selection from the **IBM® Rational® Test Automation Server service connection** drop-down list.<br><br>**Note:**<br><br>You can edit or delete the service connection that you added if required. Click **Manage**, and then select the service connection from the **Service connections** list to open the service connection. You can click **Edit** to edit the service connection |

| Field | Action |
|---|---|
| | details. If you want to delete an existing service connection details, click the **Vertical ellipsis** icon, and then click **Delete** to delete the service connection details.<br><br>Optionally, you can also create, edit, and delete a service connection from your Azure DevOps project dashboard. Open your project, click **Project settings > Service connections**, and then perform any of the following tasks:<br>▪ Create a new service connection.<br>▪ Edit or delete an existing service connection that you already added. |
| Project | Enter the name of the project containing the test. |
| Branch | Enter the branch where the test assets are stored. |
| Repository Link | Enter the repository path that is configured for the project that contains the test to run. |
| File Path | Enter the path of the test in Rational® Test Automation Server that you want to run. |
| API Test Environment | Enter the name of the environment that is configured in the test asset for the test created in Rational® Integration Tester.<br><br>**Note:** This setting is applicable only if you are running an API suite test. |

c. Expand **Control Options** and configure the settings for your task if required.

d. Expand **Output Variables** and configure the settings for your task if required.

5. Select from the following options:

a. Click **Save** to save the configured settings for the task.

**Note:** The task is not queued for a run.

You can save the task to a build pipeline and opt to run the build at a later time.

b. Click **Save & queue** to save the configurations and queue the run in the pipeline.
**Result**

The **Run pipeline** dialog box is displayed.

6. Complete the following steps:

a. Enter a comment for the test in the **Save comment** field.

b. Select the agent that you configured for the test from the **Agent pool** list.

c. Select the agent specification from the **Agent Specification** list if required.

d. Select the branch from the **Branch/tag** list.

e. Add the variables and demands for the task run from the **Advanced Options** pane if required.

f. Select the **Enable system diagnostics** checkbox for a detailed log view.

g. Click **Save and run**.
**Result**

The `pipeline summary` page displays the progress of the job run.

**Results**

You have run the tests for the application you are testing, in the Azure DevOps pipeline.

**What to do next**

You can open the job to view the task logs from the `pipeline summary` page.

You must click the task to open the **Task** page to view the test results. The **Reports information** section on the **Task** page displays the names of the report along with its corresponding URLs. The report URLs are the Rational® Test Automation Server URLs where the reports are stored. You can access the report URLs to view the test execution information at any point of time.

You can also view the test reports and logs of the test that was run on the Azure DevOps from the **Results** page on Rational® Test Automation Server. See Test results and reports overview on page 200.

## Integration with Jenkins

When you integrate IBM® Rational® Test Automation Server with Jenkins, you can run tests created for your application and available in a project on Rational® Test Automation Server from a Jenkins server by using the Rational® Test Automation Server Jenkins plugin.

### Prerequisites

You must have installed Jenkins on your computer. For more information about installing Jenkins, refer to Installing Jenkins.

You can then run the following command to start the Jenkins server to support UTF-8 character sets:

```
java -Dfile.encoding=UTF8 -jar jenkins.war
```

### Overview

You can use the Rational® Test Automation Server Jenkins plugin to integrate Rational® Test Automation Server with Jenkins.

You must have created the tests in the desktop clients and committed the test assets and test resources to a remote repository. The remote repository must be added to the project.

Depending on the type of tests you want to perform on your application, you must have created any or all of the following types of tests in the desktop clients for the application you are testing:

| Type of test | Desktop client |
|---|---|
| • Accelerated Functional Testing suites<br>• Compound tests | IBM® Rational® Functional Tester |
| • Test Suite | IBM® Rational® Integration Tester |
| • Compound tests<br>• VU Schedule<br>• Rate Schedule | IBM® Rational® Performance Tester |

You can now follow the tasks listed in the task flow table to integrate Rational® Test Automation Server with Jenkins. See Task flow for integrating Jenkins on page 251.

## Task flow for integrating IBM® Rational® Test Automation Server with Jenkins

The table shows the task flow for integrating IBM® Rational® Test Automation Server with Jenkins by using the Rational® Test Automation Server Jenkins plugin. You must perform these tasks in sequence as listed in the following table. The table also provides you the links to the information about the tasks.

| | Tasks | More information |
|---|---|---|
| 1 | Install Rational® Test Automation Server. | Installation of the server software on page 25 |

| | Tasks | More information |
|---|---|---|
| 2 | Create tests in the desktop clients for the application you are testing. | • Rational® Performance Tester documentation<br>• Rational® Functional Tester documentation<br>• Rational® Integration Tester documentation |
| 3 | Create a project on Rational® Test Automation Server and add the test assets that are created in the desktop clients to your project. | Test assets and a server project on page 280 |
| 4 | Install the latest version of the Rational® Test Automation Server Jenkins plugin form Test Workbench Resource Community | Installing the Rational® Test Automation Server Jenkins plugin on page 252 |
| 5 | Use a custom trust store if you have used an internal CA certificate. | Using a custom trust store on page 253 |
| 6 | Run Rational® Test Automation Server tests on the Jenkins server. | Running Rational® Test Automation Server tests on the Jenkins server on page 254 |

Related information

Integration with Jenkins on page 250

## Installing the Rational® Test Automation Server Jenkins Plugin

You must install the latest version of the Rational® Test Automation Server plugin on the Jenkins server before you can use the plugin for running tests that are available in your IBM® Rational® Test Automation Server project on the Jenkins server.

**Before you begin**

You must have downloaded the Rational® Test Automation Server Jenkins plugin `RTAS-Jenkins-4.0` from the IBM WebSphere, Liberty & DevOps Community portal.

1. Open the Jenkins dashboard.
2. Click **Manage Jenkins > Manage Plugins**.
3. Click **Advanced**.
4. Click **Choose File** to locate and **Open** the Rational® Test Automation Server Jenkins plugin file from the **Upload Plugin** section.
5. Click **Upload**.

**Result**

The Rational® Test Automation Server Jenkins plugin is displayed in the **Installed** tab.

**What to do next**

You can run the tests that are available in Rational® Test Automation Server on the Jenkins server. See Running Rational® Test Automation Server tests on the Jenkins server on page 254.

## Using a custom trust store

You can use a custom trust store in the Jenkins build step of a Rational® Test Automation Server Jenkins plugin to establish a trusted and secure connection between the Jenkins server and Rational® Test Automation Server.

**Before you begin**

You must have configured the certificate that is used by Rational® Test Automation Server as a trusted CA, and then install Rational® Test Automation Server. See Installation of the server software on page 25.

**About this task**

If the SSL certificate assigned to Rational® Test Automation Server is signed by an internal Certified Authority (CA), then you must download and import the CA certificate to a custom trust store. You can then use the custom trust store in the Jenkins plugin build step to establish a trusted and secure connection between the Jenkins server and Rational® Test Automation Server.

**Note:** If the internal CA certificate is already imported to the default trust store that is used by the Jenkins server, you need not use a custom trust store.

**Restriction:**

When you use Red Hat Enterprise Linux (RHEL) operating systems for Jenkins, you must run the Jenkins service with a user who has access to the custom trust store path to utilize the custom trust store feature. To change the Jenkins user, you must open the `/etc/sysconfig/jenkins` file and set the JENKINS_USER to the user who has access to the custom trust store path.

You can then run the following commands to set JENKINS_USER to another user who has access to the path of the custom trust store:

```
$JENKINS_USER= <username>
```

For example, `$JENKINS_USER= <user1>`

**Note:** You must ensure that the user account is available in the `/etc/passwd` file.

You can then run the following commands to change the ownership of the Jenkins folder:

```
chown -R username:username /var/lib/jenkins
```

```
chown -R username:username /var/cache/jenkins
```

```
chown -R username:username /var/log/jenkins
```

For example,

```
chown -R user1:user1 /var/lib/jenkins
```

```
chown -R user1:user1 /var/cache/jenkins
```

```
chown -R user1:user1 /var/log/jenkins
```

After the change of ownership is complete, run the following command to restart the Jenkins server:

```
/etc/init.d/jenkins restart
```

1. Locate the default trust store file (`cacerts` file) in your JRE directory from your computer, and then copy the file to a location of your choice on your computer.
2. Run the following command from the command prompt or terminal to import the CA certificate to your custom trust store:

   ```
   keytool -import -trustcacerts -file <path to the downloaded CA certificate with the file
    extension> -alias <custom label for the certificate> -keystore <path to the trust store>
   ```

   For example,

   ```
   keytool -import -trustcacerts -file C:\Users\ca file.crt -alias alias1 -keystore D:\cert\cacerts
   ```

   **Note:** The default password of the trust store is *changeit*. It remains the same for the custom trust store. If you want to change the password, you can run the following command, and then enter the new password:

   ```
   keytool -storepasswd -keystore <path to the trust store>
   ```

   For example, `keytool -storepasswd -keystore D:\cert\cacerts`

**Results**

You have successfully imported the downloaded CA certificate to the custom trust store.

**What to do next**

You can add the Rational® Test Automation Server tests to the Jenkins build step, and then run the tests from the Jenkins server.

## Running IBM® Rational® Test Automation Server tests on the Jenkins server

After you install the Rational® Test Automation Server Jenkins plugin on the Jenkins server, you can run tests that are available in your IBM® Rational® Test Automation Server project on the Jenkins server.

**Before you begin**

You must have completed the following tasks:

- Added the tests that you created in the desktop clients for your application to the project on Rational® Test Automation Server.
- Install the latest version of the Rational® Test Automation Server Jenkins plugin on the Jenkins server. See Installing the IBM® Rational® Test Automation Server Jenkins Plugin on page 252.
- Generated an offline user token from Rational® Test Automation Server. See Generating an offline token on page 277.
- Created a Jenkins free-style software project. See Building a software project in Jenkins.

1. Login to the Jenkins server, if you are not already logged in.
2. Open your Jenkins free-style software project.
3. Click **Configure**.
4. Click the **Build** tab.
5. Click **Add build step**, and then click **Run IBM® Rational® Test Automation Server test**.
   **Result**

   The **Run IBM® Rational® Test Automation Server test** pane is displayed.

   a. Provide details about the test run by referring to the following table.

| Field | Description |
|---|---|
| Name | Enter a name for the Jenkins build step. |
| Use Custom Trust Store | Select the **Use Custom Trust Store** checkbox if you have used an internal CA certificate and have imported the certificate to the custom trust store.<br><br>You can then enter the file path of your trust store followed by the file name in the **Custom Trust Store Path** field.<br><br>You can select the **Use Custom Password for the Trust Store** checkbox if you have modified the trust store password. Enter the new password.<br><br>**Note:** If you have not modified the trust store password, you can keep the **Use Custom Password for the Trust Store** checkbox unselected. |
| Server URL | Enter the URL of Rational® Test Automation Server. The format of the URL is as follows: *https://hostname*. |
| Offline Token | Enter the offline user token that you generated from Rational® Test Automation Server. |
| Project | Select a project from the **Project** list. The **Project** list displays the projects where you are the owner or member of Rational® Test Automation Server. |

| Field | Description |
|---|---|
| Branch | Select the branch from the **Branch** list. The **Branch** list displays the branches available in the corresponding project of Rational® Test Automation Server.<br><br>**Note:** After you select the branch from the **Branch** list, if you want to change the URL, offline user token, or the project in the **Build**, then you must close the build step. Then you must click **Add build step**, select **Run IBM® Rational® Test Automation Server test**, and then enter the details. |
| Asset Type | Select the test asset type of the test that you want to run from the **Asset Type** list. The available asset types are as follows:<br>▪ AFTSUITE<br>▪ COMPOUND<br>▪ VUSCHEDULE<br>▪ RATESCHEDULE<br>▪ APISUITE<br><br>**Note:** The test environment is mandatory if you select **APISUITE** as an asset type to run an API suite. |
| Test | Select the required test from the **Tests** list. The **Tests** list displays the available test assets from the corresponding branch in the selected project, test asset path, and the repository (that the test belongs to) from Rational® Test Automation Server based on the type of the test asset you selected from the **Asset Type** list.<br><br>**Note:** After you select the test details from the **Tests** list, if you want to change the URL, offline user token, or the project in the **Build**, then you must close the build step. You must then click **Add build step**, select **Run IBM® Rational® Test Automation Server test**, and then enter the details. |
| Test Environment | This field is mandatory only if you are running an API suite test. Based on the asset type as **APISUITE** and test from the **Tests** list that you select, the **Test Environment** list displays the available test environments from Rational® Test Automation Server. |

| Field | Description |
|---|---|
| | **Note:** The following message is displayed when you select any other asset type apart from **APISUITE** in the **Asset Type** field: `You can select Test Environment only if you are running tests of type APISUITE.` |

b. Click **Save** to save the build step.
c. Optionally, you can run multiple tests sequentially in the same job by adding multiple build steps and provide details for the tests that you want to run.

6. Click **Build Now** from the left pane to run the test on the Jenkins server.

**Results**

The Jenkins dashboard shows the progress of the test run.

**What to do next**

After the Jenkins build completes, you can view the test results. You can click the build number from the **Build History** pane on the Jenkins dashboard. You can then click **Console Output** to view a detailed log of the build from the console output.

The `Test Result` displays the status of the completed test that you ran.

**Note:** If you add multiple build steps to run multiple tests, multiple `Test Result` instances are displayed.

The **Reports information** section displays the names of the report along with its corresponding URLs. The report URLs are the Rational® Test Automation Server URLs where the reports are stored. You can access the report URLs to view the test execution information at any point of time.

You can also view the test reports and logs of the test that was run on the Jenkins server from the **Results** page on IBM® Rational® Test Automation Server. See Test results and reports overview on page 200.

## Integration with UrbanCode Deploy

When you use UrbanCode™ Deploy (UCD) for automating application deployments of your application, you can run tests created for your application and available in a project on IBM® Rational® Test Automation Server from UCD by using the IBM® Rational® Test Automation Server UCD plugin.

### Overview

You can use the IBM® Rational® Test Automation Server UCD plugin to integrate Rational® Test Automation Server with UCD.

You must have created the tests in the desktop clients and committed the test assets and test resources to a remote repository. The remote repository must be added to the project.

Depending on the type of tests you want to perform on your application, you must have created any or all of the following types of tests in the desktop clients for the application you are testing:

| Type of test | Desktop client |
|---|---|
| • Accelerated Functional Testing suites<br>• Compound tests | IBM® Rational® Functional Tester |
| • Test Suite | IBM® Rational® Integration Tester |
| • Compound tests<br>• VU Schedule<br>• Rate Schedule | IBM® Rational® Performance Tester |

You can now follow the tasks listed in the task flow table to integrate Rational® Test Automation Server with UCD. See Task flow for integrating UrbanCode Deploy.

## Installing the IBM® Rational® Test Automation Server UCD plugin

You must install the latest version of the IBM® Rational® Test Automation Server UCD plugin to integrate IBM® Rational® Test Automation Server with UCD and run tests on the UCD server.

**Before you begin**

You must have downloaded the IBM® Rational® Test Automation Server UCD plugin `RTAS-UCD-2.0` from the IBM WebSphere, Liberty & DevOps Community portal .

1. Open the UCD dashboard.
2. Click **Settings**.
3. Click **Automation Plugins** from the **Automation** pane.
4. Click **Load Plugin**.
5. Click **Choose File** to locate and **Open** the compressed IBM® Rational® Test Automation Server UCD plugin file.

   **Note:** Do not extract the IBM® Rational® Test Automation Server UCD plugin compressed file contents.

6. Click **Submit**.

**Result**

The IBM® Rational® Test Automation Server UCD plugin is displayed in the **Automation Plugins** tab.

**What to do next**

You can run the tests for the application that are available in your Rational® Test Automation Server project on the UCD server. See Running tests on the UCD server on page 271.

## Using a custom trust store for UCD integration

You can use a custom trust store in the IBM® Rational® Test Automation Server UCD plugin file to establish a trusted and secure connection between the UCD server and Rational® Test Automation Server.

**Before you begin**

You must have configured the certificate that is used by Rational® Test Automation Server as a trusted CA, and then install Rational® Test Automation Server. See Installation of the server software on page 25.

**About this task**

If the SSL certificate assigned to Rational® Test Automation Server is signed by an internal Certified Authority (CA), then you must download and import the CA certificate to a custom trust store. You can then use the custom trust store in the **IBM® Rational® Test Automation Server test** process step to establish a trusted and secure connection between the UCD server and Rational® Test Automation Server.

**Note:** If the internal CA certificate is already imported to the default trust store that is used by the UCD server, you need not use a custom trust store.

1. Locate the default trust store file (`cacerts` file) in your JRE directory from your computer, and then copy the file to a location of your choice on your computer.
2. Run the following command from the command prompt or terminal to import the CA certificate to your custom trust store:

   ```
   keytool -import -trustcacerts -file <path to the downloaded CA certificate with the file
    extension> -alias <custom label for the certificate> -keystore <path to the trust store>
   ```

   For example,

   ```
   keytool -import -trustcacerts -file C:\Users\ca file.crt -alias alias1 -keystore D:\cert\cacerts
   ```

   **Note:** The default password of the trust store is *changeit*. It remains the same for the custom trust store. If you want to change the password, you can run the following command, and then enter the new password:

```
keytool -storepasswd -keystore <path to the trust store>
```

For example, `keytool -storepasswd -keystore D:\cert\cacerts`

**Results**

You have successfully imported the downloaded CA certificate to the custom trust store.

**What to do next**

You can then run the tests that are available in your Rational® Test Automation Server project from the UCD server.

## Creating a component in UrbanCode™ Deploy

You must create a component to include artifacts and processes. The artifacts include runnable files, images, databases, configuration instructions. Whereas the processes define the activities that components can perform.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have been granted access to UrbanCode™ Deploy.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Components**, and then click **Create Component**.
3. Enter a name for the component in the **Name** field.
4. Enter the details in the other optional fields based on your requirement, and then click **Save**.
   **Result**
   The component that you created is displayed.

**Results**

You have created the component in UrbanCode™ Deploy.

**What to do next**

You must create a process for the component in UrbanCode™ Deploy. See Creating a process in UrbanCode Deploy on page 260.

---

Related information

IBM UrbanCode Deploy Documentation

## Creating a process in UrbanCode™ Deploy

You must create a process for the component to include step properties for the test that you want to run from UrbanCode™ Deploy.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have performed the following tasks:
  - Been granted access to UrbanCode™ Deploy.
  - Created a component in UrbanCode™ Deploy. See Creating a component in UrbanCode Deploy on page 260.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Components**.
   **Result**
   A list of components that are available in UrbanCode™ Deploy is displayed.
3. Select the component from the list for which you want to create a process.
4. Click the **Processes** tab, and then click **Create Process**.
   **Result**
   The **Create Process** dialog is displayed.
5. Enter a name for the process in the **Name** field.
6. Select **Operational (No Version Needed)** from **Process Type** drop-down list.
7. Verify the **Default Working Directory** field.

   The **Default Working Directory** field defines the location that the agent uses to run the process. The default value is `${p:resource/work.dir}/${p:component.name}`.

   Where `${p:resource/work.dir}` is the default working directory for the agent and `${p:component.name}` is the name of the component.
8. Click **Save**.
   **Result**
   The process that you created is listed in the **Processes** tab and the **Design** tab for the process is displayed.

**Results**

You have created the process for the component in UrbanCode™ Deploy.

**What to do next**

You must configure the process in UrbanCode™ Deploy. See Configuring the process on page 262.

---

Related information

IBM UrbanCode Deploy Documentation

## Configuring the process

You must configure the process that you created for the component to organize the steps in the process, specify the properties of the steps, and connect them.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have performed the following tasks:
  - Been granted access to UrbanCode™ Deploy.
  - Created a component in UrbanCode™ Deploy. See Creating a component in UrbanCode Deploy on page 260.
  - Created a process for the component in UrbanCode™ Deploy. See Creating a process in UrbanCode Deploy on page 260.
  - Generated an offline user token from Rational® Test Automation Server. See Generating an offline token on page 277.
  - Added the tests assets to the project on Rational® Test Automation Server.

**About this task**

When you open any process to configure, the process is displayed in the process editor. The process editor lists the plugins and steps. The required **Start** and **Finish** steps represent the beginning and the end of the process and steps are automatically placed on the design area.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Components**.
   **Result**
   A list of components that are available in UrbanCode™ Deploy is displayed.
3. Select the component from the list in which you created the process.
4. Click the **Processes** tab.
   **Result**
   A list of processes that are available for the component is displayed.
5. Select the process from the list that you want to configure.
   **Result**
   The **Design** tab for the process is displayed.
6. Click **IBM Rational Test Workbench**, and then **IBM Rational Test Automation Server** from the left menu.
7. Drag the **Run IBM Rational Test Automation Server test** step, and then drop it into the design area.

**Note:** The selected test must be placed between **Start** and **Finish** steps.

8. Specify the properties for the selected test by performing the following steps:

   a. Click the **Edit** icon.

   **Result**

   The **Edit Properties for Run IBM Rational Test Automation Server test** dialog is displayed.

   b. Specify the properties for the selected test step by referring to the following tables:

   The following table lists the required fields that you must provide to run the test from UrbanCode™ Deploy:

| Fields | Action |
|---|---|
| **Name** | Enter the name for the test step. |
| **IBM Rational Test Automation Server URL** | Enter the URL of Rational® Test Automation Server. |
| **Offline Token** | Enter the offline token that you generated from Rational® Test Automation Server. |
| **Team Space Name** | Enter the name of the team space that contains the project.<br>**Note:** The license for the team space must be configured and you must be a member of that team space. |
| **Project Name** | Enter the name of the project that are available in the team space. |
| **Repository Link** | Enter the URL of the Git repository that you added to your project in Rational® Test Automation Server. |
| **File Path** | Enter the path of the test assets that you want to run.<br>You can find the path of the test assets from the **Execution** page in Rational® Test Automation Server |

   The following table lists the optional fields that you can provide to run the test from UrbanCode™ Deploy:

| Fields | Description |
|---|---|
| **Branch Name** | Use this field to enter the name of the branch in the Git repository where you stored test assets. |
| **Custom Trust Store Password** | Use this field to enter the trust store password if you have modified the password while creating the custom trust store.<br><br>**Note:** If you have not modified the trust store password, you can retain the **Custom Trust Store Password** field blank. |
| **Custom Trust Store Path** | Use this field to enter the file path of your trust store followed by the file name if Rational® Test Automation Server uses an internal CA certificate and you have imported the certificate to a custom trust store. |
| **Datasets** | Use this field to enter the path to the dataset if you want to replace the values of the dataset during a test run.<br><br>You must ensure that both original and new datasets are in the same workspace and have the same column names. When you enter a value for the **Datasets** field, you must also include the path to the dataset. The path must be in the following format:<br><br>`/project_name/ds_path/original_ds.csv:/project_name/ds_path/new_ds.csv`<br><br>**Note:** You can override multiple datasets that are saved in a different project by adding multiple paths to the dataset separated by a semicolon. |
| **Environment** | Use this field to enter the environment details for the following types of test assets:<br>▪ APISUITE<br>▪ APITEST<br>▪ APISTUB |
| **Labels** | Use this field to add labels to test results when the test run is complete.<br><br>You can add multiple labels to a test result separated by a comma.<br><br>For example, *label1, label2* |

| Fields | Description |
|---|---|
| | After the test run is complete, then the values provided in the **Labels** field are displayed on the **Results** page of Rational® Test Automation Server. |
| **Secrets Collection Name** | Use this field to enter the name of the Secret if you created secrets collections for your project and to use Secrets at test run time.<br>This field is mandatory only if you want to run the following types of test assets:<br>▪ APISUITE<br>▪ APITEST<br>▪ APISTUB |
| **Show Hidden Properties** | Select this checkbox if you want to use an HTTP Proxy to connect to Rational® Test Automation Server.<br>After you select the **Show Hidden Properties** checkbox, you can configure the HTTP Proxy by using the following fields:<br>▪ **HTTP Proxy Host**: Use this field to enter the hostname of the HTTP proxy that you want to connect to Rational® Test Automation Server.<br>▪ **HTTP Proxy Port**: Use this field to enter the port number where the HTTP proxy is used to listen to both HTTP and HTTPS traffic.<br>▪ **HTTP Proxy User name**: Use this field to provide the username for the HTTP proxy.<br>▪ **HTTP Proxy Password**: Use this field to enter a valid password for the user that you specified in the **HTTP Proxy User name** field.<br>**Note:** You can use the **Show Hidden Properties** field only when you want to run the following types of test assets:<br>▪ APISUITE<br>▪ APITEST<br>▪ APISTUB |
| **Start Date** | Use this field to enter the date and time in the following format for running tests at the scheduled date and time:<br>`yyyy/MM/dd/HH:mm` |

| Fields | Description |
|---|---|
| | When you enter a value in the **Start date** field, the status of the test displays as `Scheduled` on the **Overview** and **Progress** pages of Rational® Test Automation Server. |
| **Variables** | Use this field to enter the name of the variable and its value if your test requires variables during the test run time.<br>You must enter the variables in the following format:<br>`name_of_the_variable=value_of_the_varibale`<br>You can add multiple variables to the test run separated by a semicolon.<br>For example, *varname1=value1;varname2=value2* |

**Note:** In addition to optional fields, you can also use the following fields from UrbanCode™ Deploy to configure your test run:

- **Working Directory**
- **Precondition**
- **Post Processing Script**
- **Use Impersonation**
- **Auth Token Restriction**

You can accept the default values or change the values based on your requirements. For more information about these fields, see the related links.

c. Click **OK** to save the properties for the test.

9. Click **Save** in the design area.

**Results**

You have configured the process for the component in UrbanCode™ Deploy.

**What to do next**

You must create a resource in UrbanCode™ Deploy. See Creating a resource in UrbanCode Deploy on page 267.

Related information

IBM UrbanCode Deploy Documentation

Process step preconditions

Post-processing scripts

User impersonation for process steps

Restricting authentication tokens

## Creating a resource in UrbanCode™ Deploy

You must create a resource to associate agents with components that you created in UrbanCode™ Deploy.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have been granted access to UrbanCode™ Deploy.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Resources**, and then click **Create Top-Level Group**.
   **Result**
   The **Create Resource** dialog is displayed.
3. Enter a name for the resource in the **Name** field.
4. Click **Save**.

**Results**

You have created the resource in UrbanCode™ Deploy.

**What to do next**

You must configure the resource. See Configuring the resource on page 267.

---

Related information

IBM UrbanCode Deploy Documentation

## Configuring the resource

You must configure the resource to add an agent and associate the agent with the component.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have performed the following tasks:

- Been granted access to UrbanCode™ Deploy.
- Created a component in UrbanCode™ Deploy. See Creating a component in UrbanCode Deploy on page 260.
- Created a resource in UrbanCode™ Deploy. See Creating a resource in UrbanCode Deploy on page 267.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Resources**.
   **Result**
   A list of resources that are available in UrbanCode™ Deploy is displayed.
3. Perform the following steps to add an agent to the resource:
   a. Click the resource from the list for which you want to add an agent.
   b. Click the **Actions** icon from the last column, and then click **Add Agent**.
   c. Select the agent from the drop-down list.

      **Note:** The **Name** field is auto populated with the name of the agent.

   d. Click **Save**.
      **Result**
      The selected agent is added to the resource and you can view the status of the agent in the **Status** column.
4. Perform the following steps to add a component to the agent:
   a. Click the agent from the list for which you want to add a component.
   b. Click the **Actions** icon from the last column, and then click **Add Component**.
   c. Select the component from the drop-down list.

      **Note:** The **Name** field is auto populated with the name of the component.

   d. Click **Save**.
      **Result**
      The selected component is added to the agent.

**Results**

You have configured the resource in UrbanCode™ Deploy.

**What to do next**

You must create an application. See Creating an application in UrbanCode Deploy on page 269.

Related information

IBM UrbanCode Deploy Documentation

## Creating an application in UrbanCode™ Deploy

You must create an application to fetch all the components together that you want to deploy.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have been granted access to UrbanCode™ Deploy.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Applications**.
3. Click **Create Applications**, and then **New Applications**.
4. Enter a name for the application in the **Name** field.
   **Result**
   The **Environments** page for the application that you created is displayed.

**Results**

You have created the application in UrbanCode™ Deploy.

**What to do next**

You must configure the application. See Configuring the application on page 269.

Related information

IBM UrbanCode Deploy Documentation

## Configuring the application

You must configure the application to associate resources with environments and define processes to run test assets.

**Before you begin**

- You must be familiar with working with UrbanCode™ Deploy.
- You must have performed the following tasks:

- Been granted access to UrbanCode™ Deploy.
- Created a component in UrbanCode™ Deploy. See Creating a component in UrbanCode Deploy on page 260.
- Created a process for the component in UrbanCode™ Deploy. See Creating a process in UrbanCode Deploy on page 260.
- Configure the process for the component in UrbanCode™ Deploy. See Configuring the process on page 262.
- Created a resource in UrbanCode™ Deploy. See Creating a resource in UrbanCode Deploy on page 267.
- Configured the resource in UrbanCode™ Deploy. See Configuring the resource on page 267.
- Created an application in UrbanCode™ Deploy. See Creating an application in UrbanCode Deploy on page 269.

1. Log in to UrbanCode™ Deploy, if you are not already logged in.
   **Result**
   The UrbanCode™ Deploy dashboard is displayed.
2. Click **Applications**.
   **Result**
   A list of applications that are available in UrbanCode™ Deploy is displayed.
3. Click the application that you want to configure from the **Name** column.
   **Result**
   The **Environments** page for the selected application is displayed.
4. Perform the following steps to create an environment for the application that you selected:
   a. Click **Create Environment**.
   b. Enter a name for the environment in the **Name** field.
   c. Click **Save**.
5. Perform the following steps to configure resources to the environment:
   a. Click the environment that you created.
   b. Click **Add Base Resources**.
      **Result**
      A list of resources that are available in UrbanCode™ Deploy is displayed.
   c. Select the checkbox to add resources to the environment.
   d. Click **Save**.

**Result**

You can view the corresponding agent and the component that you added for the resource by using the **Expand** icon.

6. Perform the following steps to add the component to the application:
    a. Click **Applications**, and then select your application from the list.
    b. Click the **Components** tab, and then **Add Components**.
    c. Select the checkbox from the drop-down list to add components to the application.
    d. Click **Save**.
7. Perform the following steps to create a process for the application:
    a. Click the **Processes** tab, and then **Create Process**.
    b. Enter a name for the process in the **Name** field.
    c. Click **Save**.
    **Result**
    The **Design** tab for the process that you created is displayed.
8. Drag the component process listed under the **Component Process Steps** option from the left navigation pane and drop it into the design area.
9. Select the component process from the drop-down list in the **Operational (No Version Needed) Process** field.
10. Click **Save**.
11. Click the **Edit** icon, and then change the name of the properties.
12. Click **OK**, and then click **Save**.

**Results**

You have configured the application to run test assets from UrbanCode™ Deploy.

**What to do next**

You can run test assets from UrbanCode™ Deploy. See Running IBM Rational Test Automation Server tests on the UCD server on page 271.

---

Related information

IBM UrbanCode Deploy Documentation

## Running IBM® Rational® Test Automation Server tests on the UCD server

After you install the IBM® Rational® Test Automation Server UCD plugin on the UCD server, you can create a `process request` that contains the test for your application, and then run the test on the UCD server.

**Before you begin**

You must have completed the following tasks:

- Added the tests that you created in the desktop clients for your application to the project on IBM® Rational® Test Automation Server.
- Installed the latest version of the IBM® Rational® Test Automation Server UCD plugin. See Installing the plugin on page 258.
- Generated an offline user token from Rational® Test Automation Server. See Generating an offline token on page 277.

**About this task**

After you have installed the IBM® Rational® Test Automation Server UCD plugin on the UCD server, you can either use an existing component in your project or create a component. You can create a component process and select the **IBM® Rational® Test Automation Server test** step to edit the *step properties* for the test you want to run. After selecting the agent, you can create an application. You can then create an application process for the application, and then submit the application process for a run.

1. Log in to the UrbanCode Deploy (UCD) server, if you are not already logged in.
2. Click **Components** from the UCD dashboard, and then click **Create Component** to create a component.

   **Note:** You can either use an existing component or create a component.

3. Create a component process in the component by performing the following steps:

   a. Open the component that you created.

   b. Click the **Processes** tab from the component dashboard, and then click **Create Process**.
   **Result**

   The **Create Process** dialog box is displayed.

   c. Enter the required values to create a component process and click **Save**.

   **Note:** All mandatory fields are marked with an asterisk (*) in the UI.

      i. Enter the process name in the **Name** field.
      ii. Select **Operational (No Version Needed)** from the **Process Type** list.

**Note:** The **Default Working Directory** field displays the folder path where the agent can download the artifacts and create temporary files.

The process that you created is listed in the **Processes** list and the **Design** tab for the process is displayed.

**Note:** The process opens in the process editor. The process editor lists the plugins and steps. The required **Start** and **Finish** steps represent the beginning and the end of the process and are automatically placed on the design area.

4. Select the process step you want to run by completing the following steps:
   a. Search for the **IBM® Rational® Test Automation Server test** process step from the left design pane.
   b. Select the **Run IBM® Rational® Test Automation Server test** process step and drag the test into the design area.
   **Result**

   The selected test is placed in between the **Start** and **Finish** steps.
5. Specify the properties for the selected test by performing the following steps:
   a. Click the **Edit** icon .
   **Result**

   The **Edit Properties for Run IBM® Rational® Test Automation Server test** dialog box of the selected test is displayed.
   b. Specify the properties for the selected test step by following the action in the table that follows.

   **Note:** All mandatory fields are marked with an asterisk (*) in the UI.

| Field | Action required for a IBM® Rational® Test Automation Server test |
|---|---|
| Name | Enter the name of the step. |
| IBM® Rational® Test Automation Server URL | Enter the URL for Rational® Test Automation Server. |
| Offline Token | Enter the offline token that you generated from Rational® Test Automation Server. |
| File Path | Enter the file path of the Rational® Test Automation Server test that you want to run. |

| Field | Action required for a IBM® Rational® Test Automation Server test |
|---|---|
| Branch Name | Select the branch where the test assets are stored. |
| Repository Link | Enter the repository path for the test to run. |
| Project Name | Enter the name of the project that contains the test to run. |
| RIT Environment | Enter the IBM® Rational® Integration Tester environment details for the test.<br><br>**Note:** This field is applicable only for an API test. |
| **Field** | **Optional action for a IBM® Rational® Test Automation Server test** |
| Working Directory | Specify an alternative path to the working directory for this step. [1] |
| Post Processing Script | Specify if you want to run any scripts after the completion of the test run. You can click **New** to add new scripts. [1] |
| Precondition | Specify any conditions that are to be completed before the test runs. You can edit the script by clicking the script displayed. [1] |
| **Use Impersonation** checkbox | Select this checkbox to run the test as a different user. [1] |
| Auth Token Restriction | Set the authentication token actions by applying token restrictions.<br><br>The **System Default** is selected by default. You can add a new token restriction or edit the one already added. [1] |
| Custom Trust Store Path | Enter the file path of your trust store followed by the file name in the **Custom Trust Store Path** field if the IBM® Rational® Test Automation Server uses an internal CA certificate and you have imported the certificate to a custom trust store. |
| Custom Trust Store Password | Enter the trust store password if you have modified the password while creating the custom trust store.<br><br>**Note:** If you have not modified the trust store password, you can keep the **Custom Trust Store Password** field blank. |

c. Click **OK** to save the properties for the test.

6. Click **Save** in the design area.
7. Click the **Resources** tab from the UCD dashboard and create a resource by clicking **Create Top-Level Group**.

**Result**

1. You need not set this property for a step when you are running an Rational® Test Automation Server test.

The created resource is displayed on the **Resource Tree** tab page.

8. Select the agent that runs the test by completing the following steps:

   **Important:** You must have already installed the agent on the UCD server that you want to use.

   a. Select the resource displayed on the **Resource Tree** tab page.

   b. Click the **Horizontal ellipsis** icon for the selected resource.

   c. Click **Add Agent**.

   d. Select the agent to add to the resource, and then click **Save**.
   **Result**

   The selected agent is added to the resource in the **Resource Tree** pane and the status of the agent can also be viewed.

   **Important:** The agent must be **Online** for the test to run.

9. Add the component to the agent by performing the following steps:

   a. Click the **Horizontal ellipsis** icon for the agent.

   b. Click **Add Component** on the list.

   c. Select the component to add to the resource, and then click **Save**.
   **Result**

   The selected component is added to the agent for the resource in the **Resource Tree** pane.

10. Create an application by completing the following steps:

    a. Click the **Applications** tab from the UCD dashboard.

    b. Click **Create Application**.

    c. Complete the details in the **Create Application** dialog box, and then click **Save**.
    **Result**

    The **Environments** tab page is displayed for the created application.

    d. Click **Create Environment** to create an environment for the application that you created.

    e. Complete the details in the **Create Environment** dialog box, and then click **Save**.
    **Result**

    The environment that you created is displayed.

f. Click the environment to open, and then click **Add Base Resources**.

g. Select the resource from the list in the **Add Resource to Environment** dialog box, and then click **Save** to add the resource to the environment.
**Result**

The resource added to the environment is displayed.

**Note:** When you add a resource to an environment, the corresponding agent and the component are displayed for the resource.

h. Click the application from the `breadcrumbs`.
**Result**

The **Environments** tab page is displayed for your application.

i. Add the component to the application by performing the following steps:
   i. Click the **Components** tab.
   ii. Click **Add Component**.
   iii. Select the component from the list in the **Add a Component** dialog box, and then click **Save**.

**Result**

The selected component is displayed on the **Components** tab page.

j. Create a process for the application by performing the following steps:
   i. Click the **Processes** tab.
   ii. Click **Create Process**.
   iii. Complete the details in the **Create an Application Process** dialog box, and then click **Save**.

**Result**

The **Design** tab page for the application process that you created is displayed.

k. Select the component process from the left pane and drag it into the design area.

**Note:** You can click the **Edit** icon to add the properties, if required.

l. Click **Save** in the design area.

11. Select the application process to run the test by completing the following steps:

a. Click **Applications** from the UCD dashboard.

b. Click the application that you configured for a test run.

c. Click the **Request Process** icon .

**Result**

The **Run Process on <environment name>** window is displayed.

d. Select the application process that contains the test from the **Process** list.

e. Click **Submit**.
**Result**

The UCD dashboard shows the progress of the application process request.

**Results**

You have used the IBM® Rational® Test Automation Server UCD plugin to integrate Rational® Test Automation Server with UCD and run the test from your project on the UCD server.

After the UCD process request runs successfully, you can view the status of the completed process request displayed as follows:

- `Success`: When the test run is successful
- `Failed`: When the test run is failed

**What to do next**

- You can view the details of the test run as a process from the UCD dashboard by performing the following action:
  - Expand the *step*. You can then expand the application process. You can then hover over the process, and then click the **Output Log** icon . The output log is displayed. You can verify the log details.
- You can also view the test reports and logs of the test that was run on the UCD server from the **Results** page on IBM® Rational® Test Automation Server. See Test results and reports overview on page 200.

# Managing access to IBM® Rational® Test Automation Server

Desktop clients and third-party integrations use offline user tokens to connect to Rational® Test Automation Server.

You can configure the desktop clients and third-party integrations to access Rational® Test Automation Server by using the offline user tokens created from the server.

Offline user tokens are used in the following cases:

- To retrieve secrets configured on the server to be used in certain tests in Rational® Integration Tester
- To enable publishing of test results and reports from desktop clients to Rational® Test Automation Server
- To enable the Resource Monitoring Service in Rational® Performance Tester to monitor a data source during a run on Rational® Test Automation Server
- To enable third-party integrations.

**About this task**

From Rational® Test Automation Server, you can create an offline user token, copy it, and then use that token in each of the desktop clients and third-party integrations.

You can also delete all the tokens, when you no longer need access or you want to prevent access to the server.

Creating a token

- Follow these steps:

  1. Click the **User** icon from the menu bar and select **Create Token**.

     The Create Offline User Token dialog box is displayed.

     **Note:** This token is not accessible after you copy it as the dialog box closes.

  2. Copy the token and paste it into a private location for ongoing reference.

     The dialog box closes.

  3. Paste the token into the desktop client UI or the third-party integrations UI where this token is required.

     See the following table for a list of where you can use the tokens and where you can find more information.

     **Table 12. Using the offline user tokens to access Rational® Test Automation Server**

     | Use the offline user token in the following places | More information |
     |---|---|
     | Rational® Integration Tester | See Rational® Integration Tester and Rational® Test Automation Server. |
     | Rational® Performance Tester | See Publishing test results to the server.<br>See Enablement of Resource Monitoring services for a schedule. |
     | Rational® Functional Tester | See Publishing test results to the server. |
     | Third-party integrations | See Integrating with other applications on page 243. |

Deleting tokens

- Follow these steps:

    1. Click the **User** icon from the menu bar and select **Delete Tokens**.

       The Delete All Offline User Tokens dialog box is displayed.

    2. Inform the desktop client users that the tokens were deleted. Fix any test automation scripts that used tokens.

# Chapter 7. Test Manager Guide

This guide describes how to manage and track your overall test effort. This guide is intended for test managers.

Related information

Managing access to server projects on page 286

## Test assets and a server project

Rational® Test Automation Server projects manage access to your test assets, which are stored in a Git repository. Projects are either public by default or private. Private projects are not discoverable by other users. You can either add your own project or you can request to be a member of another public project.

When you add your own project, you can configure it now or later. Configuring the project includes adding details about the project, adding one or more Git repositories, optionally adding secrets, classifying encrypted datasets, and adding a change management system. To run test assets that are associated with an encrypted dataset, you must categorize the encrypted datasets by creating a classification. You can also add users to your server project so they can access your test assets. To use Jira to create defects for the tests available in your project, you must configure the Jira server for your project on Rational® Test Automation Server.

Related information

Becoming a project member on page 285

Managing access to server projects on page 286

Working with Git repositories on page 290

Protecting API test assets by using secrets on page 295

Managing an encrypted dataset on page 302

Default user administration on page 58

Configuration of Jira

### Repository considerations for a server project

To collaborate with other project stakeholders, you can open test assets from a local clone of the Git repository, pull project test assets from a Git repository, and push changes made to your local test assets to a Git repository. Before you add a project and add a repository to that project, you must consider some information about repositories.

Consider the following sections about using Git repositories with IBM® Rational® Test Automation Server. For more information about installing, setting up, and using Git, see the Git documentation.

#### Git

You must install Git or upgrade the version if you already have installed Git.

### Repositories and user identities

After you install Git, you must set up your Git repository and set up access for members. You must ensure that the repository contains your test assets.

Optionally, you can use a command line utility or Git tool to access the repository, upload your test assets, fetch or pull from the repository, push to the repository, clone the repository, and other operations you want to perform in Git.

### Local and shared repositories

After you create a remote or shared repository in Git, you can create a local version of the repository by cloning the remote repository. You must ensure that your test assets are available in the remote repository and are also cloned to the local repository.

Alternatively, if your test assets are on your local system, you can set up a Git repository in the bare mode, add the project files to the local repository, and then commit and push from the local repository to the remote repository in Git by using your preferred method.

**Note:** While copying the test assets from your local system to the repository, you must ensure that you copy the entire project that contains the test assets.

### User authentication for the Git repository

The administrator can set up different types of authentication for accessing the Git repository. Rational® Test Automation Server supports the following authentication types:

- HTTP with user name and password
- HTTP without user name and password
- HTTPS with user name and password
- HTTPS without user name and password
- SSH with SSH key and passphrase
- SSH with SSH key and without a passphrase

Based on the authentication type that is set for a repository, you must provide the same authentication values in Rational® Test Automation Server when you add a repository.

### Test assets

You must complete the following tasks in the desktop client where you are authoring your test before you check in and commit the test assets to the Git repository.

| Test type | Task | More information |
|---|---|---|
| API suite in Rational® Integration Tester | Change the local stub to a remote stub. | See Test run considerations for API Suites on page 99. |
| | Add the library files. | |

## Adding details to a server project

The first step is to add a server project and provide some details about it.

**Before you begin**

To add one or more projects to manage access to your test assets, you must log in to IBM® Rational® Test Automation Server by providing the application URL in a browser.

If you are a new user, and LDAP and Active Directory are not configured, you must first sign up by completing a form that specifies user information such as an email, user name, and password, then you can log in by using that information.

**About this task**

You add a project and give it a name and description.

1. Log in to Rational® Test Automation Server.
2. From the Home page, add a project. Give it a name and if you want, add a description.
   The Details page is displayed.
3. Decide if you want a public project, which is the default, or a private project.

**What to do next**

You can add a repository to your server project.

---

Related information

Default user administration on page 58

## Adding repositories to a server project

You can add repositories to a server project to access the test assets available in the respective repository.

**Before you begin**

You must have completed the following tasks:

- Added a project on IBM® Rational® Test Automation Server.
- Been granted permission to access the repository.

**About this task**

As a project owner or a tester, you can add one or more repositories to your project. When you add a repository, the Git repository is cloned to your project. While adding a repository, you must provide the necessary authentication credentials that are set for the Git repository that you want to add to your project. For example, if the authentication type is SSH, then you must provide the Git URL, a deploy key, and a passphrase.

After you log in, from the Home page, you can add one or more repositories to your project by following these steps.

1. Open your project.
2. Click the **Repositories** tab.
3. Click **Add repository**.

   The Add repository page is displayed.
4. Enter the URL of the Git repository that you want to add to your project.
5. Click the **Expand** icon to enter the required credentials based on any of the following authentication methods configured in the repository.

   To gain access to the repository, you must use any one of the authentication methods:

| Authentication method | Credentials required |
|---|---|
| SSH | ◦ Deploy key<br>◦ Passphrase |
| HTTPS | ◦ User name<br>◦ Password |
| HTTP | ◦ User name<br>◦ Password |

   **Notes:**
   - You must have defined the authentication type and set the authentication credentials in the Git repository.
   - If you use SSH to connect to your remote repository and Rational® Test Automation Server displays an `Auth Fail` exception while using the deliver changes option, you can resolve this exception error by regenerating your SSH keys by using the -m PEM option.
6. Click **Add**.

   The Git repository is cloned on Rational® Test Automation Server.

   **Note:** Depending on the size of the repository you are cloning, it can take a few to several minutes to clone the repository.
7. **Optional:** Repeat the steps to add another repository.

**What to do next**

You can perform the following actions on the repository that you added:

- Update the authentication credentials if they are changed in the Git repository configuration.
- Delete a repository if it is no longer required.

Related information

Working with Git repositories on page 290

## Secrets configuration

If you are working with an API suite and the project test asset contains environment variables that are required for the test runs, you must configure the environment variables as secrets in a secrets collection by using the **Secrets** tab. Configuring secrets enable the API suite to run correctly on Rational® Test Automation Server.

Related information

Protecting API test assets by using secrets on page 295

## Adding users to a server project

You can add users to your server project so they can access your test assets.

**Before you begin**

You own a project.

**About this task**

As a project owner, you can add other users to your public or private project. For example, if you have a large number of test assets that must be run, you might want your test team to help run them, and see the results.

1. Add one or more users to be members of your project. Click the **Settings** icon on the Project card in the list of **My Projects** or click **Configure** in the navigation when the project is already open.
2. Add users by entering the user name of the user you want to add. You can add a partial name and then press **Enter** to see the user that you want to add.
3. Select that user and then assign a role. Click **Viewer**, **Tester**, or **Owner**. When you add a user, the default role is Viewer.

   The added user after logging in can see the owner's project in their list of **My Projects**.
4. Repeat the applicable steps to add another user.

**What to do next**

You can add or remove roles to one or more projects that you own. All users can request to be a member of another public project.

Related information

Becoming a project member on page 285

Managing access to server projects on page 286

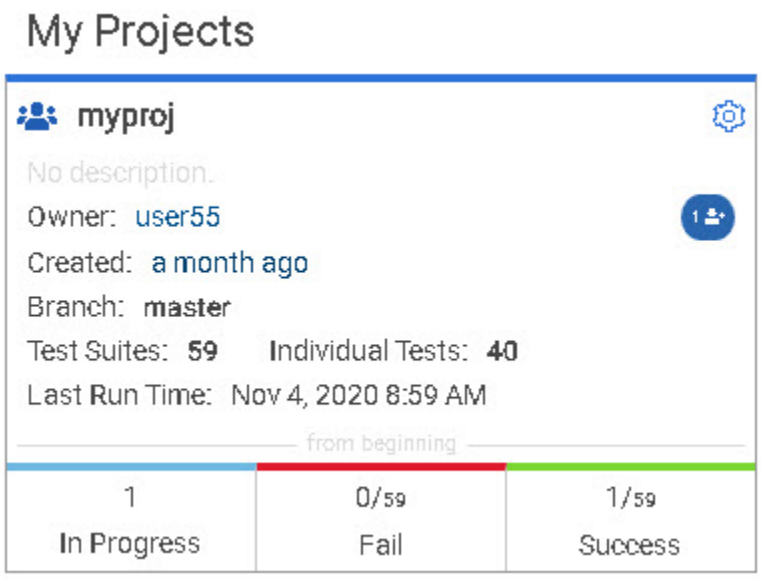
## Becoming a project member

You might want to request to be a member of another project. New users without any projects might also want to be a member of an existing project.

**About this task**

As a project owner, you can add users to your public or private projects. All users can request to become a member of a public project.

1. Request to be a member of another project. Search for the project that you want to join in the list of other projects from the project Home page. Click that Project card or the **Key** icon . Only public projects are visible in the list of projects.
2. Confirm your request when prompted. If you agree, the project owner is notified. The project owner sees a pending request **Notification** icon on their Project card.




After opening the notification, the owner can see the user that is requesting access and can accept or decline it.

If the project owner accepts your request, you are added as a member of the project with a Viewer role. A Viewer role is the default role assigned and is restricted to specific actions.

You then see the project under My Projects on the Home page.

If the project owner declines your request, follow up with the project owner.

## Managing access to server projects

You might want to add or remove a project member role from your project.

**About this task**

As a project owner, you can assign access by specifying a role when you add a user to your project. You can also assign more access or remove access for a user. Roles enable users to perform tasks on project resources such as running tests and viewing test results.

1. Choose the role in a project based on its associated actions.

| Option | Description |
|---|---|
| **Owner of a project** | ◦ All Tester's actions<br>◦ Update, archive, list, and view a project<br>◦ Assign, update, delete, and list user roles for a project<br>◦ Accept new member requests<br>◦ Create, update, and delete a dataset classification in a project<br>◦ Add, update, and remove an encrypted dataset from a classification<br>◦ Update current row for a listed dataset in a project<br>◦ Delete, list and view a report<br>◦ Update, delete, and list a repository in a project<br>◦ Get the test assets from a repository in a project<br>◦ Create, update, delete and list secrets in a project<br>◦ Create, update, delete, and list resource monitoring sources in a project<br>◦ Get the content of a resource monitoring source in a project<br>◦ Register, update, delete, and list resource monitoring agents |
| **Tester of a project** | ◦ All Viewer's actions<br>◦ List repositories in a project |

| Option | Description |
|---|---|
| | ◦ Run a test and create a report<br>◦ Stop a test while running<br>◦ Get the test assets from a repository in a project<br>◦ Get a list of datasets in a project<br>◦ Get the content of a dataset<br>◦ Update the current row for a listed dataset in a project<br>◦ Add, update, and remove an encrypted dataset from a classification<br>◦ Delete a report if you created the report |
| **Viewer of a project** | ◦ List your own roles on a project<br>◦ List running tests<br>◦ List and view a report<br>◦ List resource monitoring sources in a project<br>◦ Get the content of a resource monitoring source in a project<br>◦ List the resource monitoring agents |
| **Non-member of a project** | ◦ Request to be a member of a project |

2. Open the project that you own.
   From the project details, you can see a list of project members for your project and then add or remove a members role.
3. Add or remove the member access by using the **Menu** icon next to each user in the member list.

## Archiving or unarchiving server projects

You can archive a project when that project is no longer needed but you want to retain it for future reference. If you do want to use an archived project, you can unarchive it.

**About this task**

As a project owner,you can archive projects that are inactive. Archived projects are not visible to users or project members, but owners can show or hide their archived projects. Archived projects can be unarchived.

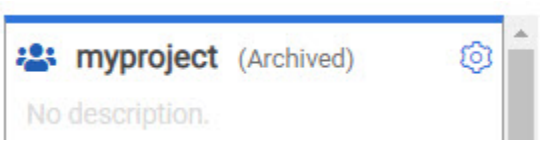
**Note:**

When you migrate projects from the previous version of Rational® Test Automation Server to V10.1.0, the project **Owner** must log in to Rational® Test Automation Server to initiate the migration of all the repositories in that project, as a result, other members can access the test assets in that project.

On migration of an archived project, the project **Owner** must log in to Rational® Test Automation Server and then navigate to the **Configure > Project > Repositories** tab to migrate the configured repositories in the archived project.

- Archive a project by following these steps:
  1. Open the project that you want to archive.
  2. From the project details, archive the project.
  3. Confirm that you want to archive the project when prompted. If you proceed, the project is archived and is hidden in your project list. To show or hide the archive projects, select the appropriate option by using the **Menu** icon from the menu bar.

     When visible, the archived project shows as archived in the list of **My Projects**.

     **Note:** The option to configure a project continues to display after you archive a project.

- Unarchive a project by following these steps:
  1. Open an archived project and unarchive it.
  2. Confirm that you want to unarchive the project when prompted. If you proceed, the project is unarchived and is shown in your project list.

## Deleting server projects

You might want to delete a project when the project is no longer needed, and you want to free up disk space.

**About this task**

As a project owner, you can delete a project that you no longer want. When you delete a project it is permanent. You cannot undo the delete project action.

After you delete the project, you can use the project name again.

1. Open the project that you want to delete.
2. From the project details, delete the project.

3. Confirm that you want to delete the project when prompted. If you proceed, the project is deleted including all of its test assets, results, and members.

**Note:** The repository that you used for the project is not deleted.

## An overview of test assets, modifications, and scheduled runs

After you configure a project with one or more Git repositories and run some test assets, you can view statistics about those test assets.

From the **Overview** page, several cards display charts, sliders, and tables of the statistics of the test assets in your server project. You can click some of the charts, sliders, or tables to see more information, such as the progress of a scheduled run or the results of a run. You can also review some statistics by date and see more details such as the proportion of test assets by type.

You can view a server project, which might contain multiple repositories and multiple branches, by selecting the name of the branch from the **Branch** drop-down list. For more information about selecting branches, see the related links.

**Test suites**

This card displays a slider of the percentage of test suites by type. A test suite is a collection of tests, including schedules, compound tests, API Suites, and AFT Suites. You can hover over the slider and see the proportion of test suites by test suite type. The number of suites shown are relative to the selected branch.

**Individual tests**

This card displays a slider of the percentage of individual tests by type that are used in the test suites. You might have performances tests, API tests, or UI tests according to the desktop clients that you used to create the tests. You can hover over the slider and see the proportion of individual tests by type. The number of tests shown are relative to the selected branch.

**Execution results**

This card displays a chart of the proportion of test suites that were run by status and by date. If no date is selected, the card displays all the run results associated with the project since it was created. You can see the run results by clicking a test suite status. You can also hover over the chart and see the test suite status and the proportion of test assets. You can view statistics by test suites run.

**Notes:**

- The results shown are generated from test suites run.
- The statistics exclude canceled test suites.
- The number of test suites and individual tests that are associated with your project are shown on the **Home** page along with a verdict summary. You can see the run results by

clicking a verdict. You can also see the last date and time when you ran one or more test assets.

- The number of suites shown are relative to the selected branch.

**Last run**

This card displays a table of the last three runs by date and time. You can see the run results by clicking each last run.

**New or modified**

This card displays a chart of new and modified test assets that were made in your Git repository for a week or for the last seven days based on a selected date range. You can hover your mouse over each bar in the chart and see the proportion of test assets by type. The data shown is relative to the selected branch.

**Scheduled runs**

This card displays a table of the next three scheduled runs by date and time. You can see the progress of a scheduled run by clicking each scheduled run.

---

Related information

Test assets and a server project on page 280

Test resource access by using the global branch

# Working with Git repositories

You can add another repository in addition to the existing repositories in your project or edit the repository settings, by updating the credentials, or by removing references to the repository.

## Adding another repository

When you created a project, you probably added a repository. If you did not add a repository when you created a project, you can add one or more repositories any time after you create a project.

## Updating the authentication credentials of the repository

If the authentication credentials of the Git repository which was added to your project are changed, then you must update the new credentials of the repository in Rational® Test Automation Server.

**Before you begin**

You must be a project owner or a tester with access to edit repositories.

**About this task**

In Rational® Test Automation Server, the repository is refreshed at regular intervals to pull the latest changes committed to the Git repository. After you update your authentication credentials of the repository, the repository is refreshed to ensure that you have continued access to the Git repository and to the latest assets in the repository.

1. Open the project.
2. Go to the Repositories page.
3. Click the **Menu** icon and click the **Change** icon.

   The Edit repository page is displayed.
4. Enter the new credentials and click **Update**.

   The new credentials are updated and your access to the repository is retained.

## Refreshing repositories manually

You can manually trigger the refresh activity for a server repository to instantly fetch and synchronize the changes from the remote repository.

**Before you begin**

You must have completed the following tasks:

- Configured the repository that contains the test assets in your project.
- Been assigned a project owner or a tester role.

1. Open the project.
2. Go to the **Repositories** page.
3. Click the **Menu** icon and select **Refresh**.

   The refresh activity is triggered.

   **Note:** The **Refresh** option is disabled when the refresh is in progress. After the refresh is completed, the **Refresh** option is enabled.

   The following table lists the icons and their description based on the status of the refresh activity:

| Refresh icon | Description |
|---|---|
| | The last refresh activity was successful. |
| | The last refresh activity was successful but errors were found while processing some of the test assets. To view the error details of the test assets for a specific repository, you can click the arrow inline with the repository name. |

| Refresh icon | Description |
|---|---|
| | You can resolve the errors of these test assets in the Git server, and then refresh the server repository, if required.<br><br>**Note:** The repository, which contains the branch that is selected in the Global branch field, displays the unprocessed test assets for that branch. Other repositories display the unprocessed test assets for their master branch. |
| | The refresh activity was not successful due to any one of the following reasons:<br>◦ Internal error<br>◦ Connection issue<br>◦ Authentication failure<br>◦ Disk quota exceeded<br>◦ Repository unreachable |

**Results**

You have manually refreshed the repository and the **Last Refreshed** field is updated with the following details:

- The time when it was last refreshed.
- Latest commit ID and message of the remote repository.

## Creating webhooks

You can configure a webhook for a repository in the Git server to notify Rational® Test Automation Server whenever there is a push event in the remote repository. Then, the refresh activity is triggered to fetch the latest changes to the server repository.

**Before you begin**

You must have completed the following tasks:

- Configured the repository that contains the test assets in your project.
- Been assigned a project owner or a tester role.

1. Open the project.
2. Go to the **Repositories** page.
3. Click the **Menu** icon and then click **Webhook**.

   The **Webhook Configuration** dialog box is displayed.
4. Copy the URL and the Secret of the repository that you want to configure.

**Note:** The URL and the Secret details are specific to a repository.

5. Click the link of the repository for which you want to configure the webhook.

   The remote repository page is displayed.

6. Paste the details, which you copied from Rational® Test Automation Server, in the appropriate fields of the remote repository.

**Results**

You have configured the webhook for a repository in the Git server. The changes of the remote repository are synchronized in the server repository.

## Deleting a repository

You can delete repositories that you no longer require in your test environment.

**Before you begin**

You must have completed the following tasks:

- Configured the repository that contains the test assets in your project.
- Been assigned a project owner or a tester role with access to delete repositories.

**About this task**

You can delete a repository that you have configured for your project.

1. Open your project and go to the Repositories page.

   You can see a list of repositories that you added to the project.

2. Delete a repository that you want to remove from the project.

   The repository is deleted and removed from the list.

**What to do next**

After you have edited the Git credentials, you can continue accessing the Git repository and the test assets. If you have deleted the Git repository, you cannot access the test assets available under that repository.

---

Related information

Adding repositories to a server project on page 282

## Adding repositories to a server project

You can add repositories to a server project to access the test assets available in the respective repository.

**Before you begin**

You must have completed the following tasks:

- Added a project on IBM® Rational® Test Automation Server.
- Been granted permission to access the repository.

**About this task**

As a project owner or a tester, you can add one or more repositories to your project. When you add a repository, the Git repository is cloned to your project. While adding a repository, you must provide the necessary authentication credentials that are set for the Git repository that you want to add to your project. For example, if the authentication type is SSH, then you must provide the Git URL, a deploy key, and a passphrase.

After you log in, from the Home page, you can add one or more repositories to your project by following these steps.

1. Open your project.
2. Click the **Repositories** tab.
3. Click **Add repository**.

   The Add repository page is displayed.
4. Enter the URL of the Git repository that you want to add to your project.
5. Click the **Expand** icon to enter the required credentials based on any of the following authentication methods configured in the repository.

   To gain access to the repository, you must use any one of the authentication methods:

| Authentication method | Credentials required |
|---|---|
| SSH | ◦ Deploy key<br>◦ Passphrase |
| HTTPS | ◦ User name<br>◦ Password |
| HTTP | ◦ User name<br>◦ Password |

   **Notes:**
   - ◦ You must have defined the authentication type and set the authentication credentials in the Git repository.
   - ◦ If you use SSH to connect to your remote repository and Rational® Test Automation Server displays an `Auth Fail` exception while using the deliver changes option, you can resolve this exception error by regenerating your SSH keys by using the -m PEM option.
6. Click **Add**.

The Git repository is cloned on Rational® Test Automation Server.

**Note:** Depending on the size of the repository you are cloning, it can take a few to several minutes to clone the repository.

7. **Optional:** Repeat the steps to add another repository.

**What to do next**

You can perform the following actions on the repository that you added:

- Update the authentication credentials if they are changed in the Git repository configuration.
- Delete a repository if it is no longer required.

---

Related information

Working with Git repositories on page 290

# Protecting API test assets by using secrets

Secrets are key-value pairs that are created for your project in IBM® Rational® Test Automation Server under a secrets collection. You can create secrets collections for your project that enable you or members in your project to use secrets at test runtime either in Rational® Test Automation Server or in desktop clients.

The secrets collections in a project in Rational® Test Automation Server has a separate access control list managed by the members with access to the secrets collections. Controlling access to secrets means controlling access to applications and systems under test. The introduction of secrets (under secrets collections) for a project has simplified managing access to separate environments. If a member of a project does not have access to a secret, for example, *a server credential* then the member cannot accidentally or maliciously run tests against that server. For example, tests that must access the database server by using the server credentials to retrieve stored data can only be run by a member if the access to the secrets is granted.

**Note:** Secrets and secrets collections are applicable to test assets authored in IBM® Rational® Integration Tester that enable running tests in defined environments. Secrets are not applicable to tests authored in Rational® Functional Tester or Rational® Performance Tester.

As a project member with the *Owner* or *Tester* role, you can create secrets collections in the project. You can grant or restrict access to the secrets collection that you create in the project.

Members with access to a secrets collection can access, edit, or delete the secrets collection in Rational® Test Automation Server and can view secrets, edit secrets, or delete secrets.

Members with access to secrets collections can grant access to or remove the following:

- Other members added specifically
- All members with a specific role

Members in the project with the *Owner* or *Tester* role and with access the secrets collection can use the secrets in the secrets collection, in tests at runtime.

If you are configuring a project to run an API Suite with tests that refer to secret values, you must configure the secrets under a secrets collection by using the **SECRETS** tab. You must complete the following tasks:

1. Create a secrets collection. See Step 1 on page 297 in Managing secrets collections on page 296.
2. Add secrets in the secrets collection created. See Step 1 on page 298 in Creating a secret in a secrets collection on page 298.
3. Grant access to project members or member roles, who can access the secrets collection. See Step 1 on page 299 in Granting access to members or member roles on page 299.

## Managing secrets collections

You can create secrets in a secrets collection for your project. Secrets are credentials required in certain tests during test runs. Secrets stored in the collection can be used by members to run tests on different environments and eliminates the need to store secrets in multiple locations. You can opt to edit or delete a secrets collection that you configured for your project any time after you create a secrets collection.

**Before you begin**

- You must have created a project on Rational® Test Automation Server. See Test assets and a server project on page 280.
- You must have completed the following tasks before you edit or delete a secrets collection:
  - Configured a secrets collection in your project.
  - Created secrets in the selected secrets collection. See Creating a secret in a secrets collection on page 298.
- You must be a member with the *Owner* or *Tester* role to create a secrets collection.
- You must be a member with access to the secrets collection to edit or delete the secrets collection. See Granting access to members or member roles on page 299.

**About this task**

You must configure secrets collections in your project so that the members of the project can use secrets contained in a collection during test runs. You can configure secrets so that you can use them in different test environments.

As a member with access to the secrets collection, you can opt to edit or delete a secrets collection configured in a project. For example, you might want to edit the secrets collection name or delete the secrets collection if the testing environment has changed and if secrets that are configured earlier are not required.

- To create a secrets collection, go to Step 1 on page 297.
- To edit or delete a secrets collection, go to Step 4 on page 297.

To create a secrets collection:

1. To create a secrets collection while configuring a new project in the Rational® Test Automation Server UI, open the **SECRETS** tab in the **Project Configuration** and create a secrets collection. Use **Add Collection**.
2. Alternatively, to create a secrets collection in an existing project, complete the following steps:
    a. Log in to Rational® Test Automation Server and from the User Interface (UI) open the project listed under **My Projects** for which you want to create a secrets collection.
    b. Open the Project Configuration page, and then open the **SECRETS** tab to create a secrets collection.
3. Enter a name for the secrets collection as its *Identifier*.

    **Tip:** You can create a secrets collection that contains secrets for a particular test environment in your project. For example, the secrets collection *test_env* can contain secrets that application testers can use in tests that they run while the secrets collection *dev_env* can contain secrets that application developers can use in tests they run.

    A message is displayed for the successful creation of the secrets collection.

    The secrets collection created is displayed.

    You can add secrets to the secrets collection you created.

To edit or delete a secrets collection:

4. Log in to Rational® Test Automation Server and from the UI open the project listed under **My Projects**.
5. Open the secrets collection from the **SECRETS** tab in the Project Configuration page.
    If there are multiple secrets collections in the project, select the secrets collection that you want from the list.
    - To edit a secrets collection, go to Step 6 on page 297.
    - To delete a secrets collection, go to Step 7 on page 297.
6. To edit a secrets collection, complete the following steps:
    a. Click the **Edit** icon to edit the selected secrets collection.

        **Note:** The **Edit** icon is displayed only for the project owner.

    b. Edit the name of the secrets collection, and update the secrets collection.

        The secrets collection is updated with the updated name.
7. To delete a secrets collection, click the **Delete** icon to delete the selected secrets collection.

    The selected secrets collection is removed from the list of secrets collections configured for the project.

**Results**

You have completed the following tasks:

- Created a secrets collection for your project.
- Edited the name of a secrets collection in your project.
- Removed a secrets collection from your project.

**What to do next**

- If you have created a new secrets collection, you must add secrets to your secrets collection.
- You must provide access to project members or member roles to the secrets collection by selecting members or member roles.

## Creating a secret in a secrets collection

You must create secrets in the secrets collections configured in your project so that the secrets contained in a collection can be used in certain tests by members of the project with access to the secrets collections during an API suite run.

**Before you begin**

You must have created a project on Rational® Test Automation Server and configured a secrets collection in your project.

You must be a member with access to the secrets collection.

**About this task**

You can also configure secrets such that the secrets can be used across different test environments by members with access to the secrets collection. Secrets correspond to the environment variables or tags that you create in a Rational® Integration Tester project specific to an environment.

1. To create a secret under a secrets collection while configuring a new project in the Rational® Test Automation Server UI, select the secrets collection listed in the **SECRETS** tab in the Project Configuration page and create a secret under the secrets collection.
2. Alternatively, to create a secret under a secrets collection in an existing project, complete the following tasks:
    a. Log in to Rational® Test Automation Server and from the UI open the project listed under **My Projects**.
    b. Open the secrets collection from the **SECRETS** tab in the Project Configuration page.
3. Enter a name for the secret as its *Identifier* and enter the `password` as the *Value* for the secret.
   For example, under the secrets collection (named as `test_env`), enter the name of the secret to access a database as `dbcred` and enter the `password` required to access the database as its value.

   A message is displayed for successful creation of the secret.

**Results**

You have created secrets in the selected secrets collection for your project.

**What to do next**

- You can view, edit, or delete the secrets created under a secrets collection any time you want.
- You can use the secrets in the tests that require these secrets during test runs.

## Granting access to members or member roles

You can grant or revoke access to the secrets collection in your project to individual members with different roles or the all members with a specific role. Without access to the secrets collection, members cannot view, create, edit, delete, or use the secrets in the secrets collection.

**Before you begin**

You must have created a project on Rational® Test Automation Server and configured a secrets collection in your project.

You must be a member with access to the secrets collection.

1. To grant access to a secrets collection while configuring a new project in Rational® Test Automation Server UI, select the secrets collection listed in the **SECRETS** tab in the Project Configuration page.
2. Alternatively, to grant access to a secrets collection in an existing project, complete the following tasks:
    a. Log in to Rational® Test Automation Server and from the UI open the project listed under **My Projects**.
    b. Open the secrets collection from the **SECRETS** tab in the Project Configuration page.
    If there are multiple secrets collections in the project, select the secrets collection that you want from the list.
3. To grant access to a secrets collection in a new project or an existing project, select from the following methods:
    - To add all members with a specific role, click the role listed under **Grant access to role**. For example, if you select **Testers**, then all members in the project with a tester role are granted access to the secrets collection. You can select any role or all the roles listed.
    - To select specific members to grant access to the selected secrets collection, enter the name or the email ID of the member in the field box and add them from the list that is displayed.

**Note:** Members added specifically are listed under **Members with access to this collection** but all the members granted access solely due to their roles are not listed.

**Important:** Irrespective of the role that the member (*Owner*, *Tester* or *Viewer*) was assigned in the project, the access to the secrets collections has to be specifically granted to the members from the **SECRETS** tab.

**Removing access to a secrets collection**

4. To remove access granted to all members with a specific role or a specific member, select from the following methods:
    - To remove all members with a specific role, click the role listed under **Grant access to role** to clear the selection. For example, if **Testers** is selected and you clear it, then all members in the project with a tester role are removed from the access list to the secrets collection.
    - To remove specific members with access to the secrets collection, select the member and click the **Delete** icon .

    **Notes:**
    - Any member with access to the secrets collection can remove access of other members specifically added or of all members with a specific role.
    - Members with access to the secrets collection can remove themselves from the access list. Members can do this when there is at least one member remaining in the list. After removing themselves, members cannot add themselves back to the access list and must be added by any of the other remaining members in the list.

**Results**

You have added members from your project or members with specific role to the access list of people who can access secrets in the selected secrets collection, or you have removed specific members or members with specific role from the access list.

**What to do next**

You can create secrets under secrets collections for your project.

## Managing secrets

You can view, edit, or delete the secrets configured under a secrets collection any time after you have created secrets or after you were granted access to the secrets collection. You can change the value of the secret by editing the secret. You can delete secrets that you no longer require in your test environment.

**Before you begin**

You must have created a project on Rational® Test Automation Server and configured a secrets collection in your project.

You must have created secrets in the selected secrets collection or the secrets collection must contain secrets.

You must be a member with access to the secrets collection.

1. Log in to Rational® Test Automation Server and from the UI open the project listed under **My Projects**.
2. Complete the following steps:

a. Open the secrets collection from the **SECRETS** tab in the Project Configuration page.
b. Optionally, select the secrets collection that you want from the list if there are multiple secrets collections in the project.

The secrets configured in the selected secrets collection are displayed.

3. Complete the steps for the task you want to perform as listed in the following table:

| Task | Steps |
|---|---|
| Viewing a secret value | Click the **Show** icon for the secret you want to view its value, which most likely is a password for the secret.<br><br>The value configured for the secret is displayed. |
| Editing a secret value | Click the **Edit** icon for the secret you want to edit, and enter a `new value` for the secret as its *Value*. The value can be a password for the secret.<br><br>**Note:** You can only change the value of the secret.<br><br>The value of the selected secret is changed. |
| Deleting a secret | Click the **Delete** icon in the row of the secret you want to delete.<br><br>After deleting it, the secrets list in the collection is removed from the list. |

**Results**

- You viewed the password configured of the secret under a secrets collection that you created or were granted access.
- You changed the secret value of the secret under a secrets collection in your project.
- You deleted and removed the secret from the selected secrets collection in your project.

**What to do next**

You can use secrets in the tests that require these secrets during test runs.

Related information

Test run configurations on page 121

Becoming a project member on page 285

# Managing an encrypted dataset

You can use encrypted datasets to limit access to confidential information such as account number or passwords. You can arrange data by an appropriate category so that project members can use datasets more effectively in certain tests and protect them.

**About this task**

A dataset can contain classified information that other members can access with permission. As a project owner, you can group encrypted datasets into different classifications and enable project members to view and edit datasets and run tests associated with the encrypted datasets. After you have created a classification, you can change the classification for a dataset. You can also delete a classification if you do not require it in your test environment.

**Notes:**

- You must grant access and provide the original password set in the desktop client to other members of the project to work with an encrypted dataset.
- A project member who has been added as a Tester role can work with the encrypted dataset.

## Creating a classification

As a project owner, you can organize encrypted datasets by creating a classification so that project members can use and protect datasets more efficiently.

**Before you begin**

You must have completed the following tasks:

- Created at least one dataset and encrypted the dataset with a password in the test assets in desktop clients and added it to the Git repository.
- Created a project in . See Test assets and a server project on page 280.
- Configured the repository that contains the test assets in your project. See Adding repositories to a server project on page 282.

**About this task**

After creating a classification, you can grant access and provide the original password set in the desktop client to other members of the project to work with an encrypted dataset.

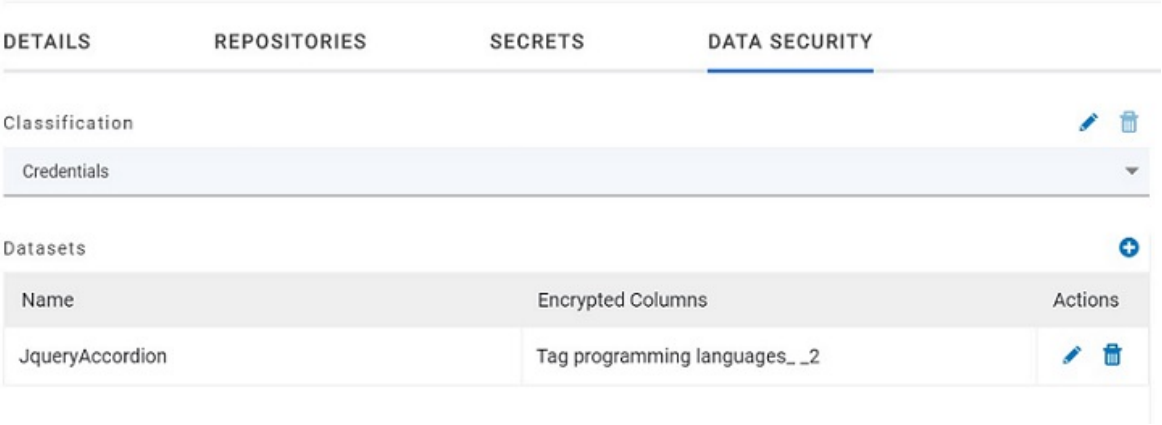
1. Go to the  URL.
2. Enter your user name and password, and then click **Login**.
3. Open your project from the  UI.
4. Click **Configure**, and then the **DATA SECURITY** tab.
5. Click **New classification** and enter a name for the classification.
6. Click **Create**.

7. Click the **Add** icon to select the encrypted dataset to become part of the new classification.

   **Note:** The **Add** icon is disabled if your repository contains only the datasets that are not encrypted.

8. Select a dataset from the list and enter a password for the dataset that was set in the desktop client, and then click the **Add** icon.

   The encrypted dataset that is added to the classification is displayed.




**What to do next**

You can grant access to other project members to use the encrypted dataset.

## Editing or deleting a classification

After you have created a classification, you can edit the name of the classification. You can also delete a classification when it is not required in your test environment.

**Before you begin**

You must have completed the following tasks:

- 
- Created at least two classifications. See Creating a classification on page 302.

**About this task**

You can opt to edit or delete a classification for your project any time after you create a classification. For example, you might want to edit the name of the classification or delete the classification if the classification that is created earlier are not required.

**Note:** You must be a project owner to create, edit, or delete a classification.

- To edit a classification, go to Step 1 on page 304.
- To delete a classification, go to Step 2 on page 304.

1. To edit a classification:

   a.

   b.

   c.

   If there are multiple classifications created in the project, select a classification that you want from the list.

   d. Click the **Edit** icon to edit the selected classification.

   e. Edit the name of the classification, and then click **Save**.

2. To delete a classification:

   click the **Delete** icon in the row of the classification that you want to delete and click **Delete** in the **Delete classification** window.

   **Note:** Before deleting a classification, you must move the added datasets to another classification else the delete icon is disabled.

**Results**

- You have edited the name for a classification in your project.
- You have deleted a classification.

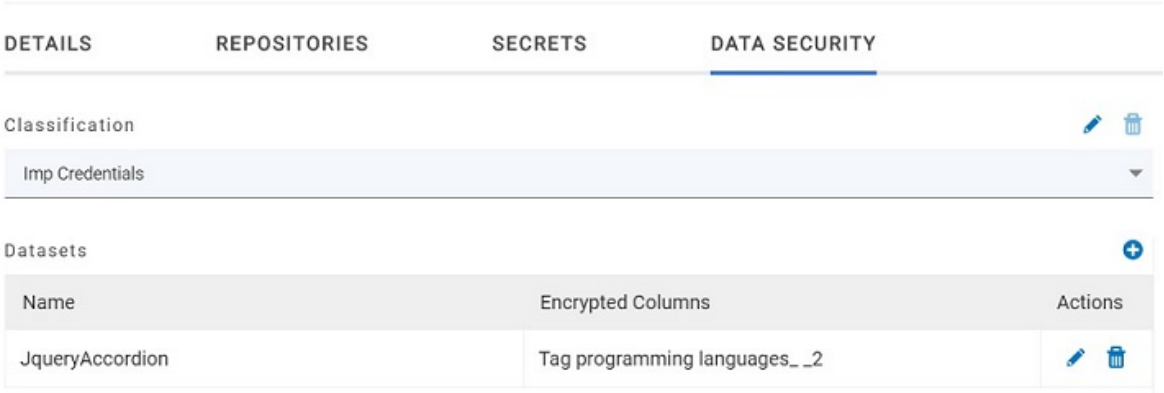
## Moving an encrypted dataset to another classification

When you add many encrypted datasets to the same classification, you can move some of them to another classification.

**Before you begin**

1.
2.
3.
4. Select a classification from the list that has the encrypted dataset.
5. Click the **Edit** icon from the **Actions** column of a dataset.
6. Select a classification from the list and enter a password in the **Change the classification for the Dataset** window.
7. Click **Save**.

   A classification for a dataset is updated successfully.




**Results**

You have moved the encrypted dataset from one classification to another.

## Removing a dataset from the classification

You can remove a datasets added to a classification when they are no longer required.

**Before you begin**

You must have completed the following tasks:

- 
- Created a classification and added the encrypted dataset to it. See Creating a classification on page 302.

1.
2.
3.

4. Select a classification from the list that has an encrypted dataset.
5. Click the **Delete** icon from the **Actions** column of a dataset.
6. Click **Remove** in the **Change the classification for the Dataset** window.

**Note:** Removing dataset from the classification also removes the password stored in  for encrypted data. You must enter the password again to gain access to the encrypted columns.

**Results**

You have removed the datasets from a classification.

## Granting classification access to members or members roles

You can grant or revoke access to the classification in your project to individual members with different roles or the all members with a specific role. Without access to the classification, members cannot view, create, edit, delete, or use the classification.

**Before you begin**

You must be a project owner and have completed the following tasks:

- 
- 
- Added one or more users to your project. See Adding users to a server project on page 284.
- Created at least one classification. See Creating a classification on page 302.

1. 
2. 
3. 
4. Select the classification from the drop-down list.
5. Choose any of the following methods to grant access to a member:
   **Choose from:**
   - To add all members with a specific role, click the role listed under **Grant access to role**. For example, if you select **Testers** then all members in the project with a tester role are granted access to the selected classification. You can select any role or all the roles listed.
   - To select specific members to grant access to the selected classification, enter the name or the email ID of the member in the **Grant access to member** field and add them from the list that displays.

**Note:** Members added specifically are listed under **Members with access to this classification** but the members added for a role are not displayed.

**Important:** Irrespective of the role that the member (Owner, Tester or Viewer) was assigned in the project, the access to the classification has to be specifically granted to the members from the **DATA SECURITY** tab.

6. Choose any of the following methods to revoke access from a member:
   **Choose from:**
   - To remove all members with a specific role, click the role listed under **Grant access to role** to clear the selection. For example, if **Testers** is selected and you clear it, then all members in the project with a tester role are removed from the access list to the classification.
   - To remove specific member with access to the classification, select the member and click the **Delete** icon .

   **Notes:**
   - Any member with access to the classification can remove access of other members specifically added or of all members with a specific role.
   - Members with access to the classification can remove themselves from the access list provided that there is at least one member in the list. After removing themselves, members cannot add themselves back to the access list and must be added by any of the other members in the list.

**Results**

You have granted or revoked the classification access to project members.

Related information

Adding details to a server project on page 282

# Managing notifications

IBM® Rational® Test Automation Server provides a feature to display notifications for different events within the user interface (UI) of Rational® Test Automation Server. You can configure an SMTP server on Rational® Test Automation Server when you want Rational® Test Automation Server to send out notifications about the different events as emails to the subscribed users.

Rational® Test Automation Server provides you the following features to receive notifications about server events:

| Channels | Description |
|---|---|
| In-app | Notifications are displayed within the UI of Rational® Test Automation Server. |
| SMTP | Notifications are sent by Rational® Test Automation Server in emails to subscribed users. |

When you register as a user in Rational® Test Automation Server, the in-app and SMTP channels are automatically subscribed for you.

### Managing notifications in the server UI

Viewing the in-app notifications or notifications in the server UI is a feature that is enabled automatically when you register as a user in Rational® Test Automation Server.

You are subscribed to the following channels for all events as the default option when you sign up on the server.

- In-app
- SMTP

If you want to receive notifications about specific events and not all events, you can change the default subscriptions and modify your subscription for the server events. You can also view the in-app notifications that are displayed for you. See Managing in-app notifications on page 308.

### Managing SMTP notifications

You, as a *Server Administrator* or *Team Space Owner* can configure an existing Simple Mail Transfer Protocol (SMTP) server so that Rational® Test Automation Server can send out notifications about the server events in emails to the registered users. See Configuring an SMTP server to manage email notifications on page 318.

You are subscribed to the following channels for all events as the default option when you sign up on the server.

- In-app
- SMTP

If you, as a registered user of Rational® Test Automation Server want to receive the email notifications about specific events and not all events, you can change the default subscriptions and modify the subscription for the server events. See Managing email notifications on page 314.

## Managing in-app notifications

You can manage the in-app notifications by changing the subscriptions that are automatically enabled for you for all server events. You can view the server events that are automatically subscribed and then change the subscription for any or all events. For example, as a team space owner, you can choose to be notified when members of the team space create new projects or delete their projects but not be notified when tests are run.

**Before you begin**

- 

**About this task**

The in-app notifications that are automatically subscribed for you, are displayed in the server UI under the following categories for the server events that occur:

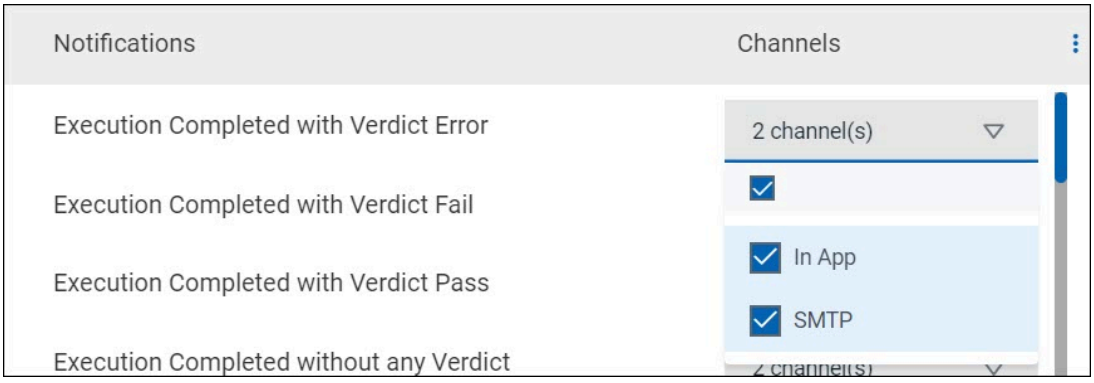
| Category | Notifications displayed |
|---|---|
| User | Include information about actions performed by team space members in the team space. |
| System | Include information about actions performed by registered users at the server level. |

1. Select from the following actions that you want to perform:
   - Modify notification settings. Go to Step 2 on page 309.
   - View notifications. Go to Step 3 on page 310.
   - Unsubscribe from the in-app notifications. Go to Step 4 on page 312.
2. Perform the following steps to modify the notification settings:
   a. Click the **User** icon on the menu bar.
   b. Click **Notification Settings**.

      The **Notification Settings** dialog is displayed.
   c. Change the language in which you want to receive the notifications, if you do not want to use the default language.

      The default language is the locale language set on the computer that hosts .

      You can notice that for each of the server events, both the channels are displayed as selected, which is the default selection unless you have modified the subscriptions.



d. Identify the server events for which you want to change the subscription.

e. Perform the actions indicated in the following table to clear the subscriptions for server events that you do not want to be notified and to select the in-app subscription for the server events that you want to be notified.

| When | Action |
|---|---|
| You do not want to receive the in-app notifications for a specific server event. | Perform the following steps:<br>i. Expand the channel list in the row of the server event.<br>ii. Clear the checkbox in the header row to clear the subscriptions to both channels.<br>iii. Click **Apply**.<br><br>You have unsubscribed from both channels and the in-app notifications for the server event is not displayed for you when the event occurs. |
| You want to receive the in-app notifications for a specific server event. | Perform the following steps:<br>i. Expand the channel list in the row of the server event.<br>ii. Select any of the following actions:<br>▪ Clear the **SMTP** checkbox.<br><br>The **In App** selection is retained while the checkbox in the header row and for the **SMTP** options are cleared.<br><br>▪ Clear the checkbox in the header row to clear the subscriptions to both channels, and then select the **In App** checkbox.<br>iii. Click **Apply**.<br><br>You have unsubscribed from the SMPT channel and retained the subscription for the in-app notifications for the server event. |

f. Go to Results on page 313.

3. Perform the following steps to view the in-app notifications:

a. Click the **Notifications** icon on the menu bar.

The **Notifications** dialog is displayed. The notifications are displayed in the following tabs:

- User
- System

The notifications in the **USER** tab is displayed as the default option.

b. View the notifications in the **USER** tab.

c. Click the **SYSTEM** tab to view the notifications at the server level.

You can find the information about the default notifications that you can view based on your role in the server as in-app notifications in the **USER** tab and **SYSTEM** tab.

| Category | Notification details | or | or | | |
|---|---|---|---|---|---|
| User notifications | Information about events at the server level or across team spaces performed by other team space owners. | ✓ | ✓ | ✗ | ✗ |
| | Information about events in the team space performed by other team space members. | ✓ | ✗ | ✗ | ✗ |
| | Information about events in the project performed by other project members. | ✗ | ✓ | ✓ | ✗ |
| | Information about events performed by you in a project. | ✗ | ✗ | ✓ | ✓ |
| System notifications | Information about events at the server level or across team spaces. | ✓ | ✓ | ✗ | ✗ |
| | Information about events in the team space performed by other team space members. | ✓ | ✓ | ✗ | ✗ |
| | Information about events in the projects at the team space level performed by other project members. | ✓ | ✓ | ✗ | ✗ |

d.

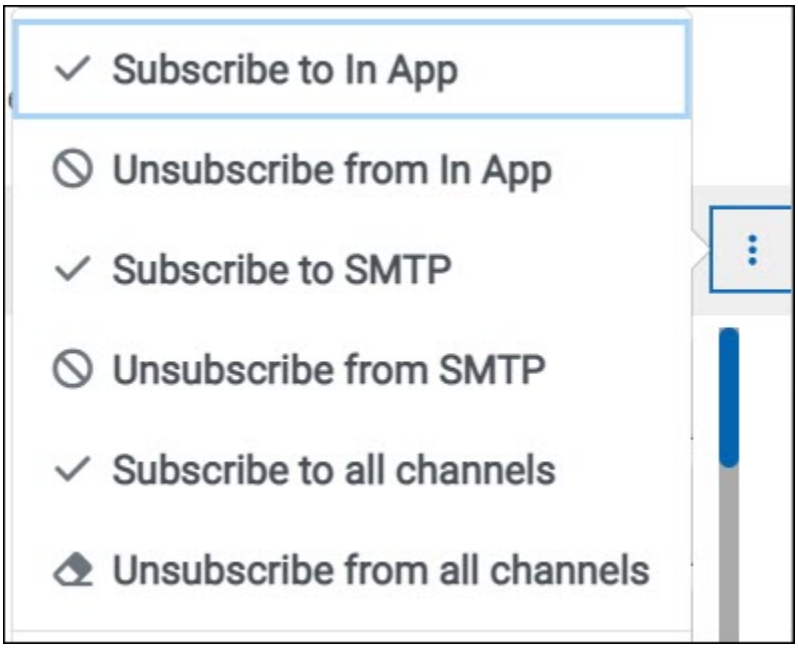
4. Perform the following steps to unsubscribe to the in-app notifications:

   a.

   b.

   c. Click the **Open action menu** icon ⋮ in the row of **Notifications**.

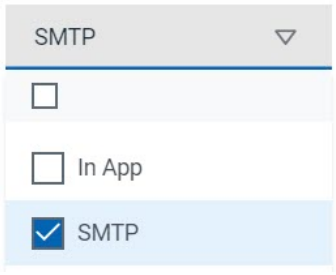
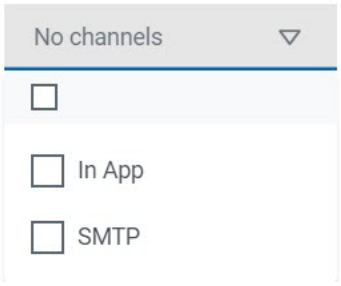
      The following panel is displayed:




      You can notice that the subscriptions to in-app, SMTP, and all channels are displayed as selected.

   d. Perform the actions indicated in the following table for the option you want:

| When | Action | Results in |
|---|---|---|
| You do not want to receive any in-app notifications. | Click the **Unsubscribe from In App** option, and then click **Apply**. | The in-app channel selection for all server events is cleared and you are unsubscribed from the in-app channel. The SMTP notifications is displayed as selected for subscription for all server events. |

| When | Action | Results in |
|---|---|---|
| | |  |
| You do not want to receive any in-app or the SMTP email notifications. | Click the **Unsubscribe from all Channels** option, and then click **Apply**. | The in-app channel and the SMTP channel selections for all server events are cleared and you are unsubscribed from both the channels. No channel is displayed as selected for subscription for all server events.  |
| You want to subscribe to the in-app notifications for all server events. | Click the **Subscribe to In App** option, and then click **Apply**. | The in-app channel for all server events is selected and you are subscribed to the in-app channel. The in-app notifications is displayed as selected for subscription for all server events. |
| You want to receive notifications as in-app and as SMTP email notifications. | Click the **Subscribe to all Channels** option, and then click **Apply**. | The in-app channel and the SMTP channel are displayed as selected for all server events and you are subscribed to both the channels. |

e.

**Results**

You completed the following tasks:

- Modified the notification subscriptions.
- Viewed the notifications displayed for you in the **USER** tab and **SYSTEM** tab.
- Unsubscribed from or subscribed to either the in-app channel or both the channels.

---

Related information

Managing notifications on page 307

## Managing email notifications

You can manage the email notifications by changing the subscriptions that are automatically enabled for you for all server events. You can view the server events that are automatically subscribed and then change the subscription for any or all events. For example, as a project owner, you can choose to be notified when members of the project add a repository or run a test in a team space but not be notified when projects are created in the team space.

**Before you begin**

- Ensured that the  or  has configured a Simple Mail Transfer Protocol (SMTP) server. See Configuring an SMTP server to manage email notifications on page 318.
-

**About this task**

The email notifications in the SMTP channel are automatically subscribed for you for all server events. You can view the server events that are subscribed in the **Notifications Settings** dialog. You can either unsubscribe from the SMPT channel or change your subscriptions for the server events for which you do not want the email notifications.

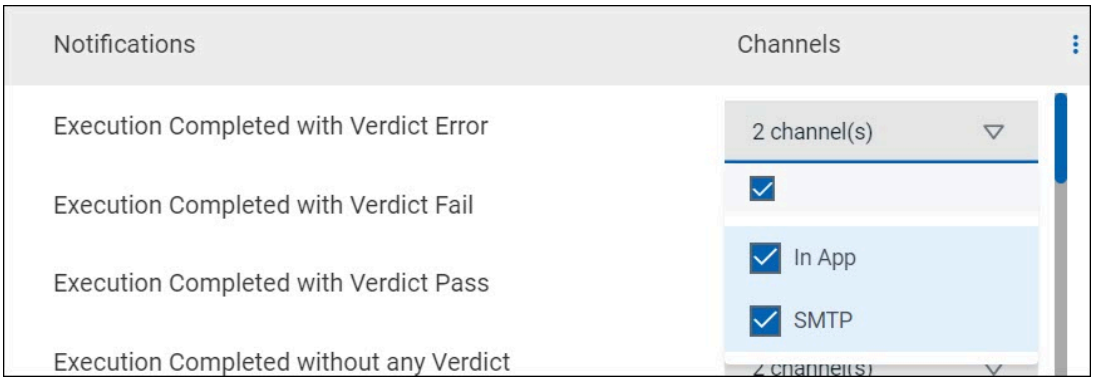
1. Select from the following actions that you want to perform:
    - Modify notification settings. Go to Step 2 on page 314.
    - Unsubscribe from the email notifications. Go to Step 3 on page 316.
2. Perform the following steps to modify the email notifications settings:
    a. Click the **User** icon on the menu bar.
    b. Click **Notification Settings**.

    The **Notification Settings** dialog is displayed.

    c. Change the language in which you want to receive the email notifications, if you do not want to use the default language.

    The default language is the locale language set on the computer that hosts .

You can notice that for each of the server events, both the channels are displayed as selected, which is the default selection unless you have modified the subscriptions.




d. Identify the server events for which you want to change the subscription.

e. Perform the actions indicated in the following table to clear the subscriptions for server events that you do not want to be notified and to select the SMTP subscription for the server events that you want to be notified in emails.

| When | Action |
|---|---|
| You do not want to receive the email notifications for a specific server event. | Perform the following steps:<br>i. Expand the channel list in the row of the server event.<br>ii. Clear the checkbox in the header row to clear the subscriptions to both channels.<br>iii. Click **Apply**.<br><br>You have unsubscribed from both channels and the email notifications for the server event are not sent to you when the event occurs. |
| You want to receive the email notifications for a specific server event. | Perform the following steps:<br>i. Expand the channel list in the row of the server event.<br>ii. Select any of the following actions:<br>▪ Clear the **In App** checkbox.<br><br>The **SMTP** selection is retained while the checkbox in the header row and for the **In App** options are cleared.<br><br>▪ Clear the checkbox in the header row to clear the subscriptions to both channels, and then select the **SMTP** checkbox.<br>iii. Click **Apply**. |

| When | Action |
|---|---|
| | You have unsubscribed from the in-app channel and retained the subscription for the email notifications for the server event. |

f. Go to Results on page 318.

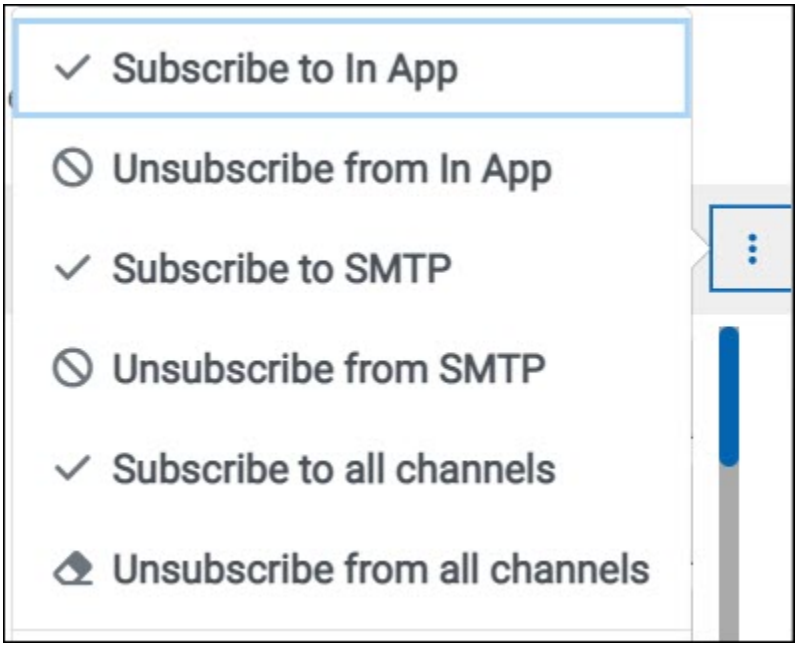
3. Perform the following steps to unsubscribe to the email notifications:

a.

b.

c. Click the **Open action menu** icon in the row of **Notifications**.

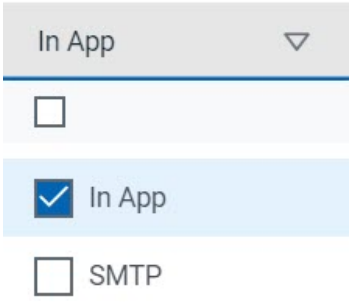
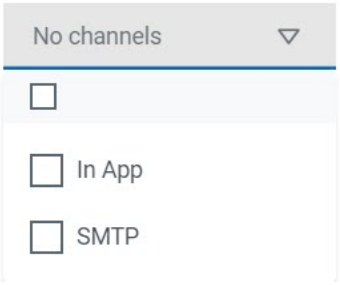
The following panel is displayed:




You can notice that the subscriptions to in-app, SMTP, and all channels are displayed as selected.

d. Perform the actions indicated in the following table for the option you want:

| When | Action | Results in |
|---|---|---|
| You do not want to receive any email notifications. | Click the **Unsubscribe from SMTP** option, and then click **Apply**. | The SMTP channel selection for all server events is cleared and you are unsubscribed from the SMTP channel. The **In App** option is displayed as selected |

| When | Action | Results in |
|---|---|---|
| | | for subscription for all server events.<br>  |
| You do not want to receive any in-app or the SMTP email notifications. | Click the **Unsubscribe from all Channels** option, and then click **Apply**. | The in-app channel and the SMTP channel selections for all server events are cleared and you are unsubscribed from both the channels. No channel is displayed as selected for subscription for all server events.<br>  |
| You want to subscribe to the email notifications for all server events. | Click the **Subscribe to SMTP** option, and then click **Apply**. | The SMTP channel for all server events is selected and you are subscribed to the in-app channel. The **In App** option is displayed as selected for subscription for all server events. |
| You want to receive notifications as in-app and as SMTP email notifications. | Click the **Subscribe to all Channels** option, and then click **Apply**. | The in-app channel and the SMTP channel are displayed as selected for all server events |

| When | Action | Results in |
|---|---|---|
| | | and you are subscribed to both the channels. |

e.

**Results**

You completed the following tasks:

- Modified the notification subscriptions.
- Unsubscribed from or subscribed to either the SMTP channel or both the channels.

---

Related information

Managing notifications on page 307

## Configuring an SMTP server to manage email notifications

Before you can receive email notifications about events that occur in Rational® Test Automation Server, you need to configure an existing Simple Mail Transfer Protocol (SMTP) server in Rational® Test Automation Server. When events occur in Rational® Test Automation Server, email notifications are sent to the registered users of Rational® Test Automation Server if they are subscribed to the SMTP notification channel.

**Before you begin**

You must have completed the following tasks:

- Ensured that an existing SMTP server is available to be connected to Rational® Test Automation Server.

**About this task**

The configuration works for emails that are supported by the SMTP server. To use this capability, you must enable IBM® Rational® Test Automation Server to connect to the SMTP server to send email notifications.

As a server administrator, you can configure email notifications so that other users can receive information about actions on the server projects.

1. Click the **Settings** icon from the menu bar.

   The **Notifications** page is displayed.

2. Click **Add Configuration**.
3. Provide the following details to configure the SMTP server:
   - A name to the configuration.
   - The host name and port number of the SMTP server.

**Note:** The default configuration uses a local SMTP server that listens to the default port 25. This SMTP server does not require any authentication.

- The email address of the sender. The email address is displayed in email notifications. The email notifications are sent to the users based on the notification settings.
- The encryption type for secure communications is when the SMTP server is on an external network. You can enable either **STARTTLS** or **SSL/TLS**.

**Note:** The SMTP server for the secure communication listens to the default port 465.

- Login credentials are required if the SMTP server requires authentication.

4. Click **Add**.

**Note:** The SMTP configuration is added after a valid connection is established with the SMTP server.

**Results**

You configured an SMTP server to connect with Rational® Test Automation Server and send out email notifications about the server events to users who have subscribed to the SMTP channel.

**What to do next**

You can modify the default subscriptions for the SMTP channel, see Managing email notifications on page 314.

---

Related information

Managing notifications on page 307

# Chapter 8. Troubleshooting Guide

This guide describes how to analyze and resolve some of the common problems that you might encounter while you work with Rational® Test Automation Server.

In addition to the list of problems that you might encounter, these resources are available to help you troubleshoot Rational® Test Automation Server problems. Known problems of Rational® Test Automation Server are documented in the form of individual technotes in the Support Knowledge Base. As problems are discovered and resolved, the knowledge base is updated and maintained with new information. By searching the knowledge base, you can quickly find workarounds or solutions to problems.

Troubleshooting documentation for Rational® Test Workbench

You can contact IBM® support if you are unable to troubleshoot the problem. Gather all the required background information and provide the details to IBM® support for investigation. For more information, see IBM Support.

## Troubleshooting issues

You can find information about the issues or problems that you might encounter while working with Rational® Test Automation Server. Details about issues, their causes and the resolutions that you can apply to fix the issues are described.

The troubleshooting issues are presented to you in the following tables based on where or when you might encounter these issues on Rational® Test Automation Server.


- Table 13: Troubleshooting issues: installation on page 320
- Table 14: Troubleshooting issues: server administration on page 322
- Table 15: Troubleshooting issues: resource monitoring on page 323
- Table 16: Troubleshooting issues: configuring test runs on page 324
- Table 17: Troubleshooting issues: test or stub runs on page 325
- Table 18: Troubleshooting issues: test results and reports on page 329


**Table 13. Troubleshooting issues: installation**

| Problem | Description | Solution |
|---|---|---|
| When you are installing the server software and you encounter errors in the scripts that are running. | At times, scripts might not appear to be running due to any of the following reasons: | You can complete any of the following tasks: |

**Table 13. Troubleshooting issues: installation (continued)**

| Problem | Description | Solution |
|---|---|---|
| | • Slow connection speeds.<br>• Insufficient CPU, memory, or disk resources.<br>• A firewall that was configured incorrectly is already enabled. | • To identify the issue, you can perform a diagnostic check by running the following command:<br>`journalctl -u k3s`<br>This command displays the log that you can use to check for the problem.<br>• Run the following command to see which pods are running and which pods are not running:<br>`kubectl get pods -A`<br>Run the following command to get details about a specific pod:<br>`kubectl describe pod -n <namespace> <pod name>`<br>• Follow the on-screen instructions to resolve the errors.<br>• Some issues can be solved by re-running the following script:<br>`ubuntu-init.sh` |
| The DNS is not working as expected. | | The DNS configuration that is used by the cluster can be displayed by using the following command:<br>`kubectl get cm -n kube-system coredns -ojsonpath="{.data.Corefile}"` |

**Table 13. Troubleshooting issues: installation (continued)**

| Problem | Description | Solution |
|---|---|---|
| | | The **forward** setting displays the nameservers that are used. For example, you might see the following in the `corefile`:<br>`.:53 {`<br>`  :`<br>`  forward . 1.1.1.1 8.8.8.8`<br>`  :`<br>`  }`<br>A script (`ubuntu-set-dns.sh`) is supplied for managing these values.<br>For example, to set the DNS values for the values shown in the previous example:<br>`sudo ubuntu-set-dns.sh --server 1.1.1.1 --server 8.8.8.8`<br>**Note:** If you do not use sudo in the command, the script runs but the configuration might be lost if the cluster is restarted.<br>To learn more about the behavior of the script, run the following command:<br>`ubuntu-set-dns.sh --help` |

**Table 14. Troubleshooting issues: server administration**

| Problem | Description | Solution |
|---|---|---|
| Less recently patched versions of OpenShift can give errors whenever OpenShift performs internal checks. | The errors are caused because OpenShift performs the internal checks that are invalid. | Apply the latest OpenShift patches when they become available. |

**Table 14. Troubleshooting issues: server administration (continued)**

| Problem | Description | Solution |
|---|---|---|
| For example, the following error is displayed:<br>`ValidationError(Route.spec.to): missing required field "weight"` | | If you do not want to apply the patches or cannot apply the patches, you can disable these checks in OpenShift by appending the following option to the helm command:<br>`--disable-openapi-validation` |
| If your user-realm role is changed when you are logged in a session, the changed role is not applied immediately or even after the browser is refreshed. | | You must log out of the session and log in again for the changed role to take effect. |
| You see the following message displayed on Rational® Test Automation Server:<br>`you can't request to join a project that has no owners` | You requested to join an orphaned project. The project is orphaned because the administrator deleted that user ID from the user registry.<br>That user registry can be either the local Keycloak user registry or an external user registry, for example LDAP that is federated with the local Keycloak registry.<br>That user ID can be deleted by the administrator in cases such as the user leaving the project or company. | Ask your administrator to take ownership of the project, and then add you as a project owner or member. |

**Table 15. Troubleshooting issues: resource monitoring**

| Problem | Description | Solution |
|---|---|---|
| You are not able to add a Prometheus server as a Resource Monitoring source. | The cause might be that you have not installed the Prometheus server at the time of server installation. | Verify that the Prometheus server was installed in Helm at the time of server installation. See Installing the server software on Ubuntu by using k3s on page 42. If not, consult |

**Table 15. Troubleshooting issues: resource monitoring (continued)**

| Problem | Description | Solution |
|---|---|---|
| | | your cluster administrator to get the Promethues server installed and configured. |

**Table 16. Troubleshooting issues: configuring test runs**

| Problem | Description | Solution |
|---|---|---|
| When you configure a run of a schedule that matches the following conditions:<br>• The schedule has two user groups configured to run on static agents when the schedule was created in Rational® Performance Tester V10.1.<br>• One of the user groups is disabled and the asset is committed to the remote repository.<br><br>Both the static agents are displayed as available for the test run in the **Location** tab of the **Execute test asset** dialog box when only one agent that is configured for the user group must be available. | The cause might be because of the following reasons:<br>• The schedule was created in Rational® Performance Tester V10.1.<br>• The user group that is disabled is not removed or deleted from the test resources.<br>• The agent configured on the disabled user group is already added as an agent to the server project and is available for selection. | To resolve the problem, select from either of the following methods:<br>• By using Rational® Performance Tester V10.1.1.<br><br>Perform the following steps:<br>1. Open the schedule in Rational® Performance Tester V10.1.1.<br>2. Save the schedule and the project.<br>3. Commit your test asset to the remote repository.<br>4. Proceed to configure a run for the schedule on Rational® Test Automation Server V10.1.1.<br>• By using Rational® Performance Tester V10.1.<br><br>Perform the following steps:<br>1. Select the disabled user group.<br>2. Click **Remove**.<br>3. Save the schedule and the project. |

**Table 16. Troubleshooting issues: configuring test runs (continued)**

| Problem | Description | Solution |
|---|---|---|
| | | 4. Commit your test asset to the remote repository.<br>5. Proceed to configure a run for the schedule on Rational® Test Automation Server V10.1.1. |

**Table 17. Troubleshooting issues: test or stub runs**

<table>
<tr><th>Problem</th><th>Description</th><th>Solution</th></tr>
<tr>
<td>You encounter any of the following issues:<ul><li>When many tests are run simultaneously on the default cluster location and you observe the following issues:<ul><li>Out of memory errors.</li></ul></li></ul></td>
<td>The issue is seen when any of the following events occur:<ul><li>Many tests are run in parallel.</li><li>The memory that is used by the tests during the test run exceeds the allocated default memory of 1 GB.</li><li>The default memory of the</li></ul></td>
<td>To resolve the problem, you can increase the resource allocation for test runs.<br><br>You can enter arguments in the Java Arguments field in the Advanced settings panel of the Execute test asset dialog box when configuring a test run.<br><br>Important: The memory settings that you configure for a test run is persisted for the test when ever you run it. You must use this setting judiciously. Configuring all tests for an increased memory limit might affect subsequent test runs or cause other memory issues when tests run simultaneously.<br><br>You can increase the resource allocation for test runs by using any of the following arguments:
<table>
<tr><th>For...</th><th>Enter the argument...</th><th>Result</th></tr>
<tr><td>Specifying the memory limit of the init container .</td><td><code>-Dexecution.init.resource.memory.limit</code></td><td>Change the memory limit of the init container from the default value of 1 Gi to 1024Mi.</td></tr>
<tr><td>Configuring a larger memory request for the</td><td><code>-Dexecution.init-.resource.memory.request=1024Mi</code></td><td>Increases the initial memory request for the init container from the</td></tr>
</table>
</td>
</tr>
</table>

**Table 17. Troubleshooting issues: test or stub runs (continued)**

<table>
<tr><th>Problem</th><th>Description</th><th>Solution</th></tr>
<tr><td>
<ul>
<li>◦ Observe that the test runs are slow with a high CPU usage.</li>
<li>◦ The Kubernetes pods are getting evicted.</li>
</ul>
• When you run an AFT suite that contains multiple Web UI tests and you observe the following issues:
</td><td>
container is not adequate for the test run.<br>• Pods are evicted due to low node memory.
</td><td>
<table>
<tr><th>For...</th><th>Enter the argument...</th><th>Result</th></tr>
<tr><td>init container to avoid pod eviction.</td><td></td><td>default value of 64Mi to 1024Mi.</td></tr>
<tr><td>Specifying the cpu request for the init container.</td><td><code>-Dexecution.init.resource.cpu.request</code></td><td>Increases the cpu limit of the init container from the default value of 50m to 60m.</td></tr>
<tr><td>Specifying a maximum heap size for the test run.</td><td><code>-Xmx4g</code></td><td>Increases the allotted 1 GB memory to 4 GB.</td></tr>
<tr><td>Specifying the memory limit of the container explicitly for the test run.</td><td><code>-Dexecution.resource-.memory.limit=<custom_memory_value>Gi</code><br>Note: You must enter the value for the memory limit that you want in <custom_memory_value>.</td><td>Increases the allotted memory to any value you specify.<br>For example, if you want to set the memory limit to 5 GB, the argument can be:<br><code>-Dexecution.resource-.memory.limit=5Gi</code>.<br>The argument enables the memory to be set to 5 GB and overrides the default memory size.</td></tr>
<tr><td>Specifying the memory request for the container used by the test run.</td><td><code>-Dexecution.resource-.memory.request</code></td><td>Increases the memory request from the default value of 64Mi to 1Gi.</td></tr>
</table>
</td></tr>
</table>

**Table 17. Troubleshooting issues: test or stub runs (continued)**

<table>
<tr><th>Problem</th><th>Description</th><th>Solution</th></tr>
<tr><td>◦ Error stating that the browser might not be installed or the browser version is unsupported.<br>◦ Error stating multiple random timeouts or an in-</td><td></td><td>
<table>
<tr><th>For...</th><th>Enter the argument...</th><th>Result</th></tr>
<tr><td>Specifying the cpu request for the container used by the test run.</td><td><code>-Dexecution.resource-.cpu.request</code></td><td>Increases the cpu limit of the container from the default value of 50m to 60m.</td></tr>
</table>
</td></tr>
</table>

**Table 17. Troubleshooting issues: test or stub runs (continued)**

| Problem | Description | Solution |
|---|---|---|
| ternal error. | | |
| You are not able to run the Istio stubs from the **Execution** page. | The cause might be that you have not enabled the service virtualization via Istio at the time of server installation. The default configuration does not enable service virtualization via Istio. | Contact your cluster administrator or if you have the privileges, configure Helm as follows:<br>• For Multi-tenant clusters<br>If the cluster is shared and the product may only virtualize services running in specific namespaces then add the following parameter to the Helm install:<br>--set execution.istio.enabled=true<br>Then enable service virtualization in specific namespaces using this command:<br>kubectl create rolebinding istio-virtualization-enabled -n {namespace} --clusterrole={my-ots}-execution-istio-test-system --serviceaccount=test-system:{my-ots}-execution<br>**Note:** Uninstalling the chart does not clean up these manually created role bindings.<br>• For Single-tenant clusters<br>If the cluster is not shared and the product may virtualize any service running in the whole cluster then add the following parameters to the Helm install:<br>--set execution.istio.enabled=true<br>--set execution.istio.clusterRoleBinding.create=true |
| | The cause might be that the fully qualified domain name is not specified in the **Host** field for the stub | Verify and ensure to add the fully qualified domain name of the server in the **Host** field when the physical transport for the stub is configured in Rational® Integration Tester. |

**Table 17. Troubleshooting issues: test or stub runs (continued)**

| Problem | Description | Solution |
|---|---|---|
| | when it was created. | |
| When HTTP stubs on Rational® Test Automation Server and called via the HTTP proxy, the calls fail with HTTP 404 errors. | | You must enable service virtualization through Istio at the time of installation of the server software. See Installation of the server software on page 25. |

**Table 18. Troubleshooting issues: test results and reports**

| Problem | Description | Solution |
|---|---|---|
| You are not able to view the Jaeger traces for the tests you ran. | The cause can be as follows:<br>• Might be that you have not installed Jaeger at the time of server installation.<br>• The Jaeger trace is not supported for the particular test you ran. | Check for any of the following solutions:<br>• Verify that Jeager was installed in Helm at the time of server installation. See Installing the server software on Ubuntu by using k3s on page 42. If not, consult your administrator to get Jeager installed and configured.<br>• Verify that the tests you ran are supported for Jaeger traces. See Test results and reports overview on page 200.<br>• Verify if you provided the program variables when you configured the test run. See Considerations for using Jaeger traces in reports on page 104. |

# Security Considerations

This document describes the actions that you can take to ensure that your installation is secure, customize your security settings, and set up user access controls.


- Enabling secure communication between multiple applications on page cccxxx
- Ports, protocols, and services on page cccxxx
- Customizing your security settings on page cccxxx
- Setting up user roles and access on page cccxxxi


## Enabling secure communication between multiple applications

The majority of communications are sent over TLS to port 443 (see Ports, protocols, and services on page cccxxx). During the installation, an X.509 certificate is generated for the user provided DNS name, which is used to connect to the server. This certificate is self-signed and hence untrusted by other applications.

This self-signed certificate must be replaced by a certificate signed by a certificate authority trusted by your organization. For more information, see X.509 Certificate User Authentication in the Keycloak documentation.

For information about how the self-signed certificate was created, see the `ssl.sh` file in the `<install-directory>/prepare/` directory.

## Ports, protocols, and services

TCP port 443 is used by the majority of communications with the server.

The port 7085 is the default port for communications with agents registered with Rational® Test Automation Server.

The ports starting from 7085, are used in pairs such as 7085 and 7086, and are allotted for the Schedule that is executed first. The next Schedule is allotted the next pair (7087,7088), and so on for the Schedules that are running simultaneously.

You must open the required ports in pairs for each of the Schedules that you want to run simultaneously.

## Customizing your security settings

You can customize your security settings through user registration.

### User registration

By default, users can sign up themselves with the server. In some environments, this self sign-up might be undesirable. It can be changed by switching off user registration. For more information, see User Registration in the Keycloak documentation.

By default, user email addresses are not verified. This verification must be enabled in production environments. For more information, see Email settings on page 60.

## Setting up user roles and access

You can manage user roles and access through single sign on (SSO) and administration only accounts.

**Single sign-on**

By default, Keycloak manages users and passwords locally. In production environments, it is normally appropriate to use single sign-on. For more information, see LDAP user administration on page 60.

**Administration only accounts**

Users in the Administrator group can discover all projects stored on the server (including private ones) and assign themselves and others roles in those projects.

For this reason, users who use the server to perform both administration and non-administration tasks must have two different accounts, one for each purpose. For more information, see Default user administration on page 58.

# Notices

This document provides information about copyright, trademarks, terms and conditions for the product documentation.

© Copyright IBM Corporation 2000, 2016 / © Copyright HCL Technologies Limited 2016, 2020

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

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

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

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

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM® Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing*
*Legal and Intellectual Property Law*
*IBM Japan Ltd.*
*19-21, Nihonbashi-Hakozakicho, Chuo-ku*
*Tokyo 103-8510, Japan*

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

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

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

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

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM® under terms of the IBM® Customer Agreement, IBM® International Program License Agreement or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

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

Statements regarding the future direction or intent of IBM® are subject to change or withdrawal without notice, and represent goals and objectives only.

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

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM®, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM®, therefore, cannot guarantee or imply reliability,

serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM® shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work must include a copyright notice as follows:
© (your company name) (year).
Portions of this code are derived from IBM Corp.  Sample Programs.
© Copyright IBM Corp. 2000, 2020.

# Trademarks

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

# Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

### Applicability

These terms and conditions are in addition to any terms of use for the IBM® website.

### Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM®.

### Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM®.

### Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM® reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM®, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM® MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

## IBM Online Privacy Statement

IBM Software products, including software as a service solutions, (Software Offerings) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offerings use of cookies is set forth below.

This Software Offering does not use cookies or other technologies to collect personally identifiable information.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM Privacy Policy at http://www.ibm.com/privacy and IBM Online Privacy Statement at http://www.ibm.com/privacy/details in the section entitled Cookies, Web Beacons and Other Technologies, and the IBM Software Products and Software-as-a-Service Privacy Statement at http://www.ibm.com/software/info/product-privacy.

# Index


**A**

accessibility
  IBM
  Rational
  Test Automation Server
    disability 24
    keyboard 24
Accessing secrets 295
administration default security
  IBM
  Rational
  Test Automation Server
  58
administration ldap security
  IBM
  Rational
  Test Automation Server
  60

**B**

backing up and restoring user data
  IBM
  Rational
  Test Automation Server
  49, 50, 51

**C**

Change
  Classification 302
  Dataset 302
  Encrypt 302
Change Classification
  Dataset 94
Configure
  Dataset 94
Create
  Classification 302
Create Dataset 83
Creating secrets 295

**D**

Dataset 94
datasets
  encrypting 91
  removing encryption 92
Delete
  Classification 302
Deleting secrets 295

**E**

Edit
  Classification 302
Editing secrets 295

**G**

getting started
  guide 16
Grant
  access 302
  Classification 302
guide
  getting started 16

**I**

IBM
Rational
Test Automation Server
  accessibility 24
  backing up and restoring user data 49, 50, 51
  default security administration 58
  disability 24
  keyboard 24
  ldap security administration 60
  managing account settings 71
  repository considerations 280
  test assets 280
IBM
Rational
Test Automation Server
project overview
  test assets statistics 289

**M**

Manage encrypted dataset 302
managing account settings
  IBM
  Rational
  Test Automation Server
  71
managing project secrets
  Rational
  Test Automation Server
  295

**O**

Offline tokens 277
Offline user token 277
offline user tokens
  Rational
  Test Automation Server
  277
overview
  server 16

**R**

Rational
Test Automation Server
  managing secrets 295
  offline user tokens 277
Remove
  Classification 302
  Dataset 302
  Encrypt 302
repository considerations
  IBM
  Rational
  Test Automation Server
  280
Revoke
  access 302
  Classification 302

**S**

Secrets
  accessing 295
  creating 295
  deleting 295
  editing 295
  viewing 295
security considerations
  overview cccxxx
server
  overview 16
  task flows 20
Set
  Current row 94
  Dataset 94

**T**

task flows
  server 20
test assets
  IBM
  Rational
  Test Automation Server
  280
test assets statistics
  IBM
  Rational
  Test Automation Server
  project overview
  289
tests
  datasets 91, 92

**V**

View
  Dataset 94, 94
  Encrypt 94
Viewing secrets 295