IBM

IBM® Rational® Test Automation Server

10.1 Documentation

June 2020

Special notice

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

Contents

Chapter 1. Release Notes....6
Description....6
What's new....6
Installing Rational® Test Automation Server....9
Known issues....9
Contacting IBM Rational Software Support....10
Chapter 2. System Requirements....11
Supported hardware....11
Operating Systems....12
Prerequisites....13
Supported software....16
Chapter 3. Getting started guide....19
Server overview....19
Supported versions of assets and resources....20
Tests supported on Rational Test Automation Server....21
Task flow: Test runs and results....22
Task flow: Server access from desktop clients....24
Accessibility features....26
Chapter 4. Administrator guide....27
Installation of Rational® Test Automation Server....27
Installation of server software on Red Hat OpenShift....27
Installation of server software on Ubuntu using microk8s....34
Backing up and restoring the user data....43
Backing up and restoring the user data on Red Hat OpenShift....43
Backing up and restoring the user data on Ubuntu....45
Uninstallation of the server software....47
Uninstalling the server software....47
Uninstalling the server software....48
Configuration of the server software....49
User administration....49
Certificate authority: Importing and extending lists....62
Copying third-party application Jars to Kubernetes....65
Licensing....67
Floating licenses....67
Chapter 5. Test Author guide....69
Datasets overview....69
Creating a dataset....69
Dataset editing....70
Publishing a dataset....75
Viewing a dataset....76
Deleting a dataset....78
Chapter 6. Test Execution Specialist guide....80
Prerequisites to running tests....80
Test run considerations for AFT Suites....80
Test run considerations for API Suites....81

Test run considerations for schedules.... 84
Managing Docker hosts....84
Test run considerations for running tests on remote Docker hosts....85
Setting up a remote Docker....85
Setting up a secure remote Docker.... 87
Copying third-party application Jars to a remote Docker host....88
Registering a remote Docker....90
Adding a remote Docker host.... 92
Editing configurations....93
Deleting a remote Docker.... 95
Unregistering a remote Docker....95
Test run configurations....96
Configuring a test for a quick run....97
Configuring a run for an AFT Suite....98
Configuring a run for an API Suite....104
Configuring a run for Compound Tests that contains traditional HTML tests....110
Configuring a run for Compound Tests that contains Web UI tests.... 116
Configuring a run for Compound Tests that contains performance tests....122
Configuring a run for Rate Schedules or VU Schedules....128
Management of running tests.... 134
Viewing the progress of running test assets....134
Checking logs.... 136
Resetting the configuration settings for a test run....137
Stopping a test run....139
Canceling a scheduled test run....140
Management of virtualized services.... 141
Starting a stub.... 142
Viewing routing rules.... 144
Viewing stub usage statistics.... 146
Stopping a stub.... 149
Test results.... 151
Test results and reports overview....151
Resource Monitoring service....161
Monitoring metrics exposed by an OpenMetrics exporter....163
Monitoring metrics collected by a Prometheus server.... 167
Monitoring host resources....171
Resource Monitoring agents.... 175
Integrating with other applications....183
Integration plugin compatibility matrix....183
Integration with Azure DevOps....184
Integration with Jenkins....191
Integration with UrbanCode Deploy.... 196
Managing access to the server 202
Chapter 7. Test Manager guide.... 205
Test assets and a server project....205
Repository considerations for a server project....205
Adding details to a server project....207

Adding repositories to a server project....207
Secrets configuration 209
Adding users to a server project....209
Becoming a project member....210
Managing access to server projects....210
Archiving or unarchiving server projects....212
Deleting server projects....213
Project overview....214
Working with Git repositories....215
Adding repositories to a server project....216
Protecting API test assets by using secrets....218
Managing secrets collections....219
Granting access to members or member roles....222
Creating a secret in a secrets collection....221
Managing secrets....223
Managing an encrypted dataset....225
Creating a classification....225
Editing or deleting a classification....226
Moving an encrypted dataset to another classification....228
Removing a dataset from the classification....228
Granting classification access to members or members roles....229
Managing notifications....230
Managing in-app notifications....231
Managing email notifications....237
Chapter 8. Troubleshooting Rational® Test Automation Server....243
Troubleshooting issues....243
Security considerations....ccl
Notices....cclii
Index....256

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 9
- Known issues on page 9
- Contacting IBM Rational Software Support on page 10

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 19.

What's new

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

- **Installation**

  Rational® Test Automation Server can only be installed on the following platforms:

  - Red Hat OpenShift 4.2+
  - Ubuntu (using MicroK8s)

  For more information, see Installation of Rational Test Automation Server on page 27.

- **Server Licensing**

  You can purchase floating licenses for Rational® Test Automation Server and install the licenses on Rational License Key Server. The administrator must configure the purchased floating licenses in Rational® Test Automation Server.

  For more information, see Floating licenses on page 67.

- **Backup and restore of user data**

  You can back up and restore the user data in Rational® Test Automation Server V10.1.0. You can also migrate the backed-up user data from the previous versions of Rational® Test Automation Server to Rational® Test Automation Server V10.1.0.

For more information, see Backing up and restoring the user data on page 43.

- **Test runs**
  - **Running AFT Suites on remote agents**

    You can now select an agent added to your project on Rational® Test Automation Server to override the agent configured in the AFT XML file in the test asset.
  - **Selecting test resources**

    You can now select either the current or previous versions of the test resources that you want to run while you are configuring the test run.
  - **Resetting test run configurations**

    You can now reset the settings you configured for a test run when you are either initiating or repeating the test run. For more information, see Resetting the configuration settings for a test run on page 137.
  - **Support for running an HTML test in a Compound Test on Rational® Test Automation Server**

    You can now, run an HTML test as a compound test on Rational® Test Automation Server and choose to override the configured browser while configuring the test. For more information, see Configuring a run for Compound Tests that contains traditional HTML tests on page 110.
- **Test results**
  - **Navigating from a result to the associated Jaeger trace**

    Now, test logs are delivered in a Jaeger format by default for executed compound tests and schedules. You can still set a Program Argument to re-enable the traditional test log format when you run the tests. For more information, see Test results and reports overview on page 151.
  - **Relaunching an execution from the Results view**

    From the Results view, you can re-execute a test with the same parameters as the ones that you previously set for the previous test run. You can also re-execute multiple tests at a time with the same parameters as the initial ones. For more information, see Re-executing tests from results on page 159.
  - **Controlling Resource Monitoring sources**

    Now you can use label matching to control what Resource Monitoring sources are collected during the execution of a Performance schedule. For more information, see Controlling Resource Monitoring sources in a schedule.
  - **Functional Test reports**

    A Unified Report is generated as a result of a functional test execution. For more information, see Test results and reports overview on page 151.

- **Resource monitoring**
  - **Querying metrics**

    Now the Resource Monitoring service can query metrics from a target monitored by a running Prometheus server with default or customized queries. For more information, see Monitoring metrics collected by a Prometheus server on page 167.

  - **Scrapping metrics**

    The Resource Monitoring service can scrap metrics exposed by an OpenMetrics exporter through metric counters. For more information, see Monitoring metrics exposed by an OpenMetrics exporter on page 163.

- **Projects**
  - **Selecting branches**

    In earlier versions, branch selection was configured on the Repositories page and applied globally to all users. Now, each user is presented with a Branch field that lists all the branches of the repositories added to your project. The list is available in the Execution, Datasets and Overview pages Rational® Test Automation Server. For more information, see Test resource access by using the global branch.

  - **Creating secret collections**

    Members with the Tester role assigned in a project, can now create secrets collections. For more information, see Protecting API test assets by using secrets on page 218.

  - **Project repository**

    You can now add a repository with multiple folders to a single project. With this improvement, each project is now restricted to a single copy of a given repository.

- **Integrations**
  - **Rational® Test Automation Server integration with Azure DevOps**

    You can now integrate Rational® Test Automation Server with Azure DevOps by using the IBM® Rational® Test Workbench extension to run tests as tasks in Azure DevOps pipelines. For more information, see Integration with Azure DevOps on page 184.

  - **Rational® Test Automation Server integration with UrbanCode Deploy**

    You can now integrate Rational® Test Automation Server with UrbanCode Deploy (UCD) by using the Rational® Test Automation Server UCD plugin to run tests on the UCD server. For more information, see Integration with UrbanCode Deploy on page 196.

- **Service virtualization**
  - **Running stubs**

    You can now run stubs that you create in Rational® Integration Tester, from the **Execution** page on Rational® Test Automation Server. You can run IBM® WebSphere® MQ stubs and HTTP stubs on

Rational® Test Automation Server. For more information, see Management of virtualized services on page 141.

- **Running stubs for Istio services**

  After you create stubs for Istio in Rational® Integration Tester that virtualize services in a Kubernetes cluster, you can run those stubs on Rational® Test Automation Server. The stubs can virtualize services that use the Istio service mesh for the following types of requests:
  - 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.
- **Stub usage statistics**

  After you run stubs on Rational® Test Automation Server, you can view the stub usage statistics from the **Stub Usage** page. For more information, see Viewing stub usage statistics on page 146.
- **Routing rules**

  You can view the details of the intercepts, proxies, or stubs with routing rules defined that are connected to Rational® Test Automation Server, from the **Routing Rules** page. For more information, see Viewing routing rules on page 144.

- **Test authoring**
  - **Using Excel Data Sources**

    Excel Data Sources defined in Rational® Integration Tester projects can now be used as Datasets in your Rational® Test Automation Server projects. Note that Excel Data Sources that define a specific worksheet to use are ignored.
  - **Authoring datasets**

    Now, you can create, edit, or delete a dataset in Rational® Test Automation Server. You can also publish the created dataset to Git repository so that other members of the project can use your dataset in their test asset run. For more information, see Datasets overview on page 69.

Installing Rational® Test Automation Server

For instructions about installing the software, see Installation of Rational Test Automation Server on page 27.

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

You can find the detailed system requirements for Rational® Test Automation Server V10.1.0.

For information about hardware and software compatibility, go to the Software Product Compatibility Reports site and complete the on-screen instructions to generate the detailed system requirements document for Rational® Test Automation Server.

Supported hardware

Rational® Test Automation Server V10.1.0 requires the following hardware, organized by type of hardware, deployment units, or components.

Installable software by Operating System

- Linux on page 11
- Windows on page 11

Linux

Hardware	Components Server	Requirement	Applicable operating systems
Disk space	Rational® Test Automation Server	64 GB	All supported Linux operating systems
Memory	Rational® Test Automation Server	12 GB	All supported Linux operating systems
Processor	Rational® Test Automation Server	4 CPUs	All supported Linux operating systems

Windows

Hardware	Components Server	Requirement	Applicable operating systems
Disk space	Rational® Test Automation Server	64 GB	All supported Windows operating systems

Hardware	Components Server	Requirement	Applicable operating systems
Memory	Rational® Test Automation Server	12 GB	All supported Windows operating systems
Processor	Rational® Test Automation Server	4 CPUs	All supported Windows operating systems

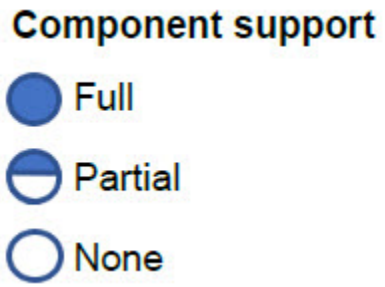
Operating Systems

The Operating systems section specifies the operating systems that Rational® Test Automation Server V10.1.0 supports, organized by operating system family.

Operating system families

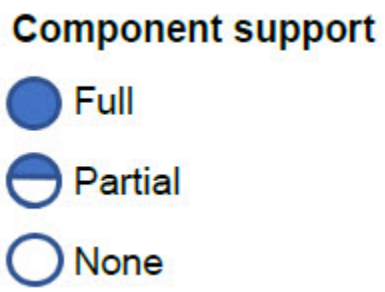
- Linux on page 12
- Windows on page 13

Linux



Operating system	Operat-ing system minimum	Hardware	Bitness	Product minimum	Compo-nents Server	Notes?
Red Hat Enterprise Linux (RHEL) 8	Base	x86-64	32, 64-Ex-ploit	10.1	●	No
Ubuntu 18.04 LTS	Base	x86-32	32	10.1	●	No

Windows



Operating system	Operating system minimum	Hardware	Bitness	Product minimum	Components Server	Notes?
Windows 10 Pro	Base	x86-64	32, 64-Exploit	10.1	●	Yes on page 13
Windows Server 2016 Standard Edition	Base	x86-64	32, 64-Exploit	10.1	●	Yes on page 13

Note: Rational® Test Automation Server for Windows is provided for evaluation purposes. It is intended for nonproduction use, for example, trials and demonstrations. It is neither supported nor recommended for production use due to insufficient stability and reliability of Docker for Windows while running Linux images at the time of release. Rational® Test Automation Server is tested on both Windows 10 and Windows Server 2016 and it operates correctly. However, it is suitable and supported for non-production use only. If native Linux is unavailable, customers may run the Rational® Test Automation Server for production use on a Linux virtual machine on their Windows host, provided that the Linux virtual machine is running a supported version of Linux.

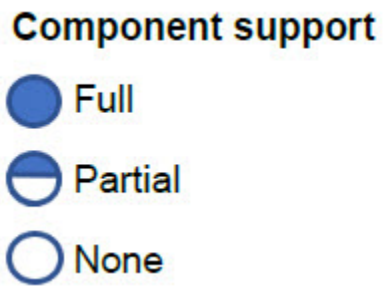
Prerequisites

The prerequisites section specifies the capabilities that Rational® Test Automation Server V10.1.0 requires,and the prerequisite products that can be used to fulfill those capabilities.

Capabilities

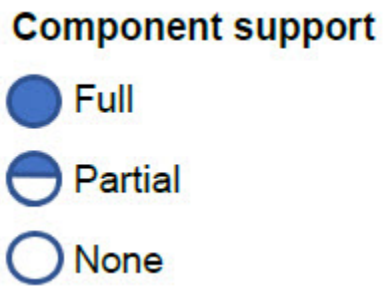
- Virtualization Management on page 14
- Web Browsers on page 14

Virtualization Management



Prerequisite	Version	Prerequisite minimum	Product minimum	Components Server	Operating system restrictions?	Notes?
Docker Community Edition (CE)	17.03 and future fix packs	17.03	10.1	●	No	No
	18 and future fix packs	18	10.1	●	No	No
	19.03 and future fix packs	19.03	10.1	●	No	No
Docker Compose	1.24 and future fix packs	1.24	10.1	●	No	No

Web Browsers



Prerequisite	Version	Prerequisite minimum	Product minimum	Components Server	Operating system restrictions?	Notes?
Apple Safari	13 and future fix packs	13	10.1		No	No
Google Chrome	78 and future fix packs	78	10.1		No	No
Google Chrome	79 and future fix packs	79	10.1		No	No
Google Chrome	80 and future fix packs	80	10.1		No	No
Google Chrome	81 and future fix packs	81	10.1		No	No
Google Chrome	83 and future fix packs	83	10.1		No	No
Microsoft Edge	44 and future fix packs	44	10.1		No	No
Microsoft Edge Chromium	Any version and future fix packs	Any version	10.1		No	[1]

1. 80, 81

Prerequisite	Version	Prerequisite minimum	Product minimum	Components Server	Operating system restrictions?	Notes?
Microsoft Internet Explorer	11 and future fix packs	11	10.1		No	No
Mozilla Firefox	69 and future fix packs	69	10.1		No	No
Mozilla Firefox	70 and future fix packs	70	10.1		No	No
Mozilla Firefox	72 and future fix packs	72	10.1		No	No
Mozilla Firefox	73 and future fix packs	73	10.1		No	No
Mozilla Firefox	74 and future fix packs	74	10.1		No	No
Mozilla Firefox ESR	68 and future fix packs	68	10.1		No	No

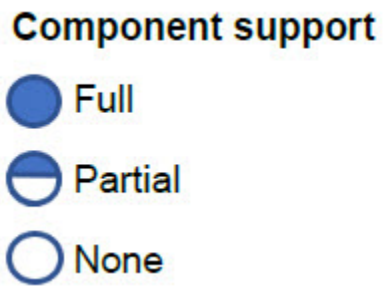
Supported software

The supported software section specifies the additional software that Rational® Test Automation Server V10.1.0 supports.

Capabilities

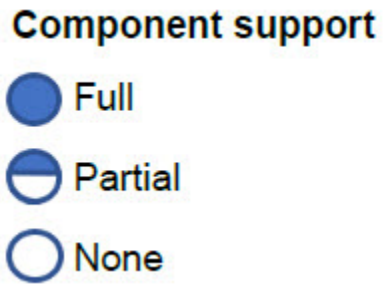
- Development Tools on page 17
- Integration Middleware on page 17

Development Tools



Supported software	Version	Supported software minimum	Product minimum	Compo-nents Server	Operating system re-strictions?	Notes?
Git	2.19.0 and future fix packs	2.19.0	10.1	●	No	No
Git	2.22.0 and future fix packs	2.22.0	10.1	●	No	No

Integration Middleware



Supported software	Version	Supported software minimum	Product minimum	Compo-nents Server	Operating system re-strictions?	Notes?
Rational® Functional Tester	10.1	10.1	10.1		No	No

Chapter 3. Getting started guide

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

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. You can also fabricate sample data to perform tests by using the data generator supported in Rational® Test Automation Server.

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.

Resource	Supported	Not supported
Assets or resources created in IBM® Rational® Performance Tester	Assets or resources created in Rational® Performance Tester which are of versions earlier or the same as Rational® Test Automation Server	Assets or resources created in any later version of Rational® Performance Tester that is later than the version of Rational® Test Automation Server.
IBM® Rational® Performance Tester Agents	Rational® Performance TesterAgents which is the same version as Rational® Test Automation Server	
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 84.
HTTP proxies	Installation of the proxy component from Rational® Test Control Panel which is the same version as Rational® Test Automation Server.	

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) Suite • Compound Test that contains Web UI tests • Compound Test that contains Traditional HTML tests	Traditional non-HTML tests. An independent Web UI test and a mobile test outside of a Compound Test or AFT Suite.
IBM® Rational® Integration Tester	API Suite	Test Suites that have scenarios with references satisfied by local stubs. Test Suites that have tests with subscribe actions that operate in the watch mode. 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 Test that contains performance tests • Rate Schedule • VU Schedule	A single test script of any test extension. Tests that belong to 32-bit test extensions and SOA Quality included in VU Schedule, Rate Schedule, and Compound Test.

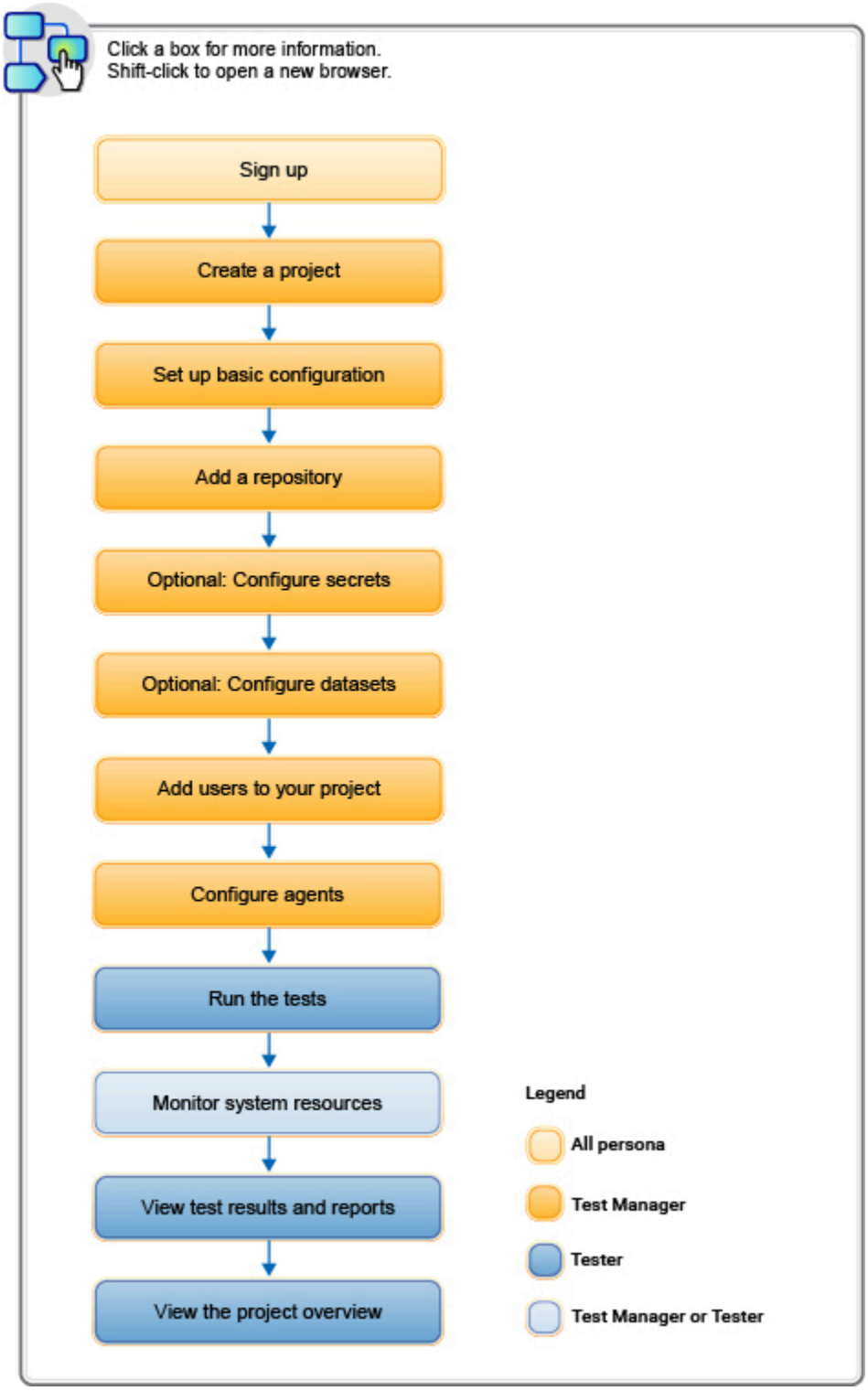
Related information

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.

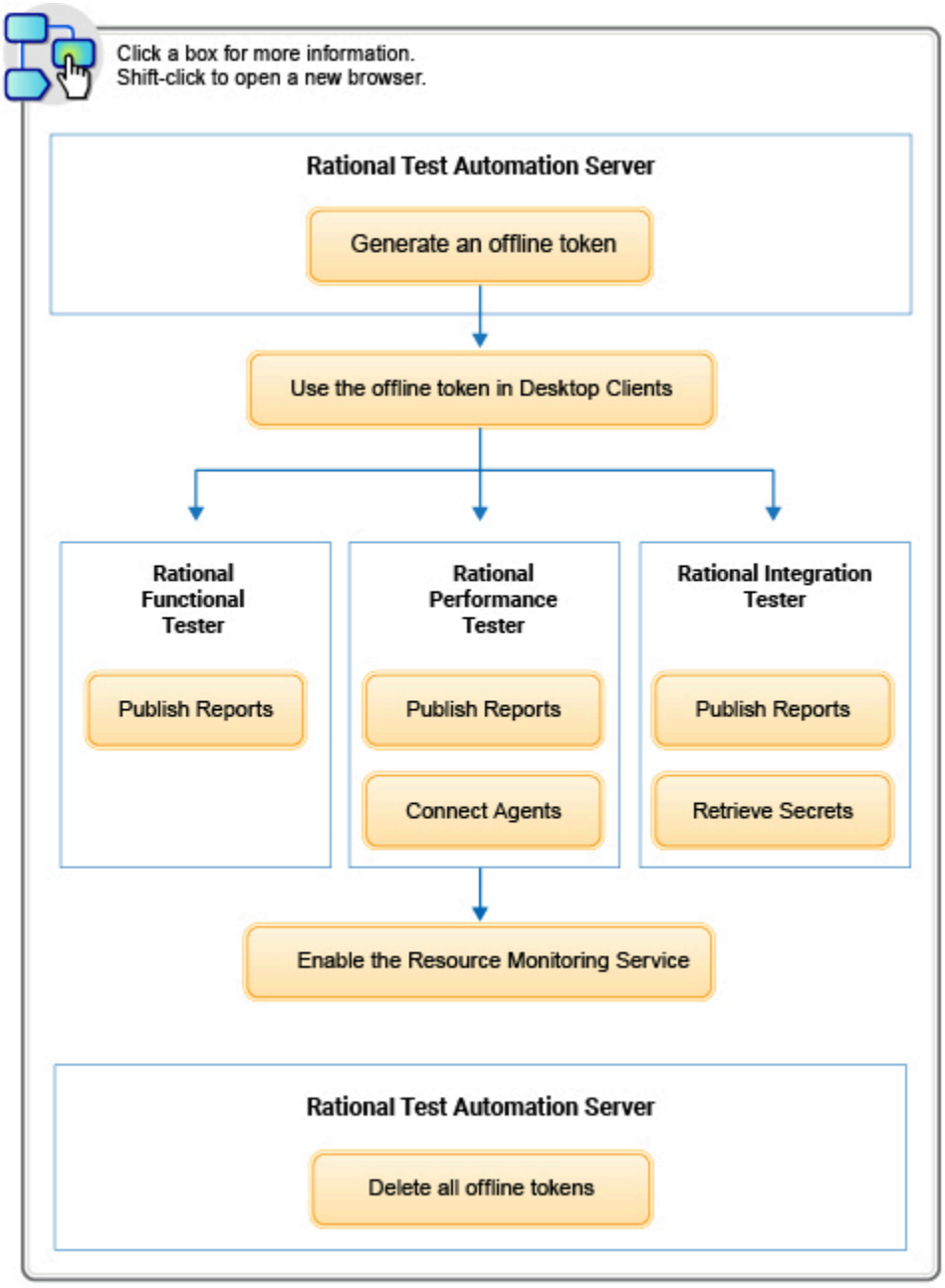
You can click the links to get more information about the tasks. The diagram also provides the tasks to be performed by personas. Rational® Test Automation Server supports user roles.



1. Default user administration on page 50
2. Test assets and a server project on page 205
3. Adding details to a server project on page 207
4. Adding repositories to a server project on page 207
5. Protecting API test assets by using secrets on page 218
6. Managing an encrypted dataset on page 225
7. Adding users to a server project on page 209
8. Adding an agent to the project for running tests
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 202
2. Test results and reports overview on page 151
3. Test results and reports overview on page 151
4. Test results and reports overview on page 151
5. Retrieving secrets
6. Enablement of Resource Monitoring services for a schedule
7. Managing access to IBM Rational Test Automation Server on page 202
8. Installing Rational Performance Tester gent

Related information

Managing access to server projects on page 210

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.

Accessibility compliance

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.

Accessing UI elements

Rational® Test Automation Server supports navigation in the UI by using different methods such as a mouse, keyboard, or touchpad.

You can use the keyboard keys such as **Tab**, arrow keys such as **UP**, **DOWN**, **LEFT**, and **RIGHT** to navigate to the different pages in the **Navigation** pane or to the different action labels in the right pane on the UI.

Chapter 4. Administrator guide

This guide, which is intended for an administrator, describes how to install IBM® Rational® Test Automation Server software. After you install the software, you can administer the server. Administration tasks can include License configuration, User management, Security, Memory and disk usage management, Backup and restore user data.

Installation of Rational® Test Automation Server

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 V4.2+
- Ubuntu (using MicroK8s)

There is no direct upgrade path from previous versions to the latest version of Rational® Test Automation Server. You must uninstall any previously installed version, and then install the latest version of the server software.

As part of the install process, you can restore user data from a backup file. The backup might have come 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.

Installation of server software on Red Hat OpenShift

Installation of Rational® Test Automation Server software on Red Hat OpenShift requires the correct hardware, software, and user privileges.

Prerequisites for installing the server software on Red Hat OpenShift

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

The following section describes each prerequisite in detail:

- Red Hat OpenShift platform on page 27
- OpenShift Kubernetes Engine on page 28
- Other software on page 28
- Persistent storage on page 28
- Service virtualization through Istio on page 30

Red Hat OpenShift platform

You must set up a Red Hat OpenShift platform with the following additional resource requirements:

- 256 GB of disk space
- 16 GB of RAM (in addition to what OpenShift offers)
- 8 CPUs

For more information about specific system requirements, see System Requirements on page 11.

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 V4.2 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 the Jaeger operator.
  - 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 have access to the OpenShift cluster with required privileges. 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
```

Other software

You must have installed the following additional softwares on Red Hat OpenShift:

- Helm V3.1.3 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. The Kubernetes framework provides persistent storage for users and administrators. The persistent storage provides an API that abstracts

details about how storage is provided than consumed. You can manage resources of a cluster by using the following APIs:

- PersistentVolume (PV) - This is a resource in the cluster that is similar to a node in a cluster resource.
- PersistentVolumeClaim (PVC) - This is a request for storage by a user.

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	Platform supported	
				Kuber-netes	OpenShift
data-{my-rtas}-keycloak-post-gresql-0	2 GB	ReadWriteOnce **CLI Abbreviation:** RWO	Users and resource owner-ship	✓	✓
data-{my-rtas}-datasets-post-gresql-0	2 GB	RWO	Edits to Datasets	✓	✓
data-{my-rtas}-execution-post-gresql-0	2 GB	RWO	Infrastructure details	✓	✓
data-{my-rtas}-gateway-post-gresql-0	2 GB	RWO	Project details	✓	✓
data-{my-rtas}-kafka-0	8 GB	RWO	Notifications between ser-vices	✓	✓
data-{my-rtas}-kafka-zookeep-er-0	8 GB	RWO	Zookeeper state	✓	✓
data-{my-rtas}-results-0	8 GB	RWO	Reports	✓	✓
data-{my-rtas}-results-post-gresql-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	✓	✓

PersistentVolumeClaim	Size	Access Mode	Content	Platform supported	
				Kuber-netes	OpenShift
data-{my-rtas}-tam-postgresql-0	8 GB	RWO	Test asset metadata	✓	✓
data-{my-rtas}-userlibs-0	1 GB	RWO	User provided third party li-braries	✓	✗
data-{my-rtas}-userlibs-0	1 GB	ReadWriteMany **CLI Abbreviation:** RWX	User provided third party li-braries	✗	✓

Note: You must substitute `{my-rtas}` with the release name that you provided for the server.

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 30.

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 completed the following tasks:

- Completed the tasks provided in the Prerequisites section. See Prerequisites for installing the server software on Red Hat OpenShift on page 27.
- Created an API key for the IBM repository from https://cloud.ibm.com to pull the images used by the server software. For more information, refer to Creating an API key.
- **Optional**. Located any user data that was stored in a backup snapshot file from V10.0.2, Fix Pack 1 or earlier. See Backing up the user data from a previous release.
- You must have access to the internet to install Rational® Test Automation Server.

About this task

As part of the installation process, you might optionally migrate user data from a previous version of the product, or restore a backup of user data from the current version of the product. You must perform different steps, depending

on which version of the user data backup file you are applying during the install. For more details, see the following table:

Backed up user data from	To restore
Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier	Perform the step 6 on page 31 and 7 on page 32.
Rational® Test Automation Server V10.1.0	Perform the step 10 on page 34.

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

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

3. Add the repository to Helm to access the server install charts by entering 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 entering the following commands:

Notes:

- You must substitute `{api-key}` with your key for the cloud.ibm.com repository.
- You can replace the `not-required@test` with the administrators email address, if desired.

```
oc create secret docker-registry cp.icr.io \
  -n test-system \
  --docker-server=cp.icr.io \
  --docker-username=iamapikey \
  --docker-password={api-key} \
  --docker-email=not-required@test
```

6. **Optional:** Run the following commands to start a pod that assists you in migrating user data:

```
#Retrieve the backup yaml file
curl  -Lo ibm-rtas-prod.tgz
 https://raw.githubusercontent.com/IBM/charts/master/repo/ibm-helm/ibm-rtas-prod-1.101.0.tgz

#Extract the file
tar xvf ibm-rtas-prod.tgz

#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
```

7. **Optional:** Run the following command to display the import-prek8s-backup pod's log, and then follow the instructions in the log to copy your backup file to where the pod is waiting to process it:

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

   Result

   When the user data backup is fully migrated, the import-prek8s-backup pod changes from `Running` to `Completed` state. To view the pod state you can run the `oc get pod import-prek8s-backup -n test-system` command.

   Progress can be monitored by displaying the log to check for updates. When the state of the pod is `Completed`, the log file displays the additional parameters that must be added to the helm install in step 8 on page 41. The following is an example that shows the additional helm arguments:

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

   Notes:

   - You must substitute the following variables value with the actual value in the command:
     - `{my-rtas}` with the release name that you provided for the server.
     - `{openshift-cluster-dns-name}` with the remainder of the DNS name you selected for the server.
     - `{rlks-ip-address}` with the IP address of the Rational License Key Server.
     - `{my-super-secret}` with a password seed that you selected.
     - `{namespace}` with the name of the namespace that you created.

- The password seed is used to create all other default passwords. You must store the password seed securely. The password seed is reused when you install the server software by using a backup of user data. This seed is used when restoring backup files either on the current or later versions of the server software.
- If you migrated user data from a previous release, you must use the same `{my-rtas}` name that you chose then, during the install step.
- 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 execution.userlibs.persistence.storageClass=ibmc-file-bronze
  ```

```
#Make sure the repository is current, and retrieve the charts required to install the server
helm repo update
helm pull --untar ibm-helm/ibm-rtas-prod --version 1.101.0

#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=accept \
  -f ibm-rtas-prod/values-openshift.yaml \
  --set global.ibmRtasIngressDomain=rtas.apps.{openshift-cluster-dns-name} \
  --set global.rationalLicenseKeyServer=@{rlks-ip-address} \
  --set global.ibmRtasPasswordAutoGenSeed={my-super-secret} \
  --set global.ibmRtasRegistryPullSecret=cp.icr.io \
  --set keycloak.keycloak.image.pullSecrets[0]=cp.icr.io
```

Optionally, to enable virtual services to use the RedHat Service Mesh, complete the following two steps:

a. Add the following parameter to the helm install parameters in the preceding command before installing the server:

   ```
   --set execution.istio.enabled=true
   ```

b. Run the following command to create the role bindings required for Istio:

```
oc create rolebinding istio-virtualization-enabled -n
 {namespace} --clusterrole={my-rtas}-execution-istio-test-system
 --serviceaccount=test-system:{my-rtas}-execution
```

Note: When you uninstall the chart, the manually created role bindings are not deleted from the namespace.

9. **Optional:** Set the following parameters, if you have installed a Jaeger operator on the cluster and you want to use it for performance and Web UI tests logs:

```
--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}
```

Note: You must substitute `{my-jaeger-dashboard-url}` with the URL of the Jaeger server.

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

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

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

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

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

where:

- `{my-rtas}` with the release name that you provided for the server.
- `{namespace}` is a name of the namespace that you created during the server install.

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 must configure the license to use Rational® Test Automation Server. See Configuring floating licenses.
- 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 43.

Installation of server software on Ubuntu using microk8s

Installation of Rational® Test Automation Server software on Ubuntu requires the correct hardware, software, and user privileges.

Prerequisites for installing the server software on Ubuntu

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

The following section describes each prerequisite in detail:

- Ubuntu server on page 35
- Non-root user on page 35
- Kubernetes cluster on page 36
- Other software on page 36
- Service virtualization through Istio on page 36

Ubuntu server

You must set up an Ubuntu server V18.04.4 or later (V20.04 is not supported) with the following minimum requirements:

- 256 GB of disk space
- 32 GB of RAM
- 8 CPUs

You must have root access to the Ubuntu server.

Note:

Depending on your testing workload, Rational® Test Automation Server might require more resources. As a best practice, 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, you can create a mount point at `/var/snap/microk8s/common/default-storage/` with at least 128 GB capacity.

For more information about specific system requirements, see System Requirements on page 11.

Non-root user

You do not require to use a root account by default and when needed you can use the sudo command to execute a single command as the root user. If you do not want to be prompted for a password each time when you run the sudo command, you can enable the password less 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.

Kubernetes cluster

Kubernetes is a portable, extensible, and open-source platform to manage containerized workloads and services. Kubernetes facilitates both declarative configuration and automation. For more information, refer to the Kubernetes documentation.

- You must have set up the Kubernetes V1.14 environment.
- You must have access to the Kubernetes cluster with required privileges. You must contact your cluster administrator to gain access to the cluster. For more information, refer to the Accessing Clusters section in the Kubernetes documentation.

Other software

You must have installed the following additional software:

- Helm V3.1.3 or later. For more information, refer to the Installing Helm section in the Helm documentation.
- OpenSSH Server. For more information, refer to the Installation section in the OpenSSH Server documentation.
- **Optional**: You can install Jaeger operator to trace test logs and Jaeger-based reports when you run tests. For more information, refer to the Jaeger operator documentation.
- **Optional**: You can install 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.

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 using microk8s on page 39.

Setting up a Kubernetes environment on Ubuntu

You can use the readily available scripts to set up a microk8s Kubernetes environment along with other configurations to install Rational® Test Automation Server software in the Ubuntu server.

Before you begin

- You must have completed the tasks provided in the Prerequisites section. See Prerequisites for installing the server software on Ubuntu on page 35
- You must have a static IP address or Domain Name Server (DNS) 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-1.101.0.tgz
 https://raw.githubusercontent.com/IBM/cloud-pak/master/repo/case/ibm-rtas-case-1.101.0.tgz
```

3. Run the following command to extract the case:

```
tar xvf ibm-rtas-case-1.101.0.tgz
```

4. Run one of the following two sets of 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:

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

```
$ cd ibm-rtas-case/inventory/ibmRtasBase/files
$ sudo ./ubuntu-init.sh
```

- To install by overriding the default name, run the following commands:

```
$ 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 select for the server.
  - *myorg.com* is a domain name of your organization.

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 login page.
- The DNS information that the Kubernetes cluster uses to resolve names.
- Certificate Authority that you must be import into the browser to prevent certificate errors.
- Instructions to confirm whether the Kubernetes environment has started.
- Instructions to access Kiali to open product logs. For more information about Kiali dashboards, refer to Related information.

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*:
     ```
     $ sudo ./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 and you can now install the server software.

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 must install the server software. See Installing the server software on Ubuntu using microk8s on page 39.

Related information

Kiali Dashboards

Jaeger Operator

Prometheus server

Installing the server software on Ubuntu using microk8s

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

You must have completed the following tasks:

- Completed the tasks provided in the Prerequisites section. See Prerequisites for installing the server software on Ubuntu on page 35.
- Created an API key for the IBM repository from https://cloud.ibm.com to pull the images used by the server software. For more information, refer to Creating an API key.
- **Optional**. Located any user data that was stored in a backup snapshot file from V10.0.2, Fix Pack 1 or earlier. See Backing up the user data from a previous release.
- You must have access to the internet to install Rational® Test Automation Server.
- You must ensure that `/snap/bin` is in your `PATH`. You can run the `export PATH="/snap/bin:$PATH"` command to set your `PATH`.

About this task

As part of the installation process, you might optionally migrate user data from a previous version of the product, or restore a backup of user data from the current version of the product. You must perform different steps, depending on which version of the user data backup file you are applying during the install. For more details, see the following table:

Backed up user data from	To restore
Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier	Perform the step 6 on page 40 and 7 on page 40.
Rational® Test Automation Server V10.1.0	Perform the step 10 on page 42.

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

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

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

Notes:

- You must substitute `{api-key}` with your key for the cloud.ibm.com repository.
- You can replace the `not-required@test` with the administrators email address, if desired.

```
kubectl create secret docker-registry cp.icr.io \
  -n test-system \
  --docker-server=cp.icr.io \
  --docker-username=iamapikey \
  --docker-password={api-key} \
  --docker-email=not-required@test
```

6. **Optional:** Run the following commands to start a pod that assists you in migrating user data:

```
#Retrieve the backup yaml file
curl  -Lo ibm-rtas-prod.tgz
 https://raw.githubusercontent.com/IBM/charts/master/repo/ibm-helm/ibm-rtas-prod-1.101.0.tgz

#Extract the file
tar xvf ibm-rtas-prod.tgz

#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
```

7. **Optional:** Run the following command to display the import-prek8s-backup pod's log, and then follow the instructions in the log to copy your backup file to where the pod is waiting to process it:

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

Result

When the user data backup is fully migrated, the import-prek8s-backup pod changes from `Running` to `Completed` state. To view the pod state you can run the `kubectl get pod import-prek8s-backup -n test-system` command.

Progress can be monitored by displaying the log to check for updates. When the state of the pod is `Completed`, the log file displays the additional parameters that must be added to the helm install in step 8 on page 41. The following is an example that shows the additional helm arguments:

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

Notes:

- You must substitute the following variables value with the actual value in the command:
  - `{my-rtas}` with the release name that you provided for the server.
  - `{my-ingress-dns-name}` with the same Ingress DNS name that you used in ubuntu-init.sh.
  - `{rlks-ip-address}` with the IP address of the Rational License Key Server.
  - `{my-super-secret}` with a password seed that you selected.
  - `{namespace}` with the name of the namespace that you created.
- The password seed is used to create all other default passwords. You must store the password seed securely. The password seed is reused when you install the server software by using a backup of user data. This seed is used when restoring backup files either on the current or later versions of the server software.
- If you migrated user data from a previous release, you must use the same `{my-rtas}` name that you chose then, during the install step.

```
#Make sure the repository is current
helm repo update
helm pull --untar ibm-helm/ibm-rtas-prod --version 1.101.0
```

```
#Install the server
helm install {my-rtas} ibm-helm/ibm-rtas-prod -n test-system \
 --set license=accept \
 --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 \
 --set keycloak.keycloak.image.pullSecrets[0]=cp.icr.io
```

Optionally, to enable virtual services to use the Istio Service Mesh, complete the following two steps:

a. Add the following parameters to the helm install parameters in the preceding command before installing the server:

```
--set execution.istio.enabled=true \
--set execution.istio.clusterRoleBinding.create=true
```

b. Run the following command to create the role bindings required for Istio:

```
kubectl create rolebinding istio-virtualization-enabled -n
 {namespace} --clusterrole={my-rtas}-execution-istio-test-system
 --serviceaccount=test-system:{my-rtas}-execution
```

Note: When you uninstall the chart, the manually created role bindings are not deleted from the namespace.

9. **Optional:** Set the following parameters, if you have installed a Jaeger operator on the cluster and you want to use it for performance and Web UI tests logs:

```
--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}
```

Note: You must substitute `{my-jaeger-dashboard-url}` with the URL of the Jaeger server.

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

a. Stop Kubernetes by running the `microk8s.stop` 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-name>
```

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

For more information about using additional parameter and overwriting the existing Persistent Volumes, see Backing up and restoring the user data on Ubuntu topic.

c. Restart Kubernetes by running the `microk8s.start` command to start Rational® Test Automation Server.

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

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

where:

- `{my-rtas}` with the release name that you provided for the server.
- `{namespace}` is a name of the namespace that you created during the server install.

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 Ubuntu on page 45.

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. For more information about the Persistent Volumes, see Persistent Volumes on page 29.

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. For more details, refer to the Velero portal.

3. Set the name of the namespace in which Rational® Test Automation Server is installed by entering 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 entering 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 entering 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 follow the steps in the Preparing your cluster to backup and restore the user data on page 43 section.

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 entering 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 by using Velero.

Before you begin

You must have backed up the user data. For more information, see Backing up the user data on page 44 section.

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

Backing up and restoring the user data on Ubuntu

To secure the user data in IBM® 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 IBM® 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 the system 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 these 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. Open a terminal and change to the directory where you extracted `ibm-rtas-case-1.101.0.tgz`.
2. Change to the `ibm-rtas-case/inventory/ibmRtasBase/file` directory. This directory contains the backup.sh and restore.sh scripts which are required to complete the backup and restore operations.
3. Stop Kubernetes by entering the following command:

   microk8s.stop

4. Create a backup of the existing user data by entering the following command:

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

   After you run this command, a backup of the local Persistent Volumes is created. The backups are created as tar archives that are compressed by using gzip (.tar.gz).

   The **create** command archives the Persistent Volumes into the <backup-file>.

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 <namespace>

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

  The syntax is -volumes <directory>

  The default path of the directory is `/var/snap/microk8s/common/default-storage`

For example, ./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.

Result

The backup is created.

5. Restart Kubernetes to start Rational® Test Automation Server by entering the following command:

   microk8s.start

Restoring the user data

You can restore the user data that is backed up in Rational® Test Automation Server V10.1.0 at any point in time after you install Rational® Test Automation Server V10.1.0. The user data from Rational® Test Automation Server V10.0.2, Fix Pack 1 or earlier, can be migrated to Rational® Test Automation Server V10.1.0 during the installation of Rational® Test Automation Server V10.1.0. You can refer to the installation instructions to migrate the user data to Rational® Test Automation Server V10.1.0.

1. Open a terminal and change to the directory where you extracted `ibm-rtas-case-1.101.0.tgz`.
2. Change to the `ibm-rtas-case/inventory/ibmRtasBase/file` directory. This directory contains the backup.sh and restore.sh scripts which are required to complete the backup and restore operations.
3. Stop Kubernetes by entering the following command:

   microk8s.stop

4. Restore the backed-up user data by entering the following command:

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

   The restore command overwrites the existing Persistent Volumes with data from the <backup_file>. This command can restore the compressed or uncompressed tar files.

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

- --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 <namespace>[:<target-namespace>]

- --release or -r: Use this parameter if the release to which the backup is being restored is different to that in the backup.

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

- --volumes or or -v: Use this parameter to back up the volumes that are associated with the namespace that you want to back up.

  The syntax is --volumes <directory>

  The directory path contains the volumes, which are expressed as an absolute path. The default path is `/var/snap/microk8s/common/default-storage`

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

For example, ./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-1* release in the *new-test-system* namespace and skips the confirmation step.

5. Restart Kubernetes to start Rational® Test Automation Server by entering the following command:

   microk8s.start

   Note: Although the pods are restored, the server might take some time to start.

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 release name that you provided for the server.

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 release name that you provided for the server.

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

```
cd ibm-rtas-case/inventory/ibmRtasBase/files
sudo ./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 49
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 bowser
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 65

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 51.

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 rockstar-keycloak-postgresql -o jsonpath="{.data.password}" |
   base64 --decode; echo
  ```

  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.

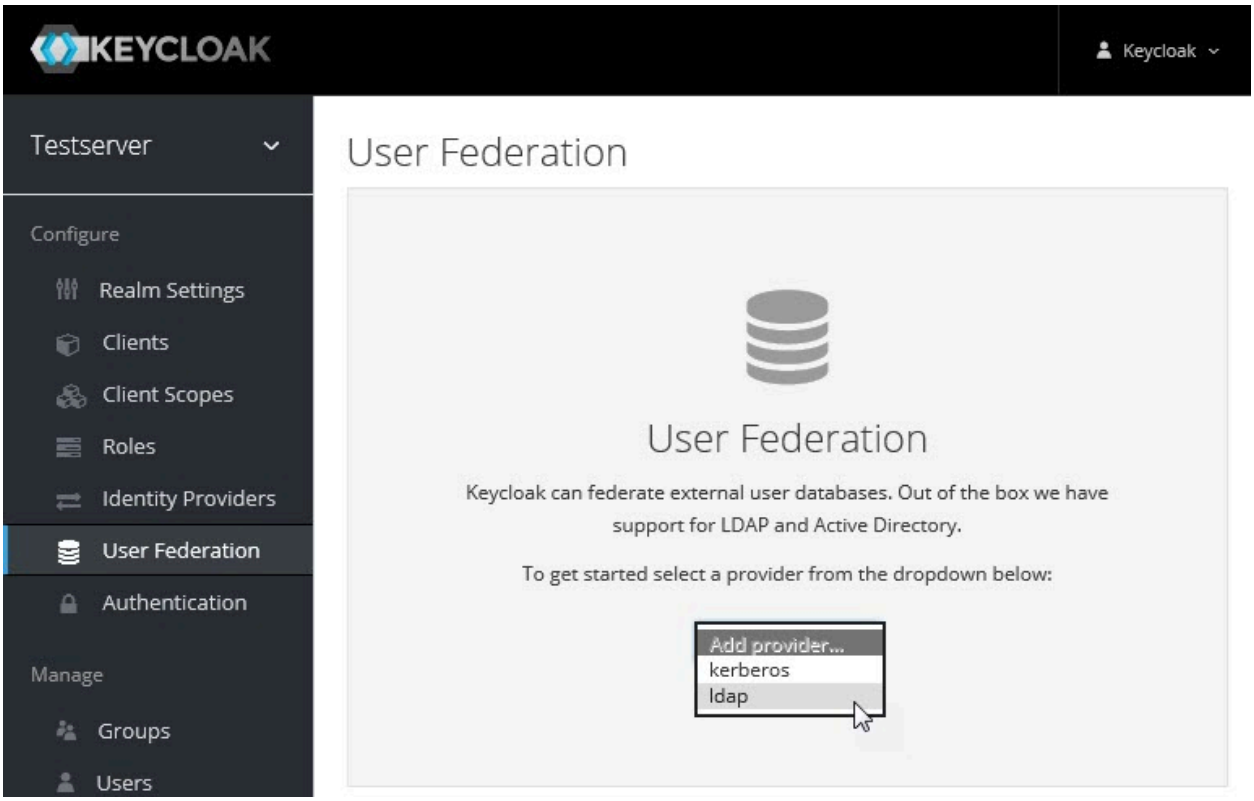
LDAP provider selection in Keycloak

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 rockstar-keycloak-postgresql -o jsonpath="{.data.password}" | base64
 --decode; echo
```

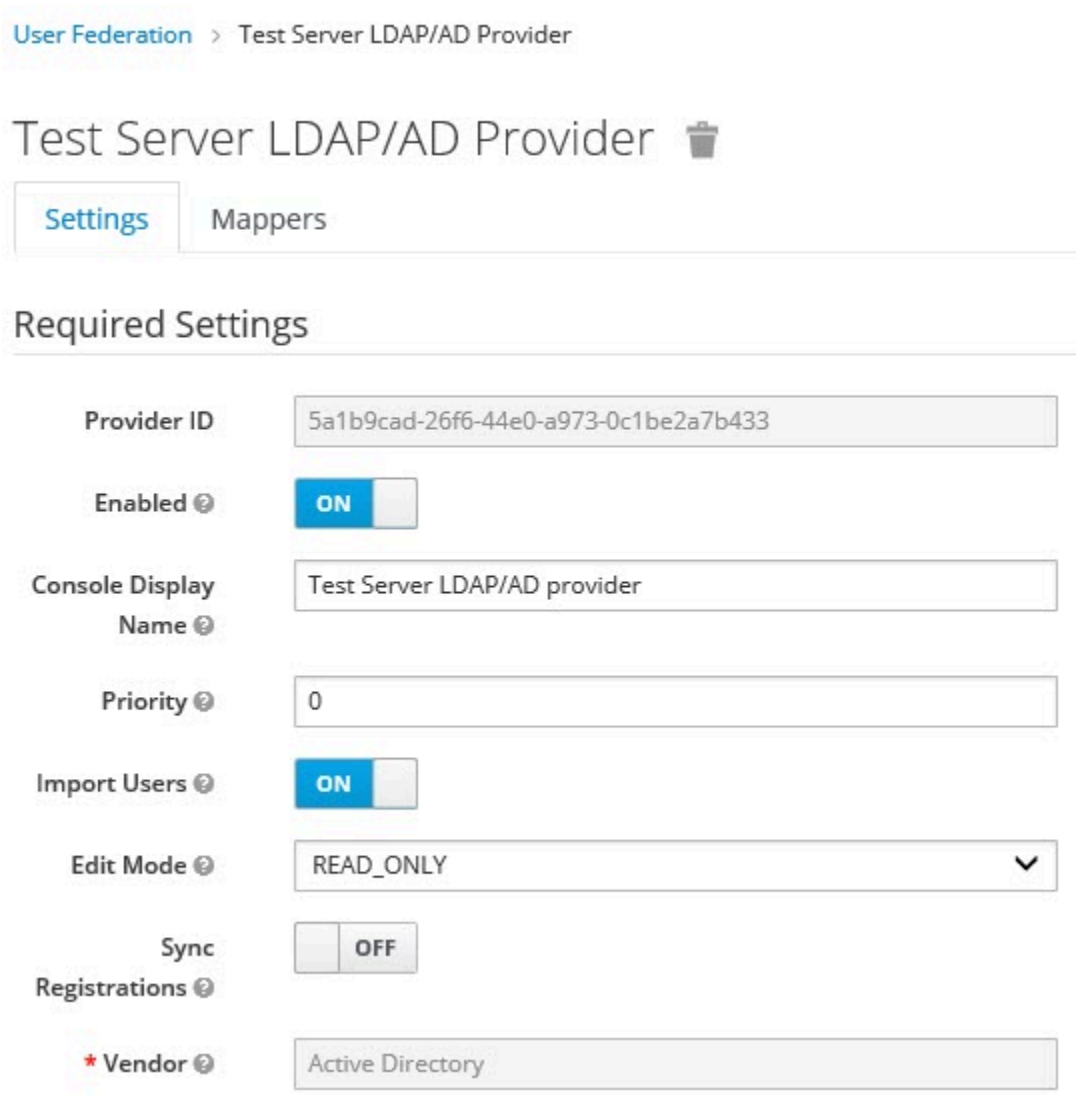
After you log in, you can go to the User Federation page to add your provider.



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.

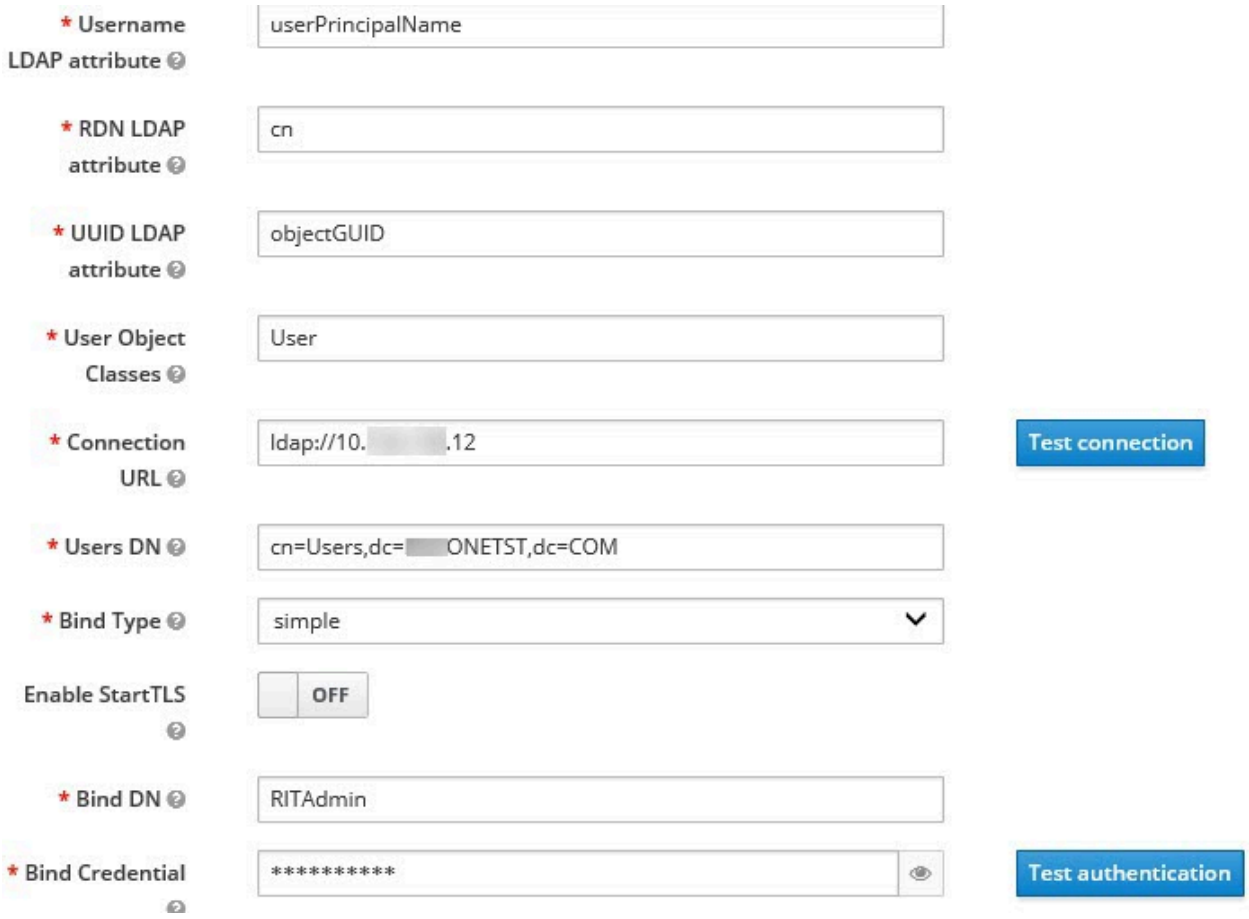
Required settings for a successful connection to your LDAP/AD provider

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.



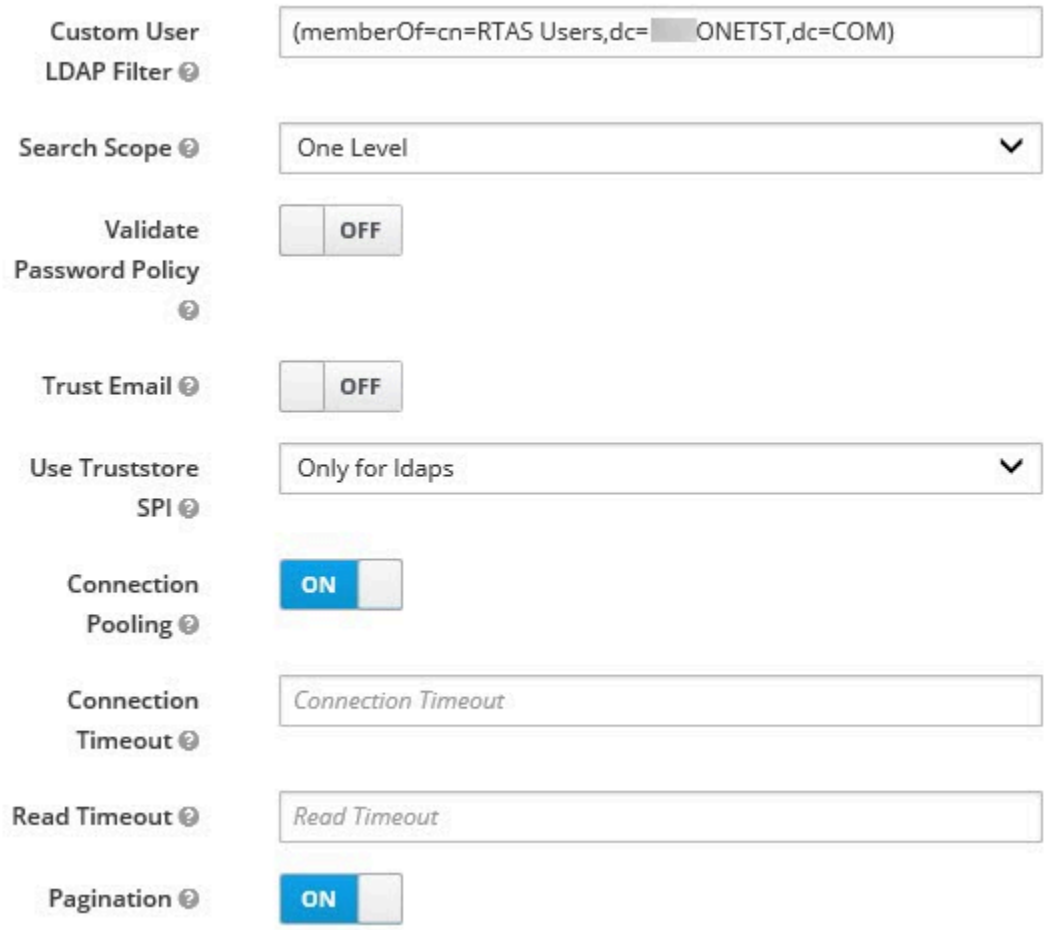
Field	Description
Console Display Name	You want to make your console name recognizable, for example, Test Server LDAP/AD Provider. This name is shown in the Keycloak Admin Console.
Priority	The priority must be set to 0 so that when a user logs in to Rational® Test Automation Server, Keycloak looks up 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. 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. 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. **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**. 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. 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.

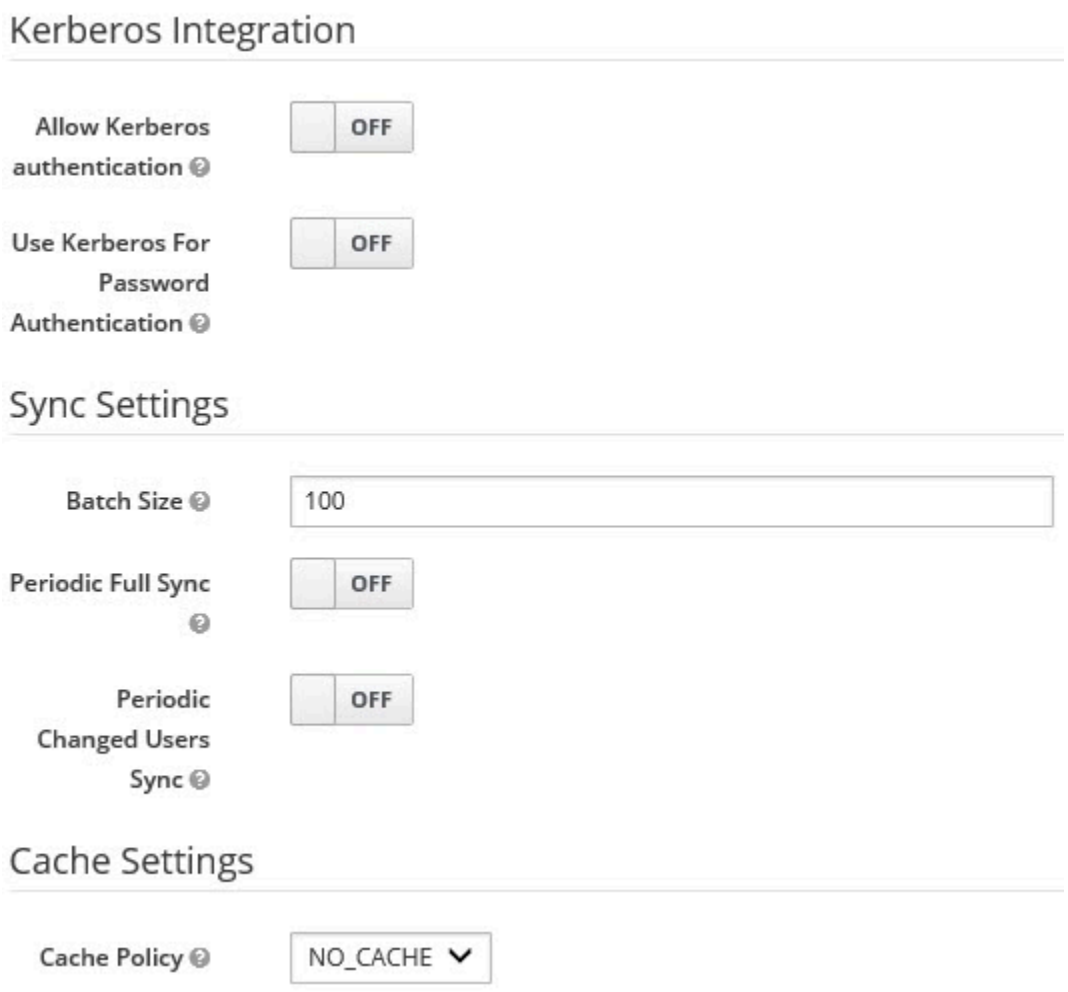


Field	Description
Username LDAP attribute	You can use the default value `cn` for the `username`, which is first name, last name, or you can use `userPrincipalName`, which is username@domain. You might want a login that matches your company style. For example, you might prefer joetester@mycompany.com instead of Joe Tester. 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>` 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 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. 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. 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. `(memberOf=cn=TESTERS,dc=MYCOMPANY,dc=COM)`

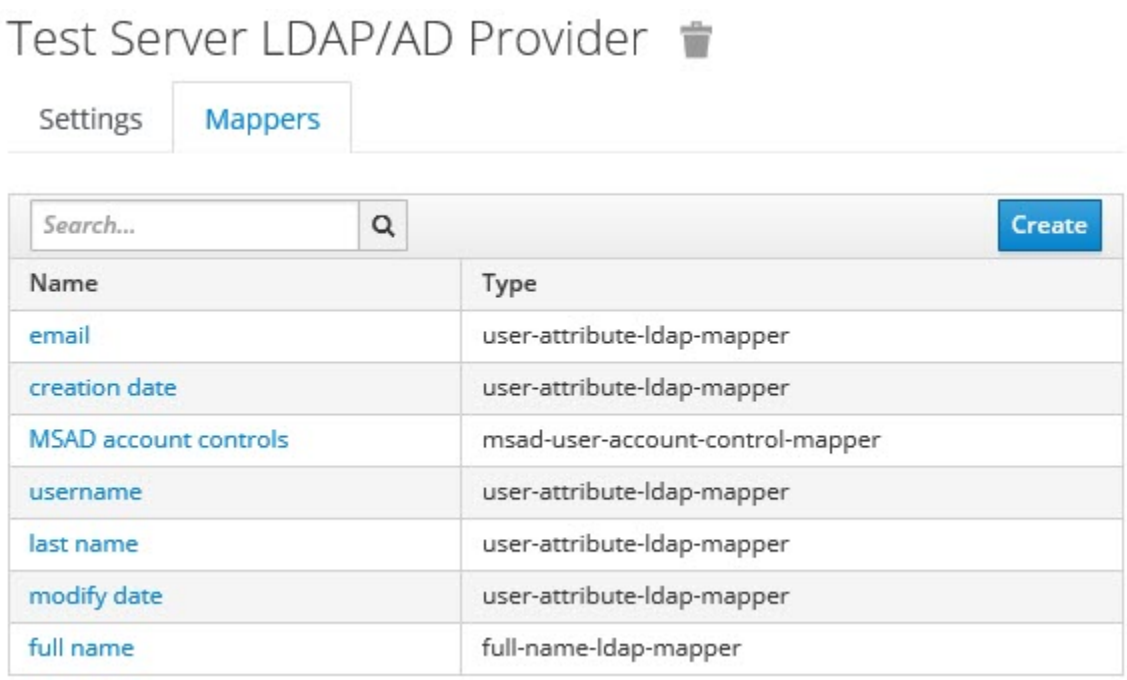


Field	Description
Batch size	You can change the number of users to import in a single transaction by using the batch size.
Cache Policy	If you have many users or performance concerns, you can change the cache policy to no cache.

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

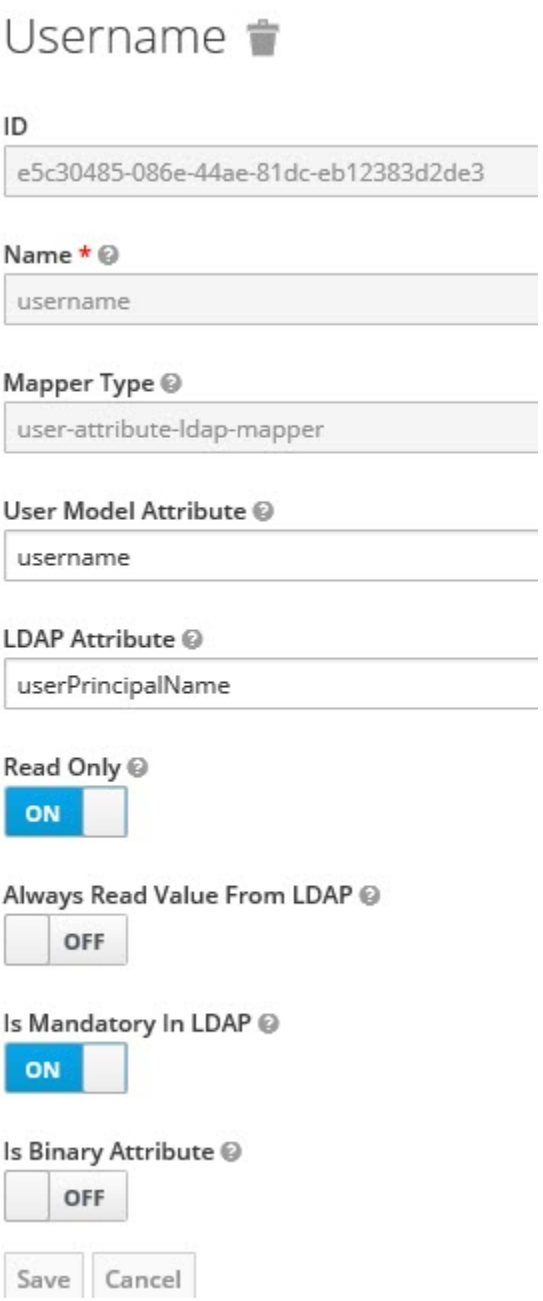
Mappers

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.



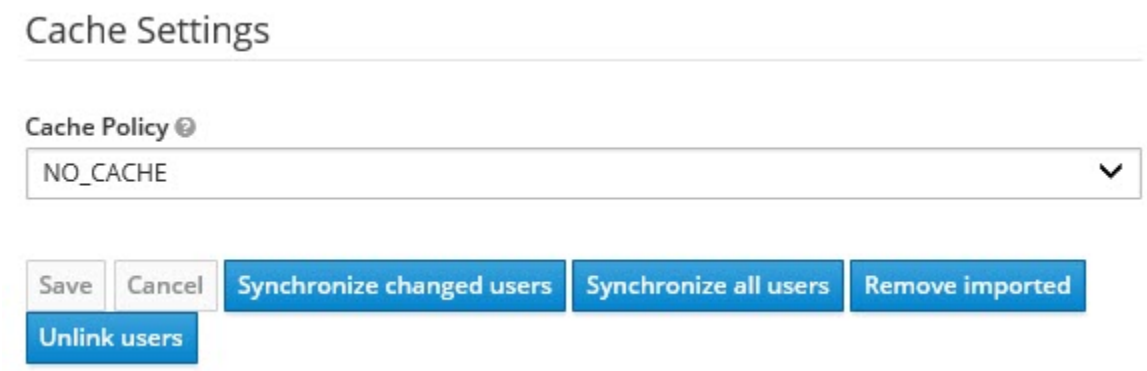
The username attribute that you specified in the **Username LDAP attribute** must match the **username** t attribute defined in the Keycloak mapper for the LDAP/AD user database to connect with Keycloak.

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.



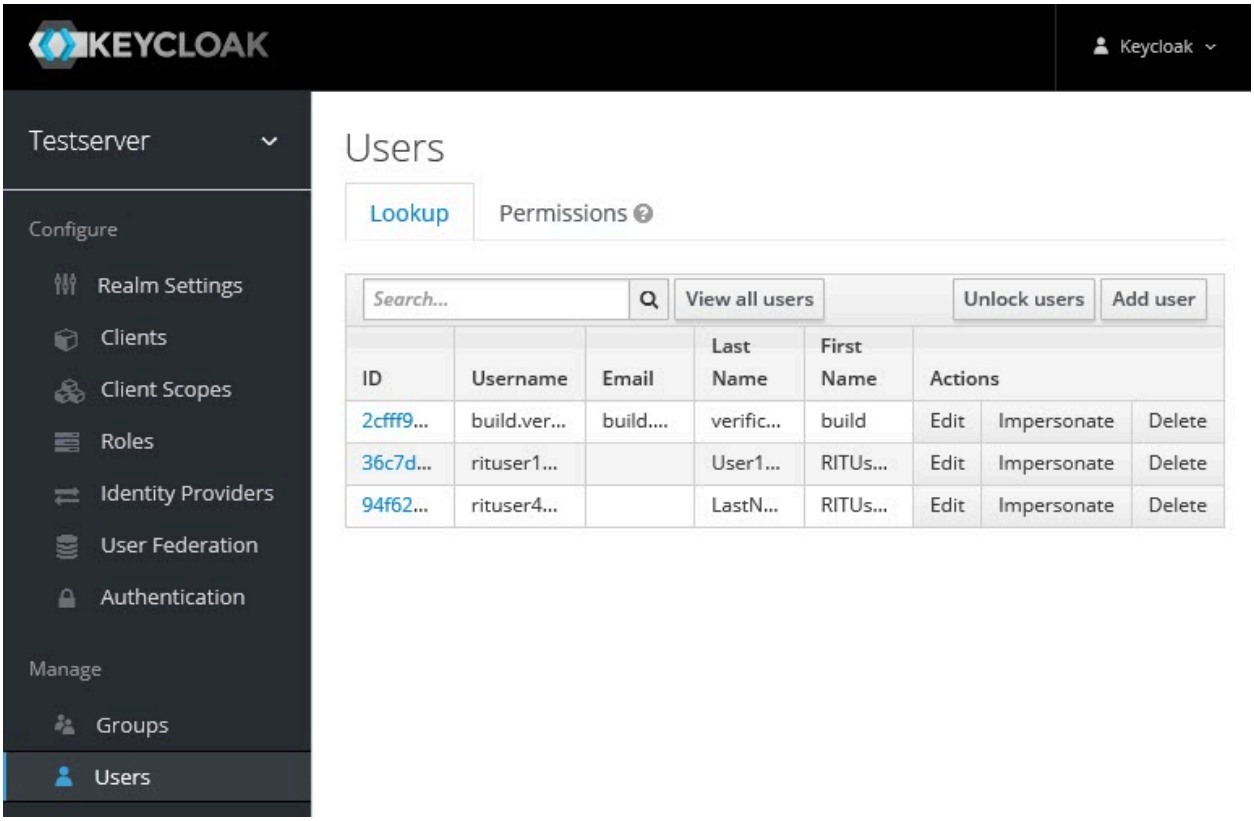
User synchronization

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 50

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 63
- Importing Certificate Authority into the Microsoft Edge browser on page 63
- Importing Certificate Authority into the Mozilla Firefox browser on page 64

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 88.

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 81.
- 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 2. 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 that you provided for the server.

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 that you provided for the server.

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 a run for an API Suite on page 104.

Licensing

You must configure the number of purchased floating licenses in Rational® Test Automation Server for the users to create a project and perform other tasks in the project.

Floating licenses

You can purchase floating licenses for Rational® Test Automation Server and install the licenses on the Rational License Key Server. You must be an administrator to configure the purchased floating licenses in Rational® Test Automation Server.

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

- 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 **Settings > Configure Licenses**.

After the administrator configures the licenses, you can start creating projects in Rational® Test Automation Server. You can view the following details in the **Available licenses** dialog box that is displayed when you click **Settings > Configure Licenses**.

Field	Description
License usage	The number of licenses that are currently in use.
Maximum number of floating licenses	The total number of floating licenses that are configured by the administrator.

A single floating license is used from the total number of floating licenses that are configured by the administrator when you perform 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.
- End your session on Rational® Test Automation Server or your session is timed out.
- Completed the test runs on any of the third-party applications that are integrated with Rational® Test Automation Server.

Chapter 5. Test Author guide

This guide, which is intended for a tester and test manager, describes how to create test assets in Rational® Test Automation Server and publish test assets to the Git repository that you have configured.

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.

When you record a test, you can perform a sequence of steps. After the recording is complete, a test is generated that captures these interactions. When you run this test, it uses the same data that you used during the recording. To use the different data in the test, you can use a dataset that contains variable data. At run time, this variable data is substituted with the original data in the recorded test.

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 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**.

When multiple repositories are added to the same project, the following events occur:

- The datasets stored in the `master` branch of all the repositories are displayed.
- All the branches in all the repositories are listed in the drop-down list.

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 207.
- Configured the Git repository in your project. See Adding repositories to a server project on page 207.

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. From the Rational® Test Automation Server UI, open your project.
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 Dataset editing on page 70.
- 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 75.

Dataset editing

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

If you are a project **Owner** or **Tester** in Rational® Test Automation Server V10.1.0 or later, you can use the CSV Editor to view and edit data in the dataset. From the **Datasets** page, you can click the **Edit** icon from the **Actions** column of the dataset to open the dataset in the CSV Editor in a web browser.

After you edit the dataset, you can publish the dataset to the Git repository or you can save the changes made to the dataset. If you save and close the edited dataset, the **Changes** page lists the edited dataset, and then later you can publish to the Git repository for other members to use. For any reason, if you want to discard the changes that you made to the dataset, you can click the **Menu** icon and then select **Discard**.

Basic tasks in CSV Editor

When you want to run a test asset with different dataset values, you can either edit the existing dataset or create a new dataset and use it during the test asset run. When the number of edits is minimal, it is easier to edit the dataset within the CSV Editor.

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.

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.

Set current row in the dataset

During the test run, if you want variable data to be selected from a current row instead of the first row in a dataset, you can right-click any cell in a row and select **Set as current row**. Also, you can set the current row from the **Datasets** page by clicking **Menu**, and then the **Configure** option.

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.

Similarly, 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.

Configure row and column settings

In the **Configure Dataset** window, you can change the row and column settings and configure the string values in a dataset that contains variable data for tests to use when they run. You can click **Menu**, and then **Configure** option to configure row and column settings

You can change the **Column header** and **Data start point** values by using the up-down control buttons. You can enter a string value in the **Treat as null** and **Treat as empty** fields to treat the entered value as null or empty during a test run.

If a dataset contains any blank cells, the value of those blank cells is interpreted as null when you select the **Treat empty text as null** field checkbox.

Content search by using Find and Replace option

When you have a large set of data in a dataset, you might want to find a specific word and replace the word with a new word in the dataset. You can do this task by using the **Find and Replace** option. You can search the content and replace with new content more effectively by selecting any of the following options:

- **Case sensitive** - To search the content that matches the letter case of the content entered in the **Find** field.
- **Match entire cell contents** - To search for cells that contain only the characters that you entered in the **Find** field.
- **Search using regular expression** – To search the pattern that matches strings.

  For example, to search a cell that contains any number between 0 to 9, you can enter \d in the **Find** field, select the **Search using regular expression** checkbox, and then click **Find**.

You can replace the individual instances by clicking **Replace** and replace every instance of the content throughout the dataset by clicking **Replace All**.

Undo and redo actions

While editing a dataset, if you want to revert the changes made to the dataset, you can use the **Undo** option.

Similarly, you can redo the action in the dataset by using the **Redo** option. You can use the undo and redo options even after you saved the dataset for all the actions performed in 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 69.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. From the Rational® Test Automation Server UI, open your project.
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 75.

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 69 and Encrypting a dataset column on page 73.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. From the Rational® Test Automation Server UI, open your project.
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 with 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 75.

Publishing a dataset

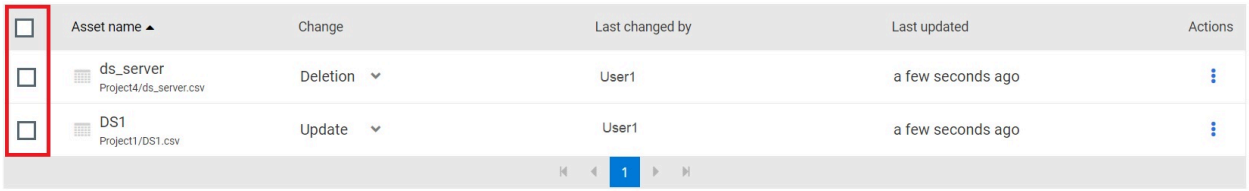
When you create or edit any datasets in Rational® Test Automation Server, you can publish your changes to the Git repository. Therefore, when you publish a dataset, other members in the project can use your dataset in their test assets run, if required.

Before you begin

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

About this task

The **Changes** page lists all updates made to the dataset assets. The status of the dataset can be creation, modifications, or deletion of datasets. You can publish all updated 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:

- **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.

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

3. From the Rational® Test Automation Server UI, open your project.
4. Go to the **Changes** page, find the dataset that you want to publish.
5. Select the checkbox preceded with 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 207.
- Configured the Git repository in your project. See Adding repositories to a server project on page 207.
- Created at least one dataset and encrypted the dataset with an encryption key. See Creating a dataset on page 69 and Encrypting a dataset column on page 73.
- Created a classification for an encrypted dataset. See Creating a classification on page 225.

1. Go to the Rational® Test Automation Server URL.
2. Enter your user name and password, and then click **Login**.
3. From the Rational® Test Automation Server UI, open your project.
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 225.

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 225

Creating a classification on page 225

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. From the Rational® Test Automation Server UI, open your project.
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, which is intended for a tester or a test execution specialist, describes tasks that you must complete before you can configure and run tests on 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.

In Rational® Test Automation Server a tester or a test execution specialist has a *Tester* role assigned in the project.

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. 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. 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. • 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.
- 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 a run for an AFT Suite on page 98.

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 82
- Important information about report configuration in API Suites on page 83

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.
- 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.	You must remove the local stubs referenced in the tests or published the stubs to Rational® Test Control Panel before you commit the test asset to the remote repository. To publish and edit stubs, see Publishing stubs.
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. The API Suite contains tests that use a transport from any of the following third-party applications: • Camel • CentraSite • CICS • Coherence • Database • IMS • Integra • JMS • SAP RFC • Software AG Universal Messaging • TIBCO EMS • TIBCO Rendezvous	You must complete the following tasks: • 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. • Copy the third-party application JAR files to the computer where Rational® Test Automation Server is installed. See Copying third-party application Jars to Kubernetes on page 65.

If...	Then...
• TIBCO SmartSockets • WebSphere Application Server Service Integration Bus (SiBus) • WebLogic • Software AG webMethods • WebSphere MQ	
You plan to run the API Suite on a remote Docker host. The API Suite contains tests that use a transport from any of the following third-party applications: • Camel • CentraSite • CICS • Coherence • Database • IMS • Integra • JMS • SAP RFC • Software AG Universal Messaging • TIBCO EMS • TIBCO Rendezvous • TIBCO SmartSockets • WebSphere Application Server Service Integration Bus (SiBus) • WebLogic • Software AG webMethods • WebSphere MQ	You must complete the following tasks: • 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. • Copy the third-party application JAR files to the computer where the remote Docker host is installed. See Copying third-party application Jars to a remote Docker host on page 88.

Important information about report configuration in API Suites

Rational® Integration Tester Test Suites must be configured 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. If you have such API Suites, you must have configured a results database in the project used in Rational® Integration Tester. 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 65.

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

For running an API Suite, see Configuring a run for an API Suite on page 104.

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 you must take into account.

Important information about JMeter tests in schedules

When you plan 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.

For running a Rate Schedule or VU Schedule, see Configuring a run for Rate Schedules or VU Schedules on page 128.

Related information

Tests configurations and test runs

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 internal Docker host of IBM® Rational® Test Automation Server.

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

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 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:

- Set up a remote Docker host system. See Setting up a remote Docker host system on page 85 or Setting up a secure remote Docker host system on page 87.
- Configure the remote Docker host. See Configuring the remote Docker host system on Rational Test Automation Server.
- Register the remote Docker host with IBM® Rational® Test Automation Server. See Registering a remote Docker host with IBM Rational Test Automation Server on page 90.

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 88.

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 92.

Related information

Tests configurations and test runs

Setting up a remote Docker host system

You must set up a system 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.

1. Locate a remote computer or Virtual Machine (VM) that acts as the remote Docker host system. The remote systems must be running on Ubuntu V18.04 or later, and has a minimum of 16 GB RAM.
2. Install Docker Enterprise Edition for a non-secure mode of connection with IBM® Rational® Test Automation Server.

For more information, refer to the related links.

3. 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
   ```

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

   $ sudo systemctl daemon-reload

   $ sudo systemctl restart docker

5. Run the following command on the remote Docker host system to get access to the Docker repository:

   $ sudo docker login <host_name> -u <username> -p <password>

   Where, *<host_name>* is the host name of the remote Docker host system, *<username>* and *<password>* are your credentials to log in.

6. Pull the following images from Rational® Test Automation Server to the remote host system:

   Docker images of: **ibm-rtw**, **ibm-rtvs**

   You can use the following command to pull the image:

   $ sudo docker pull <host_name>/cp/ibm-rtw:10.1.0

   $ sudo docker pull <host_name>/cp/ibm-rtvs:10.1.0

Results

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

What to do next

You can set up a secure remote Docker host system optionally, that enables a secure connection with Rational® Test Automation Server.

You must configure the remote Docker host on Rational® Test Automation Server.

Related information

Get Docker Engine

Setting up a secure remote Docker host system on page 87

Configuring the remote Docker host system on Rational Test Automation Server

Registering a remote Docker host with IBM Rational Test Automation Server on page 90

Adding a remote Docker host to the project for running tests on page 92

Editing configurations of a remote Docker host on page 93

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 95

Setting up a secure remote Docker host system

You must set up a system 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 system that authenticate the connection.

1. Locate a remote computer or Virtual Machine (VM) that acts as the remote Docker host system. The remote systems must be running on Ubuntu V18.04 or later, and has a minimum of 16 GB RAM.
2. IBM® Rational® Test Automation ServerInstall Docker Community Edition V18.09 or later, for a secured mode of connection with .

   For instructions, see Get Docker Engine.

   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. See the *Related Information* link.
3. Run the following command on the remote Docker host system to get access to the Docker repository:

   $ sudo docker login <host_name> -u <username> -p <password>

   Where, *<host_name>* is the host name of the remote Docker host system, *<username>* and *<password>* are your credentials to log in.
4. Pull the following images from Rational® Test Automation Server to the remote host system:

   Docker images of: **ibm-rtw**, **ibm-rtvs**

   You can use the following command to pull the image:

   $ sudo docker pull <host_name>/cp/ibm-rtw:10.1.0

   $ sudo docker pull <host_name>/cp/ibm-rtvs:10.1.0

Results

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

What to do next

You must configure the remote Docker host on Rational® Test Automation Server.

Related information

Configuring the remote Docker host system on Rational Test Automation Server

Setting up a remote Docker host system on page 85

Registering a remote Docker host with IBM Rational Test Automation Server on page 90

Adding a remote Docker host to the project for running tests on page 92

Editing configurations of a remote Docker host on page 93

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 95

https://docs.docker.com/engine/security/https/

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 65.

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 84.
- Identified the third-party application `Jar` files that are required and copied the `Jar` files. See Test run considerations for API Suites on page 81.
- 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 3. 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 90.
- You can configure the API Suite run. See Configuring a run for an API Suite on page 104.

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 system either for a secured mode or unsecured mode of connection. See Setting up a remote Docker host system on page 85 or Setting up a secure remote Docker host system on page 87.

You must have configured the remote Docker host system on Rational® Test Automation Server. See Configuring the remote Docker host system on Rational Test Automation Server.

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.

   Note: You must enter the same details for the remote Docker host that you have set in the `docker-compose.yml` file.

   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 system:

   - 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 system 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 system.

   - On a successful connection, a message is displayed.
   - On a failure to connect to the remote host system, an error message is displayed. You must resolve the error and reattempt to establish a successful connection.

Important: The remote Docker host system 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 information

Setting up a remote Docker host system on page 85

Adding a remote Docker host to the project for running tests on page 92

Editing configurations of a remote Docker host on page 93

Deleting a remote Docker host from a project on page 95

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 95

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 systems. 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 85.
- Set up the remote Docker host system. See Setting up a remote Docker host system on page 85 or Setting up a secure remote Docker host system on page 87.
- Configured the remote Docker host. See Configuring the remote Docker host system on Rational Test Automation Server.
- 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 90.

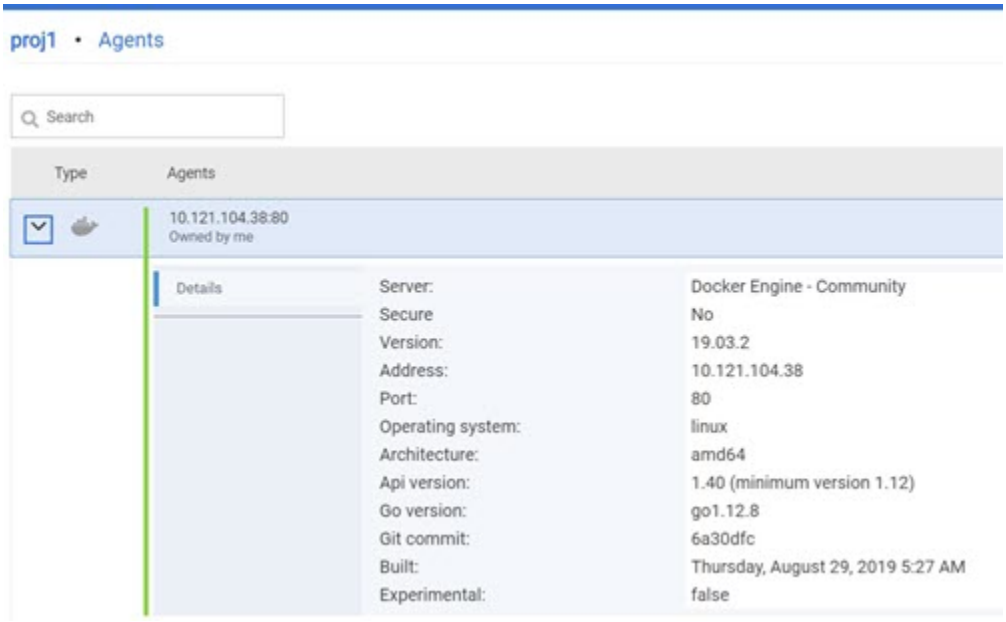
1. Log in to IBM® Rational® Test Automation Server and click **Infrastructure**.
2. Click **Add > Add docker**.

   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. See Test run configurations on page 96.

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 system instead of registering it as a new remote host if any of the parameters are changed on the remote host system. 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 system. See Setting up a remote Docker host system on page 85 or Setting up a secure remote Docker host system on page 87.
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 90.

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 system 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 system:

   - On a successful connection, a message is displayed.
   - On a failure to connect to the remote host system, 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 information

Setting up a remote Docker host system on page 85

Registering a remote Docker host with IBM Rational Test Automation Server on page 90

Adding a remote Docker host to the project for running tests on page 92

Deleting a remote Docker host from a project on page 95

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 95

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 system. See Setting up a remote Docker host system on page 85 or Setting up a secure remote Docker host system on page 87.
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 90.
3. Added the registered Docker host to a project. See Adding a remote Docker host to the project for running tests on page 92.

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 information

Unregistering a remote Docker host from IBM Rational Test Automation Server on page 95

Editing configurations of a remote Docker host on page 93

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 system. See Setting up a remote Docker host system on page 85 or Setting up a secure remote Docker host system on page 87.
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 90.

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 information

Editing configurations of a remote Docker host on page 93

Adding a remote Docker host to the project for running tests on page 92

Deleting a remote Docker host from a project on page 95

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 205.
- 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 80.

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 80 if they apply to the test you are configuring for a run.

1. Open the project that contains the test assets you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets 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 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 text 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 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Viewing the progress of running test assets on page 134

Monitoring a test run

Stopping a test run on page 139

Canceling a scheduled test run on page 140

Configuring a run for an AFT Suite

After you added the AFT Suite asset 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 80 if you are configuring 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 you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset that you want to run 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.
   - Create a filter query by using the **New filter** option. Enter the relevant parameters and apply the filter query. You can save the filter query for retrieving it from the list of saved filters.
   - Retrieve a saved filter by using the **Open filters** icon. Select and apply the saved 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 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.

- 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 earlier committed versions.

- 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 have 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 and the default version is selected for the test run.
- If you are repeating a test or you are rerunning the test 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 earlier 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 text that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, it can be used by any member of the project.

   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
   ```

   Note: You must separate the arguments or variables with a white space when you enter them in the same line or you must 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 in this step 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.

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 in this step 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 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 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 there were no agents configured in the test asset. If the test asset had agents configured, the default agent is the first one 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 in this step 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 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 have 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 either started or scheduled a test run for 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 139.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 140.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Tests configurations and test runs

Test run configurations on page 96

Configuring a run for an API Suite

After you added the API Suite asset 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 81.
  - 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 65.
  - 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 88.

1. Open the project that contains the test assets you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset that you want to run 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.
   - Create a filter query by using the **New filter** option. Enter the relevant parameters and apply the filter query. You can save the filter query for retrieving it from the list of saved filters.
   - Retrieve a saved filter by using the **Open filters** icon . Select and apply the saved 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 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.

- 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 earlier committed versions.

- 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 have 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 and the default version is selected for the test run.
- If you are repeating a test or you are rerunning the test 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 earlier 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 text that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, it can be used by any member of the project.

   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 runtime 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

   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 third-party libraries 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 you must 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 in this step 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 in this step 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 in this step 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 in this step 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 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 have 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 either started or scheduled a test run for 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 139.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 140.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Tests configurations and test runs

Test run configurations on page 96

Configuring a run for Compound Tests that contains traditional HTML tests

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 you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset that you want to run 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.
   - Create a filter query by using the **New filter** option. Enter the relevant parameters and apply the filter query. You can save the filter query for retrieving it from the list of saved filters.
   - Retrieve a saved filter by using the **Open filters** icon. Select and apply the saved 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 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.

   - 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 earlier committed versions.

- 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 have 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 and the default version is selected for the test run.
- If you are repeating a test or you are rerunning the test 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 earlier 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 text that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, it can be used by any member of the project.

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 runtime 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

    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 third-party libraries 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 you must 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 in this step 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 in this step 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 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 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 there were no agents configured in the test asset. If the test asset had agents configured, the default agent is the first one 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 in this step 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 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 have 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 either started or scheduled a test run for 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 139.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 140.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Tests configurations and test runs

Test run configurations on page 96

Configuring a run for Compound Tests that contains Web UI tests

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.

1. Open the project that contains the test assets you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset that you want to run 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.
   - Create a filter query by using the **New filter** option. Enter the relevant parameters and apply the filter query. You can save the filter query for retrieving it from the list of saved filters.
   - Retrieve a saved filter by using the **Open filters** icon . Select and apply the saved 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 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.

   - 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 earlier committed versions.

- 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 have 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 and the default version is selected for the test run.
- If you are repeating a test or you are rerunning the test 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 earlier 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 text that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, it can be used by any member of the project.

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 runtime 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

    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 third-party libraries 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 you must 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 in this step 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.

    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 in this step 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 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 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 there were no agents configured in the test asset. If the test asset had agents configured, the default agent is the first one 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 in this step 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 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 have 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 either started or scheduled a test run for 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 139.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 140.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Tests configurations and test runs

Test run configurations on page 96

Configuring a run for Compound Tests that contains performance tests

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 you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset that you want to run 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.
   - Create a filter query by using the **New filter** option. Enter the relevant parameters and apply the filter query. You can save the filter query for retrieving it from the list of saved filters.
   - Retrieve a saved filter by using the **Open filters** icon. Select and apply the saved 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 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.

   - 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 earlier committed versions.

- 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 have 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 and the default version is selected for the test run.
- If you are repeating a test or you are rerunning the test 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 earlier 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 text that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, it can be used by any member of the project.

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 runtime 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

    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 third-party libraries 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 you must 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 in this step 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.

    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 in this step 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 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 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 there were no agents configured in the test asset. If the test asset had agents configured, the default agent is the first one 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 in this step 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 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 have 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 either started or scheduled a test run for 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 139.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 140.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Tests configurations and test runs

Test run configurations on page 96

Configuring a run for Rate Schedules or VU Schedules

You can configure Rate Schedules or VU Schedules 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 schedules on page 84 if you are configuring a run for Rate Schedules or VU Schedules that have JMeter tests.

1. Open the project that contains the test assets you have added from the Git repository and click **Execution**.
2. Select the branch of the repository that contains the test assets that you want to run.

   All test assets in the selected branch are displayed on the **Execution** page.
3. Identify the test asset that you want to run 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.
   - Create a filter query by using the **New filter** option. Enter the relevant parameters and apply the filter query. You can save the filter query for retrieving it from the list of saved filters.
   - Retrieve a saved filter by using the **Open filters** icon. Select and apply the saved 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 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.

   - 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 earlier committed versions.

◦ 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 have 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 and the default version is selected for the test run.

◦ If you are repeating a test or you are rerunning the test 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 earlier 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 text that you entered is displayed for the test under the **Labels** column on the **Results** page. After you have created a label, it can be used by any member of the project.

   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 runtime 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

   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 third-party libraries 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 you must 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 in this step 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.

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 in this step 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 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 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 there were no agents configured in the test asset. If the test asset had agents configured, the default agent is the first one 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 in this step 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 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 have 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 either started or scheduled a test run for 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 139.
- You can cancel a scheduled test run, from the **Execution** page. See Canceling a scheduled test run on page 140.
- You can view the progress of the test from the **Progress** page. See Viewing the progress of running test assets on page 134.
- 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 151.

Related information

Resetting the configuration settings for a test run on page 137

Tests configurations and test runs

Test run configurations on page 96

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. 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`. 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 96

Monitoring a test run

Checking logs on page 136

Canceling a scheduled test run on page 140

Stopping a test run on page 139

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 96

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 96.

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 96

Viewing the progress of running test assets on page 134

Canceling a scheduled test run on page 140

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 96

Viewing the progress of running test assets on page 134

Stopping a test run on page 139

Management of virtualized services

You can find information about the tasks that you can perform and manage on 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: 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. This release contains access to the virtualization features in Rational® Test Automation Server as a Tech Preview. The Tech Preview is intended for you to view the capabilities offered for virtualized services supported by Rational® Test Automation Server in V10.1.0, and to provide your feedback to the product team.

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 on stubs that you can perform on Rational® Test Automation Server:

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.
- You must have 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.

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.

1. Open the project that contains the API suite with 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 in the test asset that you want to start.
6. Select the time to schedule the starting of the stub from the following options:
   - Select **Now** to initiate the start immediately after you click **Execute**.

     Important: Do not click **Execute** until you have configured the other settings in the **Execute test asset** dialog box.
   - Select **Later** and configure the date and time to schedule the starting of the stub.

Note: You can ignore the **Label** field for the stubs.

7. 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.

8. 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.

When you are done testing the stub, you must stop the running stub, see Stopping a stub on page 149.

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 that use a transport for tests in Rational® Integration Tester and committed the test resources to the remote repository.
- Copied the third-party application libraries of the transport technology. See Copying third-party application Jars to Kubernetes on page 65.
- Started the stubs on Rational® Test Automation Server and ran tests on the stubs. See Starting a stub on page 142.

About this task

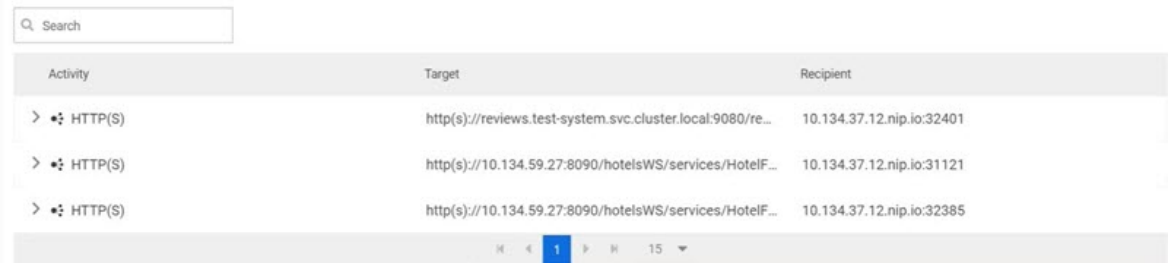
After you run the stubs that are connected to Rational® Test Automation Server, you can view details of the stubs that ran and the routing rules for the stubs, from the **Routing Rules** page.

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.

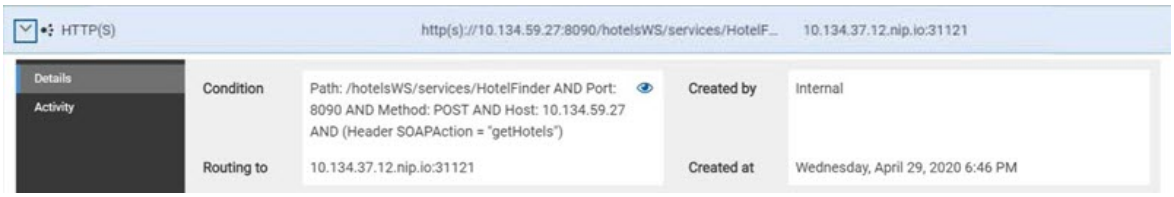


   You can find the following details about the agents or proxies:

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:

<table>
<tr><th>Tab</th><th>Description</th></tr>
<tr><td>Details</td><td>The Details panel provides the following information:
<table>
<tr><th>Option</th><th>Description</th></tr>
<tr><td>Condition</td><td>Lists the condition that causes the traffic to be recorded or routed to a virtual service.</td></tr>
<tr><td>Routing to</td><td>Lists the stub or agent to which traffic is routed.</td></tr>
<tr><td>Created by</td><td>Lists the user who created the rule.</td></tr>
<tr><td>Created at</td><td>Lists the date and time when the rule was created.</td></tr>
</table></td></tr>
<tr><td>Activity</td><td>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.</td></tr>
</table>

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 146.
- You can stop the stubs that are running on Rational® Test Automation Server. See Stopping a stub on page 149.

Related information

Management of virtualized services on page 141

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 that use a transport for tests in Rational® Integration Tester and committed the test resources to the remote repository.
- Copied the third-party application libraries of the transport technology. See Copying third-party application Jars to Kubernetes on page 65.
- Started the stubs on Rational® Test Automation Server and ran tests on the stubs. See Starting a stub on page 142.

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**, **From**, **To**, **Query step**) 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:

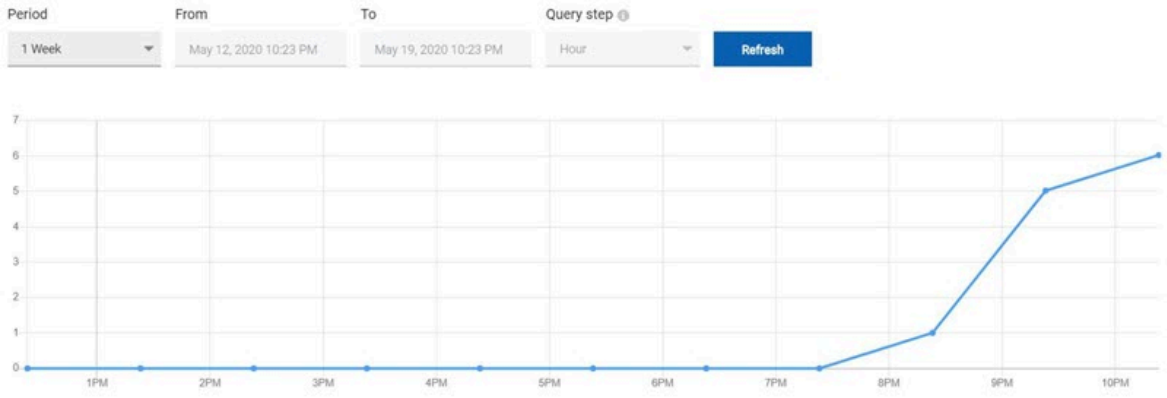
- You log out of the server and log in again.
- You log in after a server restart.

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.

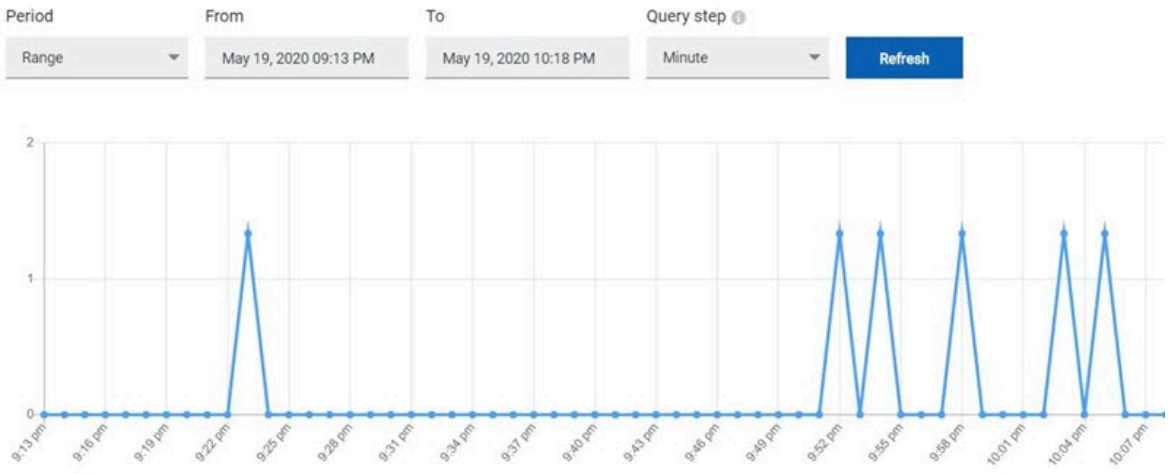


   The metric data that is reported is for the number of calls that are received by a stub. These calls are requests, messages, or other calls that are received by the stub.

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. **Note:** This is the default view.
1 Year	Displays the stub usage data that is gathered during the previous year and grouped in a daily interval.
Range	Displays the stub usage data that is gathered during the date range specified in the **From** and **To** fields. The data is grouped in the time interval that is selected from the **Query step** list. The options available for the time interval are as follows: ◦ Minute ◦ Hour ◦ Day ◦ Week ◦ Month ◦ Year

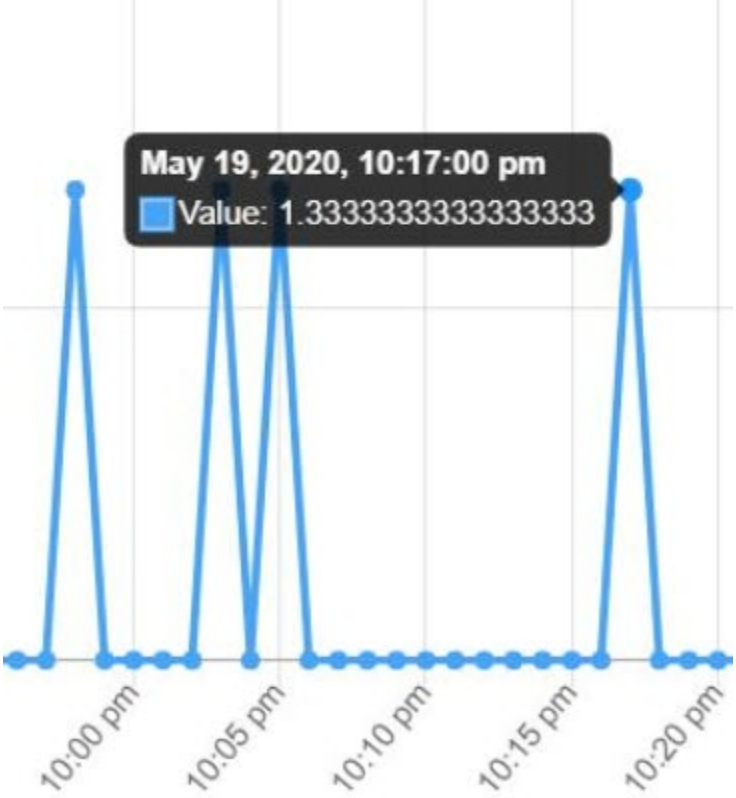
Note: You can refresh the view at any time to present the graph for the latest data by clicking **Refresh**.



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 call received.
- Number of calls 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 144.
- You can stop the stubs that are running on Rational® Test Automation Server. See Stopping a stub on page 149.

Related information

Management of virtualized services on page 141

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 142.

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:

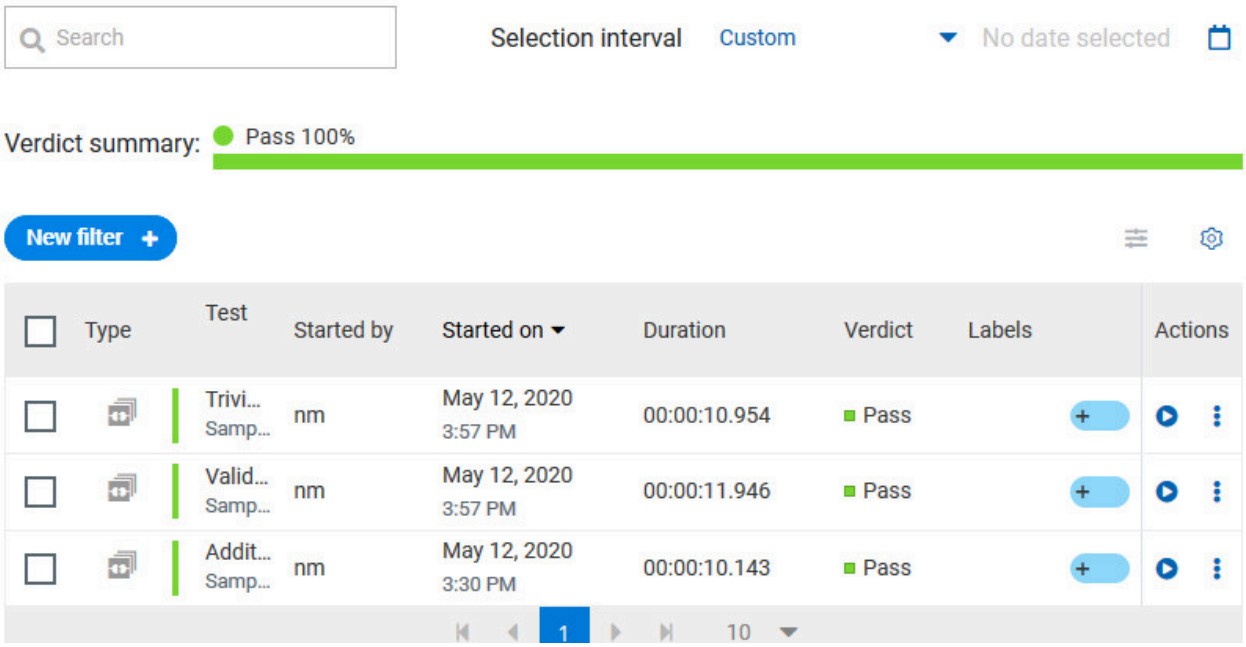
- Searching for test results. You can filter test results by using any of the following ways:
  - By using the **Search** field to search for results by name.
  - By selecting predefined time interval from the **Selection interval** drop-down list.
  - By selecting **To** and **From date** from the **Date 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.
- 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 default, 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.

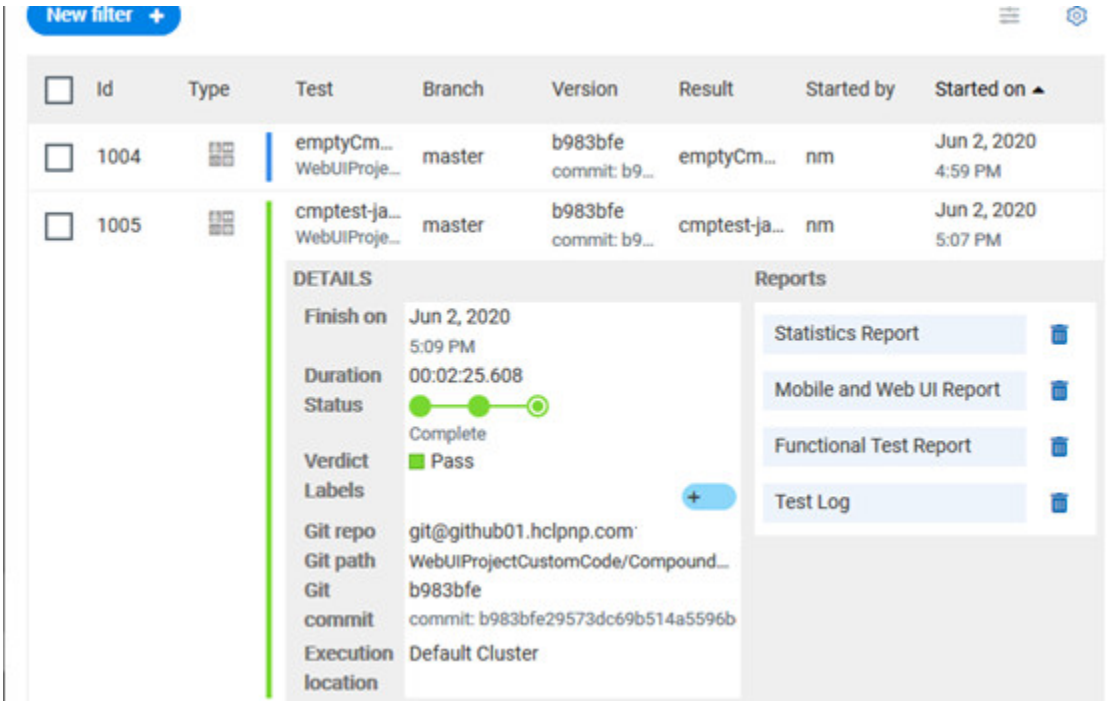
  Note: The execution location can be the default cluster or the location that you indicate to override a docker host or the agent location when you execute a test. For more information about test configuration, see the link at the end of this page.

- 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.

Test logs are delivered in a Jaeger format by default for executed compound tests and schedules. You can still set the `-history testlog` Program Argument to re-enable the traditional test log format when you run the tests.

The reports are generated for Compound Tests, Schedules, and Test Suites. They can be run from a desktop client or from Rational® Test Automation Server. The following table lists the links to help pages for each type of report.

Table 4. Analyzing Reports

Report	Tests type	Product	More information
Functional Test Report	API Suite	IBM® Rational® Integration Tester	Viewing reports
	Compound Test AFT Suite	IBM® Rational® Functional Tester	Unified Report

Table 4. Analyzing Reports (continued)

Report	Tests type	Product	More information
Mobile and Web UI Report	Compound Test	IBM® Rational® Functional Tester and IBM® Rational® Performance Tester	Mobile and web UI test result reports
	AFT Suite	IBM® Rational® Functional Tester	Unified Report
Statistics Report	Compound Test	IBM® Rational® Functional Tester and IBM® Rational® Performance Tester	Reports and counters Service Performance report From the Statistic reports, you can add counters. For more information, refer to Adding additional counters on a separate page. You can also display the statistics of counters on graphs. For more details, refer to Displaying counter data in tables or as graphs.
	AFT Suite	IBM® Rational® Functional Tester	
	VU Schedule	IBM® Rational® Performance Tester	
	Rate Schedule		
Test Log	Compound Test	IBM® Rational® Functional Tester and IBM® Rational® Performance Tester	
	AFT Suite	IBM® Rational® Functional Tester	
	VU Schedule	IBM® Rational® Performance Tester	
	Rate Schedule		

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 96

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.

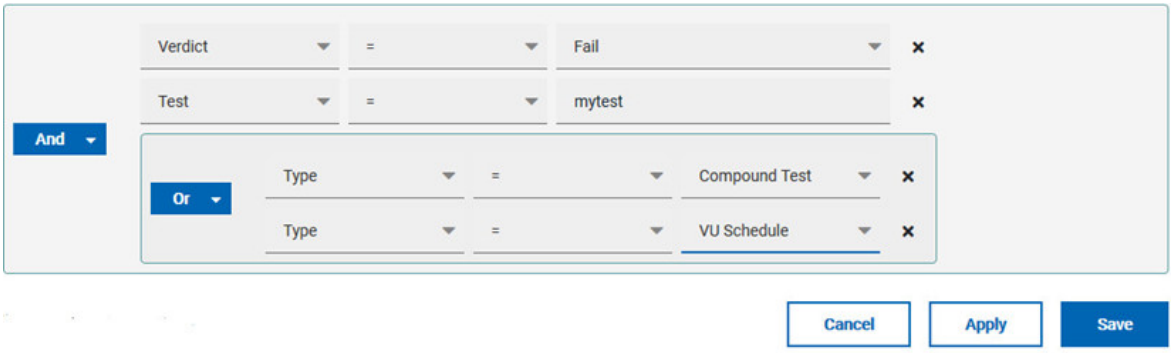
   Example

   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.



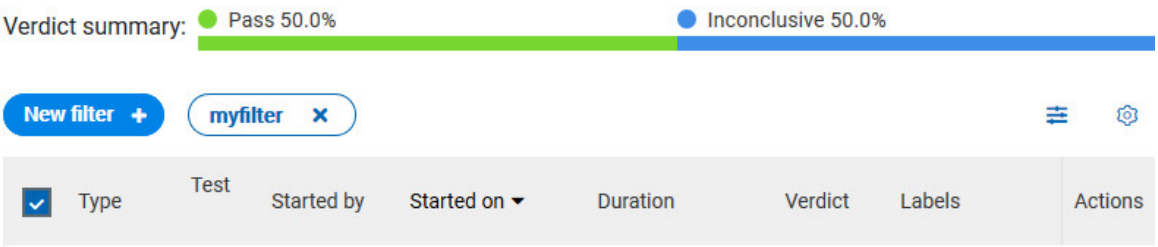
10. Click **Save**.

Proceed as follows to select your filters:

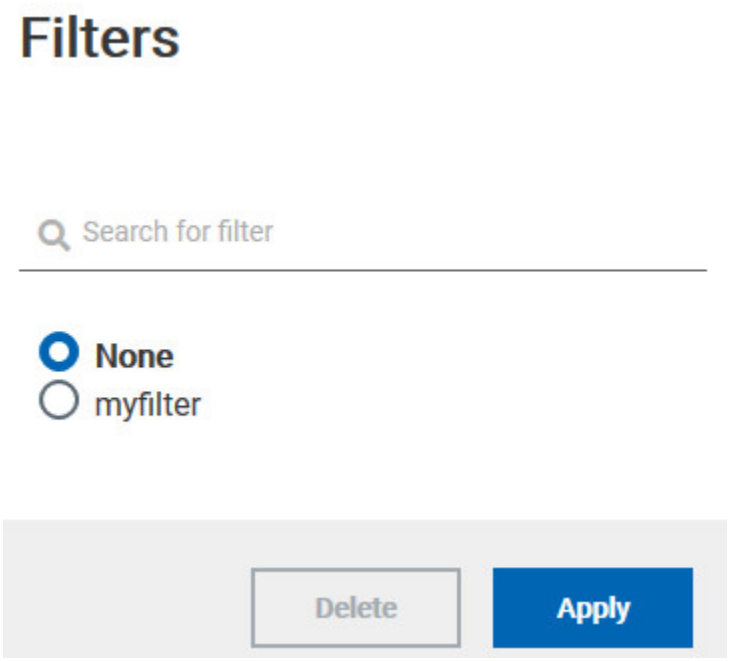
11. Enter a name for the filter and save in the dialog box that is displayed.

    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

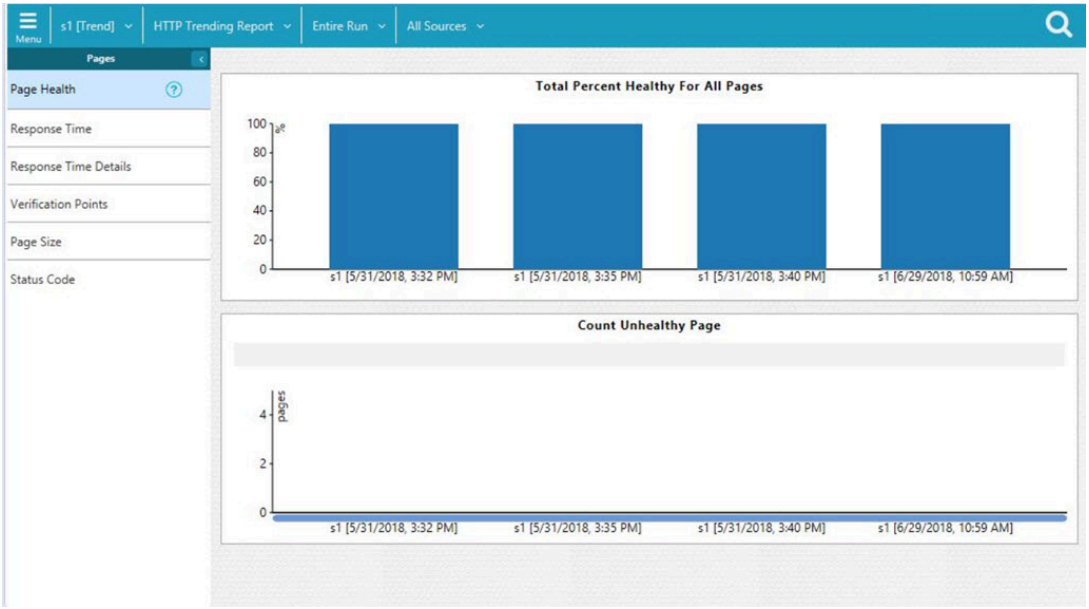
When the performance test runs are complete, you can view the trend of response time for an application over a period of time in IBM® Rational® Test Automation Server from trending reports. In addition to the response time, you can view the trend for the loops, transactions, and performance requirements 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 in the **Label** field to enter a new label, and press the **Enter** key.
5. 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.
6. Click the **Data Sources** tab and select an override option if multiple data sources were defined for the initial test run.
7. 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.
8. Click the **Location** tab to override the default cluster location if the test asset has a docker host or static agents configured.
9. Click **Advanced** to modify the advanced parameters that were set for the initial test run.
10. Enter a new **Program Argument** if applicable to the test.
11. Enter a new instance of **Java arguments** if applicable to the test.
12. Enter the environment variables that must be passed to the test run at runtime in the **Environment Variables** field.
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.

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.

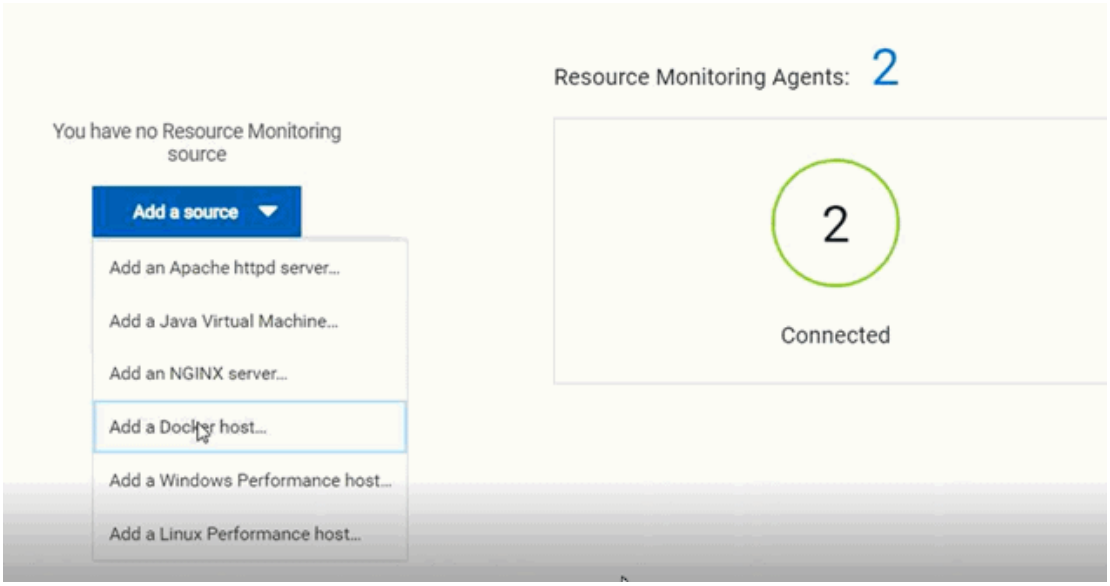
To monitor resources on a local system, you must add the source of the resource monitoring data from the Resource Monitoring page. Then you have to enter some connection settings and select the performance counters to monitor the sources. The metrics are exposed in a graph for each selected counter.

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 you can perform in Rational® Test Automation Server.

Related information

Exporters and Integrations

Starting a Java Virtual Machine on page 182

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

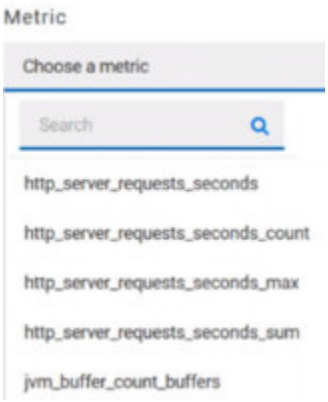
You must have at least one exporter or software to expose monitoring metrics with an OpenMetrics or Prometheus format.

You must have created a project in Rational® Test Automation Server.

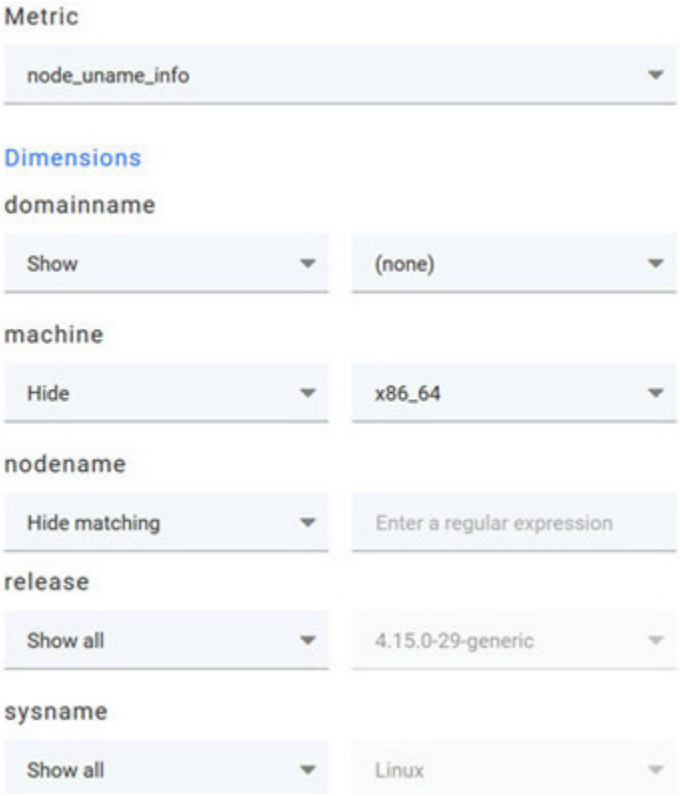
About this task

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.

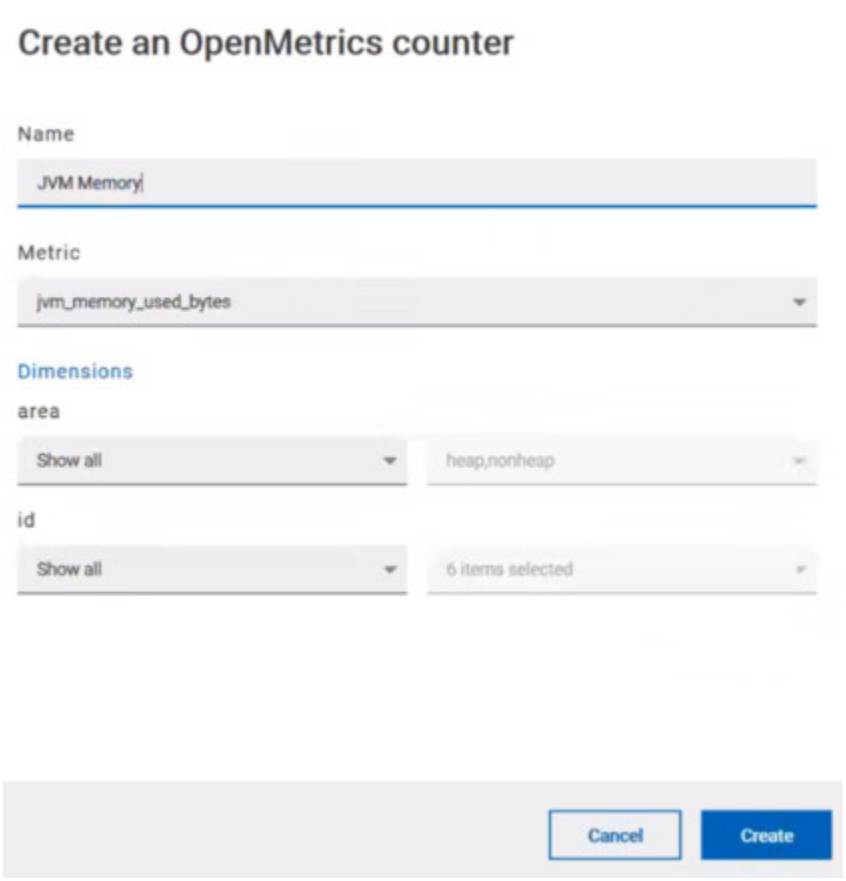


d. Select **Dimensions** from the dynamic fields when they are required.



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 161

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 161

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.

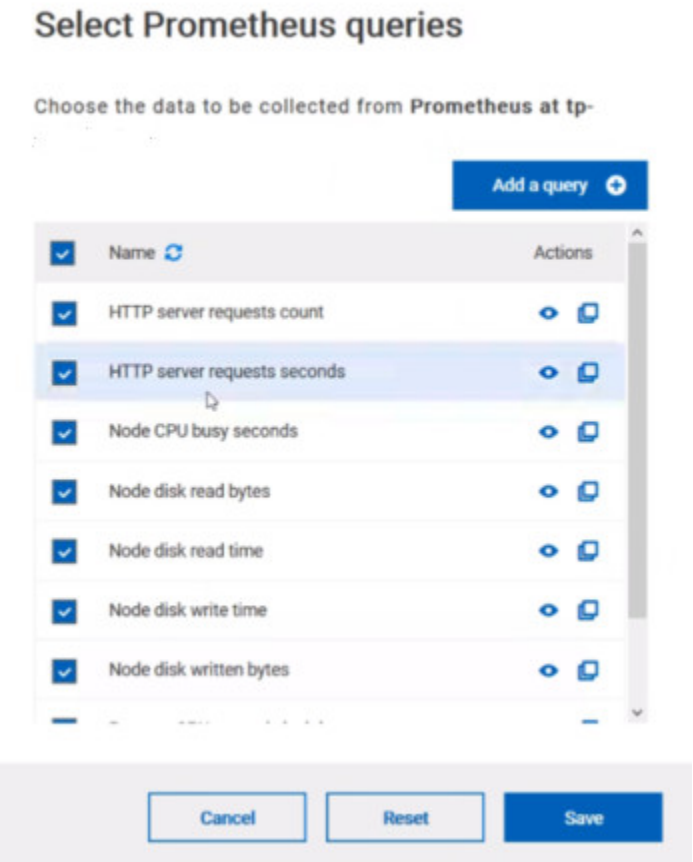
Enter the connection settings, then select queries.

1. Click **Add a source** and select **Add a Prometheus Server**.
2. In the **New server** dialog box, enter the connection settings.
    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:

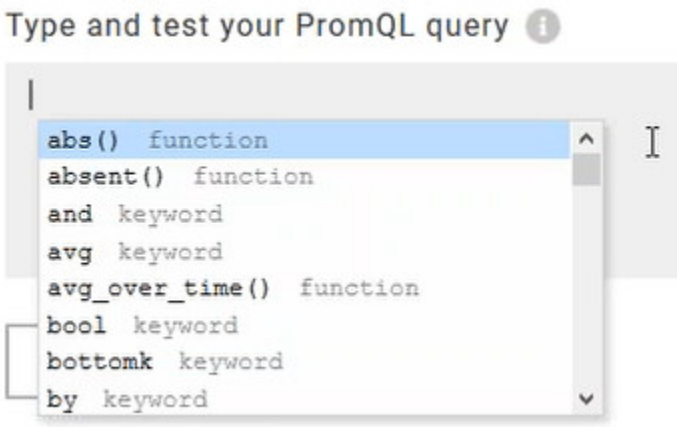
    a. Click **Add a query**.

    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.**

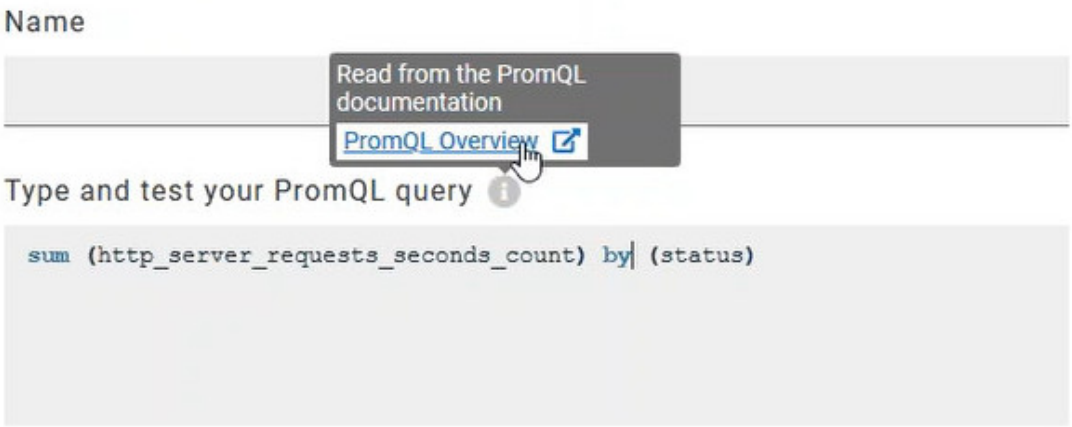
    Result

    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.

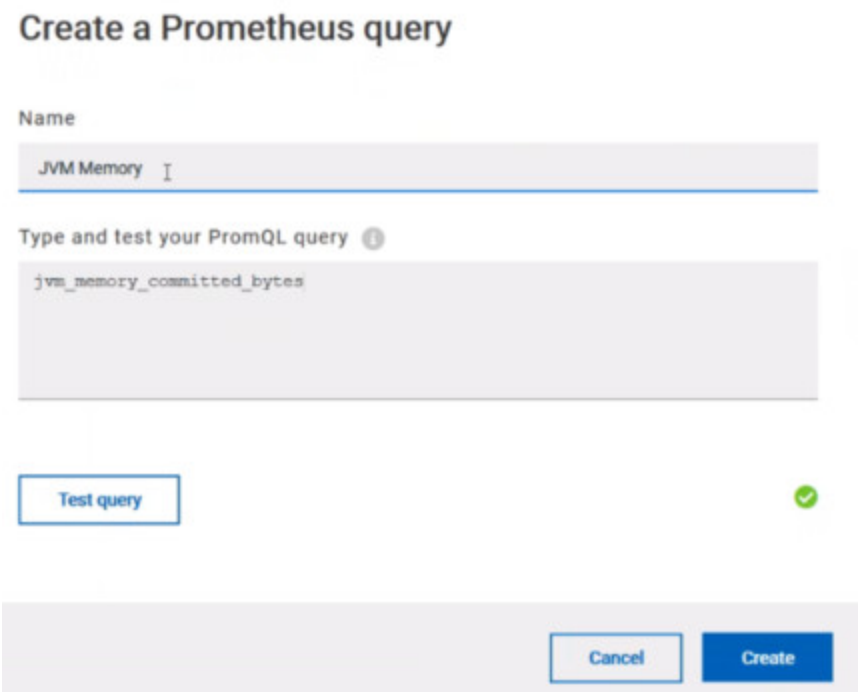
b. Rename the duplicated query in the **View the Prometheus query** dialog box.
Make sure that the name you give to the query is not already used.

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 161

Monitoring host resources on page 171

Viewing the performance metrics

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 Connecting agents to the Resource Monitoring service on page 175.

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

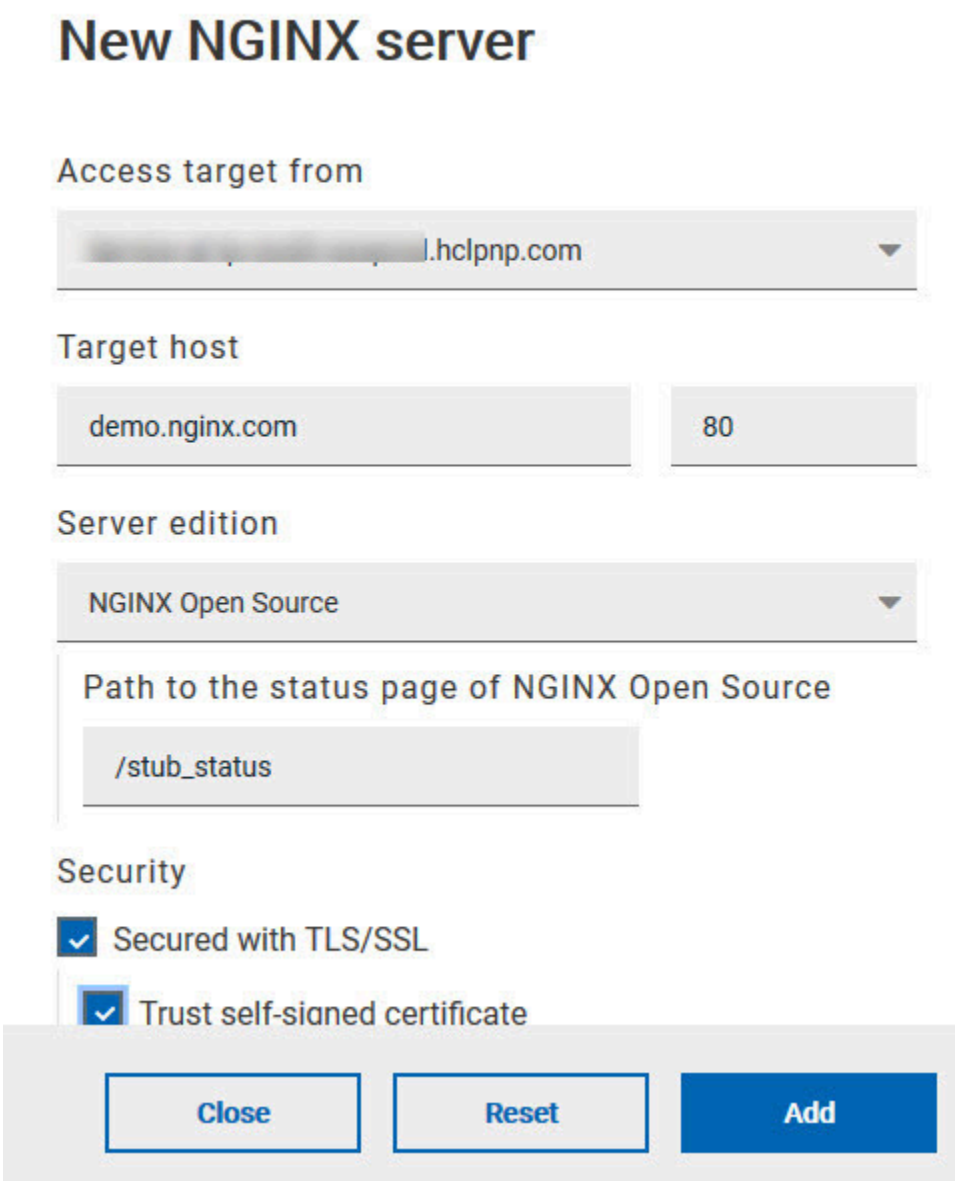
    To accept the server certificate.

    Do not verify host name

    To ignore verification of the host name in the certificate.

Require credentials

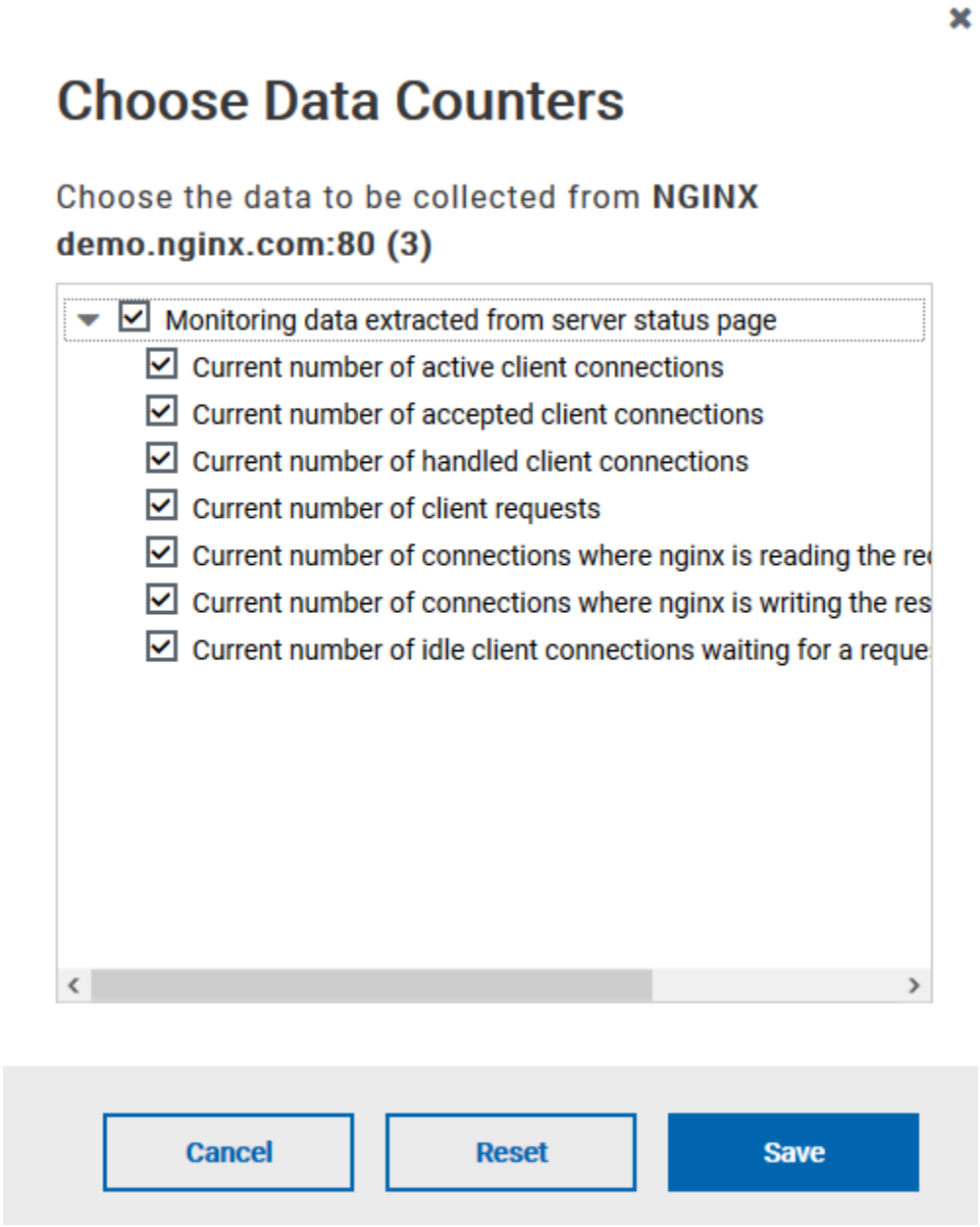
If the server requires log-in credentials, enter a **User name** and **Password**.



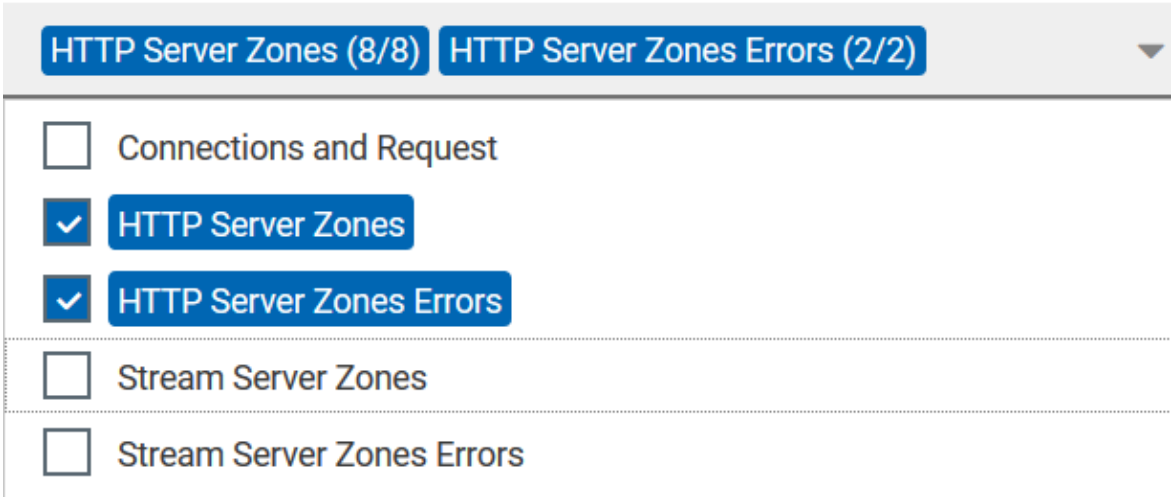
For remote hosts that are already connected to Rational® Test Automation Server through an agent, you have only the **Access target from** field enabled.

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.

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.

When the agent is connected to IBM® Rational® Test Automation Server, it is added to the main page of the Resource Monitoring service. Then, you have to add the agent as a monitoring source and select performance counters to monitor.

Related information

Resource Monitoring service on page 161

Connecting agents to the Resource Monitoring service

You must install the Resource Monitoring agents on the target host for which you want to monitor the performance statistics. The agents must be run to establish a connection with the Resource Monitoring service.

Before you begin

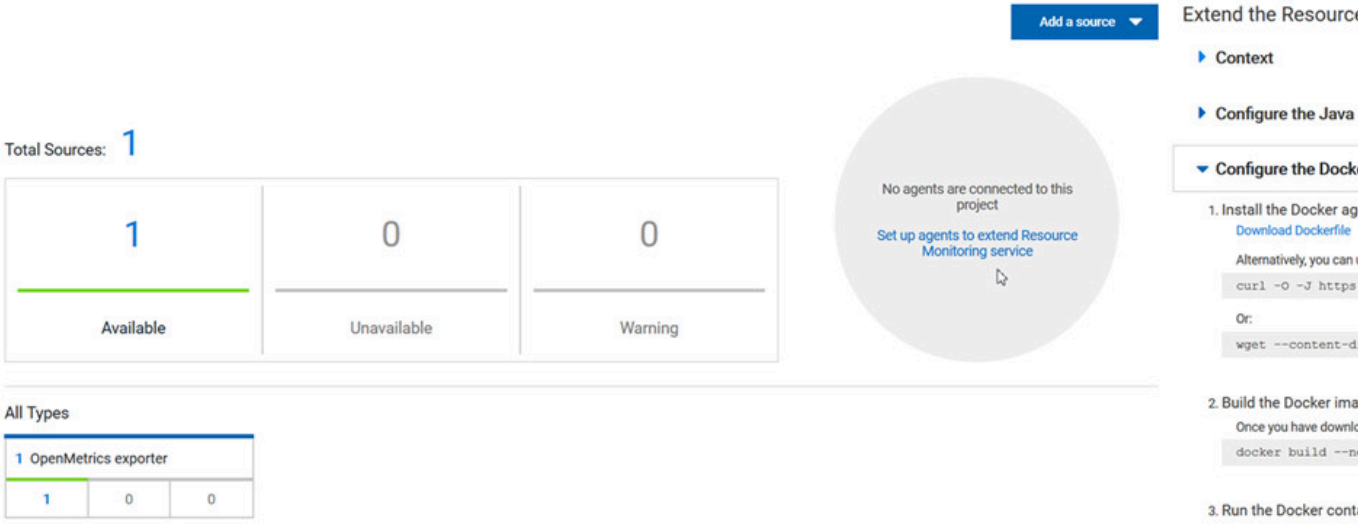
- The Resource Monitoring service does not require access to the agent host but the agent must have reached the service host over HTTPS.
- The Java agent must have been launched from a jar file and requires a Java 8 virtual machine.
- The Docker agent must have been launched in a Docker container and 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. In IBM® Rational® Test Automation Server click **Set up Agents to extend Resource Monitoring service** 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. To download the agent, click **Download jar file** for a Java agent or **Download Docker file** for 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. This step applies to Docker agents only, copy the build command and paste it in your console to build a Docker image.
4. To run your Agent, configure the command as follows:
    a. Copy the appropriate command respectfully to your environment system with the code snippets.
    b. In your console, points to the directory path 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>
```

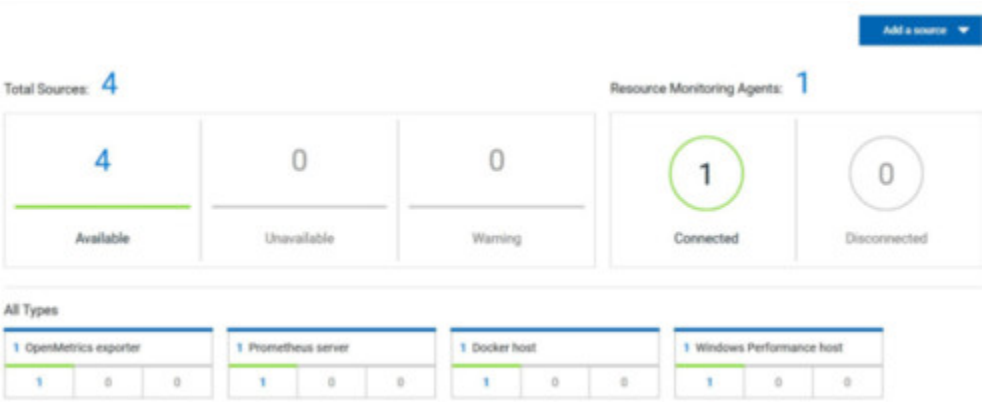
Example

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, it 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

Now, when adding a new source, you can choose where it is collected from, a service or the named agents, depending on the capabilities supported by the environment.

Starting a RM 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

Previously to starting a Resource Monitoring Agent as a service on Windows, you need to have:

- Java 8 installed on the host.
- IBM® Rational® Test Automation Server added to the PATH environment variable.
- An offline token to connect the agent securely with appropriate permissions. 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.
  - The remote firewall is not blocking access; the following Windows Firewall rules must be activated:

- 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. To set up the offline token, copy and adapt the following content to this new RMAgent-winservice.xml file:

```
<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>
```

Note:

- Replace *<service-host>* by the host name of the host that runs IBM® Rational® Test Automation Server.
- Replace *<project-id>* which is the number you find after /projects/ in the browser's URL when browsing to this project.

6. Create a new text file in the same folder and name it `auto-upgrade.bat`.
7. To upgrade the agent .jar file automatically, copy the following content to this new file:

```
@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. In the command prompt, type the following command to create a symbolic link named `RMAgent.jar` to the agent jar file:

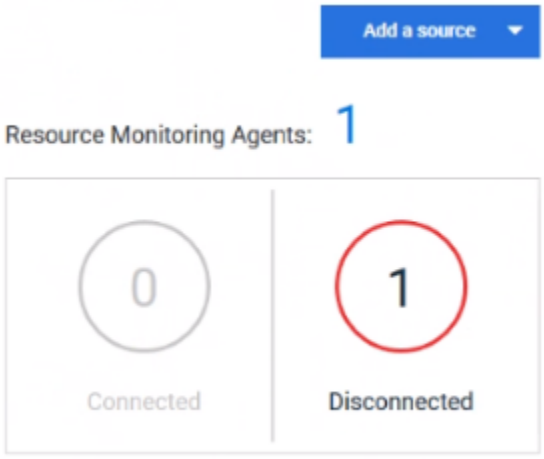
```
mklink RMAgent.jar com.hcl.test.rm.agent-<version-and-datetime>.jar
```

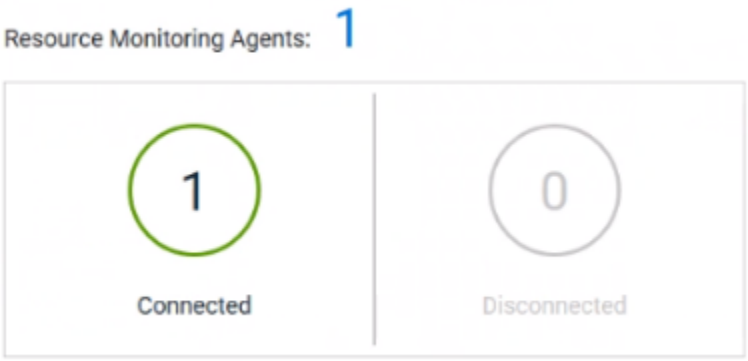
11. Type the following commands:

```
RMAgent-winservice install
```

```
RMAgent-winservice start
```

12. In the **Services** Windows application, you can check whether the service is up and running.
13. You can see in the Resource Monitoring page whether the Resource Monitoring Agent is disconnected or connected.





Results

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`. Look at those files if the Windows service for the Agent is not started or if it does not appear to be connected to the Resource Monitoring service.

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. For more details about monitoring resources, see the related links.

Related information

Other commands

winsw file

Resource Monitoring Agents

Monitoring host resources on page 171

Starting a RM 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

Previously to starting a Resource Monitoring agent as a service on Linux, you need to have:

- Java 8 installed on the host.
- IBM® Rational® Test Automation Server added to the PATH environment variable.
- An offline token to connect the agent securely with appropriate permissions. 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 is the default on most modern Linux distributions. Other ways may 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, under 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
```

Note:

- 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 will start 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 the related links.

Related information

systemd services

Resource Monitoring Agents

Monitoring host resources on page 171

Starting a Java Virtual Machine

To monitor the Resource Monitoring data from a Java Virtual Machine, you must first start the Java Virtual Machine. Some parameters must be set in the command before running the Java Virtual Machine.

1. In the command that is used to run the Java Virtual Machine, enter The IP address of the JVM (local or remote host) and the IP port.
2. **Optional:** You can optionally use authentication security data to launch the virtual machine. In this case, you must enter the name of the password file. You can also enter the name of a privilege file if needed.

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 175

Monitoring host resources on page 171

Integrating with other applications

You can integrate certain applications with IBM® Rational® Test Automation Server to run tests and view the test results.

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**
Jenkins plugin	RTAS-Jenkins-2.0

Rational® Test Automation Server	**10.1.0**
UrbanCode™ Deploy plugin	RTAS-UCD-1.0

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 • Compound tests	IBM® Rational® Functional Tester
• Test Suite	IBM® Rational® Integration Tester
• Compound tests • VU Schedule • 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 185.

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 on your computer.	Installation of Rational Test Automation Server on page 27
2	Create tests in the desktop clients for the application you are testing.	• Rational® Performance Tester documentation • Rational® Functional Tester documentation • 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 205
4	Create an organization and a project in Azure DevOps for running jobs in Azure DevOps pipelines.	Creating an organization
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 185
7	Run tests in an Azure DevOps pipeline.	Running tests in an Azure DevOps Pipeline on page 186

Related information

Integration with Azure DevOps on page 184

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 186.

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 185.
- 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 202.

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.

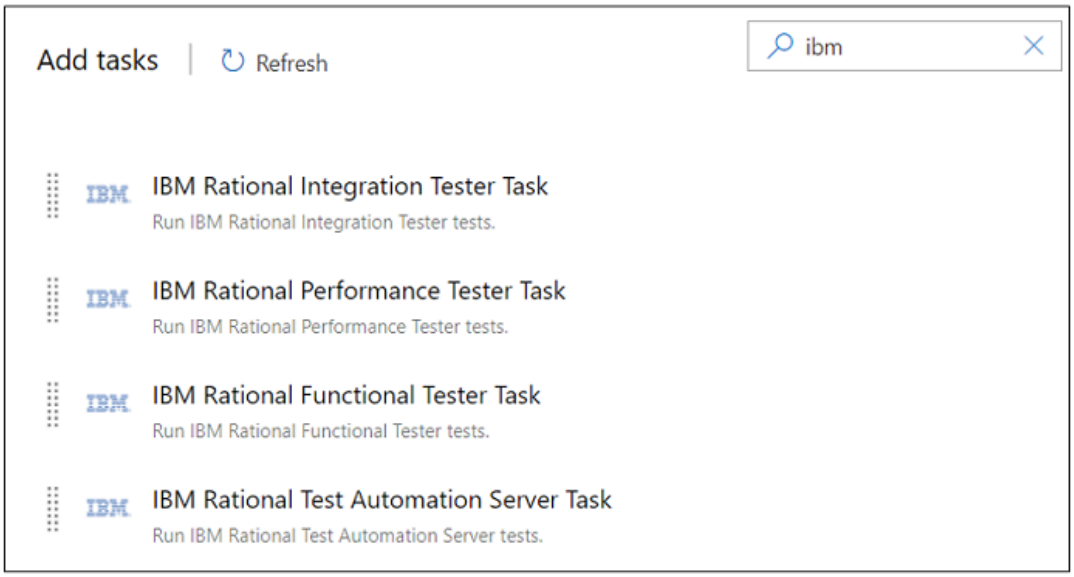
    c. Select the **Agent Specification** for the agent if required.
3. Add a task to the agent job by completing the following steps:

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 188.

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. 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: ▪ **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 token that you generated from Rational® Test Automation Server. ▪ **Service connection name** - Enter a service connection name. ▪ **Description (optional)** - Enter the details of the service connection if required. ▪ Optionally, select the **Grant access permission to all pipelines** checkbox if required. 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. **Note:** 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 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. 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: ▪ Create a new service connection. ▪ Edit or delete an existing service connection that you already added.
Project	Enter the name of the project containing the test.
Branch	Select 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.

Field	Action
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. **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 151.

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 • Compound tests	IBM® Rational® Functional Tester
• Test Suite	IBM® Rational® Integration Tester

Type of test	Desktop client
• Compound tests • VU Schedule • 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 192.

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 on your computer.	Installation of Rational Test Automation Server on page 27
2	Create tests in the desktop clients for the application you are testing.	• Rational® Performance Tester documentation • Rational® Functional Tester documentation • 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 205
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 193
5	Run Rational® Test Automation Server tests on the Jenkins server.	Running Rational® Test Automation Server tests on the Jenkins server on page 193

Related information

Integration with Jenkins on page 191

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-2.0` 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 193.

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 193.
- Generated an offline user token from Rational® Test Automation Server. See Generating an offline token on page 202.
- 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.
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.
Branch	Select the branch from the **Branch** list. The **Branch** list displays the branches available in the corresponding project of Rational® Test Automation Server. **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: ▪ AFTSUITE ▪ COMPOUND ▪ VUSCHEDULE ▪ RATESCHEDULE ▪ APISUITE **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.

Field	Description
	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. **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 151.

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 • Compound tests	IBM® Rational® Functional Tester
• Test Suite	IBM® Rational® Integration Tester
• Compound tests • VU Schedule • 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-1.0` 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 197.

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 196.
- Generated an offline user token from Rational® Test Automation Server. See Generating an offline token on page 202.

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.
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. **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. [2]
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. [2]
Precondition	Specify any conditions that are to be completed before the test runs. You can edit the script by clicking the script displayed. [2]
Use Impersonation checkbox	Select this checkbox to run the test as a different user. [2]
Auth Token Restriction	Set the authentication token actions by applying token restrictions.

2. You need not set this property for a step when you are running an Rational® Test Automation Server test.

Field	Action required for a IBM® Rational® Test Automation Server test
	The **System Default** is selected by default. You can add a new token restriction or edit the one already added. [2]

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

   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 151.

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 5. 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. See Enablement of Resource Monitoring services for a schedule.

Use the offline user token in the following places	More information
Rational® Functional Tester	See Publishing test results to the server.
Third-party integrations	See Integrating with other applications on page 183.

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, which is intended for a test manager, explains how to manage and track your overall test effort.

In Rational® Test Automation Server a test manager has an Owner or Viewer role.

Related information

Managing access to server projects on page 210

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, and classifying encrypted datasets. 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.

The following topics explain server projects and how to add and configure them.

Related information

Becoming a project member on page 210

Managing access to server projects on page 210

Working with Git repositories on page 215

Protecting API test assets by using secrets on page 218

Managing an encrypted dataset on page 225

Default user administration on page 50

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 81.
	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 50

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 ◦ Passphrase
HTTPS	◦ User name ◦ Password
HTTP	◦ User name ◦ 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 IBM® Rational® Test Automation Server displays an `Auth Fail`exception while using the deliver chnages 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 IBM® 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 215

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 218

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 210

Managing access to server projects on page 210

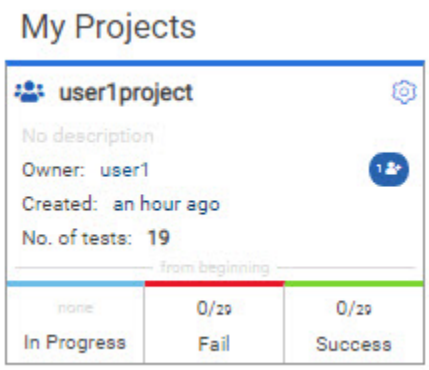
Becoming a project member

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 ◦ Update, archive, list, and view a project ◦ Assign, update, delete, and list user roles for a project ◦ Accept new member requests ◦ Create, update, and delete a dataset classification in a project ◦ Add, update, and remove an encrypted dataset from a classification ◦ Update current row for a listed dataset in a project ◦ Delete, list and view a report ◦ Update, delete, and list a repository in a project ◦ Get the test assets from a repository in a project ◦ Create, update, delete and list secrets in a project ◦ Create, update, delete, and list resource monitoring sources in a project ◦ Get the content of a resource monitoring source in a project ◦ Register, update, delete, and list resource monitoring agents
Tester of a project	◦ All Viewer's actions ◦ List repositories in a project ◦ Run a test and create a report ◦ Stop a test while running ◦ Get the test assets from a repository in a project ◦ Get a list of datasets in a project ◦ Get the content of a dataset

Option	Description
	◦ Update the current row for a listed dataset in a project ◦ Add, update, and remove an encrypted dataset from a classification ◦ Delete a report if you created the report
Viewer of a project	◦ List your own roles on a project ◦ List running tests ◦ List and view a report ◦ List resource monitoring sources in a project ◦ Get the content of a resource monitoring source in a project ◦ 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.

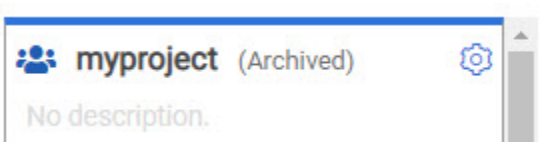
Note:

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 205

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.

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 207

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 ◦ Passphrase
HTTPS	◦ User name ◦ Password
HTTP	◦ User name ◦ 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 IBM® Rational® Test Automation Server displays an `Auth Fail`exception while using the deliver chnages 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 IBM® 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 215

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 220 in Managing secrets collections on page 219.
2. Add secrets in the secrets collection created. See Step 1 on page 221 in Creating a secret in a secrets collection on page 221.
3. Grant access to project members or member roles, who can access the secrets collection. See Step 1 on page 222 in Granting access to members or member roles on page 222.

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 205.
- 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 221.
- 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 222.

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 220.
- To edit or delete a secrets collection, go to Step 4 on page 220.

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 220.
    - To delete a secrets collection, go to Step 7 on page 220.
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. 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. **Note:** You can only change the value of the secret. 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. 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 96

Becoming a project member on page 210

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 205.
- Configured the repository that contains the test assets in your project. See Adding repositories to a server project on page 207.

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.

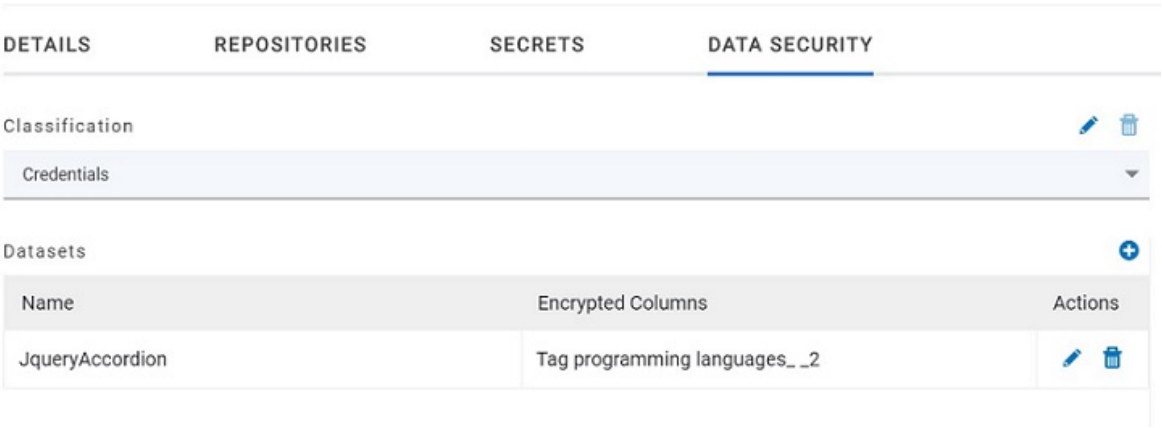
1. Go to the URL.
2. Enter your user name and password, and then click **Login**.
3. From the UI, open your project.
4. Click **Configure**, and then the **DATA SECURITY** tab.
5. Click **New classification** and enter a name for the classification.
6. Click **Create**.

7. Click the **Add** icon to select the encrypted dataset to become part of the new classification.

   Note: The **Add** button is disabled if your repository contains only the datasets that are not encrypted.

8. In the **Add encrypted dataset** window, select a dataset from the list and enter a password of a dataset that was set in the desktop client, and then click the **Add** button.

   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 225.

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 227.
- To delete a classification, go to Step 2 on page 227.

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. In the **Edit classification** window, edit the name of the classification, and 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.

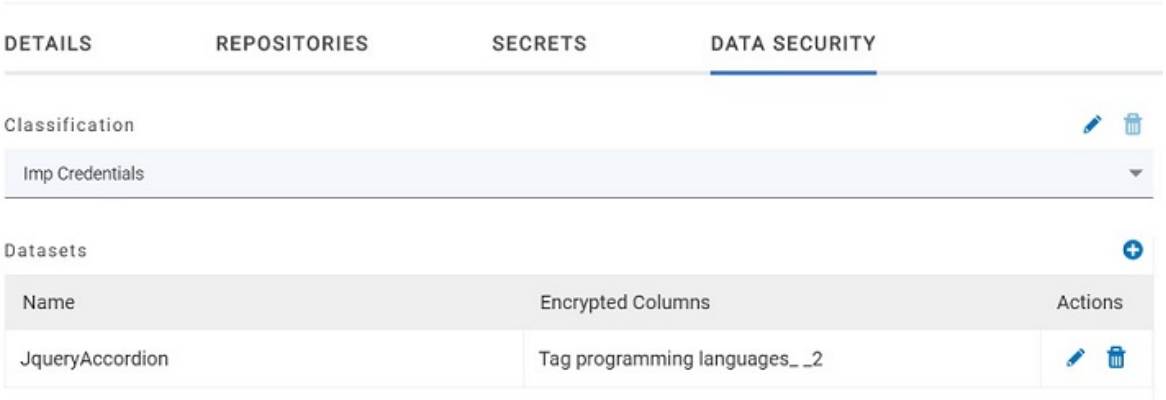
Moving an encrypted dataset to another classification

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. In the **Change the classification for the Dataset** window, select a classification from the list and enter a password.
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 225.

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. In the **Remove Dataset** window, click **Remove**.

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 209.
- Created at least one classification. See Creating a classification on page 225.

1.
2.
3.
4. Select the classification from the drop-down list.
5. To grant access, you can choose any of the following methods:
   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. To revoke access, you can choose any of the following methods:
   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 207

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 231.

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 241.

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 237.

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:

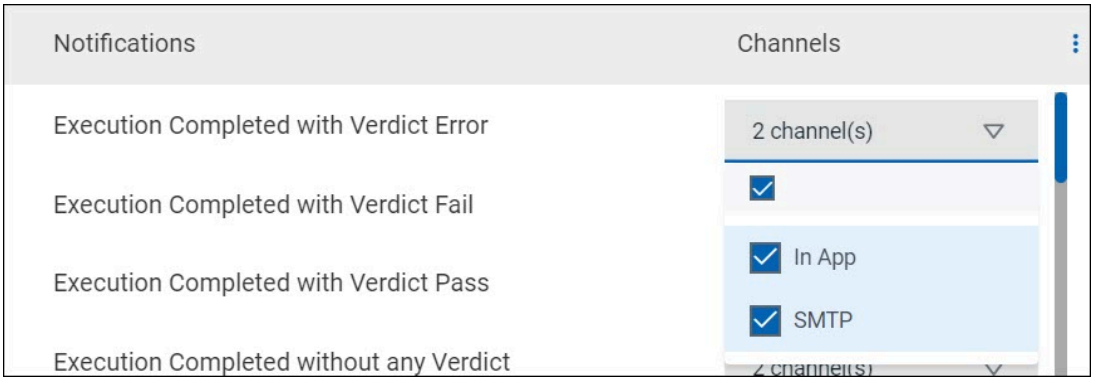
Category	Notifications displayed
User	Include information about actions performed by team space members in the team space.
System	Include information about actions performed by registered users at the server level.

1. Select from the following actions that you want to perform:
   - Modify notification settings. Go to Step 2 on page 232.
   - View notifications. Go to Step 3 on page 233.
   - Unsubscribe from the in-app notifications. Go to Step 4 on page 235.
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: i. Expand the channel list in the row of the server event. ii. Clear the checkbox in the header row to clear the subscriptions to both channels. iii. Click **Apply**. 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: i. Expand the channel list in the row of the server event. ii. Select any of the following actions: ▪ Clear the **SMTP** checkbox. The **In App** selection is retained while the checkbox in the header row and for the **SMTP** options are cleared. ▪ Clear the checkbox in the header row to clear the subscriptions to both channels, and then select the **In App** checkbox. iii. Click **Apply**. 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 236.

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.

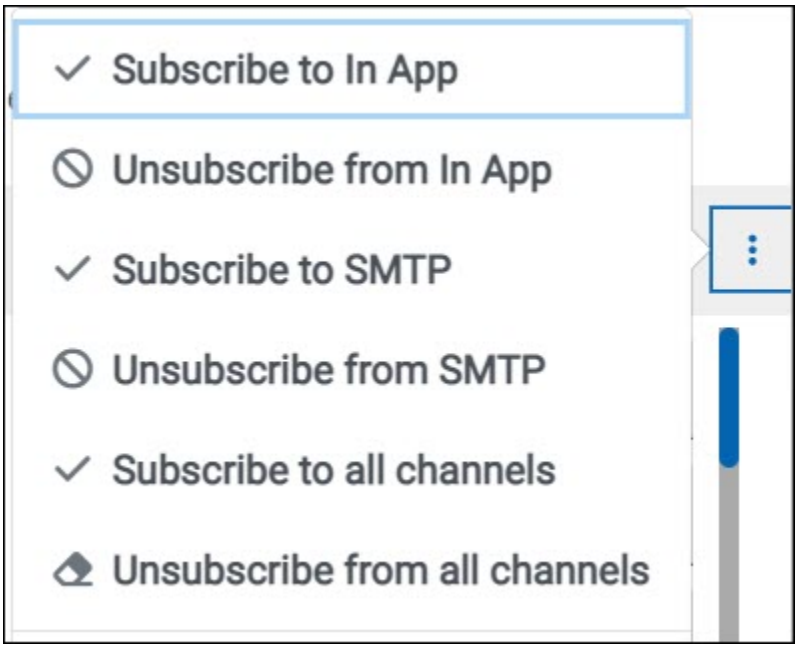
4. Perform the following steps to unsubscribe to the in-app notifications:

   a.

   b.

   c. Click the **Open action menu** icon ⋮ in the row of **Notifications**.

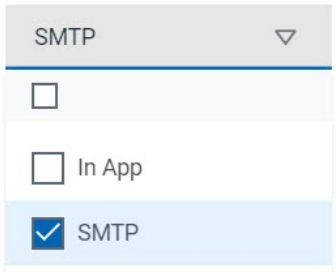
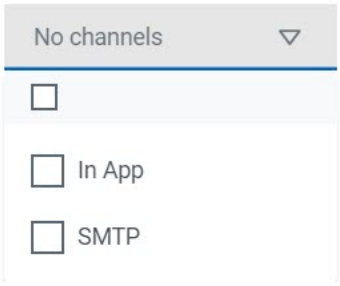
      The following panel is displayed:



      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 230

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 241.
-

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.

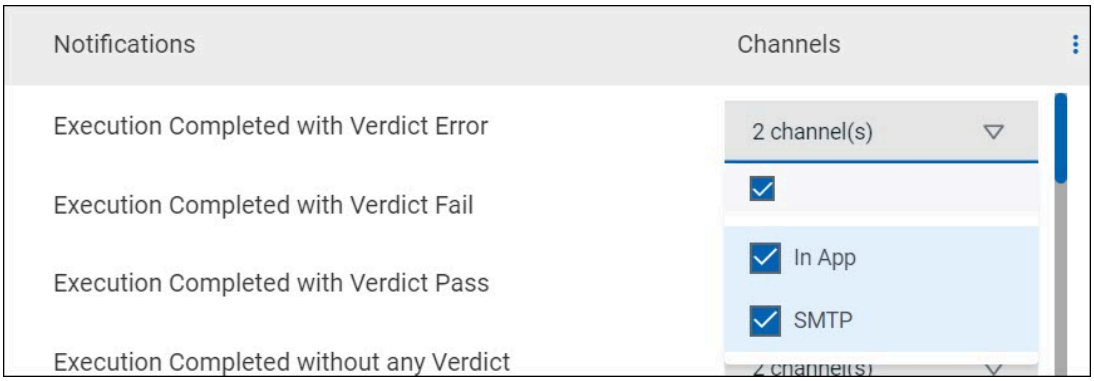
1. Select from the following actions that you want to perform:
    - Modify notification settings. Go to Step 2 on page 237.
    - Unsubscribe from the email notifications. Go to Step 3 on page 239.
2. Perform the following steps to modify the email notifications settings:
    a. Click the **User** icon 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: i. Expand the channel list in the row of the server event. ii. Clear the checkbox in the header row to clear the subscriptions to both channels. iii. Click **Apply**. 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: i. Expand the channel list in the row of the server event. ii. Select any of the following actions: ▪ Clear the **In App** checkbox. The **SMTP** selection is retained while the checkbox in the header row and for the **In App** options are cleared. ▪ Clear the checkbox in the header row to clear the subscriptions to both channels, and then select the **SMTP** checkbox. 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 241.

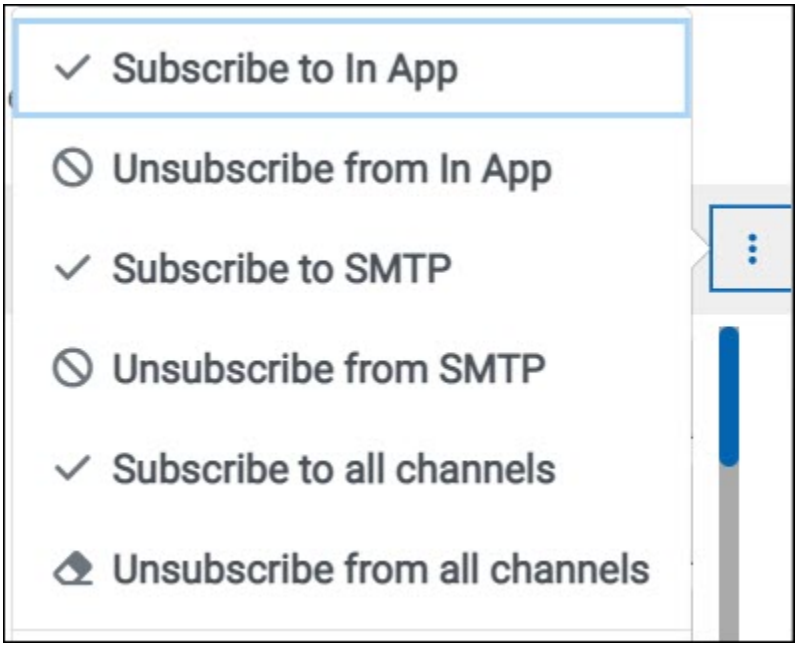
3. Perform the following steps to unsubscribe to the email notifications:

a.

b.

c. Click the **Open action menu** icon in the row of **Notifications**.

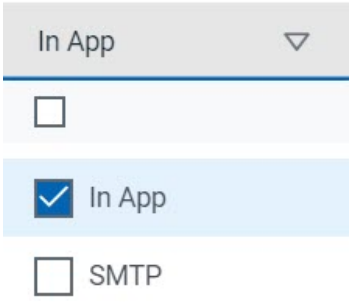
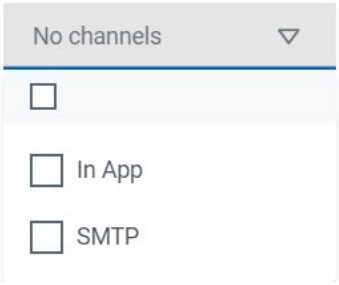
The following panel is displayed:



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. 
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 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 230

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 237.

Related information

Managing notifications on page 230

Chapter 8. Troubleshooting Rational® Test Automation Server

While working with Rational® Test Automation Server, you might encounter some problems that you can easily troubleshoot.

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 6: Troubleshooting issues: installation on page 243
- Table 7: Troubleshooting issues: server administration on page 244
- Table 8: Troubleshooting issues: resource monitoring on page 245
- Table 9: Troubleshooting issues: test or stub runs on page 245
- Table 10: Troubleshooting issues: test results and reports on page 249

Table 6. 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 6. Troubleshooting issues: installation (continued)

Problem	Description	Solution
	• Slow connection speeds. • Insufficient CPU, memory, or disk resources. • A firewall that was configured incorrectly is already enabled.	• To identify the issue, you can perform a diagnostic check by running the following command:microk8s.inspect • Run the following command to see what is not running:kubectl get pods -ARun the following command to get details about the pods: kubectl describe pod • Follow the on-screen instructions to resolve the errors. • Some issues can be solved by re-running the following script:ubuntu-init.sh
The DNS is not working as expected.		You can change the nameservers by using the following command:kubectl edit cm coredns -n kube-system The changes are applied when you restart the `coredns` pod. If you are unsure of how to do this you can run the following script that helps you manage the DNS settings: ubuntu-set-dns.sh

Table 7. Troubleshooting issues: server administration

Problem	Description	Solution
Less recently patched versions of OpenShift can give errors when ever OpenShift performs internal checks. For example, the following error is displayed:	The errors are caused because OpenShift performs the internal checks that are invalid.	Apply the latest OpenShift patches when they become available. If you do not want to apply the patches or cannot apply the patches, you can disable these checks in

Table 7. Troubleshooting issues: server administration (continued)

Problem	Description	Solution
`ValidationError(Route.spec.to): missing required field "weight"`		OpenShift by appending the following option to the helm command: --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.

Table 8. 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 using microk8s on page 39. If not, consult your cluster administrator to get the Promethues server installed and configured.

Table 9. Troubleshooting issues: test or stub runs

Problem	Description	Solution
When you run tests and you encounter any of the following issues: • Out of memory errors. • Observe that the test runs are slow with a high CPU usage.	The issue is seen when the memory that is used by the particular test during the test run exceeds the allocated default memory of 3 GB.	You can increase the memory usage for the test run in any of following ways: • Specifying a maximum heap size for the test run. • Specifying the container memory limit explicitly for the test run.

Table 9. Troubleshooting issues: test or stub runs (continued)

Problem	Description	Solution
		You can set the heap size or set the memory limit by performing the following steps when you configure the test run: 1. Click **Advanced** in the **Execute test asset** dialog box. 2. Enter any of the following arguments in the **Java Arguments** field: ◦ `-Xmx4g` Configuring this argument enables the allotted *3 GB* memory to be increased to *4 GB*. ◦ `-Dexecution.resource-.memory.limit=<custom_memory_value>Gi` **Note:** You must enter the value for the memory limit that you want in <custom_memory_value>. For example, if you want to set the memory limit to *5 GB*, the argument can be: `-Dexecution.resource-.memory.limit=5Gi`. The argument enables the memory to be set

Table 9. Troubleshooting issues: test or stub runs (continued)

Problem	Description	Solution
		to *5 GB* and overrides the default memory size. 3. Configure the other settings for the test run and run the test. **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.
You are not able to run the Istio stubs from the **Execution** page.	The cause might be that you have not enabled the service visualization 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: • For Multi-tenant clusters 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: --set execution.istio.enabled=true Then enable service virtualization in specific namespaces using this command:

Table 9. Troubleshooting issues: test or stub runs (continued)

Problem	Description	Solution
		kubectl create rolebinding istio-virtualization-enabled -n {namespace} --cluster-role={my-ots}-execution-istio-test-system --serviceaccount=test-system:{my-ots}-execution **Note:** Uninstalling the chart does not clean up these manually created role bindings. • For Single-tenant clusters 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: --set execution.istio.enabled=true --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 when it was created.	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 10. 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: • Might be that you have not installed Jaeger at the time of server installation. • The Jaeger trace is not supported for the particular test you ran.	Check for any of the following solutions: • Verify that Jeager was installed in Helm at the time of server installation. See Installing the server software on Ubuntu using microk8s on page 39. If not, consult your cluster administrator to get Jeager installed and configured. • Verify that the tests you ran are supported for Jaeger traces. See Test results and reports overview on page 151.

Security considerations for Rational® Test Automation Server

For Rational® Test Automation Server, you can take actions 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 ccl
- Ports, protocols, and services on page ccl
- Customizing your security settings on page ccl
- Setting up user roles and access on page ccl

Enabling secure communication between multiple applications

The majority of communications are sent over TLS to port 443 (see Ports, protocols, and services on page ccl). 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.

TCP port 7085+ is used by test execution agents to poll for work.

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 51.

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 51.

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 50.

Notices

© 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, 2019.

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 26
    keyboard 26
Accessing secrets 218
administration default security
  IBM
  Rational
  Test Automation Server
  50
administration ldap security
  IBM
  Rational
  Test Automation Server
  51

B

backing up and restoring user data
  IBM
  Rational
  Test Automation Server
  43, 43, 45

C

Change
  Classification 225
  Dataset 225
  Encrypt 225
Change Classification
  Dataset 76
Configure
  Dataset 76
Create
  Classification 225
Create Dataset 69
Creating secrets 218

D

Dataset 76
datasets
  encrypting 73
  removing encryption 74
Delete
  Classification 225
Deleting secrets 218

E

Edit
  Classification 225
  Dataset 70
Editing secrets 218

G

getting started
  guide 19
Grant
  access 225
  Classification 225
guide
  getting started 19

I

IBM
Rational
Test Automation Server
  accessibility 26
  backing up and restoring user data 43, 43, 45
  default security administration 50
  disability 26
  keyboard 26
  ldap security administration 51
  managing account settings 62
  repository considerations 205
  test assets 205
IBM
Rational
Test Automation Server
project overview
  test assets statistics 214

M

Manage encrypted dataset 225
managing account settings
  IBM
  Rational
  Test Automation Server
  62
managing project secrets
  Rational
  Test Automation Server
  218

O

Offline tokens 202
Offline user token 202
offline user tokens
  Rational
  Test Automation Server
  202
overview
  server 19

R

Rational
Test Automation Server
  managing secrets 218
  offline user tokens 202
Remove
  Classification 225
  Dataset 225
  Encrypt 225
repository considerations
  IBM
  Rational
  Test Automation Server
  205
Revoke
  access 225
  Classification 225

S

Secrets
  accessing 218
  creating 218
  deleting 218
  editing 218
  viewing 218
security considerations
  overview ccl
server
  overview 19
  task flows 22
Set
  Current row 76
  Dataset 76

T

task flows
  server 22
test assets
  IBM
  Rational
  Test Automation Server
  205
test assets statistics
  IBM
  Rational
  Test Automation Server
  project overview
  214
tests
  datasets 73, 74

V

View
  Dataset 76, 76
  Encrypt 76
Viewing secrets 218