Login Enterprise 6.8

User Guide

This guide is designed to help you navigate the setup and functionality of your Login Enterprise Virtual Appliance. It covers the key steps for getting started, configuring, and using your Virtual Appliance.

You'll find the information organized into chapters:

- Product Overview
- Installation and Deployment
- Configuration
- Managing Tests
- Dashboards and Reporting
- Application Script Recording

Each chapter is broken down into subchapters with dedicated sections.

Navigation tips:

- Use the Table of Contents to jump to any subchapter and section
- Check Release Notes for new features, changes, and bug fixes
- Press **Ctrl+F (Cmd+F on Mac)** to search for keywords or topics
- Follow **cross-references** to explore related resources

Outside the scope:

- Third-party licenses
- Support resources
    - Product support matrix
    - Workload/application script templates
    - Most of the troubleshooting scenarios/known issues

To access these, visit our Online Documentation Platform.

Let's get started!

Table of Contents

- Release Notes
- Product Overview
  - Value and Use Cases
    - Login Enterprise for Endpoints
  - System Requirements
    - Firewall Exemptions
  - System Architecture and Components
  - Glossary
- Installation and Deployment
  - Preparing for Installation
    - Downloads
    - Installation Checklist
  - Deploying the Appliance
    - Deploying the VHD to Microsoft Azure
    - Deploying the OVA Template to VMware vSphere
    - Deploying the OVA Template to Citrix XenServer
    - Deploying the VHDX on Microsoft Hyper-V
    - Deploying the OVA to Nutanix CALM Marketplace
    - Deploying the OVA to Nutanix AHV
  - Installation Walkthrough (Quick Start)
  - Managing Updates and Upgrades
    - Updating the Appliance
      - Updating the Appliance from v5 to v6
    - Updating Debian
- Configuration
  - Appliance
    - Initial Appliance Configuration
    - Managing Certificates
    - Configuring SSH Security and Ciphers
    - Configuring Authentication
      - Single Sign-On (SSO)
      - Configuring SAML
      - Configuring Open ID Connect (OIDC)
    - Role-Based Access Control (RBAC)
    - Managing Licenses
    - Using the Maintenance Menu
    - Getting Started with the Appliance Web Interface
    - Configuring Database Data Retention
    - Configuring External SQL Database
    - Expanding the Appliance Disk Space
    - Performing the Appliance Backup and Recovery
    - Monitoring Active Sessions
    - Monitoring Container Status

        - Monitoring Appliance Health
        - NVIDIA nVector Integration
        - Nutanix Platform Metrics Integration
    - Launchers
        - Launcher Overview and Best Practices
        - Configuring the Windows Launcher
        - Configuring the Linux Launcher
        - Configuring the IGEL 12 Launcher
            - Launcher Connections in IGEL UMS 12
            - Deploying the Launcher through the IGEL App Portal
    - Virtual Users
        - Configuring Logon Components
            - Configuring Logon Processor as a Logon Script
        - Managing Virtual User Accounts
        - Understanding Login Enterprise Engine
    - Managing Notifications
    - Configuring the Event Logger
    - Updating Browser Drivers
    - Using the Public API
- Managing Tests
    - Configuring Continuous Testing
    - Configuring Load Testing
    - Configuring Application Testing
    - Connectors
        - Citrix Netscaler 12.1 and 13.0
        - Citrix Storefront
        - Microsoft RDS
        - VMware Horizon View
        - Windows 365 (Preview)
        - Desktop Connector
        - Custom Connector
            - Configuring the Universal Web Connector
            - Deploying the AVD Connector
            - Custom Events for Connectors and Launchers
            - Named Pipes for Custom Connectors
                - Custom Connector SDK (Named Pipe)
    - Managing Session Metrics (Optional)
        - Managing Session Metrics Using the Public API
        - Session Metrics Library
        - Session Metrics Import Script (Optional)
    - Configuring Providers (Optional)
    - Configuring Environments (Optional)
    - Configuring Locations (Optional)
    - Configuring Applications for Testing
        - Troubleshooting Application Errors
    - Setting the Engine Start Timeout on a Test

- Dashboards and Reporting
  - Dashboards
    - Test Dashboard
    - Operations Dashboard
  - Test Results
    - Viewing Continuous Testing Results
    - Viewing Load Testing Results
    - Viewing Application Testing Results
  - Viewing All Events
  - Performance Measurements
    - Calculating the EUX score and VSImax
    - Monitoring Login Performance
    - Viewing Session Metrics
- Application Script Recording
  - Creating an Application Script
  - Difficult Applications
  - Using the Script Editor
  - Using the Script Recorder
    - WinApp Recorder
      - UI Inspector (Preview)
      - Understanding WinApp Recorder Implementation (FlaUI)
    - Web Recorder
  - Using the Application X-Ray
  - Scripting Functions Overview
    - Web Recorder Functions (Playwright)

Release Notes

Important Notices

- When upgrading to Login Enterprise v6.8 from a version older than 6.3, all existing API tokens will become invalid and must be manually recreated. You can do this in the Login Enterprise UI. For steps, see Adding a System Access Token
- The new UI and Public API Launcher upgrade feature introduced in v6.6 is only compatible with Windows-based Launchers already running v6.6 or higher. Launchers on older versions must be updated manually one final time
- For other upgrade limitations and detailed steps on upgrading the Appliance from v5 to v6, see Upgrading the Appliance from v5 to v6

Features

- Download a PDF report of Continuous Test results for multiple Tests
- View application performance details for Continuous Tests, including success and failure rates for each application, on the new multiple-test results page

Changes/Improvements

- The Login Enterprise UI includes accessibility improvements that address the most common high-severity WCAG 2.1 issues
- The built-in RDP connector supports the new Microsoft security prompts introduced in April 2026
- The Launcher can interact with higher-privilege Windows prompts without itself running as administrator. To use this feature, the Launcher must be installed in a secure location, such as the Program Files directory
- The Appliance supports ECC (Elliptic Curve Cryptography) certificate and its EC private key for front-end TLS
- Launcher restarts and updates can be scheduled through the Public API
- Custom Connector SDK includes an email field
- The Active Sessions page includes search and filters
- The Engine “waiting“ dialog can be moved, allowing underlying screens to remain visible
- The Engine toolbar provides more space to display the full Environment and Account names
- Script Recorder's update name option is always visible instead of appearing only on hover
- UI Inspector action list has been extended with Type to control, Focus control, Focus window, Find control, Find Window, and DumpHierarchy
- Login failure and application failure notification emails include the failure timestamp, Launcher name, Account name, and Target Host

- Login performance, Application performance, and Latency notification emails include the Target Host
- The Continuous Test overview page displays the current schedule type: Always on, Daily, or Weekly
- The list of default Windows 365 scripts has a label identifying the latest version

Bug Fixes

- Performance-above-threshold metrics on the Continuous Test overview page include disabled thresholds
- Upgrades to Login Enterprise 6.5 or later can take a long time for large databases because of a non-optimal database query
- Testing against a Citrix-published desktop with Seamless Mode enabled results in "The launcher did not respond to a session request in a timely fashion" errors. This affects both Storefront and NetScaler connectors
- The Windows 365 connector fails when a temporary window or pop-up appears and disappears during script execution
- Recorded double-click actions are labeled as left clicks in the generated comment and action name
- Application scripts fail to find the application window when mainWindowClass in the StartApplication method contains a colon
- The Save and run Test option in the UI saves changes to the Test but does not trigger a run of the Test

Product Overview

This chapter outlines the key aspects of the Login Enterprise Virtual Appliance, including its value, use cases, system requirements, firewall exemptions, and architecture.

- Value and Use Cases
  - Login Enterprise for Endpoints
- System Requirements
  - Firewall Exemptions
- System Architecture and Components
- Glossary

Value and Use Cases

- Introduction
  - Automated Testing
  - Hardware Benchmarks
  - Migrations
  - Cloud VM Instance Selection
  - Streamlined Change Process
  - SLA Reporting
  - EUX Benchmarks
  - Cost Tracking
  - Additional Use Cases and Benefits

Introduction

Login Enterprise is a comprehensive platform that optimizes and manages End-User Computing (EUC) environments, including physical endpoints (such as laptops and desktops) and virtual environments (such as Virtual Desktop Infrastructure (VDI), Server-Based Computing (SBC), and cloud-based desktop systems). It provides powerful tools to improve performance, scalability, cost management, and troubleshooting across these systems.

Automated Testing

Login Enterprise simplifies the process of testing changes in End-User Computing (EUC) environments. With tools like the Script Recorder, IT teams can automate the testing of new configurations, updates, and system migrations, ensuring that changes are validated before deployment. This helps streamline the change process by identifying potential issues early, reducing manual testing efforts, and minimizing disruptions to users.

Hardware Benchmarks

- **Physical Endpoint Management**: Login Enterprise for Endpoints provides dedicated tools for managing physical endpoints such as laptops and desktops. The platform ensures that physical hardware is properly configured, optimized, and maintained, allowing IT teams to track performance, monitor resource usage, and extend the lifespan of physical devices within the EUC environment.
- **Performance and Scalability**: Login Enterprise helps organizations manage their EUC environments by evaluating hardware configurations, ensuring scalability, and optimizing resource usage across both virtual systems and physical endpoints.
- **Capacity Management**: Using data-driven benchmarks and dashboards, teams can assess resource usage and correlate performance insights across components, optimizing cloud and hardware costs. This ensures efficient resource allocation and avoids overspending, covering both physical endpoints and virtual systems.
- **Extending Existing Investments**: The platform enables scaling environments while maintaining consistent performance and optimizing hardware investments, including

extending the lifecycle of both physical endpoints and virtual environments without compromising performance.

Migrations

- **Streamlined Migrations:** Login Enterprise supports migration teams by running A/B tests, simulating real-world failovers, and ensuring smooth transitions during technology upgrades, migrations, or new deployments. This includes A/B testing for Cloud PC and Windows OS migrations, as well as image testing to ensure that system images are compatible and function as expected in new environments and perform as expected during migrations.

Cloud VM Instance Selection

- **Optimizing Cloud Resources**: Login Enterprise helps cloud architects and infrastructure managers choose the most suitable cloud VM instances for their needs. By using benchmarking and scalable load testing, teams can assess price-to-performance ratios and select the best cloud instance to balance cost with user experience.
- **Cost Optimization**: The platform enables IT teams to evaluate cloud usage patterns and forecast costs effectively. By understanding the cost impact of resource choices, organizations can avoid surprise charges while ensuring performance remains high.

Streamlined Change Process

- **End-User Experience Focus**: Traditional monitoring tools often overlook the end-user experience, but Login Enterprise prioritizes this layer to help teams quickly identify performance issues. By monitoring metrics like login times and application performance, the tool ensures a targeted investigation into potential bottlenecks or slowdowns.
- **Predictive Maintenance**: Using historical performance data, Login Enterprise predicts future trends, helping IT teams identify potential problems before they affect users. This proactive approach helps reduce downtime and maintain seamless user experience.
- **Availability Monitoring**: The tool continuously tests the availability of resources like applications and desktops. This allows IT teams to detect issues before they impact end-users, reducing the number of help desk calls and ensuring high service availability.

SLA Reporting

- **Tracking Service Level Agreements (SLAs)**: Login Enterprise provides detailed SLA reports that demonstrate adherence to performance standards. These reports highlight areas for improvement and help EUC teams build trust with stakeholders by showcasing transparency and accountability.
- **Custom Reports and Alerts**: The platform supports creating customized reports and alert policies that notify teams about availability issues, ensuring quick responses to performance deviations and aligning IT operations with business goals.

EUX Benchmarks

- **Measuring User Experience**: The EUX (End-User Experience) score is a key metric in Login Enterprise, offering an objective measure of desktop environment performance. It tracks factors like response time, availability, and reliability, providing a comprehensive view of the user experience.
- **Proactive Issue Identification**: By setting benchmarks for user experience and monitoring changes over time, teams can detect even minor performance deviations and address issues before they disrupt users.
- **Continuous Improvement**: Login Enterprise facilitates continuous optimization of the user experience by providing clear, actionable insights into performance trends, allowing EUC managers to demonstrate progress and make data-driven decisions to improve overall system performance.

Cost Tracking

- **Managing VDI and DaaS Costs**: Login Enterprise helps organizations optimize costs in Virtual Desktop Infrastructure (VDI) and Desktop-as-a-Service (DaaS) environments by providing visibility into resource usage, helping IT teams manage costs more effectively and identify potential areas for savings.
- **Maximizing VDI Resources**: The platform tracks the total cost of ownership (TCO) and utilization of VDI resources in real-time, helping administrators plan for scalability and resource adjustments, ensuring that the system can support growing business needs.
- **Avoiding Surprise Cloud Costs**: Login Enterprise supports cloud architects and operations teams by balancing user experience with cost. By selecting the optimal mix of on-demand and always-on virtual machines, the platform helps avoid wasted spending and ensures optimal resource usage.

Additional Use Cases and Benefits

- **Role-Based Access Control (RBAC):** RBAC provides fine-grained access management for Login Enterprise, enabling administrators to define and manage custom roles with specific permissions. This functionality allows users to have isolated access to Test configurations, ensuring they can interact only with their designated Tests.
- **Cloud Architects and IT Managers**: Login Enterprise is ideal for cloud architects focused on optimizing cloud resource usage and cost management, as well as IT infrastructure managers, ensuring that all components within the EUC environment work together seamlessly.
- **Performance Analysts**: Performance analysts can leverage the platform's data and insights to identify trends and track key performance metrics, optimizing overall system behavior and user experience.
- **EUC Support Teams**: These teams benefit from advanced root cause analysis and predictive maintenance tools, ensuring rapid identification and resolution of performance issues to minimize user disruption.

In summary, Login Enterprise is an all-in-one solution for managing and optimizing EUC environments. By providing detailed performance insights, cost optimization tools, root cause analysis, and SLA compliance reports, it helps organizations enhance user experience, streamline migrations, extend the lifespan of existing systems, and proactively manage costs. With its focus on both cloud and on-premises environments, Login Enterprise supports IT teams in making data-driven decisions that improve system efficiency, reduce downtime, and maintain high service availability.

Login Enterprise for Endpoints

- Overview
- Use Cases
  - Security Agent Change Testing
  - Windows Update Change Testing
  - Application Change Testing
- Setting Up Login Enterprise for Endpoints
  - 1. Define Your Test Objectives
  - 2. Set Up Test Configuration
  - 3. Configure the Applications (Test Scripts)
  - 4. Establish Baseline Performance
  - 5. Implement Endpoint Changes
  - 6. Execute Tests with Endpoint Changes
  - 7. Iterate Testing
  - 8. Analyze Test Results
- Conclusion
- Best Practices

Overview

Deploying new security measures, updates, or applications is essential for maintaining your organization's security posture and operational efficiency. However, tools like Endpoint Detection and Response (EDR) and Data Loss Prevention (DLP) software, system updates, or application changes can sometimes introduce performance overheads that negatively impact user experience and productivity. This article explains how to use Login Enterprise to automate the testing of endpoint configurations on Windows-based physical devices, ensuring that your endpoints and critical applications maintain optimal performance without compromising security or functionality.

Balancing robust security, seamless updates, and optimal performance is a common challenge for IT organizations. Users expect smooth, uninterrupted experiences, but additional security measures or system changes can strain system resources, leading to frustration and decreased productivity. Login Enterprise provides a solution by simulating real-world user activities and collecting detailed performance metrics. This enables you to assess the impact of various endpoint configurations, whether it's security agents, system updates, or application changes, before deploying them across your organization.

Imagine quantifying the effects of new or updated security measures, system updates, or application changes on factors like application launch times, EUX scores, or CPU usage. With this data, you can make informed decisions, adjust configurations, and ensure that these changes don’t hinder daily operations.

When setting up Login Enterprise for Endpoints, you’ll follow an iterative testing process, no matter which use case you’re focusing on. Start by establishing baseline results, then implement

changes specific to the use case you're testing. After testing, review the results and compare them to your baseline. Based on this comparison, decide whether the Appliance is ready for deployment or if further adjustments are necessary.

For details on testing with different use cases, see Use Cases.

Use Cases

The following use cases describe various testing scenarios that leverage Login Enterprise for Endpoints to assess and ensure optimal performance, functionality, and reliability of physical Windows endpoints. These scenarios focus on how specific changes, such as security agent updates, Windows updates, or application modifications, impact system performance and end-user experience. By capturing key performance metrics and verifying application functionality, these Tests help ensure that system changes do not negatively affect productivity or system stability.

Security Agent Change Testing

This use case evaluates the impact of modifying or installing a security agent on a physical Windows endpoint.

- Start by establishing a baseline by running Tests on a standard Windows image while capturing key performance metrics, such as logon times, application responsiveness, and EUX score, using Login Enterprise for Endpoints.
- Once the baseline is recorded, install or update the security agent, and rerun the Tests.

This helps measure any performance changes and verify that applications continue to operate correctly. The process provides a comprehensive view of both performance and functionality, ensuring that changes to the security agent do not unintentionally hinder end-user productivity or system stability.

Windows Update Change Testing

This use case assesses the effects of Windows updates on endpoint performance and application functionality.

- The process begins by capturing baseline performance data on a physical Windows endpoint using Login Enterprise for Endpoints.
- After applying the Windows update, rerun the Tests to compare measurements such as system responsiveness and resource utilization.

This method provides clear insights into the impact of the update on overall system performance while verifying that essential applications continue to run smoothly. By monitoring any deviations, you can ensure that the update maintains both system reliability and user productivity.

Application Change Testing

This use case analyzes the effects of changes made directly to key applications installed on physical endpoints.

- The testing process starts with establishing a baseline by running Tests to capture the normal performance characteristics of critical applications, including their launch times, response rates, and stability under typical usage conditions.
- After an application update or configuration change is applied, execute the same set of Tests using Login Enterprise for Endpoints.

This systematic approach ensures that modifications do not lead to functionality issues or performance degradation, supporting a smooth transition when updating or altering essential applications.

Setting Up Login Enterprise for Endpoints

1. Define Your Test Objectives

Start by clearly outlining your goals for testing:

- **Identify Critical Applications and Functionality**: List essential applications for user productivity (e.g., Microsoft Office, custom applications, etc.).
- **Set Performance Metrics and Thresholds**: Establish acceptable performance levels for metrics such as:
  - EUX score
  - Session metrics (e.g., CPU usage, memory consumption)
  - Application launch and action times
  - Login durations
- **Collaborate with the Security Team**: Work with teams (e.g., Security, IT, Operations) to understand the specifics of planned endpoint changes (e.g., security agent deployments, Windows updates, or application updates) that may affect performance.

Example and Walkthrough

NOTE: The objective thresholds provided in this section are for demonstration purposes only. These examples should be adjusted according to your specific testing requirements.

- The EUX score should not be more than 1 point lower than the baseline.
- Logon time should not exceed 4 additional seconds.
- Microsoft Edge app performance should not show a performance delta greater than 33%.

For example, if any of these thresholds are exceeded during Security Agent Change Testing, Windows Update Change Testing, or Application Change Testing, the changes will not be approved. However, the changes can be deployed if the thresholds remain within acceptable limits.

2. Set Up Test Configuration

Correctly configuring the test environment is critical, as it directly impacts the accuracy of the pass/fail decisions during the analysis of endpoint changes. Proper configuration ensures that the test results are reliable and meaningful, whether you are testing Security Agent Changes, Windows Updates, or Application Changes.

Example and Walkthrough

Configuring the testing scenario:

1. **Download the Login Enterprise Launcher**. To learn how to do this, see Downloading and installing the Login Enterprise Launcher.
2. **Configure the Launcher (if not already set up)**.

Important Steps to Remember:

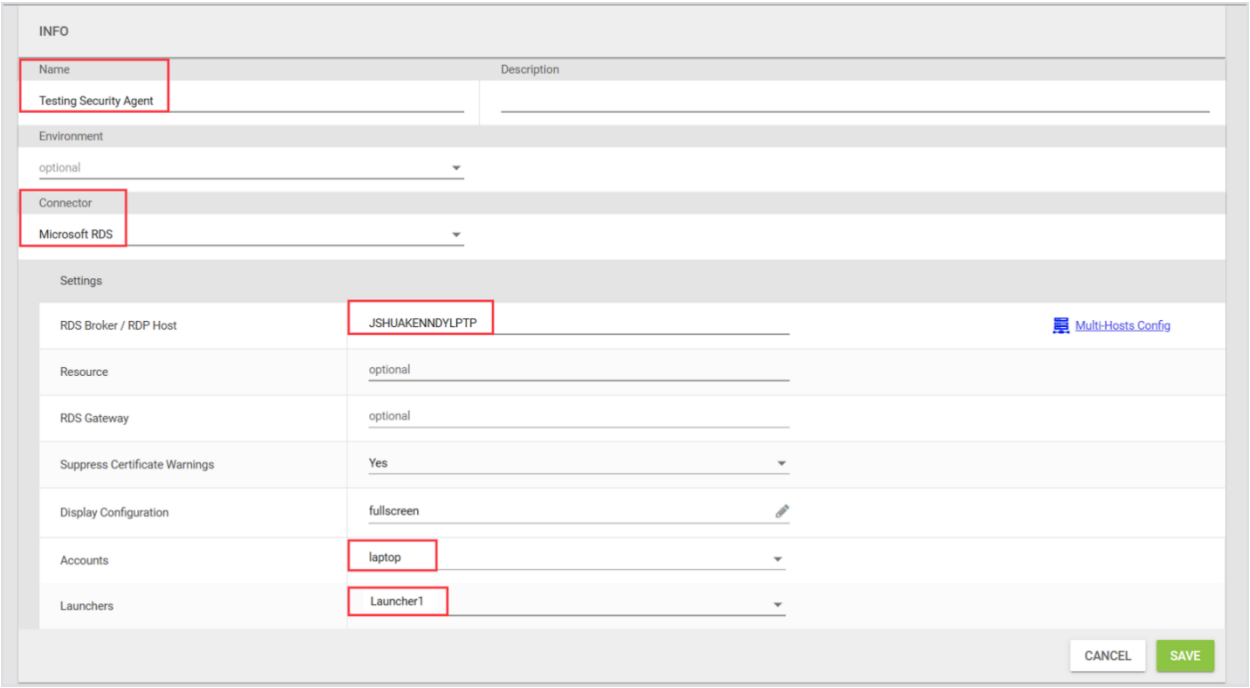
- Keep the Launcher application running throughout the entire testing process.
- Prevent the Launcher screen from locking by disabling the relevant policy, if necessary.
- Before beginning the Login Enterprise testing, manually test a successful connection from the Launcher Host to the device under test.

To learn about Launcher best practices, see Launcher Overview and Best Practices.

3. **Add the Launcher to a Launcher Group**. To learn how to do this, see Launcher Groups and Locations.
4. **Set Up the Physical Endpoint for Baseline Testing**. Ensure the physical endpoint is configured to match the baseline setup for testing. This may include: no endpoint changes installed, a base configuration of the security agent, Windows update configuration, or the initial "A" configuration of any applications. Also, ensure that all desired applications, updates, and configurations are applied to the endpoint.
5. **Add the Login Enterprise Logon Script to the Physical Endpoint**. For more information on the Logon component setup and configuration process, see Logon Components.
6. **Add the Windows Test User** that will be logged into the endpoint under test. For more information, see Creating Virtual Users and Groups.
7. **Add the Test User to a Group**. For more information, see Creating Account Groups.
8. **Create a New Load Test Scenario and Connection**. For more information, see Creating a Load Test.
9. Configure the following parameters for the Test:

Field	Description
Name	A descriptive name for the test scenario.
Connector	Select Microsoft RDS

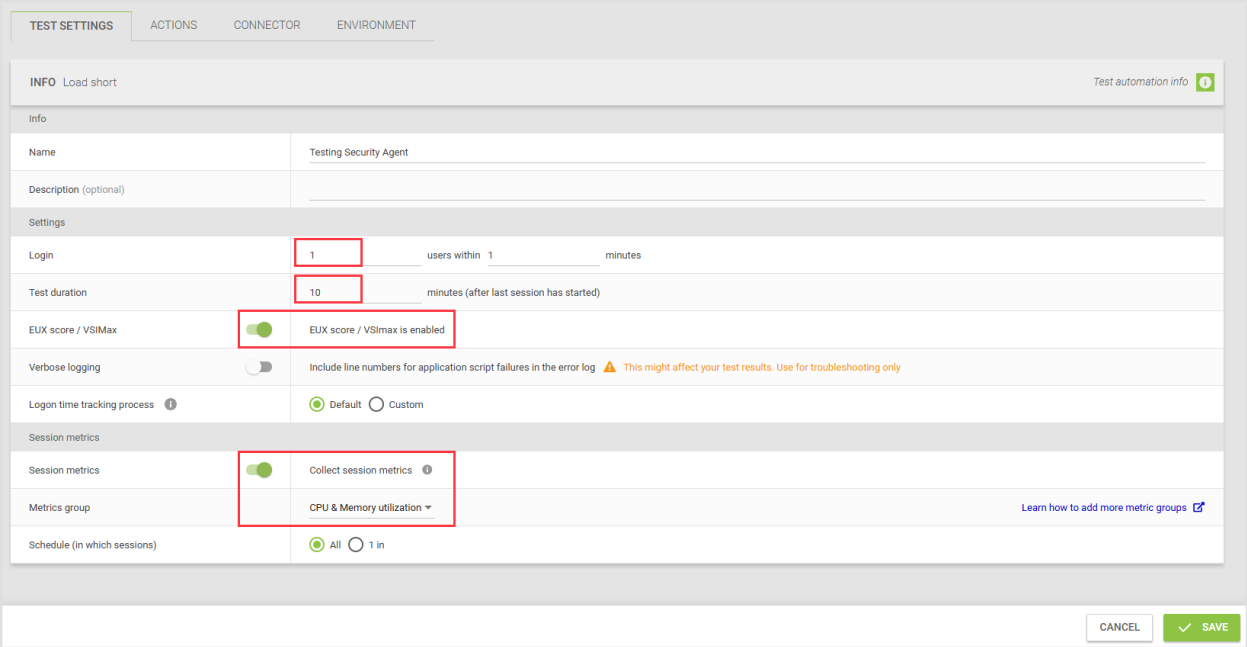
RDS/RDP Broker/Host	Enter the hostname of the endpoint under test.
Accounts	Select the newly created account group that contains the Test User for the endpoint.
Launcher	Choose the Launcher Group that can successfully RDP into the endpoint.



10. Click **Save** to apply the changes.
11. In **Load Tests**, click on the newly created Test and configure the following Test settings:

Field	Description
Login	Set to 1 user for the individual endpoint.
Test Duration	Set the duration to a realistic timeframe. Longer Tests (e.g., 1 hour or more) are more representative of an employee's workday and provide more accurate results. Shorter Tests yield quicker results but may miss issues only exposed in longer Tests. A 1-hour Test duration is recommended, but you can set it to the maximum workday duration (typically 8 hours) if needed.
EUX score	Enable this option to gather EUX metrics. This helps compare Test results and make informed decisions. For more information on the EUX score, see EUX Score and VSImax.

Session Metrics	Enable CPU and Memory utilization metrics. Custom metrics can be added if needed, but the default metrics group provides a quick start for testing. For more information on Session Metrics, see Session Metrics



12. Click **Save** to apply the test settings.

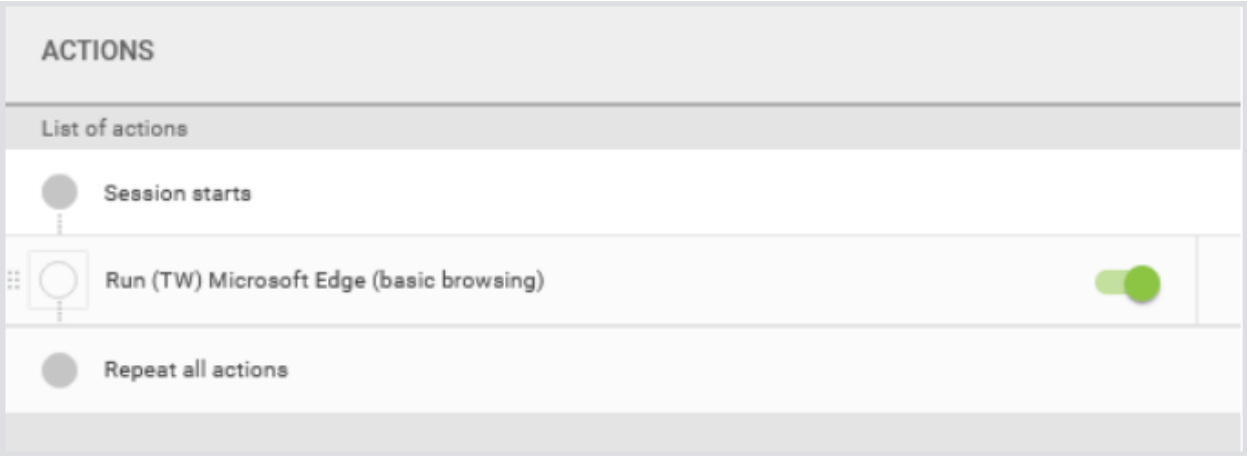
3. Configure the Applications (Test Scripts)

While Login Enterprise gathers login, EUX, and session metrics, it's also important to include Application testing (workloads) in your Test. These workloads should focus on the key applications essential for employee success, those that are crucial to business operations, high-risk, or frequently used in employee roles. Application testing ensures that these critical applications perform optimally, regardless of the underlying changes being tested (such as security agent updates, system updates, or application changes).

Example and Walkthrough

In the Load Test configuration, the **Actions** section defines the Application Workloads to be run. For this example, we'll set up a simple workload that involves looping Microsoft Edge: opening the browser, browsing the web, and closing it, with intermittent EUX measurements taken.

In this example, Microsoft Edge is added to the **Actions** section. For more information, see Adding Applications to a Load Test.



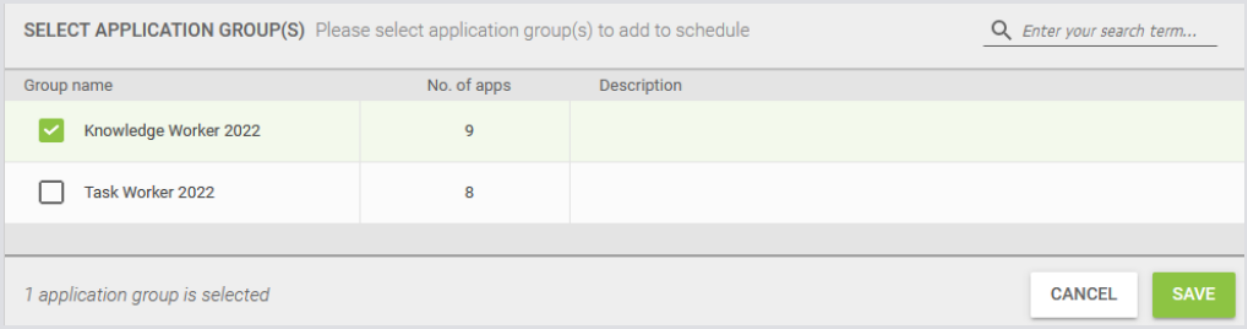
You can configure different Test scenarios here to run against the applications critical to your organization's success.

1. **Creating Application Workloads**. Workloads can be easily created using the no-code Login Enterprise Script Recorder (For more information, see Script Recorder), which allows you to record and upload user interactions.
    - These scripts can then be uploaded to Login Enterprise and used in the testing scenario to simulate real-world Application usage. For more information, see Importing Application scripts.
2. **Tailor Workload Sequences**. Adjust the sequences to fit different user roles and performance requirements. For example, simulate user behavior by including realistic actions like:
    - Opening Applications
    - Typing and navigating menus
    - Using appropriate wait times to simulate real-world use

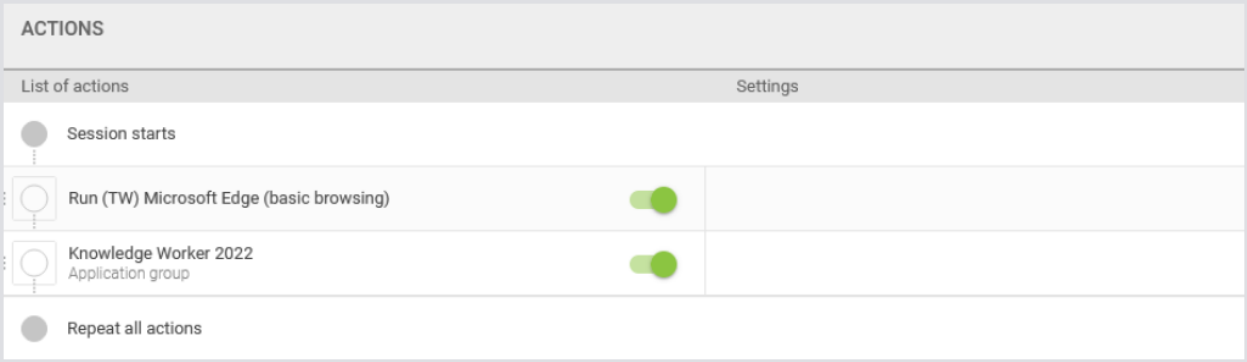
Use timers to measure specific actions, such as how long it takes to open a critical page in an app or to complete an important task. For more information, see StartTimer.

3. **Setting Up Out-of-the-Box Workload Groups**. Set up the **Knowledge Worker** or **Task Worker** workload group (for more information, see Knowledge Worker and Task Worker), which simulates typical office tasks. Note that Microsoft Office must be installed and licensed on the physical endpoint for this to work. To configure the workload:
    - Navigate to the **Actions** section of the Test configuration.
    - Click **Add action(s)** (the + icon) and select **Application Group(s)**.

- Select the preferred Application Group.



- Ensure the Application Group has been added to the Actions.



- Select Microsoft Edge, as it is typically pre-installed on Windows devices.

NOTE: While a Login Enterprise Test scenario doesn't require Application Workloads, it is highly recommended to include them for more comprehensive testing.

4. Establish Baseline Performance

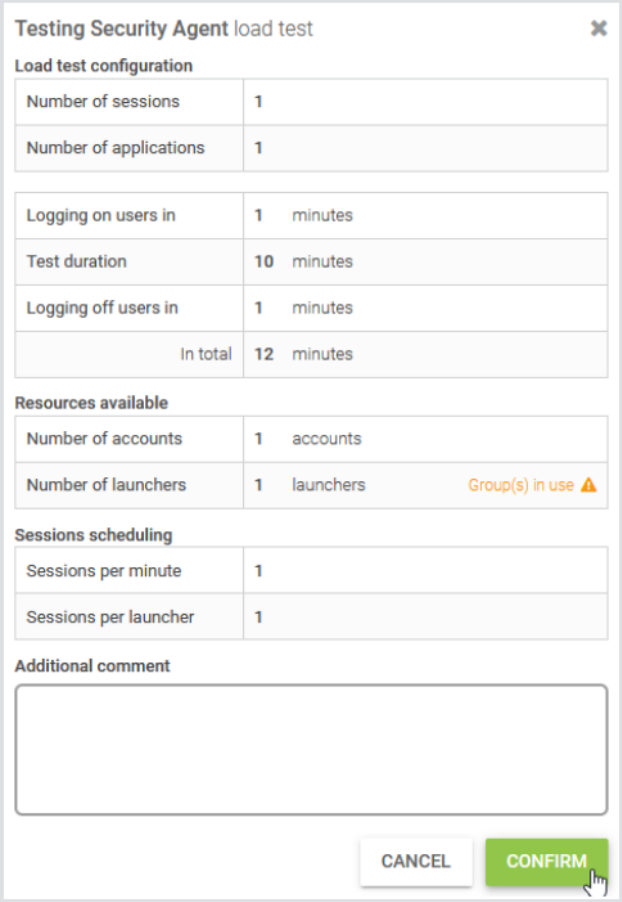
Before introducing any changes (such as security agent updates, system updates, or application changes), run Baseline Tests to capture performance metrics without the added changes.

1. **Execute Baseline Tests**: Run the configured Test on the target physical endpoint without any security agents or changes applied.
2. **Observe Virtual Appliance Results**: Review the performance metrics, including Application performance, session metrics, login times, and EUX scores.
3. **Validate Results**: Ensure there are no existing performance issues before introducing any changes.

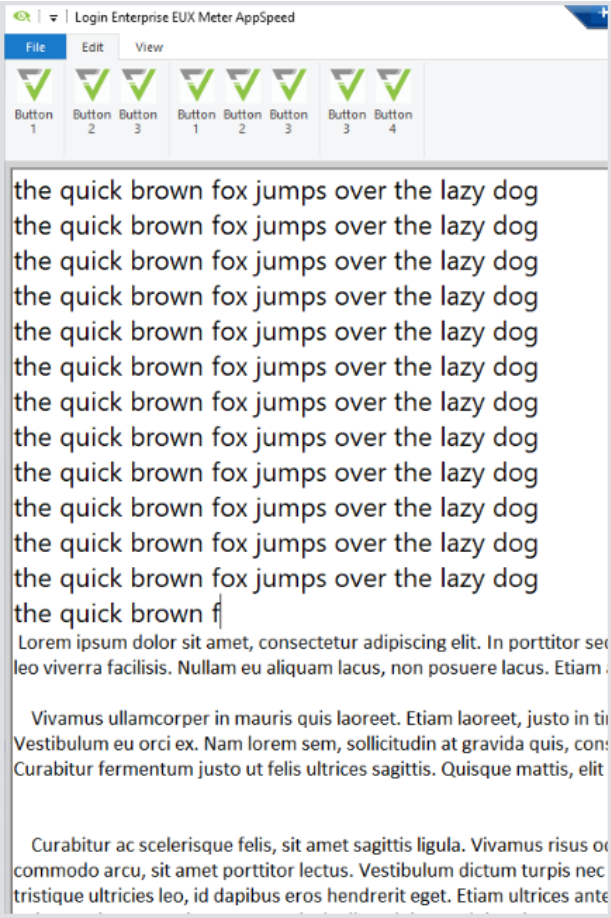
Example and Walkthrough

1. **Ensure Launcher Is Running**. Confirm that the Launcher application is running and the user has logged out of the endpoint under test.

2. **Start the Load Test**. For more information, see Starting a Load Test.
3. **Confirm Test Start**. In the confirmation dialog, click **Confirm** to initiate the Load Test. This will start the testing process.

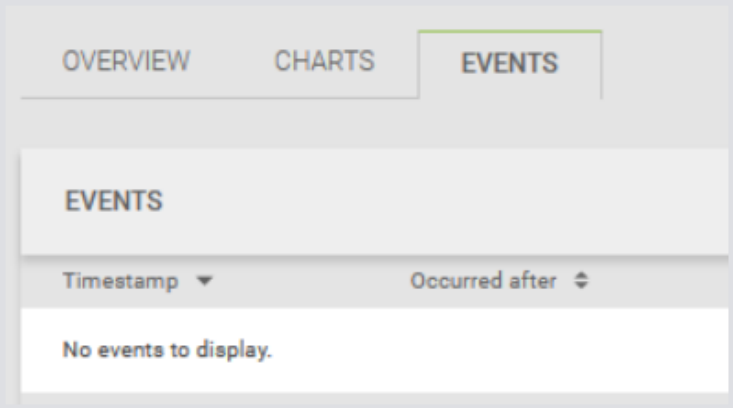


4. **Monitor Test Invocation**
   The launcher will invoke the Test session on the endpoint under test. The logon script should automatically trigger the defined testing process on the endpoint.

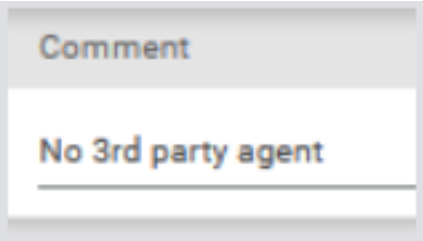


5. **Navigate to Dashboard**. After starting the Test, in the **Login Enterprise sidebar menu**, navigate to **Dashboard** to monitor the Test countdown, which will indicate when the Test is complete. For more information, see Dashboard.

6. **Review Preliminary Results**. Once the Test is complete, review the preliminary results to ensure there were no issues during the Test Run. For more information, see the Results page. Check **Events** for any errors or warnings.



7. **Add Comments to Test Results**. Add a comment to the Test results to indicate the mode of testing.



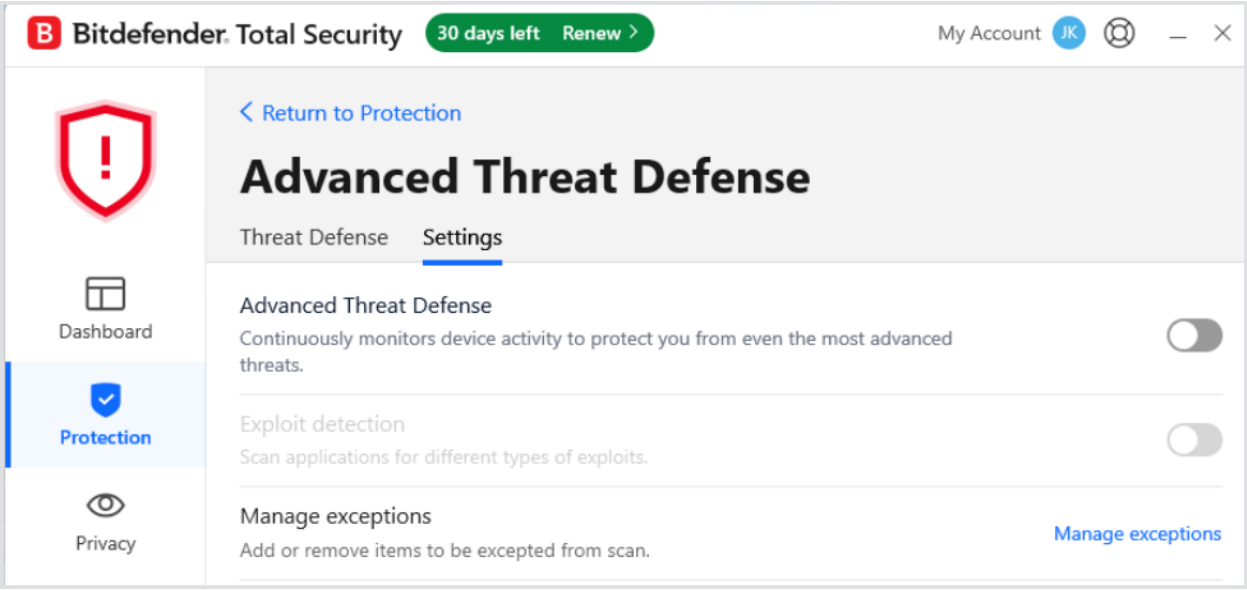
5. Implement Endpoint Changes

Introduce the endpoint changes to the Test endpoint. This could involve installing or updating any software, applying system configurations, or modifying settings to assess their impact on performance. Here are the steps (general approach):

1. **Install/Update Endpoint Software**: Follow the vendor's instructions for deploying or updating the software (e.g., security agents, applications, or updates) on the test endpoint.
2. **Apply Configurations**: Implement necessary configurations, such as enabling or disabling features, adjusting policies, or applying updates.
3. **Document Changes**: Keep a detailed record of all installations, updates, and configuration changes for future reference.

Example and Walkthrough

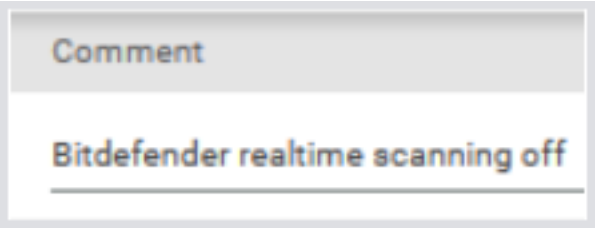
This could involve introducing a new application, updating a security agent, or applying a Windows update. In this example, BitDefender was installed with real-time scanning temporarily disabled:



6. Execute Tests with Endpoint Changes

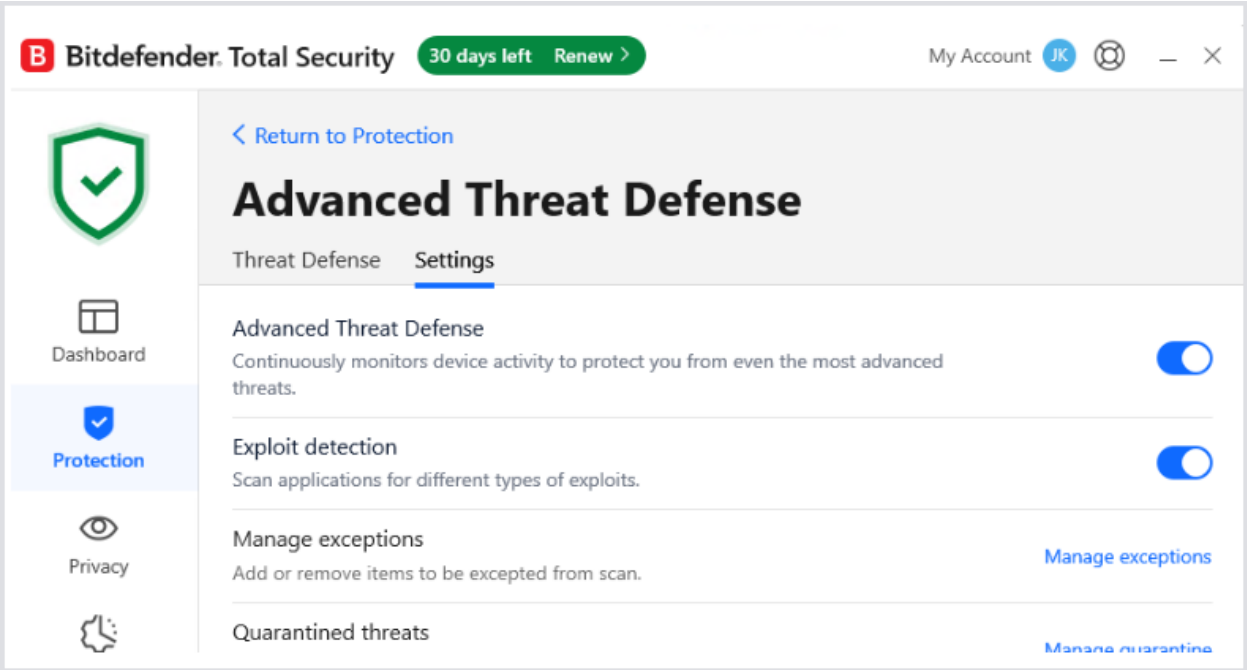
After applying the endpoint changes, execute the tests again to measure the impact of the changes on performance and system behavior. Here are the steps to follow (general approach):

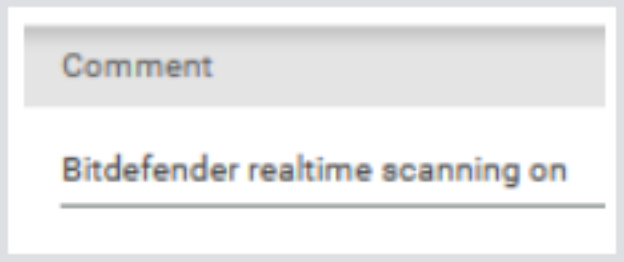
1. **Use the Same Test Configuration**: Ensure the configuration remains the same as in the Baseline Tests, with the only difference being the changes made to the endpoint.
2. **Verify Test Results**: After running the tests, review the results to ensure that the test was completed successfully and no issues were encountered.
3. **Add a Comment to the Test Run**: Provide a comment to indicate that the test includes endpoint changes (whether it's a software update, security agent change, or application modification).



7. Iterate Testing

Test any additional scenarios that need to be compared to previous results. For example, in this Test, BitDefender's real-time scanning was enabled to measure the impact compared to the previous Test.





8. Analyze Test Results

After running the Tests, compare the baseline data with the post-change results. For details, see Comparing Multiple Load Tests.

1. **Utilize Virtual Appliance Reporting**: Access detailed graphs and reports in the Virtual Appliance to review the Test outcomes.
2. **Focus on Key Metrics**: Identify any deviations in critical metrics such as:
    - CPU usage
    - Memory consumption
    - Application launch and performance times
    - Login times
    - End User Experience scores
3. **Assess Against Thresholds**: Compare the results with the predefined thresholds to determine if any metrics exceed acceptable levels.

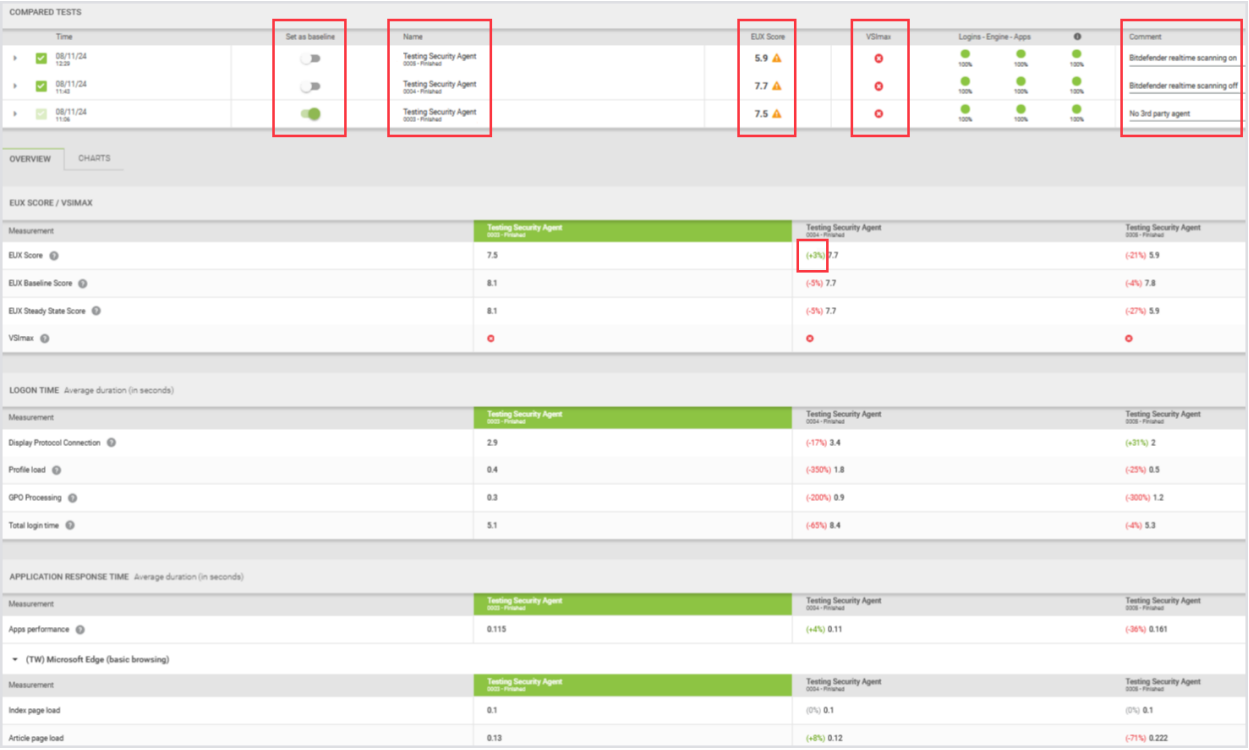
Example and Walkthrough

1. In this example, we compare 3 sets of Test results:

NOTE: Load Testing is preferred over Application Testing in this scenario, as Load Testing provides valuable metrics such as End User Experience and login times, which Application Testing does not.

2. The default comparison view in Login Enterprise is the **Overview** page. This page allows you to compare results from 2 to 5 Load Tests. It also lets you set a Baseline Test and view key metrics like EUX, login times, and Application response times in a structured format. The Overview page dynamically updates when the Baseline Test is changed, highlighting differences while maintaining consistent sorting for easy comparison. Pay attention to the highlighted items (from left to right) to identify significant deviations.

1. **Set as Baseline**: You can select the Test to use as the baseline. By default, the oldest Test will be selected as the baseline.
2. **Cross-Reference Test Results**: Under the **Name** column, you can cross-reference the Test number (shown under the Test name) with the Test Comment column. This helps identify which results correspond to which Test. For example, in this case, Test 0003 is the Baseline Test and corresponds to the "No 3rd party agent" Test (as indicated in the comment).
3. **Review EUX Scores**: The high-level EUX scores are prominently displayed in the Overview. Higher scores indicate better user experience and system performance.
4. **VSImax Value**: In this testing, the "VSImax" column will show red X icons, as the Tests were conducted with only one user per endpoint. This is expected, as "VSImax" represents the maximum number of concurrent users that can use the system while maintaining an acceptable EUX score. Since the Tests are single-user, this metric isn't applicable and will show as a red X by default.
5. **Compare Test Iterations**: In the Overview tables (e.g., EUX score / VSImax), the green column headers represent the Baseline Test. The columns to the right display the percentage deviation from the baseline. For example, a +3% deviation means the current Test performed 3% better than the baseline.
    1. **Positive (Green)**: Indicates better performance than the baseline.
    2. **Negative (Red)**: Indicates worse performance than the baseline.



3. **Open the Charts**
   The **Charts** page displays metrics for EUX, Logins, Applications, and Session Metrics. For details, see the Overview page and Charts and Graphs.

Example: If a login action takes 300% longer than the baseline, it would indicate a failed result based on the Test objectives defined earlier. This would signal that further configuration or changes to the security agent are needed, prompting further testing.

5. **Generate a PDF Report**: Optionally, you can generate a PDF report for easy sharing with stakeholders. For details, see Generating a Load Test PDF Report and Configuring a Load Test PDF Report. Save and share the generated PDF report as needed.

Conclusion

This conclusion is based on the results from the example testing provided earlier in this article (see the Define Your Test Objectives section) and is compared to the defined Test objective thresholds. This helps assess the impact of the changes made to the endpoint, whether they involve security agents, application updates, or other configuration adjustments.

1. **End User Experience Score**
   - **Threshold**: The End User Experience score can't be more than 1 point lower than the baseline.
   - **Result**: The baseline score (with no security agent) was compared to the score with the security agent and real-time scanning enabled. The difference was 1.6 points, which represents a **-21%** decrease.
   - **Outcome**: **FAIL**—since 1.6 is greater than the allowable 1-point difference.

Measurement	Testing Security Agent 0003 - Finished	Testing Security Agent 0004 - Finished	Testing Security Agent 0005 - Finished
EUX Score	7.5	(+3%) 7.7	(-21%) 5.9

2. **Logon Time**
   - **Threshold**: Logon time can't exceed 4 seconds longer than the baseline.
   - **Result**: The total logon time difference was **3.3 seconds** (8.4 seconds minus 5.1 seconds).
   - **Outcome**: **PASS**—since 3.3 seconds is less than the allowable 4-second difference.

LOGON TIME Average duration (in seconds)

Measurement	Testing Security Agent 0003 - Finished	Testing Security Agent 0004 - Finished	Testing Security Agent 0005 - Finished
Display Protocol Connection	2.9	(-17%) 3.4	(+31%) 2
Profile load	0.4	(-350%) 1.8	(-25%) 0.5
GPO Processing	0.3	(-200%) 0.9	(-300%) 1.2
Total login time	5.1	(-65%) 8.4	(-4%) 5.3

3. **Microsoft Edge Apps Performance**
   - **Threshold**: Microsoft Edge apps performance can't deviate by more than 33%.
   - **Result**: The total percentage difference in performance was **-36%**.
   - **Outcome**: **FAIL**—since -36% exceeds the allowable -33% threshold.

APPLICATION RESPONSE TIME Average duration (in seconds)			
Measurement	Testing Security Agent 0003 - Finished	Testing Security Agent 0004 - Finished	Testing Security Agent 0005 - Finished
Apps performance	0.115	(+4%) 0.11	(-36%) 0.161
(TW) Microsoft Edge (basic browsing)			

Since the defined performance thresholds were not met due to the changes introduced by the security agent, further discussions with the security team are necessary. Now, objective results are available to present and address the performance issues identified during testing.

Best Practices

1. **Test Across Different Devices**. Different hardware may respond differently:
    1. **Include Diverse Endpoints**: Conduct Tests on diverse endpoints, including different device types and ages, based on what's in production or to determine the best value endpoint hardware to purchase. Conduct a bake-off if needed.
    2. **Adjust Thresholds for Older Hardware**: Set realistic expectations for older hardware to avoid unrealistic comparisons.
    3. **Plan for Upgrades**: Identify older endpoints that may need replacement or upgrades.

2. **Test Before Each Security Agent Change**. Organizations that don't test changes (such as security updates, application modifications, or endpoint configurations) before deployment risk unintentionally breaking applications, causing them not to start, or worsening endpoint performance, which can ultimately lead to lost productivity.
3. **Iterate Results**. If issues are found during testing, take corrective actions:
    1. **Adjust Security Configurations**: Modify settings that are causing performance issues, such as adjusting scan frequencies or creating exclusions for certain processes.
    2. **Consult Vendors**: Reach out to the security software provider for optimization tips and best practices.
    3. **Rerun Tests**: Rerun Tests to evaluate whether the changes improved the situation and met the defined performance thresholds.

Investing in thorough endpoint testing and optimization helps improve operational efficiency and fortifies your organization's overall security posture. In a world where both security threats and performance expectations are high, taking these proactive steps is crucial.

System Requirements

- Overview
- Login Enterprise Virtual Appliance
  - Hypervisor
- Hardware requirements
  - External SQL database (optional)
  - Encryption technology
  - Certificates
- Login Enterprise Launcher machine
  - Operating systems
  - Software requirements
    - Antivirus and app blocker exclusions
  - Hardware requirements
- Digital Workspace
  - Operating systems
  - Software requirements
- Connectivity and permissions
  - HTTPS configuration (port 443)
  - Active Directory (AD) authentication
  - Launcher connectivity
  - Virtual Users

Overview

This section outlines the requirements for the successful implementation of the Login Enterprise Virtual Appliance. To learn how the different components of the Login Enterprise interact with each other, see the System Architecture and Component Overview.

For information on Login Enterprise licensing, see Managing Licenses. If you need a license file, request one here.

Login Enterprise Virtual Appliance

The Login Enterprise Virtual Appliance is delivered as a fully configured virtual machine. This virtual machine is provided as an OVA or VHD(X) file.

Hypervisor

A hypervisor, also known as a virtual machine monitor (VMM), is a software or hardware component that allows multiple operating systems to run on a single physical machine. It facilitates the creation and management of virtual machines, which are isolated instances of an operating system running on a host system.

For optimal performance, Login Enterprise is compatible with the following hypervisors:

Hypervisor	Versions	File
VMware vSphere Hypervisor (ESXI)	6.5 to 8.0	OVA file
Microsoft SCVMM (Hyper-V)	Server 2012 and greater	VHDX file
Citrix Hypervisor	8.2	OVA file
Microsoft Azure	-	VHD file
Nutanix AHV	AHV-20230302.102001 and greater	OVA file

The guest OS type in the OVA template is "debian9_64Guest," and the hardware level version is 13. It is to maintain compatibility with ESXi 6.5 and newer. OVA has VM tools installed, so once the VM is deployed from the OVA, the VM tool will automatically change the guest OS type to the highest version supported by the ESXi host.

Hardware requirements

For the Login Enterprise Appliance to perform its functions accurately, you need to configure the correct hardware requirements.

NOTE: To eliminate external performance influences, we recommend NOT configuring the Appliance on a heavily used hypervisor.

Configuration	CPU	Memory	Disk
Recommended	4(v)CPU	8GB RAM	100GB

With the recommended configuration, Login Enterprise supports a maximum of 2000 concurrent Load test users. For Continuous tests, Login Enterprise supports a maximum of 250 concurrent users, which also requires an External Database.

External SQL database (optional)

By default, Login Enterprise is delivered using an internal database. However, you can add an external SQL database.

NOTE: We recommend using the External Database option when you plan to run tests with more than 75 Concurrent Continuous Test users, or more than 1000 concurrent Load test users.

For information on the external database requirements and configurations, see External SQL database.

Encryption technology

Information is necessary to be provided to security teams regarding the type of encryption that Login Enterprise utilizes in our products. Requests for security details must be substantiated before you implement new software solutions in a production environment.

Encryption types

Login Enterprise utilizes the following types of encryption throughout the software.

Network communications with the following components occur over encrypted TCP / IP using SSL.

- Management Console
- Launchers
- Engine

Protocols and ciphers are used for encrypted communication over HTTPS.

- SSL Protocols
  - TLSv1.2
  - TLSv1.3
- Cipher Suite
  - EECDH+ECDSA+AESGCM
  - EECDH+aRSA+AESGCM
  - EECDH+ECDSA+SHA256
  - EECDH+aRSA+SHA256
  - EECDH+ECDSA+SHA384
  - EECDH+ECDSA+SHA256
  - EECDH+aRSA+SHA384
  - EDH+aRSA+AESGCM
  - EDH+aRSA+SHA256
  - EDH+aRSA:EECDH
- Disabled Ciphers
  - aNULL, eNULL, MEDIUM, LOW, 3DES, MD5, EXP, PSK, SRP, DSS, RC4, SEED

Passwords and Secure Custom Fields are stored utilizing AES-256-bit encryption and are FIPS compliant.

AES operational mode is the .NET core default mode CBC.

For more information, see the Microsoft documentation.

Certificates

By default, Login Enterprise is deployed using a self-signed certificate. The best practice is to employ your own certificate. To learn how to manage certificates for Login Enterprise, see Managing Certificates on the Appliance.

NOTE: Login Enterprise supports certificates that use RSA and ECC (Elliptic Curve/ECDSA) keys, including password-protected private keys for either key type.

Login Enterprise supports certificates with RSA or ECC encryption.

Login Enterprise makes use of the Debian certificate requirements. For each certificate to work correctly, there are a number of allowed Ciphers:

Allowed ciphers:

- EECDH+ECDSA+AESGCM
- EECDH+aRSA+AESGCM
- EECDH+ECDSA+SHA256
- EECDH+aRSA+SHA256
- EECDH+ECDSA+SHA384
- EECDH+ECDSA+SHA256
- EECDH+aRSA+SHA384
- EDH+aRSA+AESGCM
- EDH+aRSA+SHA256
- EDH+aRSA:EECDH

Disallowed ciphers:

- !aNULL
- !eNULL
- !MEDIUM
- !LOW
- !3DES
- !MD5
- !EXP
- !PSK
- !SRP
- !DSS
- !RC4
- !SEED

For more information, see the OpenSSL Documentation.

Properties

Login PI 3.0 and greater / Login Enterprise

Certificates

Deploying a certificate is a common practice for enhancing the security and trustworthiness of a product. By default, Login Enterprise is deployed using a self-signed certificate. A best practice is to employ your own certificate. To learn how to manage your certificates for Login Enterprise, see Managing Certificates.

Login Enterprise Launcher machine

The Login Enterprise Launcher machine initiates sessions to the target environment. It must be a separate virtual or physical machine.

A Launcher has a maximum capacity of 30 concurrent sessions. If you use Custom Connectors, the maximum capacity may vary. Additionally, the launcher needs to run in an interactive session to ensure that the appropriate client software can be executed, and graphic content from the protocol is displayed.

For more information on connectors and connection configurations, see Connectors.

Operating systems

To run the Launcher software, Login Enterprise is compatible with the following Launcher operating systems:

OS Vendor	Versions
Windows	Windows 10 and greater Windows Server 2016 and greater
IGEL	IGEL OS 12
Debian	Debian 10, Debian 12
Ubuntu	Ubuntu 20.04 LTS
Fedora	Fedora 33

Software requirements

To ensure the Launcher is working accurately, the Launcher needs to have certain software available. All Launchers need a vendor-specific connector to connect to the Test platform.

Windows

- Microsoft .NET Framework 4.8 or higher
- Citrix Receiver / Citrix Workspace

- VMware Horizon Client

Linux

- Curl
- XDOTool
- OpenSSL
- Free RDP
- Citrix Receiver / Citrix Workspace
- VMware Horizon Client

Antivirus and app blocker exclusions

To ensure optimal performance and prevent conflicts, certain files and processes need to be excluded from antivirus scans and app blocking. Add the following exclusions to your antivirus or app blocker to ensure the Login Enterprise runs correctly.

Launcher machine

- Enterprise Launcher software
  - C:\Program Files\Login VSI\Login Enterprise Launcher\LoginEnterprise.Launcher.UI.exe
- Universal Web Connector
  - C:\Program Files\Login VSI\Universal Web Connector\UniversalWebConnector.exe

Target VM

- Login Executable
  - LoginPI.Logon.exe
    - The path will depend on the policy/configuration of the environment
- Enterprise engine
  - %TEMP%\LoginPI\Engine\LoginPI.Engine.exe
- Login Enterprise EUX Applications
  - %TEMP%\LoginEnterprise\EuxApp
    - AppSpeed.exe
    - CompressionSpeed.exe
    - CpuSpeed.exe
    - DiskSpeed.exe
    - eux2023.zip
- Login Enterprise EUX work files
  - My Documents\eux2023
  - %LOCALAPPDATA%\eux2023

Hardware requirements

For optimal functionality of Launchers, you need to configure the minimum hardware requirements. The following specifications represent the minimum requirements applicable to machines running Windows, Linux, and Azure environments:

Configuration	CPU	Memory	Disk
Minimum	2(v)CPU	4GB RAM	1GB free space

- For information on Launchers overview and best practices, see Launchers Overview and Best Practices.
- To learn about the Login Enterprise Windows Launcher setup, configuration, maintenance, and more, see Login Enterprise Windows Launcher.

Digital Workspace

The Digital Workspace is a virtual or physical machine that you need to have if you want to perform a Test.

Operating systems

To run the target environment software, Login Enterprise is compatible with the following operating systems:

OS Vendor	Versions
Windows	Windows 10 and greater Windows Server 2016 and greater

Software requirements

To ensure the target environment is working accurately, you need to have certain software available on the platform.

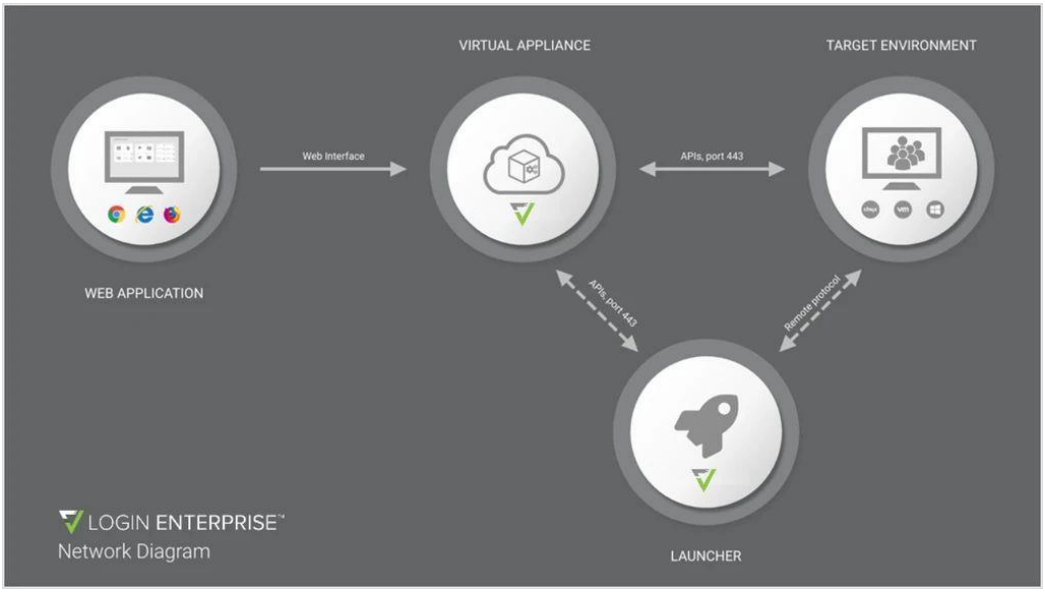
- Microsoft .NET Framework 4.8 or higher

Connectivity and permissions

For all the Login Enterprise components to work accurately, there are a few additional requirements. They are as follows:

HTTPS configuration (port 443)

The Login Enterprise Appliance must be reachable by the Login Enterprise Launcher and the Digital Workspace via HTTPS traffic, i.e. port 443. For more information, see the diagram and the table below:



Source	Target	Port Number	Protocol
Launcher Machines	Target Machines	3389	RDP
Launcher Machines	Citrix Servers	80/443	HTTP/HTTPS
Launcher Machines	Citrix Servers	1494	ICA Sessions
Launcher Machines	Horizon Connection Server	80/443	HTTP/HTTPS
Launcher Machines	Horizon VDI/Blast	8443	Blast
Launcher Machines	Horizon VDI/PCoIP	4172	PCoIP
Launcher Machines	Login Enterprise Virtual Appliance	443	Default TCP
Login Enterprise Virtual Appliance	Launcher Machines	443	Default TCP
Login Enterprise Virtual Appliance	Target Machines	443	Default TCP
Login Enterprise Virtual Appliance	AD Server (Optional)	135, 136, 137, 138, 139, 3268	TCP/UDP

Login Enterprise Virtual Appliance	AD Server (Optional)	445	TCP
Launcher Machines	AD Server (Optional)	135, 136, 137, 138, 139, 3268	TCP/UDP
Launcher Machines	AD Server (Optional)	445	TCP

For a list of external URLs that Login Enterprise may access, see Firewall Exemptions.

Active Directory (AD) authentication

To use your organization's Active Directory credentials (UPN) to authenticate and access the Appliance, you need to configure AD Authentication. To learn more, see Authentication.

Launcher connectivity

The Launcher must have HTTPS (port 443) access to the Login Enterprise Appliance. Additionally, it requires access to the connection broker of your vendor when connecting to a remote machine. That is, if you choose Citrix Virtual Apps and Desktops deployment, the Launcher must be able to reach the StoreFront server.

Virtual Users

The Virtual Users that you've configured for Login Enterprise must be able to:

- Log in to the Digital Workspace
- Run the logon script. For details, see Logon components.
- Connect to the Login Enterprise Appliance through the 443 (HTTPS) port
- Have full access to the users' %temp% directory
- Be able to initiate and have the required licenses for the applications to be tested through automation, including Office 365 licenses

For more information on the Virtual Users, see Accounts.

Firewall Exemptions

- Overview
- Allowlisted URLs
  - Diagnostics
  - Appliance Updates
  - Selenium Driver Updates (items in {} are dynamic values)
    - Chrome (pre version 114)
    - Chrome (post version 114)
    - Edge
    - Firefox
  - Debian
  - Docker
  - Azure
  - Internal Workloads (Task and Knowledge Worker)

Overview

This page serves as a comprehensive reference for the external websites and endpoints that the Login Enterprise appliance may access. It includes a list of all the specific URLs that the appliance could try to reach, which would require allowlisting on the customer's firewall.

Allowlisted URLs

Diagnostics

- https://lediagnostics.azurewebsites.net

Appliance Updates

- https://loginvsidata.s3-eu-west-1.amazonaws.com/LoginPI3_ISO/public/
- https://loginvsidata.s3-eu-west-1.amazonaws.com/LoginPI3_ISO/public/latest.txt
- https://loginvsidata.s3.eu-west-1.amazonaws.com/LoginPI3_ISO/public/
- https://loginvsidata.s3.eu-west-1.amazonaws.com/LoginPI3_ISO/public/latest.txt

Selenium Driver updates (items in {} are dynamic values)

Chrome (pre version 114)

- https://chromedriver.storage.googleapis.com/LATEST_RELEASE
- https://chromedriver.storage.googleapis.com/LATEST_RELEASE_{majorRelease}
- https://chromedriver.storage.googleapis.com/{version}/chromedriver_win32.zip

Chrome (post version 114)

- https://googlechromelabs.github.io/chrome-for-testing/last-known-good-versions.json

- https://googlechromelabs.github.io/chrome-for-testing/latest-versions-per-milestone-with-downloads.json
- https://edgedl.me.gvt1.com/edgedl/chrome/chrome-for-testing/{version}/win64/chromedriver-win64.zip

Edge

- https://msedgedriver.microsoft.com/LATEST_STABLE
- https://msedgedriver.microsoft.com/LATEST_RELEASE_{majorRelease}_WINDOWS
- https://msedgedriver.microsoft.com/{version}/edgedriver_win64.zip

Firefox

- https://api.github.com/repos/mozilla/geckodriver/releases/latest
- https://github.com/mozilla/geckodriver/releases/download/v{VersionValues}/geckodriver-{VersionValues}-win64.zip

Debian

- https://deb.debian.org/debian
- https://ftp.debian.org/debian/
- https://security.debian.org/debian-security

Docker

- https://download.docker.com
- https://registry-1.docker.io
- https://docker.com
- https://docker.io

Azure

- https://portal.azure.com/
- https://management.azure.com

Internal Workloads (Task and Knowledge Worker)

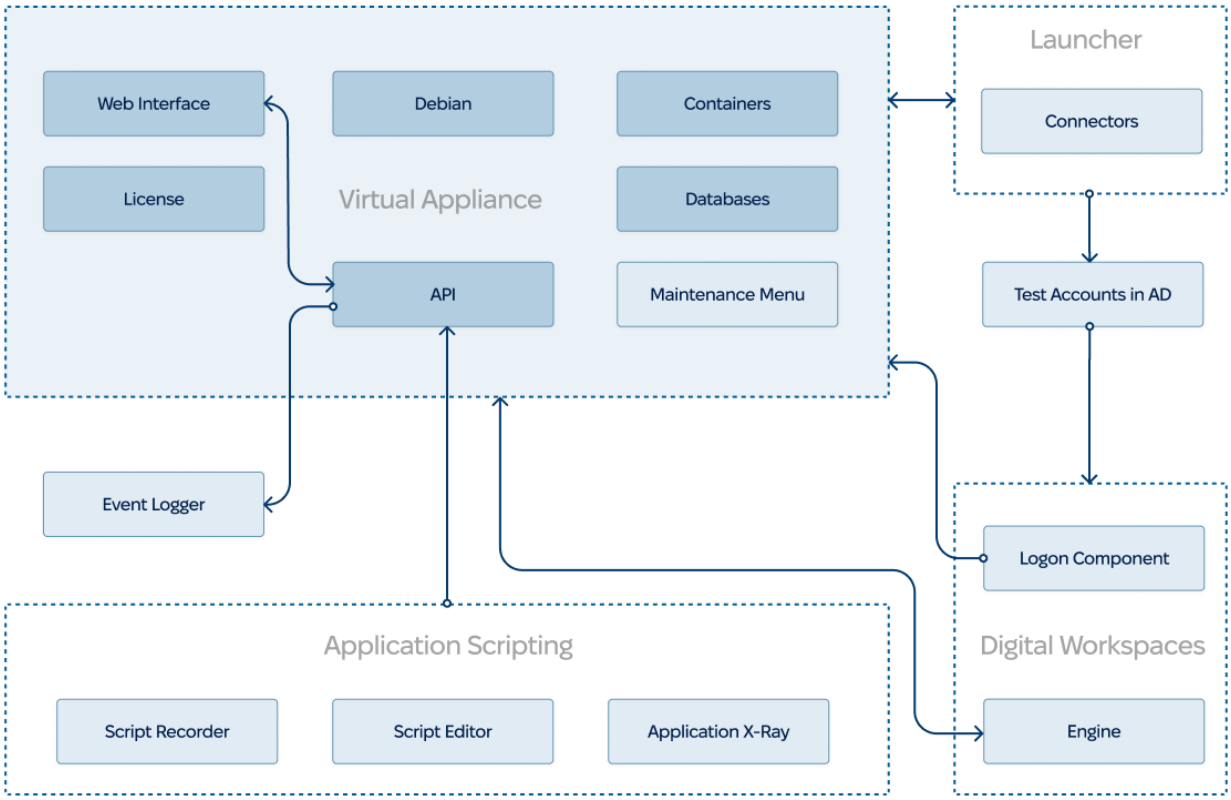
- https://euc.loginvsi.com/customer-portal/knowledge-worker-2023

System Architecture and Components

- Introduction
- Virtual Appliance
  - Web Interface
  - API
  - Debian
  - Containers
  - License
  - Databases
  - Maintenance Menu
- Launcher
  - Connectors
- Digital Workspaces
  - Engine
  - Logon Component
  - Test Accounts in Active Directory
- Application Scripting
  - Script Recorder
  - Script Editor
  - Application X-Ray
- Monitoring
  - Event Logger
- Encryption

Introduction

This section provides high-level summaries of the various components of the Login Enterprise platform, explaining their functions and how they work together within the system. It is designed to give you a clear understanding of each component's role and its integration, helping you effectively utilize Login Enterprise for testing, monitoring, and optimizing your environments.



Encryption

All internal communication for Login Enterprise occurs over port 443.

Virtual Appliance

The Login Enterprise Virtual Appliance is a fully configured virtual machine designed for deployment on hypervisors such as VMware ESXi, Microsoft Hyper-V, Citrix Hypervisor, and Microsoft Azure. It serves as the core of the Login Enterprise platform, hosting a Debian-based operating system, Docker containerization, and databases, as well as a web interface for configuring and managing test scenarios and visualizing results with graphs and charts. Delivered as an OVA or VHD(X) file, the appliance requires a minimum of 4 vCPUs, 8GB RAM, and 100GB disk space for optimal performance, supporting up to 2000 concurrent load test users or 250 continuous test users with an external database.

For more details, see Configuring the Virtual Appliance.

Web Interface

The Login Enterprise Virtual Appliance is a web-based interface that serves as the central hub for managing and configuring all aspects of the Login Enterprise platform. This intuitive web interface allows you to set up test configurations, upload application workloads, monitor tests in progress via the Dashboard, and review detailed performance results. It also allows you to manage and configure Launchers, Accounts, alerts/events, measurements, Environments, and

Providers. The web interface streamlines managing and optimizing your Login Enterprise setup, making it the command center for all your testing needs. It uses HTTPS for secure and encrypted communication, ensuring the safety of your data and configurations.

For more details, see Getting Started with the Login Enterprise Web Interface.

API

The Login Enterprise platform offers a comprehensive Public API built on REST Web API standards, enabling you to perform various tasks programmatically that are typically done through the Appliance web interface. These include automating configurations, running tests, retrieving data, and processing results. The API supports integration with external systems for advanced use cases such as bulk administration, analytics, and triggering IT workflows based on results or alerts.

To access the API, you can generate tokens with specific permission levels (Read-Only, Schedule, or Configuration) to control what operations can be executed. These tokens ensure secure and flexible API interactions.

Login Enterprise provides tools like Swagger (an API playground) for testing and experimenting with API requests and ReDoc for in-depth documentation of API endpoints. This robust API framework empowers you to streamline workflows, integrate with third-party tools, and achieve efficient data management within your IT environments.

For more details, see Using the Public API.

Debian

The Login Enterprise Virtual Appliance operates on a Debian-based Linux environment, providing a stable and secure foundation for managing performance testing and application compatibility. This Debian OS supports appliance updates, security enhancements, and optimizations, ensuring reliable operation and compatibility with the Login Enterprise platform's features.

For more details, see Updating Debian on the Virtual Appliance.

Containers

Login Enterprise employs containerization technologies such as Docker and Portainer to manage core platform functionalities efficiently. Each container serves a specific role, analogous to Windows services, ensuring modularity and stability within the Login Enterprise Virtual Appliance. These containers power essential operations, streamlining performance, scalability, and maintenance.

For troubleshooting, the **Container Status** page within the Login Enterprise interface offers insight into the health of each container. Accessible through the **System** menu, this page

provides a detailed status for all containers. Healthy containers are marked accordingly, but deviations can be investigated by downloading log files directly from the platform. These logs can be reviewed using tools like SEQ or sent to support for further assistance.

This functionality supports operational transparency, helping maintain Login Enterprise's reliable performance and ease of management.

For more details, see Monitoring Container Status.

License

Login Enterprise requires a valid license to ensure the platform functions fully and supports Virtual User sessions, testing, and workload management. During the initial setup, a 3-day grace license is provided, but it must be replaced with a valid license to continue testing without interruption. Notifications in the platform interface inform users when licenses are nearing expiration or have expired. Licenses can be updated at any time through the Licensing menu in the Appliance web interface to ensure uninterrupted platform operation.

For more details, see Managing Licenses.

Databases

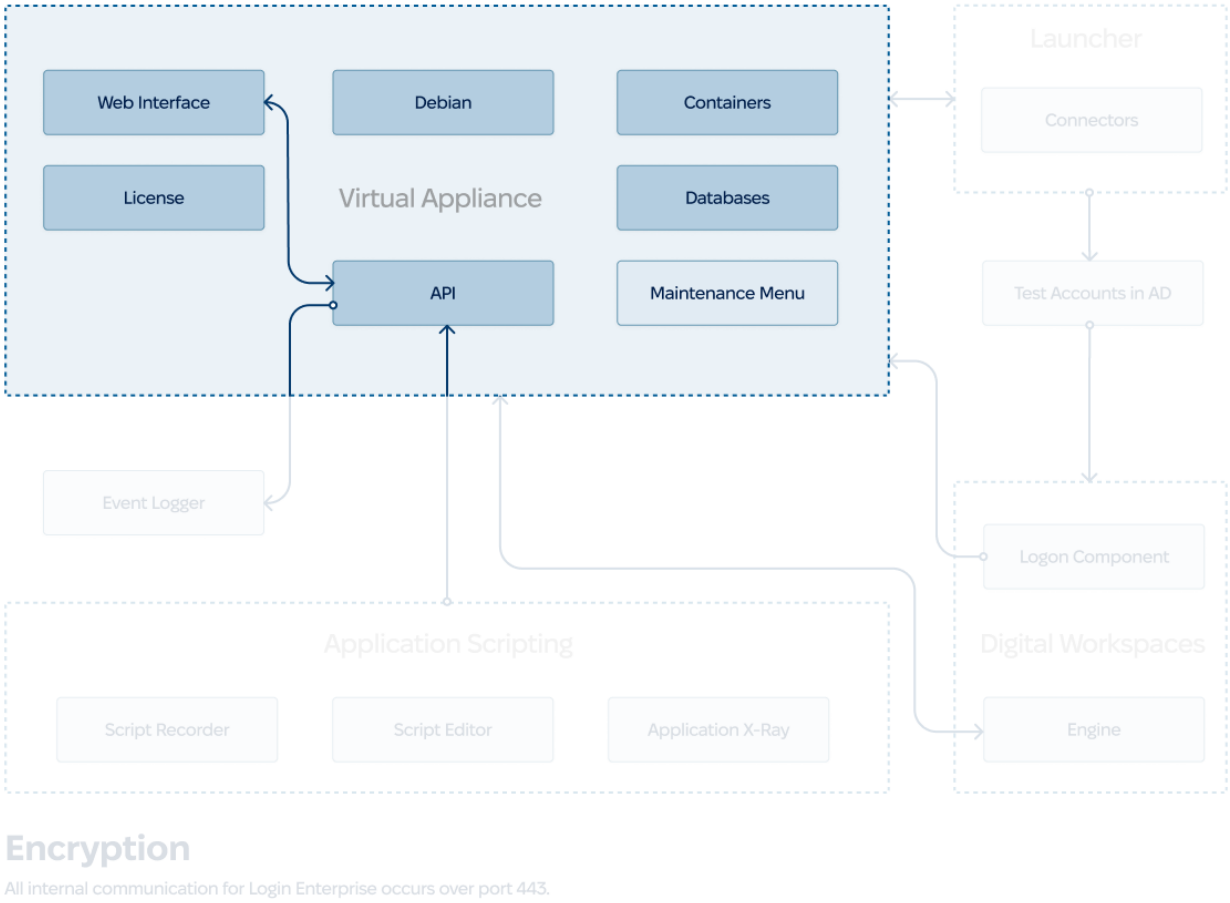
Login Enterprise provides flexibility in database configuration, allowing you to choose between the default internal PostgreSQL database or an external Microsoft SQL Server database, depending on what best suits your needs. While the internal database is included by default and simplifies setup and management, opting for an external SQL database can offer benefits such as centralized administration, enhanced scalability, and improved compliance with organizational policies. The choice ultimately depends on what is most advantageous for your specific requirements. For example, the internal database may provide ease of use, while an external SQL database allows for features like high availability and advanced security. Additionally, Login Enterprise employs a time-series database, powered by InfluxDB, to store and process platform and session metrics.

For more details, see Configuring Database Data Retention and Configuring External SQL Database.

Maintenance Menu

The Login Enterprise Maintenance Menu, accessible over SSH, provides a user-friendly interface for managing, updating, and troubleshooting the appliance. This menu allows you to configure system settings, manage network interfaces, perform updates, and handle certificates, all within a secure environment. Key features include restarting Login Enterprise services, performing database maintenance, enabling or disabling the SSH server, and accessing a bash shell for advanced troubleshooting. It simplifies appliance management and supports smooth operation with tools like Ping and Netstat for network diagnostics.

For more details, see [Using the Maintenance Menu](#).



Launcher

The Login Enterprise Launcher acts as an endpoint that connects the Login Enterprise virtual appliance to target environments like VMware, Citrix, and others. Available for Windows, Linux, and IGEL, the Launcher facilitates up to 30 concurrent Test sessions, executing predefined scenarios to measure performance and ensure targets function as expected. Organizations can monitor real-world latency and performance by strategically positioning Launchers in specific locations, aiding in Continuous, Load, and Application Testing. Launchers leverage API keys for secure credential handling, operate autonomously during tests, and offer flexibility for deployment and scaling across various environments.

For more details, see [Launchers Overview and Best Practices](#).

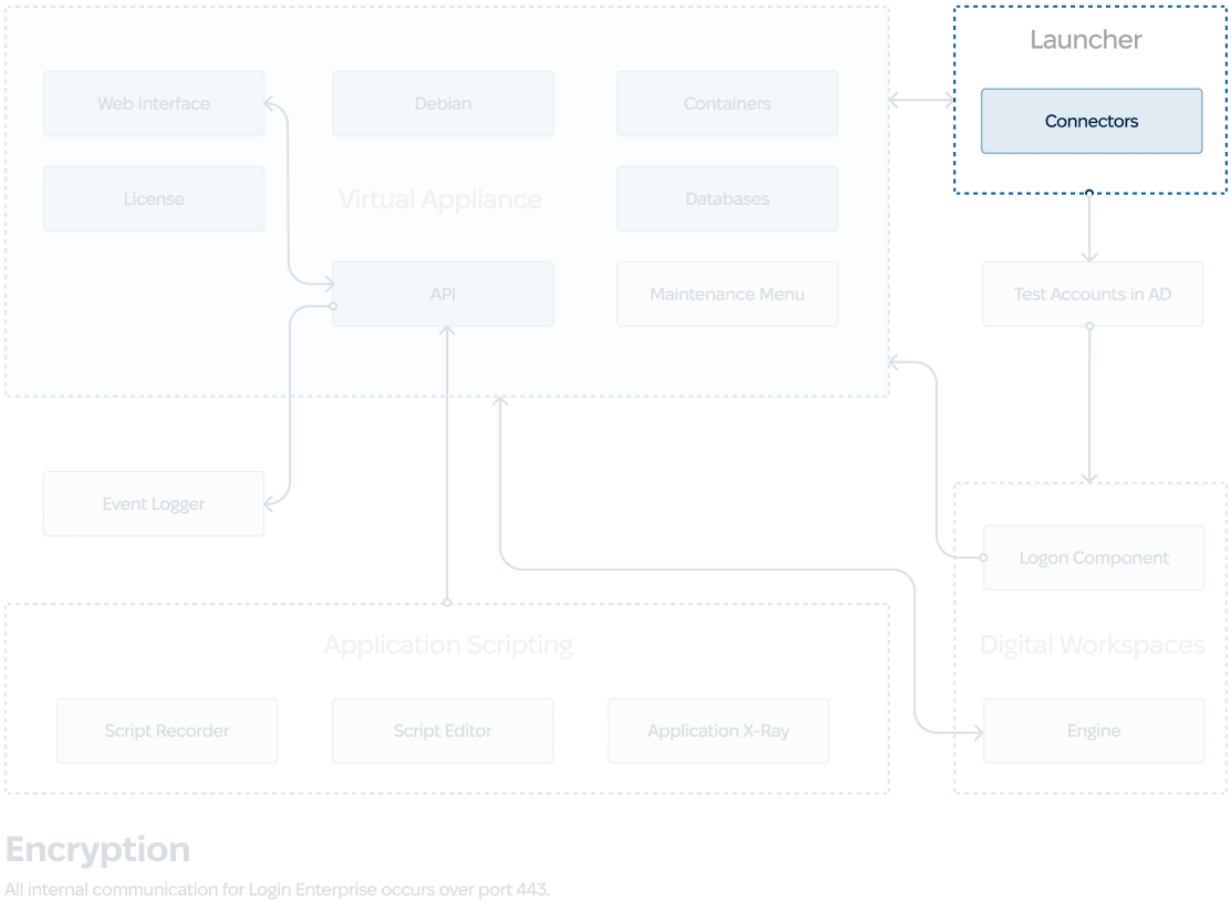
Connectors

Login Enterprise simplifies establishing connections to testing targets using a variety of pre-configured and customizable connectors, including Citrix, VMware Horizon, Microsoft RDS, and the Universal Web Connector (UWC). These connectors provide the necessary configuration

details to connect to target environments, enabling effective testing and automation of sessions to published applications and desktops.

The UWC enhances flexibility by supporting customized scripts for gateways or unique configurations, such as time-based authentication or advanced connection requirements. It automates login processes, launches applications, and retrieves session files, making it ideal for environments not supported by standard connectors. With both pre-configured and custom options, Login Enterprise offers reliable compatibility and adaptable connection setups for your testing scenarios.

For more details, see Connectors.



Digital Workspaces

In Login Enterprise, a digital workspace represents a target environment or a desktop where Virtual User workloads are executed and measured, serving as the foundation for testing and validation. The software is designed to be agnostic, offering unparalleled flexibility to test and monitor virtually any Windows-based infrastructure. Whether the target is a physical endpoint, such as a laptop or desktop, or a virtual resource like a VDI session, published application, or server-hosted desktop, Login Enterprise ensures seamless integration and performance testing.

Using remoting protocols, the Launcher connects to these environments via test accounts, simulating realistic user interactions to provide detailed performance metrics. Supported environments span across cloud, on-premises, and hybrid configurations, including Microsoft Azure, VMware Horizon, Citrix Cloud, Amazon EC2, Microsoft RDS, and beyond. For virtual environments, the solution connects to published virtual resources through brokers and hosts, navigating resource pools to test virtual desktops or applications. For physical endpoints, the Launcher interacts directly, enabling comprehensive insights into end-user experience on actual hardware.

This flexibility empowers organizations to perform bake-offs, validate infrastructure decisions, and monitor performance across diverse setups. By supporting a wide array of platforms and deployment models, whether cloud-native, hybrid, or traditional on-premises, Login Enterprise adapts to the unique requirements of each organization. The result is a highly versatile tool that meets the demands of modern IT environments while simplifying performance validation and monitoring workflows.

Engine

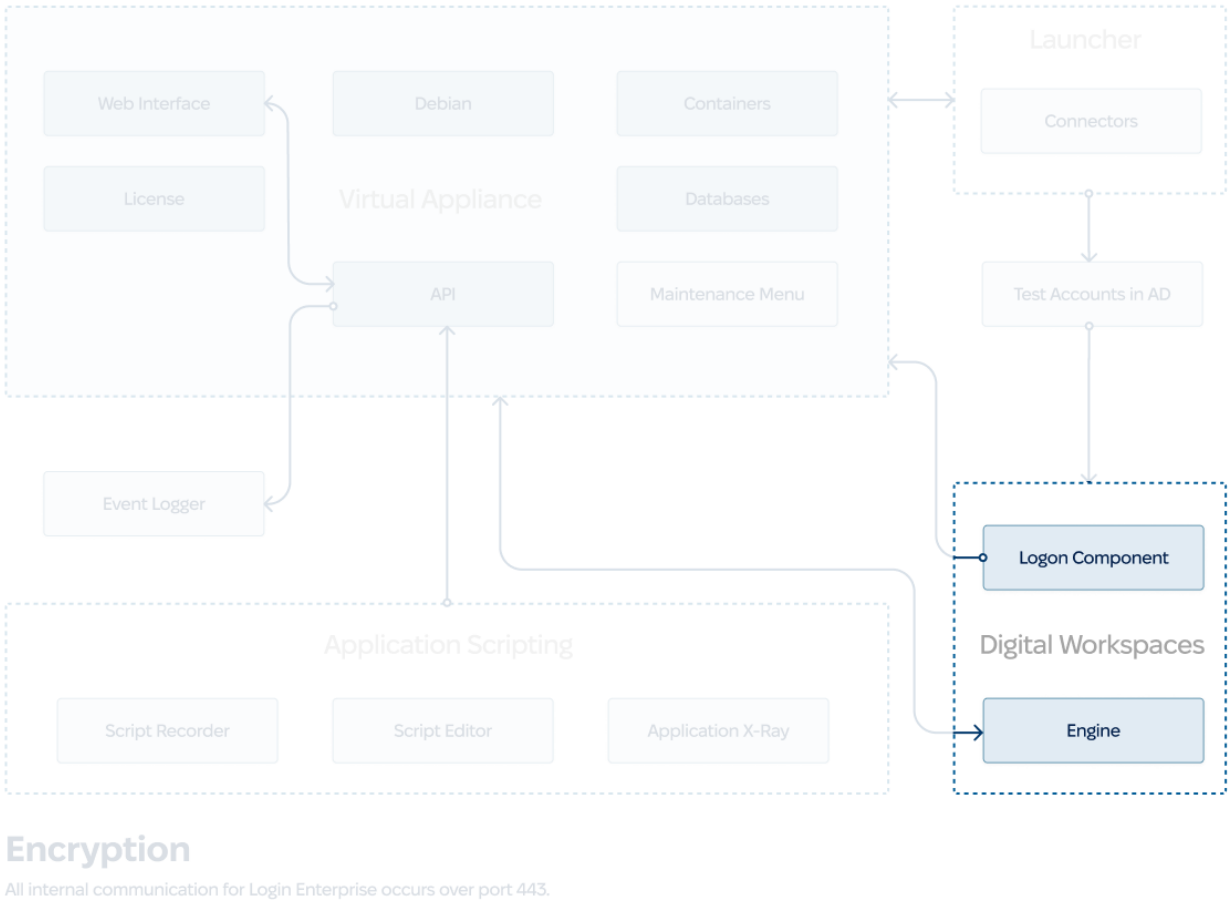
The Login Enterprise Engine is a key element in running test sessions initiated by the Logon Component. When a session is designated to execute a test, the Logon Component deploys the Login Enterprise Engine to the target digital workspace desktop environment. The engine drives Virtual User interactions as specified by workloads, interacting with applications and collecting detailed performance measurements. These metrics are communicated back to the Login Enterprise Virtual Appliance, enabling precise evaluation and actionable insights into system and application performance.

For details, see Engine.

Logon Component

The Login Enterprise Logon Component is a feature designed to initiate Virtual User sessions and synthetic operations by connecting to the Login Enterprise Virtual Appliance. It ensures that the platform communicates with the appliance to determine test parameters and activates the engine that executes tests. The Logon Component can be deployed as either an executable or a fallback PowerShell script, depending on the environment's compatibility. While the Logon Executable is the recommended option for most modern systems, the script serves legacy environments that lack support for the executable. Both methods provide flexibility in configuration, allowing integration through AD user profiles, GPO filtering, or UEM tools to suit organizational needs.

For more details, see Configuring Logon Components.



Test Accounts in Active Directory

Login Enterprise Test Accounts are **Active Directory (AD) user accounts** that must already exist in your AD infrastructure before being configured in the Login Enterprise platform. These accounts enable testing across desktops, profiles, and both virtual and physical environments by mirroring the structure of your AD. Login Enterprise does not create these accounts in AD; instead, it relies on their prior existence to perform Continuous, Load, and Application tests. By simulating real-world user interactions, these accounts allow the platform to measure performance accurately. Test Accounts can be configured in Login Enterprise via bulk creation, support customizable fields for connecting to target platforms, and integrate with custom connectors, streamlining the testing process

For more details, see Managing Virtual User Accounts and Configuring Authentication.

Application Scripting

Script Recorder

The Login Enterprise Script Recorder enables zero-coding script creation by capturing and recording user inputs, such as mouse clicks and keyboard interactions, during the recording process. Integrated with the Script Editor, it simplifies the creation of scripts, workloads, and workflows for application, load, and continuous testing by identifying UI elements and automating wait logic. This tool reduces script creation time from hours to minutes, allowing users to validate and adjust workflows effortlessly without requiring coding expertise. It supports both in-house and third-party applications, streamlining testing and ensuring a seamless end-user experience.

For more details, see Using the Script Recorder.

Script Editor

The Login Enterprise Script Editor simplifies creating and managing application workflows for Application, Load, and Continuous Tests. It includes functions for designing, debugging, and optimizing workflows, such as syntax error detection, command suggestions, and detailed debugging options. Supporting both executable and browser-based applications, it also integrates `// TARGET:` comments for CLI automation and includes the Script Recorder for capturing user interactions.

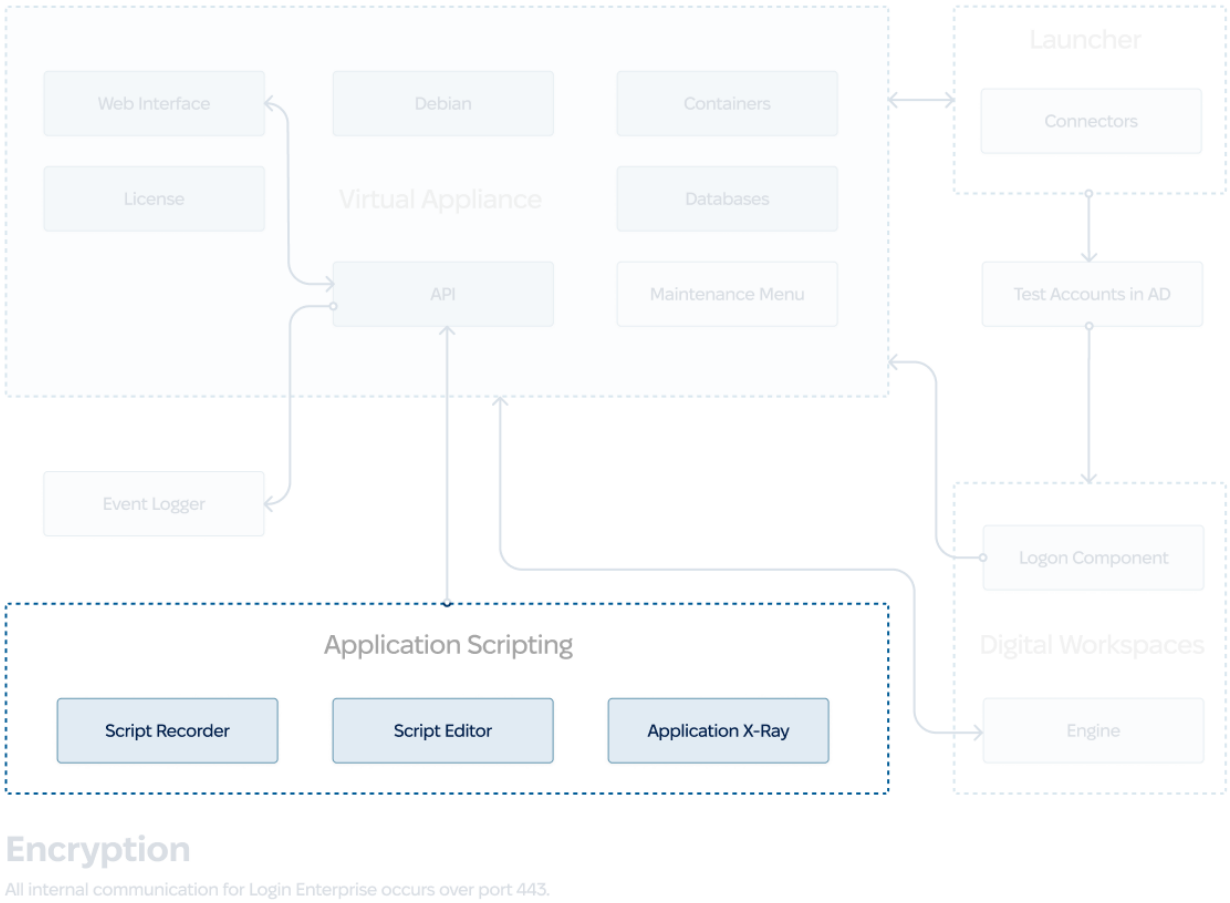
With an extensive library of scripting functions, such as commands for managing files, browsers, and Windows elements, the Script Editor allows precise testing and workflow automation. Functions like `If Else`, `Try Catch`, and `ForEach` enable dynamic and adaptable scripting, making it easier to validate workflows and automate workloads tailored to specific needs.

For more details, see Using the Script Editor.

Application X-Ray

The Login Enterprise Application X-Ray tool simplifies the creation of application workloads by providing a straightforward interface for capturing and scripting user interactions with desktop or web-based applications. Included in the Login Enterprise platform, X-Ray allows you to identify and interact with UI elements, simulate user actions, and generate ready-to-use C# code snippets for workload automation. Supporting Win32 applications and web browsers, X-Ray can bind to applications, detect controls, and execute actions like typing or clicking, making it easy to create and test reliable workflows. This tool enables you to efficiently build and refine synthetic workloads for testing application performance and compatibility across a wide range of environments.

For more details, see Using Application X-Ray.



Monitoring

Event Logger

The Login Enterprise Event Logger integrates with the Windows Event Log Service, providing real-time logging of events from the Login Enterprise Appliance to a Windows machine's Event Viewer. This seamless integration enables organizations to monitor system activity effectively while leveraging tools like Microsoft SCOM for advanced analysis. It tracks functional issues, such as login failures or application errors, and technical problems such as database connection errors, assigning specific event numbers for precise tracking. The Event Logger simplifies system monitoring and helps maintain optimal performance by enhancing visibility and delivering timely insights.

For more details, see Configuring the Event Logger.

Encryption

Login Enterprise utilizes advanced encryption technology to ensure secure communications and data protection. Network interactions between components, such as the Appliance Web Interface, Launchers, and Engine, are encrypted using SSL with protocols TLSv1.2 and TLSv1.3. The platform supports a range of secure ciphers while excluding weaker ones, such as MD5 and RC4,

to maintain high-security standards. Passwords are encrypted with AES-256-bit in CBC mode, adhering to FIPS compliance. Additionally, Debian certificate requirements ensure the proper use of trusted ciphers for authentication and communication. This encryption framework provides a reliable foundation for protecting sensitive data and maintaining the integrity of Login Enterprise environments.

For details, see Encryption.

Glossary

This glossary comprehensively overviews the Login Enterprise Virtual Appliance's essential concepts, terms, acronyms, and features. It also includes industry-standard terms. Use it as a reference to learn about the Login Enterprise components and how they work together.

A - **B** - **C** - **D** - **E** - **F** - **G** - **H** - **I** - J - K - **L** - **M** - N - **O** - **P** - Q - **R** - **S** - **T** - **U** - **V** - **W** - X - Y - Z

A

Access Control

Measures and technologies that regulate who or what can access resources within a computing environment. It includes authentication (verifying identity), authorization (determining permissions), and auditing (monitoring access). Access control ensures that only authorized users, applications, or devices can access specific resources, enhancing security and protecting against unauthorized access. To learn more, see Configuring Active Directory Authentication.

Access Token

A credential that represents the authorization granted to a user or application to access specific resources. Access tokens are commonly used in OAuth 2.0 authentication flows, where they are issued by an authorization server and presented to a resource server to gain access to protected resources. To learn more, see Adding an access token.

Account

A username/password pair used to trigger connections and identify connections once they start (usually an AD account). To learn more, see Accounts.

Account Capacity

The full set of Test accounts available to a Test or a set of Tests.

Account Group

A subset of the full list of accounts available to the system that can be selected by name pattern matching or explicit selection. To learn more, see Creating Account Groups.

Active Session

How many users are actively logged on and probably testing during a Test scenario running.

AD Timeout

The time it takes for the AD domain controller to respond. To learn more, see Configuring Active Directory.

Appliance

The Linux-based Login Enterprise control center.

Appliance Admin user

The original superuser account created during Appliance setup. It exists outside of LDAP and the Access Control UI. It has unrestricted access to all Tests and system areas by default. To learn more, see Role-Based Access Control.

Application (App)

A configured step that can be associated with a Test, consisting of a C# script that performs user-like actions in a Test session, and other configuration information. It is not limited to a single end-user application, it could start and interact with multiple programs. Normally, though, it will test one program. To learn more, see Applications.

Application Script

The script portion of the Application (App) above.

Application Registration ID

App registration represents Login Enterprise as a registered application in Azure. Find the app registration ID on the App registration tab in the Azure portal.

Application Test

Application Tests are a specific type of Login Enterprise Test configuration, used for checking that your user software still operates, and operates within your existing operational parameters. Like Load Tests, these are one-time, on-demand Tests that return data and events specific to this Test. They are not scheduled Tests that run periodically. To learn more, see Application Testing.

Application Secret

Application secret is a parameter that enables Login Enterprise to authenticate against the app registration in Azure. Create it in the Provider definition to allow access to Azure billing information.

B

Baseline EUX Score

The Baseline EUX Score represents the best possible performance of the system and is the average EUX Score of the best 5 minutes of the Test. This score indicates how the system performs when it is not under stress. Typically, the Baseline Score is captured at the beginning of the Test before the system is fully loaded. To learn more, see Baseline and Steady state EUX scores.

Built-In Admin role

A system-defined role introduced in v6.1. It always has full access to all Tests and Test Runs, even though it does not appear in the Access Control tab. The role cannot be deleted, renamed, or modified, and only user or group membership can be changed. See Built-In Admin for more details. To learn more, see Built-In Admin.

C

CapEx

CAPEX (Capital Expenditure) refers to money spent on acquiring, repairing, updating, or improving fixed company assets, such as on-premises IT infrastructure. To learn more, see the Operations dashboard.

Connection Broker

A VDI software for directing an incoming connection for a user to a specific VDI resource.

Connector

A Login Enterprise software that is used to interact with a Connection Broker to launch a remote connection to a target desktop. To learn more, Connectors.

Container Status

Login Enterprise is comprised of a number of Docker Containers. The Container Status page allows you to see the status of each container in the system.

Continuous Test

A Continuous Test is a specific type of Login Enterprise Test configuration. Also referred to as Continuous Monitoring, these are Tests that run back to back, intended to keep a running monitor of your end-user experience, alerting you quickly to problems in your infrastructure without waiting for users to begin to complain. To learn more, see Continuous Testing.

D

Data Retention

The time that Login Enterprise will retain your testing results and log entries. You can set this as low as you need to preserve data for a time but not allow unlimited database expansion. You can choose separate data retention periods for Raw Measurement Data, Event Data, or EUX Raw Data. To learn more, see Data retention.

Domain Controller

The LDAP or Active Directory server that Login Enterprise can authenticate against.

E

Engine

The tool on target machines that retrieves and executes Application scripts, and reports results back to the Appliance. To learn more, see Engine.

Environment

The environment enables Test grouping. Use it to define cost and capacity for the Operations dashboard. To learn more, see Environments.

Event

A specific message with a time stamp that can be generated automatically by the Appliance or from an Application script. Events can be viewed in the UI or retrieved from the API, showing significant events with timestamps in your Test history. Events can optionally trigger e-mail alerts. To learn more, see Events.

Event Logger

A tool for transporting Test events from the Appliance into a Windows Event Log so they can be further processed by third-party tools like Splunk, Datadog, and Solar Winds. To learn more, see Event logger.

Executable

A program that can be run in Windows. To learn more, see Logon executable.

EUX Metrics

The component scores from performance data that will be combined into the final EUX Score.

EUX Raw Data

EUX raw data exceeding the number of defined days will be deleted. To retain all EUX data, toggle off this feature. EUX data is collected during Tests with EUX metrics enabled. The Login

Enterprise EUX Score offers an objective means to assess how the overall experience evolves with an increasing number of users logging into a specific infrastructure or as changes are implemented that may affect the entire system. To learn more, see the EUX raw data, e.g. for Continuous Tests (this feature also works with Load Testing).

EUX Score

End User Experience is a specific set of tools that produce measurements to closely approximate how a human would feel about using this machine. A score from 0 to 10. To learn more, see EUX score and VSImax.

External SQL Database

A Microsoft SQL Server database can be used to store Login Enterprise data instead of the internal database in the Appliance. Choose this for production use. Set this during initial configuration, because you cannot migrate an existing configuration to the External SQL Database. To learn more, see External SQL database.

F

FQDN

Fully-qualified domain name: a hostname plus its domain, e.g. server.subdomain.domain.com.

G

GPO Processing

The time it takes for Windows to transmit and enforce any Group Policy Objects (GPOs) that apply to a Windows machine. This happens as part of the Windows logon phase and can affect the total amount of time it takes to log in.

Grace License

Login Enterprise comes built-in with an automatic license that is valid for 3 days from the time of installation. You must replace this with a real license to continue operating the software. To learn more, see License.

H

Host URL

Any host's exposed URL. For example, the Login Enterprise Virtual Appliance URL, e.g. https://server.subdomain.domain.com/.

I

Integration Test

A software testing approach that assesses the interactions and interoperability among individual software components or modules within a system. It verifies that integrated components function together as intended and exchange data accurately. Integration Tests aim to identify defects related to interface communication, data flow, and integration points between software modules, ensuring cohesive system operation. These Tests can occur at different levels, such as component integration, subsystem integration, or end-to-end integration, depending on system complexity. Integration testing is typically conducted after unit testing and before system testing in the software development lifecycle to detect and resolve integration issues early, ensuring overall software quality and reliability.

Internal API

If you perform actions in the Login Enterprise web interface, the interactions are occurring within the internal API.

L

Launcher

A system that uses a Connector to initiate remote connections to target machines. To learn more, see Launchers Overview and Best Practices.

Launcher Group

A subset of the full list of Launchers available to the system that can be selected by name pattern matching or explicit selection. To learn more, see Launcher Groups.

Launcher Hostname

The Windows name reported by the Launcher.

Legacy Scripting Toolset

A set of tools that can be used with VSCode, to approximate the functionality now available with the Script Editor. To learn more, see the Script Editor.

Legal Notice Tool

Legal notices at session login are designed to be manually clicked, and can't be skipped. They exist in space after the Launcher starts a session and before the Login Processor and Engine take over and start controlling the system. The Legal Notice tool runs on the Launcher, watches for new sessions, and provides a helpful Enter keypress on each new remote session to hopefully

clear the legal notice. Please note that this tool is not officially supported by Engineering. Customers have been known to write their own, too.

Load Test

Load Tests are a specific type of Test available in Login Enterprise. They're designed to simulate multiple active users all running in parallel, to determine the point at which an additional user would impact existing users noticeably. Login Enterprise will launch multiple sessions on Target machines in parallel, ramping up to a specific number of sessions, with those sessions running the defined Application scripts in a repeating loop for the duration of the Test. By default, it includes the EUX data collection and tries to calculate a VSI Max number for your Test configuration. To learn more, see Load Testing.

Local AppData I/O Score

The results of a Test used as part of EUX testing which specifically exercises disk operations in C:\Users\%USERNAME%\AppData\Local, to test that portion of a user's profile. To learn more, see Determining the EUX score.

Location

A geographical location and name that can be assigned to a Launcher for visualization purposes. To learn more, see Locations.

Login Processor

Used synonymously with a Logon Executable. A Login Enterprise software that will run on your target machines, and download and run the Engine to execute Tests.

Logon Script

A logon component/mechanism that targets use to invoke the actual virtual user testing. It's like a bootstrapper. Note the Logon script is the legacy way of doing it and the Logon Executable is the newer way. To learn more, see Logon Components.

M

Measurement

There's a wide range of measurements that the Login Enterprise can take. WMI/Perfmon polling, Launcher protocol latency, custom workload timers, logon timers, etc.

Mixed CPU I/O Score

The results of a Test used as part of EUX testing which simultaneously exercises CPU and disk operations by approximating the in-memory compression of a Test file in the user's profile. To learn more, see Determining the EUX score.

MSI Installer

Windows Installer or Microsoft Installer. A technology developed by Microsoft for installing, maintaining, and removing software applications on the Windows operating system. It consists of installation metadata, files, and instructions required to install an application. The Windows Installer service, integrated into Windows OS, manages the installation process defined by the MSI package, including file copying, registry modifications, and system configuration changes. MSI installers support features like customization, silent installation, repair, uninstallation, and rollback to ensure reliable software deployment. Administrators and developers use tools like Microsoft's Orca to create, customize, and package MSI installer files for distribution.

My Documents I/O Score

The results of a Test used as part of EUX testing which specifically exercises disk operations in the My Documents portion of a user's profile.

O

OpEx

OPEX (Operational Expenditure) covers ongoing business expenses, including recurring costs for software or cloud services. See Operations dashboard.

P

PerfMon

Performance Monitor. A native Windows tool enabling real-time monitoring and analysis of system performance. It offers a graphical interface to observe various performance counters like CPU utilization, memory usage, disk activity, and network traffic. Users can configure PerfMon to display specific counters, set up logging and alerts, and generate reports to diagnose performance issues and optimize system performance. It's widely used by system administrators, IT professionals, and developers to troubleshoot problems, tune configurations, and monitor the health of Windows-based computers and servers. To learn more, see Performance Monitor (PerfMon).

Performance Test

A software testing method focused on assessing the responsiveness, throughput, scalability, stability, and resource usage of a system or application under specific workload conditions. It measures various performance metrics like response times, transaction throughput, and resource consumption to evaluate the system's ability to meet performance requirements and user

expectations. Performance Tests simulate different user scenarios, load profiles, and stress conditions to identify bottlenecks and scalability limitations. Types of performance Tests include Load Testing, stress testing, endurance testing, and scalability testing. Overall, performance testing ensures that software applications deliver acceptable performance levels, maintain responsiveness under peak loads, and sustain user satisfaction and productivity.

Port

Network Port. A port is a communication endpoint used by software to enable communication between computers or devices. It's identified by a number, such as 80 for HTTP (web) traffic or 443 for HTTPS (secure web) traffic. Ports allow different applications and services to operate simultaneously on a single device. To learn more, see Port configuration.

Profile Load

User Profile Load. When a user logs into a system, their personalized settings, preferences, and configurations are loaded from their user profile. This process is sometimes referred to as "profile loading." This can include things like desktop background, application settings, browser bookmarks, etc. To learn more, see Login performance.

Provider

Desktop service providers enable Login Enterprise to establish a connection, facilitating the collection of real-time data for the operations dashboard. To learn more, see Providers.

R

Raw Data Export

Detailed numeric Test result data from a completed Test. To learn more, see Exporting Raw Data.

Raw Measurement Data

The maximum number of days for retaining raw measurement data, collected by the virtual user during Continuous Testing. The collected data includes logon, application, and latency metrics. The metrics are visualized in the Dashboard and Reporting Charts (Hourly-view only) pages.

Read-Only Group

The Active Directory group defined that contains users given Read Only access to the Login Enterprise web interface.

RDS Broker

Remote Desktop Services Broker. A key component in the Microsoft Remote Desktop Services (RDS) infrastructure. It's responsible for coordinating the distribution of Remote Desktop (RD) sessions across multiple Remote Desktop Session Host (RDSH) servers. Acting as a central management point, the RDS Broker handles session load balancing, reconnection, and persistence for users accessing Remote Desktop resources in distributed environments. By optimizing resource utilization and directing users to the most suitable RDSH server based on factors like session availability and user preferences, the RDS Broker enhances user experience. Moreover, it provides high availability and fault tolerance capabilities to ensure continuous access to Remote Desktop resources, even in case of server failures or network disruptions. To learn more, see Microsoft RDS.

RDP Host

Remote Desktop Protocol Host server or simply a terminal server. A computer system within the Microsoft Remote Desktop Services (RDS) infrastructure that hosts Remote Desktop (RD) sessions and applications for remote users. It executes and manages user sessions, providing a remote desktop or application environment via the Remote Desktop Protocol (RDP). RDP Hosts support multiple user sessions simultaneously, enabling users to access applications, and files, and perform tasks remotely. They facilitate centralized desktop and application delivery, allowing organizations to consolidate resources and provide remote access to computing resources for users from different locations.

RDS Gateway

Remote Desktop Services Gateway. A Microsoft Remote Desktop Services (RDS) component that enables authorized remote users to securely access internal network resources from outside the corporate network or over the internet. It acts as a gateway server, providing secure connections using Remote Desktop Protocol (RDP) over HTTPS. RDS Gateway allows organizations to enforce access policies, authenticate users via Active Directory, and encrypt communications to protect sensitive data and maintain network security. It supports network-level authentication (NLA), and SSL encryption, and integrates with Active Directory for user authentication and access control. To learn more, see Microsoft RDS.

S

Script Editor

A tool for building our C# Application scripts and examining programs for what controls are accessible and how to interact with them. To learn more, see Script Editor.

Script Recorder

A tool that captures and records the sequence of user interactions with a computer system or software application. It records user inputs, such as mouse clicks, keyboard actions, and other commands, creating a script that can be replayed to automate repetitive tasks or test software

functionality. Script Recorder v1 is integrated into Script Editor. To learn more, see Script Recorder.

Secret

Confidential information like passwords, keys, or tokens, vital for secure authentication, authorization, or encryption in software systems. Protecting secrets from unauthorized access is crucial, requiring secure storage, encryption, and access controls. Best practices include securely generating, storing, rotating, and revoking secrets to maintain system integrity. In Login Enterprise, secrets authenticate component requests, ensuring their validity and identity.

Session

A Test user that is actively logged into a target and running Test scenario-defined testing from Login Enterprise

Session Metric

Session Metrics are Windows performance metric data that are collected by the Engine throughout a Test Run. They are user-defined, so you can collect any Windows counter that the Virtual User has access to. When the Test Run starts, the Engine collects all of these user-defined performance counters for the duration of the Test and reports them back to the Appliance as part of your Test results. To learn more, see Session Metrics.

SLA Report

A Service Level Agreement report. A document summarizing a service provider's performance in meeting the terms of the SLA, a contract outlining expected service levels. It includes metrics like uptime, response time, and availability. In the context of Login Enterprise, a Continuous Test scenario can generate an SLA report as a PDF based on configured parameters. This report serves to demonstrate key performance indicators (KPIs) or validate if required functionality and performance are met or exceeded in a production environment, crucial for stakeholders. To learn more, see SLA reports.

SSL/TLS

Secure Sockets Layer/Transport Layer Security. SSL and TLS are cryptographic protocols for secure communication over computer networks. SSL was developed by Netscape, succeeded by TLS, and maintained by the IETF. They encrypt data transmission between client and server, ensuring confidentiality, integrity, and authenticity. SSL/TLS uses cryptographic algorithms to establish secure connections, negotiate encryption parameters, authenticate parties, and verify data integrity. They secure protocols like HTTPS, SMTP, and FTPS, enabling secure web browsing, email, file transfer, and online transactions. To learn more, see Encryption.

Steady State EUX Score

The Steady State period represents the time after all users have logged in (login storm) and the system starts to normalize. The Steady State EUX score is the average of the EUX scores captured 5 minutes after all sessions are logged in, until the end of the Test. To learn more, see the Baseline and Steady state EUX scores.

T

Target

An individual end-user machine where a Test will run. Target could also mean the platform as a whole that is hosting the targets (in this case, also referred to as a Target host).

Tenant ID

Azure Tenant is an instance of Azure Active Directory (Azure AD). Find the tenant ID on the App registration tab in the Azure portal.

Test

In Login Enterprise, a Test is the complete configuration describing how a Test should operate. It includes a collection of Application scripts that will perform a series of actions and take measurements as defined in those Application scripts; the Accounts; Launchers; Connector configuration; scheduling information; operational parameters like Session Metric selection, EUX enablement, and logging options.

Test Duration

A time span indicating how long a Load Test is set to run as a whole or during just the steady state segment of a Load Test (after the Test users have ramped up). To learn more, see Load Testing overview.

Total Login Time

The complete amount of time from when the Connector begins the login process for the user session to the point where the final user-session executable (usually Windows Explorer) is up and running. To learn more, see Measuring total login time.

Trusted Certificate

A digital certificate issued by a reputable certification authority (CA) and acknowledged as valid by client software or operating systems. These certificates are issued by CAs that follow industry standards and verification practices, ensuring the security of certificate issuance. Client applications and OSs maintain a trust store containing public keys of trusted CAs, allowing certificates signed by these authorities to be deemed trustworthy. Trusted certificates are essential for secure communication, online transactions, and data exchange, ensuring authenticity and integrity in digital identities. To learn more, see Managing certificates.

U

Unit Test

Automated Tests that check individual parts (units) of software to ensure they work correctly in isolation, typically fast, focused, and deterministic.

Untrusted Certificate

A digital certificate is not accepted as valid by client software due to reasons such as expired validity, mismatched signatures, or unknown issuers. Client applications may warn users about potential security risks, like data interception, if they choose to communicate using such certificates. To learn more, see Managing certificates.

User

An account or an actual human user interacting with a target.

UWC

Universal Web Connector. For VDI systems where Login VSI doesn't provide a Connector, or where the built-in Connector doesn't quite work, you can build your own using an Engine-like scripting system for interacting with web-based connection broker interfaces. To learn more, see the Universal Web Connector.

V

VDI

Both an individual virtual Windows desktop and the entire infrastructure for managing and accessing those individual desktop machines.

Verbose Logging

Detailed logging mode in software apps that produces extensive log messages, including debug info and diagnostic data. It's useful for troubleshooting and debugging but can increase log file sizes and resource usage.

Virtual User

Sometimes also referred to as a Robotic, Synthetic, or Test User.

VSImax

The maximum number of concurrent active VDI sessions your hardware will support without compromising responsiveness. To learn more, see the EUX score and VSImax.

W

Web Automation Driver

Software for automating web browser interactions, offering APIs to control behavior programmatically. It's used for tasks like testing, web scraping, and automating workflows. Examples include Selenium WebDriver and Microsoft WebDriver. Login Enterprise employs Selenium drivers.

Workload

Used synonymously with Application script. A Workload is what a virtual user uses to perform user-like interactions within the Windows targets.

WMI

Windows Management Instrumentation. A suite of system management tools within Windows OS. It offers a standardized framework to access and manage system info, settings, and resources locally or remotely across a network. Developers and admins use WMI to automate tasks, monitor system performance, retrieve hardware/software info, and perform management operations through scripts or apps. WMI uses a hierarchical namespace structure and supports querying with WQL (WMI Query Language), similar to SQL. To learn more, see Windows Management Instrumentation (WMI).

Back to Product Overview

Installation and Deployment

This chapter provides a comprehensive guide to the installation and deployment of the Login Enterprise Virtual Appliance. It begins with the preparations necessary for a successful setup, detailing the Virtual Appliance requirements, navigating the Downloads page, and offering a checklist to ensure technical readiness.

Following that, it explores various deployment methods, including deploying a VHD to Azure, setting up Microsoft Azure Virtual Desktop (AVD), and deploying OVA templates on VMware vSphere and Citrix XenServer, as well as VHDX on Microsoft Hyper-V, and utilizing the Nutanix CALM Marketplace.

Then, it offers a basic walkthrough of setting up a new Appliance and running a simple Test. Finally, it includes steps for updating the Virtual Appliance and Debian.

- Preparing for Installation
  - Downloads
  - Installation Checklist
- Deploying the Appliance
  - Deploying the VHD to Microsoft Azure
  - Deploying the OVA Template to VMware vSphere
  - Deploying the OVA Template to Citrix XenServer
  - Deploying the VHDX on Microsoft Hyper-V
  - Deploying the OVA to Nutanix CALM Marketplace
  - Deploying the OVA to Nutanix AHV
- Installation Walkthrough (Quick Start)
- Managing Updates and Upgrades
  - Updating the Appliance
    - Updating the Appliance from v5 to v6
  - Updating Debian

Preparing for Installation

This chapter describes the essential preparations for downloading and setting up the Login Enterprise Virtual Appliance. It explains how to obtain downloads for deploying Login Enterprise in different environments and provides a comprehensive checklist to ensure you have everything needed for successful installation.

For detailed information, refer to the corresponding sections within this subchapter.

- Downloads
- Installation Checklist

Downloads

To access the Virtual Appliance downloads, including versions for Hyper-V, Citrix, and Azure, as well as updates for existing installations, you must be signed in with your Zendesk account. If you do not have an account, contact us to request access.

Once signed in, you will also have access to SHA256 hashes. These hashes are used to verify the integrity of the downloaded files, ensuring that they have not been tampered with during transmission. Additionally, you can download the Linux Launcher and IGEL Launchers.

Installation Checklist

- Introduction
- Before you start: Gather the following resources
- Account checklist
- Launcher resources
- Server and broker information
- Additional information to verify

Introduction

This checklist outlines the resources and information required before beginning the installation of the Virtual Appliance. Ensure that all necessary components, configurations, and authorizations are in place before proceeding.

Before you start: Gather the following resources

1. **Hypervisor Information**
    - Ensure that your hypervisor is on the supported list:
        - VMware ESXi 5.5 - 7.0 (OVA file)
        - Microsoft Hyper-V - Server 2012 R2 and higher (VHDX file)
        - Citrix Hypervisor 8.2 (OVA file)
        - Microsoft Azure (VHD file)
    - Ensure that you have the necessary permissions to make changes to the hypervisor, or coordinate with the Infrastructure Admin team if needed.
2. **Hardware Resources for Virtual Appliance**
    - Ensure the following resources are available for the **VM hosting the Virtual Appliance**:
        - **CPU**: 4 vCPU
        - **RAM**: 8GB RAM
        - **Storage**: 100 GB virtual Hard Disk
    - Download the Virtual Appliance in your preferred format (OVA, VHD, or VHDX). For details, see Downloads.
3. **Networking and DNS Information**
    - Obtain the Fully Qualified Domain Name (FQDN) for the Virtual Appliance.
    - Ensure that the appropriate DNS settings are in place.
4. **Active Directory (AD) Accounts**
    - Create the necessary Virtual User accounts in Active Directory and ensure they are licensed for the appropriate systems and applications (e.g., RDS CALs, Citrix, Office365, etc.)
    - Configure the Logon Executable via Group Policy for Virtual Users at the user level (not machine level).
    - Create an AD account for the Launcher to log in, and consider creating a service account for sending email alerts.

Account checklist

Before you begin the installation, verify that your **user accounts** and **logon configurations** are set up correctly:

1. **Test User Login**
    - Log in with one of your test user credentials to access the target environment, applying the same authentication method as your end users
        - This ensures that users can authenticate and access the virtual desktops or applications they will need.
2. **Active Directory User Permissions**
    - Ensure that the Virtual User accounts are correctly synchronized with Active Directory and authorized to access the target resources (e.g., Citrix, VMware, etc.)
    - Ensure that the Admin AD User Group has appropriate administrative access to the required resources.
    - Ensure that the Read-only AD User Group has the necessary read-only access to the required resources.
3. **Logon Configuration**
    - Ensure that the Logon Executable is configured to start during the logon process via Group Policy at the user level (not the machine level).
        - There are several ways to fulfill this requirement:
            - Group Policy to execute Logon Executable at Windows logon
            - NETLOGON/ Logon Script for the Virtual User AD User
            - shell:startup (For single-user OS scenarios, not recommended for production)
            - shell:common startup (For multi-session OS scenarios, not recommended for production)

Launcher resources

1. **Launcher Hardware**
    - Ensure that the Launcher machines meet the minimum hardware requirements:
        - 2 vCPU
        - 4GB RAM
        - Minimum available disk space: 1GB free virtual hard disk space.
2. **Logon Configuration**
    - Configure auto-login for the Launcher machines if necessary.
3. **Launcher Software**
    - Download and prepare the Launcher software for installation on the required machines.
        - Windows 7 and later (32/64-bit) or Windows Server 2012R2 or later
        - Microsoft .NET Framework 4.8
        - Client Connector Application (Citrix Workspace, VMware Horizon Client, Universal Web Connector)
        - Login Enterprise Launcher

4. **Client Connector**
    - Ensure that the Client Connector software (e.g., Citrix Workspace, VMware Horizon, RDP, UWC) is installed and ready on the Launcher machine.

Server and broker information

1. **Broker Configuration**
    - Ensure that your brokers (e.g., Citrix, VMware Horizon) are configured and ready to accept connections from Virtual Users.
2. **Resource Pools/VMs**
    - Ensure that the resource pools and VMs needed for Virtual Users and Launchers are set up in your hypervisor or virtualization environment
3. **Certificates**
    - Obtain the necessary certificates (if applicable) for secure communication and ensure they are installed and configured for the Virtual Appliance and Launchers.

Additional information to verify

1. **Permissions and Access Rights**
    - Ensure you have the appropriate permissions to install and configure software on the necessary machines (Launcher, Target Environment, etc.). If not, coordinate with the relevant teams.
2. **Licensing Requirements**
    - **Virtual User Licensing**: Ensure that each Virtual User has the correct licensing to interact with the target resources (e.g., RDS CALs, Citrix, etc.)

Deploying the Appliance

This subchapter describes the various methods and steps for deploying the Login Enterprise Virtual Appliance across different platforms. It covers the installation process in detail, including:

- Deploying the VHD to Microsoft Azure
- Deploying the OVA Template to VMware vSphere
- Deploying the OVA Template to Citrix XenServer
- Deploying the VHDX on Microsoft Hyper-V
- Deploying the Appliance to Nutanix CALM Marketplace
- Deploying the OVA to Nutanix AHV

Deploying the VHD to Microsoft Azure

- Overview
- Uploading VHD to Azure
- Creating an image
- Configuring the cloud-init file
  - Configure the Virtual Appliance password
  - Configuring the Virtual Appliance domain name
- Uploading the cloud-init file to Azure Cloud Shell
- Creating the Virtual Machine
- Custom requirements
  - Adding a Public IP
  - Integrating with existing networks

Overview

This section outlines the steps to deploy the Login Enterprise Virtual Appliance to Azure. If you haven't already, download the VHD from the Downloads page (Be sure to select the Microsoft Azure download option).

The instructions below outline a general deployment pattern and may not account for all environment-specific requirements. Organizations with existing Azure networking infrastructure, such as virtual networks, subnets, or network security groups (NSGs), may need to modify the CLI examples to fit their architecture.

NOTE: While key configuration concepts are covered, these steps are not a comprehensive deployment reference, and additional customization may be necessary.

Uploading VHD to Azure

Although the VHD is compressed in a ZIP file, it remains large. For best results, download it from an Azure VM with sufficient bandwidth to ensure a fast and reliable transfer. The same Azure VM can also be used to upload the file in the next step.

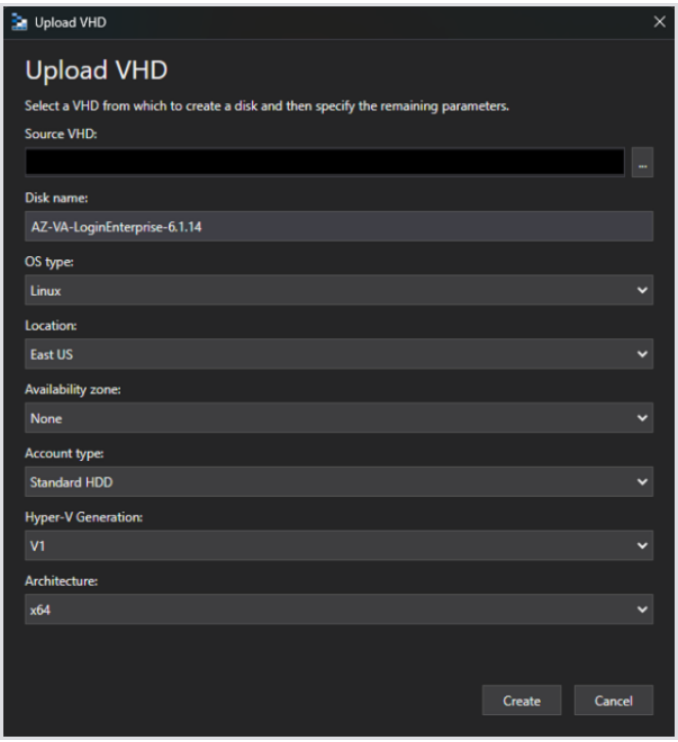
You need to upload the VHD to Azure using Microsoft Azure Storage Explorer. You can download it by clicking the link below. Note that this will start the download immediately. Storage Explorer allows you to upload a raw .VHD file directly to a resource group as a managed disk, simplifying the deployment process.

Download Microsoft Azure Storage Explorer

After signing in to your Azure tenant, navigate to the subscription where you want to upload the disk.

1. In the **left-hand menu**, select **Disks**, then choose the resource group where the managed disk will be created.
2. Select **Upload** and complete the Upload VHD Wizard form.

Make sure to select **Linux** as the operating system type and **Generation 1 (V1)** for Hyper-V generation, as Generation 2 (V2) disks are not compatible with Login Enterprise VHDs.



Creating an image

This section provides an Azure CLI walkthrough for automating the Login Enterprise deployment process. Due to dependencies on cloud-init files, it is recommended to use Azure Cloud Shell, which supports file uploads.

Before proceeding, verify the Hyper-V generation of the uploaded VHD. If Generation 2 (V2) was used, you must reupload the disk using Generation 1 (V1), as V2 is not supported for Virtual Appliance VHDs.

```
# Variables

$resourceGroupName = "" # The resource group where the image will be created

$imageName = "" # The name for the new image

$sourceDisk = "" # The managed disk or VM name to use as the source

$storageSku = "" # The storage performance tier for the image
```

```
# Create the image

az image create --resource-group $resourceGroupName --name $imageName --
source $sourceDisk --storage-sku $storageSku --os-type Linux
```

Once the command has run successfully, a managed disk will appear in the specified resource group and can be used to deploy a virtual machine.

Configuring the cloud-init file

This section includes a `.yml` file, which serves as a cloud-init configuration file. You can download the file from here. It is passed as custom data during virtual machine creation. The file sets the Virtual Appliance's password and configures the expected domain name, which is used to define its fully qualified domain name (FQDN).

Configure the Virtual Appliance password

The 5th line of the downloaded cloud-init file contains a placeholder password. Before deployment, you must replace this with your own password, encoded as a Base64 string. The Appliance expects the password in this format.

To convert your password to Base64, use the following PowerShell command:

```
[System.convert]::ToBase64String([System.Text.Encoding]::ASCII.GetBytes("<pwd
_here>"))
```

Once you have the Base64-encoded string, replace the placeholder value in your local copy of azure.yml. This password will be used by the root user to access the Virtual Appliance.

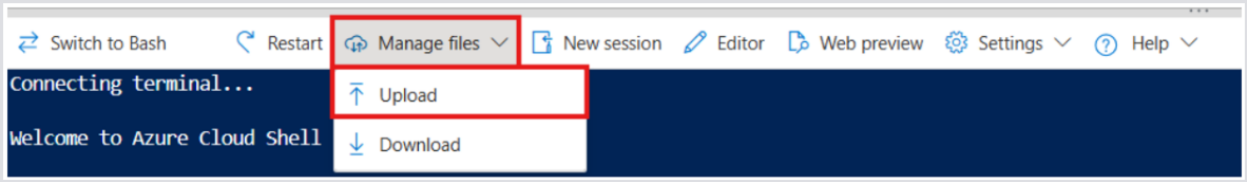
Configuring the Virtual Appliance domain name

In the downloaded azure.yml file, the fourteenth line contains a placeholder for the Appliance's expected domain name. This domain name, combined with the Virtual Machine's hostname, is used to configure the HTTPS listener.

If you are using custom DNS, replace the placeholder with your organization's domain, e.g., "contoso.com." For instance, if the hostname is "logent" and the domain name is "contoso.com.", the Appliance will be configured to listen for incoming requests at "logent.contoso.com."

Uploading the cloud-init file to Azure Cloud Shell

After configuring the cloud-init file, upload it to your Azure Cloud Shell session. In Cloud Shell, open the **Manage files** menu and select **Upload**. This provides *local* access to the file within Cloud Shell.



Creating the Virtual Machine

With the cloud-init file uploaded to your Cloud Shell session, you can now create the Virtual Machine resource. Before proceeding, verify that the YAML file is present in your Cloud Shell environment. For example, run the following command to list the contents of the current directory:

```
dir
```

You should see the uploaded azure.yml file listed in the output.

Login Enterprise requires 4vCPU and 8GB RAM; ensure you choose an appropriate virtual machine.

To deploy the Virtual Machine, use the following Azure CLI command:

```
# Variables

$resourceGroup = "" # The RG where VM will be created

$location = "" # Azure region to create VM in

$vmName = "" # Name for the new VM

$vmSize = "" # VM size (SKU)

$imageName = "" # VM image name to deploy

$storageSku = "" # Storage tier of OS disk eg Standard_LRS

# Create the Virtual Machine, Vnet, Snet, NSG, OS Disk, no Public IP

az vm create --resource-group $resourceGroup --location $location --name
$vmName --size $vmSize --image $imageName --custom-data "./azure.yml" --
authentication-type "password" --admin-username "azureadministrator" --admin-
password "fakePassword123!" --storage-sku $storageSku --os-disk-size-gb 128 -
-public-ip-address ""
```

This command creates the following resources within the specified resource group:

- A Virtual Network and subnet
- A Network Security Group (NSG) associated with the subnet
- A Network Interface linked to the subnet and NSG

- An OS disk (128 GiB). Note that only 100 GiB is required, but you are billed for the full 128 GiB.

The provided examples do not assign a public IP address to the Virtual Machine. As a result, the virtual Appliance will not be accessible over the internet. In hybrid environments, accessing the Appliance may require private connectivity solutions such as Azure ExpressRoute or a Site-to-Site VPN.

Custom requirements

As noted, these instructions do not cover every unique scenario or its specific dependencies.

By default, when creating a Virtual Machine, Azure assigns a public IP address. The deployment command above disables this default behavior.

Adding a Public IP

In many environments, a public IP address is not desirable, so the example command above does not configure one. To add a public IP to the Login Enterprise Virtual Machine, remove the parameter that disables it from the command, and include the parameter that sets the DNS label for the public IP.

```
az vm create --resource-group $resourceGroup --location $location --name
$vmName --size $vmSize --image $imageName --custom-data "./azure.yml" --
authentication-type "password" --admin-username "azureadministrator" --admin-
password "fakePassword123!" --storage-sku $storageSku --os-disk-size-gb 128 -
-public-ip-address-dns-name $vmName
```

By allowing Azure to create a Public IP for the VM and specifying a DNS Name for the Public IP, the Appliance will now be publicly available. Ensure the YAML file domain name corresponds with the one used here.

Integrating with existing networks

In some environments, the Login Enterprise Virtual Machine may need to use existing virtual networks and/or subnets. For example, these networks might already have connectivity to on-premise networks via Site-to-Site VPN or ExpressRoute.

If the Virtual Machine should be attached to an existing Virtual Network and/or subnet, create a Network Interface (NIC) configured to use those networks. Then, use the `--nics` parameter with the `az vm create` command to specify the `resourceId` of the Network Interface that the Virtual Machine should use.

By default, the example `az vm create` command will create a NIC if the `--nics` parameter is not used. When used, Azure will interpret the Virtual Network, Subnet, and Network Security Group to use based on the specified NIC configuration. This requires manually creating and configuring the Network Interface, which is out of scope for this section.

Deploying the OVA Template to VMware vSphere

- Overview
- Deploying an OVA Template in VMware vSphere

Overview

This section provides step-by-step instructions for deploying an OVA template on VMware vSphere.

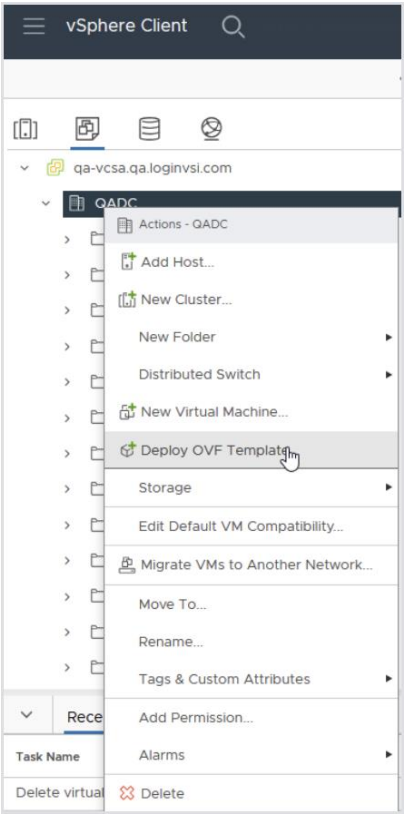
Before you begin, make sure to download the OVA template.

Follow these instructions to deploy the Virtual Appliance, power it on, and prepare for the next steps in configuration.

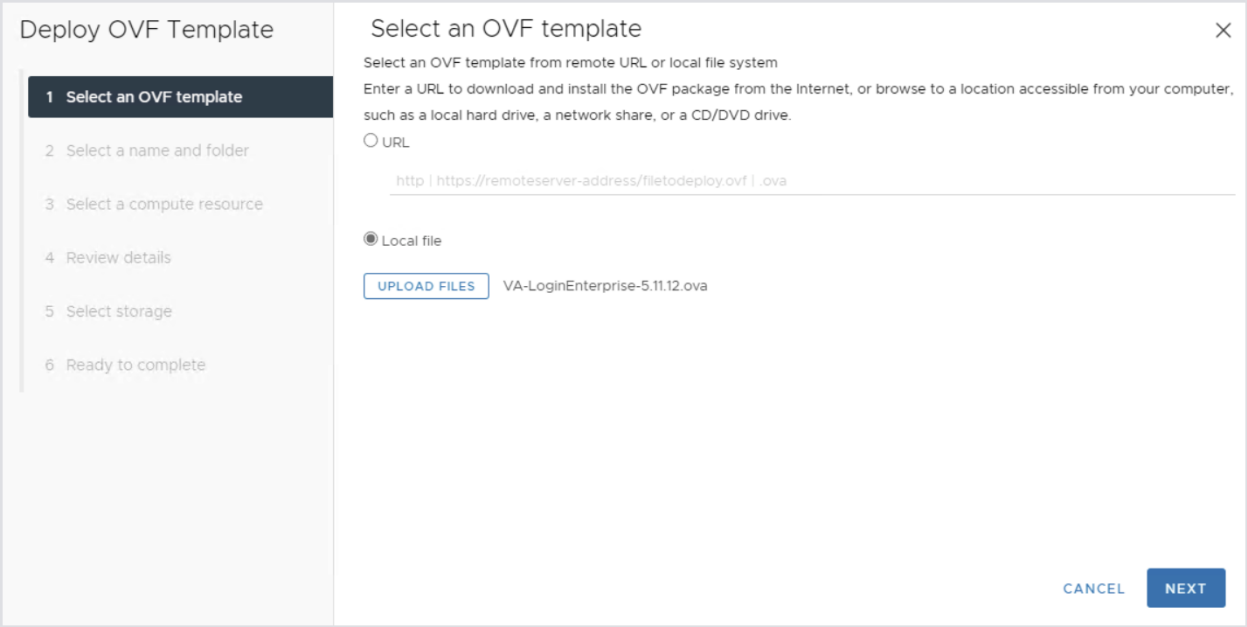
Deploying an OVA Template in VMware vSphere

1. Sign in to the VMware vSphere Web Client.

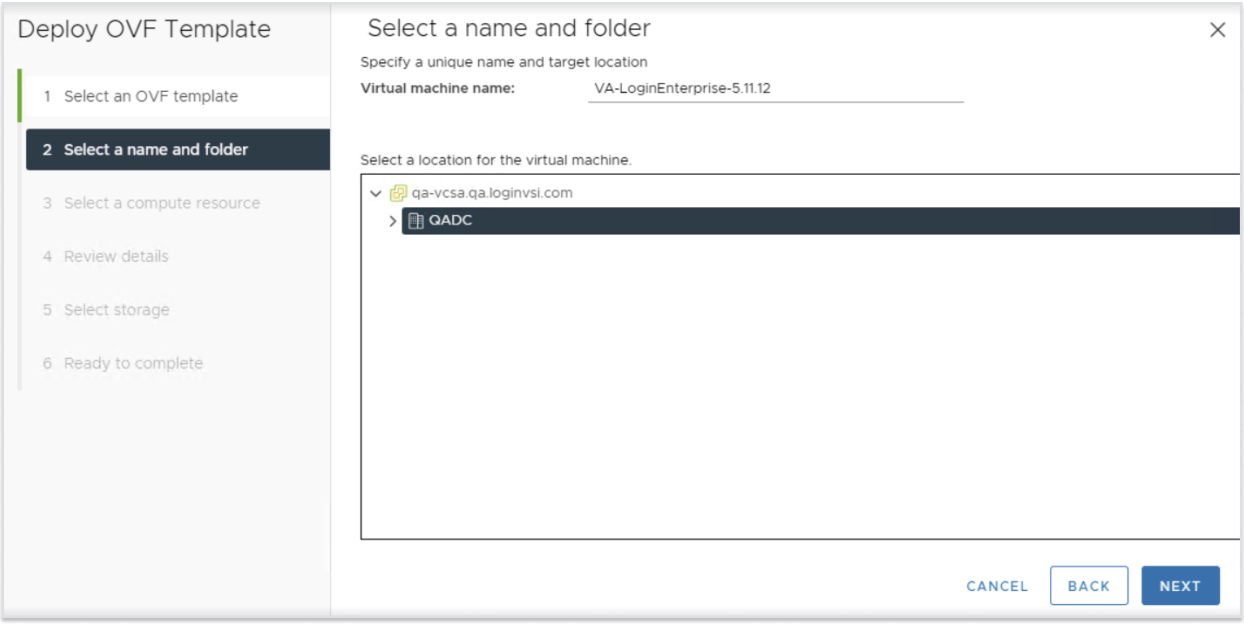
2. Right-click the target resource and select **Deploy OVF Template**.



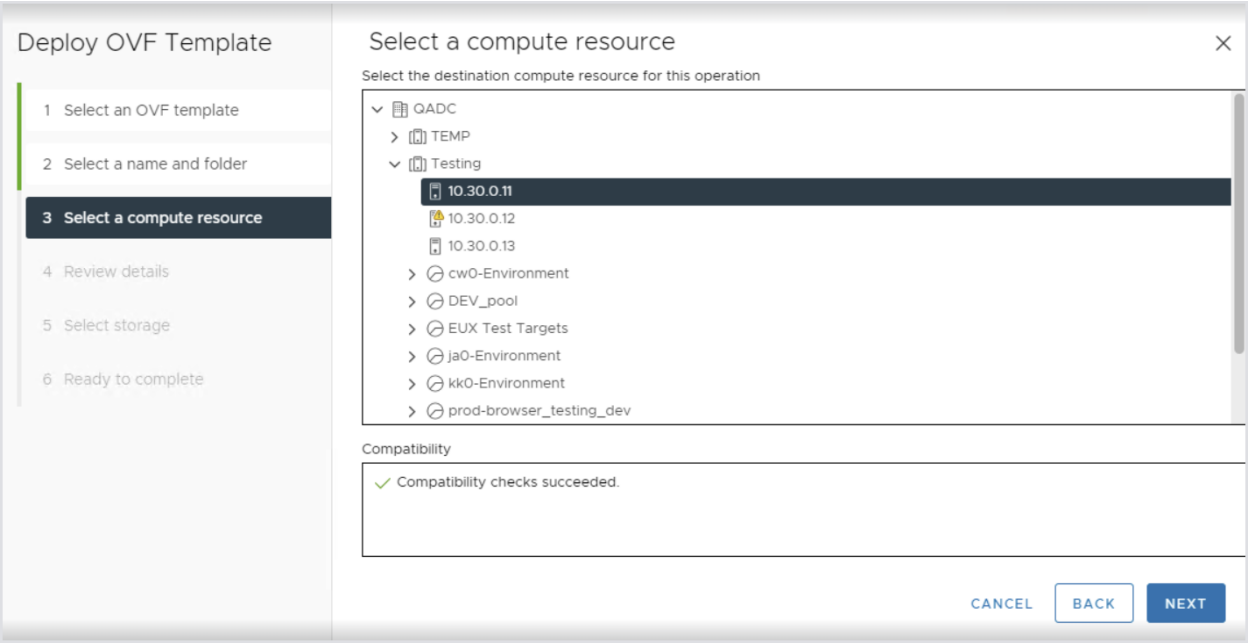
3. In **Deploy OVF Template**, click **Select an OVF template**, and then click **Next**.



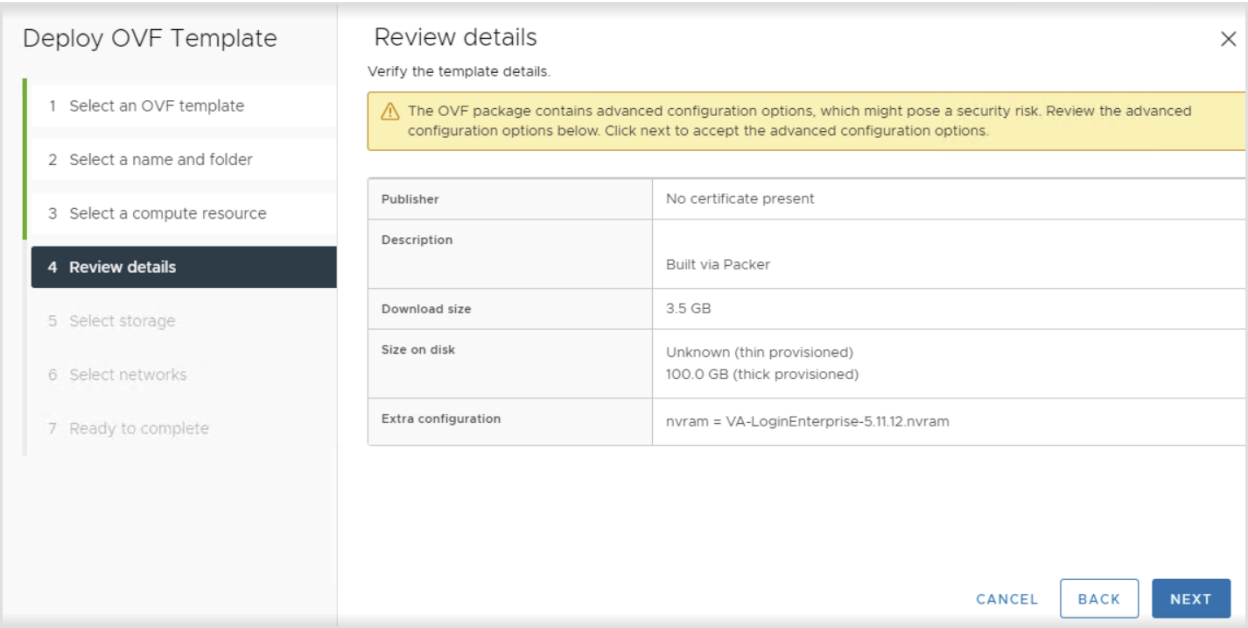
4. Select a name and folder, then click **Next**.



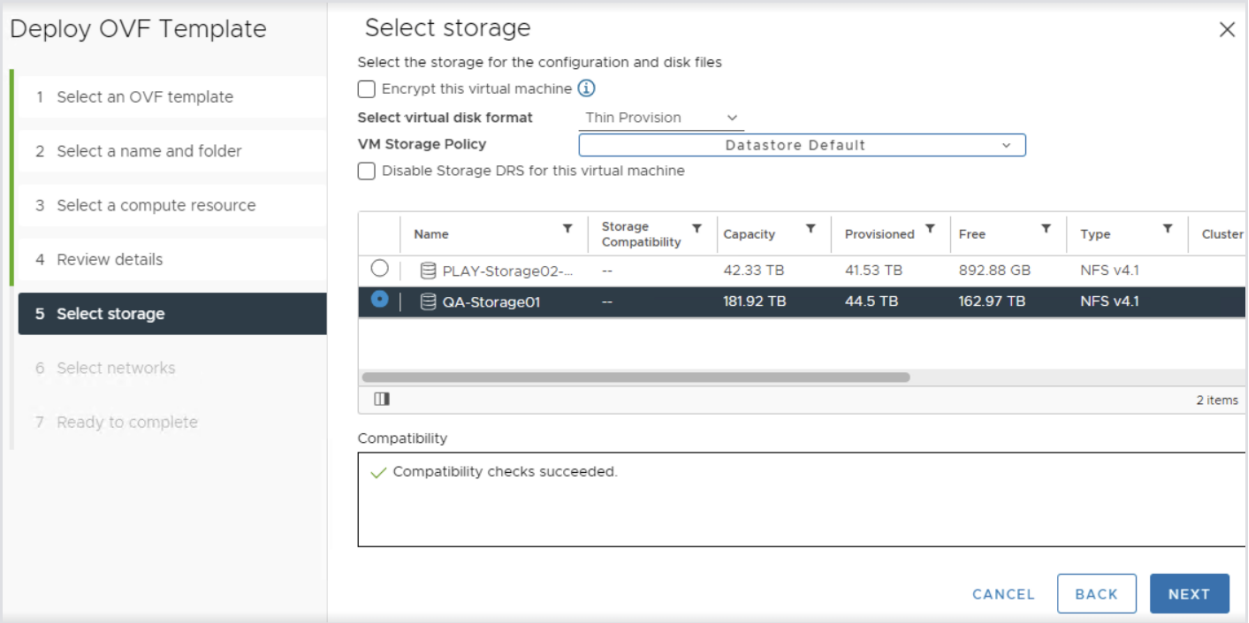
5. **Select a compute resource** and click **Next**.



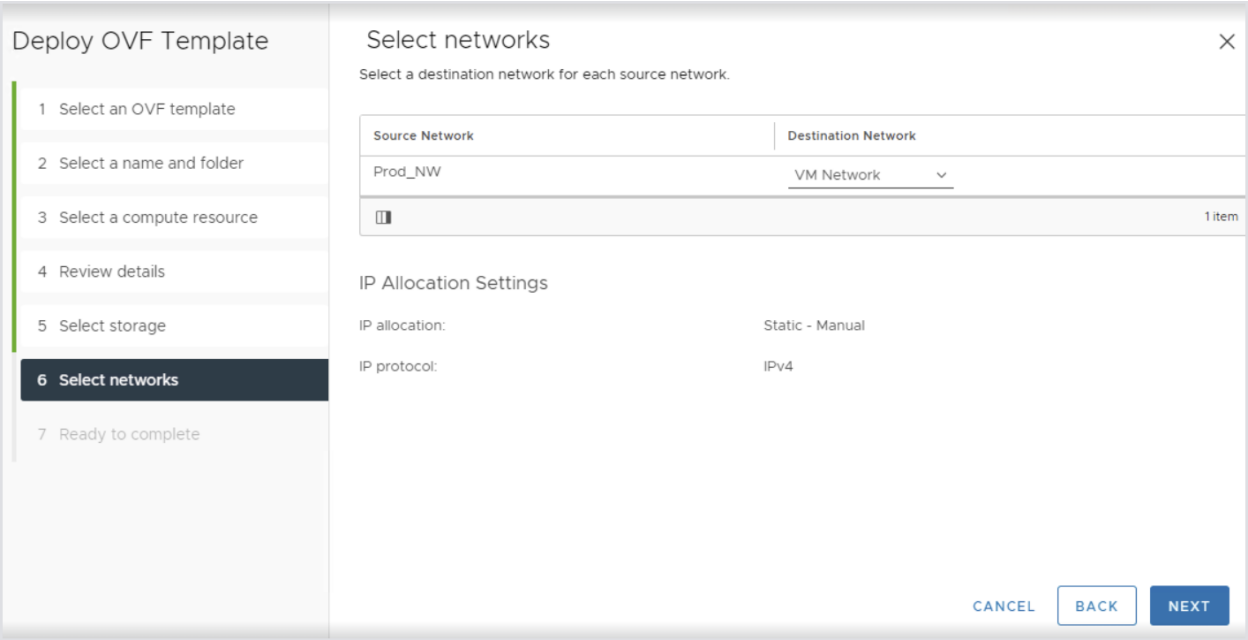
6. Review the deployment details, then click **Next**.



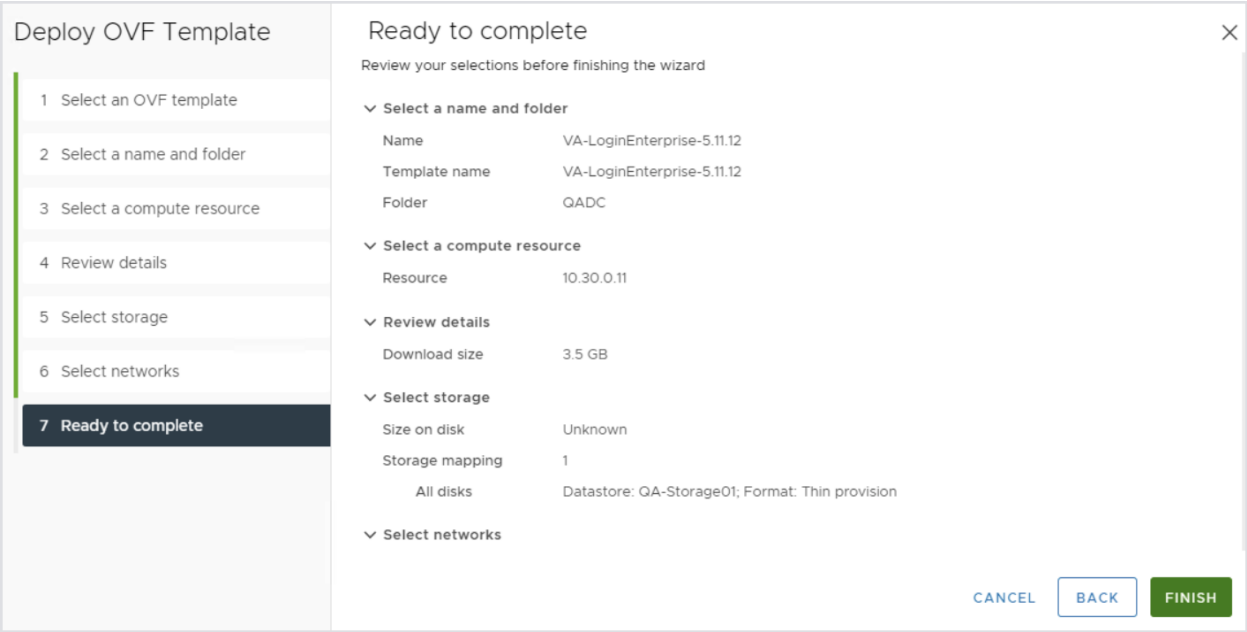
7. Select the storage for the VM, then click **Next**.



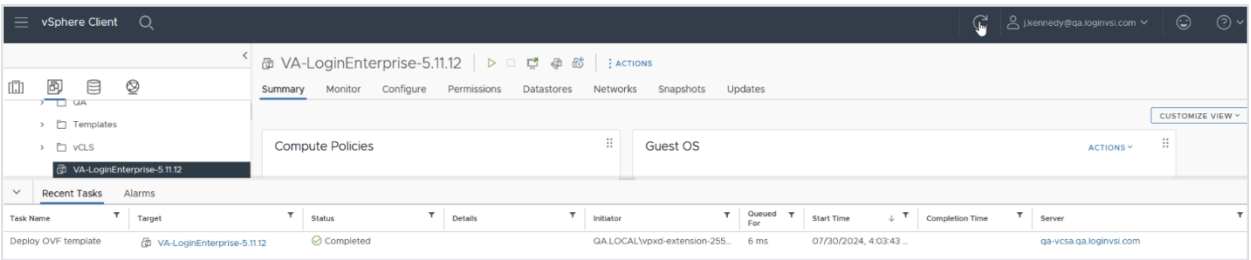
8. Select the network for the VM, then click **Next**.



9. In **Ready to Complete**, review the deployment, then click **Next**.



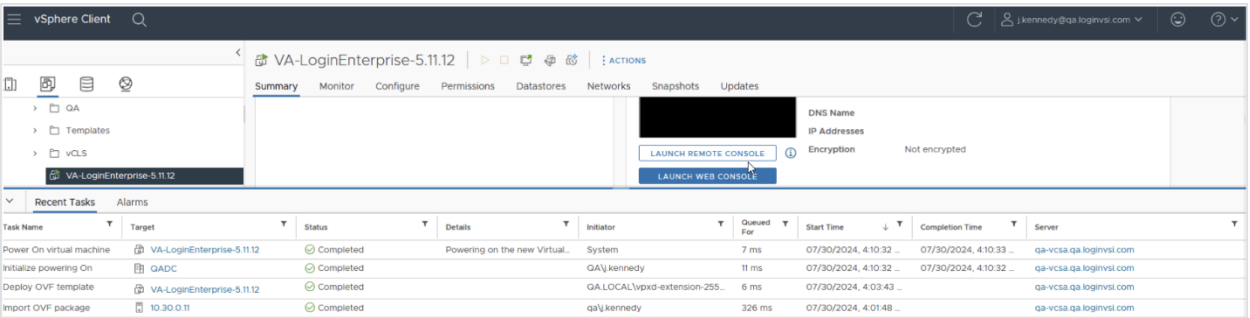
10. Wait for the deployment to complete, then click **Refresh**.



a. Click on the deployed resource.

b. Click the **Play** button to power on the Login Enterprise virtual appliance and begin booting.

11. Click **Launch Remote Console** to open the console for the virtual appliance.



You have now successfully deployed the OVA template and powered on the VM. The next step is to [configure the Virtual Appliance.]

Deploying the OVA Template to Citrix XenServer

- Overview
- Prerequisites
- Steps

Overview

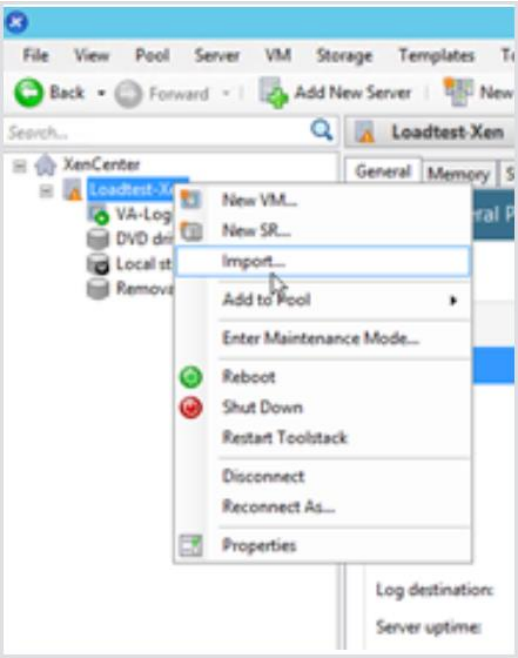
Deploying an OVA template on Citrix XenServer involves importing the OVA file, configuring various settings, and starting the virtual machine (VM). Below are step-by-step instructions to help you through the process.

Prerequisites

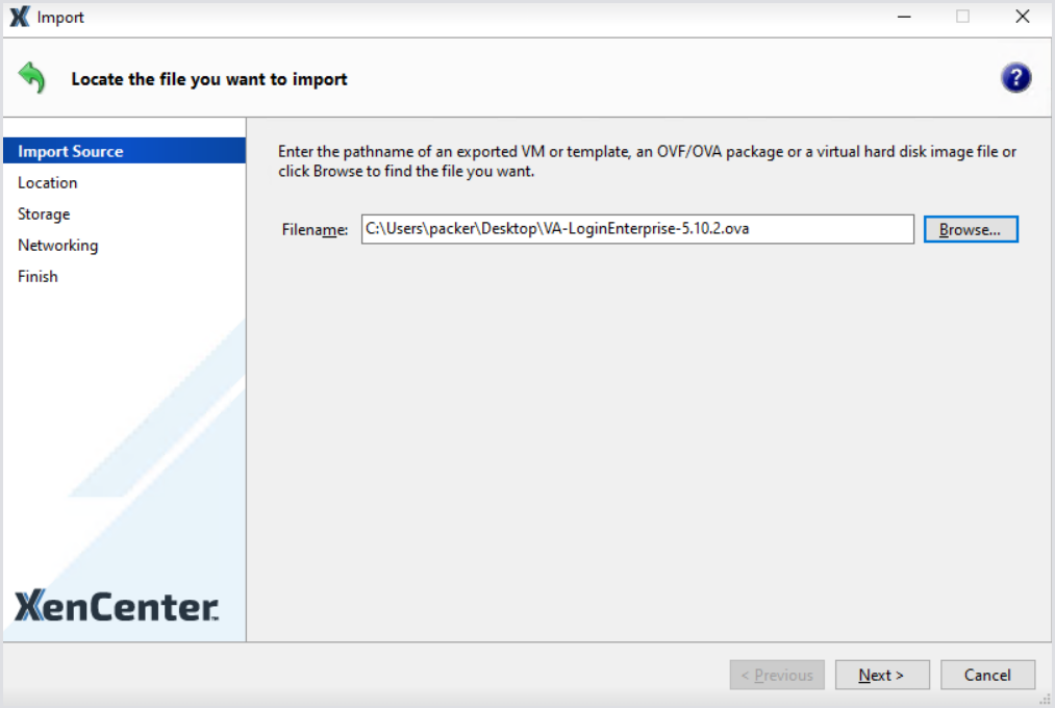
1. **Citrix XenServer**: Ensure that your Citrix XenServer is properly installed and running.
2. **XenCenter**: Install XenCenter on your management workstation.
3. **OVA File**: Download the OVA template you wish to deploy.

Steps

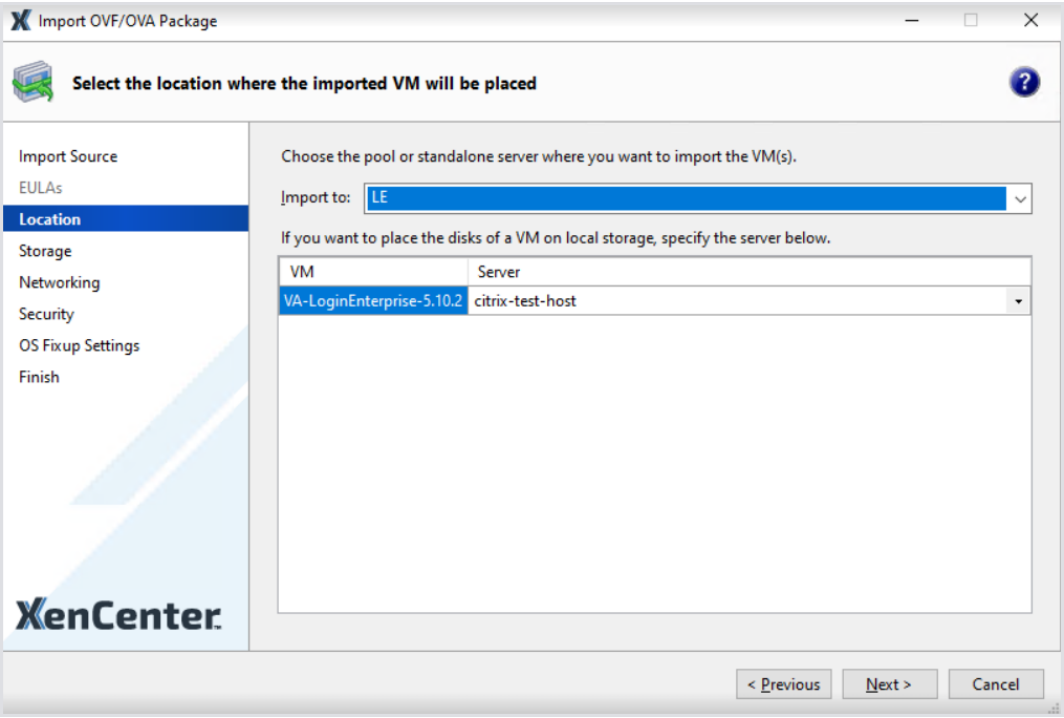
1. **Launch XenCenter**: Open XenCenter on your management workstation and connect to your XenServer.
2. **Import the OVA File**: Navigate to the **File** menu and select **Import**.



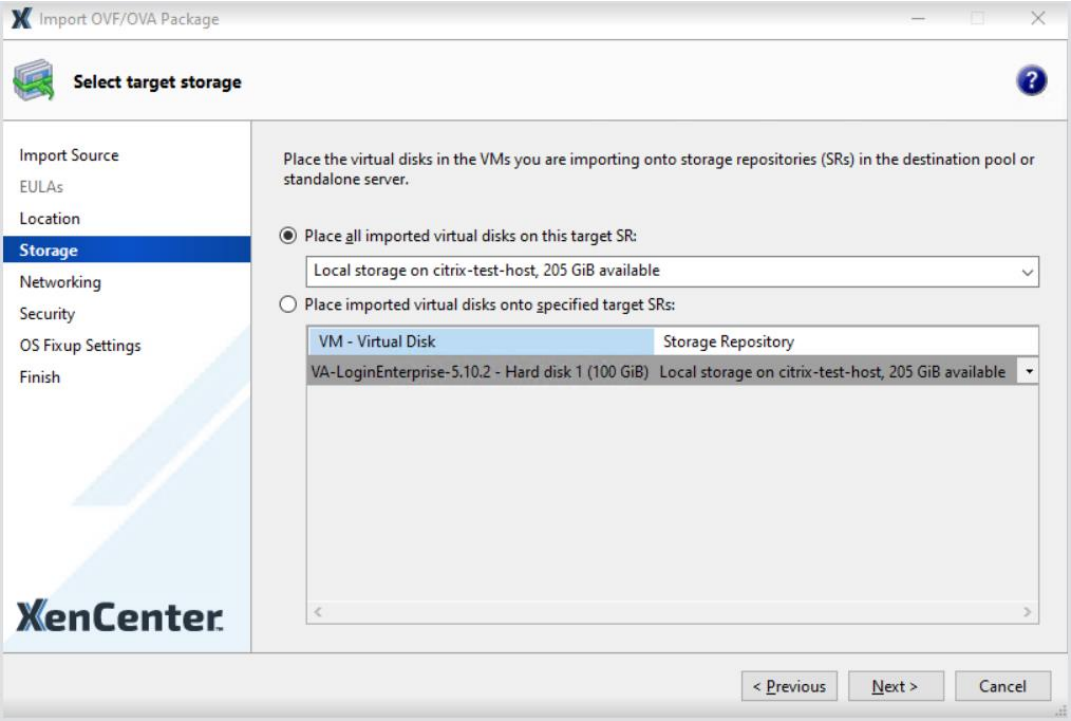
3. In the **Import wizard**, browse to the location of your OVA file and select it. Click **Next**.



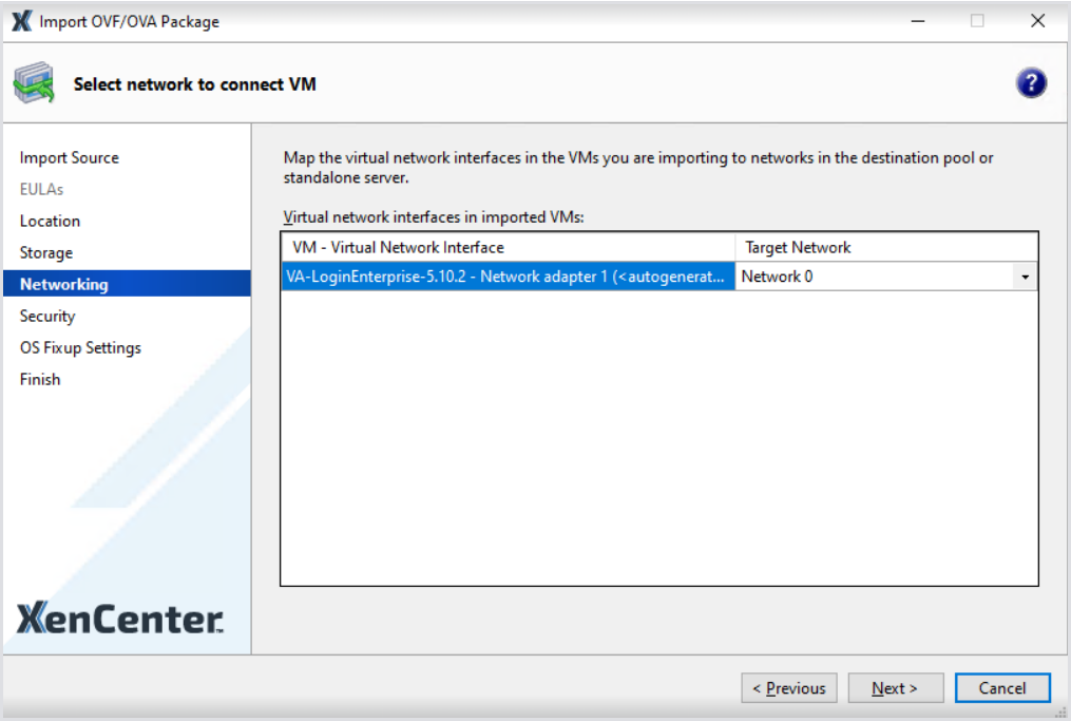
4. **Select a Server**: Choose the XenServer where you want to deploy the Virtual Appliance, and click **Next**.



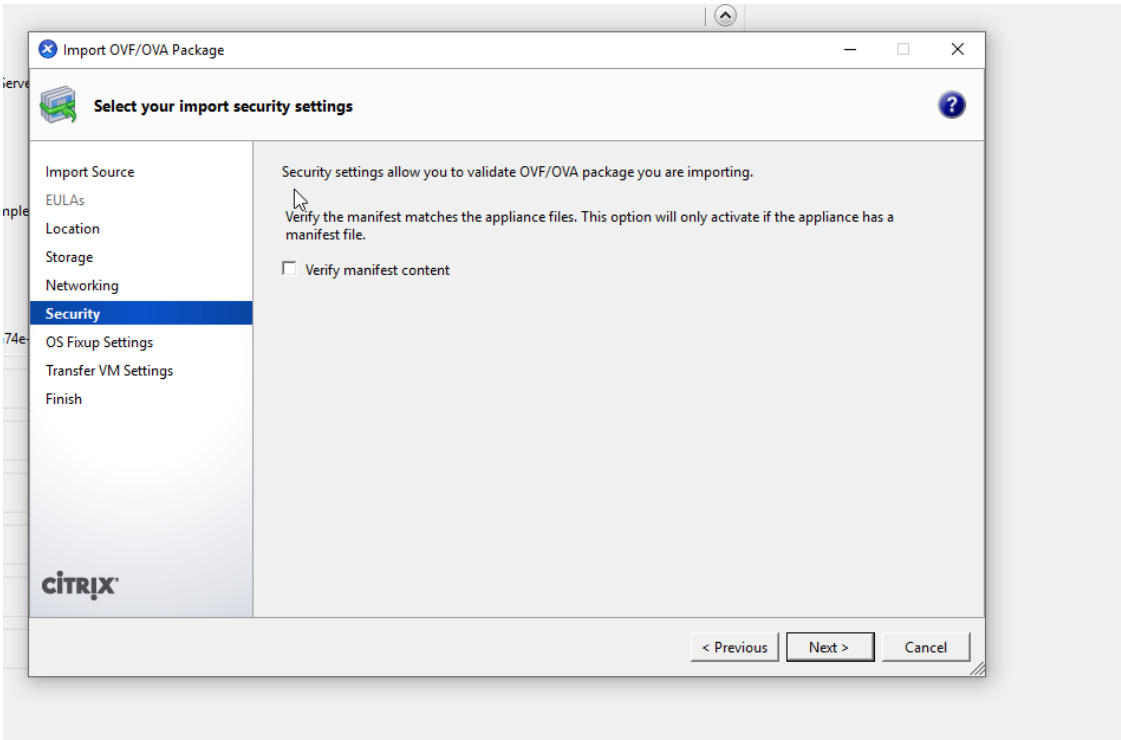
5. **Storage configuration**: Select the storage repository (SR) where you want the virtual machine (VM) to reside. Click **Next**.



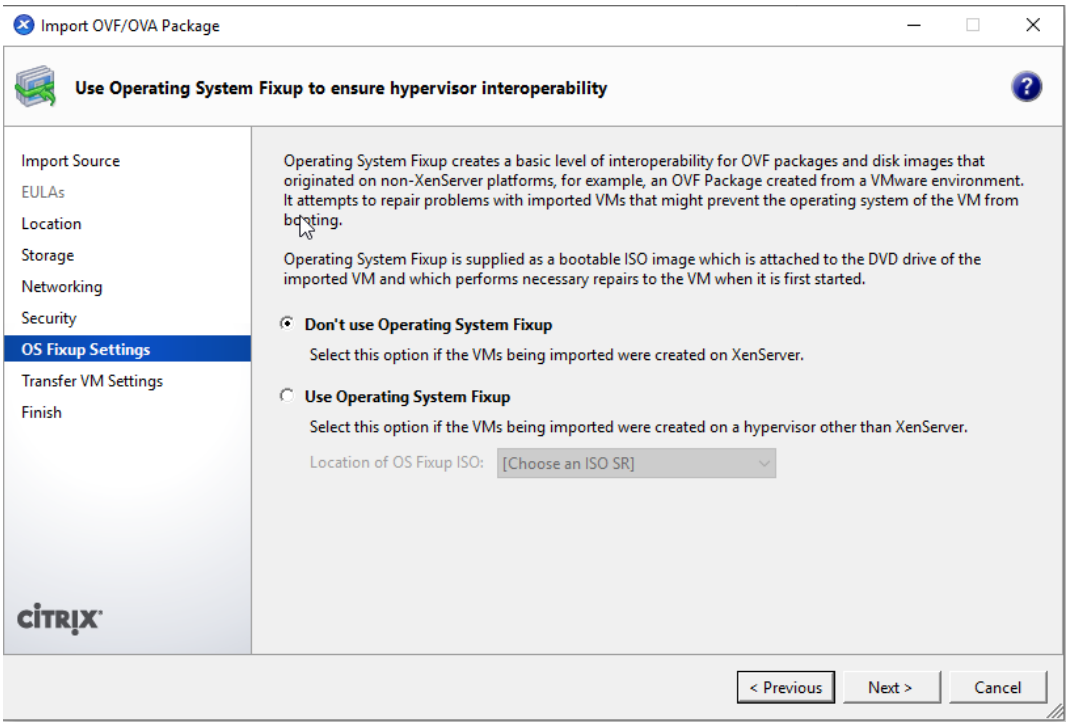
6. **Network configuration**: Assign the appropriate network settings to the VM. Choose the network interface card (NIC) that the VM will use to connect to the network. Click **Next**.



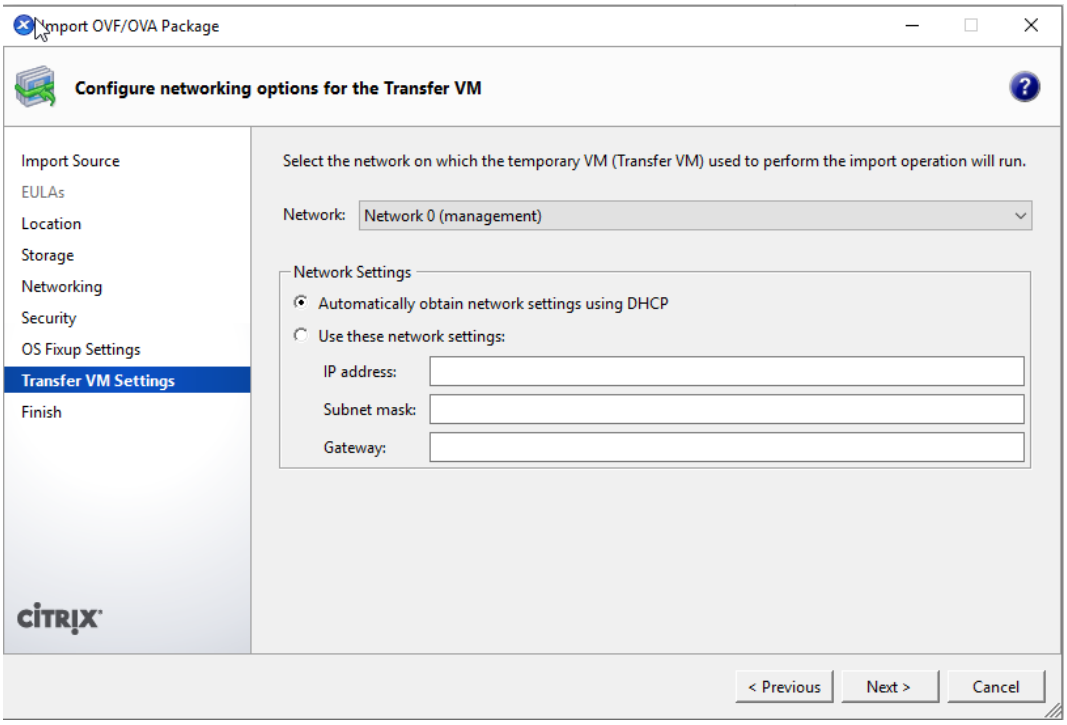
7. **Security configuration**:
    1. **Validate OVF/OVA Package**: You will see a window titled "Security settings allow you to validate OVF/OVA package you are importing."
    2. **Verify the manifest matches the appliance files**: This option will only be available if the appliance has a manifest file.
    3. **Verify Manifest Content**: Check the box labeled "Verify manifest content" to ensure that the manifest file matches the appliance files. This verification helps ensure the integrity and authenticity of the OVA package. Click **Next**.



8. **OS Fixup Settings**:
    1. In the OS Fixup settings window, ensure that the option **Don't use Operating System Fixup** is selected.
    2. This setting ensures that no additional fixup operations are performed on the operating system of the imported VM. Click **Next**.



9. **Transfer VM Settings**:
    1. In the Transfer VM Settings window, leave the option **Automatically obtain network settings using DHCP** selected.
    2. Note that setting up the NIC for the Virtual Appliance will be done from the console after the machine is started. Click **Next**.



10. **Finish**: Review the configuration settings you have chosen. If everything looks correct, click **Finish** to start the import process.
11. **Monitor the Import Process**: The import process will begin, and you can monitor its progress in XenCenter. This may take some time, depending on the size of the OVA file and the performance of your XenServer
12. **Start the VM**: Once the import process is complete, the new VM will appear in XenCenter. Select the VM and click the Start button to power it on.
13. **Post-Deployment Configuration**: After the VM starts, configure the NIC for the Virtual Appliance from the console according to your network requirements. You may also need to configure additional settings, such as network configuration, installing VM tools, and other customizations, according to your needs.
14. **Verify the Deployment**: Verify that the VM is running correctly. Check for proper network connectivity, disk performance, and other parameters to ensure the VM is operating as expected.

By following these instructions, you should be able to deploy an OVA template on Citrix XenServer smoothly. If you encounter any specific issues, refer to the Citrix XenServer documentation or contact us.

Deploying the VHDX on Microsoft Hyper-V

- Overview
- Step 1: Download the VHDX File
- Step 2: Open Hyper-V Manager
- Step 3: Creating a New Virtual Machine
- Step 4: Configuring Virtual Machine Settings
- Step 5: Starting the Virtual Machine
- Step 6: Install Hyper-V Integration Services (if needed)
- Additional tips

Overview

Deploying a VHDX on Microsoft Hyper-V involves creating a virtual machine (VM) and attaching the VHDX file to it. Below are step-by-step instructions to help you through the process.

Step 1: Download the VHDX File

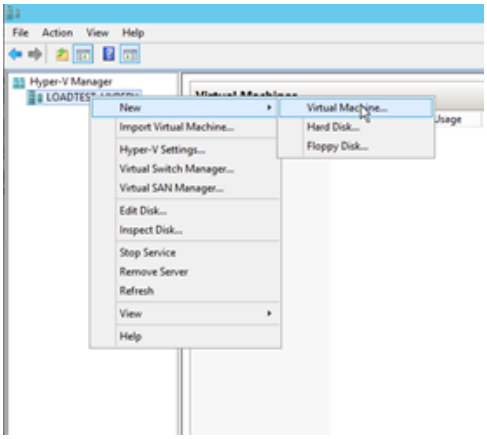
Download the appropriate VHDX file. (Please note that to access the Downloads page, you need to sign in.) Save the VHDX file in a place accessible from your Hyper-V host

Step 2: Open Hyper-V Manager

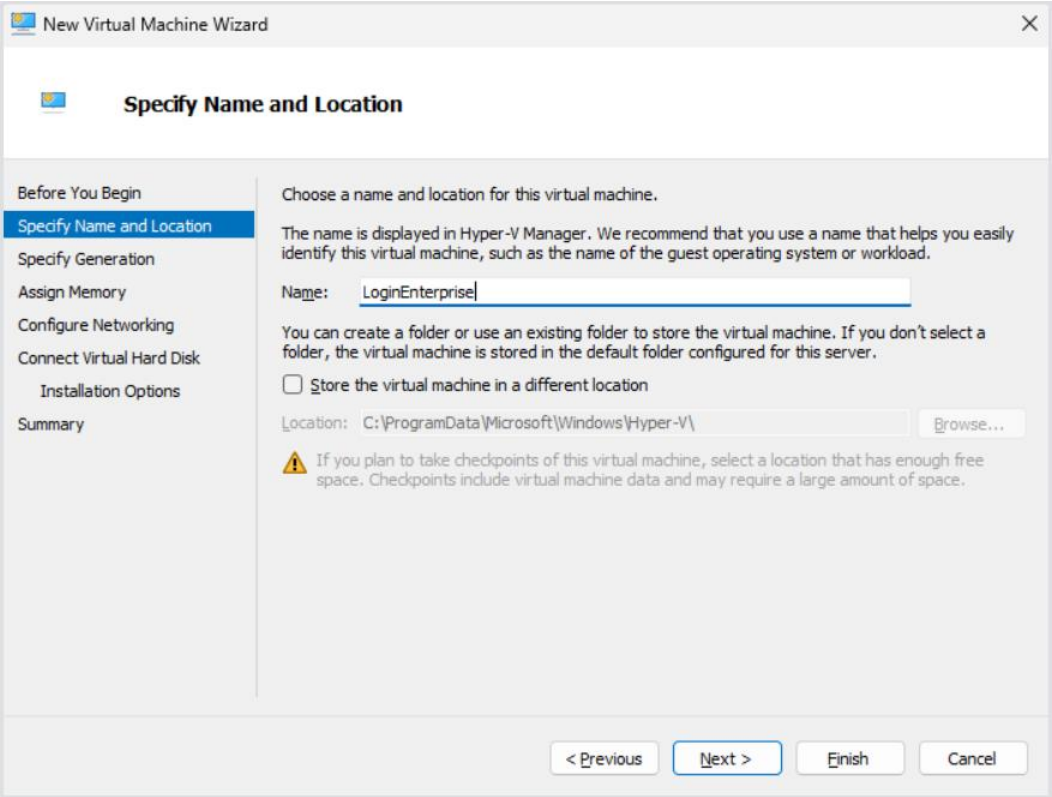
Launch Hyper-V Manager: Open Hyper-V Manager on your Windows server. You can find it by searching in the Start menu.

Step 3: Creating a New Virtual Machine

1. **Start the New Virtual Machine Wizard**: In the **Hyper-V Manager**, right-click on your hostname in the left-hand pane, then select **New > Virtual Machine**. This will open the New Virtual Machine Wizard.

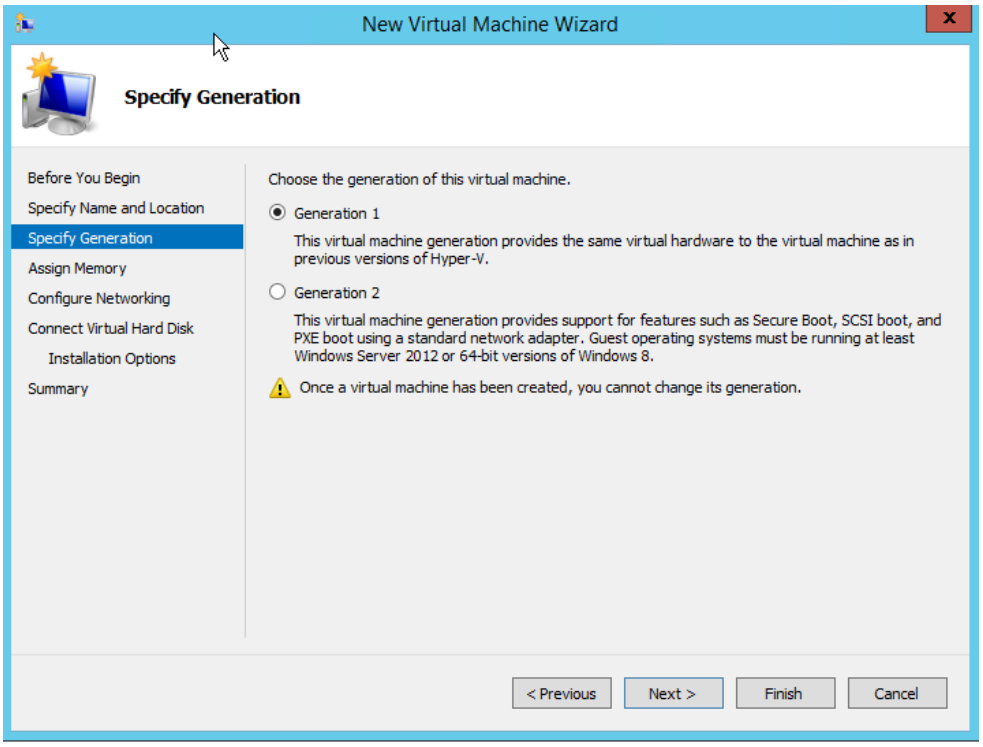


2. **Specify Name and Location**:
    1. **Name**: Enter a name for your Virtual Machine.
    2. **Location**: Choose the folder where you want to store the VM configuration files. Click **Next**.

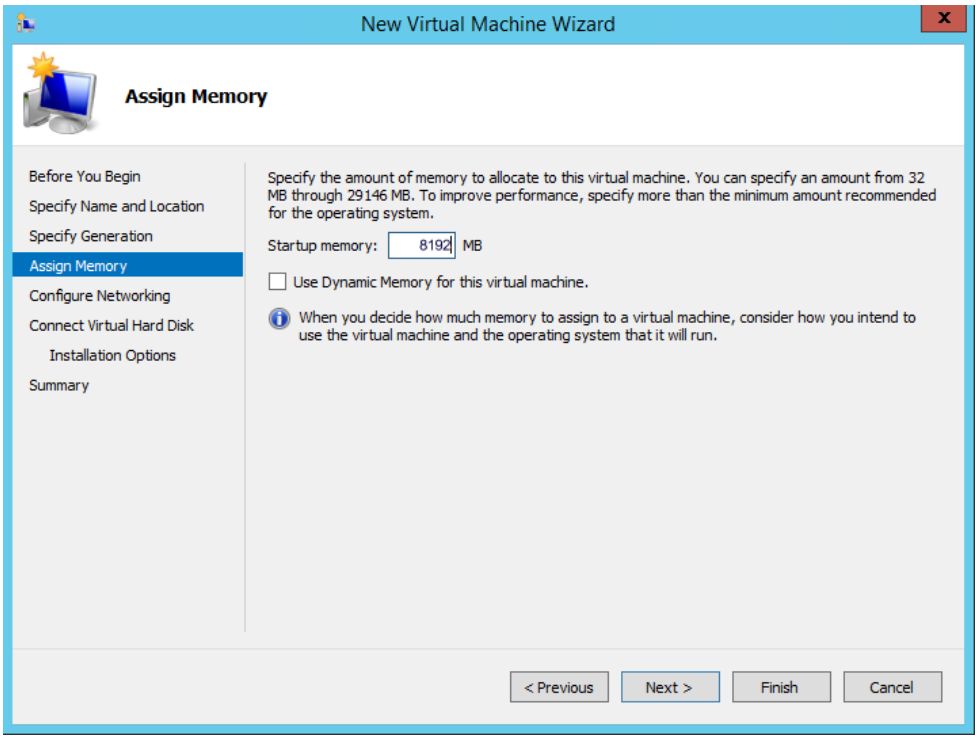


3. **Specify Generation**: Select **Generation 1**, and click **Next**.

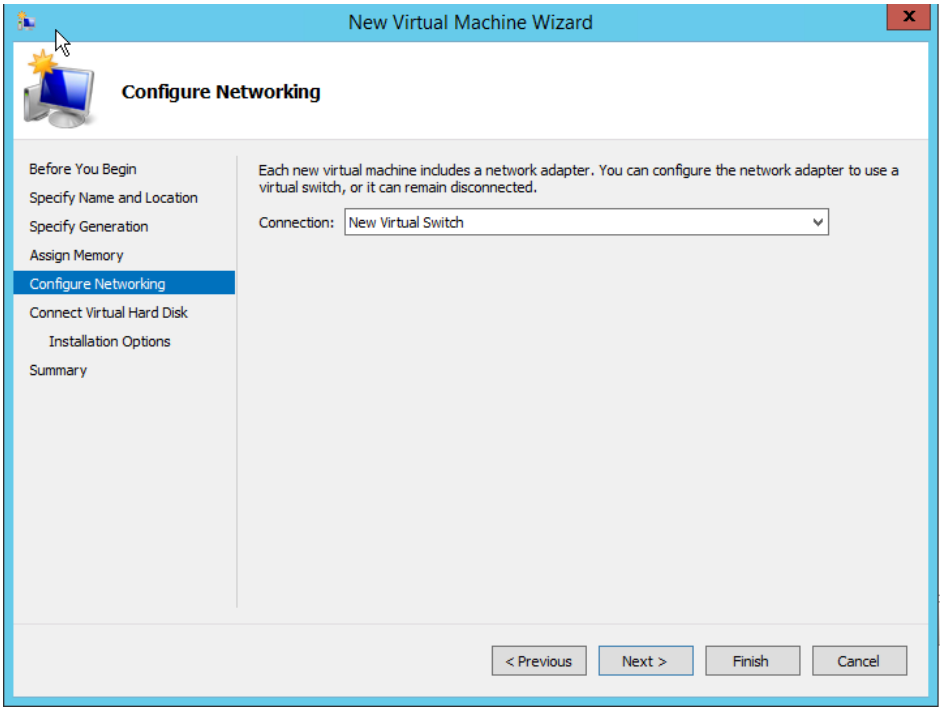
NOTE: Generation 1 uses legacy BIOS-based boot architecture, making it compatible with older systems and software. In Azure, it is suitable for workloads requiring legacy configurations but lacks faster boot times and enhanced security features. Ideal for infrastructures where UEFI support is unnecessary.



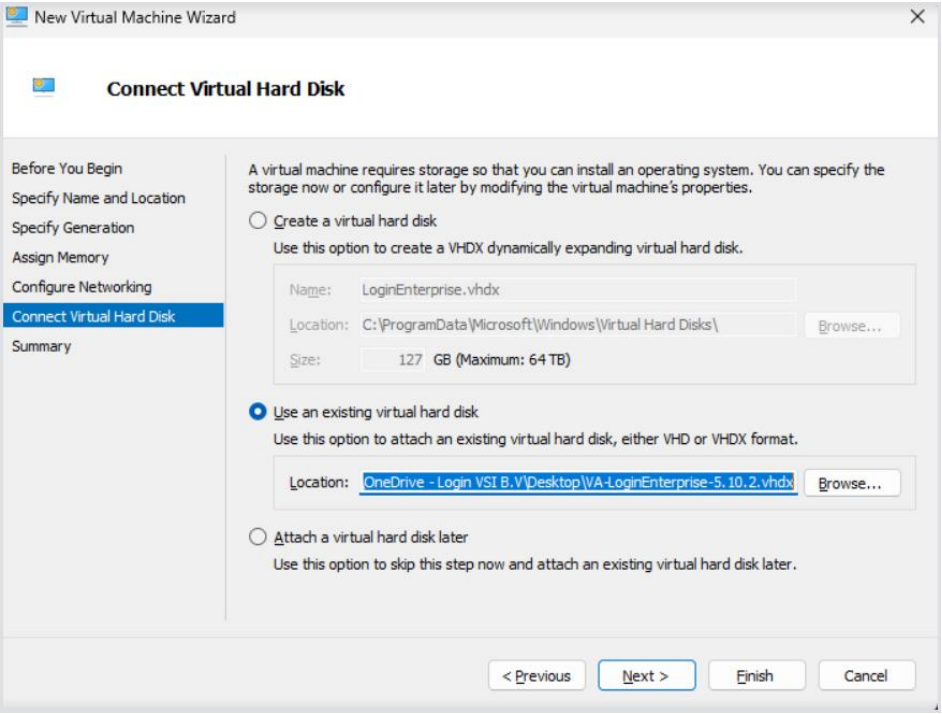
4. **Assign Memory**:
    1. Specify the amount of startup memory for the VM. For example, enter **8192** MB for **8** GB of RAM.
    2. Optionally, enable Dynamic Memory if required. Click **Next**.



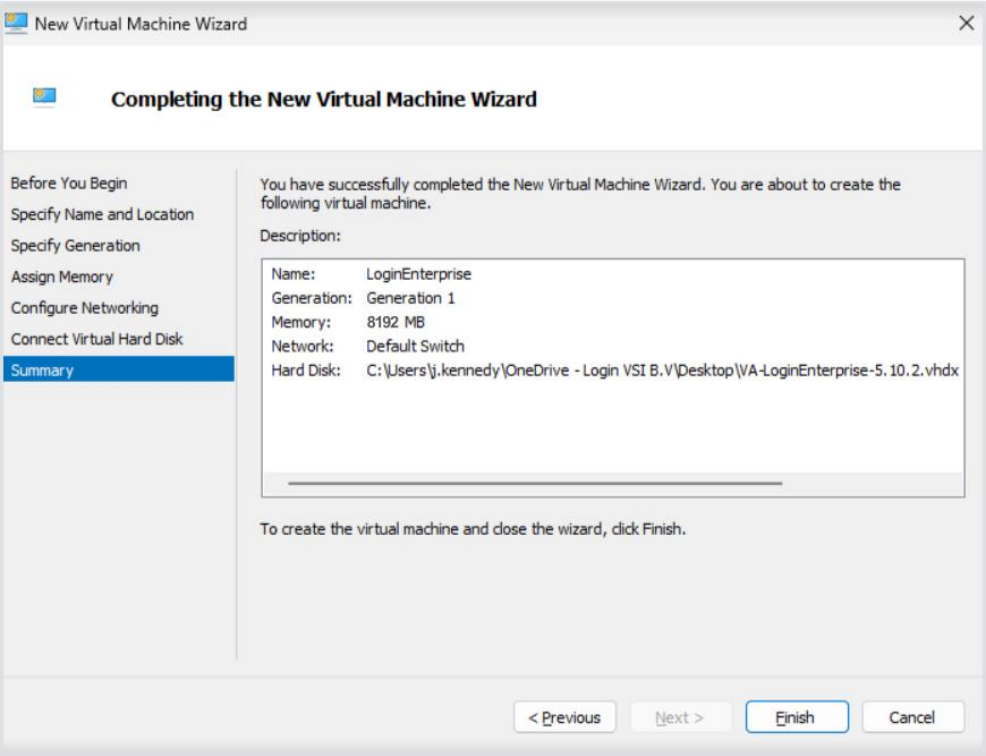
5. **Configure Networking**: Choose a virtual switch for the VM to connect to. If you don't have one configured, you can set it up later. Click **Next**.



6. **Connect Virtual Hard Disk**:
    1. Select **Use an existing virtual hard disk**.
    2. Browse and select your VHDX file. Click **Next**.

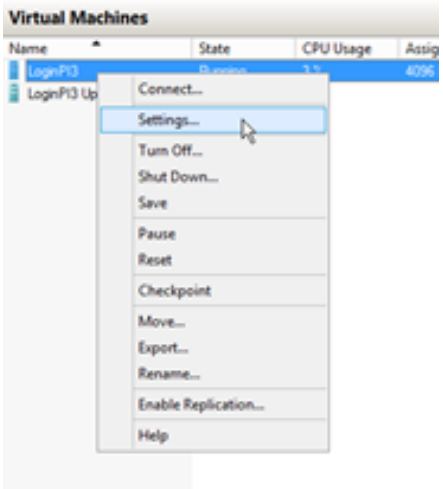


7. **Summary**: Review your settings and click **Finish** to create the virtual machine.

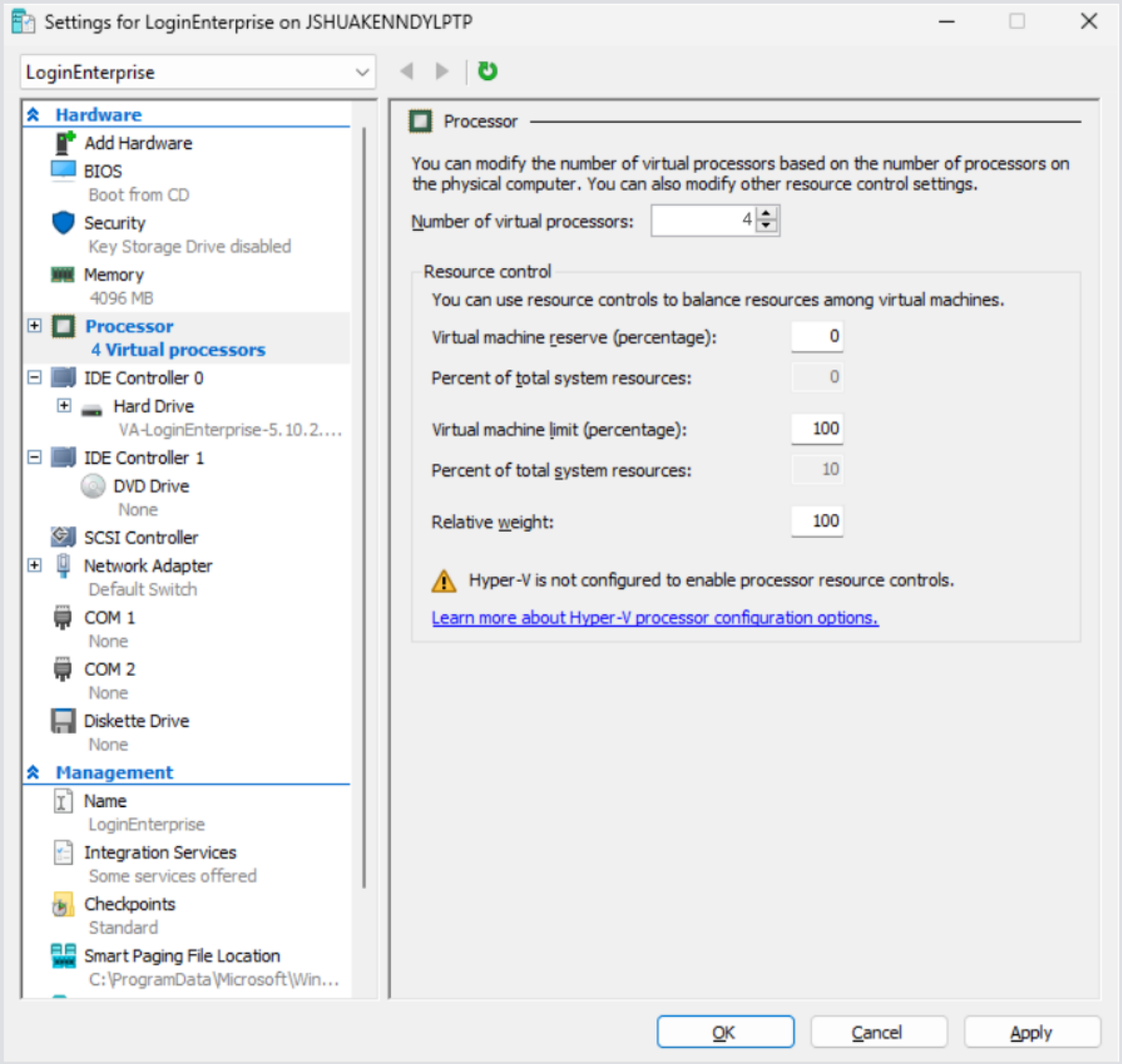


Step 4: Configuring Virtual Machine Settings

1. **Adjust VM Settings**: Right-click the newly created VM in Hyper-V Manager and select **Settings**.



2. Go to the **Processor** settings and scale the **number of virtual processors** to a minimum of **4**.
3. Click **OK**, and start your Virtual Machine.



Step 5: Starting the Virtual Machine

1. **Start the VM**: Right-click the VM and select **Start**.
2. **Connect to the VM**: Right-click the VM again and select **Connect** to open the Virtual Machine Connection window.
3. **Complete the Setup**: Follow the on-screen instructions to complete the setup of the operating system inside the VM.

Step 6: Install Hyper-V Integration Services (if needed)

1. **Install Integration Services**: If the guest operating system doesn't have Hyper-V Integration Services installed, you may need to install it. This enhances the performance and compatibility of the VM.

- In the Virtual Machine Connection window, go to **Action > Insert Integration Services Setup Disk**
- Follow the prompts inside the VM to install the services.

Additional tips

- **Networking Configuration**: Make sure to configure your virtual network settings to allow the VM to communicate with other machines on your network.
- **Backup**: Regularly back up your VHDX files and VM configuration to prevent data loss.

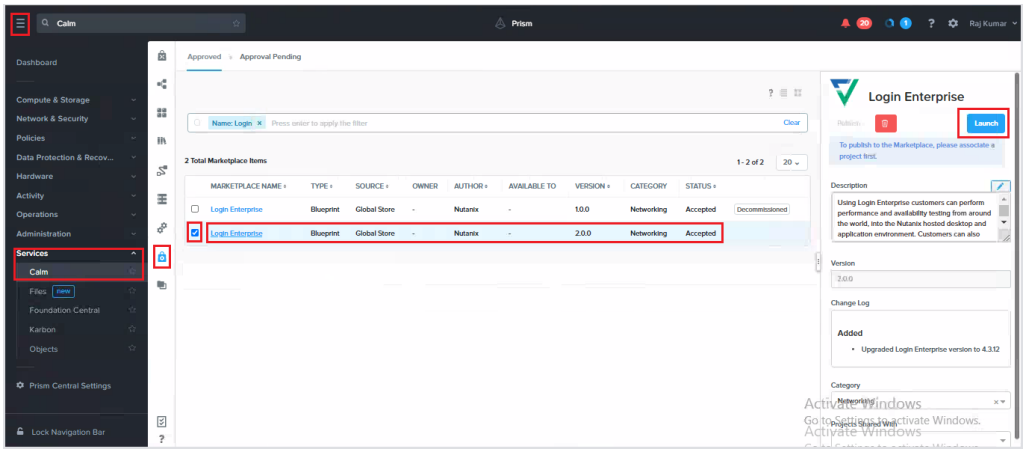
By following these instructions, you should be able to successfully deploy a VHDX file on Microsoft Hyper-V. If you encounter any specific issues, please refer to the Hyper-V documentation or contact us.

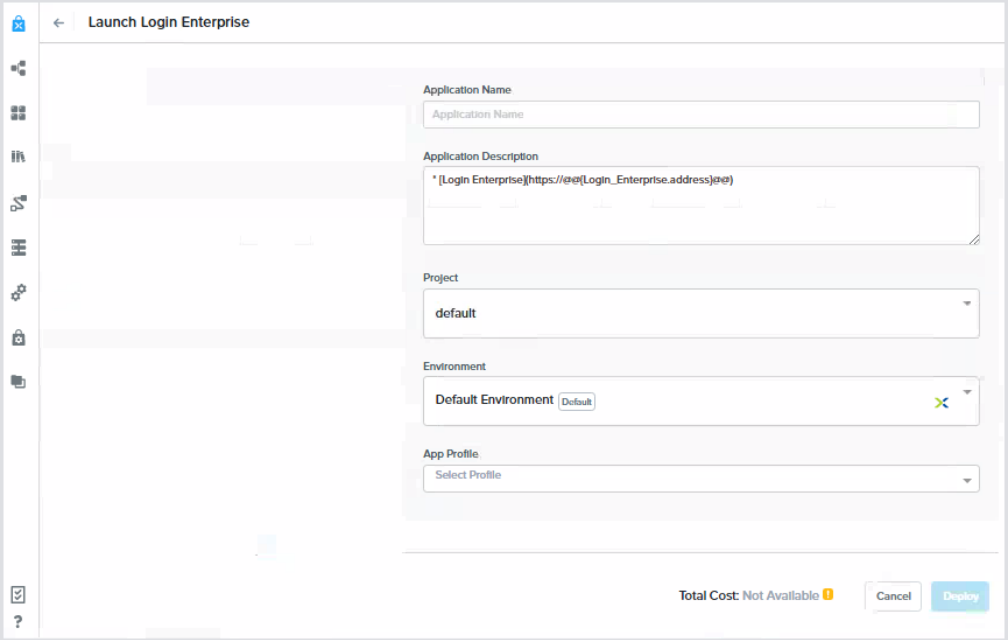
Deploying the OVA to Nutanix CALM Marketplace

This section covers how to deploy the Login Enterprise Virtual Appliance in Nutanix's Acropolis (AHV) via Nutanix's CALM Marketplace.

- To learn more about **Marketplace**, see Marketplace.
- For information on how to **Launch Blueprint** from the Marketplace, see Launching a Blueprint from the Marketplace.

The following screenshots show the **"Login Enterprise"** in **"Market place Manager"** Searched for and Selected. After, click **Launch**, which will navigate to the Launch page, then follow the steps in Launching a Blueprint from the Marketplace.





Deploying the OVA to Nutanix AHV

- Overview
- Prerequisites
- Deployment steps
  - Upload from a local file
  - Upload via URL
- Conclusion

Overview

This section walks you through deploying the Login Enterprise Virtual Appliance (OVA) into a Nutanix AHV environment. By running Login Enterprise on Nutanix, you combine Nutanix's enterprise-grade infrastructure, high availability, rapid scalability, and simplified management with Login Enterprise's powerful endpoint performance testing and automation.

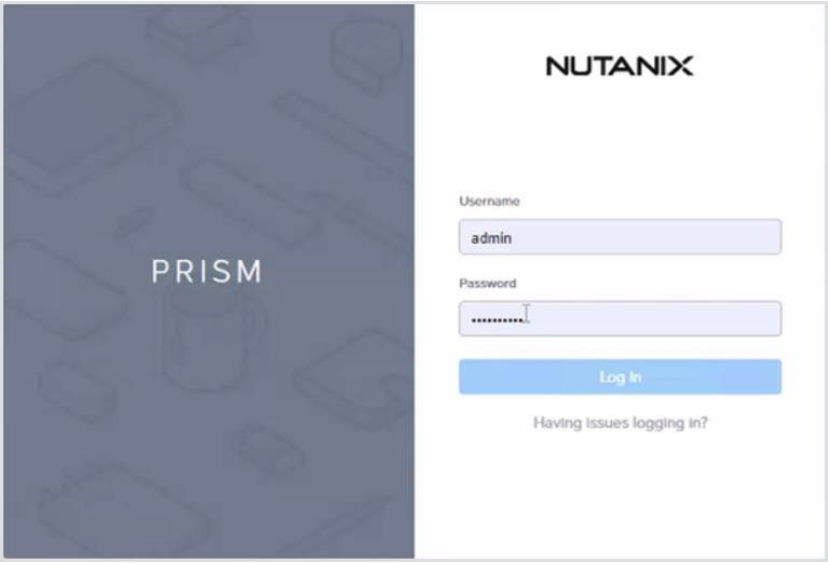
Prerequisites

Before you begin, ensure you have:

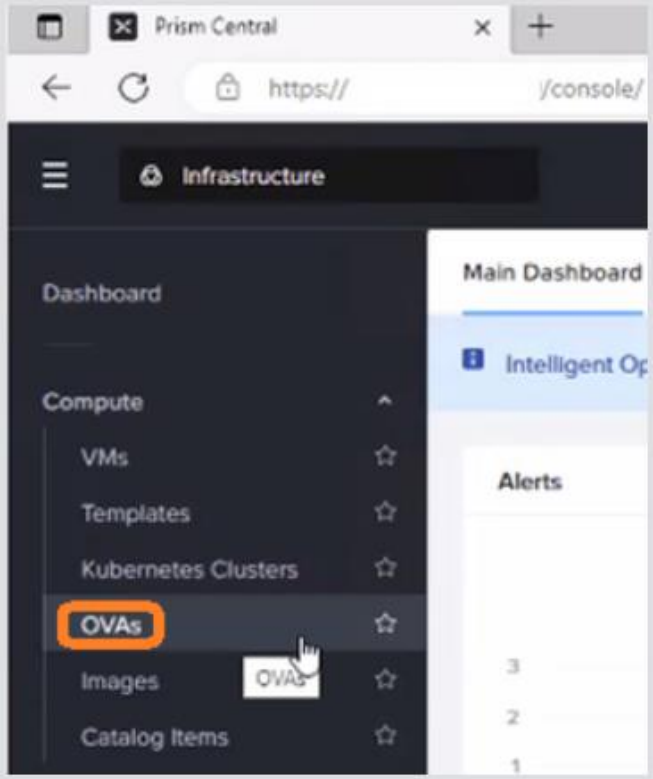
1. Nutanix Prism Central access and an AHV cluster are already configured.
2. Administrator privileges in Prism Central to upload and deploy OVAs.
3. Subnet defined for the VM's network interface.
4. Login Enterprise OVA download URL (or the .ova file downloaded locally). Refer to Downloads.

Deployment Steps

1. Log in to Prism Central with administrator credentials.



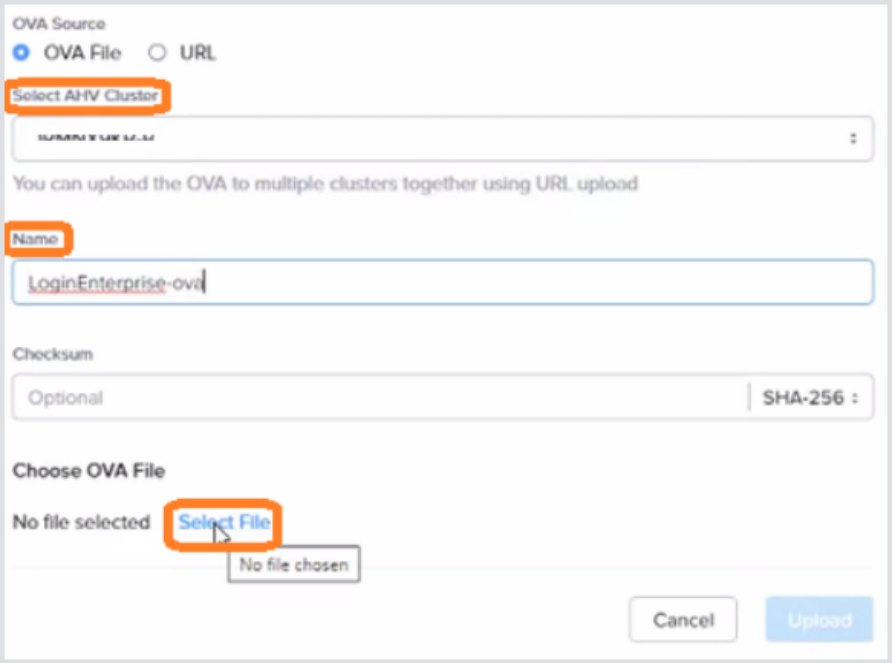
2. From the main menu, select **OVAs**.



3. Click **Upload OVA** to start the upload. You can upload from a local file or upload via URL.

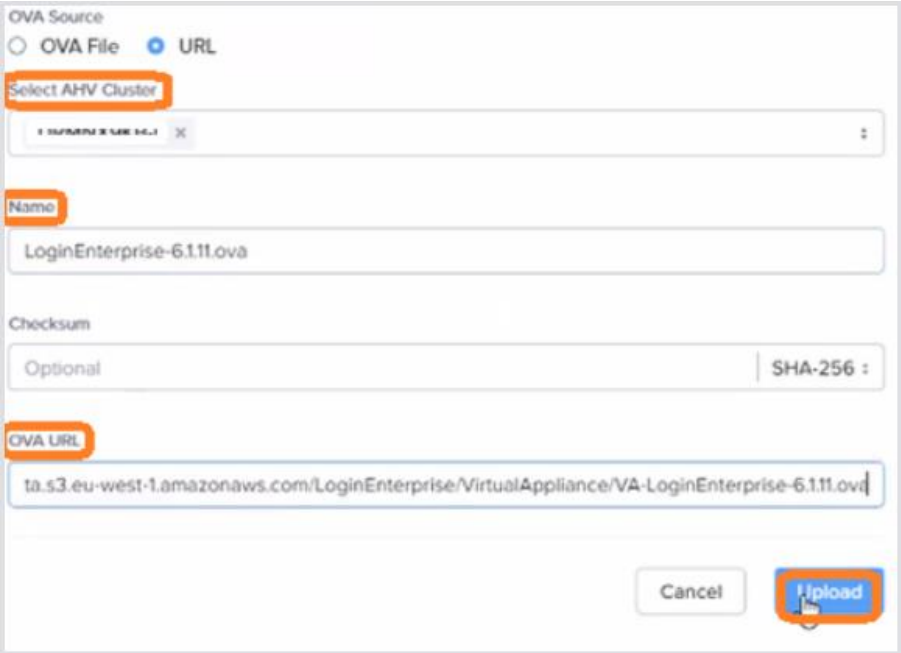
Upload from a Local File

1. In the OVA Source dialog, ensure **OVA File** is selected.
2. (If required), choose the target AHV cluster.
3. (If required) Enter a **Name** for the OVA.
4. Click **Select File**, browse to your downloaded Login Enterprise .ova, and open it.
5. Monitor the upload progress until complete.

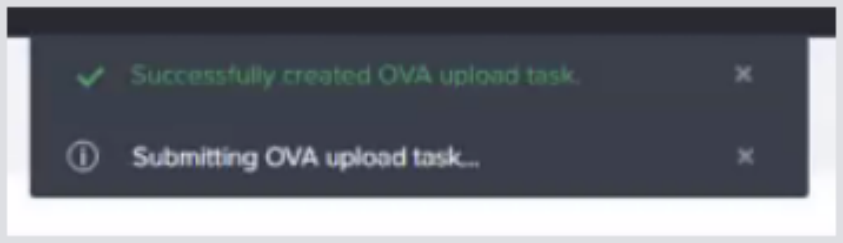


Upload via URL

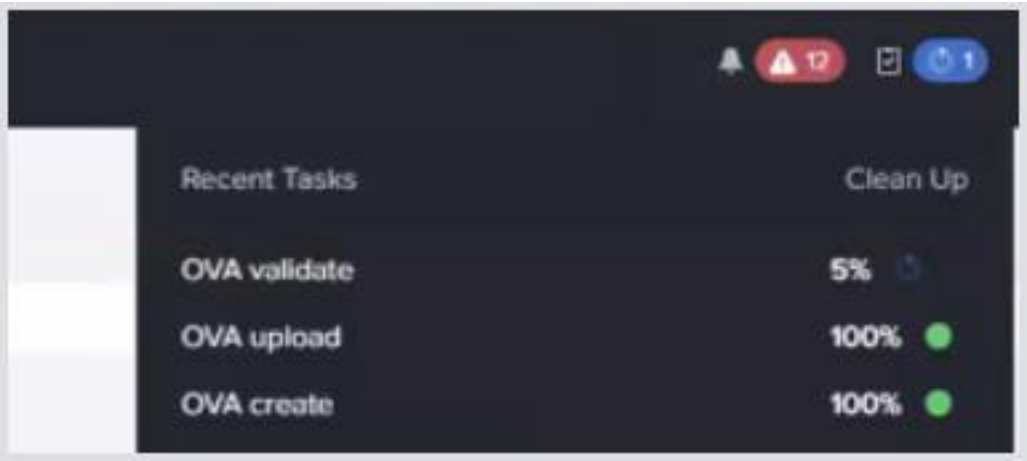
1. In the **OVA Source** dialog, select **URL**.
2. Enter a **Name** for the OVA.
3. Paste the Login Enterprise OVA download URL. The .ova URL can be retrieved from Downloads, under the Nutanix download (get the URL from the Download button for it).



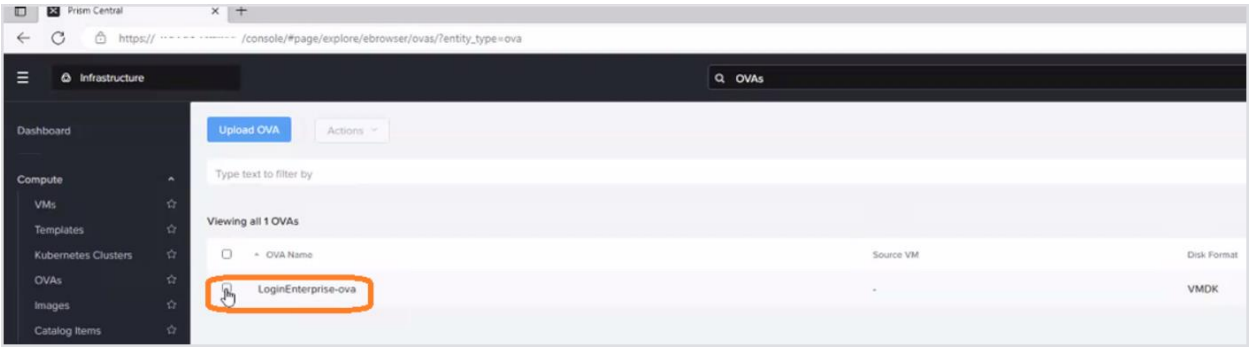
4. Click **Upload** and monitor the task creation.



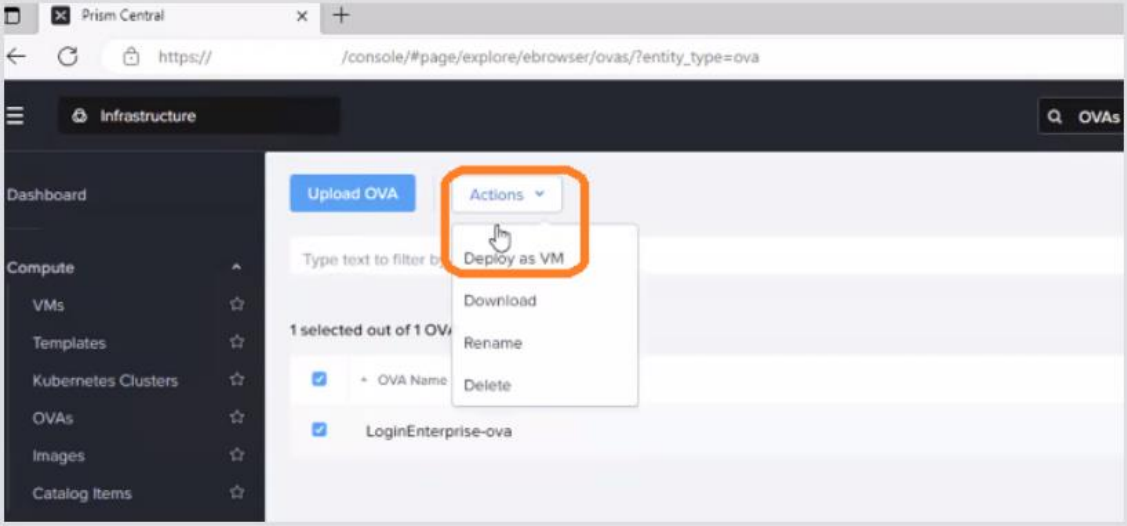
4. **Confirm OVA availability**.
    1. Watch the real-time updates (upper right).



2. Once uploaded, verify the OVA appears on the **OVAs** page.

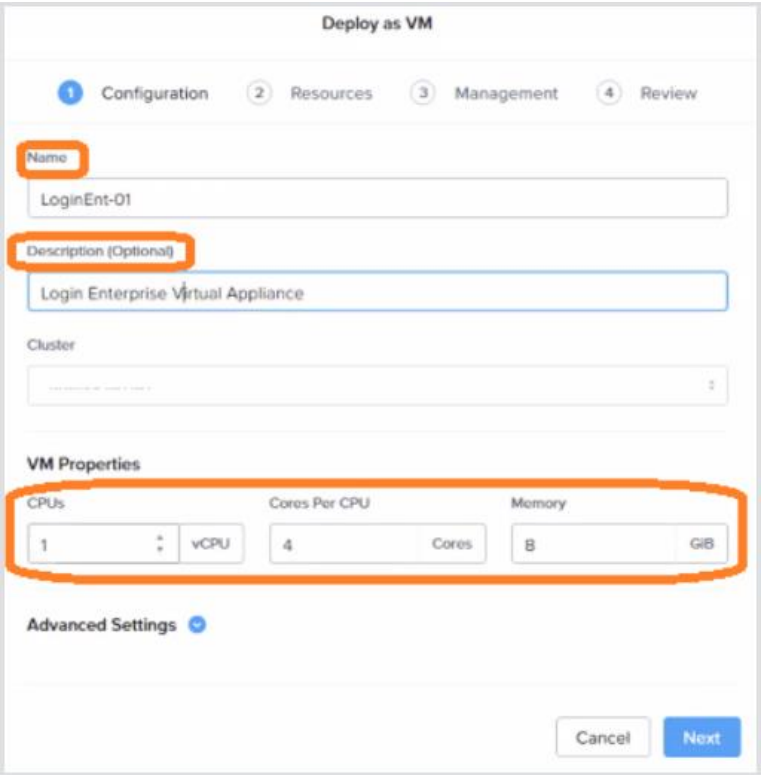


5. **Deploy as a VM**.
    1. Select the new OVA, click **Actions > Deploy as VM**.



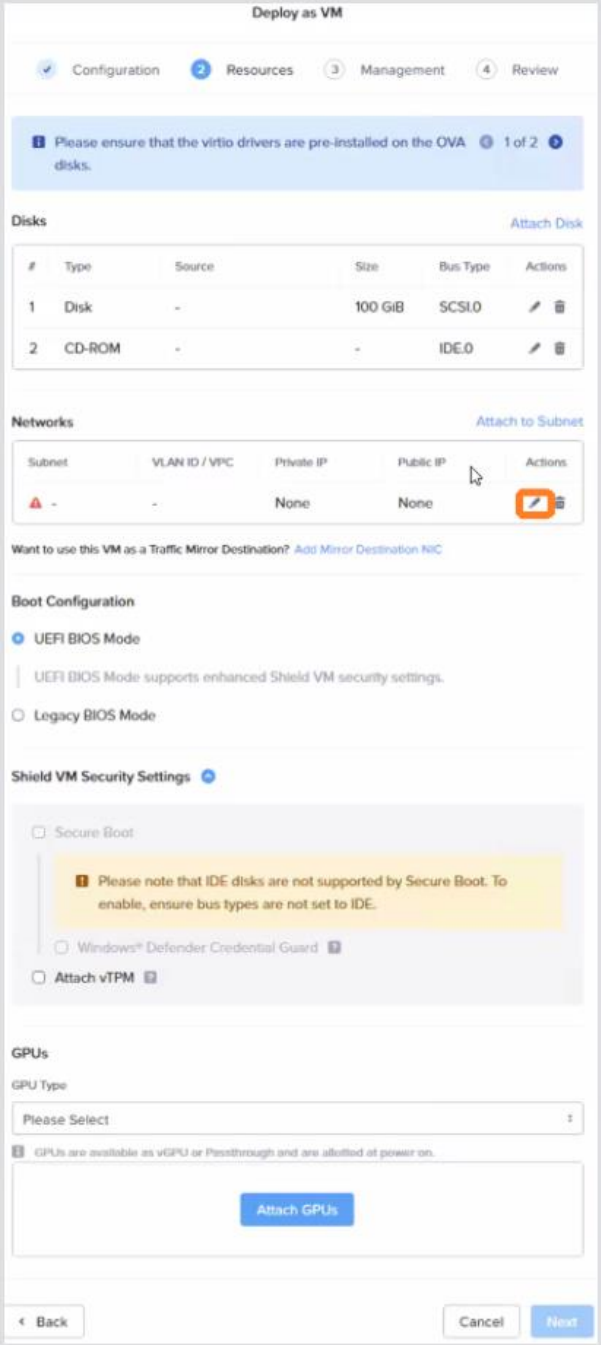
2. In **Deploy as VM**:

Field	Description
Name	Enter your desired VM name.
Description	Optionally, describe the VM.
CPU & Memory	Default (1 vCPU × 4 cores, 8 GB) is sufficient; adjust later if needed.

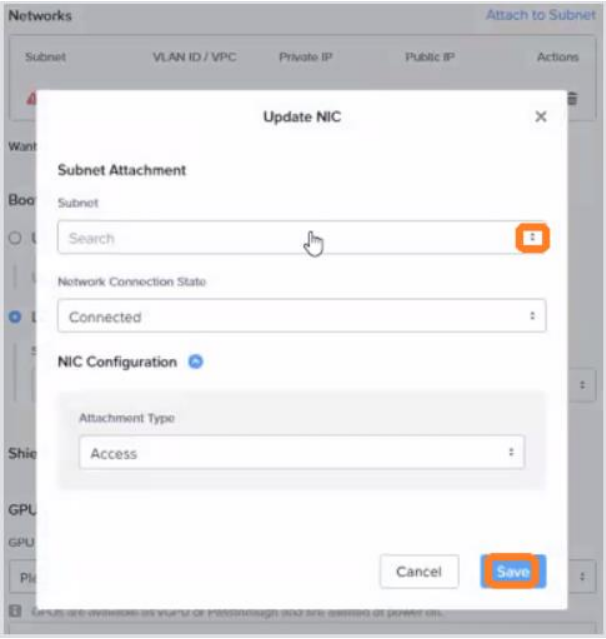


3. Click **Next**.

6. **Configure Networking**.
   1. Under **Resources**, click the pencil icon next to **Networks**.

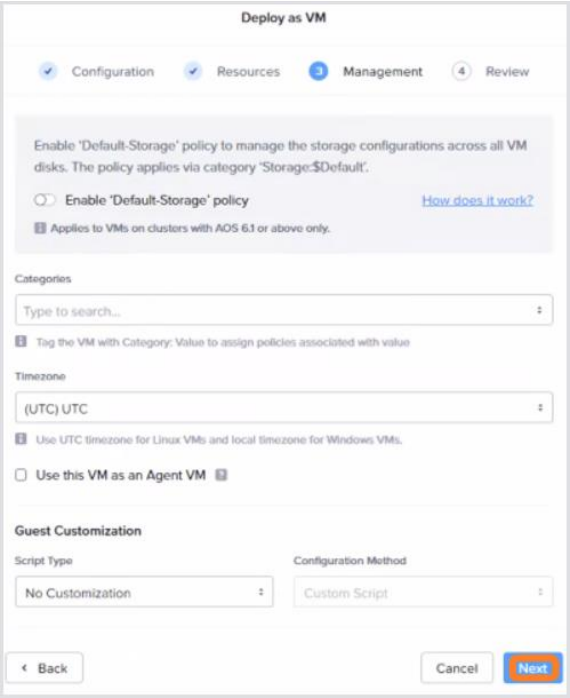


   2. In **Update NIC**, choose the appropriate **Subnet**.



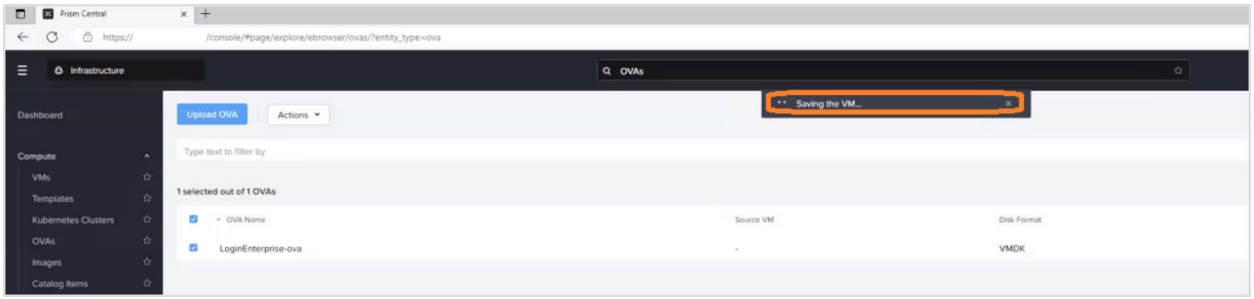
3. Click **Save** and then **Next**.

7. In **Resources**, under **Boot Configuration**, select **Legacy BIOS Mode**.
8. **Finalize Management Settings**.
    1. Review any management options; leave defaults unless changes are required.
    2. Click **Next**.

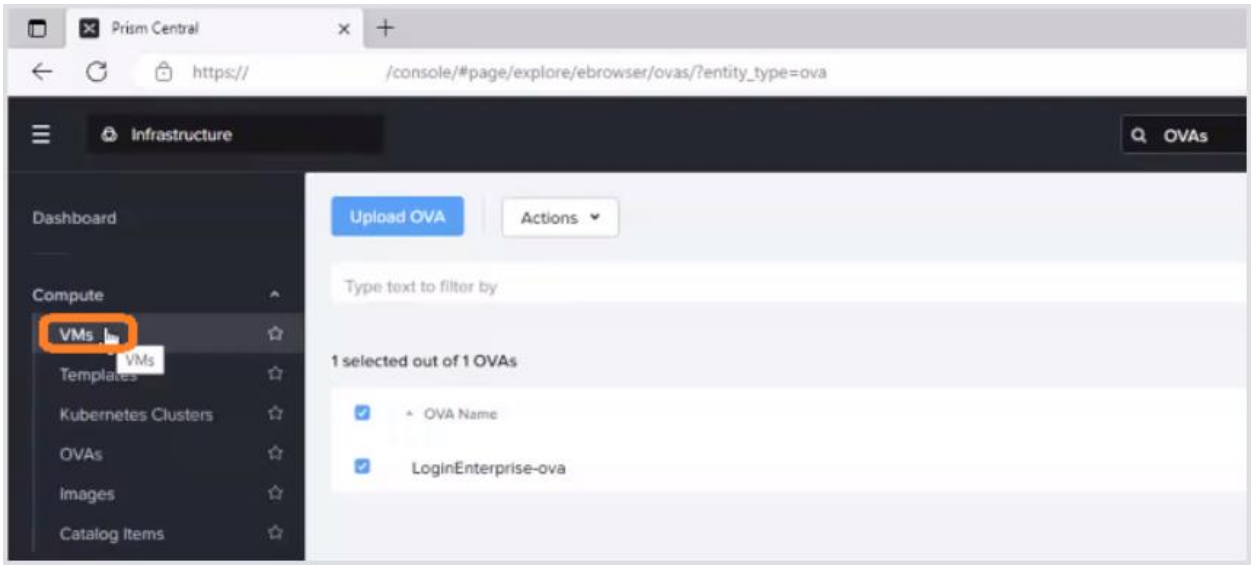


9. **Review & Deploy**.

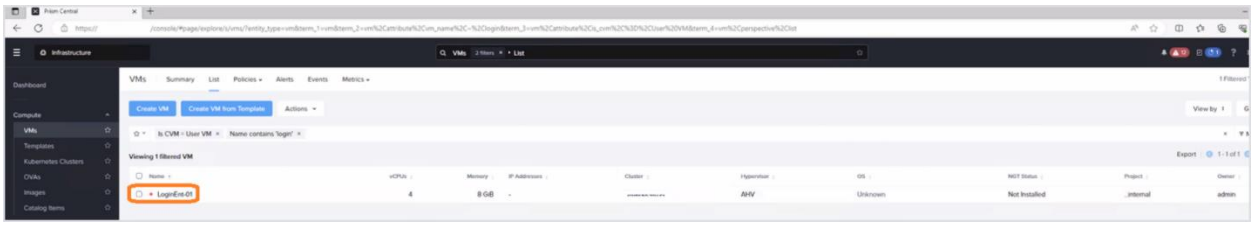
1. On the **Review** page, confirm your settings.
2. Click **Finish**.
3. Observe the **Saving the VM…** popup.



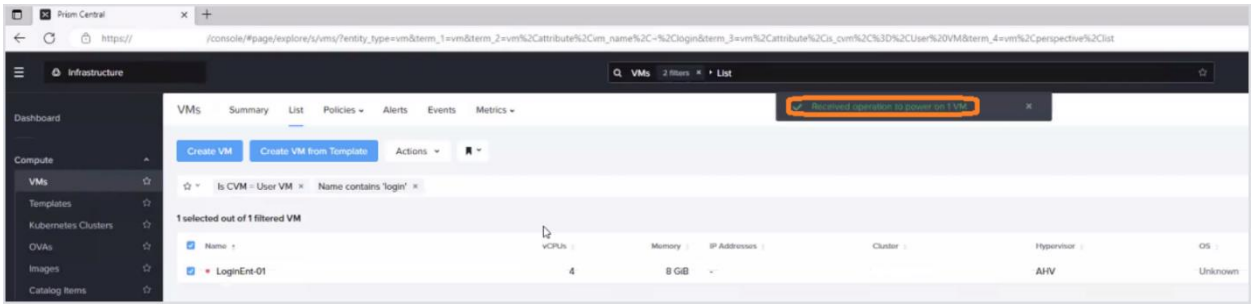
10. **Power on the VM**.



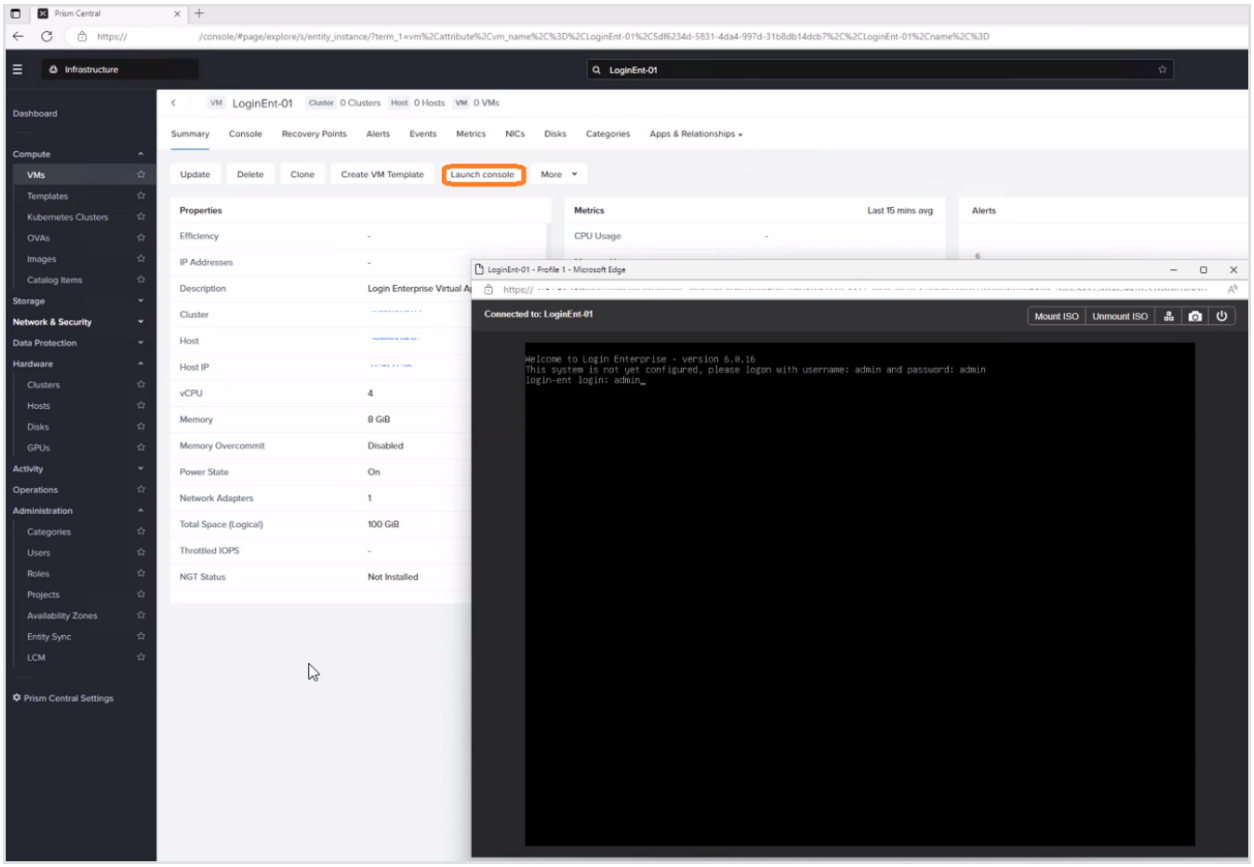
1. Go to **VMs**, select your new **Login Enterprise VM**.



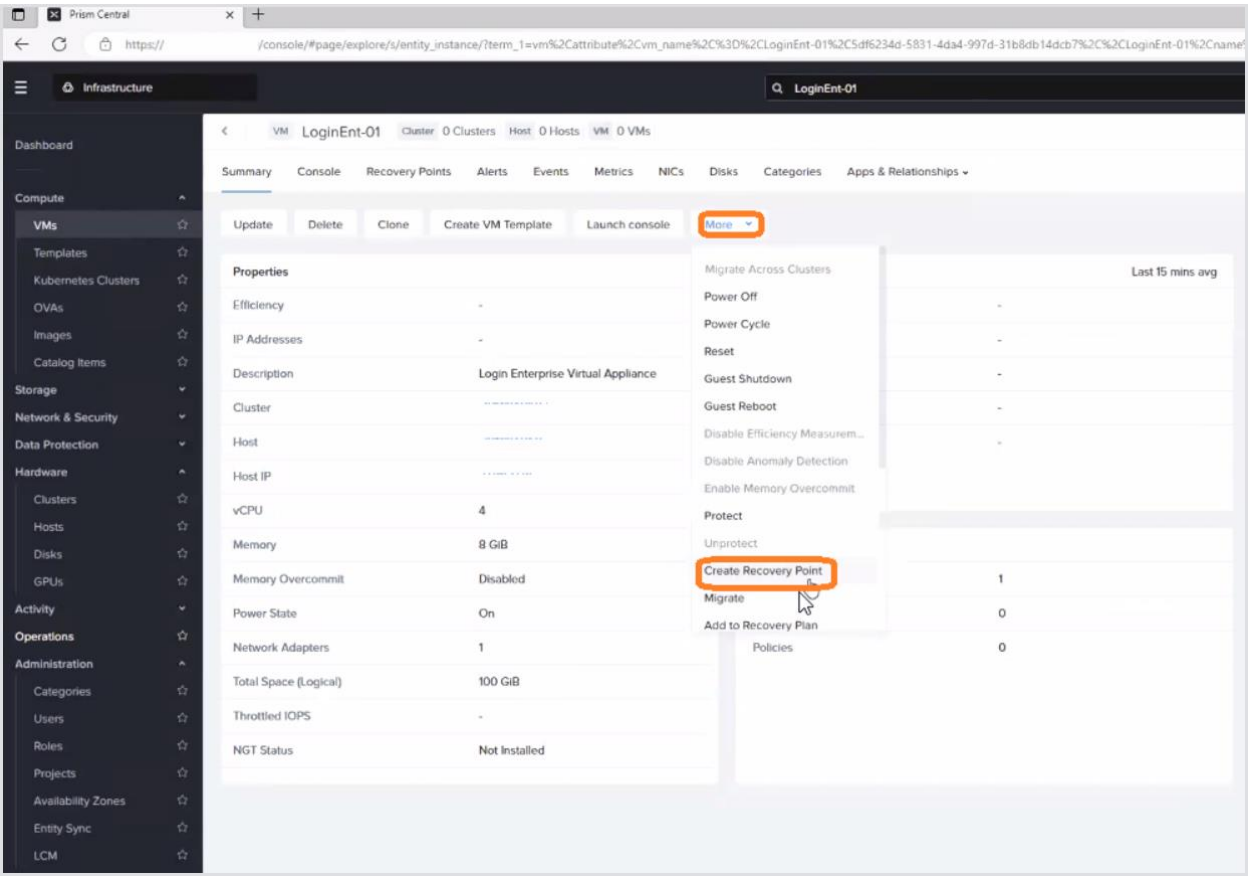
2. Click **Actions > Power On**.
3. Wait for the "Received operation to power on 1 VM…" notification.



10. **Access the Console**.
    1. Select the VM, click **Launch Console**.
    2. Verify that the Login Enterprise setup interface appears.



12. **(Best Practice)**. After initial setup, and before/after any Login Enterprise upgrades or maintenance, create a Nutanix recovery point via the **VMs > [Your VM] > More > Create Recovery Point**.



Conclusion

Once the appliance is running, you can begin configuring Login Enterprise tests directly within your Nutanix-hosted VM, leveraging both platforms for seamless performance monitoring and automation. For next steps, see [Initial Appliance Configuration](#).

Installation Walkthrough (Quick Start)

- Overview
- Pre-installation Checklist
- Step 1: Deploy the Appliance
- Step 2: Upload Certificates
- Step 3: Configure Active Directory
  - Create Admin and Read-Only User Groups
  - Create Test Accounts
  - Configure Test Accounts
- Step 4: Configure the Appliance
  - Licensing
  - LDAP
    - Set up LDAP Connection
    - Assign Roles with LDAP Groups or Users
  - E-Mail
  - Accounts
- Step 5: Build a Launcher
- Step 6: Configure End-User VM Pool
- Step 7: Create and Execute a Test
  - Option 1: Run a Test Using the Desktop Connector
  - Option 2: Configure and Run a Test Using Microsoft RDS
- Step 8: View Results

Overview

This section is designed to help you quickly get started with your new Appliance, ensuring you can set it up and run a simple Test to begin experiencing its value. The steps outlined below will guide you through the essential processes required to deploy and configure the Appliance, enabling you to start using it.

NOTE: This is an overview of the basic setup steps. For more in-depth information and advanced configuration options, we encourage you to take the introductory training lab and explore the user interface after completing the initial setup.

By following this walkthrough, you'll be able to deploy the Appliance, configure key settings, create and execute a Test, and view the Test results.

Pre-installation Checklist

Your Login Enterprise system will be composed of the following elements:

- Login Enterprise Appliance VM: You will deploy this VM from our VM template
- Launcher VM: a Windows machine that will be used to originate user sessions
- End-user VM pool: a set of machines that will host your Test sessions

- A domain user group for managing appliance access
- A set of domain user accounts to host the Tests

To deploy the Appliance, you will need the following information:

- FQDN for the Appliance
- IP address and network configuration for the Appliance (or a decision to use DHCP-assigned addressing)
- Certificate for the FQDN (Optional)
- Password to use for the admin account
- Information about an MS SQL Server to host your data (Optional)
- Domain controller information (Optional)

Login Enterprise Appliance hostname:	
Login Enterprise Appliance domain name:	
Login Enterprise Appliance IP address*:	
Login Enterprise Appliance Subnet + Mask*:	
Login Enterprise Appliance Default Gateway*:	
Login Enterprise Appliance DNS Server(s)*:	
Login Enterprise Appliance DNS suffix*:	
Login Enterprise Appliance admin password:	
MS SQL Server instance**:	
Launcher Windows VM:	
Connection Broker:	
Resource Pool:	
Test Accounts/Name Pattern:	
Domain Controller Address***:	
Read-only AD User Group***:	
Admin AD User Group***:	

*** Optional, only necessary if not using DHCP for network configuration.**

**** Optional, only necessary if choosing an external database.**

***** Optional, only if you are enabling LDAP logins to the Appliance UI**

Step 1: Deploy the Appliance

Please follow the platform-specific deployment instructions in Deploying the Appliance, and then continue with the initial configuration.

Use the VM settings in our template when you import the appliance VM, including 4 CPUs and 8GB RAM. The VM will automatically provision as a 100GB boot/data disk.

NOTE: Azure, by default, provisions a 32GB boot disk. This is likely too small for a production system with an Internal Database, so you may need to expand the virtual disk and the file system. You can expand the disk in the future if you do not want to do it at the initial setup.

Use your hypervisor's normal process to import the template. The deployment process of all possible hypervisors is beyond the scope of this document. You must connect the VM to an appropriate subnet and make other configuration choices.

Power on the VM. Eventually, you will see this login prompt:

```
Welcome to Login Enterprise - version 5.10.2
This system is not yet configured, please logon with username: admin and password: admin
login-ent login:
```

As it says, log in using the default credentials: username: `admin`, password: `admin`.

Provide the network configuration in the next few prompts: computer name (also called the hostname), domain, and network configuration. Select `d` for DHCP or `s` for static. The example below shows a Static network configuration.

```
Starting LoginVSI configuration tool...
Welcome to Login Enterprise - version 5.10.2
 Please help answer the following questions to get the system setup...
 Please enter your computername (e.g login-ent):login-ent
 Please enter your domain dns suffix (e.g contoso.com):loginvsi.lab
 Please choose a network configuration mode dhcp or static (D/s) s
Configuring network using static address...
 Please provide the new ip address (192.168.7.144): 192.168.3.10
 Please provide the netmask (255.255.255.0): 255.255.255.0
 Please provide the default gateway (192.168.7.1): 192.168.3.1
 Please provide the nameservers (run 192.168.7.1) 10.0.0.10
 Please provide the dns search suffix (loginvsi.lab) loginvsi.lab
```

NOTE: If your subnet contains DHCP, your Appliance will automatically acquire a DHCP address initially. You can even use SSH to log in instead of the console (using the default admin/admin credentials) to perform the initial configurations if your subnet has DHCP.

After the network configuration (if any), the system will prompt you to set the name and password of the default administrator account. Then it will automatically reboot. It's normal to preserve the `admin` account name but set a different password.

```
Please enter a name for admin user: admin

Please enter a new password for admin:

Please confirm the new password:

Password updated successfully

Setting fqdn: login-ent.loginvsi.lab
Setting hostname: login-ent
Rebooting...
```

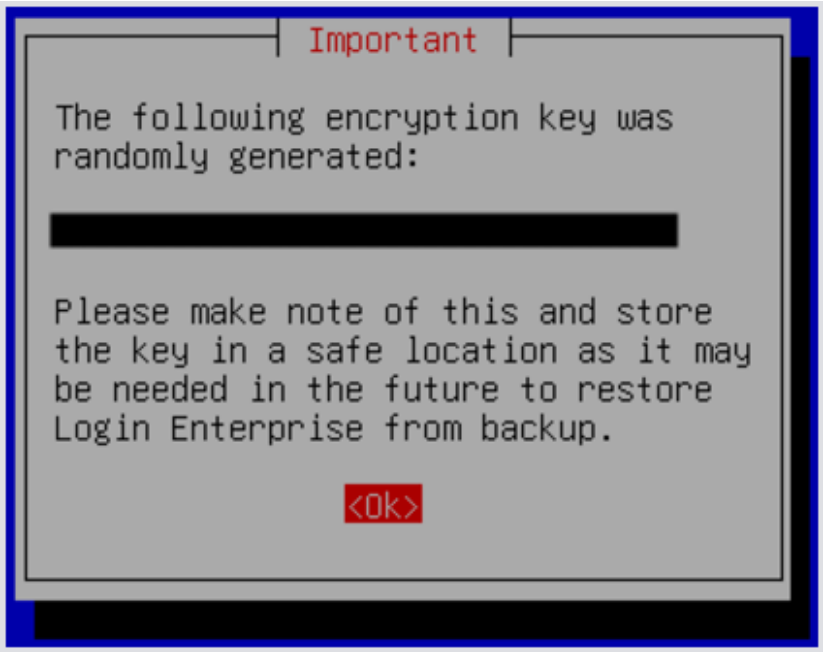
After the reboot, you are prompted to log in with the new administrator credentials you just set:

```
        Welcome to Login Entperise - version 5.10.2
        This appliance is not yet configured!
        Logon with username and the password you set previously

login-ent login: _
```

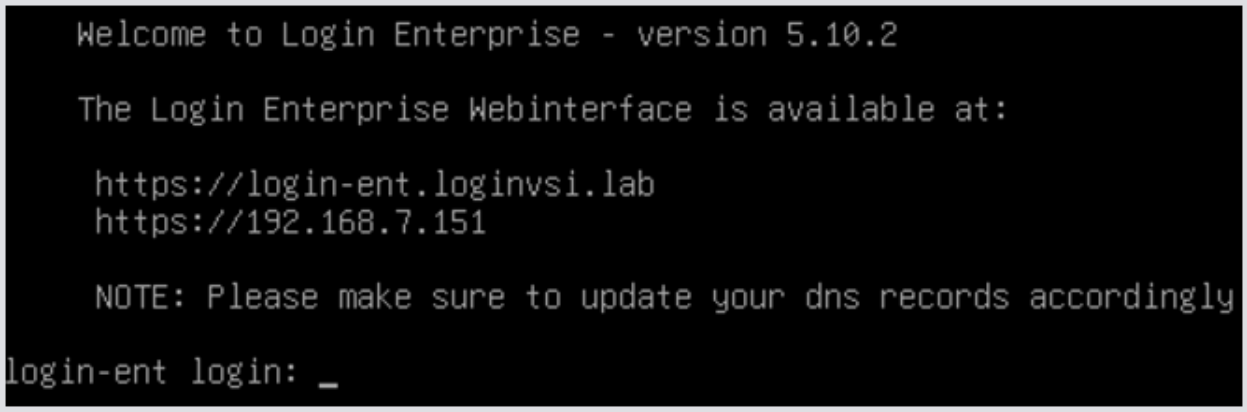
When you do that, the system will perform several automatic configuration operations. It will then prompt you whether to create a new encryption key or enter one. The encryption key is only used to encrypt sensitive data in the configuration database, like passwords. Connections to the Appliance and between the containers are handled over HTTPS with the applied SSL Certificate (self-signed or custom).

```
Login Enterprise Encryption

An encryption key is required to protect
sensitive information within Login
Enterprise.

Please select an option:

   1) Generate new encryption key
   2) Enter encryption key from backup

                <Ok>
```

Normally, you would allow the system to generate a new encryption key. You would only use an existing key if you are trying to reconnect to an existing external MS SQL Server Appliance database that already has its own encryption key, and you do not have access to the original Appliance VM that created the database. Then, you would need to manually type in the existing key. Otherwise, the system will generate a key for you, which you should screen-shot for future reference.



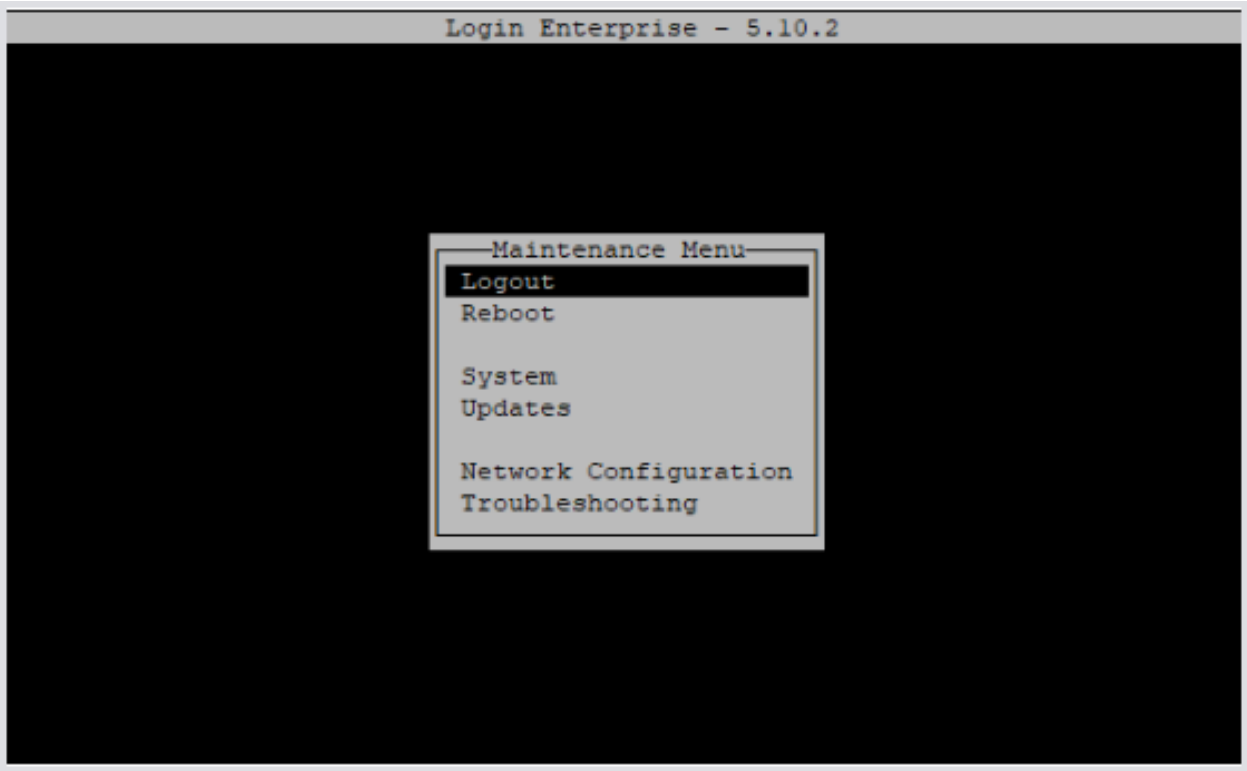
The system will automatically reboot. After it completes, you will see a screen like this on the console informing you that the appliance is up and operational. If you have a certificate for the Appliance, you can add it now. Otherwise, you can leave the console and move to the Appliance web UI.



NOTE: If you have not already added the DNS record for this host to your local DNS server, you should pause to do that now.

Step 2: Upload Certificates

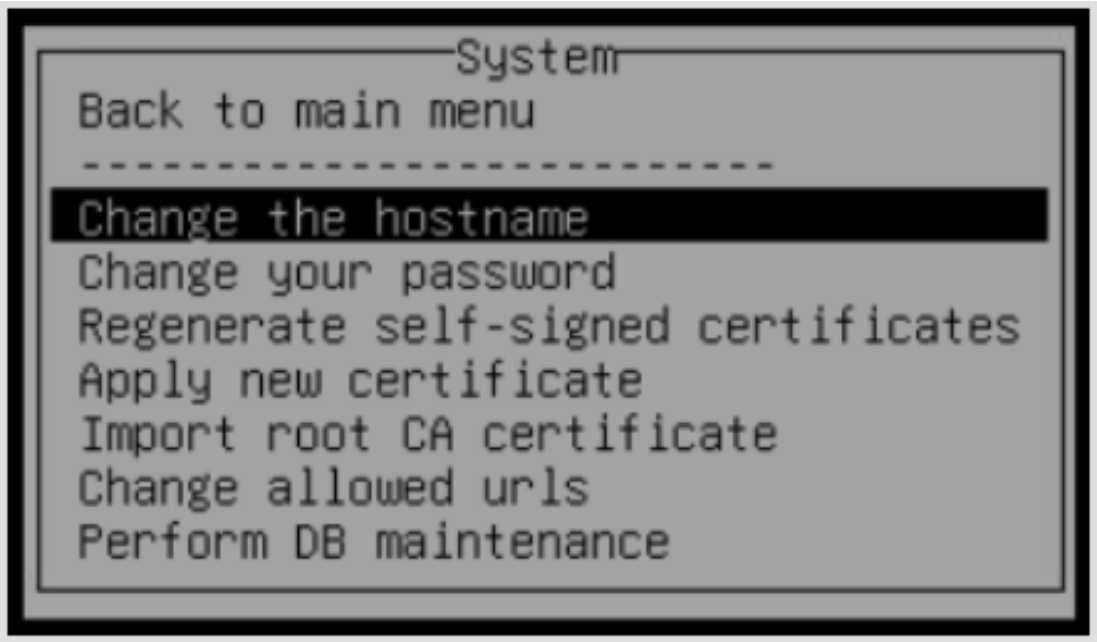
From the VM console or with an SSH client, log in to the Appliance as the local `admin` account. You will see a menu like this:



If you need to readjust your network configuration, you can do that now. Otherwise, the only configuration you are likely to need is uploading your machine certificate and any required root certificates for your domain.

TIP: If you are going to use the self-signed certificate, you can skip the rest of this section. The self-signed certificate has already been generated. Just remember that when you access the Login Enterprise UI, it will show up as untrusted in your browser, and you will have to acknowledge that. To install any certificates, you will need to upload them to the system first. We support the SFTP protocol for transferring files to and from your appliance. WinSCP is a popular Windows tool that supports SFTP.

Using the arrow keys, navigate to the **System** menu entry shown in the screenshot above. Click **Enter**. You will see this menu:



Install your Root CA Certificates first. These are required for the appliance to communicate with any external hosts in your network, but they may also be required for your appliance to talk to itself. We use your server certificate for internal communications as well, so if your server cert comes from your local CA, we won't be able to use your cert internally until the root CA certificate is imported.

Using your SFTP client, upload your root and any intermediate certificates to a directory on the appliance. `/certificates` is the normal folder to put these into. Select **Import root CA certificate** from the above menu. It will prompt you with:

```
Please enter the path to the root ca file (.crt):
```

Enter the full path of the file you uploaded, e.g.,`/certificates/myrootca.crt`. Click **Enter**. The Login Enterprise system will offer to restart your services. If you have multiple certificates to upload, don't restart the services until you have uploaded each of them. It will save time.

Next, upload the server certificate to the Appliance. Use the same directory, /certificates. Then, select Apply new certificate from the above menu. It will prompt you for the certificate file location and will then prompt you for the certificate password. Then it will restart the Login Enterprise services again and return you to the menu. Your web services are now using the new certificate.

NOTE: If your Appliance hangs while starting services, especially if it hangs at "10 services" remaining, there is a problem with your certificate, the upstream certificates, the network connection to the CA, or some other issue preventing the system from using the provided certificates. See Managing Certificates for more information or contact us.

Step 3: Configure Active Directory

After building your Appliance, it's worthwhile to set up your directory services to support the operation of Login Enterprise. There are 3 main functions you need to deal with:

1. Create at least one User Group (Optional)
    - You can create one or more Active Directory user groups to organize your test accounts. Alternatively, you can use individual AD accounts without assigning them to a group.
2. Create Test Accounts
3. Configure Test Accounts to run our Login Processor at login

Create Admin and Read-Only User Groups

Login Enterprise supports logging into the Appliance web UI using LDAP/AD accounts. Below, you will configure the LDAP integration by telling the Appliance where your Active Directory (AD) is.

User access is managed using Role-Based Access Control (RBAC). You can connect multiple AD groups and assign custom permission levels to each group. While you may choose to create Admin and Read-Only groups as a starting point, you have full flexibility to define roles and permissions that suit your environment.

In AD, you can either create at least one user group or identify specific AD usernames (in email format) that you want to grant access to the Appliance. These groups or users will later be configured in the Access Control section of the Appliance.

The placement of users in the Active Directory (AD) tree is not a concern for this process; we will query them by the name you give us. In those user groups, you can add AD accounts you want to log in as. You can change the membership of these groups at any time. We verify group membership on every login attempt. (This also means that if your domain controllers are down, you cannot log in as a domain user.)

Create Test Accounts

To perform Tests, the Appliance needs to have a set of AD accounts it can use to run Tests. We use those accounts to log in to your connection broker. The Appliance needs to be told the name and password of every one of these accounts.

It's easiest if you create a sequence of numbered accounts because the Appliance has a way to bulk-create sequential account records like that. So, for instance, you can create `loginvsi0001 .. loginvsi0100` in Active Directory, and it's straightforward to create records for those accounts in Login Enterprise. It's also easiest if you set the passwords to the same.

NOTE: (If you need to have accounts with different naming conventions and different passwords, contact us for a script that can use the Login Enterprise API to upload a list of accounts from a CSV file.)

Place these accounts in an OU specific to them so you can easily adjust permissions and policies.

Configure Test Accounts

We have one requirement for your test accounts: they must run our Login Processor at login. The Login Processor is a program named `LoginPI.Logon.exe`, and you can download it from the appliance itself. At this point, you don't need to download the program, but you do need to decide where and how you want to execute it.

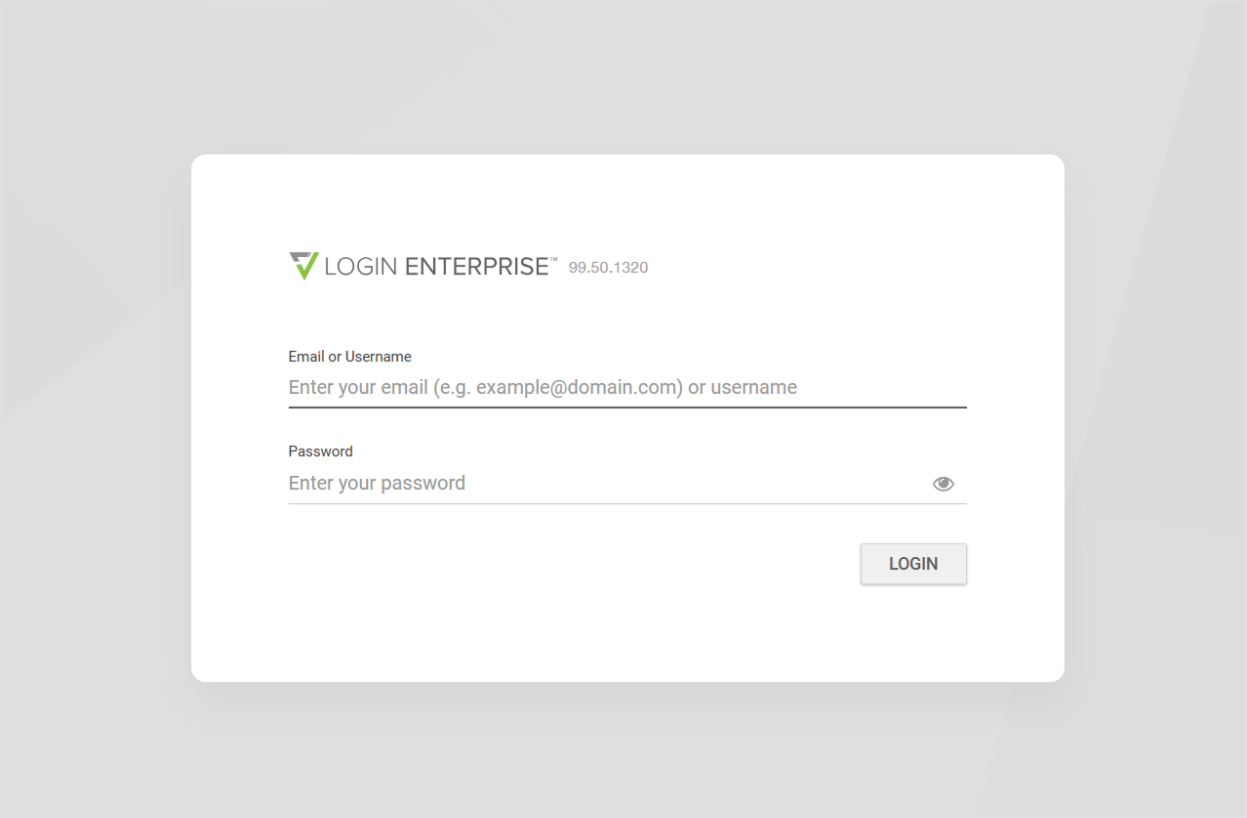
There are multiple options. In our training lab, for instance, we have you place it in your domain's NETLOGON share. You can put it on some other file share that your test accounts will have access to. You can even embed it in your VDI image's local C: drive. (Please remember that you will need to replace `LoginPI.Logon.exe` every time you upgrade the appliance, so a file share might be the easiest option in the long run.)

Once you have decided where you want to put the executable, you need to configure it as a login script for these accounts. You can configure the individual account properties, or you can create a GPO that applies to the OU you put the accounts in. For most situations, a GPO is easier. You create it once for all accounts instead of having to modify each account (potentially hundreds) manually.

Step 4: Configure the Appliance

Licensing

Go to a web browser on a machine that has network access to your appliance IP address. Point your web browser to the appliance FQDN or IP address. If you are using the self-signed certificate, you will likely have to acknowledge an insecure connection. Your login page will look like this:

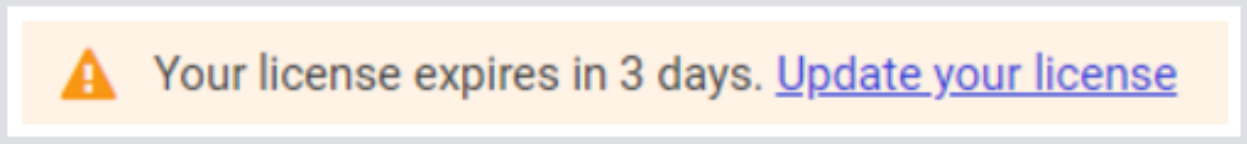


NOTE: This login page UI is relevant for Login Enterprise 6.2 and higher.

Provide your local `admin` username and password. After agreeing to the EULA and acknowledging the release notes, you will be presented with a few options for a quick tour of the appliance and its functionality. Close that.

Your first task here is to upload the license you got from Login VSI. The default grace license gives you 3 days, a maximum of 3 sessions, and a maximum of 3 Applications. (An application in this context is a testing script.) Since the Appliance comes pre-installed with 21 Application Scripts, that effectively means you cannot add any custom scripts of your own.

At the top of the browser page, you will see this warning:



Click the blue link. Click the button that says **Upload License**, and select your license .lic file from your local machine. This is the file you got from Login VSI to enable the software. The UI will update the table to show you the terms of your license. Click **Save**. You can upload an updated license at any time, but it is not possible to return to the grace license without recreating the appliance from the template.

You can continue with the grace license if you need to, at least for the next 72 hours.

LDAP

TIP: If you do not wish to enable LDAP authentication to the appliance, you can skip this section.

Next, you may want to configure the domain connection information. You can configure LDAP integration to allow users to log in to the Appliance web UI using domain credentials, instead of relying only on the built-in local admin account.

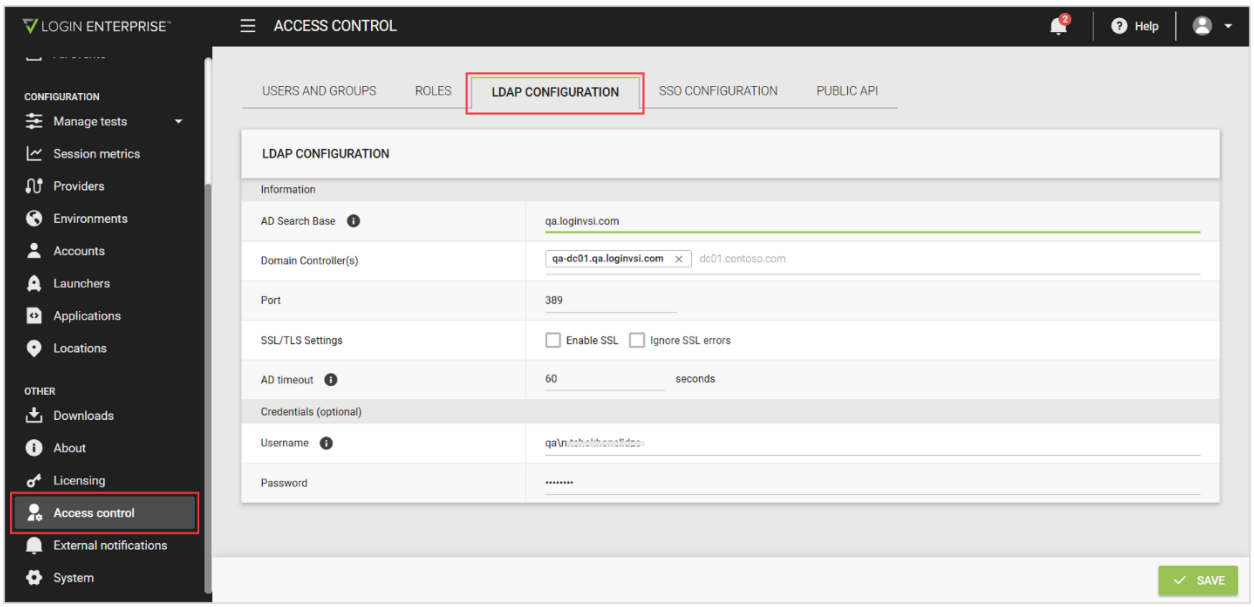
NOTE: LDAP authentication is used only for access to the Appliance UI. It does not apply to console or SSH access, which is limited to local Linux accounts.

This configuration is separate from the Test Accounts used in your test environments. You can use unrelated domains for Test Accounts if needed; the Appliance does not validate those accounts against LDAP.

Set up LDAP Connection

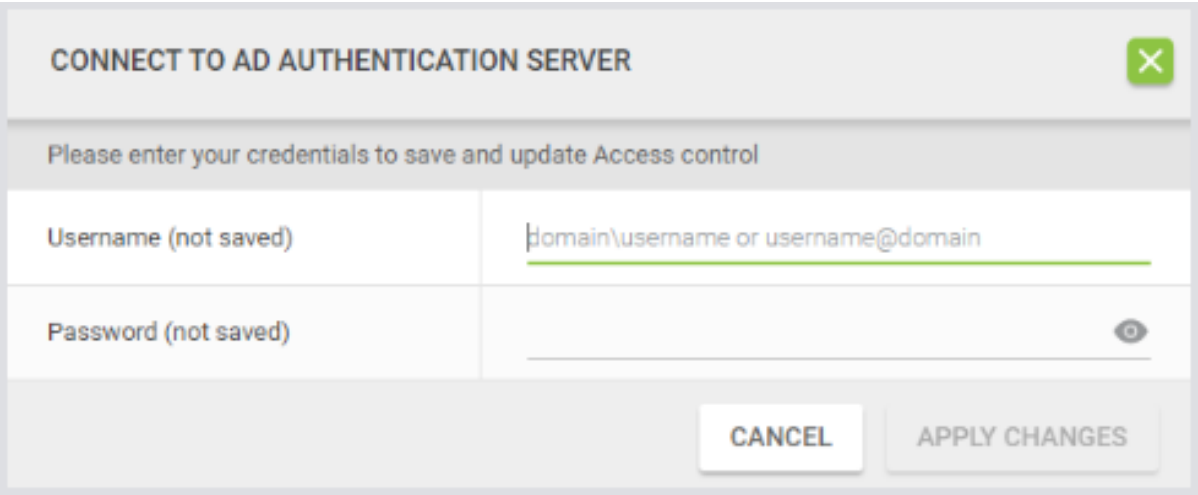
1. In the **Login Enterprise sidebar menu**, navigate to **Other > Access control**.
2. In the **LDAP** tab, fill in:
    1. Domain Name
    2. One or more Domain Controllers (DCs)
    3. Port
    4. SSL Options

You can enter up to five Domain Controllers. Use **Tab** after each entry to add another. The system treats them as a failover group; if one DC is unavailable, it will try the next.



3. Click **Save Changes**. You will be prompted to enter a domain account and password.

- These credentials are used only to verify access to the DC.
- They are not saved or stored by the Appliance.
- You cannot save the configuration unless the connection is validated.



Assign Roles with LDAP Groups or Users

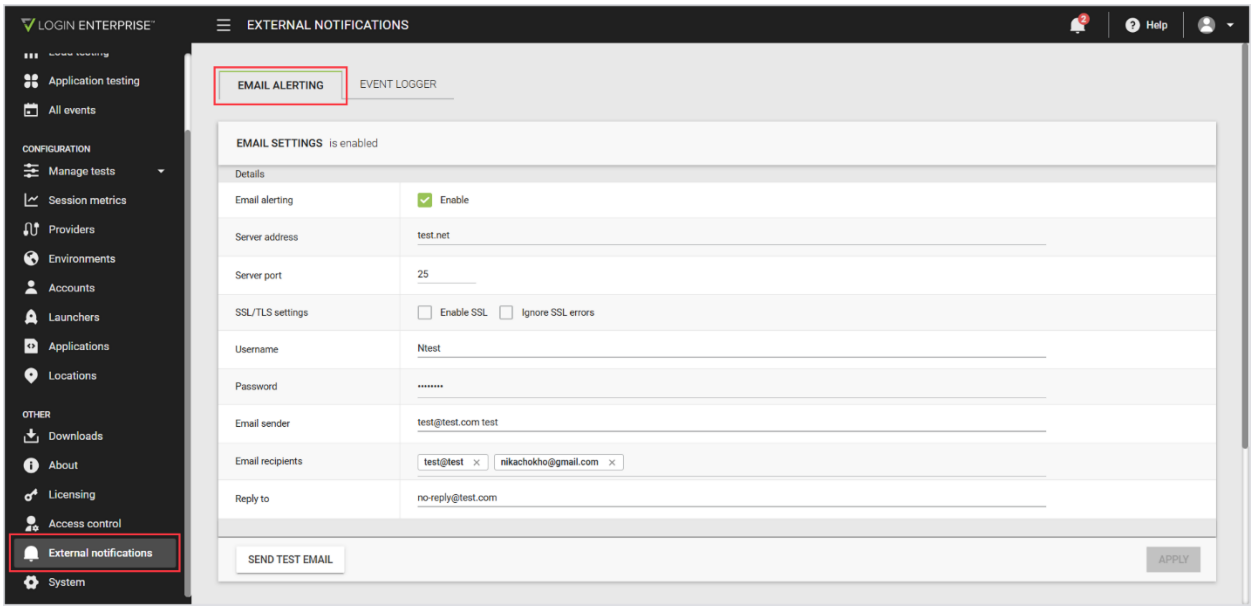
Once LDAP is connected, you need to grant access to specific users or groups and assign them roles.

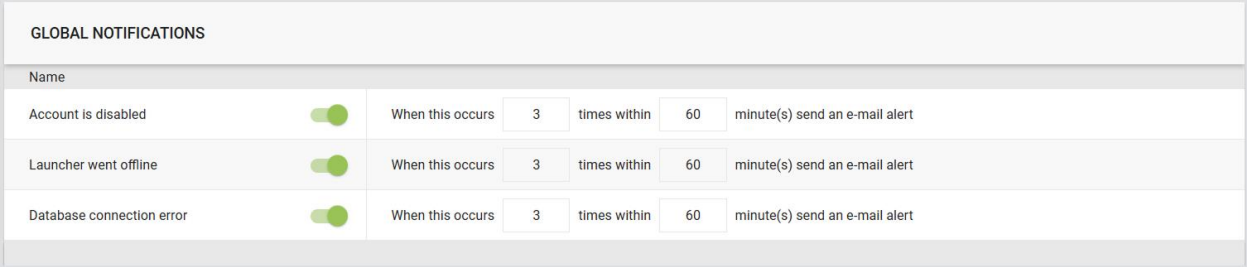
1. Go to the **Users and Groups** tab.
2. Add an **LDAP Group** or an **LDAP User**.
   - The group or user must already exist in your LDAP. The Appliance does **not** create or manage LDAP objects.
3. Assign a role:
   - Use a **Built-in Role** such as **Built-In Admin** for full access.
   - Or create a **Custom Role** with limited permissions.

For detailed instructions on creating and managing custom roles, see [Role-Based Access Control (RBAC)](#).

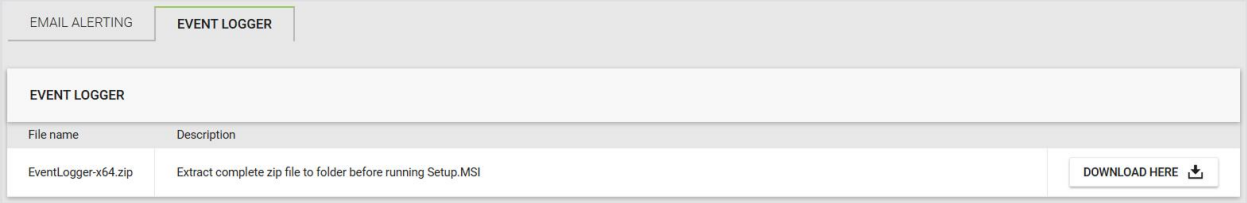
E-Mail

In the **Login Enterprise sidebar menu**, navigate to **External notifications**. Here, you can configure an outbound e-mail server for Test and Event notifications.





There are also tabs for an **Event Logger** package you can install in Windows to receive every event generated within Login Enterprise, as well as information about our **Public API**. These are both beyond the scope of this section.



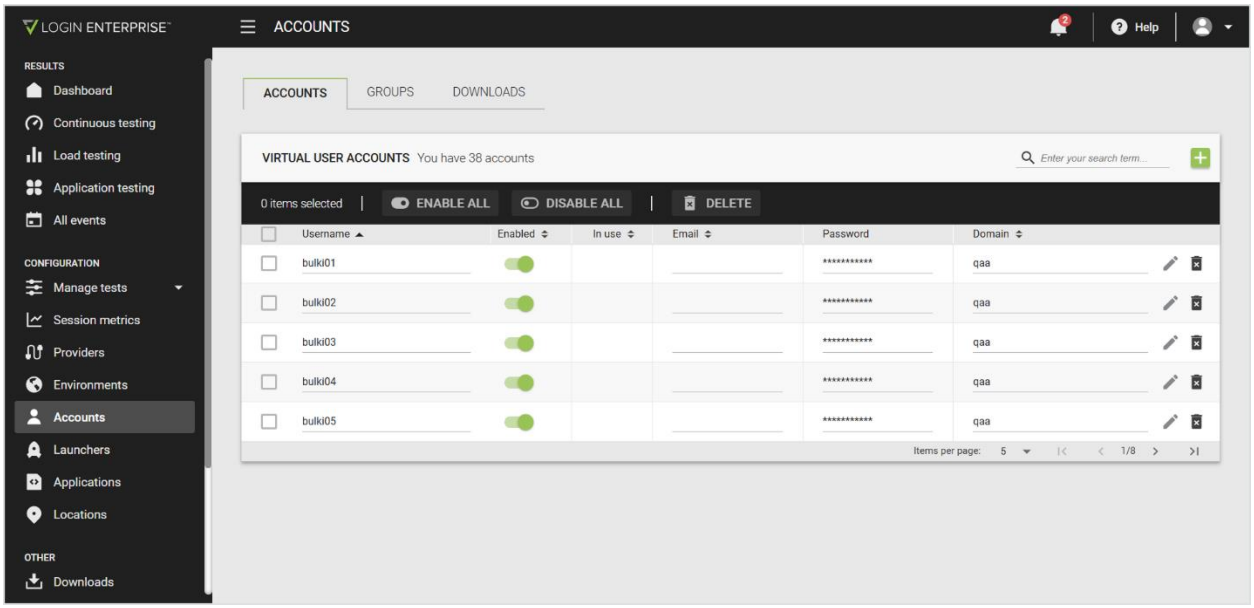
Accounts

You need to tell the appliance what Test Accounts you have created. We don't use these accounts internally at all, and we don't verify them in your active directory. These are simply a set of name/password pairs (with some additional attributes) that we provide to your connection broker. The broker will handle authentication and connection routing.

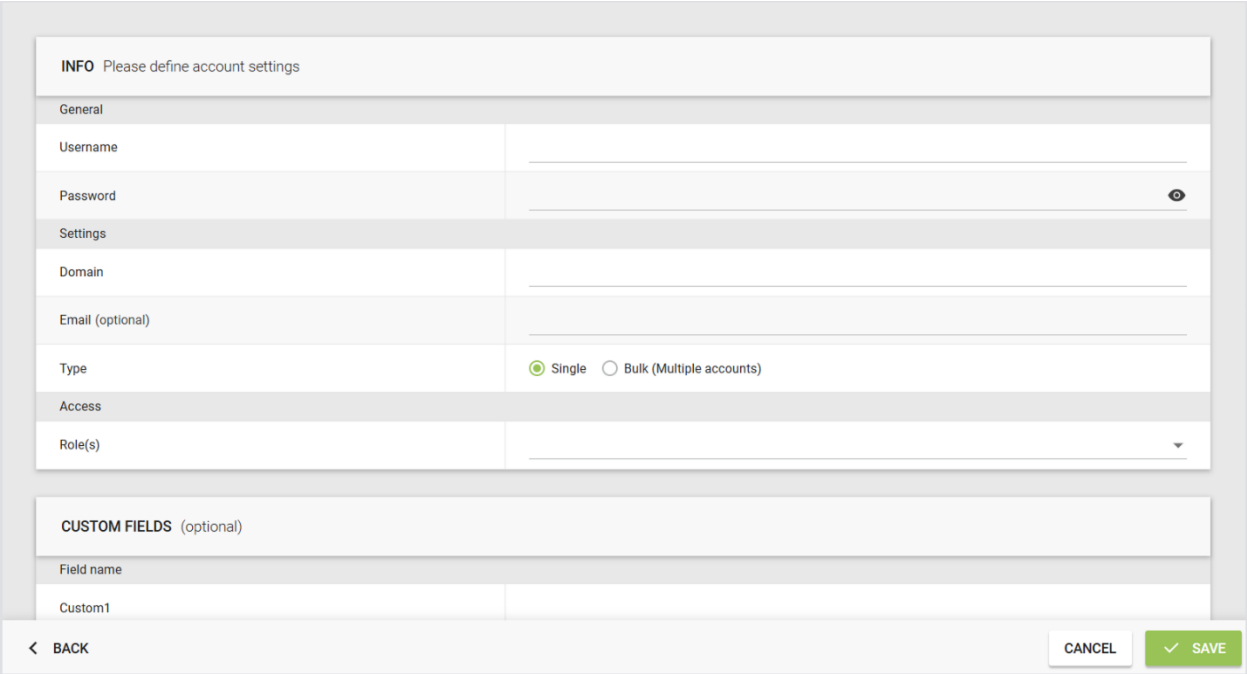
NOTE: There's no requirement that your accounts be part of the LDAP server you have configured for Appliance logins. We simply provide strings to the connector, which it passes to

your connection broker. You can support multiple domains with your testing. Just make sure you keep your usernames unique.

In the **Login Enterprise sidebar menu**, navigate to **Configuration > Accounts**.

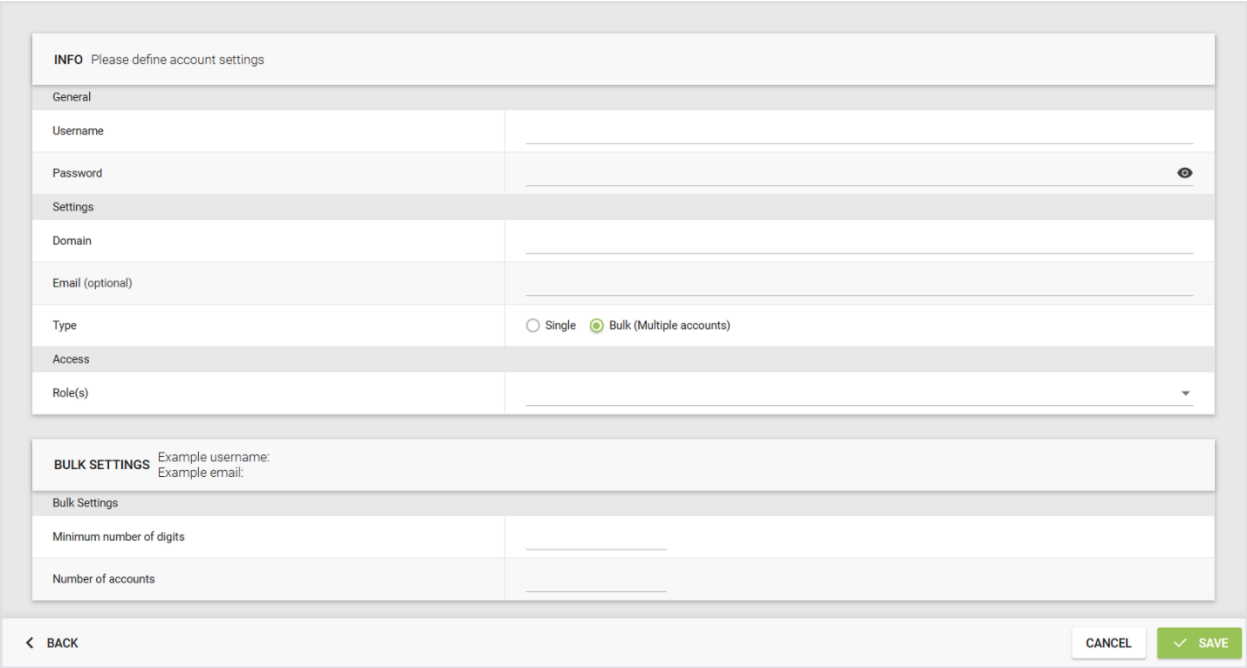


First, create some accounts. Click the green + icon to create either a single account or a set of them. The single-account dialog looks like this:



Provide a name, a password (which is encrypted in the database and is not shown in the UI), and a domain. These fields are provided to the Launcher whenever we need to start a new connection.

Or you can click **Bulk Accounts**. To create 100 sample accounts, see Create Test Accounts. You would fill in the **Bulk Accounts** fields like this:

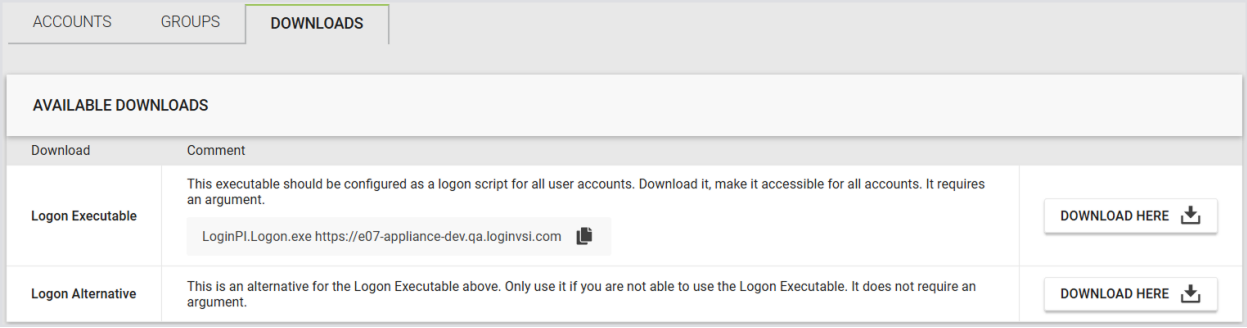


NOTE: This assumes the passwords for all your accounts will be the same. If that's not true, you will need to update them.

The Public API provided by Login Enterprise offers API endpoints for updating account information, including passwords. If you need to script account updates or account creation, contact us for more information.

You can see there is also an Account Groups tab, which you can use to create subsets of your accounts for specific Tests. When you build a Test, you assign it an Account Group. All accounts are automatically members of the "All" group, but you can create sub-lists for different domains, accounts with different permissions, or whatever you need.

Finally, the last tab in the **Accounts** section is called **Downloads**. This is where you get the `LoginPI.Logon.exe` login processor executable discussed above.



Click **Download** next to **Logon Executable**. Save that in whatever location you decide on in the **Configure Test Accounts** section above. Now, when you log in as one of these users, our Login Processor should start automatically.

Logon Executable Verification

1. After logging in as a Test User, do you see the **Login Enterprise icon** and the **Stop Engine** button?
2. If the **Stop Engine** button is missing, it indicates that the **Logon Executable** hasn't run successfully. Without this, Virtual Users will not function.

Step 5: Build a Launcher

A Launcher is simply a regular Windows machine with the Appliance's software installed and running. A Launcher is bound to a specific Login Enterprise Appliance, so you must download the software from the Appliance to install it. The software contains server keys for communicating with the Appliance.

You may build as many Launchers as you need. Launchers have no enforced maximum number of sessions, but Login VSI recommends you provision 1 Launcher for every 15 simultaneous sessions you expect to support. The exact number of sessions an individual Launcher is capable of will depend on the performance of the remote connection system and the hardware configuration of the VM.

You may add and remove launchers at any time. The Login Enterprise Appliance is aware of all the Launchers that are connected (the Launcher application is running) and will only delegate user sessions to those Launchers. Launchers have no persistent data to preserve. You can stop a Launcher, reconfigure it with different hardware, and start it again at any time.

The Launcher does not have any account or authentication requirements. When the Login Enterprise Launcher software starts, it registers with the Login Enterprise appliance. It will receive directions to start sessions and will then execute the remote client software for the desired user and endpoint. It does not need any special permissions for this and does not need to be run as an Administrator. However, your company's security requirements may need you to join the Launcher to the domain, apply certain GPOs, and/or have special permissions for the account that will run the Launcher software.

The Launcher software must run as a user-mode program in an interactive session. It cannot run as a non-interactive service. Normally, we expect this to run on the VM console so that there is no danger of a remote session for the Launcher timing out or getting disconnected. If you choose to connect to the Launcher VM over a remote client and start the Launcher software within that window, you must not minimize that window while our software is running. Windows will report the resolution change (to 0x0) to our remote sessions, and your Application Scripts will start to behave unpredictably.

Login Enterprise recommends the following:

- Designate a specific account to run the Launcher software
- Disable any automatic screen-locking for this account
- Configure Windows to perform an AutoLogin for this account so that, on machine restart, this account will automatically log in. Otherwise, you will have to manually log this account in after an error.
- AutoLogin is a software package built by Microsoft and packaged in their Sysinternals repository.
- Configure Windows (or the service account) to automatically start the Launcher software at login, so on any failure, your system will automatically recover.

Then, start provisioning your system. Build (or identify) an appropriate Windows VM. We recommend 2 CPUs and 4GB RAM as a good starting point. It is not necessary to join the launcher to the domain, but it may be desirable in your environment. Log in as the designated Launcher user.

Install any additional software your system will require. Normally, this is at least the connection client software for your VDI system. If you are using RDP, that is already included in Windows.

To install the Login Enterprise Launcher software, you will need to download the Launcher Installer from the Appliance UI after the Appliance is set up. When you install that software on a Windows machine and run it, it will connect to the Appliance and register. You do not need to configure a Launcher in the Appliance; it simply shows up.

Each Launcher Installer (the MSI and JSON files together constitute the Installer) is specific to the Appliance from which it was downloaded. If you have multiple Appliances, you will have multiple, separate Launcher Installer packages. A Launcher is bound to a single Login Enterprise system. If you want to direct a Launcher to another Appliance, you will need to run the installer from that Appliance. You cannot have multiple Launchers installed on a single host.

Execute the MSI and allow the software to complete the installation. There are no options for the installer; it simply runs. It already knows what to do. During the installation process, it will download the server certificate from the Appliance and install it locally. If there is a problem making an HTTPS connection to the Appliance, the installer will error out.

- For more details on the Launcher setup and configuration, see Configuring the Windows Launcher.

- For the general Launcher Overview and best practices, see Launcher Overview and Best Practices.

After installation, verify that the Launcher shows Connected and that the Appliance shows the Launcher as online. If you configured AutoLogin and a startup script, reboot and verify that your system reconnects.

In the Login Enterprise UI, you can also configure Launcher Groups either by selecting specific Launchers or by specifying a name pattern. This would allow you, for instance, to group Launchers by geographic location if you're trying to test widely distributed systems and only want to test connections where the client (the Launcher) is geographically near the server (the end-user desktop).

Step 6: Configure End-User VM Pool

Login Enterprise operates by logging into a VM as an interactive user, having our software start, and remotely controlling that VM. To work, you need to configure some kind of remote access to a pool of available VMs. That pool can be as small as 1.

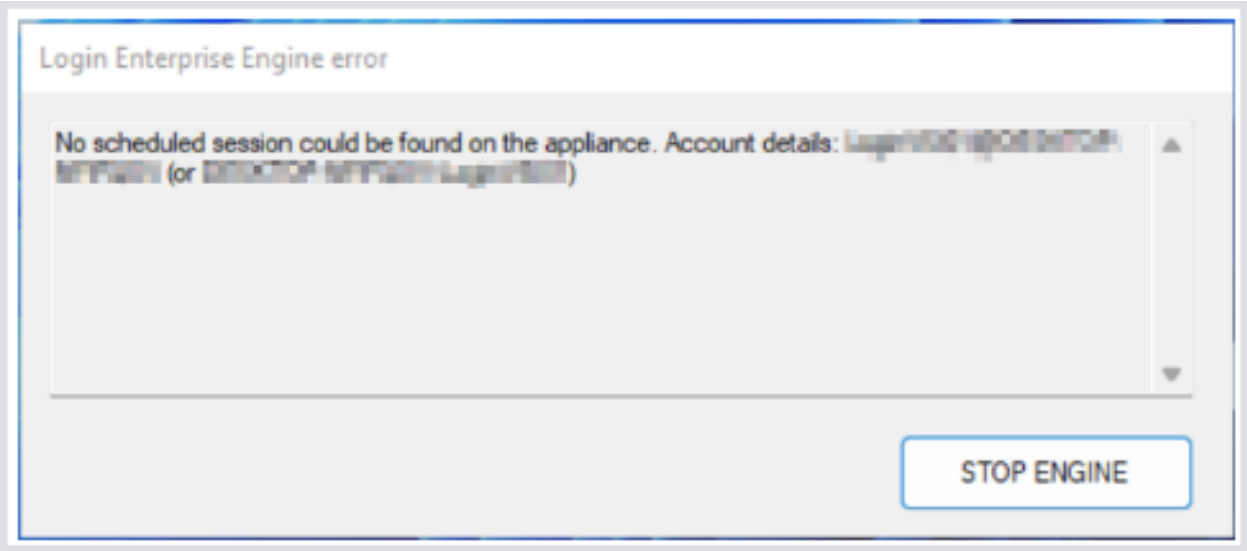
For the end-user VM pool, you need to decide how you want to handle that. Normally, you already have a connection broker. You could set up a special VM pool for testing or allow your test sessions to run on your normal end-user images.

If you do not already have an existing pool of VMs to use, you can deploy normal Windows machines. There are no particular specifications for these machines. If you decide to embed our Login Processor in your machine image, make sure you do that, and make sure the path matches what you set in your Login Script definition. If you intend to measure the experience of your existing users, provision your test machines with the same hardware and policy configurations as your end users.

Configure your connection broker to allow your set of test users to access the VM pool you want.

NOTE: If you don't have a broker available or you just want a simple walk-through, you can make a very simple Test by creating a single machine that is allowed to accept RDP connections from your desktops. RDP does not require a real broker; we can simply run MSTSC to connect to the designated machine using the appropriate credentials.

Test the connection manually. Using the name and password of one of your Test Users, manually connect to the desired pool or host. When your account logs in, you should get this error:



You get that error because no Login Enterprise Test is running for the user you have logged in as. If you want to work with the desktop, click **Stop Engine** now. Otherwise, our software will log you off to allow the account/machine to be available for the next Test. If you see that dialog, then the Login Processor is correctly configured.

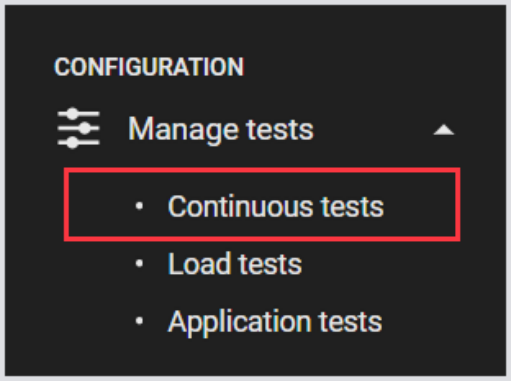
Step 7: Create and Execute a Test

Before configuring your full test setup with a connection broker (like Microsoft RDS), it's recommended first to verify your environment using the **Desktop Connector**. This simpler setup allows you to trigger a test manually from your own desktop machine, helping ensure the rest of your configuration is functioning as expected.

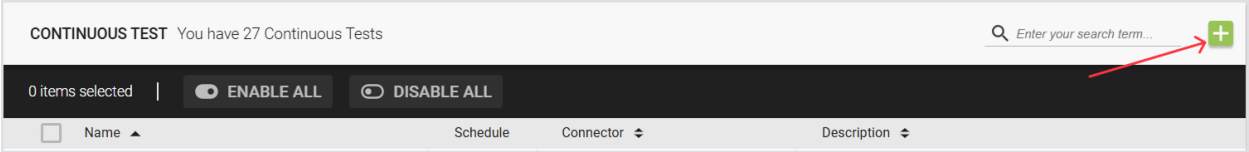
Option 1: Run a Test Using the Desktop Connector

The **Desktop Connector** enables you to run a test without relying on a remote-access system. Instead, you manually launch the test from a desktop session using a downloadable command.

1. Go to **Manage Tests > Continuous Tests** in the **Configuration** section.

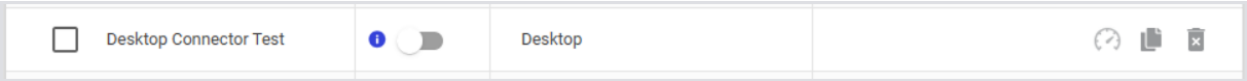


2. Click the green + icon to create a new Test.



3. Enter a **Test Name** and set the **Connector** to **Desktop**.
4. Optionally fill in the following:
    1. **Description**
    2. **Environment**
    3. **Access**

5. Once the Test is saved, you can add steps and configure the test as you desire. Then go back to the list of Tests, enable your Test by clicking the slider:



6. Click the blue icon to get the command to start the Test:

7. Run this command from your desktop session to initiate the Test.

The Desktop Connector does not require additional configuration parameters. It's ideal for:

- Verifying basic test setup before configuring full remote access.
- Troubleshooting or running ad hoc Tests from machines with active user sessions.
- Situations where remote login automation isn't feasible.

Limitations:

- No Logon or Latency measurements.
- Only application-related failures are captured.
- Notifications and thresholds can still be configured.
- You cannot change a Test from the Desktop Connector to any other connector, or vice versa.

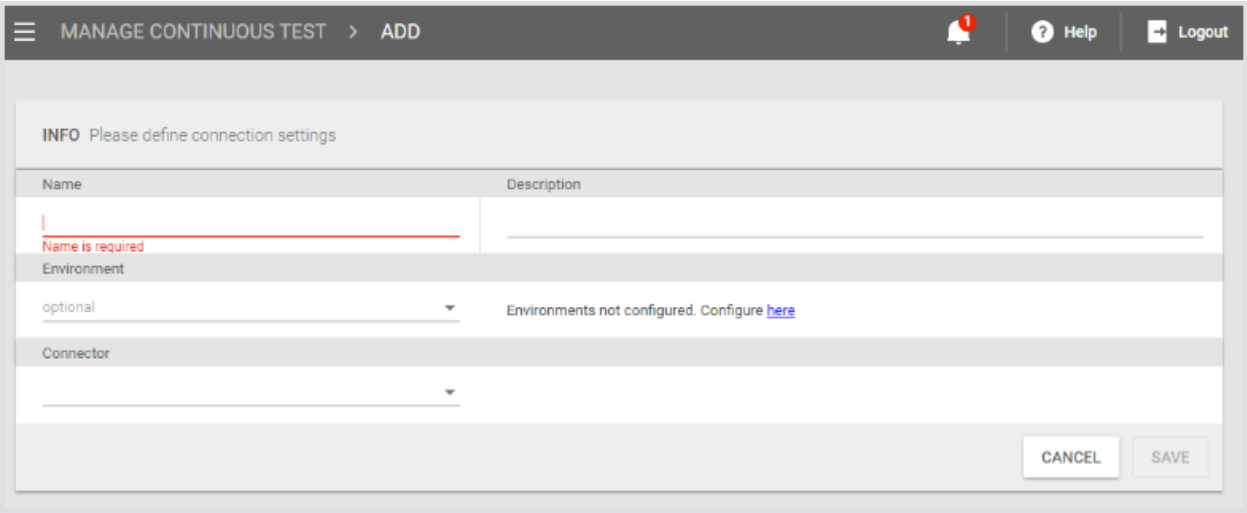
For more information on the Desktop Connector, see the Desktop Connector.

Option 2: Configure and Run a Test Using Microsoft RDS

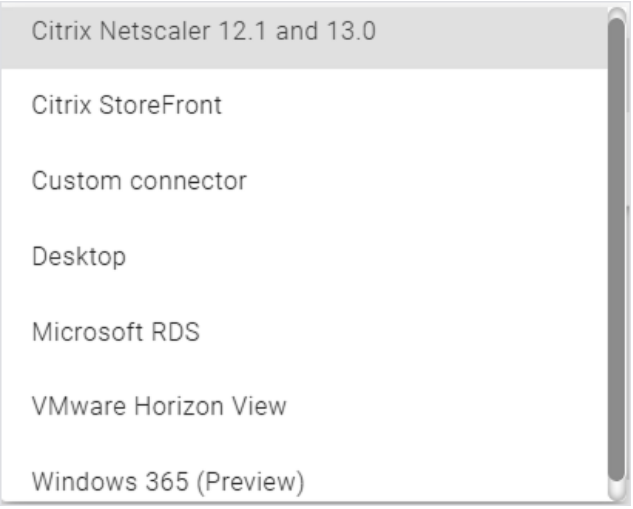
Once you've confirmed that the Desktop Connector works, you can move on to configuring a Test using a connection broker such as **Microsoft RDS**.

Create a simple Test just to verify that we can successfully initiate a login and get results, e.g., a Continuous Test. For this:

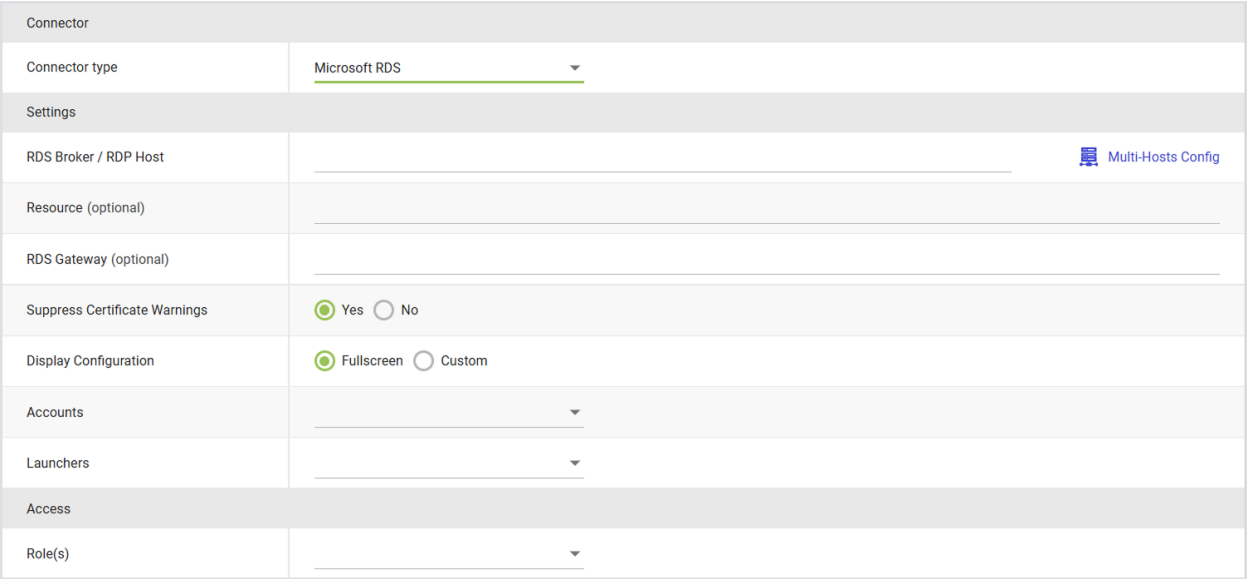
1. Go to **Manage Tests > Continuous Tests** in the **Configuration** section.
2. Click the green + icon to create a new Test.



3. Enter a **Test Name** and choose a **Connector**. This acts as the connection broker for your remote desktop pool.

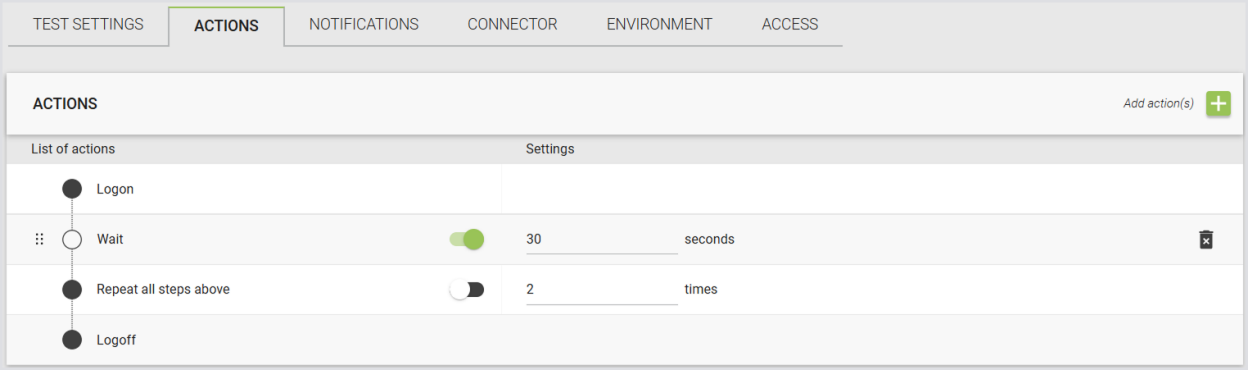


4. Select **Microsoft RDS** or another broker type suitable for your environment.
5. For a single-host RDP, just enter the hostname and select the Account and Launcher groups. Select **All** for both. This means that all of your configured accounts and connected Launchers are available for this Test.



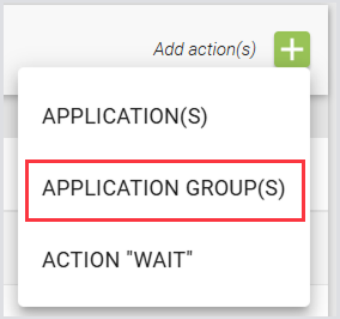
After you hit **Save**, you will be taken to the full **Test Configuration** page. For this walk-through, leave most things set to the default. We'll configure the Test to run one of our preconfigured Test suites and to run one session at a time.

First, tell it what to do. Select **Actions** from the tab menu:

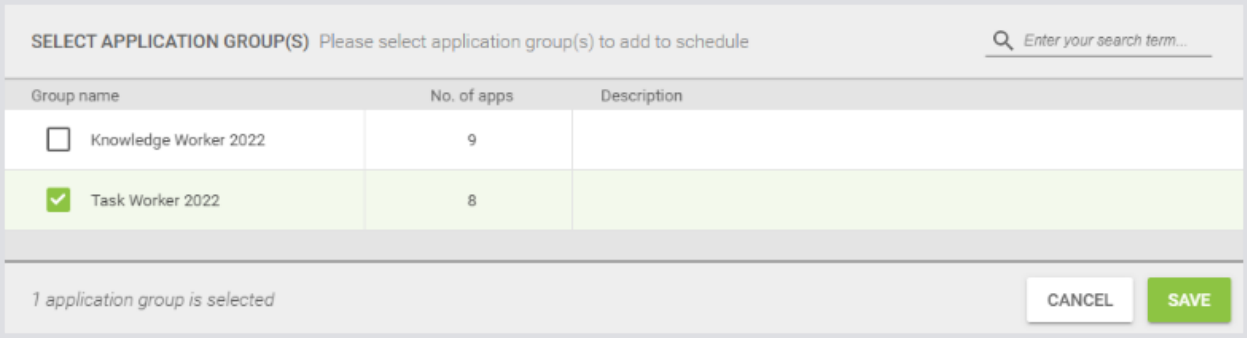


By default, a Test is just "login, wait 30 seconds, and log out."

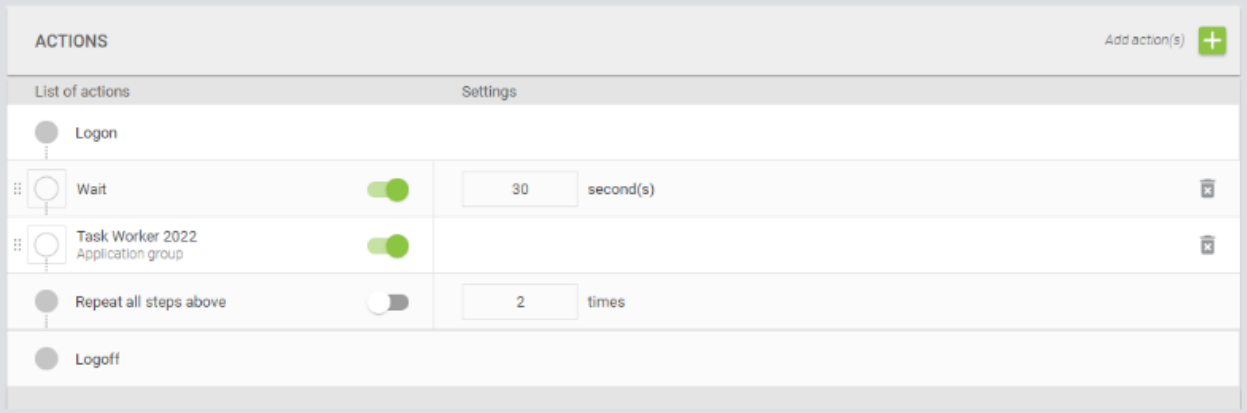
Click the green + and select **Application Group**:



Select **Task Worker** (a lower-power user) and hit **Save**.

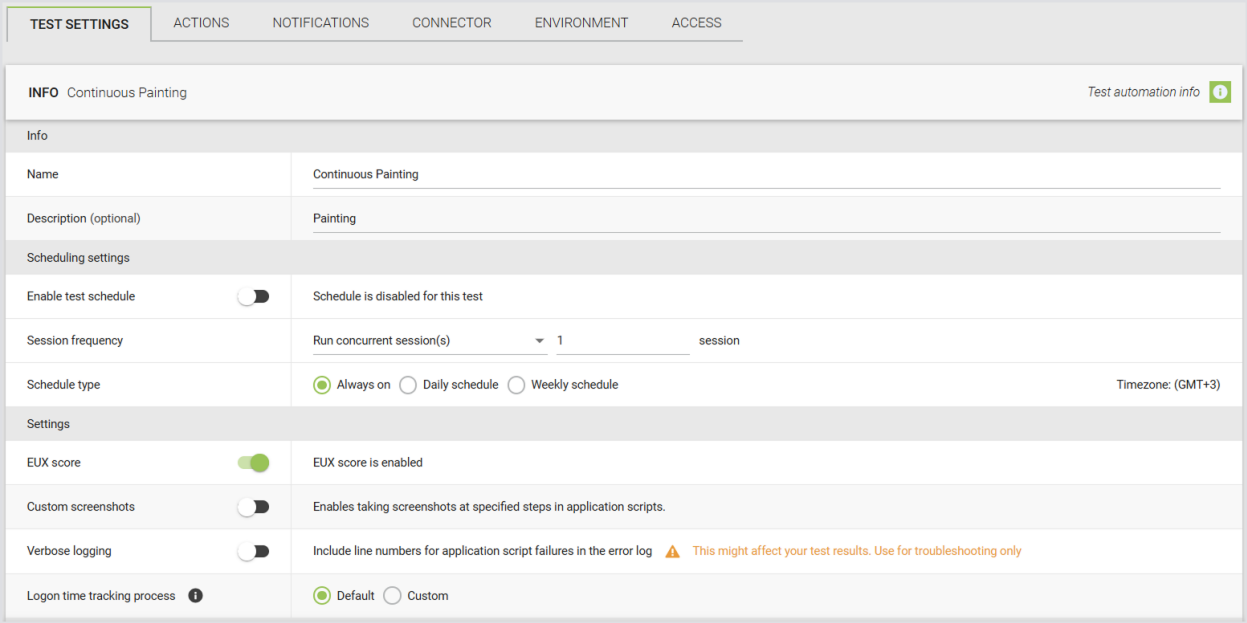


Then your new Test group is included in your Actions list:



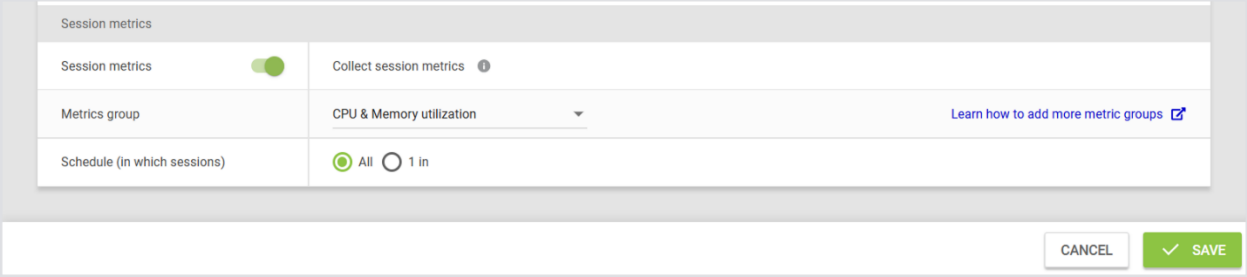
Now, configure the test schedule to have it run one Test at a time, back-to-back.

Change the schedule to **Run Concurrent Sessions** for 1 session.



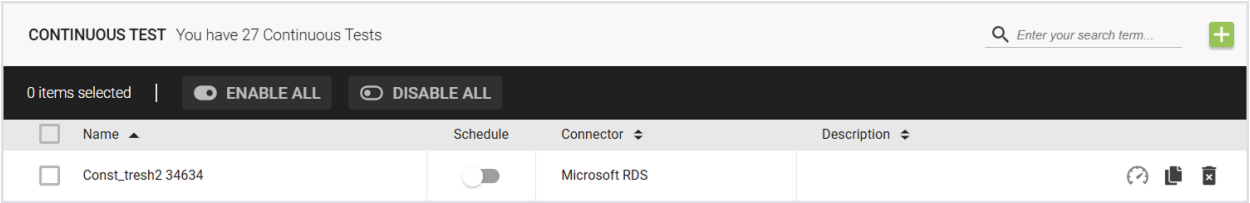
IMPORTANT: Don't enable the schedule yet. The moment you click **Enable Schedule**, the Tests will start running. Don't do that until the rest of the Test is configured.

There's something more interesting to look at: turn on **Session Metrics**.



Once you've finished configuring your Test, remember to click **Save** to apply Test configurations.

At the top of the screen, click **Manage Continuous Test** so you can see the list and enable our new Test. Click the **Schedule** switch on your Test. It will turn green, and your test will start.



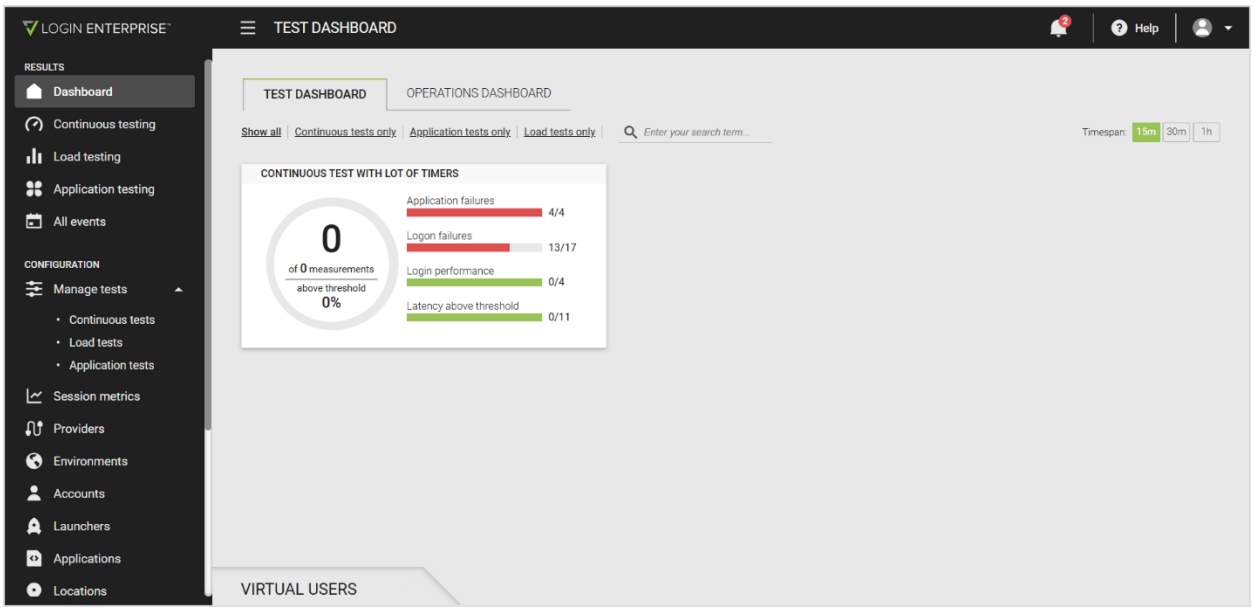
For more information on Microsoft RDS and other connection options, see Connectors.

Step 8: View Results

The full set of results available is beyond the scope of this section. As a quick example of what you can do, look at the Dashboard. Click **Dashboard** in the **Results** menu section.

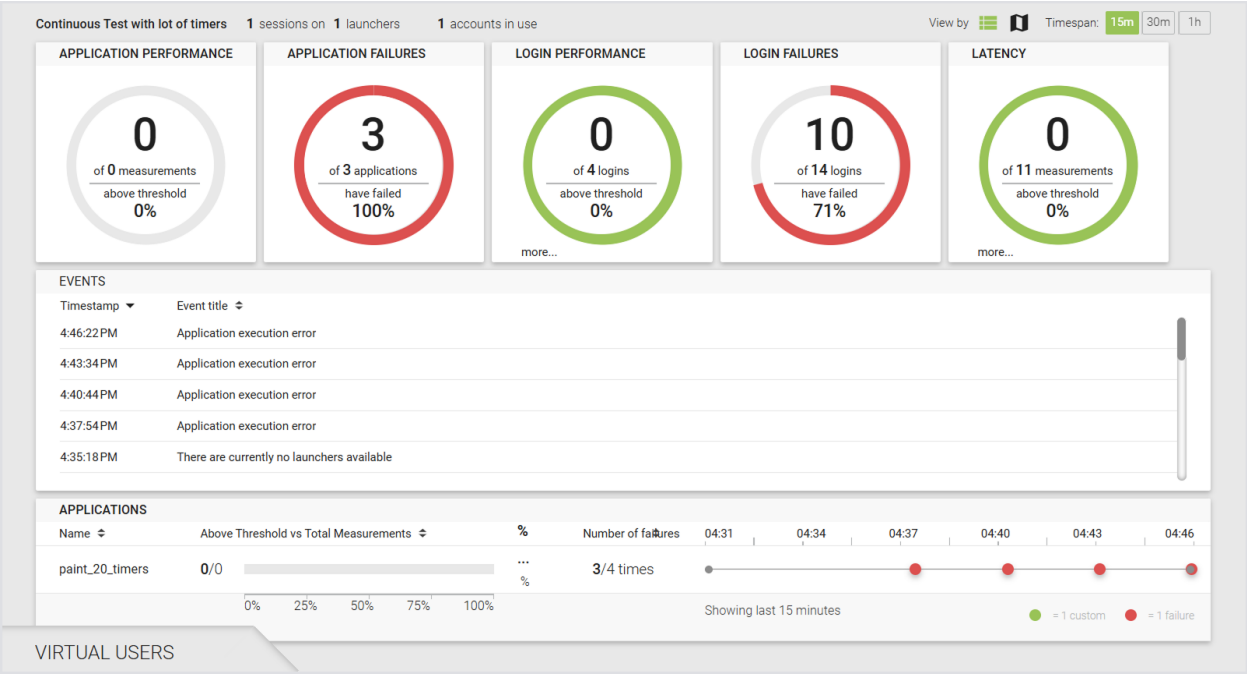
NOTE: An example below illustrates the old Continuous Test results page. The page was changed in Login Enterprise 6.6. For detailed information, refer to [Viewing Continuous Testing Results](#).

This shows you information on all Tests currently running.

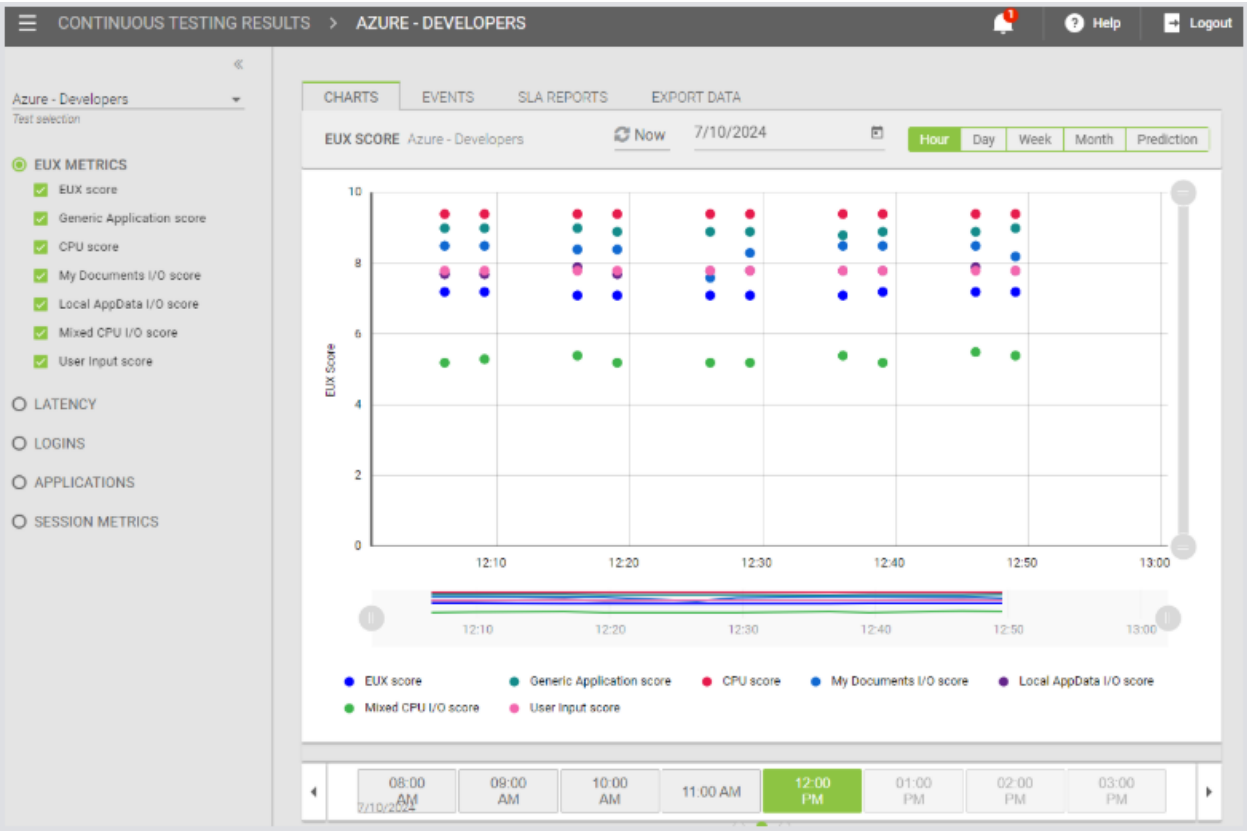


You can see simple bar graphs about login success, login performance, defined latency thresholds, and Application Test failures. You can click on one of these tiles to get more details about those metrics.

That will show you which Tests are failing and what events have been reported about your tests. For instance, if your logins are failing because of a misconfiguration somewhere, click on the tile, look at the Events table, and open up each event for more information.



Instead, go down to the **Continuous Tests** section of the menu to look at some of the performance graphs. Immediately under the **Dashboard** menu entry, click **Continuous Tests**. That defaults to the Charts tab, where you can see metric data for your Test Runs laid out.



First, note the pull-down at the top left. You need to select which test you want to look at. For this walk-through, you have only one so it should be shown by default. In other systems, you should always pick your test first.

Let's look at EUX Metrics – the relative performance of our experience-measuring tools. Click **EUX Metrics** and then select all the metrics. You will see a chart like the one above, where you can see the overall EUX score and the component score of each of our applications. To view a subset of these, you can either uncheck the metric or click on the color in the legend at the bottom.

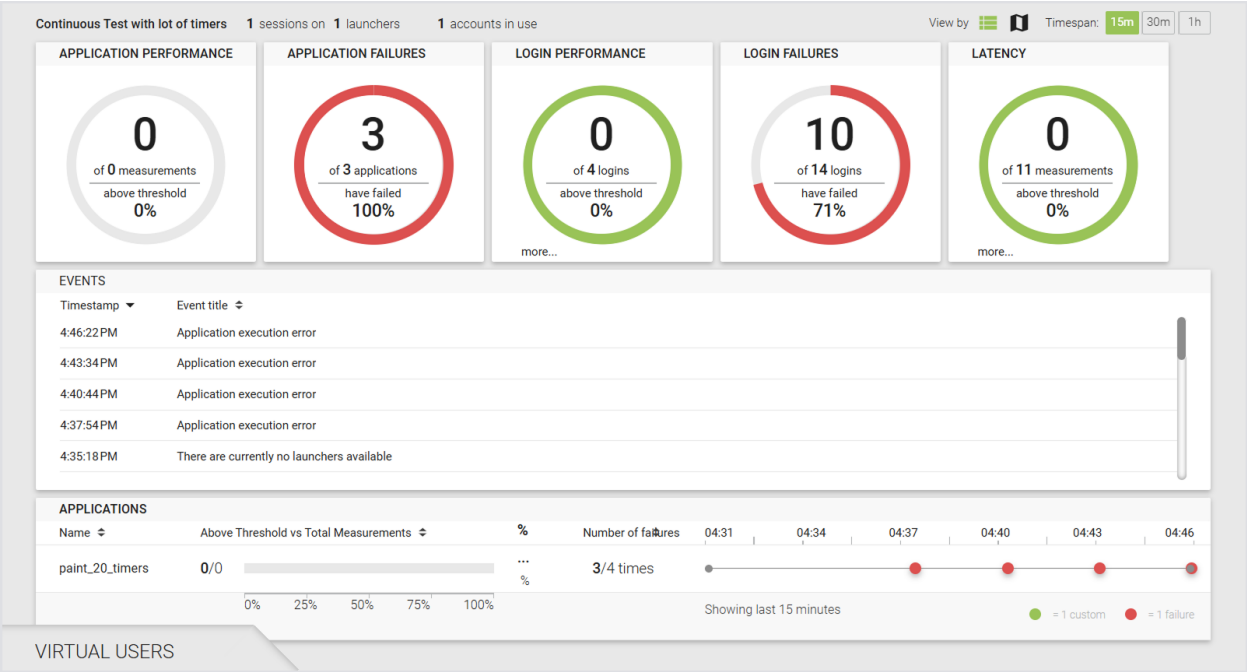
At the top of the graph, you can choose your date and your reporting period. It defaults to "hours", with a scrolling selector at the bottom for which hour. You can also look at the entire day, an entire month, or even a linear projection of the view into the future.

For instance, you can look at the Session Metrics for the month of June like this:

The other metric headings have the same behavior. Select the metric type you want, and that will open up the list of available metrics. Select whatever you want to show in the graph and select your time frame.

The next tab, **Events**, is where you can get more details about failures in the tests. Any Test failure, any metric that exceeded a configured threshold, or any login problem will create an Event. If your script includes CreateEvent commands, those events will appear here, too.

The Events list shows all of the events for all Continuous Tests, rather than allowing you to filter by test. It is limited to displaying 1 hour of events at a time. Please note it is 1-hour blocks, i.e., 12-1. It is not possible to specify the hour of time the events are shown for. One hour of events on a large system can still be many thousands. At the top, you can choose the day, and at the bottom, you can choose the hour, as well as how many events will be displayed per page of the results.



Every error event that is generated by a Test running on an end-user desktop will include a copy of the Engine Log as a downloadable object. See the download link on the right. Many events, including informational events generated by the script, can also include screenshots. Events that reflect problems outside of the end-user desktop, like login failures, have no logs available.

TIP: For every event here, there is a triangle to the left. Click on that to get more details about the error, including any additional messages, as well as the account, host, and launcher involved.

Conclusion

That's a very basic walk-through of setting up a new Appliance and getting a simple Test running. There is a lot more detail than can be covered here. We strongly recommend you take the introductory training lab and explore the UI now that you've had a basic introduction to its mechanics.

Managing Updates and Upgrades

This subchapter guides you in maintaining your system's performance and security through updates and upgrades.

You will learn how to update the Virtual Appliance to ensure optimal functionality, as well as how to update Debian to keep your environment secure and up to date.

For further details, refer to the specific sections linked within this subchapter.

- Updating the Appliance
  - Updating the Appliance from v5 to v6
- Updating Debian

Updating the Appliance

- Overview
- Important notice on upgrading Login Enterprise
- Updating Login Enterprise from v5 to v6
- Debian Linux upgrades
- Update notifications
- Configuring Login Enterprise proxy settings
- Online update of Login Enterprise
- Offline update of Login Enterprise
- Offline update without a virtual CD-ROM drive
- System update of the Login Enterprise operating environment
- Disk cleanup in the Appliance upgrade process
- Updating Launchers

Overview

While we recommend reviewing the instructions below, here's a video on updating your Login Enterprise Virtual Appliance and Launchers.

The Login Enterprise upgrade can be initiated from the VM console or an SSH session. You can upgrade with or without Internet access. If you want to upgrade directly from the Internet, ensure the following URLs are whitelisted in your environment:

https://loginvsidata.s3-eu-west-1.amazonaws.com/LoginPI3_ISO/public/
https://loginvsidata.s3.eu-west-1.amazonaws.com/LoginPI3_ISO/public/

We recommend backing up the Appliance before performing the update. One suggested method is to take a snapshot using the hypervisor management system.

Important notice on upgrading Login Enterprise

When upgrading your Login Enterprise Virtual Appliance, it's essential to follow a recommended upgrade path based on your current version and the underlying Debian version. This will help ensure a smoother transition and minimize potential compatibility issues.

Current Version	Upgrade Path	Target OS
3.x	→ 4.0–4.6 → 4.7–5.8 → 5.9–5.14	Debian 9 → Debian 10 → Debian 12
4.0	→ 5.8 → 5.9–5.14	Debian 10 → Debian 12
5.0	→ 5.14 → any 6.x release	Debian 12

Key notes:

- Upgrading directly from versions **3.x or 4.0 to 5.14** without following this path is not recommended, as updates from Debian 9 to 12 are unsupported.
- While you can choose to stay on an older Debian version, for example, Debian 9 or 10, when running Login Enterprise 5.14, updating Debian is **strongly recommended** for optimal performance and security.
- Version 6 requires Debian 12, so you must allow the System Upgrade to Debian 12 before you can upgrade to any version 6 release.

Important considerations:

- The upgrade will not migrate the old license from the 3.x legacy database, nor will it carry over SLA report schedules configured in 4.0. After upgrading, you will need to upload a new license due to format changes and reconfigure your SLA report schedules **(This only applies when updating from 3.x/4.0 directly to 5.14)**.
- All configuration and licensing information is preserved in all upgrades after 4.3.
- Version 5 Launchers are not supported in version 6: you must upgrade your Launchers when you upgrade the Appliance.
- Please review the release notes for any version-specific updates or changes.

If you have any questions or concerns about updating the Virtual Appliance, contact us.

Updating Login Enterprise from v5 to v6

For detailed steps, see Updating the Appliance from v5 to v6.

Debian Linux upgrades

Starting v5.9, Login Enterprise is based on Debian Linux 12. Systems upgrading from before 5.9 have the option to migrate to Debian 12 during the upgrade. However, you can also choose to remain on Debian 10.

If you are still on Debian 10, any upgrade to 5.9 and above that includes a “System Update” will migrate you to Debian 12. If you are already at Debian 12, System Update just performs a normal security/patching update like the menu entry below. So, in any version, simply upgrade and select the option to perform a System Update to migrate to Debian 12.

If you are already at Login Enterprise version 5.14 and wish to upgrade Debian, you can perform an upgrade to the same version. That is, attempting to upgrade 5.14 to the latest (also 5.14) will run, and then offer to perform the System Update.

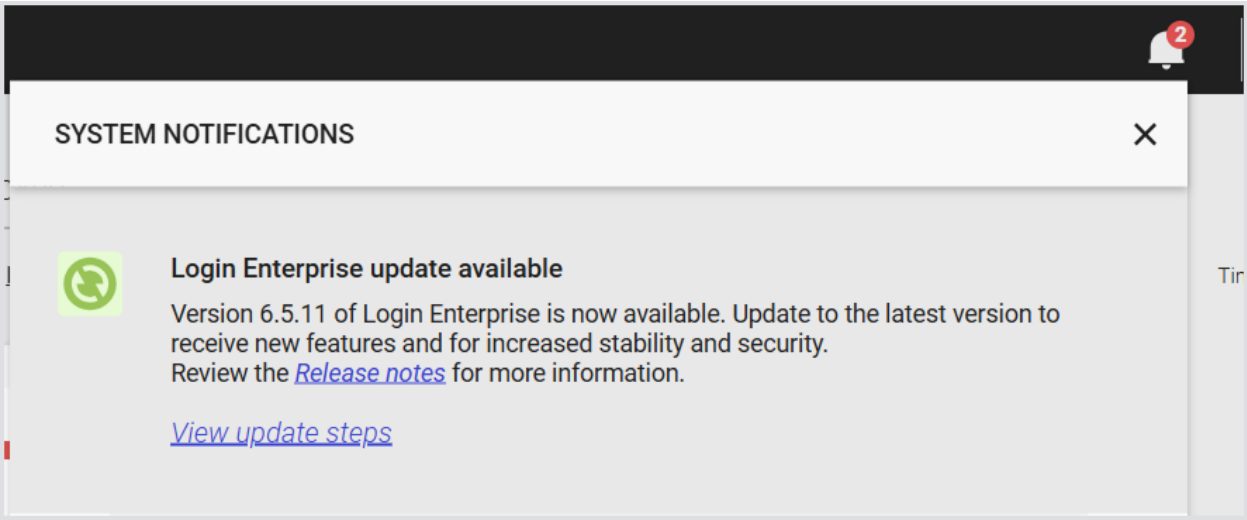
IMPORTANT: Upgrading to Login Enterprise 6.0 and later requires that you are already on Debian Linux 12. Please perform a Debian Linux 12 upgrade before attempting to upgrade to Login Enterprise 6.0.

Update notifications

When a new version of the Virtual Appliance is available, Login Enterprise shows a system notification on the Dashboard page.

NOTE: Upgrade notifications are only available for Appliances with Internet access. Notifications will not be displayed for offline Appliances.

Below is an example of a system notification:



Configuring Login Enterprise proxy settings

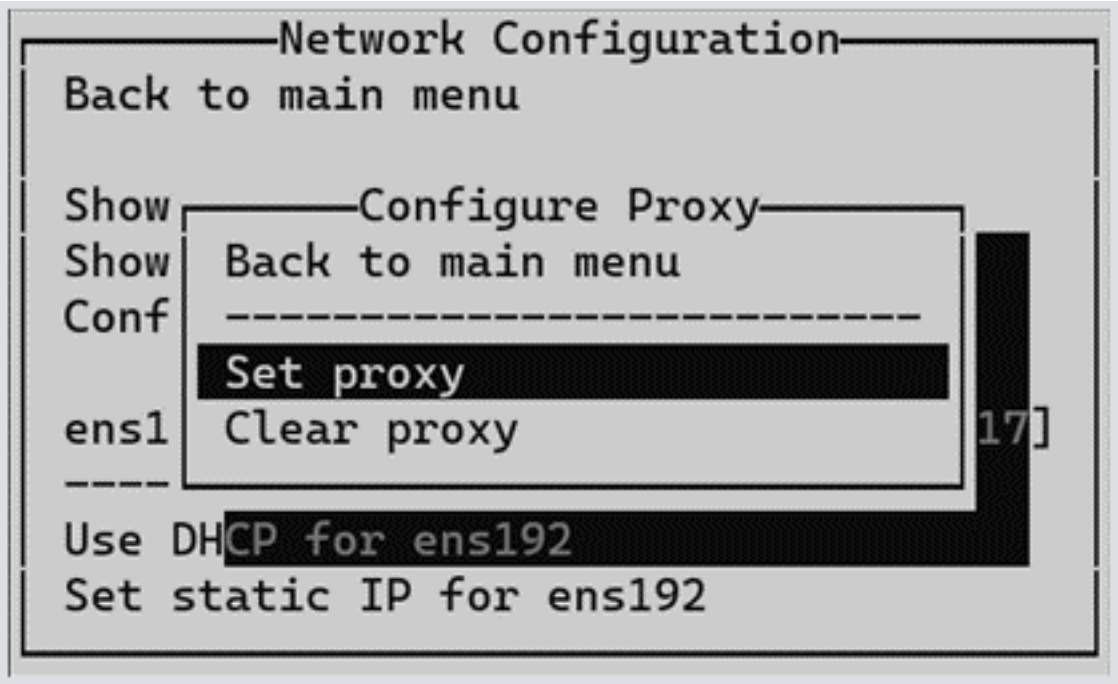
Before performing an online update, configure the proxy settings on the Virtual Appliance.

If your organization uses a proxy within its internal network, you may want to route traffic to and from your Virtual Appliance through this proxy.

To set up a proxy, access the Maintenance Menu of the Appliance. Log onto the Virtual Machine hosting the Appliance (for example, by using a terminal and logging in via SSH).

```
PS C:\Users\giedr> ssh admin@p16-appliance.play.loginvsi.com
admin@p16-appliance.play.loginvsi.com's password:
```

After logging in successfully, navigate through the graphical menu of the Virtual Appliance. In the "Maintenance Menu," go to the "Network Configuration" section, and select the "Configure Proxy" sub-menu. This will display two options:



- **Set Proxy** - Allows you to configure the proxy settings:
  - Define the proxy URL, pointing to the proxy server, so that all HTTP requests are routed through it. You can specify a port number along with domain names or IP addresses if necessary. To ensure proper configuration and minimize user errors, basic validation is applied: no empty values or whitespace are allowed, and all inputs must start with either `http://` or `https://`. Only one proxy can be configured at a time.
  - You can also save a list of HTTP proxy exceptions to bypass the proxy for specific web servers. This list should consist of domain names or IP addresses for which the HTTP proxy should not be used, and it must be entered as a comma-separated list. A few default exceptions are hard-coded and cannot be removed (including `127.0.0.1`, `localhost`, and `::1`).

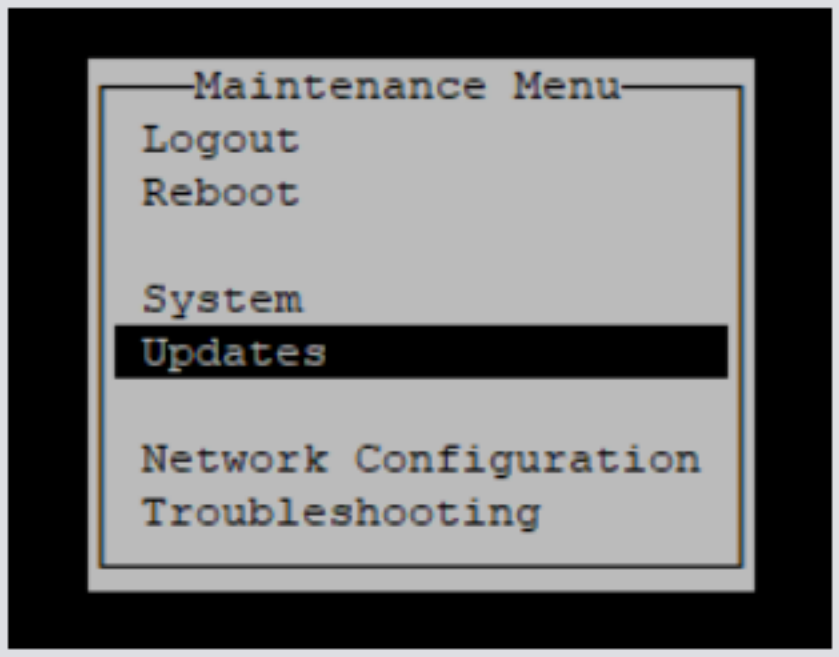
```
Enter proxy URL (e.g.: http://my-proxy:1234/): http://10.40.205.5:3128/
Enter addresses that should be excluded from proxying (e.g.: my-domain,10.0.0.1,some-domain.com):
```

- **Clear Proxy** - Allows you to quickly remove any configured proxy settings without additional prompts for further specification.
  - After making any changes to the proxy settings, you must restart any containers that use the proxy to ensure the changes take effect. Currently, only the **loginenterprise** container utilizes the proxy, so it will be restarted automatically following any changes.
  - **NOTE:** Proxy settings are configured solely for the following menu items: **Updates / Update from the Internet**, **Updates / System Update from the Internet**, and **Troubleshooting / Open Bash Shell**.

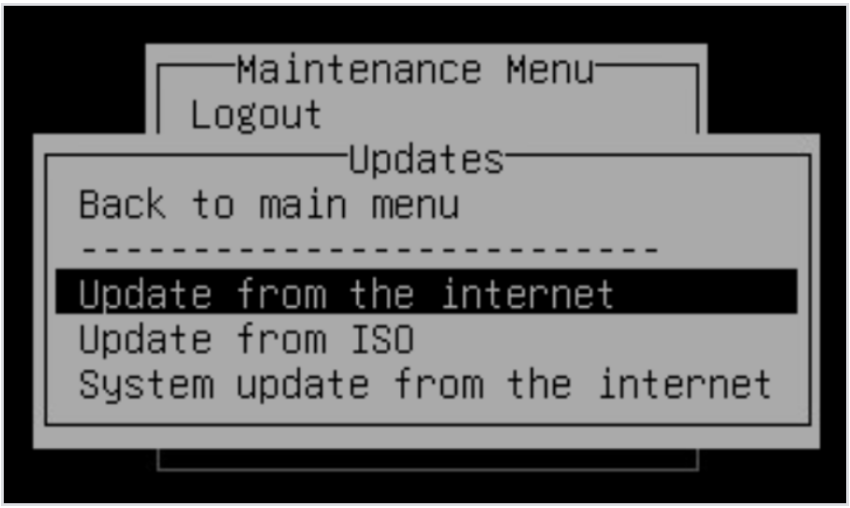
Once the proxy configuration has been successfully set up, access the Maintenance Menu. From there, navigate to the **Updates** sub-menu, where you will find options for upgrading both the Login Enterprise software and the Debian OS. For details, see the steps below.

Online update of Login Enterprise

1. Open the Appliance console and log in to access the **Maintenance Menu**.
2. In the **Maintenance Menu**, select **Updates**.



3. In the **Updates** menu, select **Update from the Internet**.



4. Wait until the ISO file download is complete.

```
Downloading LoginEnterprise-5.11.12.iso
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  3 3196M    3  110M    0     0  40.9M      0  0:01:18  0:00:02  0:01:16 40.9M
```

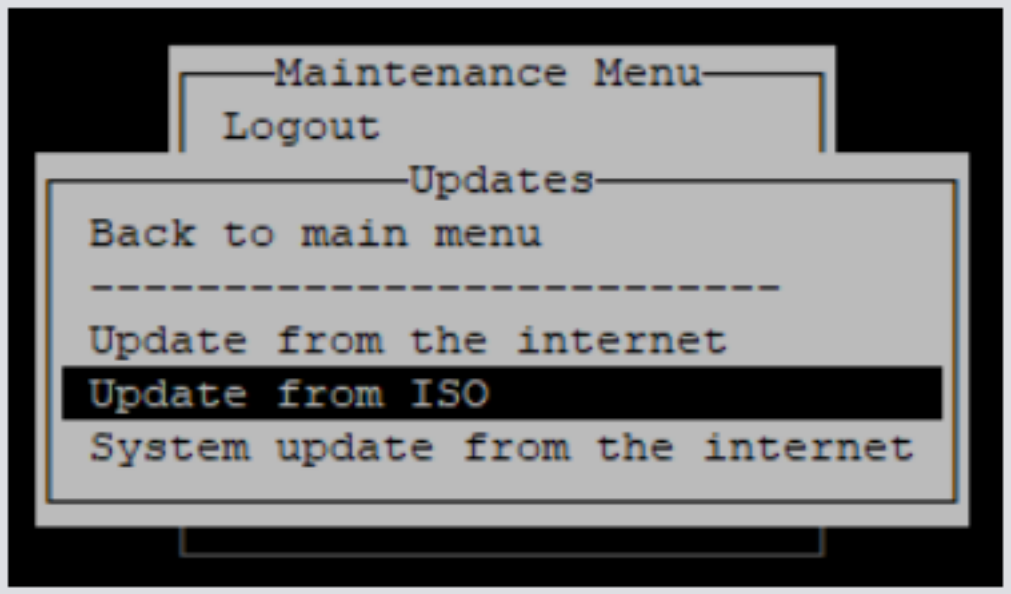
5. Afterward, you'll see the following message:

```
# # # #
```

8. Press **Y** to perform database maintenance or **N** to skip it.

```
# # # #
```

2. In the **Updates** menu, select **Update from ISO**.



Afterward, you'll see the following message:

```
Using username "admin".
Linux P20-Appliance 4.9.0-8-amd64 #1 SMP Debian 4.9.144-3.1 (2019-02-19) x86_64
Last login: Mon Sep 23 14:21:17 2019 from 10.101.101.143
Make sure the update iso is connected to the VM, press enter to continue
```

3. After pressing **Enter**, you'll be prompted to confirm the update.

```
# # # #
```

Once the update is initiated, please wait for it to complete. Press **Enter** when finished to return to the menu.

For optimal performance, reboot the Login Enterprise Virtual Appliance after the update.

Your Login Enterprise Appliance is now updated. Remember to update/upgrade all existing Launchers after each Appliance update.

Offline update without a virtual CD-ROM drive

In some cases, certain hypervisors may not offer a virtual CD-ROM device to attach the ISO file to. However, you can still manually perform the update by copying the ISO file into the Appliance itself.

1. Using an SFTP client (such as WinSCP, ensuring you specify the SFTP protocol), upload the Login Enterprise update ISO to the `/tmp` directory.
2. Log in to the Appliance as the `admin` account via SSH or the console, and navigate to the **Troubleshooting > Open bash shell** menu entries, to get to the Linux Bash prompt.
3. Create the mount directory if it does not already exist:

```
mkdir /mnt/cdrom
```

4. Mount the ISO you uploaded.

```
mount -o loop /tmp/enterprise-update.iso /mnt/cdrom
```

5. Once that file is mounted, run the update process manually with this command:

```
loginvsid update offline
```

6. Reboot the Appliance. After the reboot, the ISO file will no longer be mounted, and you can delete it from `/tmp` to save space. (The `/tmp` folder is not special, you could instead choose to upload it to any other unprotected folder, such as `/home/admin`).

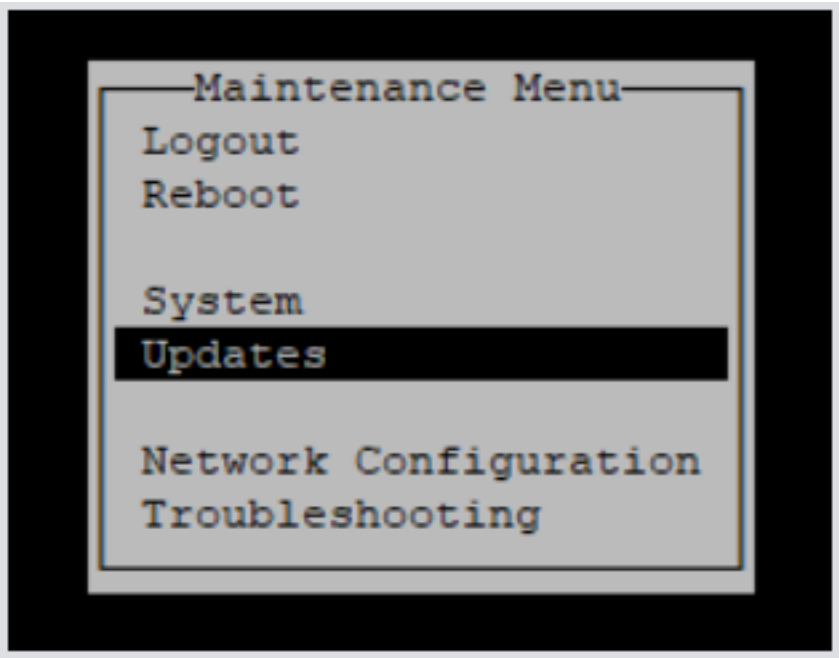
System update of the Login Enterprise operating environment

Within Login Enterprise, there's a built-in feature in the update menu called 'System update from the Internet.' This option will download and install security patches and other software updates for all installed Linux/Debian software. We recommend integrating this into your default update schedule for infrastructure servers.

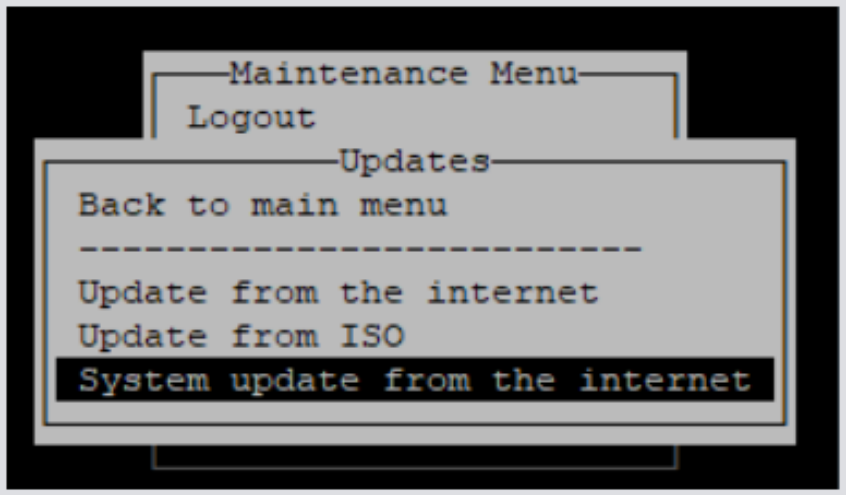
NOTE: It will not perform the major Debian 10 to Debian 12 upgrade. That must be done through a full Login Enterprise upgrade.

To utilize this update, navigate to the Virtual Appliance and log in until you see the **Maintenance Menu**.

1. In the **Maintenance Menu**, select **Updates**.



2. In the **Updates** menu select **System update from the internet**.



Upon selection, it will promptly begin downloading and installing the latest updates from the internet.

```
Get:26 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 libglib2.0-data all 2.50.3-2+deb9u1 [2,518 kB]
Get:27 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 libglib2.0-bin amd64 2.50.3-2+deb9u1 [1,615 kB]
Get:28 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 libglib2.0-0 amd64 2.50.3-2+deb9u1 [2,691 kB]
Get:29 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 libicu57 amd64 57.1-6+deb9u3 [7,705 kB]
Get:30 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 linux-image-4.9.0-11-amd64 amd64 4.9.189-3 [39.2 MB]
Get:31 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 linux-image-amd64 amd64 4.9+80+deb9u9 [7,132 B]
Get:32 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 rsync amd64 3.1.2-1+deb9u2 [393 kB]
Get:33 https://snapshot.debian.org/archive/debian/20190923 stretch/main amd64 unzip amd64 6.0-21+deb9u2 [172 kB]
Fetched 152 MB in 22s (6,756 kB/s)
apt-listchanges: Reading changelogs...
```

After completion, press **Enter** to return to the **Maintenance Menu**.

For optimal performance, reboot the Login Enterprise Virtual Appliance after the update completes.

Congratulations, you have now successfully updated the Appliance system.

Disk cleanup in the Appliance upgrade process

This feature is designed to address excessive memory usage on the Appliance disk. Two primary factors contribute to this issue:

- **Old Docker images:** These may remain in storage even after they are no longer used.
- **Unused blocks on a mounted filesystem:** These can accumulate over time and take up space.

To prevent unnecessary memory usage, the cleanup of old Docker images is performed during each Virtual Appliance update, just before you're prompted to confirm the Debian update.

```
Extracting updated binaries, this might take a while
Stopping LoginVSI services...[DONE]
Clearing deprecated messages and queues
Removing old Docker images
```

The second issue, unused blocks on the mounted filesystem, can be addressed using the **fstrim** command. This command trims the disk and discards any blocks that are no longer in use by the mounted filesystem.

NOTE: The underlying device must support physical discard operations for this to work. To verify this, the value in the /sys/block/{device_name}/queue/discard_max_bytes file must not be zero. The script used for this implementation will perform this check before executing any trim operation.

IMPORTANT: Executing fstrim too frequently may negatively impact the lifespan of lower-quality SSD devices. For most desktop and server systems, a trimming frequency of once a week is generally sufficient.

Changing the schedule

The script checks for support of discard operations and then enables a system timer for weekly trimming (by default, this is scheduled for Mondays at 12 PM UTC) if it is not already enabled.

If you prefer a custom schedule, you can modify the timing by editing the **fstrim.timer** file:

```
nano /lib/systemd/system/fstrim.timer
```

```
  GNU nano 3.2                                  /lib/systemd/system/fstrim.timer

[Unit]
Description=Discard unused blocks once a week
Documentation=man:fstrim

[Timer]
OnCalendar=Sat 23:00
AccuracySec=1h
Persistent=true

[Install]
WantedBy=timers.target
```

```
root@E02-APPLIANCE-RC:/loginvsi# systemctl status fstrim.timer
● fstrim.timer - Discard unused blocks once a week
   Loaded: loaded (/lib/systemd/system/fstrim.timer; enabled; vendor preset: enabled)
   Active: active (waiting) since Tue 2022-11-29 18:49:10 UTC; 4s ago
  Trigger: Sat 2022-12-03 23:00:00 UTC; 4 days left
     Docs: man:fstrim
```

The timer is set up during the initial-run startup, ensuring it applies both after updates and for new deployments.

If the timer has not been enabled, the script will execute the first trim operation itself. This is done to eliminate concerns that the initial scheduled trim might interfere with testing results, as it could execute while testing is in progress. By running the first trim when no services are active, subsequent scheduled trims can complete more quickly.

```
Cleaning up deprecated files
Discarding unused filesystem blocks
Setting timer for weekly cleanup of unused filesystem blocks
Login Enterprise started
Update completed
```

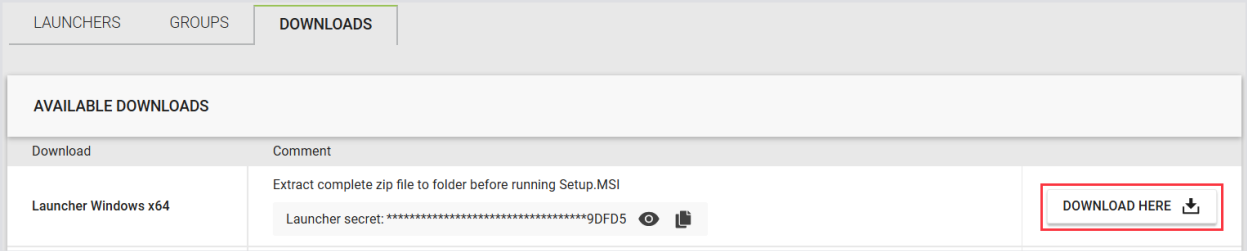
This feature operates quietly, meaning the process is fast, runs in the background, and requires no user interaction. The only indications that cleanup has been performed are terminal messages and entries in the log files. The response text from the cleanup of old Docker images is logged in **loginvsi/update.log**, while the output from filesystem trimming can be found in **loginvsi/startup.log**.

```
  GNU nano 3.2                                        startup.log

Creating service VSI_reportsworker
Creating service VSI_contentdelivery
Creating service VSI_systemconfigurationDb
Creating service VSI_identityserver
Creating service VSI_publicapi
Creating service VSI_systemworker
Creating service VSI_sessionmetricstatisticsDb
/: 89.5 GiB (96054403072 bytes) trimmed on /dev/sda1
Created symlink /etc/systemd/system/timers.target.wants/fstrim.timer → /lib/systemd/system/fstrim.timer.
● fstrim.timer - Discard unused blocks once a week
   Loaded: loaded (/lib/systemd/system/fstrim.timer; enabled; vendor preset: enabled)
   Active: active (waiting) since Mon 2022-12-05 07:04:28 UTC; 27s ago
  Trigger: Mon 2022-12-12 00:00:00 UTC; 6 days left
     Docs: man:fstrim

Dec 05 07:04:28 qa-4910 systemd[1]: Started Discard unused blocks once a week.
```

Updating Launchers

Launchers do not update automatically. You must manually update the Launchers, which is a straightforward task. We strongly recommend updating the Launchers after each Appliance update.

1. Disable all active environments to prevent them from launching sessions during the update process.
2. Log in to the Launcher machine and close the 'Login Enterprise Launcher' application.
3. Once completed, you can download the new Launcher files from the **Login Enterprise sidebar menu > Launchers > Download Launcher setup**, e.g. https://myvirtualappliance/configuration/launchers location.



4. After downloading, extract the ZIP file and run the 'Setup.msi'.
5. Once installed, you can enable the environments to resume operation, and the Appliance will continue from where it left off.

For more detailed information, see Launcher life cycle and best practices.

Updating the Appliance from v5 to v6

- Overview
- Upgrade limitations
- Steps to update to v6
- Steps after upgrading
- Role-based access control (RBAC)

Overview

This section outlines the update process for migrating from Login Enterprise v5 (5.0 to 5.14) to any v6 release.

IMPORTANT:

- Login Enterprise v5.8 and older can't be updated directly to v6. However, Login Enterprise v5.9.8 through v5.14.7 can be updated directly to v6 **if you initially installed one of those versions, or if you have performed the system update to Debian Linux 12 as part of the upgrade to one of those versions**.
- Before upgrading to v6, you must upgrade Debian to v12. This upgrade is available for Login Enterprise v5.9 and higher. For details, see Updating Debian on the Appliance.
- Once you are on Debian Linux 12, you can upgrade from Login Enterprise v5 to any v6 release.

1. Before updating, ensure you have a backup or snapshot of your Virtual Appliance. If you are using the optional SQL database for data storage, back it up as well.
2. Mount and attach the Login Enterprise v6 ISO to your existing Virtual Appliance via your virtualization management platform.

Upgrade Limitations

- When upgrading to Login Enterprise v6.3, all existing API tokens will become invalid and must be manually recreated. You can do this in the Login Enterprise UI. For steps, see Adding a System Access Token.
- Process tracking for custom connectors and secure custom fields is supported only when using v6.3 Launcher
- Launchers from previous versions of Login Enterprise (before v6.0) are not compatible with v6. Therefore, all Launchers must be upgraded when you upgrade to Login Enterprise v6. This only applies when upgrading from v5 to v6
- The upgrade to Login Enterprise v6 and higher will be blocked under the following circumstances:
    - If you're on Login Enterprise v5.8 or lower
    - If you're on Login Enterprise v5.9–5.14 but still using Debian 10
- In these cases, you'll need to perform specific actions before upgrading:

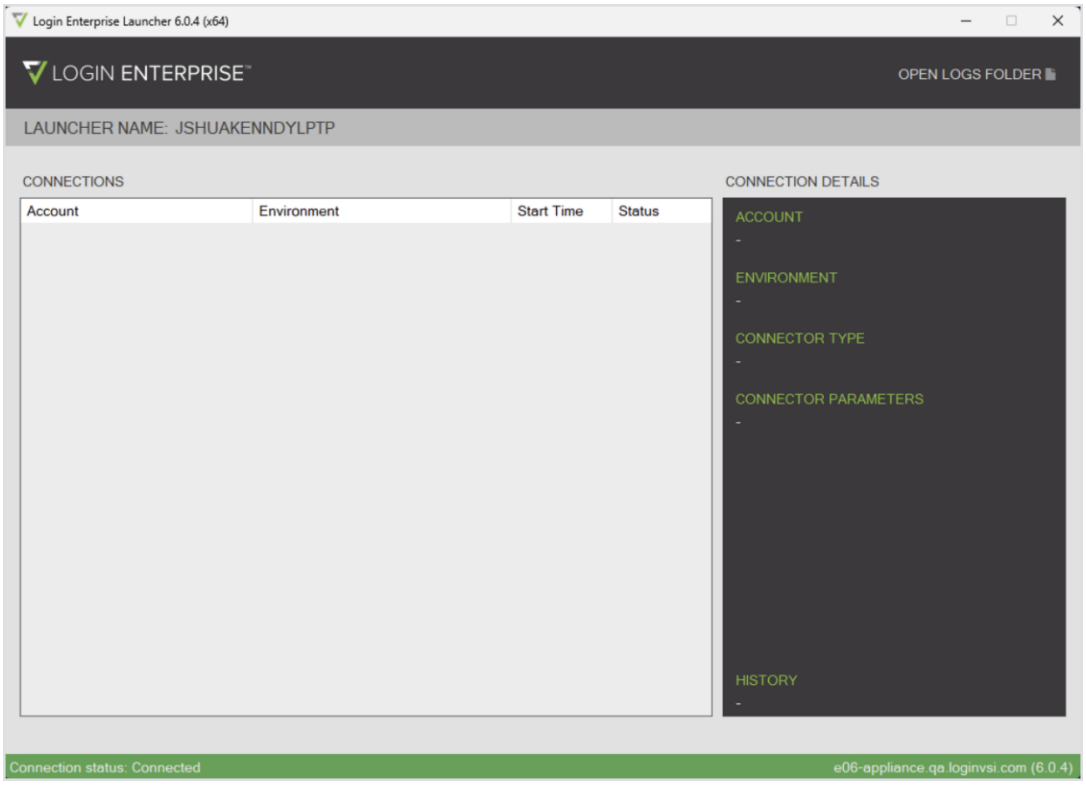
    - **For v5.8**: You must update Linux before upgrading. To do this, perform an upgrade to v5.14 and select "Perform System Upgrade"
    - **For v5.9 without Debian 12**: You must update Linux before upgrading. Upgrade to v5.14 and select "Perform System Upgrade". If you're already on v5.14, perform the upgrade to v5.14 again with the "System Upgrade" option
- If you currently log in using an LDAP account with the format `DOMAIN\USERNAME` (UNC format), please note that this is no longer supported in versions 6.0 and higher. You must use the format `USERNAME@DOMAIN` (UPN format) to log in

Steps to Update to v6

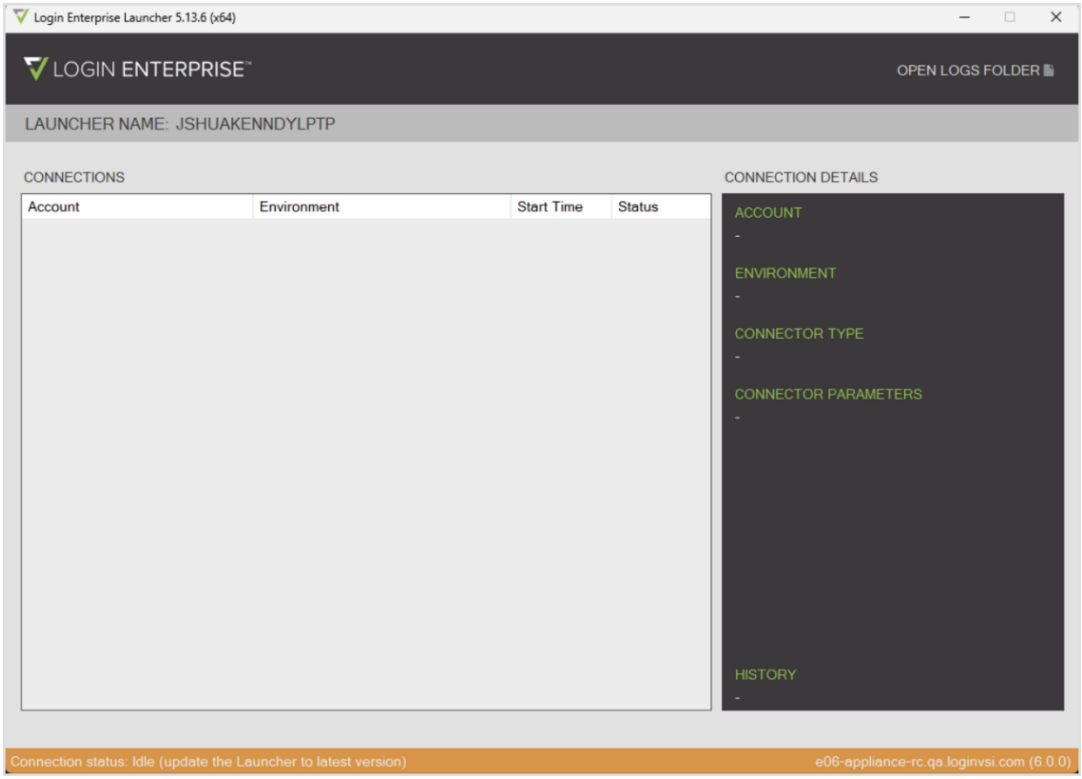
For detailed steps, see the Online Update of Login Enterprise.

Steps after Upgrading

- All Login Enterprise Launchers connected to your Login Enterprise Virtual Appliance will need to be updated. For details, see Launcher Maintenance and Updates.
    - The Login Enterprise v6 Launcher needs to be downloaded from your Login Enterprise. For details, see Downloading and Installing the Launcher.
    - To update the Login Enterprise Launcher agent on each Launcher host, the Launcher installer/updater needs to be run on each by either:
        - Silent installation. For details, see Silent Installation.
        - Graphical Launcher installation, invoked by running the Launcher installer (double-clicking on it if in the Downloads folder, for example) and following the installation prompts in the installation interface.
- To verify that the new Launcher installations are successful, run the Launchers, and the Launcher agent UI will look like this, with the new version showing in the top window banner (*Login Enterprise Launcher v6.0.4* in this example), and the bottom a green banner reading "*Connection status: Connected.*"



- If a Launcher hasn't been updated yet, the bottom orange banner would read "*Connection status: Idle (update the Launcher to the latest version)*".



Role-Based Access Control (RBAC)

Login Enterprise v6.0 introduces the first version of role-based access control (RBAC). Currently, RBAC is only supported with LDAP (Active Directory) authentication and requires LDAP users and groups for access management.

If you have LDAP set up, you can enable RBAC. Without LDAP integration, RBAC cannot function. To learn more about RBAC, use cases, permission dependencies, and more, see Role-Based Access Control (RBAC).

Updating Debian on the Appliance

- Updating Debian 9 to 10
  - Online update (Strongly recommended)
  - Offline update
- Updating Debian 10 to 12
  - Isolated/Offline environments
- Troubleshooting
  - Handling missing packages in Login Enterprise updates

Updating Debian 9 to 10

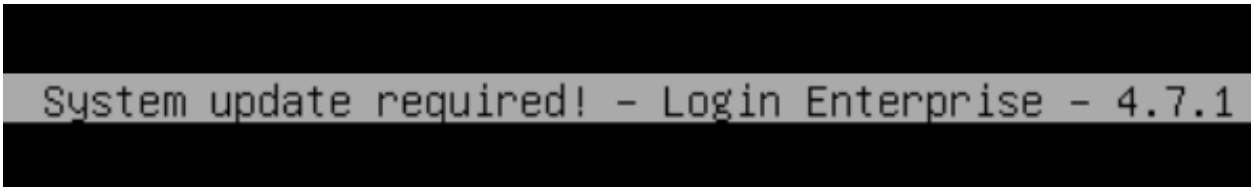
With the release of Login Enterprise 4.7, we recommend updating your Debian version to 10.

NOTE: This only applies to Login Enterprise Appliances, upgraded to 4.7 from 4.6 or lower. New installations using 4.7 and higher will have Debian 10 already installed. Older versions 4.6 and lower can't update to Debian 10.

Verification

To check if you need to upgrade to Debian 10, open an SSH or console connection to your appliance and log in to the console.

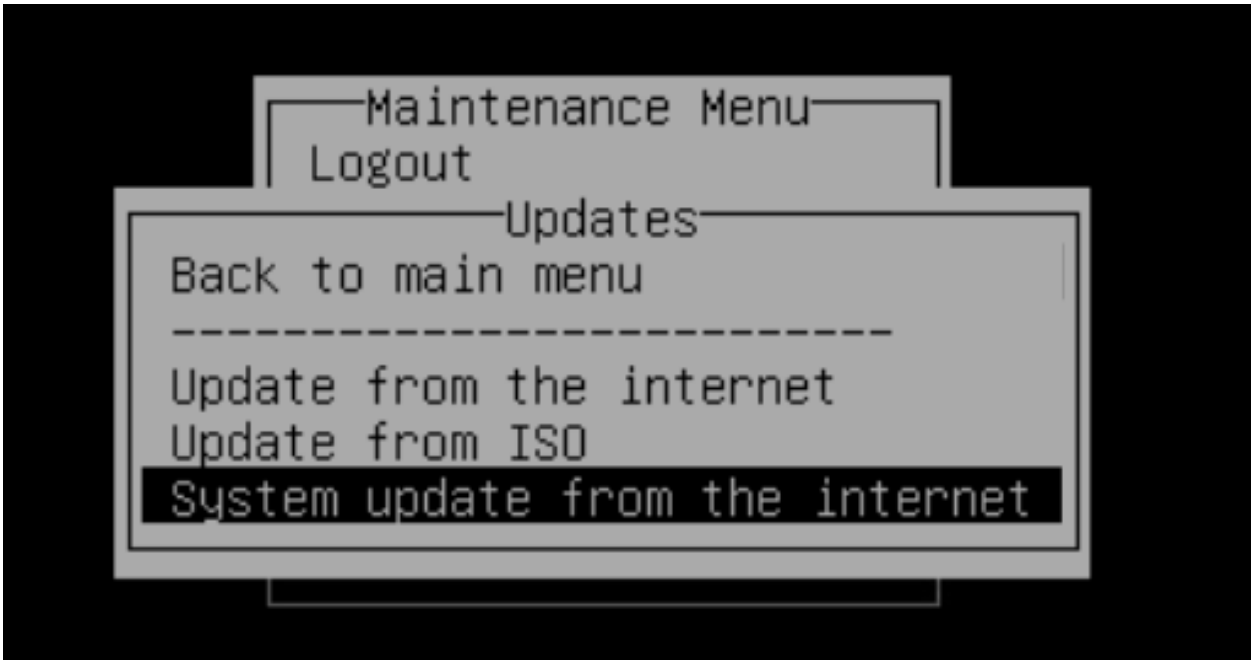
At the top of the screen, you can see a message stating, "System Update Required!"



NOTE: If you do not see this, no update is required.

Online update (Strongly recommended)

The update sequence is very simple and straightforward when your appliance is connected to the internet. Go to the **Updates** section on your Appliance console and select **System update from the internet**.



When initiated you are asked for verification of the update.

```
WARNING: YOU ARE ABOUT TO UPDATE THE OPERATING SYSTEM
MAKE SURE YOU HAVE A SNAPSHOT/BACKUP OF THE SYSTEM BEFORE PROCEEDING
ARE YOU SURE? [y/N]_
```

Once the update is initiated, Debian will start downloading the necessary files to complete the update. This process can take about 10 minutes, depending on internet speed and hardware.

When the update process is completed, you are asked to press **Enter**.

When you log out and log in to the console again, you can see that the message at the top of the screen is gone.

Offline update

With the release of 4.7, you can do an offline update to Debian 10. This happens during the upgrade to Login Enterprise 4.7.

The update process is similar to the default update from iso process as described in a different section. The only difference is that after the Important Notice, you are shown a different message that explains the importance of the Debian upgrade.

```
# # # #
```

```
root@loginenterprise:/home/admin# lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description:    Debian GNU/Linux 10 (buster)
Release:        10
Codename:       buster
root@loginenterprise:/home/admin#
```

Updating Debian 10 to 12

Since the 4.7 release, Login Enterprise has been running on Debian 10. However, Debian 10's LTS period ends on June 30th, 2024. For details, see Debian Long-Term Support Matrix.

Starting with the 5.9 release, new Login Enterprise installations will come with Debian 12 by default. Existing Login Enterprise installations (4.7-5.8) can upgrade Debian OS from 10 to 12 by updating to the latest Login Enterprise release.

NOTE: To receive the latest security updates on your System, we strongly advise upgrading to Debian 12 after the Login Enterprise upgrade is completed.

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#                        SYSTEM UPDATE RECOMMENDED                            #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#                                                                             #
#  Login Enterprise appliance system update to Debian 12 is available.       #
#  This major OS update can only be performed via the Login Enterprise       #
#  update ISO.                                                                #
#                                                                             #
#  Please note that Debian 10's LTS period ends on June 30th, 2024.          #
#  To receive the latest security updates on your System, we strongly        #
#  advise upgrading to Debian 12 after the Login Enterprise update           #
#  completes.                                                                 #
#                                                                             #
#  To proceed with the Debian 12 update after the Login Enterprise update,   #
#  press 'Enter' or answer 'Yes' to the prompt below.                         #
#                                                                             #
#  If you wish to remain on the current version of Debian, answer 'No'.      #
#  When you're ready to update to Debian 12, rerun the Login Enterprise      #
#  update from the latest update ISO.                                         #
#                                                                             #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
Perform system update from ISO after Login Enterprise update completes? [Y/n]
```

Isolated/Offline environments

The Login Enterprise upgrade ISO contains all the necessary packages to facilitate a Debian 10 to 12 upgrade in an offline scenario—no internet connectivity is required.

Troubleshooting

Handling missing packages in Login Enterprise updates

Starting with Login Enterprise 5.10, the system will detect if the Debian package manager cannot locate or download the required package. In such cases, the appliance will:

- Provide information on how to supply the missing package.
- Prompt the operator to decide whether to continue with the update.

This scenario is likely if your appliance is behind a firewall that restricts access to external systems.

Manually retrieving missing packages

If automatic package updates are not possible, you can manually retrieve the missing packages, upload them to the appliance, and then retry the update. This process should resolve the issue.

Error message example

When you run a Login Enterprise update and choose to include a system update, the process may fail with messages similar to the following:

```
After this operation, 883 MB of additional disk space will be used.
Err:1 https://deb.debian.org/debian bookworm/main amd64 libtcl8.6 amd64
8.6.13+dfsg-2
Could not connect to debian.map.fastly.net:443 (146.75.30.132). - connect
(113: No route to host) Unable to connect to deb.debian.org:https:
Err:2 https://deb.debian.org/debian bookworm/main amd64 tcl8.6 amd64
8.6.13+dfsg-2
Unable to connect to deb.debian.org:https:
Err:3 https://deb.debian.org/debian bookworm/main amd64 tcl-expect amd64
5.45.4-2+b1
Unable to connect to deb.debian.org:https:
Err:4 https://deb.debian.org/debian bookworm/main amd64 expect amd64 5.45.4-
2+b1
Unable to connect to deb.debian.org:https:
Err:5 https://deb.debian.org/debian bookworm/main amd64 javascript-common all
11+nmu1
Unable to connect to deb.debian.org:https:
Err:6 https://deb.debian.org/debian bookworm/main amd64 libxapian30 amd64
1.4.22-1
Unable to connect to deb.debian.org:https:
Err:7 https://deb.debian.org/debian bookworm/main amd64 psmisc amd64 23.6-1
Unable to connect to deb.debian.org:https:
Download complete and in download only mode
```

```
E: Failed to fetch
https://deb.debian.org/debian/pool/main/t/tcl8.6/libtcl8.6_8.6.13+dfsg-
2_amd64.deb Could not connect to debian.map.fastly.net:443 (146.75.30.132). -
connect (113: No route to host) Unable to connect to deb.debian.org:https:
E: Failed to fetch
https://deb.debian.org/debian/pool/main/t/tcl8.6/tcl8.6_8.6.13+dfsg-
2_amd64.deb Unable to connect to deb.debian.org:https:
E: Failed to fetch https://deb.debian.org/debian/pool/main/e/expect/tcl-
expect_5.45.4-2+b1_amd64.deb Unable to connect to deb.debian.org:https:
E: Failed to fetch
https://deb.debian.org/debian/pool/main/e/expect/expect_5.45.4-2+b1_amd64.deb
Unable to connect to deb.debian.org:https:
E: Failed to fetch https://deb.debian.org/debian/pool/main/j/javascript-
common/javascript-common_11+nmu1_all.deb Unable to connect to
deb.debian.org:https:
E: Failed to fetch https://deb.debian.org/debian/pool/main/x/xapian-
core/libxapian30_1.4.22-1_amd64.deb Unable to connect to
deb.debian.org:https:
E: Failed to fetch
https://deb.debian.org/debian/pool/main/p/psmisc/psmisc_23.6-1_amd64.deb
Unable to connect to deb.debian.org:https:
Warning: The following package(s) couldn't be downloaded:
libtcl8.6_8.6.13+dfsg-2_amd64.deb MD5Sum:017076f7f43bc9eee3355b21ec42cd16
tcl8.6_8.6.13+dfsg-2_amd64.deb MD5Sum:02ca287da37704e9df6ee0b06bd479c4
tcl-expect_5.45.4-2+b1_amd64.deb MD5Sum:88db84877ab2d4711b173e16f88cb6c5
expect_5.45.4-2+b1_amd64.deb MD5Sum:52a0d8ad1f8c18df2dd31fe6e39bfd63
javascript-common_11+nmu1_all.deb MD5Sum:f045ec32a84a135b9c218a227a07ead0
libxapian30_1.4.22-1_amd64.deb MD5Sum:f0b31ea1fb3e186af9266aad1166130e
psmisc_23.6-1_amd64.deb MD5Sum:2a758475a05809ab3080de96f9478401
Check the log at /loginvsi/missing_packages. You may need to download these
packages manually and upload them to the "/var/cache/apt/archives" directory.
Do you want to proceed with the update? (Make sure you have a snapshot to
revert to) [y/N]:
```

Handling download errors

If multiple packages fail to download (as indicated by the `Err:` and `E:` lines in the error message example above), the system will detect these errors and prompt you at the end, asking whether you want to continue.

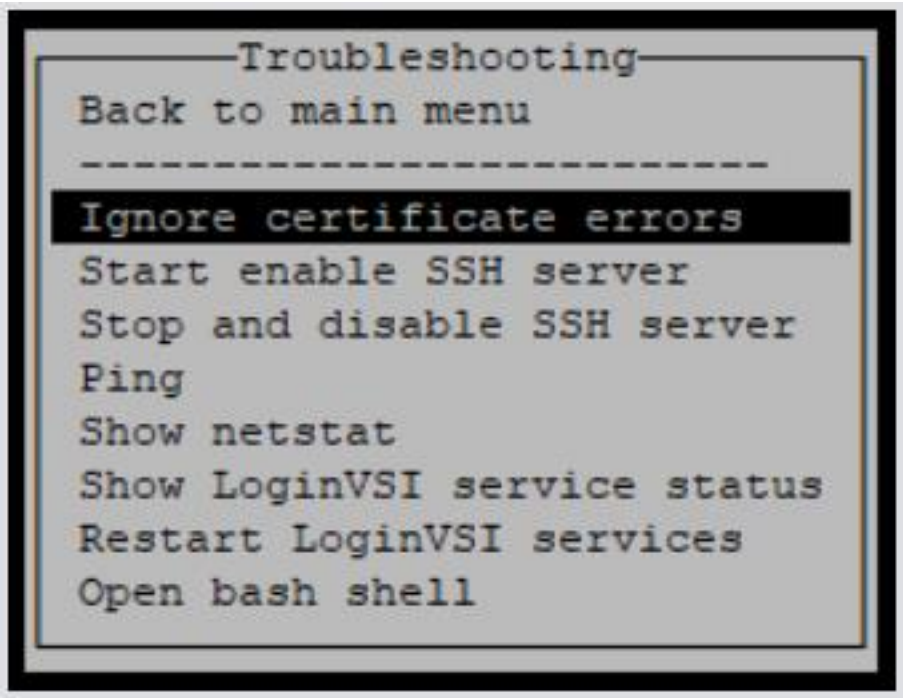
Unless you are certain it is safe to proceed (or you have a VM snapshot to revert to), you should respond with `n` and follow the instructions below to manually provide the missing packages to your appliance.

Common errors and solutions

- `Unable to connect to deb.debian.org.`
  - This error suggests that the appliance cannot reach the Internet at all, possibly due to basic connectivity issues or DNS errors like `Temporary failure resolving 'http://deb.debian.org'`

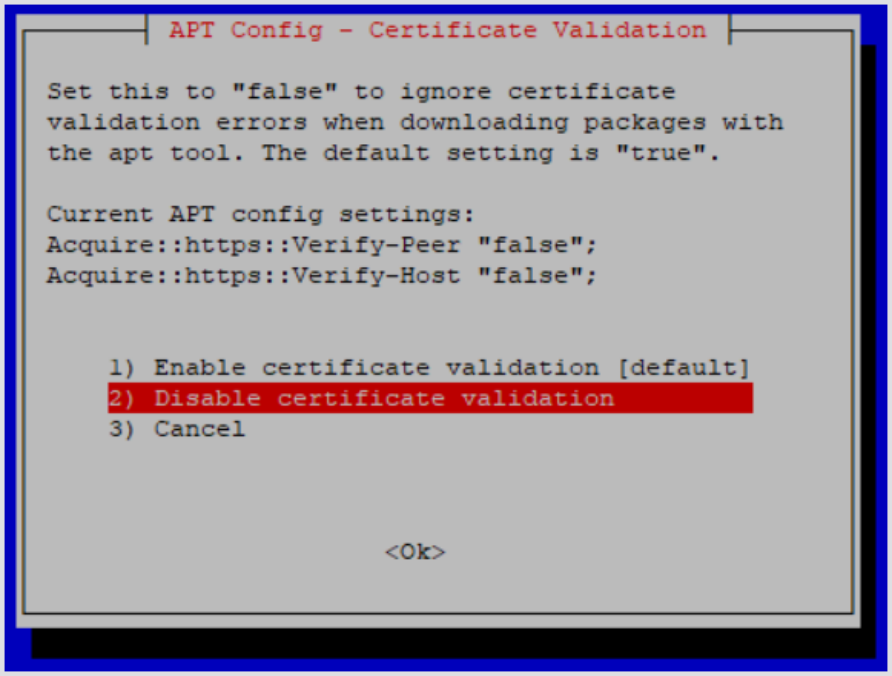
- `Certificate verification failed: The certificate is NOT trusted.`

- This error indicates that you can reach the Internet, but something interfered with the certificate on `deb.debian.org`, such as a packet-inspecting firewall.
- To resolve this:
  - Respond with `n` to continue the upgrade.
  - Go to the **Troubleshooting** menu and select **Ignore certificate errors**.



- Choose **Set apt config** to tell the Debian package manager to ignore certificate errors.

- Try the upgrade process again.



Manually downloading and uploading packages

If you cannot use the Internet to download additional packages directly to the appliance, follow these steps:

1. Download the required packages to a machine with Internet access.

2. Upload the packages to the appliance.

Once the packages are uploaded, you can safely rerun the upgrade process.

If the upgrade process fails, follow these instructions:

1. Check the log file at `/loginvsi/missing_packages`.
    1. This file contains failure information and URLs that Debian attempted to use.
    2. The log file will look similar to this:

```
'https://deb.debian.org/debian/pool/main/t/tcl8.6/libtcl8.6_8.6.13+dfsg-
2_amd64.deb' libtcl8.6_8.6.13+dfsg-2_amd64.deb 1034548
MD5Sum:017076f7f43bc9eee3355b21ec42cd16
'https://deb.debian.org/debian/pool/main/t/tcl8.6/tcl8.6_8.6.13+dfsg-
2_amd64.deb' tcl8.6_8.6.13+dfsg-2_amd64.deb 120328
MD5Sum:02ca287da37704e9df6ee0b06bd479c4
'https://deb.debian.org/debian/pool/main/e/expect/tcl-expect_5.45.4-
2+b1_amd64.deb' tcl-expect_5.45.4-2+b1_amd64.deb 133128
MD5Sum:88db84877ab2d4711b173e16f88cb6c5
'https://deb.debian.org/debian/pool/main/e/expect/expect_5.45.4-
2+b1_amd64.deb' expect_5.45.4-2+b1_amd64.deb 165996
MD5Sum:52a0d8ad1f8c18df2dd31fe6e39bfd63
```

```
'https://deb.debian.org/debian/pool/main/j/javascript-common/javascript-
common_11+nmu1_all.deb' javascript-common_11+nmu1_all.deb 6260
MD5Sum:f045ec32a84a135b9c218a227a07ead0
'https://deb.debian.org/debian/pool/main/x/xapian-core/libxapian30_1.4.22-
1_amd64.deb' libxapian30_1.4.22-1_amd64.deb 1100156
MD5Sum:f0b31ea1fb3e186af9266aad1166130e
'https://deb.debian.org/debian/pool/main/p/psmisc/psmisc_23.6-1_amd64.deb'
psmisc_23.6-1_amd64.deb 258696 MD5Sum:2a758475a05809ab3080de96f9478401
```

2. Download the log file using WinSCP or another SFTP client with the `admin` account.

3. Manually download the necessary packages listed in the log file.
    1. On any machine with Internet access, download each URL listed in the log file at `/loginvsi/missing_packages`.
    2. Remove any single-quote characters from the URLs.

4. Upload the packages to the appliance:
    1. Using WinSCP or another SFTP client, log in as `admin`.
    2. Change the remote directory to `/var/cache/apt/archives`.
    3. Upload all the DEB files you downloaded.

5. Rerun the upgrade process.

NOTE: For your system, the packages may vary and could be numerous.

Handling additional package discoveries

After downloading an initial set of files, Debian Linux may discover additional packages it did not initially identify. You may need to repeat this process. The previous DEB packages will still be present on the appliance, so you only need to download and upload the newly listed ones.

Continuing with missing packages

You can choose to continue with the upgrade process even if there are missing packages by responding with `y`. Most missing packages are for non-system components, and Linux can operate safely without upgrading them. At worst, specific programs may not function properly while the rest of the system is upgraded. If you have a snapshot to revert to, you can always attempt to continue the upgrade if you think the remaining packages are unimportant.

Back to Installation and Deployment

Configuration

This chapter outlines the steps to set up and configure the essential features of the Login Enterprise Virtual Appliance, including the Virtual Appliance itself, authentication, licensing, Launchers, Virtual Users, email notifications, event logging, and browser driver updates.

- Appliance
  - Initial Appliance Configuration
  - Managing Certificates
  - Configuring SSH Security and Ciphers
  - Configuring Authentication
    - Single Sign-On (SSO)
    - Configuring SAML
    - Configuring OpenID Connect (OIDC)
  - Role-Based Access Control (RBAC)
  - Managing Licenses
  - Using the Maintenance Menu
  - Getting Started with the Appliance Web Interface
  - Configuring Database Data Retention
  - Configuring External SQL Database
  - Expanding the Appliance Disk Space
  - Performing the Appliance Backup and Recovery
  - Monitoring Active Sessions
  - Monitoring Container Status
  - Monitoring Appliance Health
  - NVIDIA nVector Integration
  - Nutanix Platform Metrics Integration
- Launchers
  - Launcher Overview and Best Practices
  - Configuring the Windows Launcher
  - Configuring the Linux Launcher
  - Configuring the IGEL 12 Launcher
    - Launcher Connections in IGEL UMS 12
    - Deploying the Launcher through the IGEL App Portal
- Virtual Users
  - Configuring Logon Components
    - Configuring Logon Processor as a Logon Script
  - Managing Virtual User Accounts
  - Understanding Login Enterprise Engine
- Managing Notifications
- Configuring the Event Logger
- Updating Browser Drivers
- Using the Public API

Appliance

This subchapter provides comprehensive guidance on setting up and managing Login Enterprise.

- Initial Appliance Configuration
- Managing Certificates
- Configuring SSH Security and Ciphers
- Configuring Authentication
  - Single Sign-On (SSO)
  - Configuring SAML
  - Configuring OpenID Connect (OIDC)
- Role-Based Access Control (RBAC)
- Managing Licenses
- Using the Maintenance Menu
- Getting Started with the Appliance Web Interface
- Configuring Database Data Retention
- Configuring External SQL Database
- Expanding the Appliance Disk Space
- Performing the Appliance Backup and Recovery
- Monitoring Active Sessions
- Monitoring Container Status
- Monitoring Appliance Health
- NVIDIA nVector Integration
- Nutanix Platform Metrics Integration

Initial Appliance Configuration

- Overview
- Prerequisites
- Accessing the Virtual Appliance
- Logging in
- Initial configuration
- Network configuration
- Admin username and password update
- Finalizing configuration
- Daemon startup and Appliance restart
- Completion of configuration

Overview

This section provides step-by-step instructions for configuring the Login Enterprise Virtual Appliance for the first time. By following these instructions, you will set up essential parameters, such as hostname, network configuration, and administrative access. Upon successful configuration, you will be able to access the web interface and perform further maintenance tasks.

Prerequisites

Before configuring the Appliance, gather the following information:

- **Appliance Hostname**
- **Appliance Fully Qualified Domain Name (FQDN)**
- **IP Address** (not required if using DHCP)
- **Netmask (Subnet)** (not required if using DHCP)
- **Gateway Address** (not required if using DHCP)
- **DNS Address** (not required if using DHCP)
- **Search Domain (FQDN)**
- **Admin Username for Configuration**
- **Admin Password for Configuration**

Accessing Virtual Appliance

1. Power on the Virtual Appliance.
2. Open the console in a new window.

You will see a welcome message with the following login details:

- Username: admin
- Password: admin

Logging in

In the Login field, enter the default credentials, and press **Enter**:

```
Welcome to Login Enterprise - version 4.0.10
This system is not yet configured, please logon with username: admin and password: admin
login-ent login: admin
Password: _
```

Initial configuration

Once logged in, the system will prompt you to provide the following information:

1. **Preferred Hostname:** Enter your desired hostname. For example, loginenterprise
2. **Preferred Domain:** Enter your desired domain. For example, demo.lab
3. **Network Configuration Mode:** Choose the desired network configuration mode, DHCP (D) or Static(s)

Network configuration

The system will prompt you to configure the network settings based on your selection:

- **If you select Static:** You will be prompted to provide the following information:
  - **IP Address:** Enter the desired IP address.
  - **Netmask:** Enter the subnet mask.
  - **Default Gateway:** Enter the default gateway.
  - **Nameservers:** Enter the DNS server addresses.
  - **DNS Search Suffix:** Enter the DNS search suffix.

```
Configuring network using static address...
 Please provide the new ip address (10.100.10.168):
 Please provide the netmask (255.255.255.0):
 Please provide the default gateway (10.100.10.1):
 Please provide the nameservers (10.10.0.31 10.10.0.32)
 Please provide the dns search suffix (demo.lab)
 Please enter a new password for admin:
```

- **If you select DHCP:** The system will automatically configure the necessary network settings.

Admin username and password update

After completing the network configuration, you will be prompted to enter an updated admin username and password:

- **Admin username:** If "admin" is acceptable, simply press **Enter**. Otherwise, enter your desired username.
- **Admin password:** Enter a new password for the admin account.

```
Starting LoginVSI configuration tool...
Welcome to Login Enterprise - version 99.99.4
Please help answer the following questions to get the system setup...
 Please enter your computername (e.g login-ent):andrius-01
 Please enter your domain dns suffix (e.g contoso.com):qa.loginvsi.com
 Please choose a network configuration mode dhcp or static (D/s)
Configuring network using dhcp...

Please enter a name for admin user: admin
```

```
Please enter a name for admin user: andrius

Please enter a new password for andrius:
Please confirm the new password:
Password updated successfully
```

Finalizing configuration

The system will update the password and continue running. After a brief moment, you will be prompted to log in using the admin username and the password you set earlier.

```
        Welcome to Login Entperise - version 5.10.2
        This appliance is not yet configured!
        Logon with username and the password you set previously

login-ent login: _
```

When you do that, the system will perform several automatic configuration operations. It will then prompt you whether to create a new encryption key or enter one. The encryption key is only used to encrypt sensitive data in the configuration database, like passwords. Connections to the appliance and between the containers are handled over HTTPS with the applied SSL Certificate (self-signed or custom).

Normally, you would allow the system to auto-generate a new encryption key. You should only use an existing key if you plan to reconnect to an existing external MS SQL Server appliance database that already has its own encryption key, and you do not have access to the original appliance VM that created the database. In this case, you would manually type in the existing key. Otherwise, the system will generate a key for you. Be sure to screen-shot this key for future reference:

The system will automatically reboot. After it completes, you will see a screen like this on the console informing you that the Appliance is up and operational. If you have a certificate for the Appliance, you can add it now. Otherwise, you can leave the console and move to the appliance web UI.

```
   Welcome to Login Enterprise - version 5.10.2

   The Login Enterprise Webinterface is available at:

    https://login-ent.loginvsi.lab
    https://192.168.7.151

    NOTE: Please make sure to update your dns records accordingly

login-ent login: _
```

If you have not already added the DNS record for this host to your local DNS server, you should pause to do that now.

Daemon startup and Appliance restart

Once you have logged in, the system will start the Login VSI daemon and automatically restart the Appliance.

Completion of configuration

After a few minutes, you will see a welcome message indicating that the Appliance has been successfully configured. The message will include the following important information for accessing the web interface:

- **IP-Based URL**
- **FQDN URL**

```
   Welcome to Login Enterprise - version 4.0.10

   The Login Enterprise Webinterface is available at:

    https://loginenterprise.loginvsi.com
    https://10.101.99.50

    NOTE: Please make sure to update your dns records accordingly

loginenterprise login:
```

NOTE: Please create an A record for the host to access the FQDN URL.

You have now successfully configured the Virtual Appliance. The next step is to access the Maintenance Menu.

Managing Certificates on the Appliance

- Overview
- Default self-signed certificate
- Installing a new certificate
- Importing a root CA certificate
- Switching back to self-signed certificates
- Troubleshooting
  - Problems after applying a new certificate
  - Allowed URLs
  - Some certificate requirements
  - Ignoring certificate errors on Debian
    - Enabling "Ignore certificate errors" option
    - Ignoring certificate errors for apt
    - Ignoring certificate errors for curl
    - Disabling "Ignore certificate errors" option
    - Restoring certificate validation for apt
    - Restoring certificate validation for curl
  - Login Enterprise services not starting

Overview

Login Enterprise uses certificates to provide increased security as a standard part of its infrastructure.

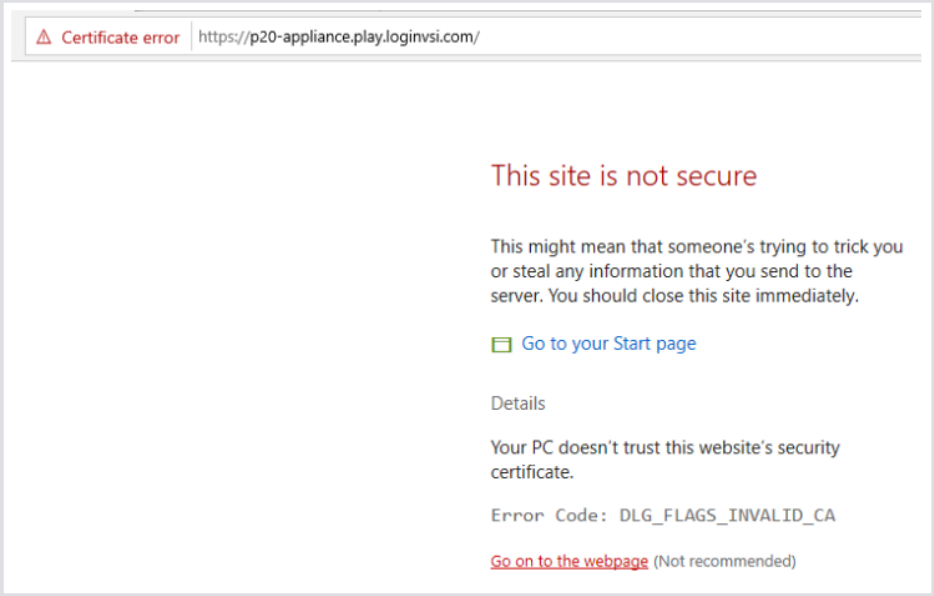
It is common for enterprises to use their own Certificate Authority (CA), and therefore, Login Enterprise lets you customize which certificates are used.

Certificates are used by browsers for the Web UI and internal communications between the launchers and the targets being tested. The launcher installs it automatically, and the target engine ignores SSL errors by default. However, the target engine stops ignoring SSL errors after importing your own certificate.

IMPORTANT: If you regenerate a certificate, install a new server certificate, or change the hostname, any installed Launchers must be reinstalled whenever your certificate changes.

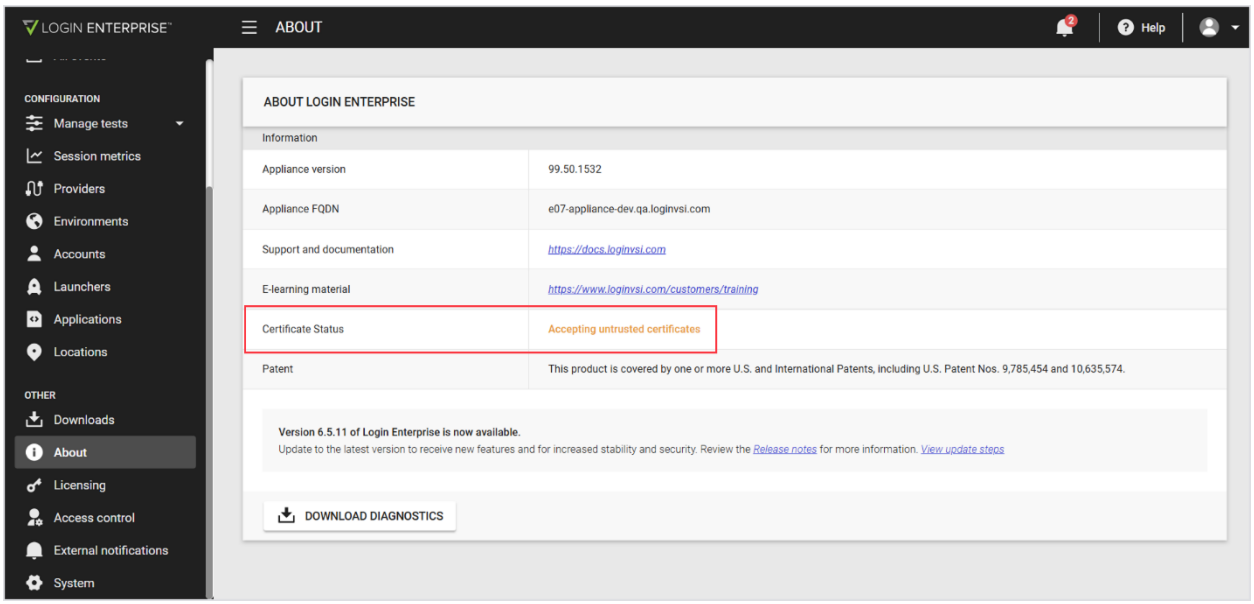
Default self-signed certificate

By default, Login Enterprise runs with a self-signed certificate on the Virtual Appliance. This lets you get Login Enterprise up and running quickly and requires you to proceed even though the certificate is untrusted.



To verify the certificate status:

In the **Login Enterprise sidebar menu**, navigate to **Other > About**:



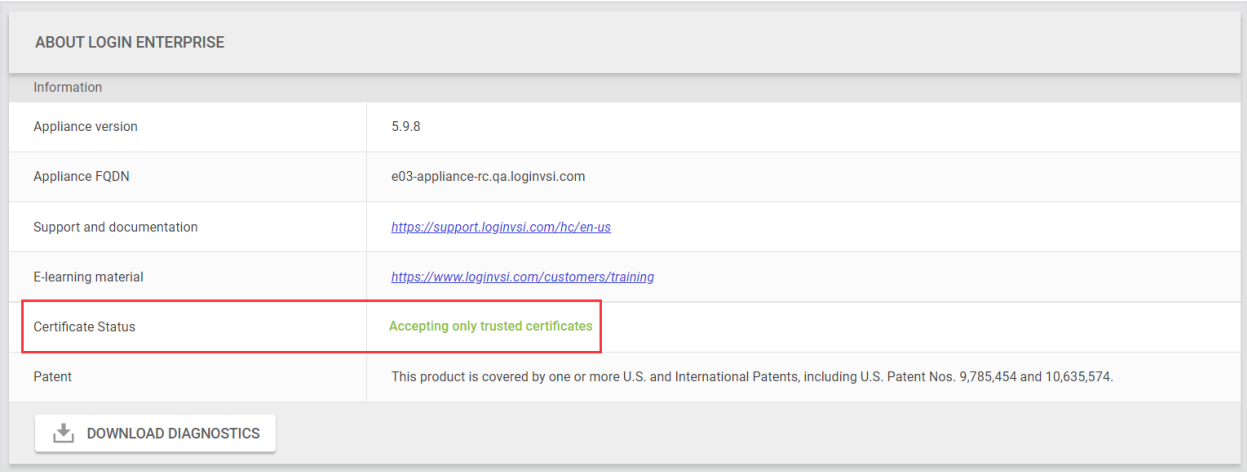
Installing a new certificate

If your new certificate was issued by an internal Enterprise or Standalone CA, we recommend importing the root CA certificate and any intermediary certificates before proceeding with the new certificate install. This will ensure that the certificate chain is trusted on the appliance. For more information, see Importing a root CA certificate. For the certificate requirements, see Encryption.

NOTE: The uploaded PFX may contain an RSA or ECC certificate with its matching key.

To install a new certificate:

1. Log in to the appliance console or use SSH to connect as `admin`. For more information, see the Maintenance menu.
2. Use WinSCP or an equivalent SFTP client to upload the `pfx` certificate to the `/certificates` folder in the root of the appliance, e.g. `/certificates/mycert.pfx`
   - If using WinSCP, select the SFTP protocol for the connection type
3. Select **Apply new certificate** from the appliance console menu.
   a. Specify the full path to the `pfx` file, including the filename, e.g. `/certificates/mycert.pfx`, and press **Enter**.
   b. Specify the password for the `pfx` file and press **Enter**.
4. This will automatically switch the product to use the Trusted SSL mode. To verify the certificate status, refresh the Web UI to see that it is trusted, or in the **Login Enterprise sidebar menu**, navigate to **Other > About**. The certificate status will be available in the Information section:

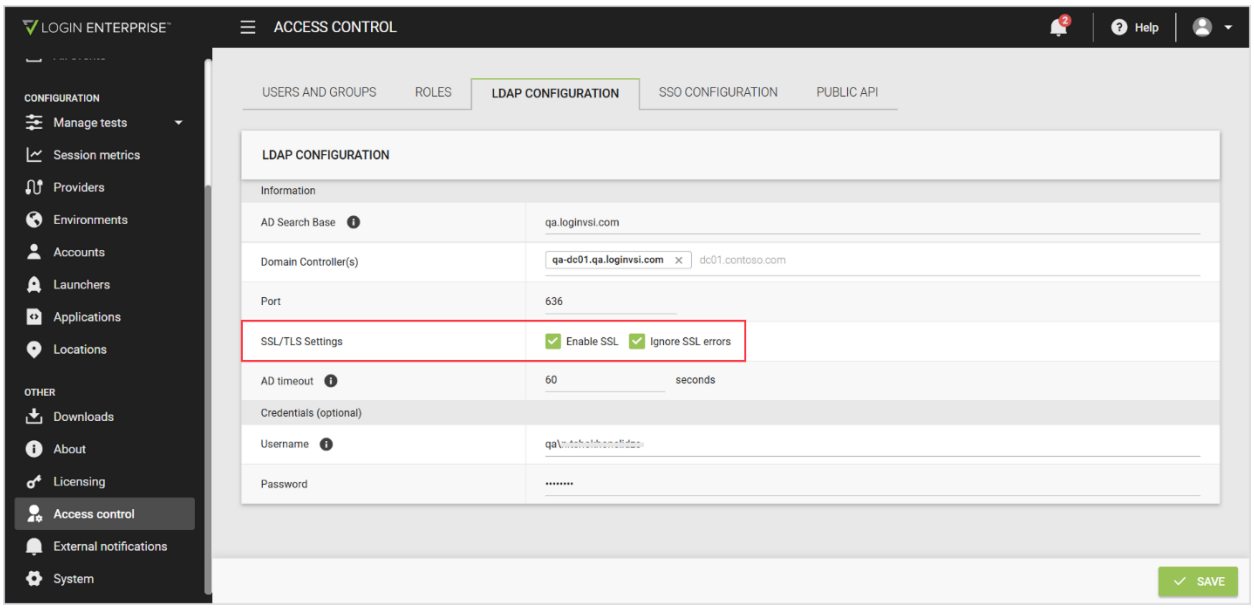


Importing a root CA certificate

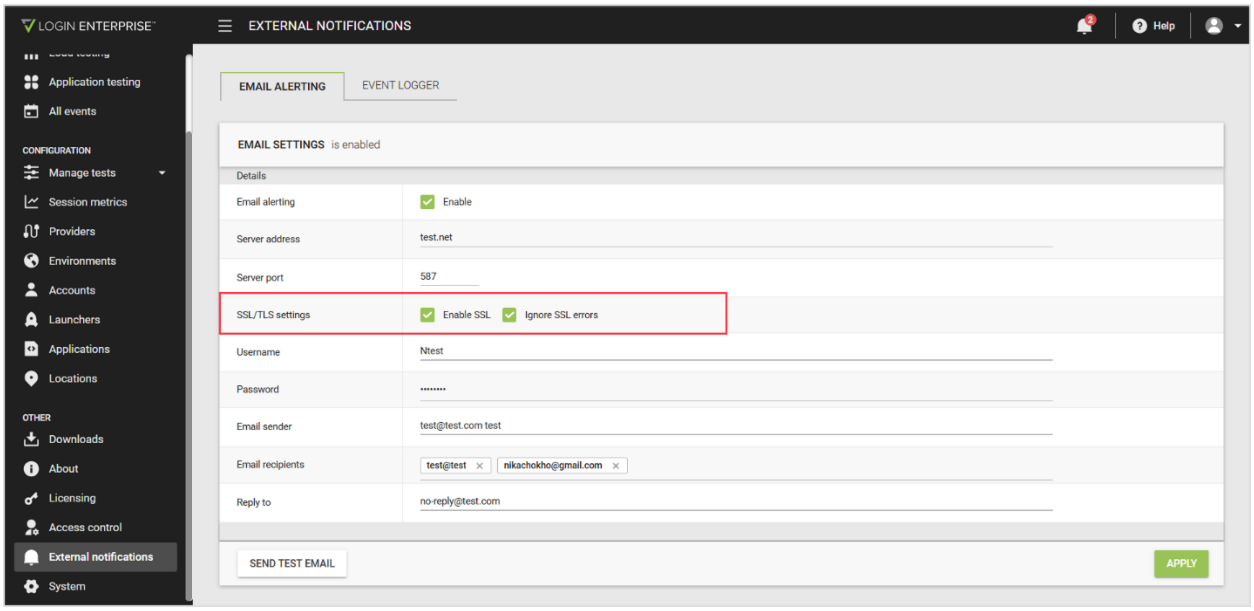
Login Enterprise lets you import root Certificate Authority (Root CA) certificates. This functionality is useful when you have an internal CA that generates certificates for your organization. Once a valid certificate is imported—if you use an internal CA—SSL can be enabled for the AD authentication, Email, and Database configuration settings.

There are 3 areas in the Login Enterprise where you can enable SSL:

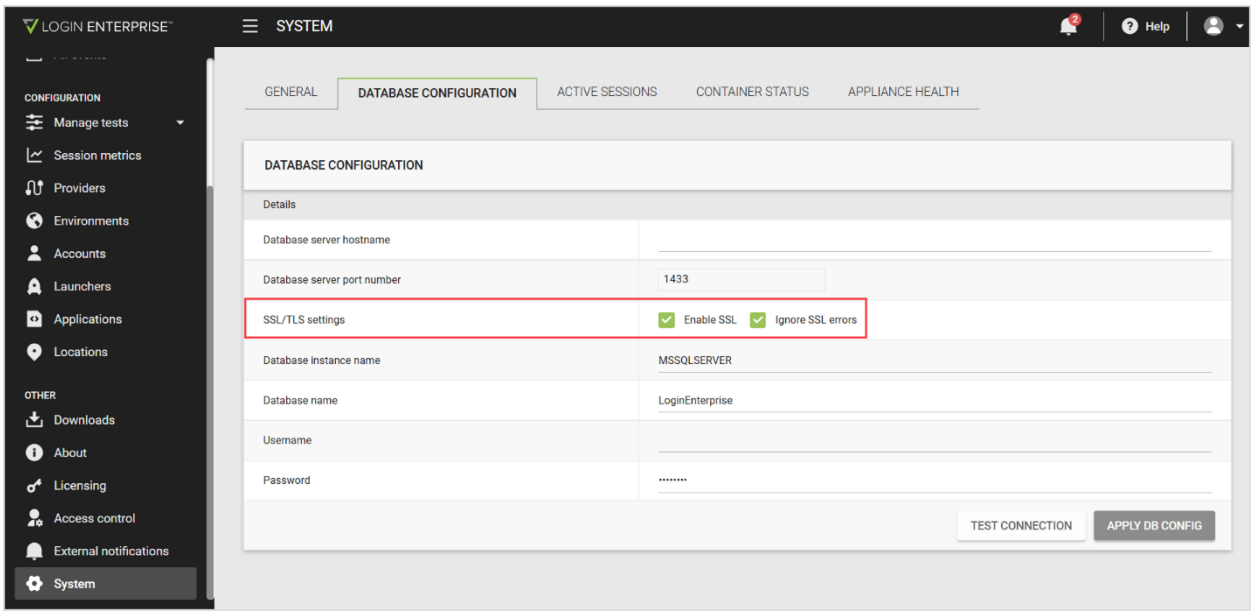
- **Access control > LDAP**



- **External notifications > Email alerting**

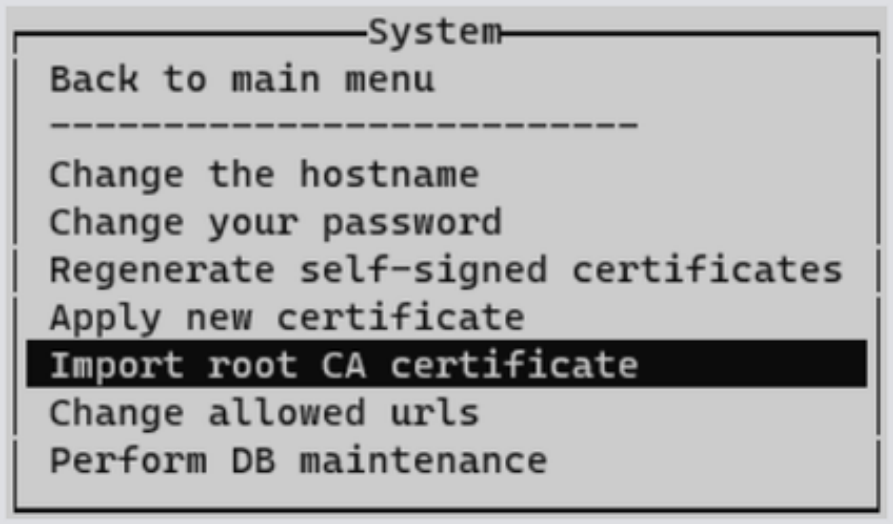


- **System > Database configuration**



To import a root CA certificate:

1. Use WinSCP or an equivalent SFTP client to upload a root CA certificate to the `/certificates` folder in the Appliance, e.g. `/home/admin/mycert.crt`
2. Log in to the appliance console. For more information, see the Maintenance menu.
3. Navigate to **Main > System > Import root CA certificate**.



4. Specify the full path to the newly uploaded `.crt` file, e.g. `/home/admin/myCert.crt`
5. Restart the Login VSI services when prompted to apply the changes.

```
Please enter the path to the root ca file (.crt): /home/admin/mycert.crt
Updating certificates in /etc/ssl/certs...
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
Are you sure you want to restart the LoginVSI services? [y/N]: y
Stopping LoginVSI services...[DONE]
Wed 30 Nov 2022 04:09:48 PM UTC Applying database changes. This will take a while...
Database migration completed.
Stopping migration services...[DONE]
Cleaning up deprecated files
Login Enterprise started

Press Enter to return to Pdmenu.
```

Switching back to self-signed certificates

At times certificate installation may fail. If this is the case and you'd like to reuse the self-signed certificate supplied with the appliance, take the following steps:

1. Log in to the appliance console. For more information, see the Maintenance menu.
2. Select **Regenerate self-signed certificates** from the appliance console menu.
3. This will automatically reset the product to accept any SSL certificate.

Troubleshooting

Problems after applying a new certificate

For problems after applying a new certificate, first make sure your root CA and any intermediate certificates are properly imported/installed on the appliance.

If the certificate being applied was issued by an External CA (DigiCert, Sectigo, GlobalSign, etc.), make sure your appliance public certificates are up to date. For this, run `apt install ca-certificates` from Bash.

If the certificate was issued by an Internal Enterprise or Standalone CA, ensure the certificates in the trust chain (root and intermediates) have been uploaded to the appliance in the `.pem` (base64) format with the `.crt` extension and imported through the **pdmenu > Import root CA certificate**.

```
System
Back to main menu
--------------------------
Change the hostname
Change your password
Regenerate self-signed certificates
Apply new certificate
Import root CA certificate
Change allowed urls
Perform DB maintenance
```

When a certificate is successfully imported, it should show 1 or more was “added.”

```
Updating certificates in /etc/ssl/certs...
rehash: warning: skipping ca-certificates.crt,it does not contain exactly one certificate or CRL
rehash: warning: skipping pfxcacerts.pem,it does not contain exactly one certificate or CRL
1 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
```

If it shows “0 added”, it could mean the certificate was already imported or that the certificate file extension was not `.crt`.

```
Updating certificates in /etc/ssl/certs...
0 added, 0 removed; done.
Running hooks in /etc/ca-certificates/update.d...
done.
```

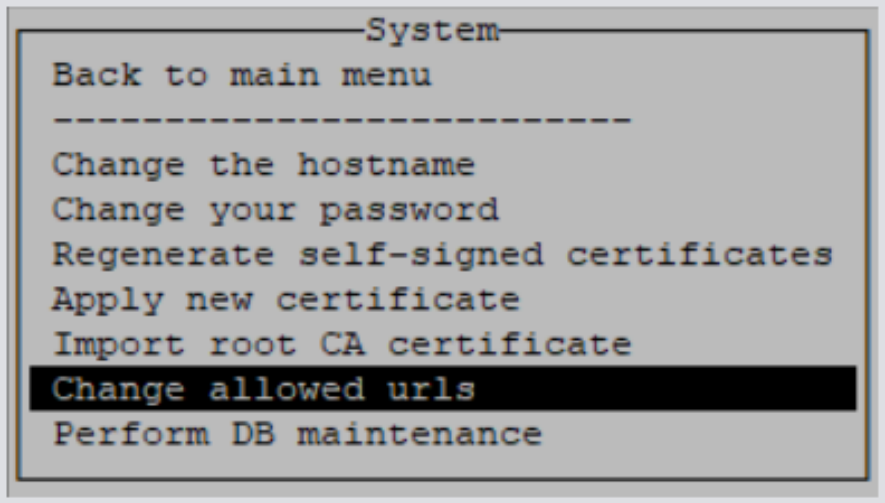
When using the **Import root CA certificate** option through **pdmenu**, the certificate will be copied to `/usr/local/share/ca-certificates/`, and a symbolic link to that file is created with the `.pem` extension in `/etc/ssl/certs/`. To verify that the certificates exist in the proper directory, run the commands below in Bash.

```
ls -la /usr/local/share/ca-certificates/
ls -la /etc/ssl/certs/ | grep "local/share"
```

```
root@RC-app-pool-1:/tmp# ls -la /usr/local/share/ca-certificates/
total 28
drwxr-xr-x 2 root root 4096 Apr 12 17:01 .
drwxr-xr-x 7 root root 4096 Apr 11 12:50 ..
-rw-r--r-- 1 root root 1545 Apr 11 13:23 CA.crt
-rw------- 1 root root 4810 Apr 11 19:06 pfxcacerts.crt
-rw-r--r-- 1 root root 1918 Apr 12 17:01 QA-DC01-CA.crt
-rw-r--r-- 1 root root 2190 Apr 12 17:01 QA-DC02.crt
root@RC-app-pool-1:/tmp# ls -la /etc/ssl/certs/ | grep "local/share"
lrwxrwxrwx 1 root root     39 Apr 11 13:23 CA.pem -> /usr/local/share/ca-certificates/CA.crt
lrwxrwxrwx 1 root root     47 Apr 12 17:01 pfxcacerts.pem -> /usr/local/share/ca-certificates/pfxcacerts.crt
lrwxrwxrwx 1 root root     47 Apr 12 17:01 QA-DC01-CA.pem -> /usr/local/share/ca-certificates/QA-DC01-CA.crt
lrwxrwxrwx 1 root root     44 Apr 12 17:02 QA-DC02.pem -> /usr/local/share/ca-certificates/QA-DC02.crt
```

Allowed URLs

The allowed URLs list contains a list of URLs that are allowed to be used to access the Login Enterprise and can be modified in the **System > pdmenu**. The first entry in the allowed URLs list should contain the FQDN of the certificate that was applied to the appliance. The appliance uses the first entry in the allowed URLs list as the ServerURL in the `appsettings.json` file for applications like the Launcher, Event logger, and appliance guard.



Some certificate requirements

Whatever the SANs of your certificate are, they must be:

1. DNS resolvable to your appliance, and
2. One of them must be the first entry in your ALLOWED_URLS key in the `/loginvsi/.env` file.

If your local hostname is not listed in the SAN field, and you can't regenerate the certificate to include it, consider changing the hostname to match the certificate. However, your ALLOWED_URLS, and any client attempting to access the appliance must use a name in the SAN field, including a wildcard certificate pattern match, if a SAN exists. Otherwise, it must match the Subject field.

If you change the hostname, make sure you reset pi_guard along with restarting services to get the containers recomposed.

```
systemctl restart pi_guard
loginvsid restart
```

Ignoring certificate errors on Debian

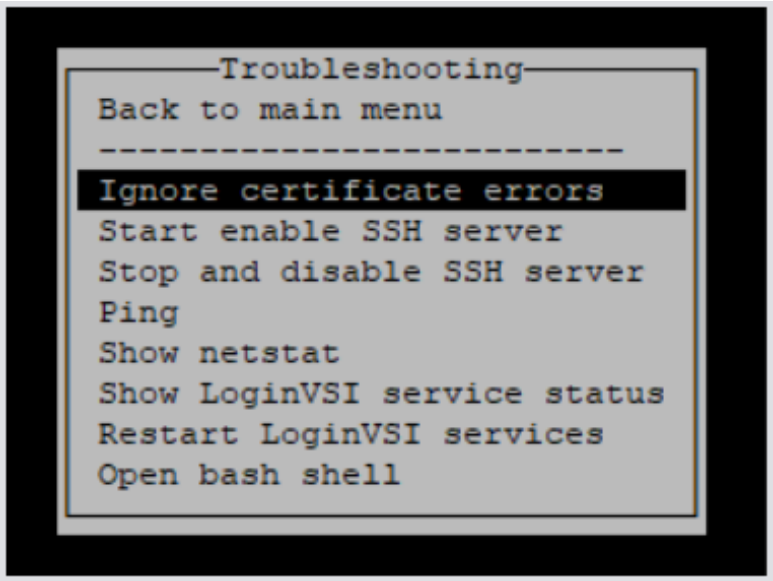
We understand that some customers operate in environments where SSL communications are inspected or intercepted, leading to certificate verification errors when downloading packages with `apt` or our ISO with `curl`. These errors can disrupt your workflow and make it difficult to manage software packages efficiently.

To address this, we've introduced a feature in the PDMenu that detects SSL certificate issues during the update process. When a certificate problem is identified, you'll be prompted to allow insecure connections if needed. This feature modifies the commands to ignore certificate errors, specifically by adding the `--insecure` flag to `curl` and adjusting the configuration for `apt`.

The following sections provide step-by-step instructions on how to enable and disable the "Ignore certificate errors" option for both `apt` and `curl`.

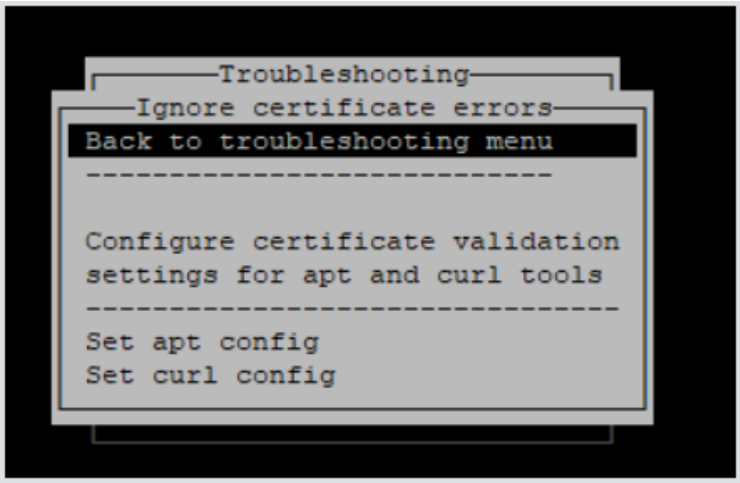
Enabling "Ignore certificate errors" option

1. **Open PDMenu**: Launch the PDMenu from your application menu or by running `pdmenu` from the terminal.
2. **Navigate to the Troubleshooting section**: Use the arrow keys to select `Troubleshooting` and press `Enter`.
3. **Select "Ignore certificate errors"**: Find the option labeled "Ignore certificate errors" and press `Enter`.



4. **Configure certificate validation settings**.

- You will see the following submenus:
  - `Set apt config`
  - `Set curl config`



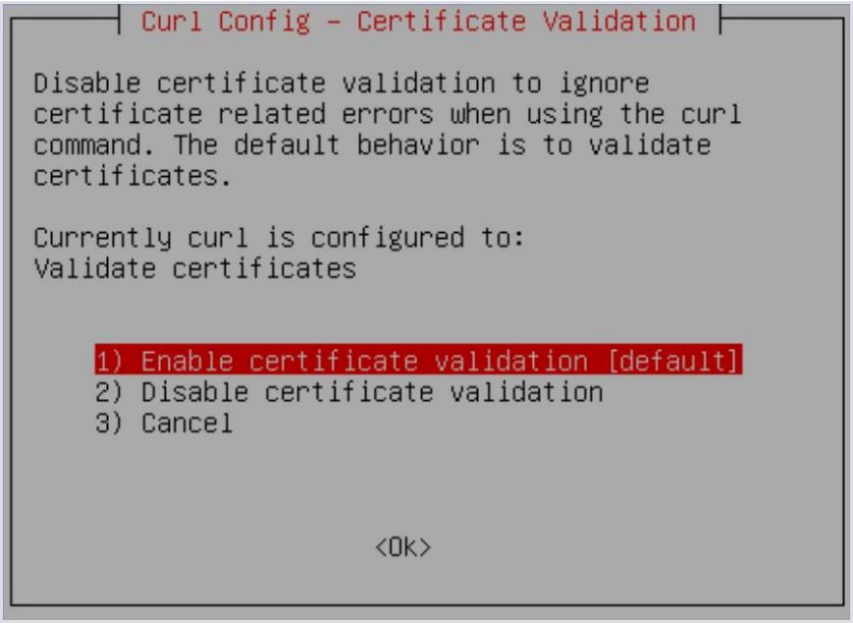
Ignoring certificate errors for `apt`

1. **Select "Set apt config"**: Use the arrow keys to highlight `Set apt config` and press `Enter`.
2. **Enable ignoring certificate errors**: Follow the on-screen prompts to configure `apt` to ignore certificate errors. This will add the necessary configuration to the `apt` settings.

After configuring `apt` to ignore certificate errors, you can verify the changes by performing operations that involve SSL connections. In cases where a certificate is expired, untrusted, or cannot be validated, appropriate error messages will be displayed, confirming that certificate validation is enabled.

Ignoring certificate errors for `curl`

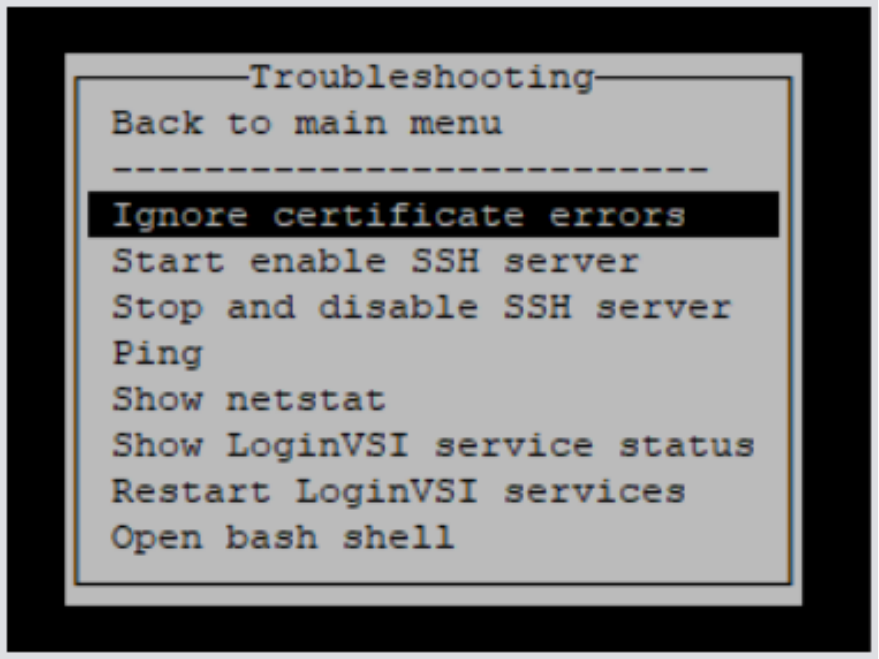
1. **Select "Set curl config"**: Use the arrow keys to highlight `Set curl config` and press `Enter`.
2. **Enable ignoring certificate errors**: Follow the on-screen prompts to configure curl to use the `--insecure flag`. This will adjust the `curl` command settings accordingly.



After configuring `curl` to ignore certificate errors, you can verify the changes by performing operations that involve SSL connections. In cases where a certificate is expired, untrusted, or cannot be validated, appropriate error messages will be displayed, confirming that certificate validation is enabled.

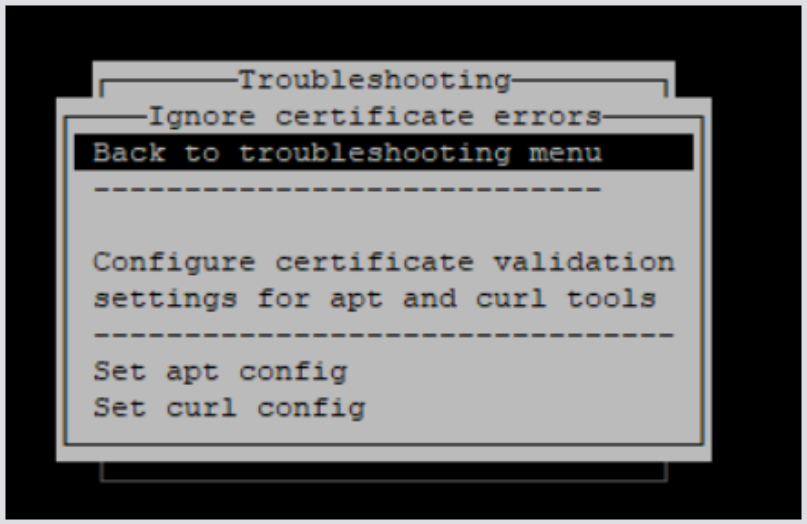
Disabling "Ignore certificate errors" option

1. **Open PDMenu**: Launch the PDMenu from your application menu or by running `pdmenu` from the terminal.
2. **Navigate to the Troubleshooting section**: Use the arrow keys to select `Troubleshooting` and press `Enter`.
3. **Select "Ignore certificate errors"**: Find the option labeled “Ignore certificate errors” and press `Enter`.



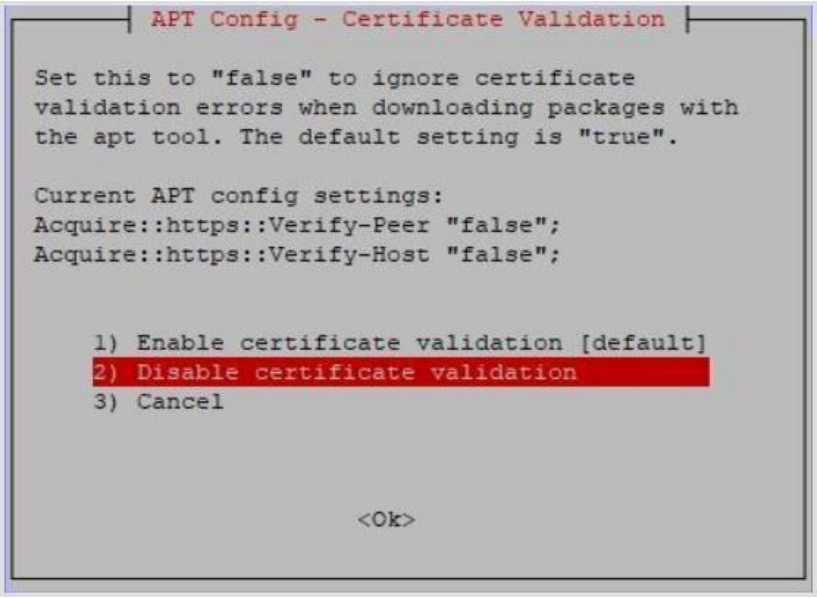
4. **Configure certificate validation settings**.

- You will see the following submenus:
    - `Set apt config`
    - `Set curl config`



Restoring certificate validation for `apt`

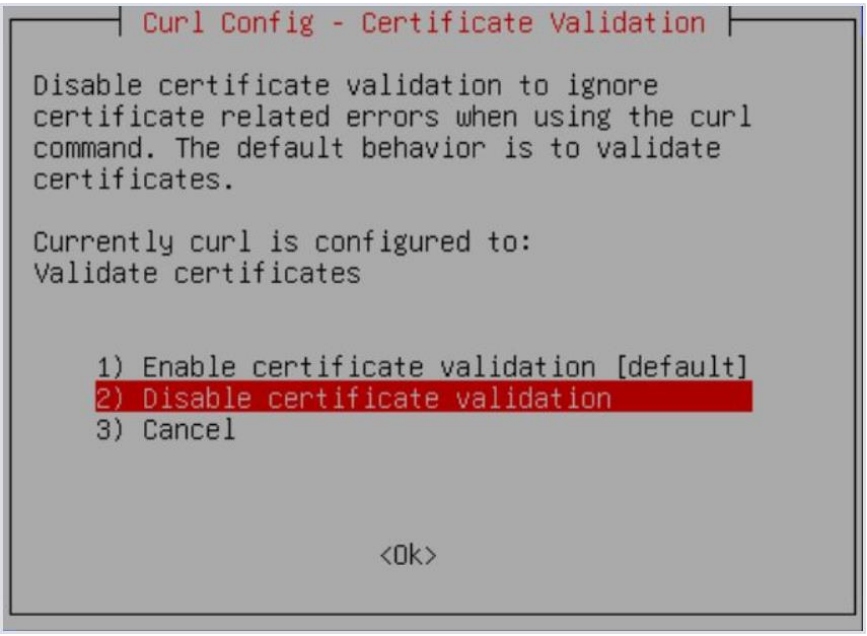
1. **Select "Set apt config"**: Use the arrow keys to highlight `Set apt config` and press `Enter`.
2. **Disable ignoring certificate errors**: Follow the on-screen prompts to remove or comment out the configuration that ignores certificate errors for `apt`.



The certificate validation will be ignored.

Restoring certificate validation for `curl`

1. **Select "Set curl config"**: Use the arrow keys to highlight `Set curl config` and press `Enter`.
2. **Disable ignoring certificate errors**: Follow the on-screen prompts to remove the `--insecure flag` from `curl` commands.



By following these instructions, you can effectively manage SSL certificate issues using the PDMenu interface, ensuring smooth package management. Remember to re-enable certificate validation once the immediate issues are resolved to maintain the security and integrity of your system.

Login Enterprise services not starting

Various configuration errors can cause some of the Login Enterprise microservices to fail to start. The specific number of services not starting depends on the major version of Login

Enterprise, but they have the same underlying cause. Below are some common factors known to cause this issue that are worth considering.

Starting 8 Services (v6) or 10 Services (v5)

SSL Certificate-related issues

Since the Appliance uses SSL communication between the containers, certificate problems can cause the appliance to be unable to determine that a service has started.

1. FQDN of the certificate does not match the first entry in the Allowed URLs or the Hostname of the appliance.
2. A Certificate or Certificates missing (typically Root or Intermediate Certs).
3. Certificates not meeting the minimum requirements.
4. Certificates not including the correct purpose (Server Authentication).
5. FQDN of the certificate does not resolve to an address reachable from the appliance.

The first step would be to:

1. Log in to the Appliance through SSH using the admin account.
2. Go to **Troubleshooting** > **Open bash shell**, and run the command:

```
openssl s_client -connect FQDN:443 -showcerts | more
```

The FQDN you specify should be the name used in your certificate, set as the first entry in the Allowed URLs setting. Read the output to determine if there is a problem with an unreachable host, a name mismatch, or missing root CA certificates.

Alternatively, reverting to self-signed certificates allows the appliance to generate and apply its own SSL certificate. For more information, see Switching back to SSL certificates.

For troubleshooting the 'Services not starting' message, review the contents of the most recent /loginvsi/logs/ApplianceGuard*.log file. Or, use the following command:

```
journalctl -u pi_guard
```

If you want to follow the `pi_guard` log, use one of the following commands:

```
tail -f /loginvsi/logs/ApplianceGuard*.log
```

```
journalctl -u pi_guard -f
```

Based on the error messages that are encountered in the `pi_guard` log, you will need to take the appropriate action to resolve the issue.

1. For missing Root or Intermediate Certs, see Importing a root CA certificate. Please note that the steps will need to be repeated for each Root or Intermediate Cert as only one is allowed at a time.
2. If you see a message related to a name mismatch in the `pi_guard` logs, check whether the FQDN of the Cert and the first entry in Allowed URLs match. For information on allowed URLs, see Managing and adding Virtual Appliance URLs.
3. If your FQDN is unreachable because it is intentionally outside the available network, you can work around this by adding the certificate's FQDN to the localhost lines in /etc/hosts.
4. Or, you can regenerate your certificate to include both the external FQDN and the internal hostname as SANs in the certificate, and specify the internal hostname as the first Allowed URL entry.

For the encryption requirements for the SSL Certificate, see Encryption.

NOTE: RSA keys must be at least 2048 bits. ECC keys of 256 bits or stronger are supported.

Starting 7 Services (v6) or 9 Services (v5)

If you are using an external Microsoft SQL Server database, the system will fail to come up if that database is unavailable. It will get stuck waiting at "9 Services". Verify that the SQL server is operational and that your specific database is being served. If you are unsure whether you are using an internal or external database:

1. Log in to the Appliance through SSH using the admin account.
2. Go to **Troubleshooting** > **Open bash shell**, and run the command:

```
grep COMPOSEFILE /loginvsi/.env
```

The path of the Docker Compose file listed there will tell you whether it's internal or external. If you don't know what your configured database server is, or it has changed, and you need to update the configuration, contact us.

Starting 8 Services (v5)

If you have run out of disk space, you may get stuck in a state waiting for 8 services to start up. You may still be in this state after expanding the disk and the file system. To see whether you are out of disk space:

1. Log in to the Appliance through SSH using the admin account.
2. Go to **Troubleshooting** > **Open bash shell**, and run the command:

```
df -k /
```

If you are out of space, follow the instructions on expanding the Login Enterprise disk space to recover. If you still can't get all the services started, contact us.

In rare instances, you may see a message stating 'Starting 8 Services' that does not progress. This is likely to occur on an Appliance where an external database has been configured. If the database connection was set up in 4.6. x or prior and has not been reapplied since it was set up, the related connection string may be missing some key parameters.

The resolution to this issue is to log in to Login Enterprise and reinitialize the database connection information. Once the connection is reinitialized, you may need to restart the services or reboot the Appliance for the services to come back online.

APPLIANCE_GUARD_URL mismatch issue

Appliance Guard is the service that monitors the Login Enterprise services. If it is misconfigured, your services will be unable to start correctly. To fix the mismatch issue, take the following steps:

1. Access the appliance bash shell via the troubleshooting menu.

2. Check the IP address of the Docker network adapter: Use the following command to check the IP address associated with the Docker network adapter:

```
ifconfig docker_gwbridge
```

This command will display detailed information about the specified network adapter, including its IP address.

3. Update the APPLIANCE_GUARD_URL: Once you have identified the IP address associated with the `docker_gwbridge` interface, you need to update the `APPLIANCE_GUARD_URL` configuration accordingly. This involves manually editing a configuration file. Edit the file `/loginvsi/.env` and find the `APPLIANCE_GUARD_URL entry`.

4. Restart services: Run `loginvsid restart` or select the Restart LoginVSI Services in the Troubleshooting section of the menu. Wait for the services restart to complete.

5. Test connectivity: Test the connectivity of your appliance to ensure that it can communicate properly using the updated APPLIANCE_GUARD_URL configuration.

By following these steps, you should be able to resolve the `APPLIANCE_GUARD_URL` mismatch issue and ensure that your appliance functions correctly within the Docker environment using the `docker_gwbridge` interface.

Configuring SSH Security and Ciphers

- Overview
- Allowed MACs
- Disallowed ciphers
- Customizing ciphers
- Configuration file
- Verifying SSHD configuration changes
- Modifying the configuration
- Additional resources

Overview

This section outlines the security configurations for SSH (Secure Shell) and SSHD (SSH Daemon) to ensure the use of ciphers, in compliance with `seclevel=2`. It details the allowed Message Authentication Codes (MACs) and ciphers and provides instructions for disabling weak ciphers. The minimum TLS Version is TLS 1.2. For the OpenSSH specifications, see OpenSSH: Specifications.

Allowed MACs

The following MACs are permitted for SSH and SSHD:

- umac-128-etm@openssh.com
- hmac-sha2-256-etm@openssh.com
- hmac-sha2-512-etm@openssh.com
- umac-128@openssh.com
- hmac-sha2-256
- hmac-sha2-512

Disallowed ciphers

The following cipher is disallowed:

- chacha20-poly1305@openssh.com (due to the Terrapin vulnerability)

Customizing ciphers

You can remove any ciphers from the `sshd_config` file for any reason.

Configuration file

The following configuration is for the SSH server (SSHD).

The relevant section in `/etc/ssh/sshd_config` will appear something like this:

Message Authentication Codes

```
macs umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-sha2-512-
etm@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512
```

Ciphers

```
ciphers -chacha20-poly1305@openssh.com
```

Specify allowed host key algorithms

```
HostKeyAlgorithms ssh-rsa,ssh-ed25519
```

In this example, only RSA and ED25519 host key algorithms are allowed. You can customize the list based on your security requirements and preferences.

Make sure to restart the SSHD service after making changes to the configuration file for the changes to take effect:

```
sudo systemctl restart sshd
```

Verifying SSHD configuration changes

To verify whether the modification to the SSHD configuration file successfully disabled the ssh-rsa algorithm, follow these steps:

1. **Check for existing configuration**: Open the SSHD configuration file (`/etc/ssh/sshd_config`) and ensure that the HostKeyAlgorithms parameter includes only secure algorithms such as `ssh-ed25519`. If this parameter is not present, or if `ssh-rsa` is still included, proceed to the next steps to modify the configuration.

2. **Modify the SSHD configuration**: If necessary, add or modify the HostKeyAlgorithms parameter in the SSHD configuration file to include only secure algorithms such as `ssh-ed25519`. You can add this parameter to the bottom of the file on a new line if it's not already present.

3. **Restart SSHD service**: After making the modification, restart the SSHD service to apply the changes:

```
sudo systemctl restart sshd
```

4. **Test SSH connection**: Use an external SSH client to attempt an SSH connection to the server in question. You can specify the `-o` option to enforce the use of a specific algorithm:

```
ssh -o HostKeyAlgorithms=ssh-rsa user@hostname
```

5. **Observe the connection outcome**: If the SSH connection is denied or fails, it indicates that the `ssh-rsa` algorithm has been successfully disabled.

6. **Verify SSHD configuration**: Additionally, you can verify the SSHD configuration settings using the sshd `-T` command:

```
sshd -T | grep -i hostkeyalgorithms
```

Ensure that the output reflects the updated HostKeyAlgorithms setting without the `ssh-rsa` option.

Modifying the configuration

1. Open the `sshd_config` file in a text editor.
2. Locate the ciphers or MACs directive.
3. Remove the cipher from the list or add a new one.
4. Save the file.
5. Restart the SSHD service or reboot the appliance.

Additional resources

To learn how to disable weak SSH ciphers for Linux VMs, see [How to disable weak SSH ciphers for Linux VMs].

Configuring Authentication

- Overview
- Configuring Active Directory
- The multi-domain controller in LDAP
  - Configuring multi-domain controllers
  - Using multi-domain controllers
- Security requirements
- Multi-domain environment configuration
- Troubleshooting
  - Login problems with LDAP/Active Directory

Overview

This section focuses on configuring authentication for users logging into the Login Enterprise web interface. This is distinct from Test Users, which are used specifically for running Tests within the Virtual Appliance.

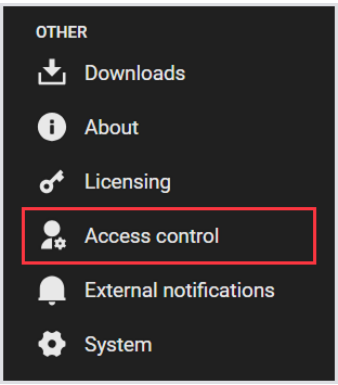
To enhance user access to the Login Enterprise Virtual Appliance, administrators can integrate it with Active Directory (AD). This integration allows for more granular control over permissions within the organization, moving away from the reliance on local administrator credentials.

Administrators can also specify an Active Directory LDAP Authentication timeout. This feature addresses scenarios in larger, more complex domains where the default timeout of 55 seconds may be insufficient for successful authentication.

Login Enterprise supports Role-Based Access Control (RBAC), enabling administrators to define custom roles with specific permissions. RBAC provides flexibility in managing user access, as administrators can create roles tailored to different user needs. For more information on how roles and permissions work, see the Role-Based Access Control (RBAC).

Configuring Active Directory

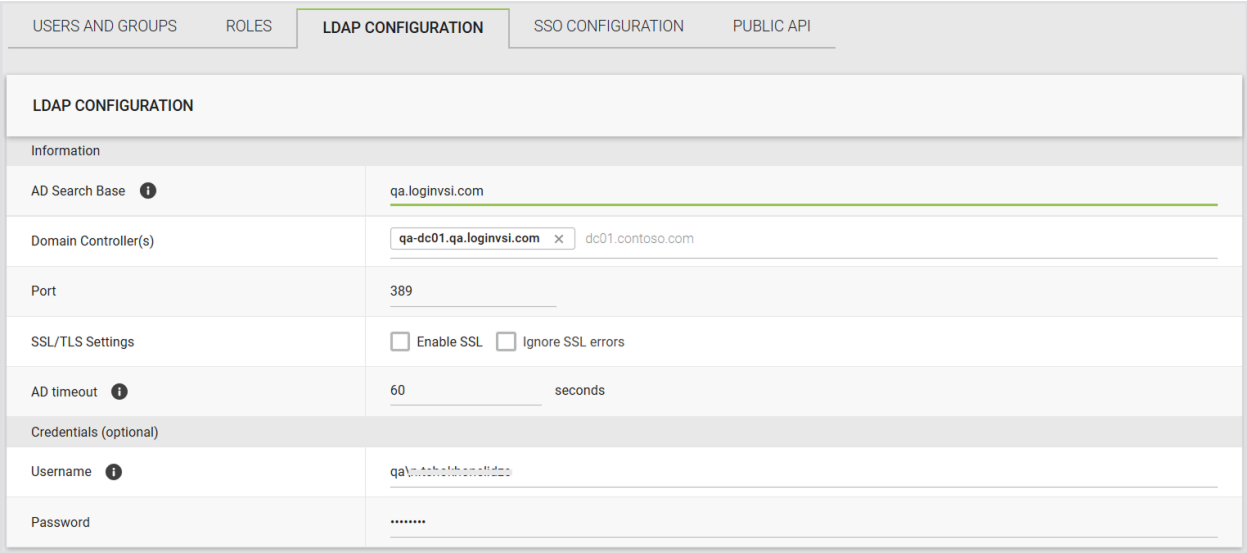
1. In the **Login Enterprise sidebar menu**, navigate to **Other > Access Control**.



2. In **Access Control**, select **LDAP** from the tab bar menu and provide the following:

Field	Value
AD Search Base	The fully qualified domain name (FQDN) of the domain used for authentication.
Domain controller	The domain controller responsible for processing the requests. You can configure up to 5 Domain Controllers for LDAP authentication**.**
Port	The port utilized for LDAP.
SSL/TLS Settings	• Select **Enable SSL** for Login Enterprise to use SSL/TLS encryption to establish a secure connection with the AD server. This is the recommended choice to enhance security and protect sensitive information during authentication. • Select **Ignore SSL** errors to instruct Login Enterprise to bypass SSL certificate validation errors that may occur during the connection process. While this option can be useful in certain situations, e.g. dealing with self-signed certificates, it also introduces potential security risks by allowing connections to proceed without proper certificate validation.
AD Timeout	The Active Directory controller's response time. You can extend the default 60-second AD timeout up to 180 seconds.
Credentials (optional)	When SSO is **not** enabled, these credentials are optional. However, when SSO **is** enabled, Login Enterprise must authenticate to LDAP to look up user group membership and permissions after login. For this reason, a valid LDAP username and password must be saved when using SSO.

3. Click **Save Changes** to apply your AD authentication integration.



If you configured LDAP without SSO, credentials behave differently depending on whether you filled them in:

1. **Credentials entered on the page**: Login Enterprise uses these credentials to perform a connectivity check when saving.
2. **Credentials left empty**: You'll be prompted with a pop-up to enter credentials. These are used only for a one-time connectivity check and are **not stored** in Login Enterprise.

After saving, you can log out and log back in using your domain account (if SSO is enabled). For more information on SSO, see Single Sign-On (SSO).

The multi-domain controller in LDAP

The support for multiple domain controllers allows the configuration of up to five domain controllers for LDAP authentication. This ensures that if the primary domain controller becomes unavailable, the system will automatically attempt to connect with the next available controller in the list. The feature enhances the system's availability and reliability by enabling a fail-over mechanism across multiple domain controllers (DC).

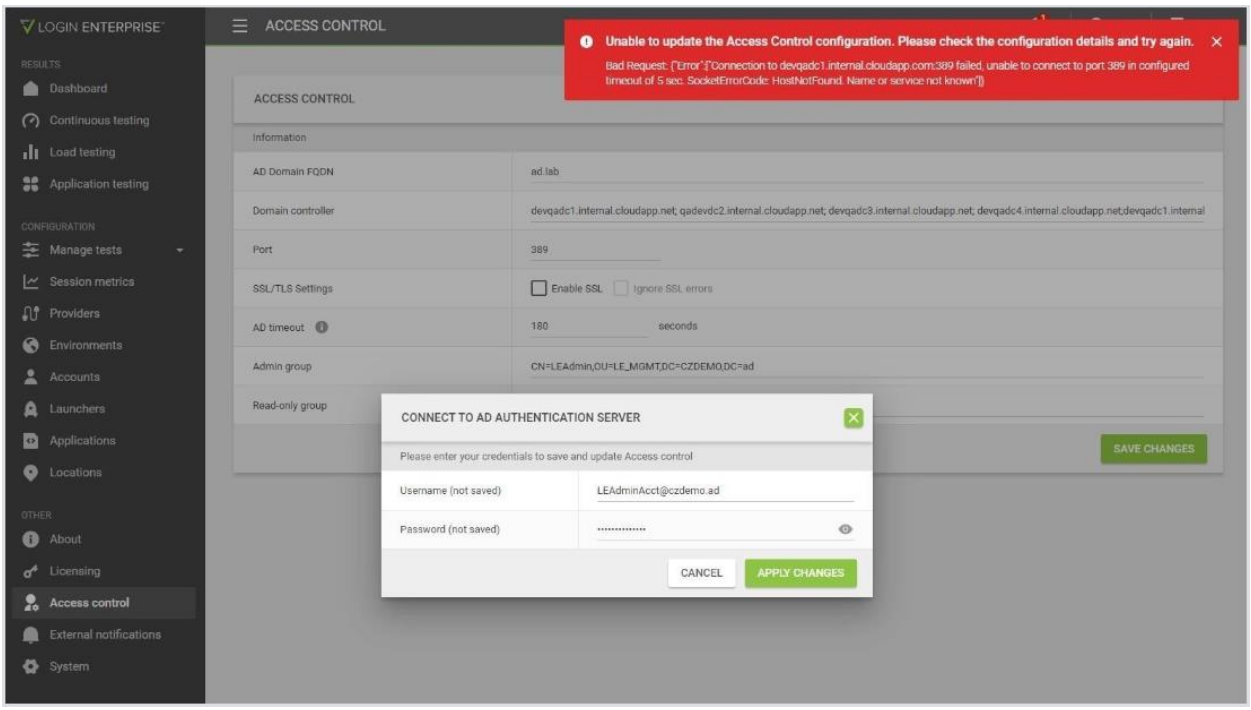
To configure and utilize it, as a Login Enterprise administrator, you need to have administrative access to Login Enterprise.

Configuring multi-domain controllers

1. **Access the Access Control panel**: Navigate to the LDAP settings in your system's administration console.
2. **Update Domain Controllers list**: Input the domain controllers in the order of priority in the field Domain controller as a semicolon-separated list. For example:

```
devqadc1.internal.cloudapp.net; qadevdc2.internal.cloudapp.net;
devqadc3.internal.cloudapp.net; devqadc4.internal.cloudapp.net;
devqadc5.internal.cloudapp.net
```

NOTE: All DCs must be available, and the configured port (389) must be open. Otherwise, the application throws an error message with incorrect DCs:



3. **Save and apply changes**: Confirm the settings and apply them to enable the feature.
4. **Log off and log in again**.

Using multi-domain controllers

Once configured, the system will handle authentication requests across the specified domain controllers without further user intervention.

1. In case one of the DCs is unavailable (stopped) the corresponding info is logged to the IdentitySerever container log:

```
[19:34:28 INF] Attempting to connect to LDAP server
[19:34:28 INF] Connecting to devqadc5.internal.cloudapp.net:389 (ssl: False)
[19:34:28 ERR] Connection failed to
ldap://devqadc5.internal.cloudapp.net:389: Connect Error (Ldap result #91
Connect Error)
LdapException: Unable to connect to server devqadc5.internal.cloudapp.net:389
(91) Connect Error
System.Net.Internals.SocketExceptionFactory+ExtendedSocketException
(00000005, 0xFFFDFFFF): Name or service not known
[19:34:28 INF] Connecting to devqadc1.internal.cloudapp.net:389 (ssl: False)
[19:34:28 INF] Connected to LDAP server
```

```
[19:34:28 INF] Attempting to bind to LDAP server with username:
LEAdminAcct@czdemo.ad
[19:34:28 INF] Binding user credentials
[19:34:28 INF] Bind succeeded, authentication DN: LEAdminAcct@czdemo.ad
```

2. The same if the port is closed:

```
[09:50:45 INF] Connecting to devqadc5.internal.cloudapp.net:389 (ssl: False)
[09:50:45 INF] Checking if port 389 is open
[09:50:45 INF] Connecting to 10.0.0.9:389
[09:50:50 ERR] Connection failed to ldap://devqadc5.internal.cloudapp.net:389
due to configured timeout of 5 sec for port open check.
[09:50:50 INF] Connecting to devqadc1.internal.cloudapp.net:389 (ssl: False)
[09:50:50 INF] Checking if port 389 is open
[09:50:50 INF] Connecting to 10.0.0.5:389
[09:50:50 INF] Port 389 is open
[09:50:50 INF] Connected to LDAP server
[09:50:50 INF] Attempting to bind to LDAP server with username:
LEAdminAcct@czdemo.ad
[09:50:50 INF] Binding user credentials
[09:50:50 INF] Bind succeeded, authentication DN: LEAdminAcct@czdemo.ad
[09:50:50 INF] Successfully bound to LDAP server
```

Security requirements

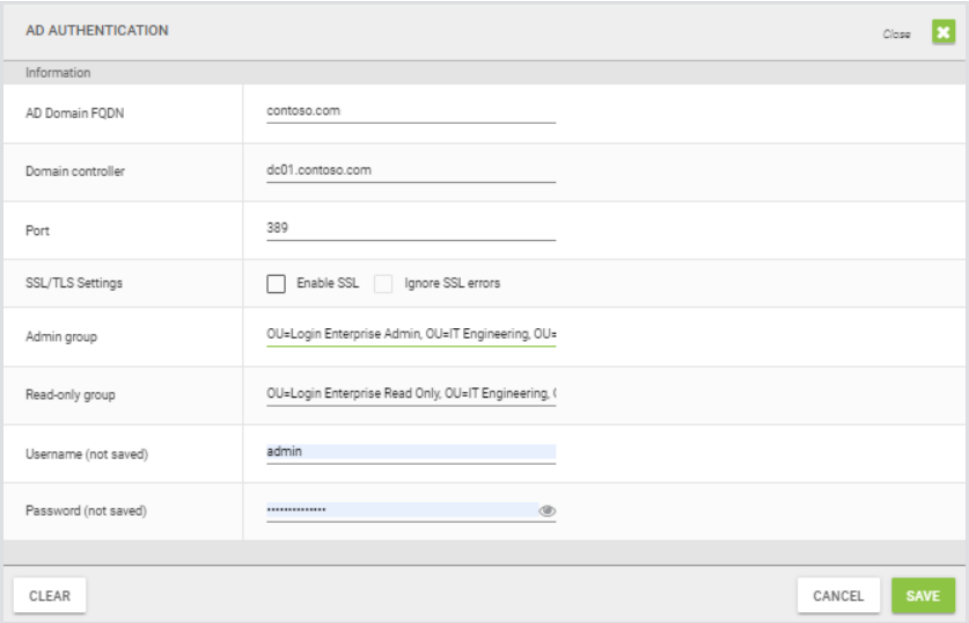
The minimum requirements for secure AD Authentication / LDAPS (port 636) are:

- Protocol TLS 1.2
- 2048-bit certificate

Multi-domain environment configuration

Login Enterprise supports multi-domain configuration. This setup requires a different configuration of the above AD Authentication parameters.

Because the Group name is not guaranteed to be unique across multiple domains, the Group's distinguished name is required. Thus, the configuration will look more like the example below:



Login VSI has tested and supports using a single parent domain with two direct subdomains. Resources from any domain can be used in this configuration. Users can reside in a subdomain while the domain controller and groups are used from the top-level domain.

NOTE: When you have multiple domains and need to specify groups using their Distinguished Names, you need to log in with UPN name format: either just the username or username@domainname.dom. You can't use domainname\username format. It will report an authentication failure.

Troubleshooting

Login problems with LDAP/Active Directory

Issue overview

When you attempt to log in to the Login Enterprise Appliance using an Active Directory (LDAP) account, the web page initiates a request to the appliance, which in turn executes an LDAP query against the configured directory server. If an unexpected issue occurs within the appliance, it may respond with a 504 Server Error to the web page. It's important to note that this 504 error originates from the appliance's backend service rather than directly from the directory server.

Troubleshooting 504 server error during the AD login process

NOTE: This is a workaround/temporary solution. If you have any questions or concerns, contact us.

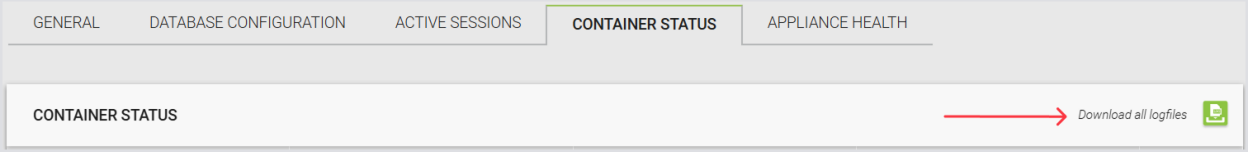
If you get errors logging in to a domain, verify that Login Enterprise can talk to the domain at all. For this:

1. Log in to Login Enterprise with the local admin account.
2. In the **Sidebar menu**, scroll down and find **Other > Access Control**.
3. In **Access Control**, verify that you can save your existing AD configuration by clicking **Save Changes**. (You might need to make a trivial change to test saving the settings).
4. Pay attention to whether the **Save Changes** action completes within the "AD timeout" setting configured on that page.
5. If your AD test is taking too long, your authentication attempt will take the same amount of time. Our AD test is nothing more than test-authenticating with the provided user account.
6. Try changing to a different domain controller, in case the one you have currently selected is impaired.

How to access log files

LDAP errors are reported in the Login Enterprise container logs, in the IdentityServer log. You can access the log files in 2 ways:

- From the **Login Enterprise sidebar menu > System > Container Status > Download all log files**.



- Using SSH to log in to the Appliance as the local `admin` user and looking at files in `/loginvsi/logs`.

Look for files named `IdentityServer{date-time}.log`. For example: `IdentityServer20240215.log`

Single Sign-On (SSO)

- Overview
- How SSO Works
- Choosing an SSO Method

Overview

Single sign-on (SSO) allows users to sign in once and access Login Enterprise without repeated authentication prompts. Authentication is delegated to an external identity provider (IdP), which verifies user identities and manages sign-in.

SSO streamlines the sign-in experience and helps organizations enforce security policies such as multi-factor authentication (MFA), reduce password management overhead, and improve overall security.

SSO can be implemented using standard protocols such as SAML or OpenID Connect (OIDC), depending on your identity provider and application requirements.

How SSO Works

In Login Enterprise, authentication and authorization are handled separately:

- **Authentication** is performed by an external identity provider (IdP) using single sign-on (SSO).
    - This can be configured using SAML or OpenID Connect (OIDC).
- **Authorization** is handled by LDAP.

After a user signs in through the identity provider, Login Enterprise queries LDAP to determine the user's group membership (for example, whether the user belongs to the required Active Directory groups) and assigns access accordingly.

NOTE: Because authorization depends on LDAP, it must be configured even when SSO is enabled. For details, see Configuring Authentication.

Choosing an SSO Method

1. **SAML** is a widely adopted standard supported by many identity providers. It is often used in enterprise environments and offers broad compatibility across different systems.
2. **OpenID Connect (OIDC)** is a modern authentication protocol built on OAuth 2.0. It supports modern authentication flows and integration with newer applications.

When you choose a method, consider the capabilities of your identity provider and your organization's existing setup:

- Choose **SAML**: if you require compatibility with a wide range of identity providers or are working with existing enterprise SSO configurations.
- Choose **OpenID Connect (OIDC)**: if your identity provider supports OIDC. When you configure OIDC, you can select **Microsoft Entra ID** or **AD FS** as the identity provider.

For configuration steps, see Configuring SAML and Configuring OpenID Connect.

Configuring SAML

- Overview
- How SAML Works
- Prerequisites
- Configuring SAML in Login Enterprise
- Testing the Configuration

Overview

While OpenID Connect is a modern and recommended protocol, many organizations continue to rely on SAML, which remains widely adopted and supported by most identity providers. To improve compatibility with existing enterprise environments, Login Enterprise supports SAML-based SSO in addition to OpenID Connect.

SAML provides a standardized, identity provider–agnostic approach. If an identity provider supports SAML, it can typically be integrated without provider-specific configuration. In contrast, OpenID Connect integrations may involve provider-specific differences at the implementation level, although the configuration experience for users is often similar across providers.

This section describes how to configure SAML-based SSO in Login Enterprise. The examples use Okta as the identity provider. Steps may vary depending on your provider, such as Microsoft Entra ID or others.

SAML SSO in Login Enterprise supports both identity provider (IdP)-initiated and service provider (SP)-initiated login flows.

How SAML Works

After SAML is configured:

- Users are redirected to the identity provider for authentication.
- If the user already has an active session with the identity provider, they are signed in automatically.
- Otherwise, the user is prompted to enter their credentials.

User access in Login Enterprise is determined by the roles assigned to the user. For details, see Role-Based Access Control (RBAC).

Prerequisites

Before configuring SAML-based SSO, ensure that the following requirements are met:

- An LDAP configuration is in place. For more information, see Configuring Authentication.
- An identity provider (such as Okta) is configured.
- A SAML application is created in your identity provider.
- Users and/or groups are assigned to the application in your identity provider.
- You have obtained the following values from your identity provider:
    - Identity Provider metadata URL
    - Identity Provider identifier (issuer)
    - Signing certificate or key

For instructions on creating and configuring a SAML application in Okta, see the Okta documentation.

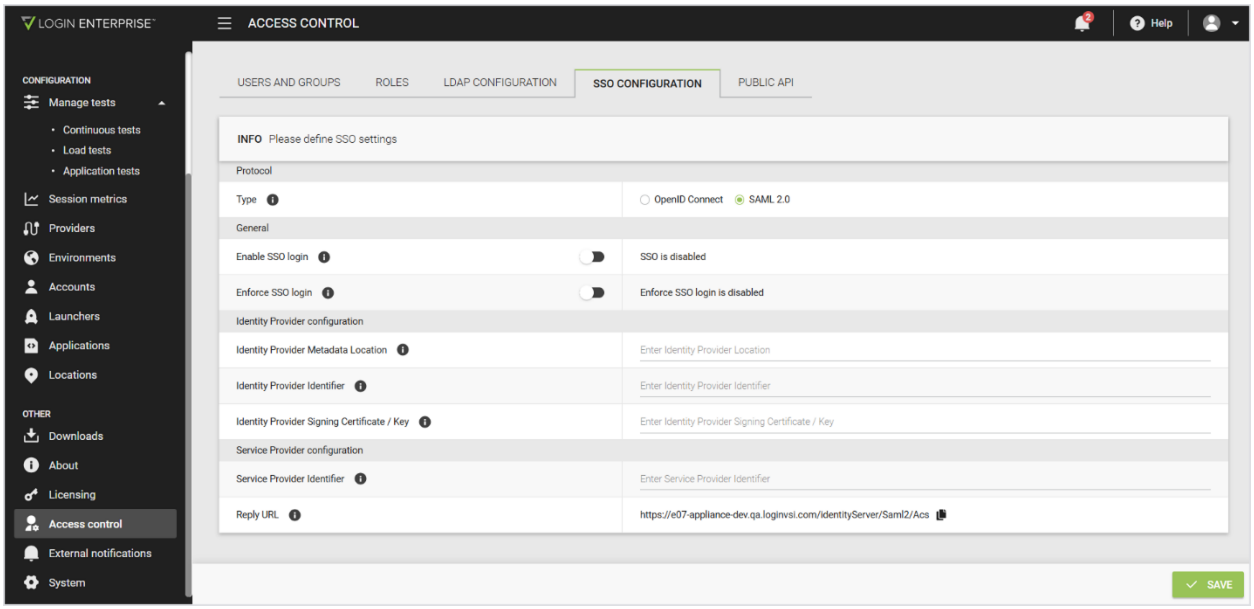
Configuring SAML in Login Enterprise

1. Sign in to Login Enterprise using an account with administrative privileges.
    1. Using a local admin account is recommended to prevent potential lockout if SSO is misconfigured.
2. In the **Login Enterprise sidebar menu**, navigate to **Access Control** > **SSO configuration**.
3. Configure the following:

NOTE: Before configuring the Identity Provider fields, obtain the required values from your identity provider. These values are typically available in the provider's SAML application settings or metadata. Because the exact steps and field names vary by provider, refer to your identity provider's documentation.

Protocol	
Type	The protocol used for SSO. For SAML, select the **SAML 2.0** radio button.
General	
Enable SSO login	When disabled, the SSO login option isn't available on the login page.
Enforce SSO login	When enabled, users can sign in only with SSO or a local admin account.
Identity Provider configuration	
Identity Provider Metadata Location	The URL to the SAML metadata XML provided by your identity provider.
Identity Provider Identifier	The unique identifier of your identity provider in the SAML configuration.

Identity Provider Signing Certificate / Key	The signing certificate or key used to validate SAML responses from the identity provider.
Service Provider configuration	
Service Provider Identifier	The Assertion Consumer Service URL that the identity provider redirects users back to after sign-in.
Reply URL	The Assertion Consumer Service URL that the identity provider redirects users back to after sign-in.

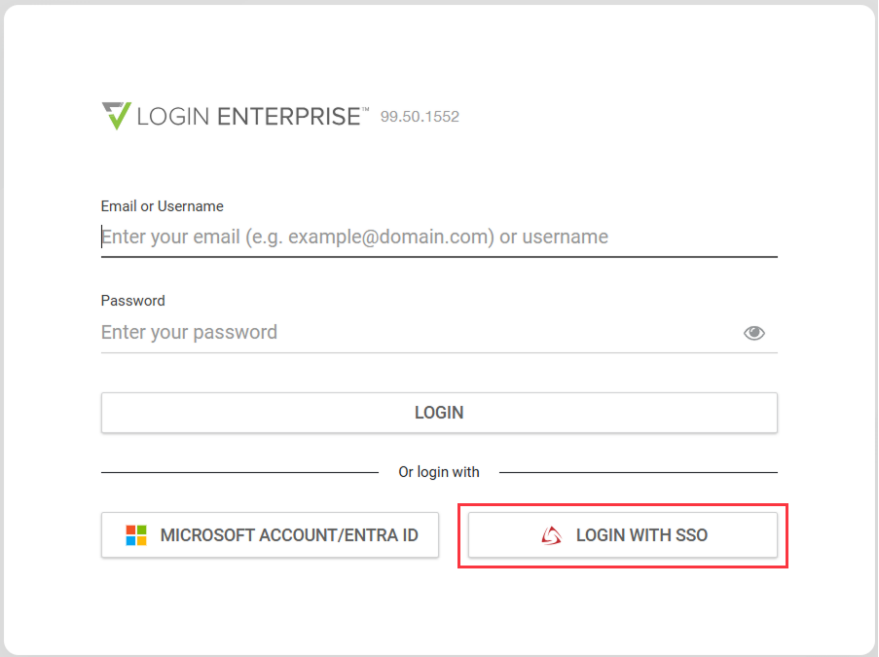


4. Save the configuration.

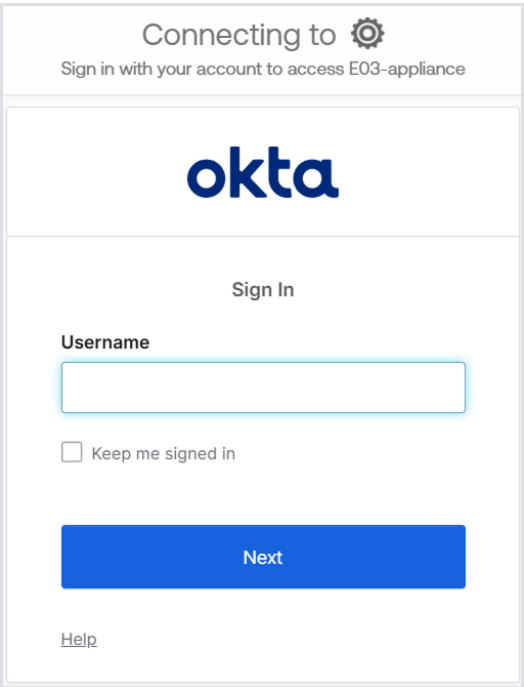
Testing the Configuration

After you configure SAML-based SSO, verify that authentication works as expected.

1. Sign out of Login Enterprise.
2. Select **Log in with SSO**.



3. You are redirected to your identity provider's sign-in page (for example, Okta).
    1. If you already have an active session with your identity provider, you are signed in automatically.
    2. Otherwise, you are prompted to enter your credentials.



4. Optionally, confirm that you're assigned the correct roles based on LDAP group membership.

Configuring OpenID Connect (OIDC)

- Overview
- Prerequisites
- Configuring OIDC in the Identity Provider
  - Microsoft Entra ID
  - AD FS
- Configuring OIDC in Login Enterprise
- Testing the Configuration
- Break-Glass Access

Overview

OpenID Connect (OIDC) is an authentication protocol used for Single Sign-On (SSO). It allows users to authenticate through an external identity provider (IdP).

Login Enterprise relies on the identity provider to verify the user's identity and uses LDAP for authorization. LDAP group membership is used to assign roles in Login Enterprise.

Login Enterprise supports the following OIDC providers:

- Microsoft Entra ID
- AD FS

NOTE: Only one SSO provider can be active at a time.

Prerequisites

Before configuring OIDC, make sure that the following requirements are met:

1. An LDAP configuration is in place. For more information, see Configuring Authentication.
2. LDAP credentials are available.
   1. These credentials are optional when saving the LDAP configuration without SSO.
   2. These credentials are required when SSO is enabled, because Login Enterprise uses them to query user and group information for role assignment.
3. A supported OIDC-compatible identity provider is available, such as Microsoft Entra ID or AD FS.
4. An application is registered in the identity provider.

For AD FS, an administrator must have permission to create and configure an OIDC application group or client in AD FS.

Configuring OIDC in the Identity Provider

To configure OpenID Connect (OIDC), you must register an application in your identity provider. This application is used to authenticate users who sign in to Login Enterprise.

Microsoft Entra ID

For guidance on configuring single sign-on in Microsoft Entra ID, see the Microsoft documentation.

Configuration options such as multi-factor authentication (MFA), conditional access, and user assignment are managed in Microsoft Entra and are outside the scope of this documentation.

AD FS

Before configuring AD FS in Login Enterprise, an administrator must configure the OIDC application in AD FS. For guidance on configuring an OpenID Connect application in AD FS, see the Microsoft documentation.

The AD FS administrator must create an application group or client and generate a client secret. After the Redirect URL is available in Login Enterprise, add it as a redirect URI in AD FS.

Configuring OIDC in Login Enterprise

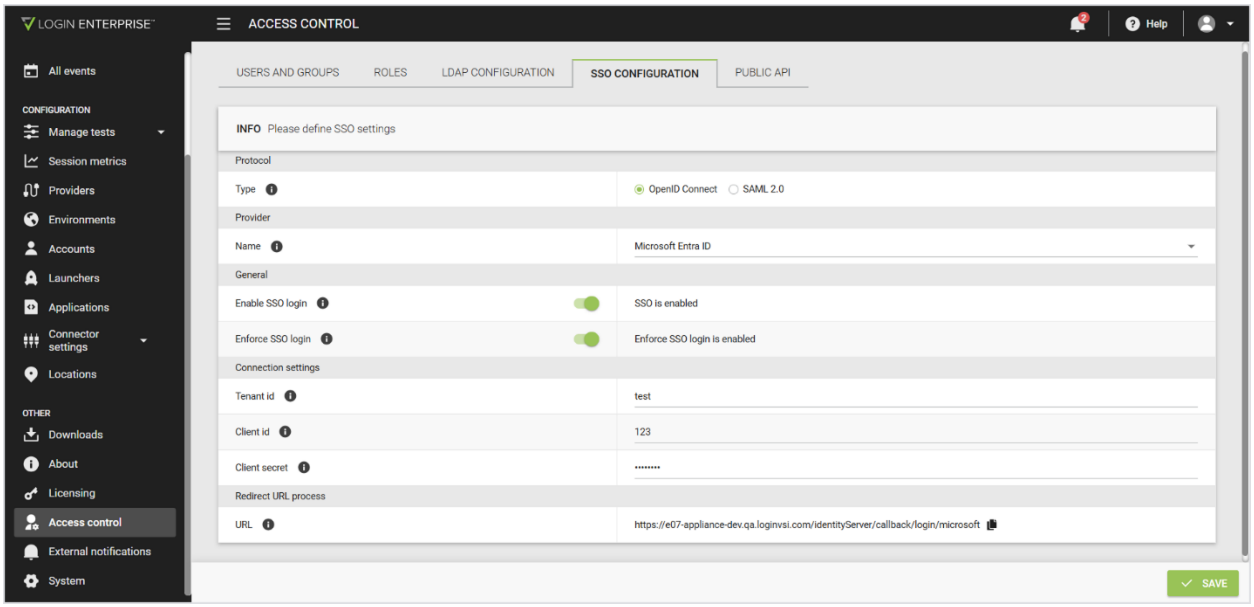
1. Sign in to Login Enterprise using the local admin account.
2. In the **Login Enterprise sidebar menu**, navigate to **Access Control** > **SSO configuration**.
3. Configure the following:

Protocol	
Type	The protocol used for SSO. Select **OpenID Connect**.
Provider	
Name	The identity provider used for SSO. Select one of the following options: • Microsoft Entra ID • AD FS
General	
Enable SSO login	When disabled, the SSO login option isn't available on the login page.

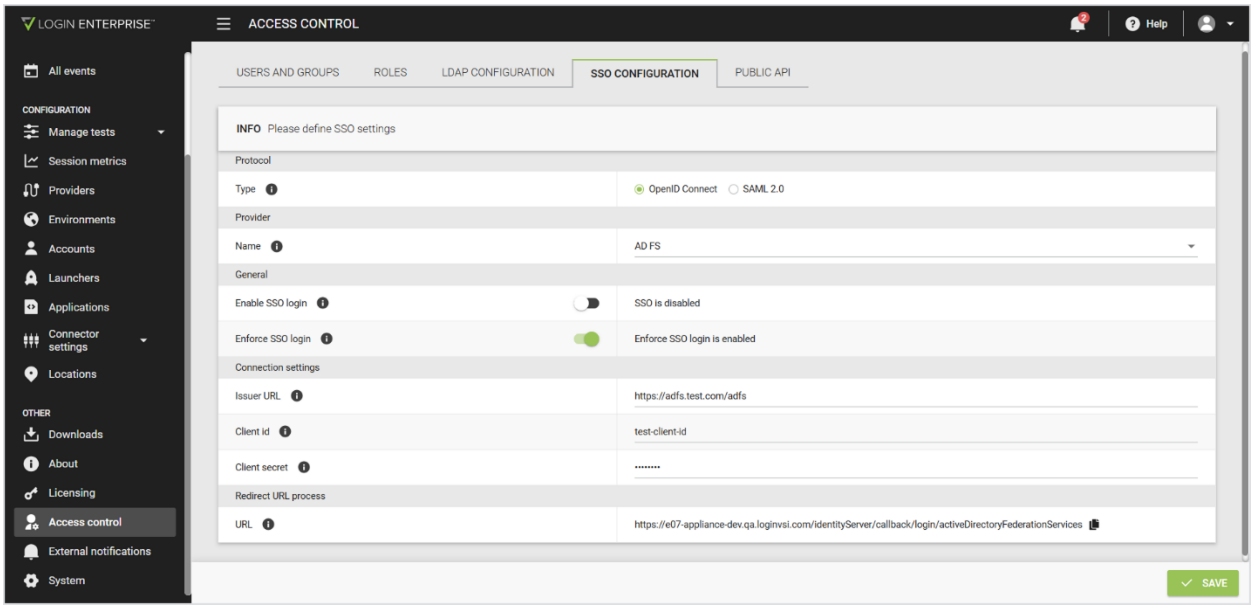
Enforce SSO login	When enabled, users can sign in only with SSO or the admin account.
Connection settings (Microsoft Entra)	
Tenant ID	The Microsoft Entra tenant (directory) ID to authenticate users. This value is a GUID and determines which directory is used for sign-in.
Client ID	The application (client) ID of the Microsoft Entra app for SSO.
Client Secret	The client secret for the Microsoft Entra app registration, used to authenticate the application.
Connection settings (AD FS)	
Issuer URL (required)	The AD FS server base URL used as the OpenID Connect issuer, format `https://<adfs-fqdn>/adfs`
Client ID (required)	The client identifier registered for Login Enterprise in the AD FS application group.
Client Secret (required, masked)	The shared secret generated for the Login Enterprise server application in AD FS.
Redirect URL process	
Redirect URL	The Login Enterprise-generated URL that Microsoft Entra or AD FS uses to redirect users after successful sign-in. **NOTE**: The URL cannot be edited in Login Enterprise, but can be modified when pasting it into the identity provider's configuration. You can add multiple URLs in Microsoft for multiple domains, if needed.

4. Save the configuration.

Here's an example of the MS Entra ID configuration:



An example of the AD FS configuration looks as follows:



Testing the Configuration

To verify that OIDC authentication is configured correctly:

1. Sign out of Login Enterprise.
2. Select the appropriate identity provider sign-in option. The available options depend on the identity providers that are configured and enabled:
    1. **Sign in with your organization** for AD FS.
    2. **Microsoft account / Entra ID** for Microsoft Entra ID.

3. Sign in using a user account from the selected identity provider.
4. Optionally, confirm that the user is assigned the correct roles based on LDAP group membership.

NOTE: If the redirect URL is misconfigured, sign-in may fail. Ensure that the redirect URL configured in MS Entra ID or AD FS matches the Redirect URL process field in Login Enterprise.

Break-Glass Access

The built-in admin login remains available even when OIDC authentication is enabled and **Enforce SSO login** is turned on.

This account provides break-glass access if OIDC authentication is unavailable or misconfigured.

Role-Based Access Control (RBAC)

- Overview
- Migration of Existing LDAP Groups
- Why Use RBAC?
- Key Functionalities
- General Rules
- Built-In Admin
- Configuring RBAC
- Assigning or Removing Test Access from a Role
  - Assigning Test Access
  - Removing Test Access
- Managing Access to Applications and Application Groups
  - Assigning Roles to Applications and Groups
  - Managing Application Access from the Roles UI
- Managing Access to Test Runs
  - Granting Access to a Test Run
  - Granting Access to Multiple Test Runs
  - Rules and Exceptions
- Managing Access to Accounts and Account Groups
  - View and Filter Access
  - Assigning Accounts and Account Groups in Bulk
- Managing Access to Launchers and Launcher Groups
  - Assigning Roles to Launchers
    - From the Launchers overview page
    - From the Roles page
  - Managing Access to Launcher Groups
    - Assigning Roles to Launcher Groups
- Managing Roles and Permissions
- Understanding Permission Dependencies
- Endpoints for Access Management

Overview

Role-Based Access Control (RBAC) provides fine-grained access management for Login Enterprise, enabling administrators to define and manage custom roles with specific permissions. This functionality allows users to have isolated access to Test configurations, ensuring they can interact only with their designated Tests.

Before RBAC, Login Enterprise supported a single permission level, limiting its usability in environments with multiple users or teams with diverse responsibilities. All users who logged into the system shared the same permissions, creating security and operational challenges. RBAC addresses this limitation by allowing custom roles with specific permissions, enabling segregated access to Test configurations, and enhancing security and usability for larger organizations.

IMPORTANT: The RBAC functionality is currently limited to LDAP (Active Directory) for authentication and requires LDAP users and groups for access management. Without LDAP integration, RBAC cannot function.

Migration of Existing LDAP Groups

With RBAC, a migration process automatically creates roles and user groups corresponding to two LDAP groups (Admin and Read-only), provided you have previously set them up.

Migrated Roles:

- **Admin**: All permissions are enabled.
- **Read-only**: Only read permissions are enabled, allowing access to view Test Results, Applications, Session Metrics, and Test Configurations.

LDAP Group Mapping:

- **Admin**: Display name = Admin, Group name = [configured group name]
- **Read-only**: Display name = Read-only, Group name = [configured group name]

NOTE: This migration ensures that users assigned to the Admin or Read-Only LDAP groups can still log in after the upgrade. However, note that these migrated roles are not automatically assigned to Tests. As a result, users may not be able to see any Test data unless the appropriate roles are assigned to the Tests.

To resolve this, Login Enterprise provides API endpoints that allow administrators to bulk update Tests and Test Runs to assign or replace the roles. For details, see Endpoints for Access Management.

Why Use RBAC?

Benefit	Explanation
Granular access control	Enables precise control over Test configurations by defining custom roles and permissions.
Increased customer satisfaction	Provides a flexible and secure access management solution that improves user experience.
Enhanced security	Allows administrators to assign specific permissions to roles, such as Read, Execute, and Read-Write.
LDAP-based authentication support	Integrates with LDAP to support organizational authentication needs.
Improved stability and scalability	Ensures the system remains stable and scalable while implementing role-based access control.

Key Functionalities

- **Custom Roles**: Users with Access Control Read+Write permission can create custom roles with specific permissions.
- **Permissions Management**: Assign permissions to roles at the Test configuration level.
- **Permission Types**: Support for read, execute, and read-write permissions with an additive model (e.g., execute on a Test).
- **LDAP Integration**: Add LDAP users and groups to roles for authentication purposes.
- **Access Control UI**: A new 'Access Control' UI allows you to manage roles and permissions.
- **Fallback**: A super user retains full control over the system.

General Rules

- **LDAP Requirement**: RBAC can only be used in combination with LDAP.
    - **LDAP Group**: When using the LDAP group type, login credentials are verified against LDAP. The system checks if the user exists in the specified LDAP group as configured in Login Enterprise.
    - **LDAP User**: When using the LDAP user type, login credentials are verified against LDAP. If the user exists, access is granted. **NOTE:** Only the User Principal Name (UPN) format is supported for this user type; other variants, such as SAM account names, are not supported.
- **Appliance Admin user**: The original superuser account created during Appliance setup. It exists outside of LDAP and the Access Control UI. It has unrestricted access to all Tests and system areas by default.
- **Built-In Admin role**: A system-defined role introduced in v6.1. It always has full access to all Tests and Test Runs, even though it does not appear in the Access Control tab. The role cannot be deleted, renamed, or modified, and only user or group membership can be changed. See Built-In Admin for more details.
- **Role Assignments to Tests**: Roles can be assigned to Tests, and the permissions of the assigned role define what the user can do with the Test. For example, if role 1 has access to Test 1 with "read-only" results permission, the user can view Test results but cannot modify the Test configuration.
- **Test Results Access**: Test results are tied to the roles assigned to the Test. Users can view all results for a Test they have access to and can assign access to individual Test Runs.
- **Role Assignments to Applications**: Roles can also be assigned to individual Applications and Application Groups. Users can only access and configure Applications and Application Groups that are explicitly assigned to their role. If a user has access to the Test results, they can see all Applications referenced in those results, including those they don't have access to.
- **Accounts and Account Group Access**: Roles can also be assigned to individual Accounts and Account Groups. Users can only access and configure Accounts and Account Groups that are explicitly assigned to their role. However, when a user starts a

Test, the system automatically uses all assigned Account Groups and their associated Accounts, including those the user does not have direct access to.

- **Combined Role Permissions**: When multiple roles are assigned to a user, the user will inherit the combined permissions of all assigned roles.
- **UI Permissions**: The UI will only display menu items and options for which the user has permission.
- **Permissions for Creating/Updating Tests**: To create or update Tests, a user must have read permissions for both Launchers and Accounts. Also, to add Workloads or Session Metrics, the user must have read permissions for Applications and user Session Metrics.
- **Deleting Users**: When a user is removed, they are only removed from Login Enterprise, not from the LDAP.
- **Deleting Roles**: Deleting a role will also remove the role from all the Tests it's assigned to.

Built-In Admin

The Built-In Admin role is a special, non-editable role introduced in version 6.1 to ensure consistent administrative access across Login Enterprise. This role is distinct from both custom roles and the original built-in admin *user* account (now referred to as the **Appliance Admin User**).

NOTE: This section describes the Built-In Admin role, not the Appliance Admin user. The Appliance Admin user is the internal superuser account created during the appliance setup and is not managed via the Access Control UI.

Role Characteristics

- Automatically granted **full permissions** to all Tests and Test Runs.
- **Cannot be deleted**, renamed, or have its permissions modified.
- The **only allowed action** is adding or removing users or LDAP groups assigned to the role.
- **Does not appear** in the Access Control tab of individual Tests, as it is not technically assigned to them.
- **Cannot be manually removed** from any Test or Test Run.

Role Behavior

- **On Test creation**, the Built-In Admin role is automatically granted access, even though it is not visible in the UI.
- **On upgrade to version 6.2**, the Built-In Admin role is automatically created and granted implicit access to all existing Tests and Test Runs.
    - The Built-in Admin role will also automatically get access to all Applications and Application Groups.
    - Existing roles will automatically get all Applications and Application Groups assigned to them.

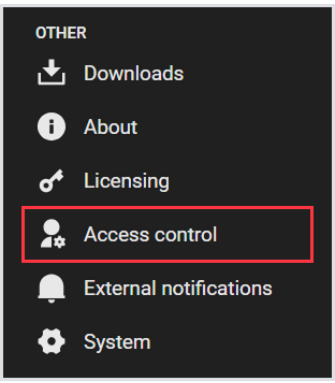
NOTE: This behavior is enforced by the system and does not rely on Test-level role assignments. The access logic recognizes the Built-In Admin role at runtime.

Migration and Compatibility

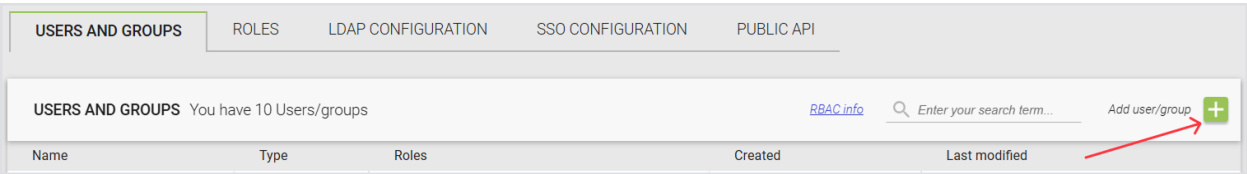
- In version **6.0**, LDAP-enabled appliances automatically generated an `admin` role. This role was editable, deletable, and not automatically assigned to Tests.
- When upgrading from **6.0 to 6.1 or higher**, the system:
  - Preserves the existing `admin` role.
  - Adds the new **Built-In Admin role**, which may result in both roles appearing in your environment.
- When upgrading from **5.14 or earlier to 6.1**:
  - Only the Built-In Admin role is created.
  - The previously configured admin LDAP group is assigned to the Built-In Admin role.

Configuring RBAC

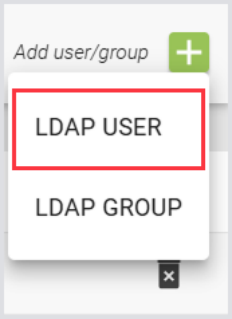
1. Navigate to the **Access Control** page to start managing roles and permissions for users and Tests.



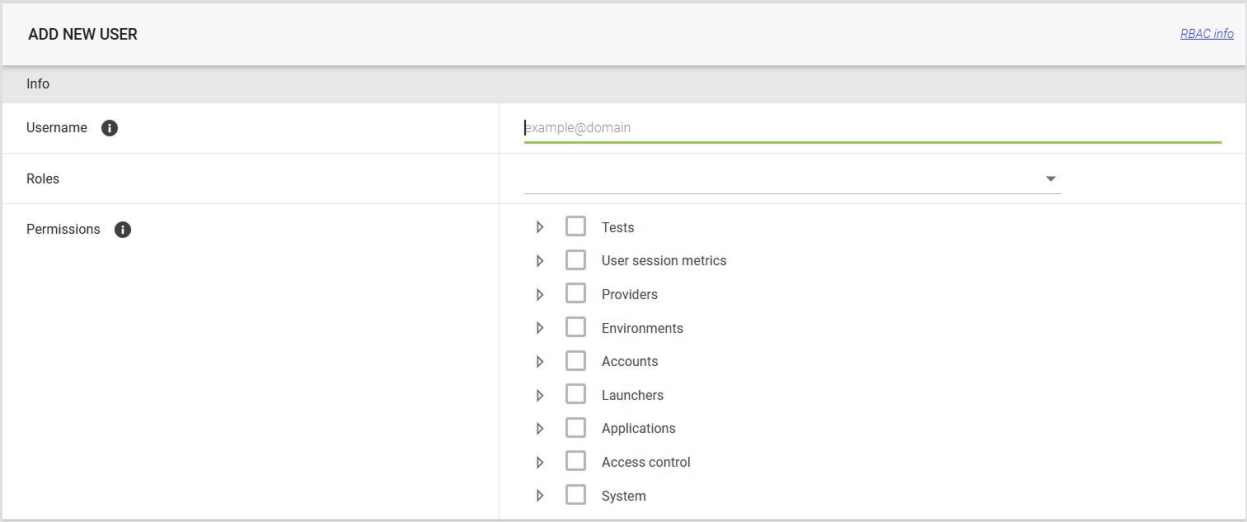
2. To add a new user, click the green "+" to add a user/group.



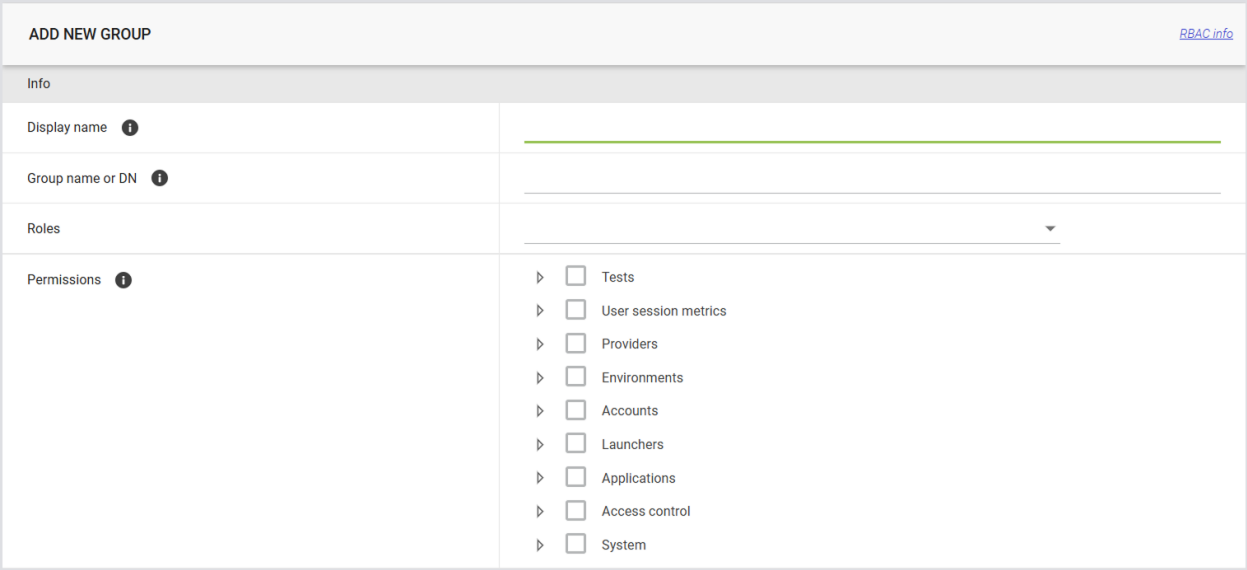
3. Select **LDAP User**.



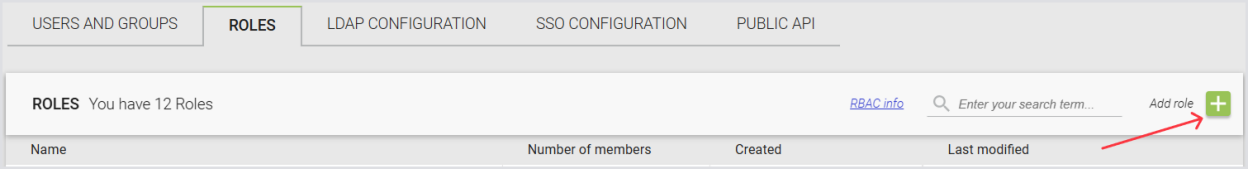
This allows you to define group-based roles and assign permissions accordingly.



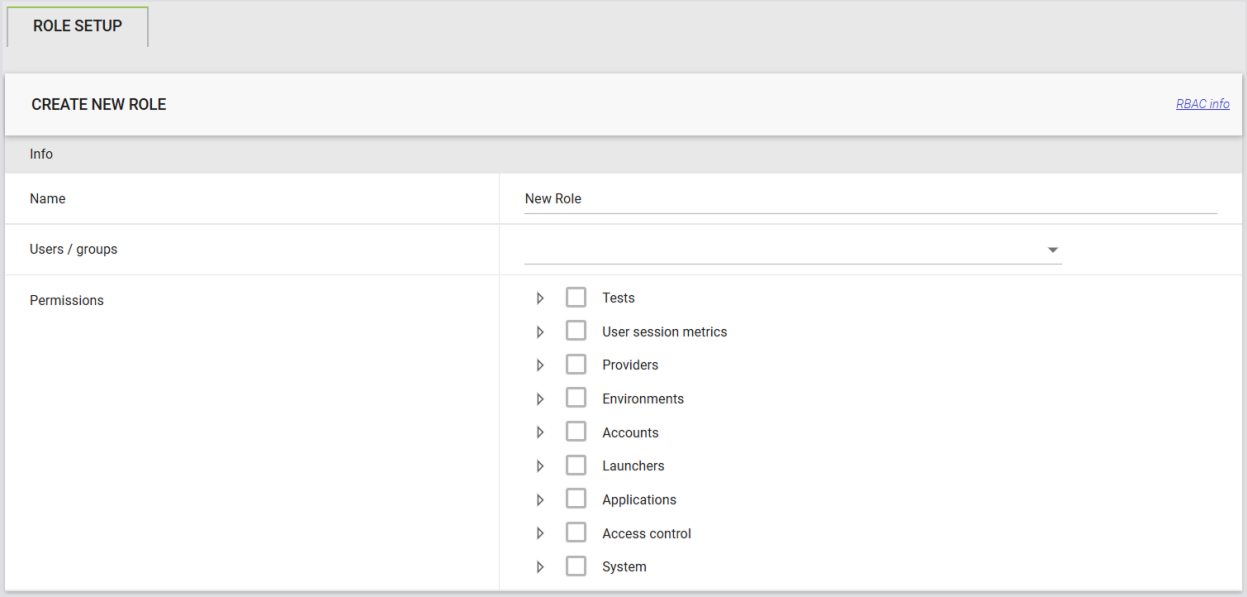
4. To create a new group of users, click the green "+" to add a user/group, and select **LDAP Group**.



5. In **Access Control**, select **Roles** from the tab menu and click the green "+" icon to add a role.

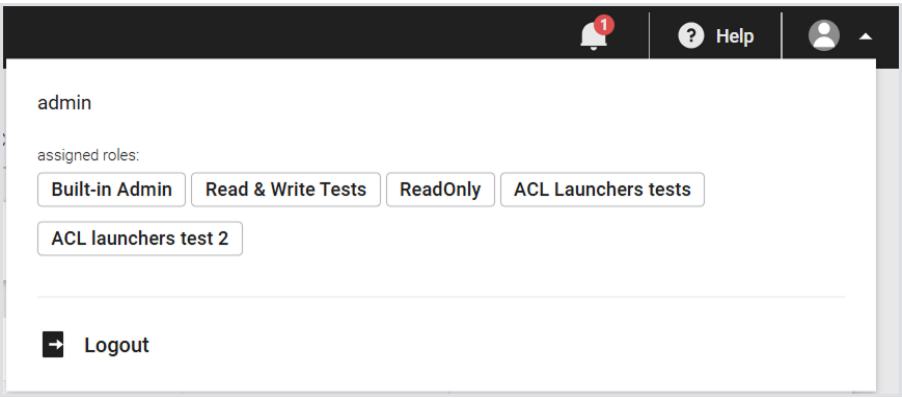


6. Assign permissions based on the access level you want the role to have.



NOTE: The list of permissions for both the user and the user group cannot be adjusted during the user creation or editing steps. The permissions displayed are based on the roles linked to the user to be created. To change permissions, you need to modify the role, not the user.

Click the **User Profile** icon on the top right to see the assigned roles and permissions for each user.



TIP: You can update Tests with roles in bulk using the Public API.

Assigning or Removing Test Access from a Role

You can assign or remove access to multiple Tests directly from a Role configuration. This allows you to manage Test access at scale without relying on the API or editing each Test individually.

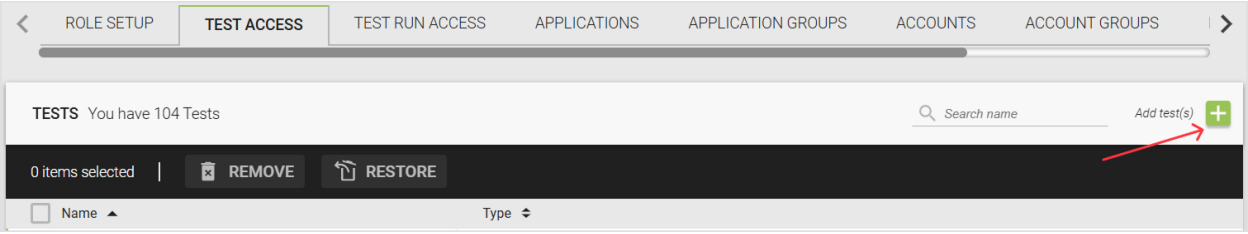
The Tests tab in the Role's details view includes:

- A searchable, sortable table of Tests assigned to the role.
- Options to add or remove one or more Tests.
- Optional filters to narrow the list by Test type.

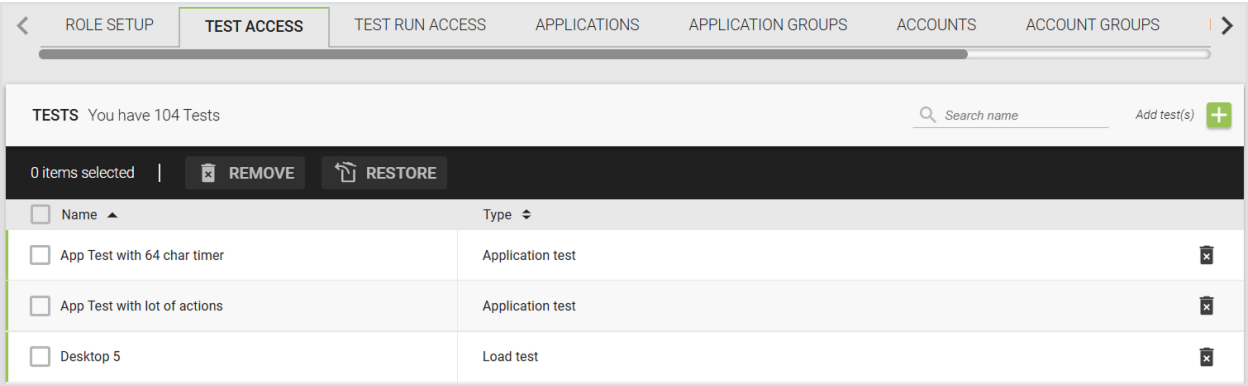
IMPORTANT: Only Tests that the current user has permission to access are displayed. Tests that are assigned to the role but not visible to the current user will **not** appear in the list.

Assigning Test Access

1. In the Role's details view, select **Test Access** from the tab menu, then click the green "+" to add Tests.



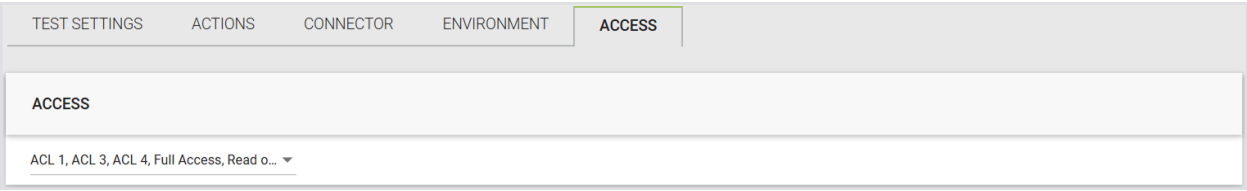
2. Use the search or filters to find the Tests you want to assign.
3. Select one or more Tests and confirm.
4. The newly added Tests will appear at the top of the list, highlighted with a green line.



5. Click **Save** to apply the changes. The green lines will disappear once the changes are saved.

You can also assign Role(s) to Test (Access) from the Test Configuration. This allows you to assign roles to determine which users can access and modify Tests. To do this:

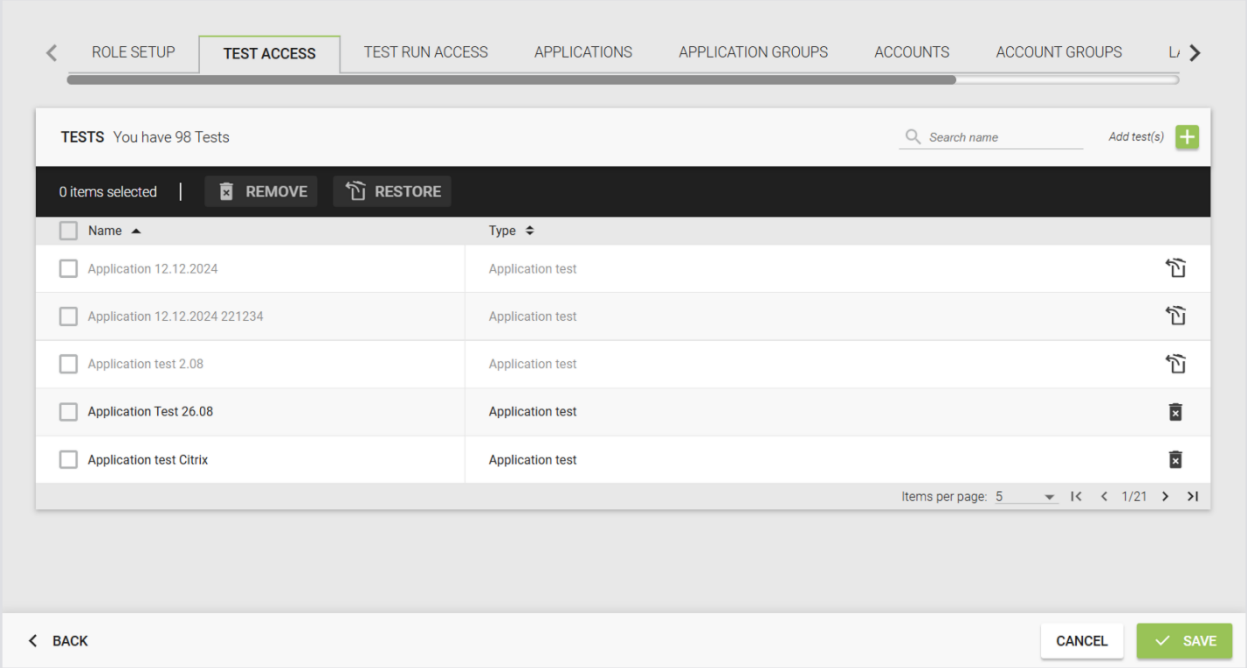
1. In the **Login Enterprise sidebar menu**, go to **Configuration** > **Manage Tests**.
2. Select the Test (e.g., Load Test), then select **Access** from the tab bar menu.



Removing Test Access

1. In the Role's details view, select **Test Access** from the tab menu.
2. Use the filters or search to locate the Tests you want to remove.
3. Select one or more Tests, then click **Remove**.
4. The removed Tests will remain in the list but appear greyed out, indicating they are marked for removal.

5. Click **Save** to apply the changes. The greyed-out rows will disappear once the changes are saved.



Managing Access to Applications and Application Groups

You can control which users can view or manage specific Applications and Application Groups by assigning access roles. This helps ensure that only authorized users, the “owners” of the Application, can manage its settings or include it in Tests.

Assigning Roles to Applications and Groups

You can assign one or more roles to an Application or Application Group to control who can view or manage them. You can only assign roles that you have permission to access.

When creating:

- **Application**: Use the Assign Roles option to define who can access the Application.
- **Application Group**: Use the Assign Roles option to grant access during Group creation.

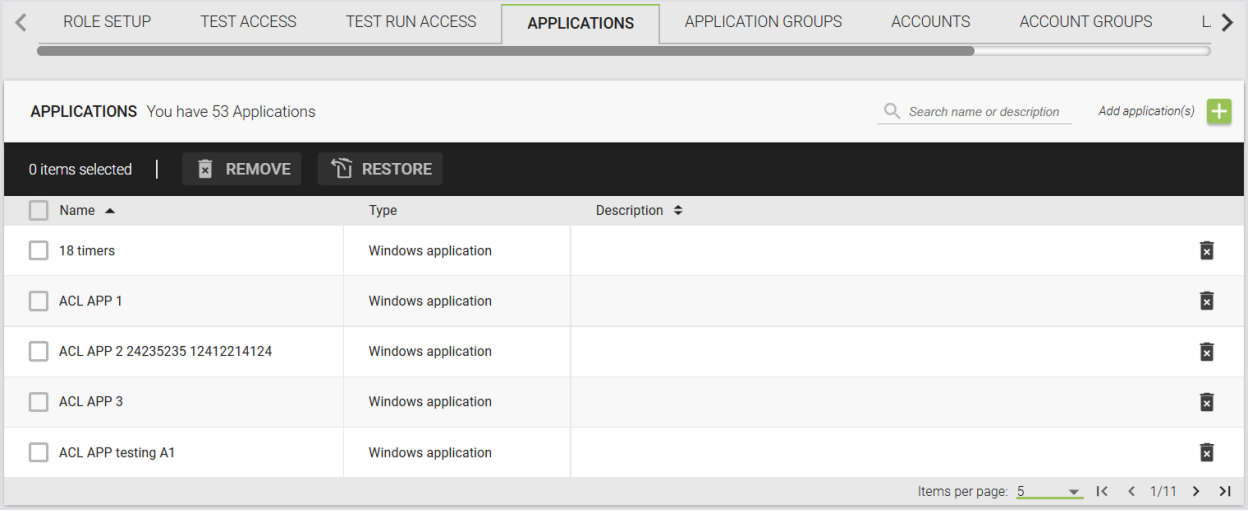
When editing:

- Modify assigned roles directly in the Application or Application Group configuration.
- You can add or remove roles at any time.

Managing Application Access from the Roles UI

When viewing a role, you can manage associated Applications and Groups:

- See a list of Applications and Groups assigned to the role.
- Add one or more Applications or Groups to the role.
- Remove Applications or Groups from the role.



Deleting Roles

When a role is deleted, it is automatically removed from all Applications and Application Groups it was assigned to.

Test Visibility and Behavior with Restricted Apps

- Users can see all Test results, even if the Test includes Applications they don't have access to.
- The Test configuration page displays all Applications assigned to a Test, including those without user access.
- Users cannot add Applications they don't have access to into a Test.
- If a Test already contains Applications without user access, users can perform actions such as:
    - Removing those Applications from the Test
    - Changing their settings
    - Reordering them within the Test

- Once a no-access Application is removed from a Test, users cannot add it back because it will not appear in the Add Application list.

Application Group Behavior

- Application Groups will display all included Applications, even if the user doesn't have access to some of them.
- Users cannot add Applications they don't have access to into a Group.
- If a Group already contains no-access Applications, users can remove, reorder, or change settings for those Applications.
- Users can add new Applications to the Group only if they have access to those Applications.

You can perform bulk role assignments using the Public API, specifically:

- Retrieve Applications not assigned to any role.
- Assign one or more roles to one or more Applications in a single request.

Managing Access to Test Runs

You can grant access to specific Test Runs directly through the UI. This lets you share selected Test Runs with other roles without giving access to the entire Test configuration.

There are two types of access for Test Runs:

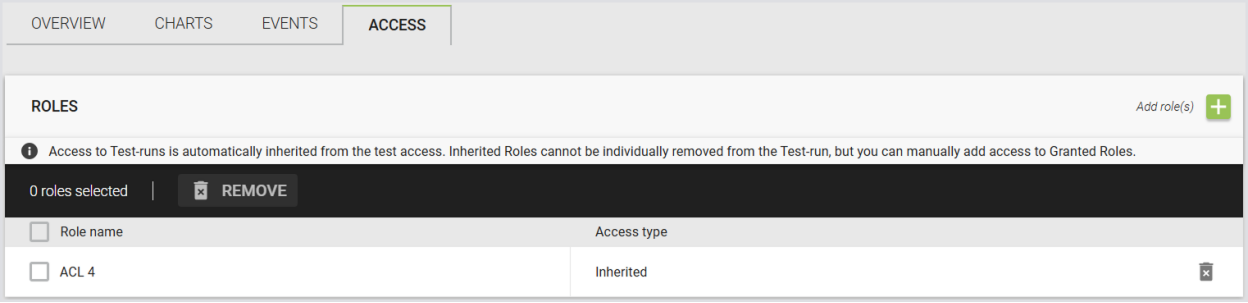
1. **Inherited access**: Access that comes from the parent Test configuration. You cannot remove inherited access from a Test Run. If the parent Test is deleted, the access level of roles inherited from that Test changes to "Granted" and can then be removed.
2. **Granted access**: Access assigned directly to a Test Run. You can manage granted access independently of the parent Test configuration.

This section summarizes the key actions, rules, and special cases that apply when assigning or removing roles from Test Runs.

NOTE: Continuous Tests are out of scope. Because this type of testing doesn't generate separate Test Runs, the setting doesn't apply.

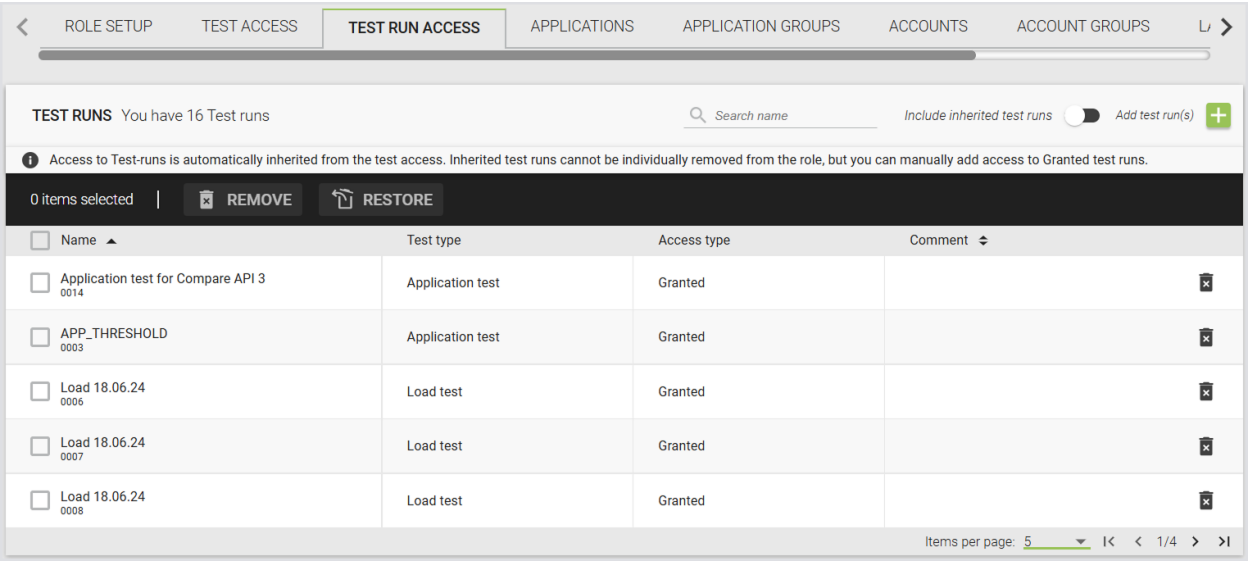
Granting Access to a Test Run

1. In the **Login Enterprise sidebar menu**, go to the **Results** > e.g., **Load Testing**.
2. Open the Test Run, and select the **Access** tab.
3. Check what roles have access to the Test Run.
4. Use the green "+" on the right to add roles.

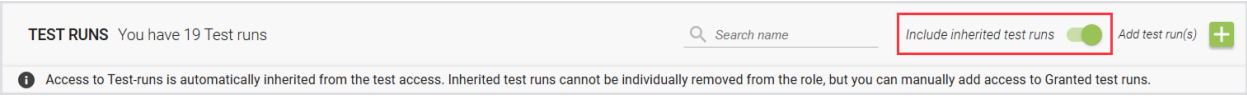


Granting Access to Multiple Test Runs

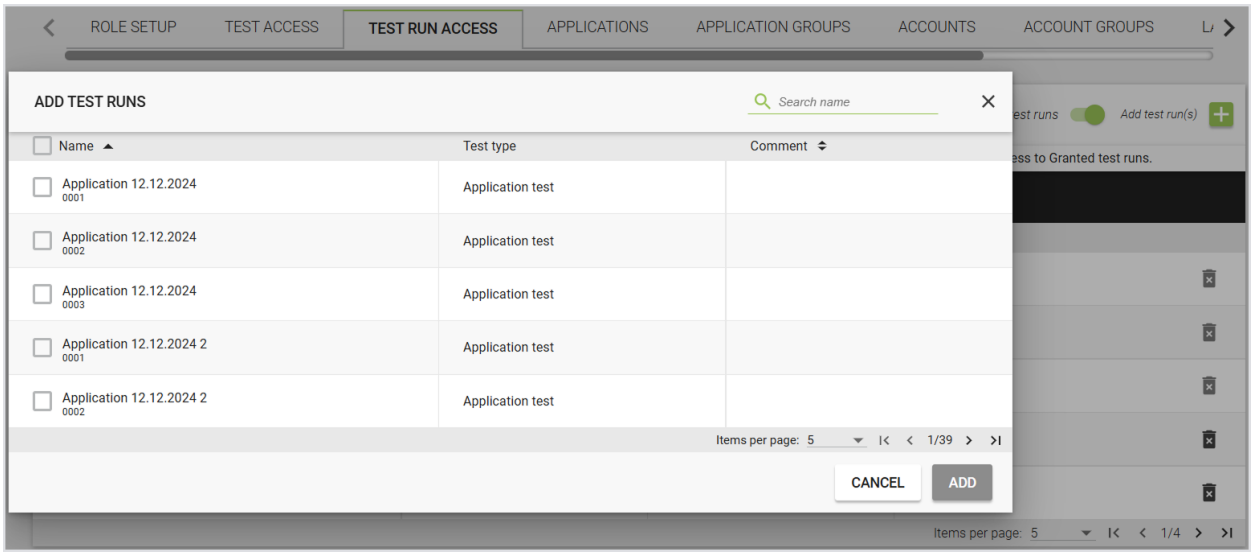
1. Navigate to **Access Control > Roles**.
2. Click on a Role and then select the **Test Run Access** tab.
3. View all Test Runs assigned to the role.



4. Use the toggle to hide or show Test Runs inherited from the Test configuration. This is helpful when a role has access to many Test Runs. Hiding inherited Test Runs lets you focus on the “Granted” (non-inherited) Test Runs.



5. Click the green “+” on the right to add access to more Test Runs.



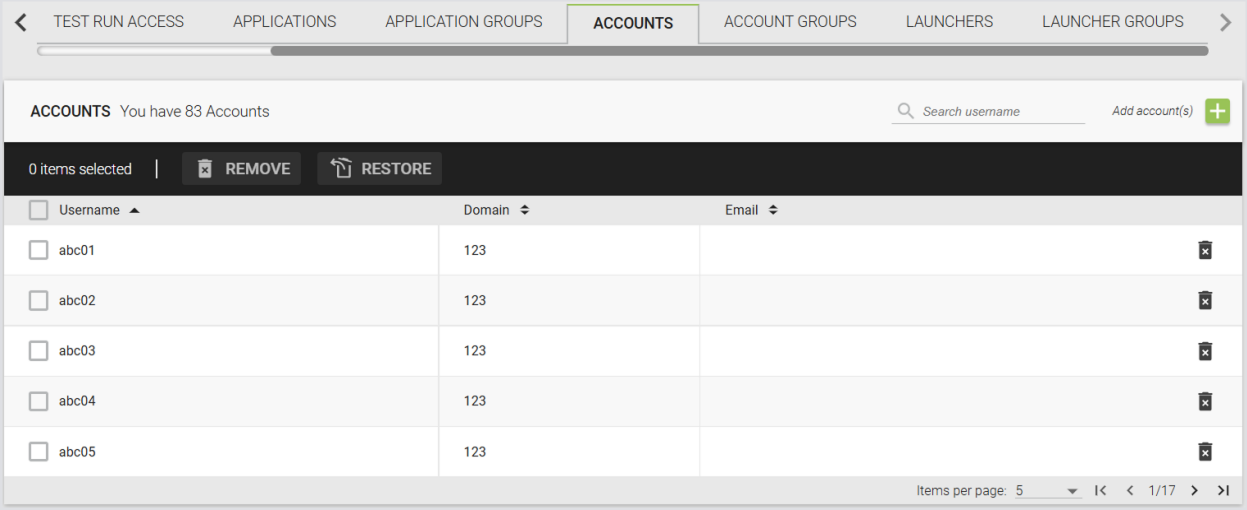
Rules and Exceptions

- **Minimum role requirement:** You cannot remove a role if it would leave a Test Run without any assigned roles. There must always be at least one role assigned to each Test Run.
- **Removing non-inherited ("Granted") Test Runs from a role:** You can remove access to non-inherited Test Runs from a role, unless it is the only role assigned to that Test Run.
- **Removing inherited Test Runs from a role:** Test Runs inherited from the Test configuration can be removed from the role within the Role > Test Run Access tab.
- **Viewing Test Run results:** When you have access to a Test Run, you can see all results, even if you do not have access to one of the applications included in the Test.

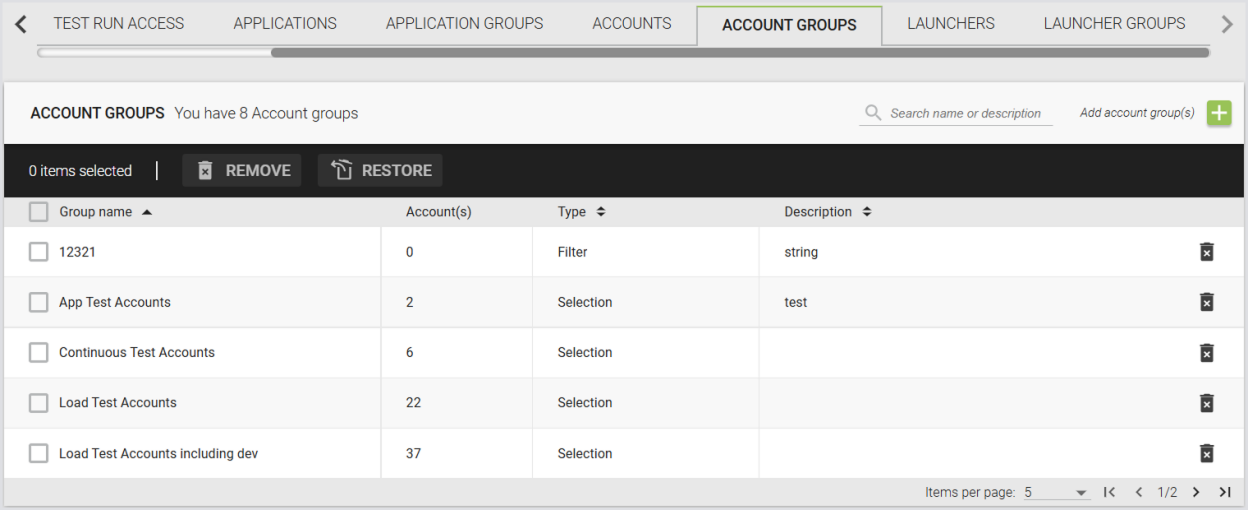
Managing Access to Accounts and Account Groups

You can give roles access to accounts and account groups to control who can view or modify them. You can also view relationships between roles, accounts, and account groups. For example:

- **From the Accounts page:** View which account groups the account belongs to and which roles are assigned to it.



- **From the Account Groups page:** See which accounts and roles are assigned to the group.



Access Behavior when Configuring or Running a Test

When configuring or running a Test:

- A user can select only the account groups assigned to their role.
- A user can view all account groups assigned to a test, including those they don't have access to.
- A user can't add account groups that they don't have access to.
- A user can remove such groups if necessary.

When a Test runs, it uses all account groups assigned to it, regardless of the access rights of the user who started the Test.

Login Enterprise Upgrade Behavior

After upgrading from v6.0–v6.3 to v6.4, all existing roles will have access to all accounts and account groups.

Assigning Accounts and Groups in Bulk

You can assign multiple accounts and account groups to a role at once from the **Account Access** > **Roles** section.

Managing Access to Launchers and Launcher Groups

Launchers are automatically created in the Appliance when the Launcher software is installed on a Launcher machine. After a Launcher registers itself, you need to assign roles to control access manually. You can manage access either through the UI or by using the Public API.

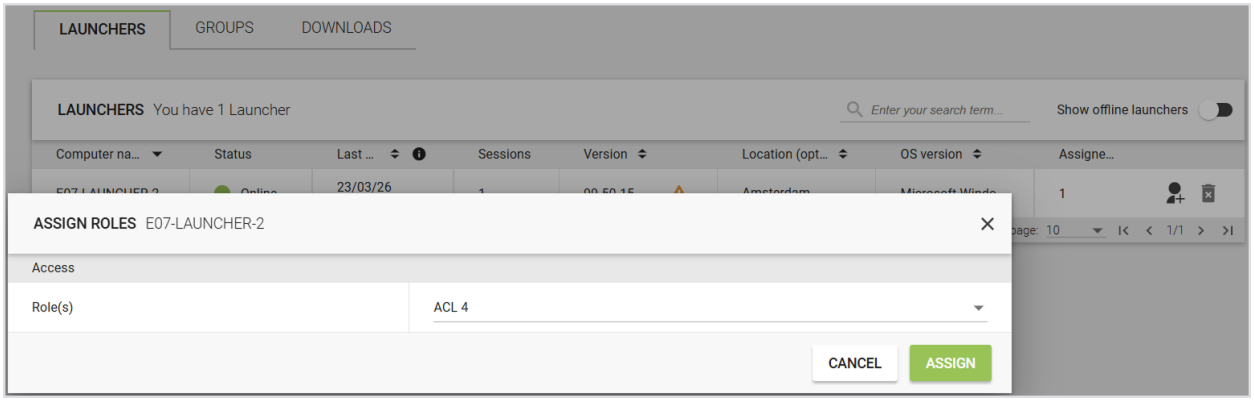
NOTE: Access management for Launchers in the UI is handled from the Launchers overview page or from the Roles page.

Assigning Roles to Launchers

You can assign roles to launchers from either the **Launchers** overview page or the **Roles** page. Both methods provide the same level of access.

From the Launchers overview page

1. In the **Login Enterprise sidebar menu**, navigate to **Launchers**.
2. In the **Assigned roles** column, select the **Add roles** (human-plus) icon for the launcher you want to manage.
3. In the dialog box, select one or more roles from the drop-down list, and click **Assign**.

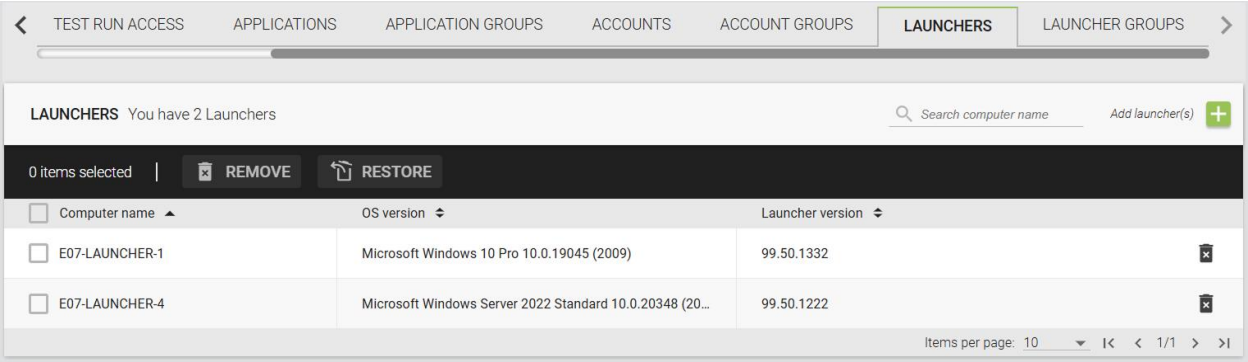


The selected roles are granted access to the Launcher.

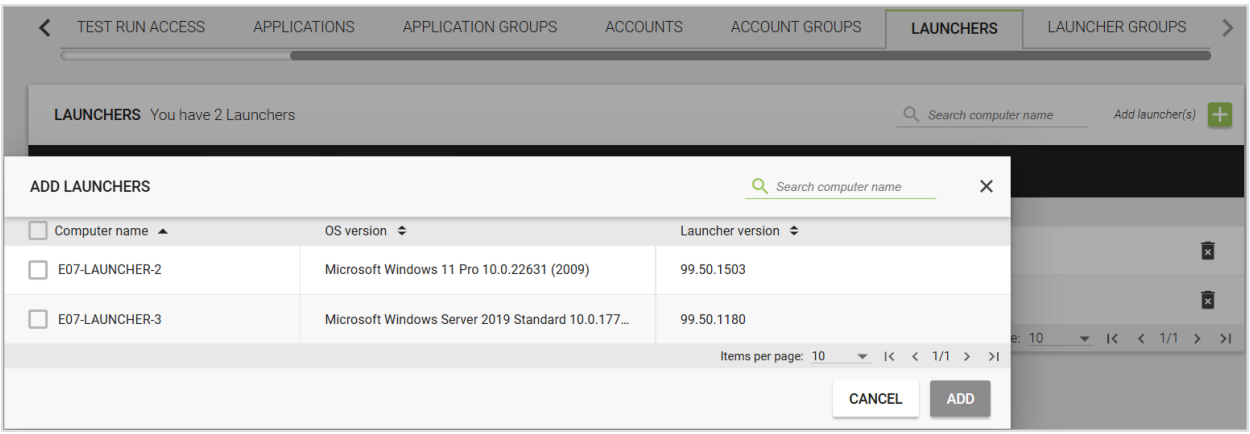
From the Roles page

1. In the **Login Enterprise sidebar menu**, navigate to **Access Control** > **Roles**.

2. Select a role, then select **Launchers** from the tab to view the Launchers currently assigned to the role.



3. Click the green "+" to add Launchers, select one or more launchers, and then click **Add**.



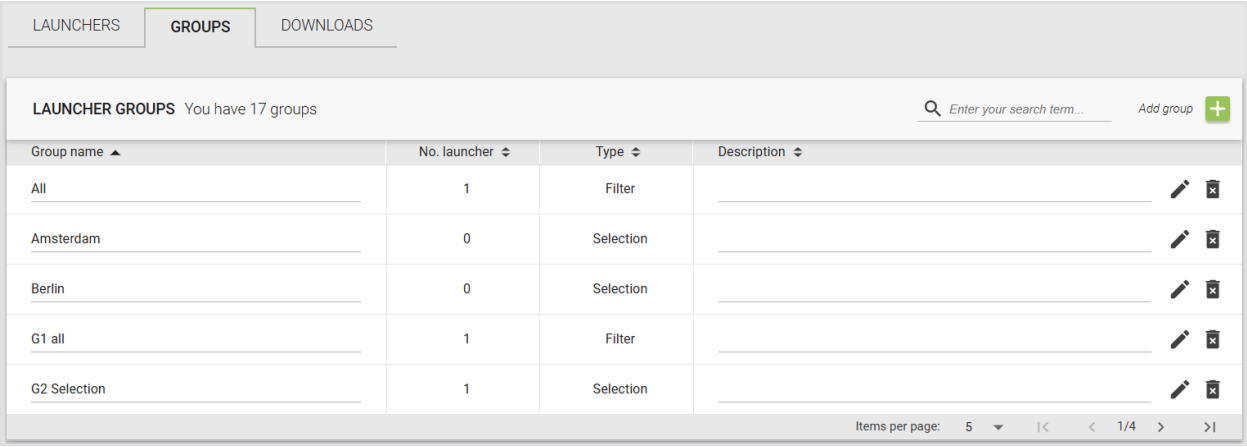
The selected Launchers are now accessible to users with that role.

Managing Access to Launcher Groups

Launcher Groups have a dedicated page where you can create and edit groups. You can add Launchers in one of the following ways:

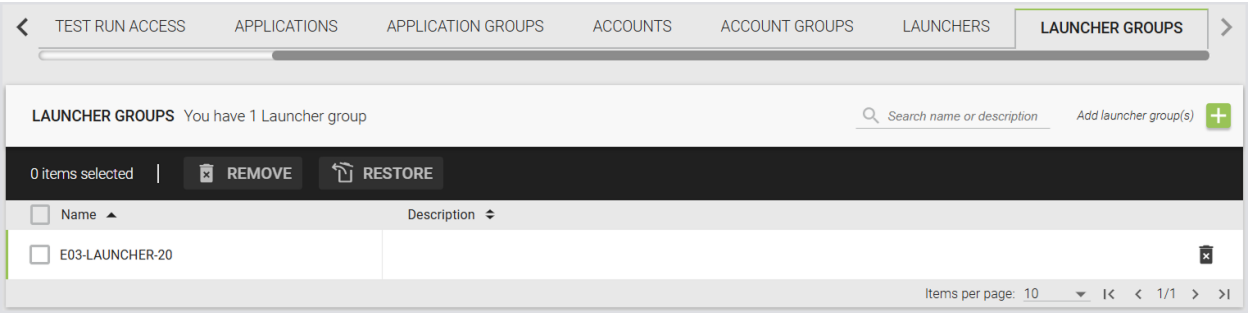
1. **Select Launchers manually**: You can only select Launchers that you already have access to.
2. **Use a filter**: When defining a filter, only the Launchers you have access to are shown while configuring the group.

NOTE: When running a Test that uses a Launcher Group defined by a filter, the system does not evaluate the access of the user who created the group. As a result, the Test may run on additional launchers beyond those visible to the group creator. This behavior is consistent with how Account Groups work.



Assigning Roles to Launcher Groups

Roles can also be assigned to Launcher Groups to control access. You manage role assignments for Launcher Groups in the same way as for other access-controlled components.



Once a role is assigned to a Launcher Group, users with that role gain access according to the group configuration.

Managing Roles and Permissions

Main Category	Subcategory	Permission Type	Description
Tests			
	Results	Read	Gives read access to Test results (all Test types).
		Read and write	Gives permissions to view and delete results and edit the comment (all Test types). Also, it gives access to the Dashboard.
		Read	Gives access to view Test config (all test types).

	Configuration	Read and execute	Gives access to view the Test config and to start and stop the Tests.
		Read, execute, and write	Gives access to view, start/stop, and edit Tests.
	Access Control	Read	Allows viewing the Access tab in the Test configuration to see which roles have access to a specific Test. Users will be able to see the roles assigned to the Test, but only the ones they have access to.
		Read and write	Allows adding or removing roles from a Test. Users can only assign and/or remove roles they have access to.
Session metrics		Read	Gives access to view Session Metrics.
		Read and write	Gives access to add, edit, and delete Session Metrics.
Providers		Read	Gives access to view Providers.
		Read and write	Gives access to create, edit, and delete Providers.
Environments		Read	Gives access to view Environments.
		Read and write	Gives access to create, edit, and delete Environments.
Accounts	Configuration	Read	Gives access to view Accounts and Account Groups.
		Read and write	Gives access to add, edit, and delete Accounts and Account Groups.
	Access Control	Read	Allows viewing the Roles field in Account and Account Group configuration to see which roles have access to a specific Account or Account Group. Users will be able to see the roles assigned to the Account, but only the ones they have access to.
		Read and write	Allows adding or removing roles from an Account or Account Group. Users can only assign and/or remove roles they have access to.
	Configuration	Read	Gives access to view Launchers, Launcher Groups, and Locations.

Launchers		Read and write	Gives access to add, edit, and delete Launcher Groups.
	Access Control	Read	Allows viewing role assignments for Launchers and Launcher Groups. Cannot assign or remove roles.
		Read and write	Allows viewing and assigning roles to Launchers and Launcher Groups.
Applications	Configuration	Read	Gives access to view Applications and Application Groups.
		Read and write	Gives access to add, edit, and delete Applications and Application Groups.
	Access Control	Read	Allows viewing the Access tab in Application and Application Group configuration to see which roles have access to a specific Application or Application Group. Users will be able to see the roles assigned to the Test, but only the ones that they have access to.
		Read and write	Allows adding or removing roles from an Application or Application Group. Users can only assign and/or remove roles they have access to.
Connectors	Windows365	Read	Allows viewing the Windows 365 connector configuration, including the active script (default or custom) and available scripts. Users can view and download scripts but cannot upload, delete, or modify them.
		Read and write	Allows full management of the Windows 365 connector configuration. Users can upload, replace, and delete custom scripts and change the active script.
Access Control		Read	Allows viewing the Access Control section in the UI, including all users and roles the user has access to.
		Read and write	Allows creating, editing, and deleting users and roles within the Access Control section, limited to those the user has access to. To add access to specific components on roles in this section, you also need access control permissions for the specific component.

System			
	Configuration	Read	Gives access to view External Notifications, general System Settings, Database Configuration, Active Sessions, Container Status, and Appliance Health.
		Read and write	Gives access to update External Notifications, general System Settings, Database Configuration, Active Sessions, Container Status, and Appliance Health.
	Components		
		Downloads	Gives access to view the Downloads page and the Download tabs within the Launchers, Accounts, and Application pages. Also includes access to the Event Logger tab in External Notifications.

Understanding Permission Dependencies

To manage and assign Launchers, Accounts, Applications, and Environments, users must have specific permissions. The following outlines the required permissions for each:

- To view and assign **Launchers** to Tests (Config), the user must have **read** permissions for Launchers.
- To view and assign **Accounts** to Tests (Config), the user must have **read** permissions for Accounts.
- To view and assign **Applications** to Tests (Config), the user must have **read** permissions for Applications.
- To view and assign **Session Metrics** to Tests (Config), the user must have **read** permissions for Session Metrics.
- To view and assign **Environments** to Tests (Config), the user must have **read** permission for **Environments**.
- To view and assign **Tests** to Environments, the user must have **read** permissions for **Environments** and **Test Configuration**.
- To access the **Downloads** tab for Launchers, Accounts, and Applications, the user must have **read** permissions for **System > Components > Downloads**.
- To view **Session Metrics** in Continuous Test results, the user must have **read** permissions for **Session Metrics**.
- To view the **threshold line** in the charts of Continuous Test results, the user must have **read** permissions for **Test Configuration**.
- To view and assign roles to Tests, Applications, or Accounts, the user must have **Read** or **Read and write** permissions for **Access Control** for each of the specific components.
- To view and assign access to specific components (Tests, Applications, or Accounts) on roles via the Access Control section, the user must have **Read** or **Read and write**

permissions for both **Access Control (global)** and for the **related component** (Tests, Applications, or Accounts).

Other Permission Details

- The **About** and **Licensing** pages are visible to all users, regardless of their permissions. However, to perform actions on these pages, users must have **System Config** permissions (read/write).
- To view the **Operations Dashboard**, the user must have **read** permissions for **Environments**.

Endpoints for Access Management

You can manage access to Tests, Test Runs, Applications, and Application Groups directly through the Virtual Appliance web interface. This is the recommended method for most scenarios. If needed, you can also perform access management tasks programmatically by using the Public API.

Managing Licenses

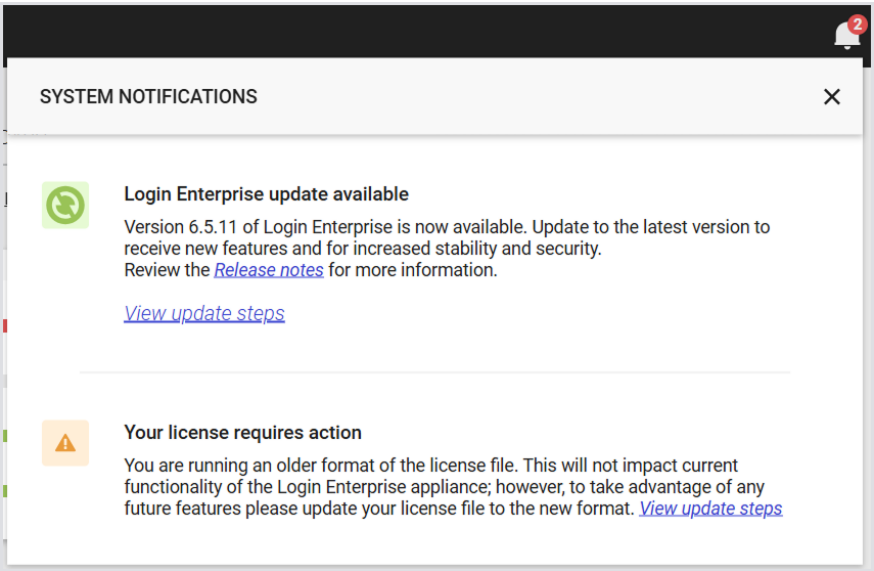
- Overview
- License notifications
- Updating a License

Overview

To ensure proper testing, you need to upload the Login Enterprise license to the Virtual Appliance. This license contains important information, including the expiration date, the number of permitted Virtual Appliance Launchers, the allowed number of Test Users, and the maximum number of Test Applications.

A newly set up Appliance includes a default 3-day grace period. Uploading your valid license will replace this grace period. New virtual user sessions cannot be launched without a license during testing. However, access to previously gathered data remains unrestricted.

Alerts in the top menu section display a notification icon when grace licenses or production licenses are about to expire. Below is an example of such an indication.

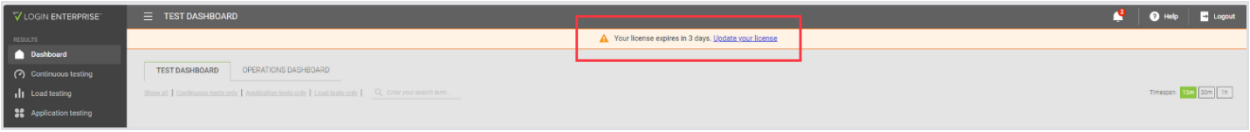


NOTE: If you do not have a license, contact us. For assistance with an existing license, reach out to the Login VSI Customer Success or Support teams.

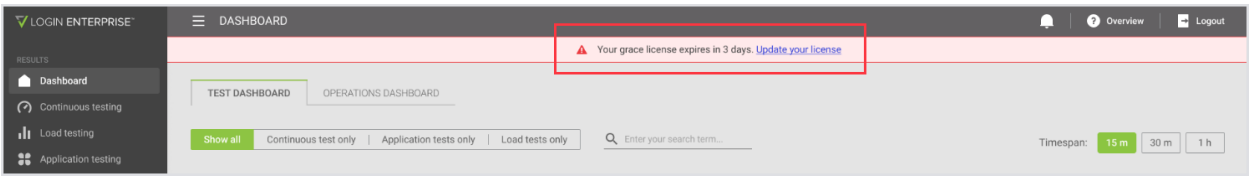
License notifications

Notifications regarding your license status appear in the top panel of the Test dashboard. Below are examples of messages indicating a license that is about to expire and a license that has already expired:

- **License about to expire**: Your license expires in 3 days. Update your license.



- **License already expired**: Your grace license expires in 3 days. Update your license.

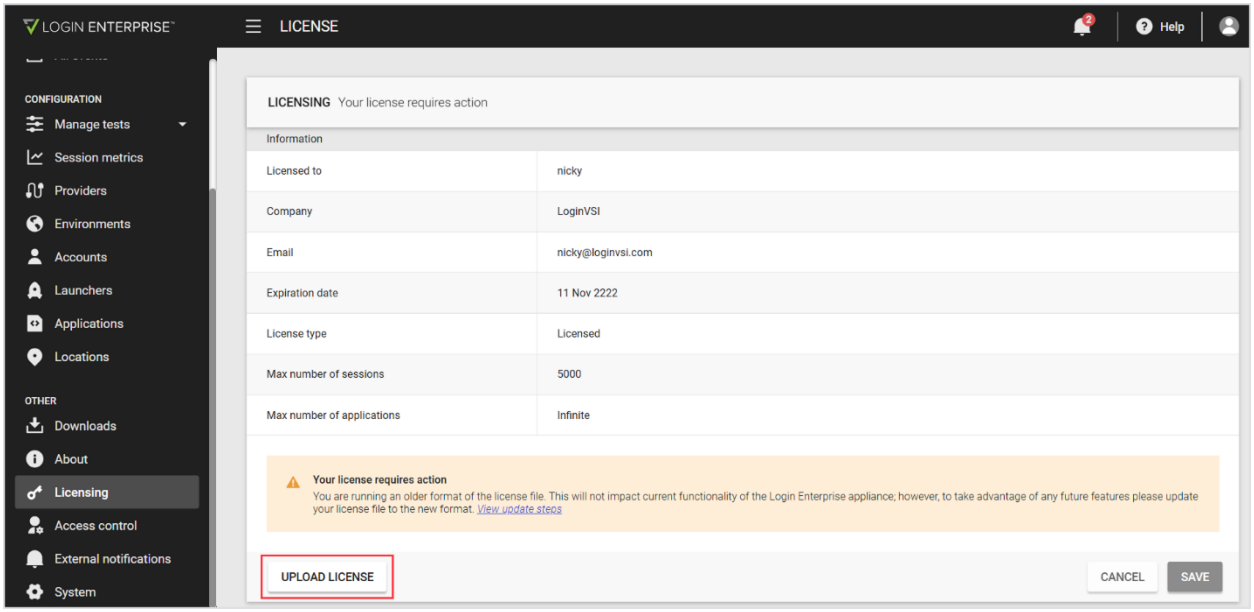


Updating a License

You can import the license, e.g., license.lic at any time during the grace period or after it has expired. To request a license, contact your sales representative or support. Save the valid license on a drive accessible via the Login Enterprise web console.

To update a License:

1. In the **Login enterprise sidebar menu**, navigate to **Other > Licensing**.
2. In **Licensing**, click **Upload license**.



3. Select the license, then click **Open** to import it. Once the license is imported, you will see the details of the license.

4. After importing the file, click **Save** to apply the changes.

The new license information will be available when you go back to the **Sidebar menu > Other > Licensing**.

Using the Maintenance Menu

- Overview
- Accessing the Maintenance Menu
- Using the Maintenance Menu
  - System
    - Change the hostname
    - Change your password
    - Regenerate self-signed certificates
    - Apply new certificate
    - Import root CA certificate
    - Change allowed URLs
    - Perform DB maintenance
  - Updates
    - Update from the internet
    - Update from ISO
    - System update from the internet
  - Network configuration
    - Show network interfaces
    - Show routes
    - Configure proxy
    - Use DHCP
    - Set static IP
  - Troubleshooting
    - Ignore certificate errors
    - Start enable SSH server
    - Stop and disable SSH server
    - Ping
    - Show netstat
    - Show LoginVSI service status
    - Restart LoginVSI services
    - Open bash shell

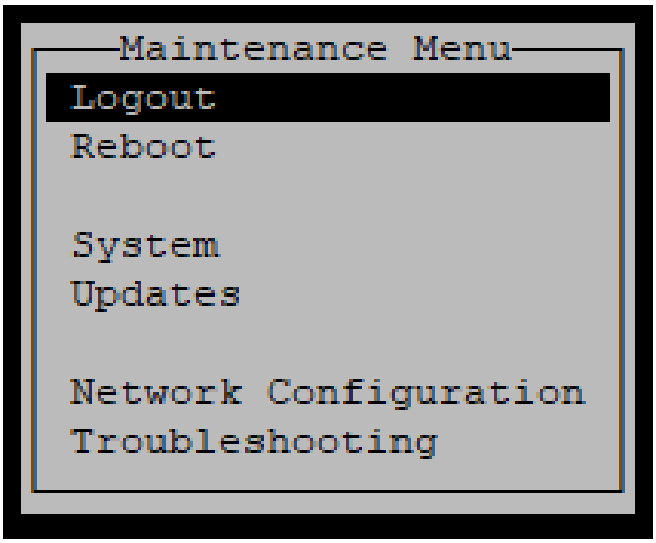
Overview

The Maintenance Menu allows you to manage and troubleshoot your Virtual Appliance. This section provides step-by-step instructions for accessing and using the various options available on the Maintenance Menu.

```
Welcome to Login Enterprise - version 5.13.6

The Login Enterprise Webinterface is available at:

 https://rc-app-pool-6.qa.loginvsi.com
 https://10.30.55.12

 NOTE: Please make sure to update your dns records accordingly

RC-app-pool-6 login:
```

Accessing the Maintenance Menu

Ensure you are logged in successfully to view the Maintenance Menu. To access the menu, enter your previously defined username and password. Once logged in, you will have access to the following options:



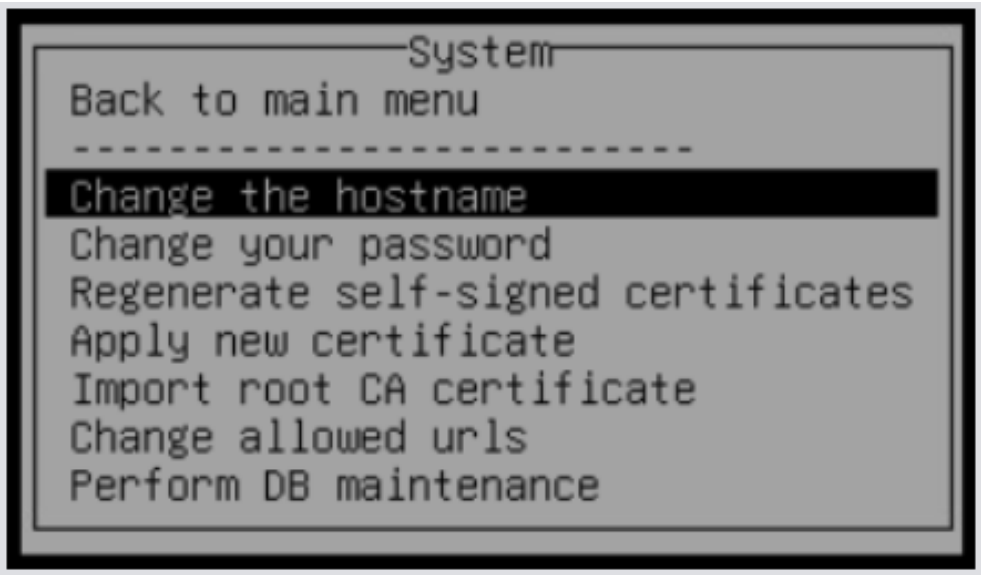
- **Logout**: Logs you out of the Maintenance Menu.
- **Reboot**: Reboots the Virtual Appliance.
- **System**: Provides additional functionalities such as changing the hostname, updating the Admin password, and managing certificates. For more details, see System.
- **Updates**: Offers different methods for updating the Virtual Appliance. For more details, see Updates.
- **Network configuration**: Allows you to check and modify network interfaces and routes. For more details, see Network configuration.
- **Troubleshooting**: Provides tools and options for diagnosing and resolving issues. For more details, see Troubleshooting.

Using the Maintenance Menu

System

This submenu gives you additional options, with which you can:

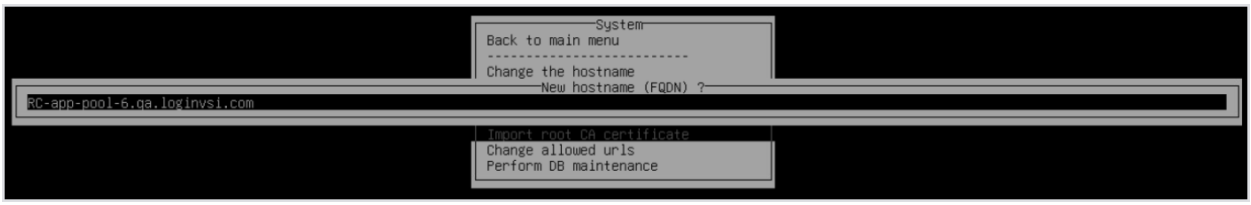
- Change the hostname
- Change your password
- Regenerate self-signed certificates
- Apply new certificate
- Import root CA certificate
- Change allowed URLs
- Perform DB maintenance



IMPORTANT: If you regenerate a certificate, install a new server certificate, or change the hostname, any installed Launchers must be reinstalled whenever your certificate changes.

Change the hostname

This option allows you to change the hostname at the Virtual Appliance level.



NOTE: When you change the hostname of your Virtual Appliance, it is recommended to regenerate the self-signed certificates, provided you are using them instead of a custom certificate. This ensures that the certificates reflect the new hostname.

Best practice: After changing the hostname, regenerate the self-signed certificates and restart the relevant services.

Change your password

This option opens a wizard to change the Admin password of the Virtual Appliance.

NOTE: Changing the password affects both the login credentials for the Maintenance Menu and the Login Enterprise web interface, as both use the same account (admin).

Reasons to change the password:

- To revoke access for employees who are no longer with the company.
- To comply with security policies that require password changes after a specified period.

Regenerate self-signed certificates

This option allows you to regenerate the self-signed certificate in use.

Reasons to regenerate SSL certificates:

- **Configuration changes:** If you change the appliance's hostname, IP address, or other network settings, the existing certificate may become invalid. Regenerating self-signed certificates ensures they align with the updated configuration and maintain secure connections.
- **Compliance:** Certain security policies or audits may require the periodic regeneration of certificates to ensure that communication channels remain secure and up to date.

Apply new certificate

This option allows you to add a custom PFX certificate to the Virtual Appliance. For more information, see Installing new certificates.

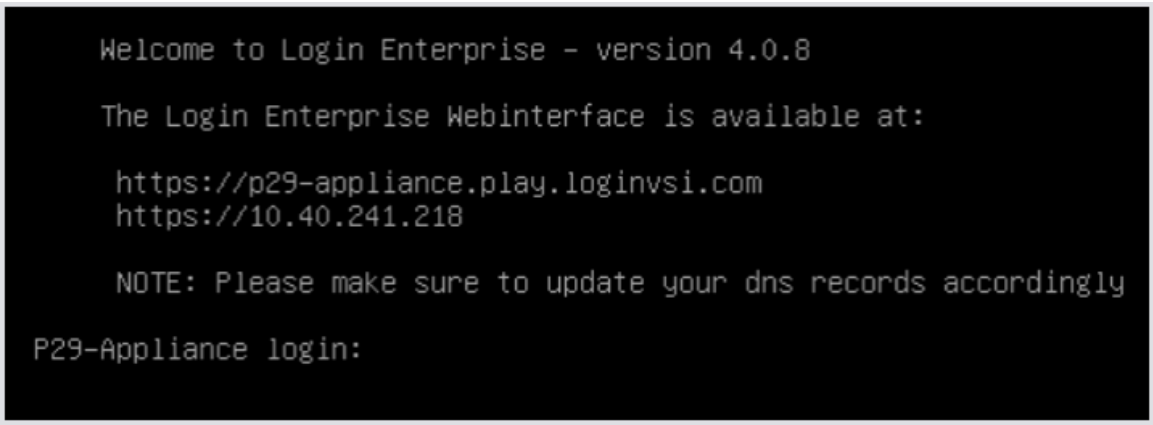
Import root CA certificate

Use this option to import a Root Certificate Authority (CA) certificate when applying custom certificates. This is essential for establishing trust between your custom PFX certificate and the system. For more information, see Importing a root CA certificate.

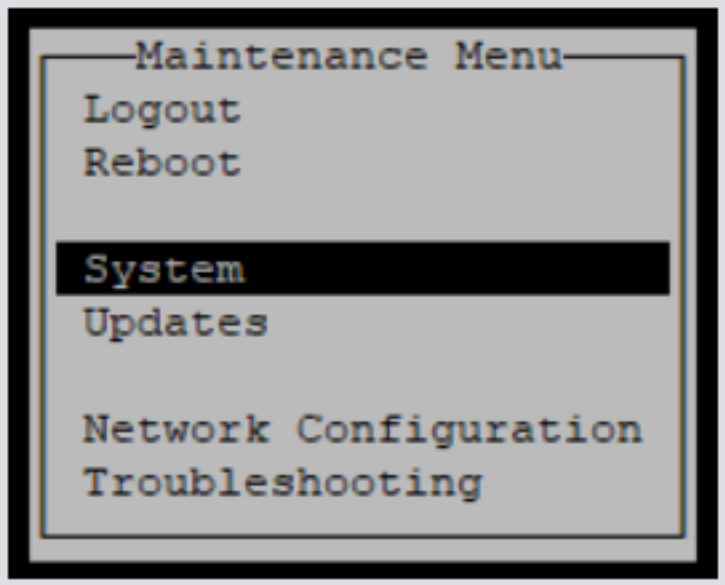
Change allowed URLs

This option allows you to add or modify URLs that the Virtual Appliance listens to. In some scenarios, you may prefer to use a URL other than the hostname or IP address of the Appliance. Following the steps below will help ensure your Virtual Appliance is accessible via the desired URL and that your environment is properly configured to recognize it.

1. Open a connection to the Virtual Appliance console.

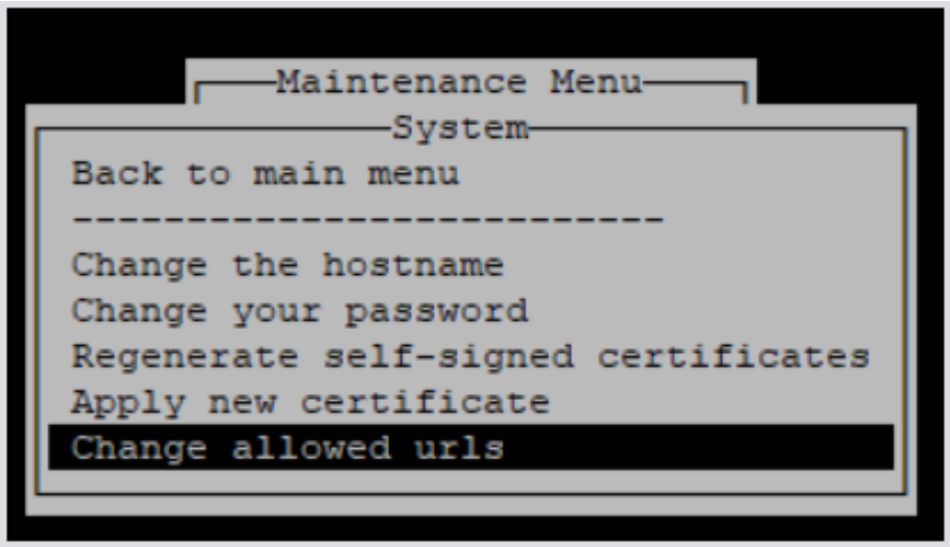


2. Log in to the Appliance and select **System**.

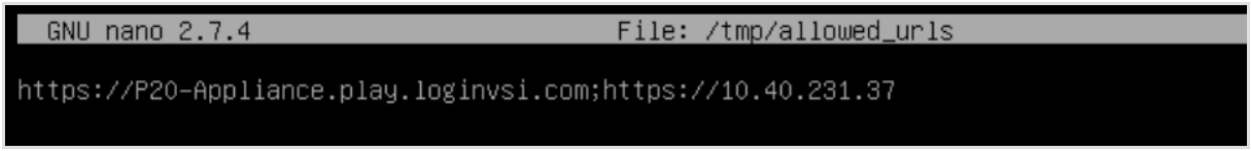


NOTE: Before you proceed, make sure to stop running your Tests.

3. From the **System** menu, select **Change Allowed URLs** (Allowed URLs refer to the new URL where the Virtual Appliance will be accessible).



4. This takes you to the built-in GNU nano editor. The values are semicolon-delimited.



5. Delete any unnecessary URLs and add the new URL as needed.

The first entry in the list must be the name in the certificate. If your certificate has multiple SANs, the first entry must be one of them. The remaining URLs can be in any order. Make sure your URLs are valid and that you do not have a ; (semicolon) at the end of the line. Semicolons must only be used between each URL in the list. For example:

```
  GNU nano 2.7.4                    File: /tmp/allowed_urls

https://P20-Appliance.play.loginvsi.com;https://10.40.231.37;https://MyNew.Url.Co.Uk
```

6. Press **Ctrl + O** to save the changes, then hit **Enter** to confirm overwriting the existing file.

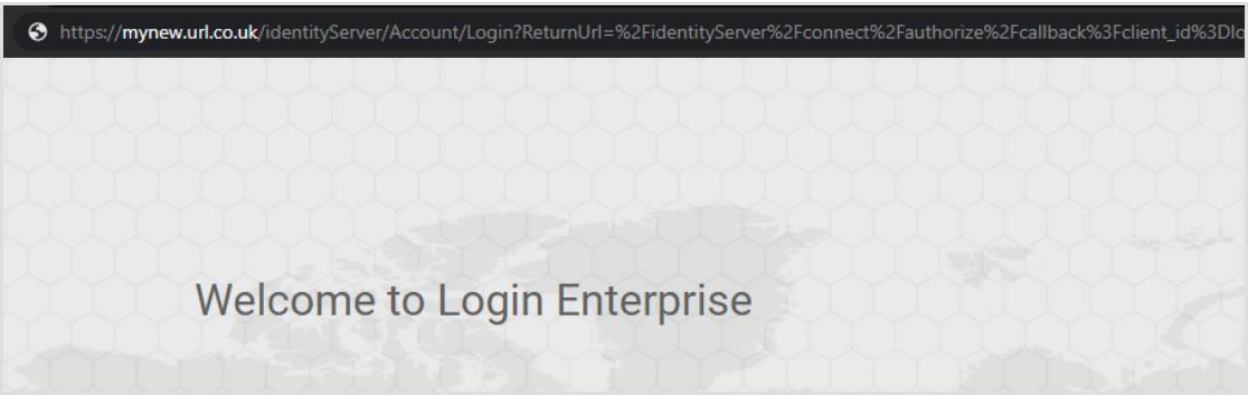
```
File Name to Write: /tmp/allowed_urls
```

7. Press **Ctrl + X** to exit GNU nano.

```
Press Enter to return to Pdmenu.
```

8. From the main menu, select **Reboot** to restart the Appliance.

After rebooting, verify access to the new IP address of the Virtual Appliance front end using a web browser:



Ensure that the rest of your environment recognizes the new Appliance URL so that all parts of the network can connect to it.

After configuring the new URL, ensure that you update the Launchers to the latest version. Even if the version number remains the same, Login Enterprise updates the Launcher files with the new URL name.

For more detailed information, see Launcher life cycle and best practices.

Perform DB maintenance

This option optimizes your database by rebuilding indexes and updating statistics, which improves query performance and reduces fragmentation. This process automatically adapts to the type of database in use—whether it's the internal PostgreSQL database or an external Microsoft SQL database, depending on your setup.

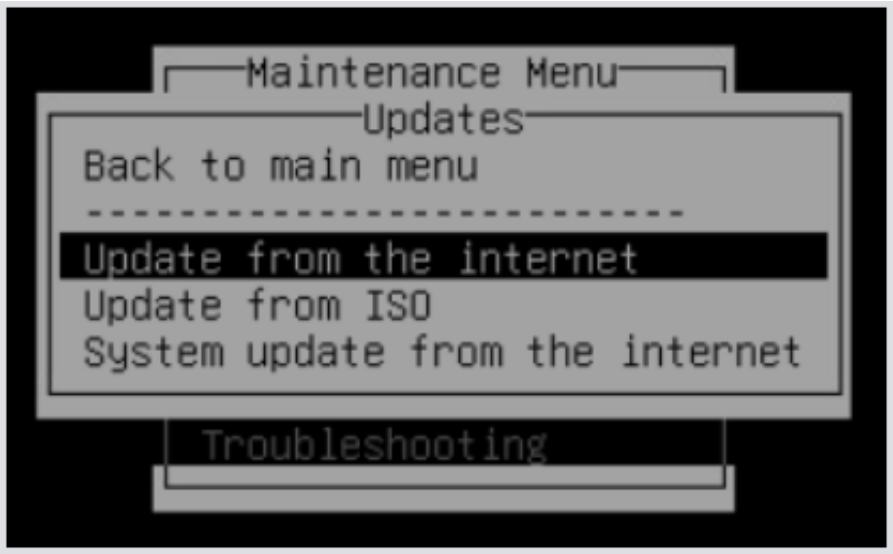
IMPORTANT: Login Enterprise services will be temporarily stopped during maintenance. It's important to schedule this process accordingly. Always back up your database or Virtual Appliance before proceeding to prevent potential data loss.

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#                                                                         #
#                        DATABASE MAINTENANCE                             #
#                                                                         #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
#                                                                         #
#  Database (DB) maintenance will rebuild the index of the Login          #
#  Enterprise DB tables and update its statistics.                        #
#                                                                         #
#  A highly fragmented database may see improved DB query                 #
#  performance after DB maintenance.                                      #
#                                                                         #
#  Depending on the database size, it may take some time for DB           #
#  maintenance to complete.                                               #
#                                                                         #
#  It is recommended to backup your database (external DB) or             #
#  virtual appliance (internal DB) before proceeding.                     #
#                                                                         #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
Configured database type is Internal PostgreSQL Database
LE services will be stopped in order to perform DB maintenance.
Continue with DB maintenance? [y/N]: y
Stopping Login Enterprise services...[DONE]
Login Enterprise started

Press Enter to return to Pdmenu._
```

Updates

This submenu provides methods for updating the Virtual Appliance:

- Update from the Internet
- Update from ISO
- System Update from the Internet



Update from the internet

This option initiates an update over the internet, bringing the virtual appliance to the latest available version. For more information, see the Online update of Login Enterprise.

Update from ISO

This option initiates an update from an attached ISO applied to the VM. For more information, see the Offline update of Login Enterprise.

System update from the internet

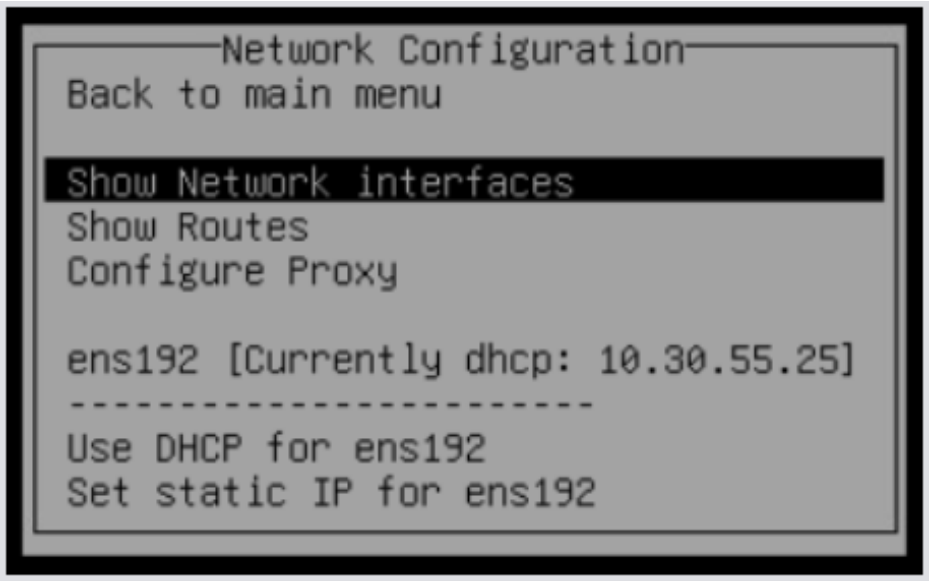
This option initiates Debian system updates over the Internet. For more information, see the System update of the Login Enterprise operating environment.

Network configuration

The submenu allows you to:

- Show Network interfaces

- Show routes
- Configure Proxy
- Use DHCP
- Set Static IP



Show network interfaces

This option allows you to verify specific details, such as:

- **Interface status (UP/DOWN)**: Ensure the network interface is active. If it's down, the device will not communicate.
- **IP address**: Verify the correct IP assignment. An incorrect or missing IP can cause connectivity issues.
- **Subnet mask and gateway**: Check for proper network routing. Mismatched settings can prevent access to other devices or networks.
- **MAC address**: Confirm the correct hardware identifier for security or access control.
- **Link speed/duplex**: Ensure the network operates at expected speeds and in full-duplex mode to avoid performance issues.
- **Errors/packet loss**: Look for network errors or dropped packets, which may indicate hardware or configuration issues.
- **Active interface**: Verify that the correct interface, e.g., wired or wireless, is being used.

These checks help quickly identify and resolve connectivity, configuration, or performance issues.

```
Login Enterprise - 5.13.6
Show Network interfaces
docker0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 172.17.0.1  netmask 255.255.0.0  broadcast 172.17.255.255
        ether 02:42:7f:76:37:88  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
docker_gwbridge: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.126.1  netmask 255.255.255.0  broadcast 192.168.126.255
        ether 02:42:33:f8:4b:5e  txqueuelen 0  (Ethernet)
        RX packets 2645808  bytes 235175844 (224.2 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1647399  bytes 210730355 (200.9 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
ens192: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.30.55.25  netmask 255.255.0.0  broadcast 10.30.255.255
        ether 00:50:56:96:83:e0  txqueuelen 1000  (Ethernet)
        RX packets 13950999  bytes 1058943211 (1009.8 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 372329  bytes 33222886 (31.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 260768  bytes 24551202 (23.4 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 260768  bytes 24551202 (23.4 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
veth5307320: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 3e:21:f4:59:f8:91  txqueuelen 0  (Ethernet)
        RX packets 33  bytes 2482 (2.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 45  bytes 4146 (4.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
veth9768499: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 1a:ce:a3:f7:d8:08  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 1  bytes 42 (42.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
veth0439b7e: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        ether 7e:17:bb:fc:fe:30  txqueuelen 0  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
```

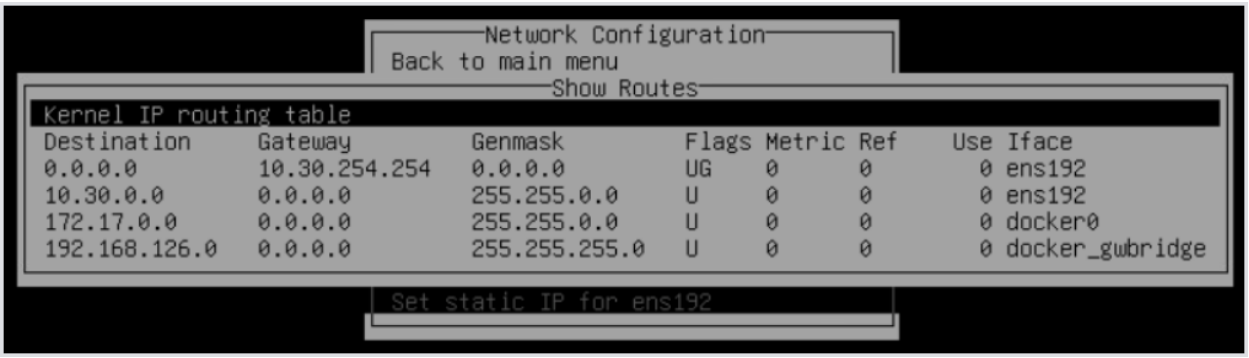
Show routes

This option allows you to review networking details, such as the following:

- **Routing table**: Confirm active network routes to ensure traffic is routed correctly to both internal and external networks.

- **Default gateway**: Validate the correct default gateway for routing outbound traffic.
- **Subnet routes**: Ensure specific routes to critical subnets or VPNs are configured and functioning properly.
- **Troubleshooting**: Diagnose network connectivity issues by verifying that routes to particular destinations exist and identifying any incorrect routes.
- **Interface assignment**: Verify which network interfaces are managing specific routes to ensure proper traffic management.

These checks help maintain network efficiency and facilitate troubleshooting.

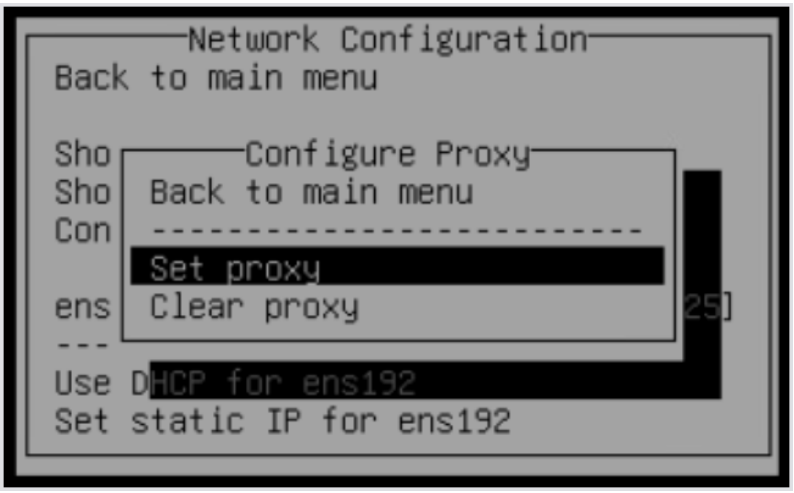


Configure proxy

This option allows you to manage proxy settings for your network configuration.

- **Set proxy:** Configure the proxy settings to direct network traffic through a specified proxy server.
- **Clear proxy:** Remove any existing proxy settings to revert to a direct connection.

Managing these settings can help optimize network performance and enhance security.

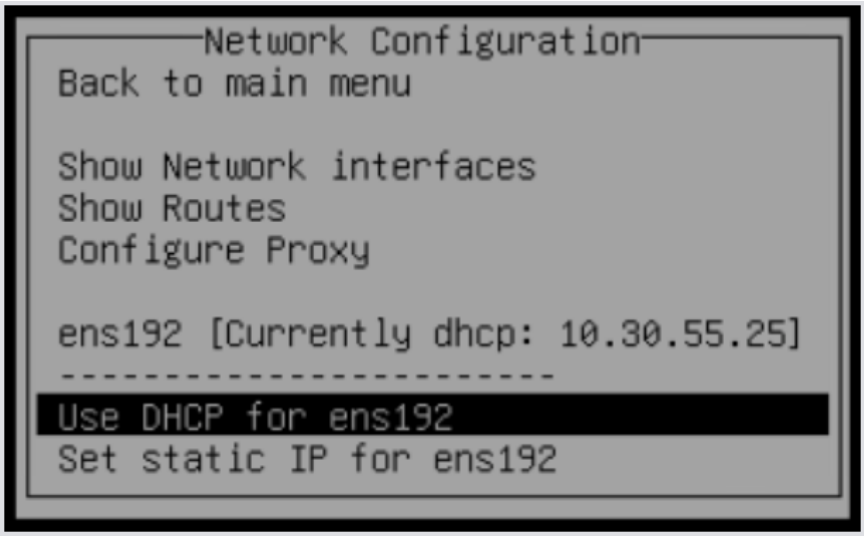


Use DHCP

Select this option to automatically obtain an IP address and network settings from a DHCP server. This simplifies network configuration and allows for dynamic address allocation. For more information, see Configuring the Virtual Appliance for the first time.

Set static IP

Select this option to manually configure a fixed IP address for the Virtual Appliance. This is useful for ensuring consistent network identification, especially for servers or devices that require a stable connection.



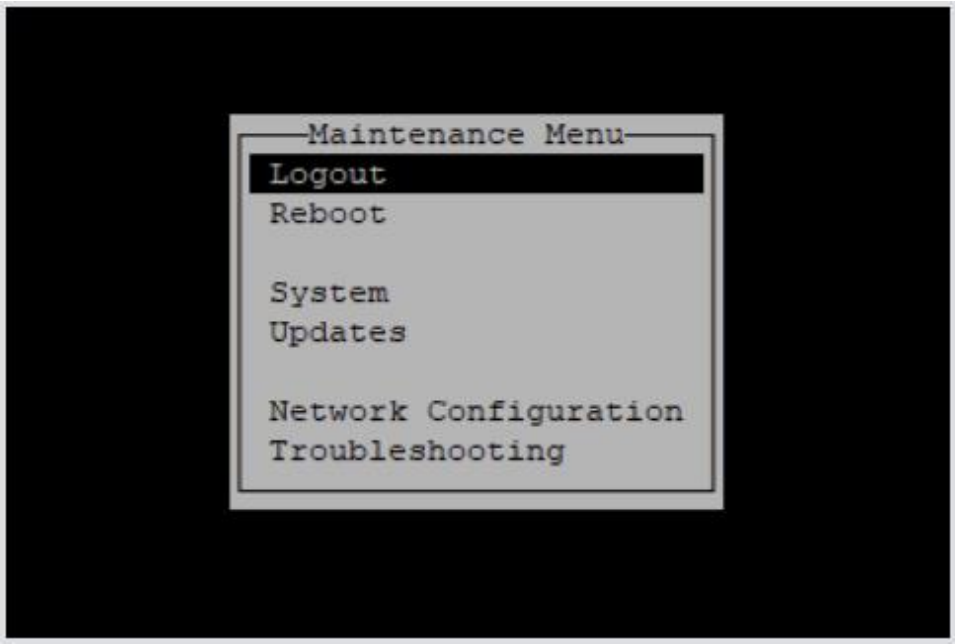
The steps below will help you change the IP address of your Login Enterprise Virtual Appliance and update the allowed URLs accordingly.

Changing the IP address may be necessary for various reasons, such as reconfiguring network settings, migrating to a different network, or resolving IP conflicts.

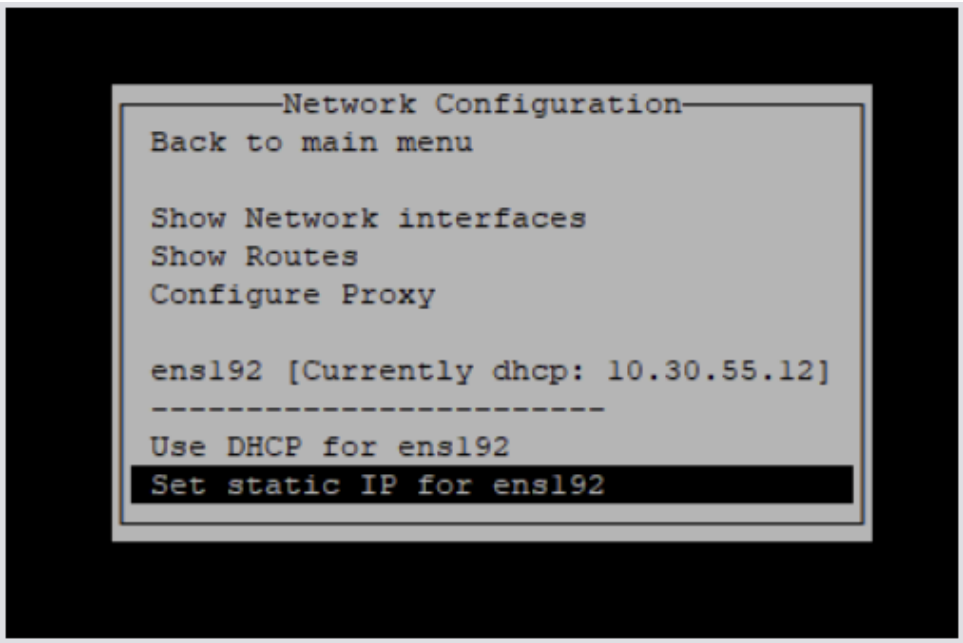
NOTE: After changing the IP address, you must manually update the ALLOWED_URLS list to include the new IP address.

Changing the IP address

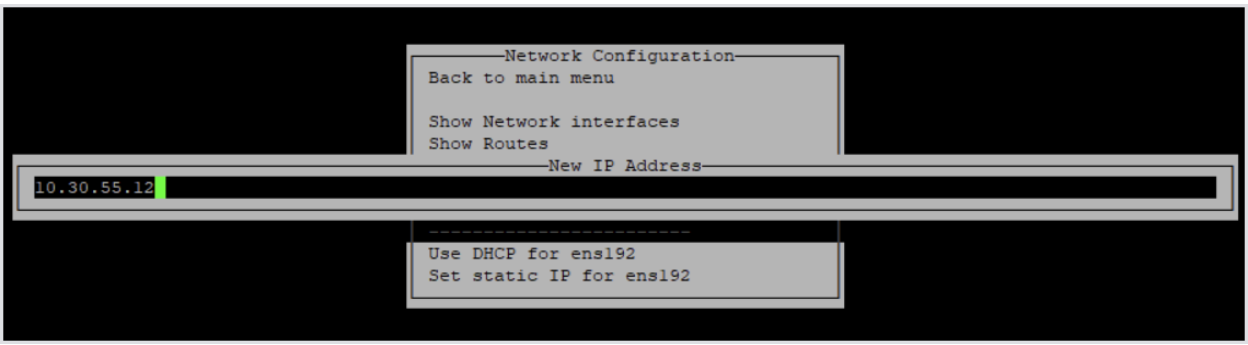
1. **Access the Maintenance Menu**: Console or SSH into the Appliance to access the Maintenance Menu.

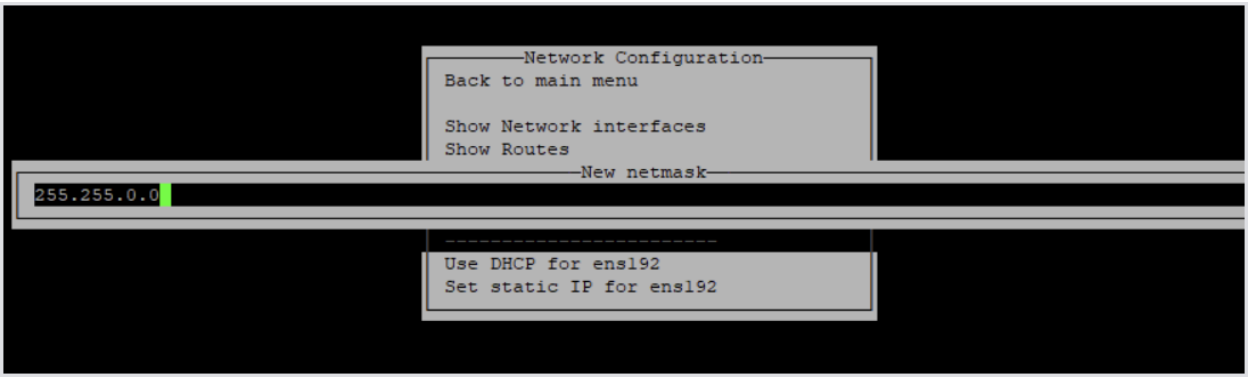


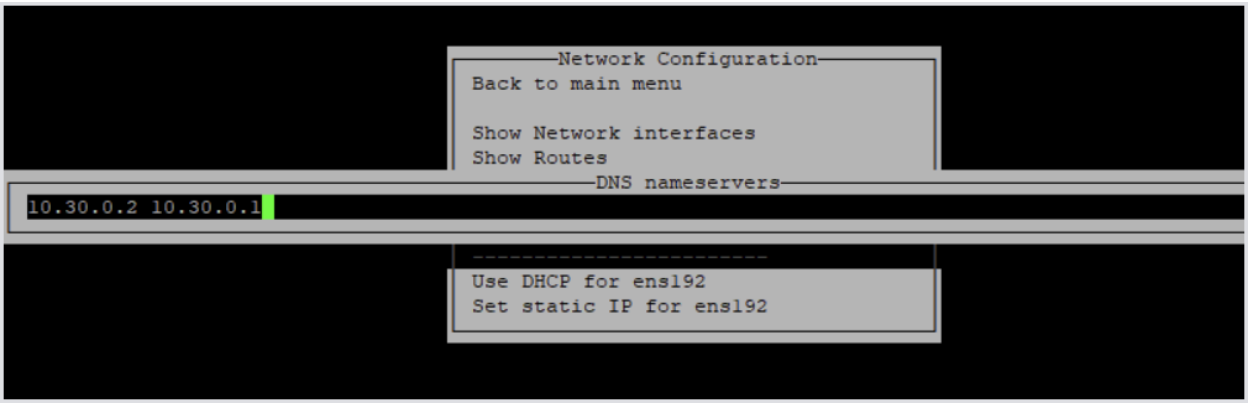
2. **Navigate to Network Configuration**: Select **Network Configuration**, then choose the **Set Static IP** option.

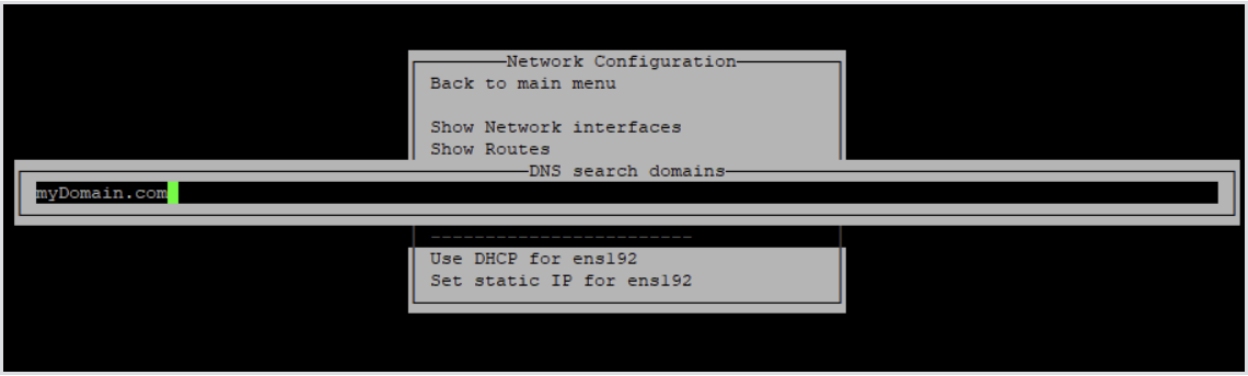


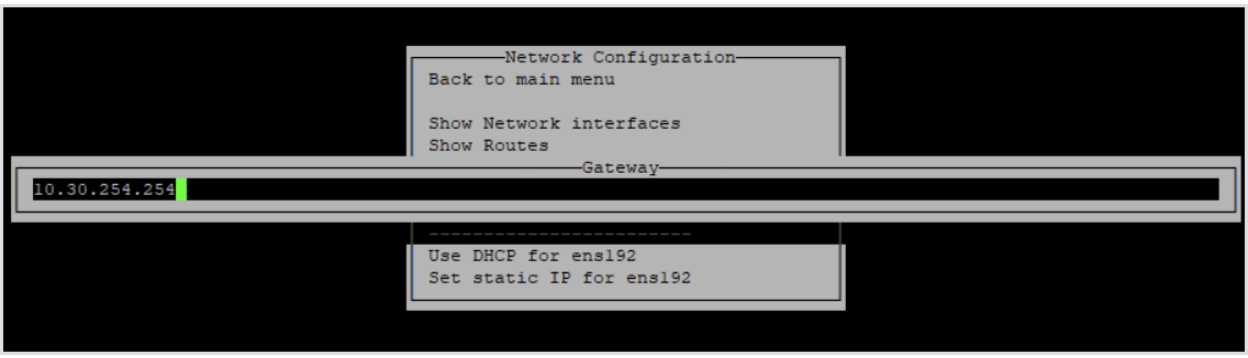
3. **Enter networking details**: Follow the prompts to enter the networking details. A new field will appear each time you press **Enter**.


Network Configuration
Back to main menu
Show Network interfaces
Show Routes
New IP Address
10.30.55.12
Use DHCP for ens192
Set static IP for ens192

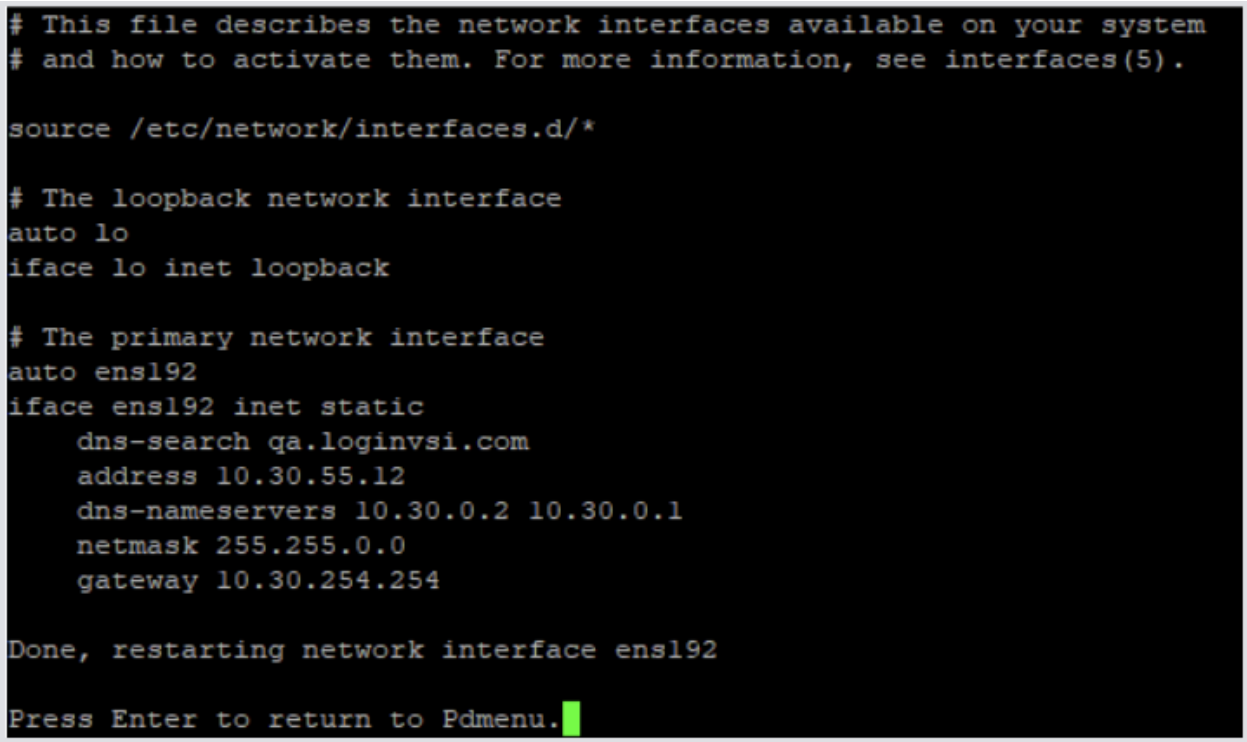

Network Configuration
Back to main menu
Show Network interfaces
Show Routes
New netmask
255.255.0.0
Use DHCP for ens192
Set static IP for ens192


Network Configuration
Back to main menu
Show Network interfaces
Show Routes
DNS nameservers
10.30.0.2 10.30.0.1
Use DHCP for ens192
Set static IP for ens192


Network Configuration
Back to main menu
Show Network interfaces
Show Routes
DNS search domains
myDomain.com
Use DHCP for ens192
Set static IP for ens192



4. **Completion confirmation**: Once you've entered all necessary information, the screen will display a confirmation message.



Updating the Allowed URLs

For information on updating the allowed URLs, see Changing the Allowed URLs.

Troubleshooting

This submenu provides options to help diagnose and resolve issues. The options are as follows:

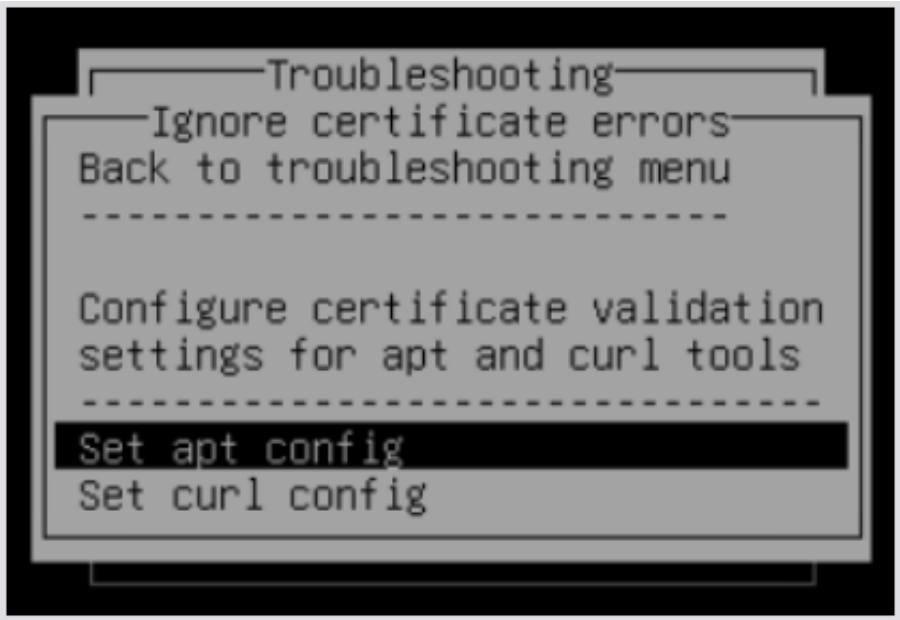
- Ignore Certificate Errors
- Start Enable SSH Server
- Stop and disable SSH server

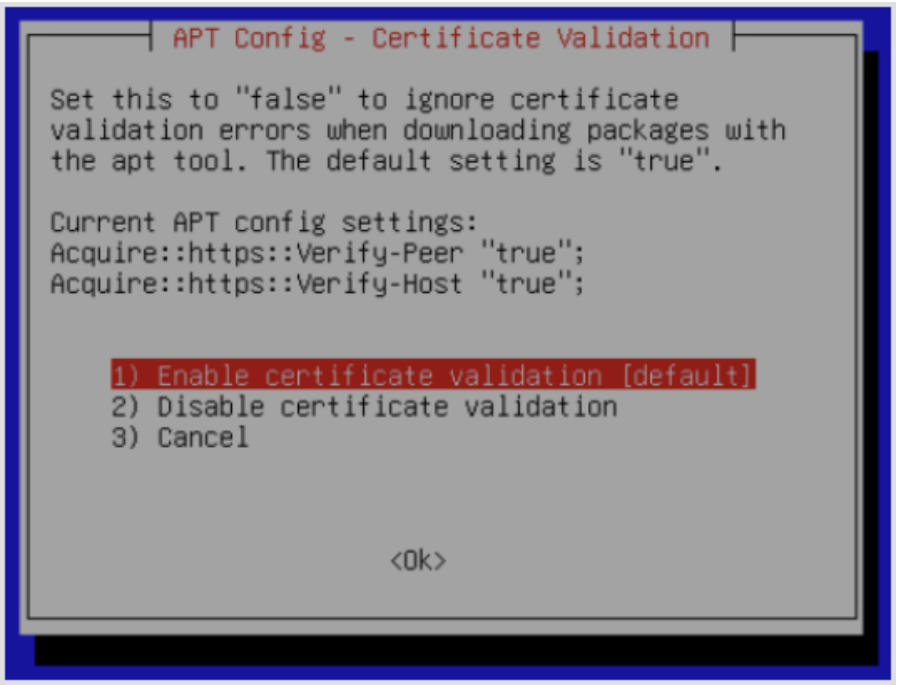
- Ping
- Show netstat
- Show LoginVSI service status
- Restart LoginVSI services
- Open bash shell

Ignore certificate errors

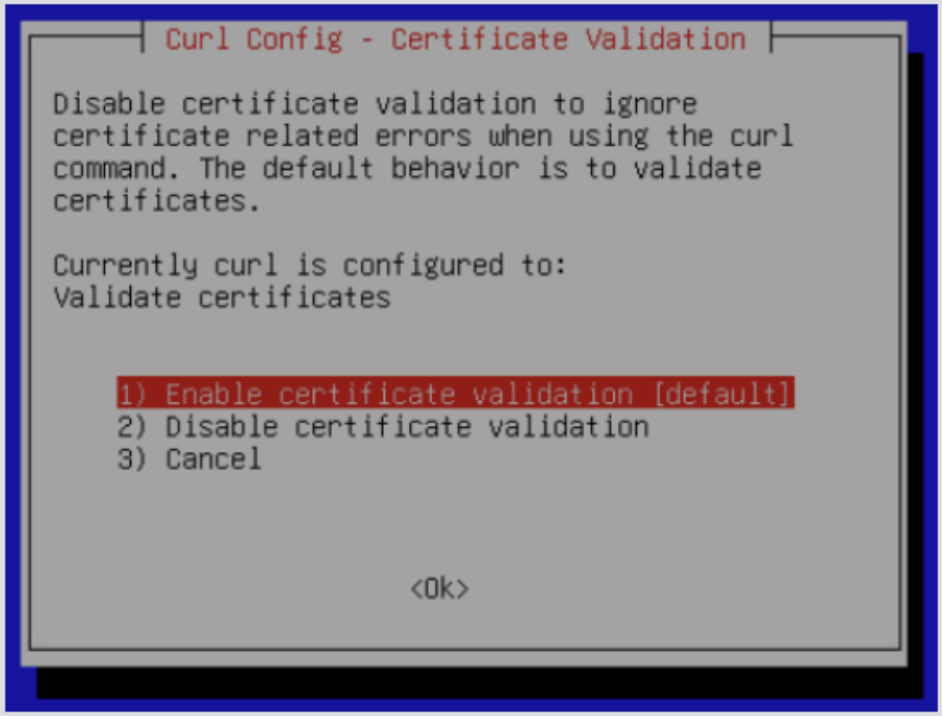
This option allows you to configure certificate validation settings for `apt` and `curl`. By adjusting these settings, you can choose to ignore certificate errors, which may be useful in certain troubleshooting scenarios or when working with self-signed certificates.

- **APT config menu**:
  - **Acquire::https::Verify-Peer** and **Verify-Host** are set to "true" by default, meaning `apt` validates certificates when downloading packages over HTTPS.
  - **Disable certificate validation** turns both settings to "false," allowing downloads without SSL checks. Useful for trusted internal repos or testing environments.





- **cURL config menu:**
  - **Validate certificates** is the default for `curl`, ensuring SSL validation for HTTPS requests.
  - **Disable certificate validation** allows `curl` to skip SSL checks, equivalent to using `--insecure`. This can be used in testing or for trusted servers.



IMPORTANT: Both options reduce security and should be used cautiously in production environments, as ignoring certificate errors can expose your system to potential risks.

Start enable SSH server

Select this option to start the SSH service, allowing remote SSH access to the appliance. Highlight this option and press **Enter** on your keyboard to run it.

Stop and disable SSH server

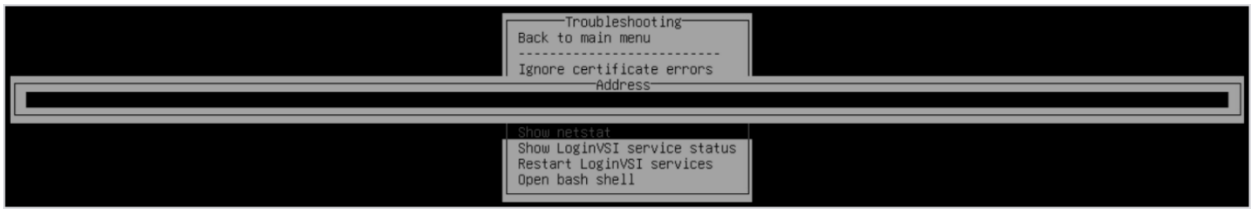
Select this option to stop the SSH service and disable remote access, preventing further SSH connections. Highlight this option and press **Enter** on your keyboard to execute it.

NOTE: Use the appropriate option based on your security needs. Enabling SSH allows remote access for management, while disabling SSH enhances security by blocking such access.

Ping

The Ping option allows you to test network connectivity by sending ICMP echo requests to a specified address or hostname. When selected, a field will appear where you can enter the desired IP address or hostname to check if the destination is reachable.

This tool helps you quickly verify network status and diagnose connection issues.




```
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=118 time=1.41 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=118 time=1.33 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=118 time=1.43 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=118 time=1.39 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=118 time=1.30 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=118 time=1.37 ms
64 bytes from 8.8.8.8: icmp_seq=7 ttl=118 time=1.52 ms
64 bytes from 8.8.8.8: icmp_seq=8 ttl=118 time=1.29 ms
64 bytes from 8.8.8.8: icmp_seq=9 ttl=118 time=1.44 ms
64 bytes from 8.8.8.8: icmp_seq=10 ttl=118 time=1.45 ms

--- 8.8.8.8 ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9014ms
rtt min/avg/max/mdev = 1.289/1.392/1.516/0.066 ms

Press Enter to return to Pdmenu.
```

Show netstat

The Show Netstat option provides an overview of networking details without requiring any configuration. This section is designed for users who know what they're looking for in terms of network diagnostics.

- **View network connections:** Quickly see active network connections and their status.
- **Interface statistics:** Check statistics for each network interface to monitor performance.
- **Routing information:** Access routing details to understand how traffic is managed.
- **Troubleshooting:** Identify potential issues and unauthorized connections, helping to resolve connectivity problems efficiently.

This option allows for easy monitoring of network performance without the need to manually enter netstat commands.

```
Show netstat
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN
tcp6       0      0 :::443                  :::*                    LISTEN
tcp6       0      0 :::80                   :::*                    LISTEN
tcp6       0      0 :::22                   :::*                    LISTEN
tcp6       0      0 :::8080                 :::*                    LISTEN
tcp6       0      0 :::7946                 :::*                    LISTEN
tcp6       0      0 :::2377                 :::*                    LISTEN
tcp6       0      0 127.0.0.1:43242         127.0.0.1:443           CLOSE_WAIT
udp        0      0 0.0.0.0:68              0.0.0.0:*
udp        0      0 0.0.0.0:4789            0.0.0.0:*
udp6       0      0 :::7946                 :::*
```

Show LoginVSI service status

This option displays the Docker containerized microservices hosted by the Virtual Appliance. This section functions similarly to viewing the "Login Enterprise Services" page in the Login Enterprise user interface.

- **View microservices:** See a list of all active microservices running in Docker containers.
- **Service status:** Check the current status of each microservice.

NOTE: This option is for viewing purposes only; no configuration changes can be made here.

```
Show LoginVSI service status
ID             NAME                          MODE         REPLICAS   IMAGE                                          PORTS
v5birrzwb8ke   VSI_contentdelivery           replicated   1/1        loginvsi/contentdelivery:3.1.121
rjn37hbtgwxv   VSI_contentdeliveryDb         replicated   1/1        loginvsi/postgres:13.3-le
foiz7smj9oyx   VSI_environments              replicated   1/1        loginvsi/environments:4.8.43
j12tuu7nt83v   VSI_environmentsDb            replicated   1/1        loginvsi/postgres:13.3-le
ow75cugl4alm   VSI_eventnotificationsworker  replicated   1/1        loginvsi/eventnotificationsworker:3.1.69
tt1hs4khpoed   VSI_eventshub                 replicated   1/1        loginvsi/eventshub:3.1.44
oobq2rm36g86   VSI_functionaleventsworker    replicated   1/1        loginvsi/functionaleventsworker:3.1.61
i1i3xv5r5q0i   VSI_grafana                   replicated   1/1        loginvsi/grafana:7.5.17.7-le
t3pjwjvfhaol   VSI_identityserver            replicated   1/1        loginvsi/identityserver:3.1.64
z0wl04pc1wdw   VSI_identityserverDb          replicated   1/1        loginvsi/postgres:13.3-le
kb5uzs6kq5l6   VSI_influx                    replicated   1/1        loginvsi/influxdb:2.7.6-le
y6jtel39ktq0   VSI_ingress                   replicated   1/1        loginvsi/ingress:3.1.35                        *:80->2080/tcp, *:443->2443/tcp
y5wats578tjr   VSI_loadtests                 replicated   1/1        loginvsi/loadtests:3.1.42
j7c1z0djtgwt   VSI_loadtestsDb               replicated   1/1        loginvsi/postgres:13.3-le
ycq29l3myz6o   VSI_loginenterprise           replicated   1/1        loginvsi/loginenterprise:4.8.78
dp3m0pkav7mg   VSI_loginenterpriseDb         replicated   1/1        loginvsi/postgres:13.3-le
r6mwx1ymd0x7   VSI_loginscheduleworker       replicated   1/1        loginvsi/loginscheduleworker:3.1.78
h2uafw5exbc4   VSI_messagebroker             replicated   1/1        loginvsi/rabbitmq:3.8.2-le
mzyuw77uk41g   VSI_portainer                 replicated   1/1        loginvsi/portainer:1.23.1
x8eu82fr7nbc   VSI_publicapi                 replicated   1/1        loginvsi/publicapi:3.1.104
2vmzmshypfcj   VSI_rawdataexports            replicated   1/1        loginvsi/rawdataexports:3.1.42
oeeqpqzlgr0w   VSI_rawdataexportsDb          replicated   1/1        loginvsi/postgres:13.3-le
qy56hmwqmkkk   VSI_rawdataexportsworker      replicated   1/1        loginvsi/rawdataexportsworker:3.1.55
0egmvpbnns3t   VSI_reports                   replicated   1/1        loginvsi/reports:3.1.62
pdbz02r8evxj   VSI_reportsDb                 replicated   1/1        loginvsi/postgres:13.3-le
dnpw23956eku   VSI_reportsworker             replicated   1/1        loginvsi/reportsworker:3.1.55
mc1mnr0s47em   VSI_sessionmetricstatistics   replicated   1/1        loginvsi/sessionmetricstatistics:3.1.45
qxqh188r4p82   VSI_sessionmetricstatisticsDb replicated   1/1        loginvsi/postgres:13.3-le
hu1fo98sz7vx   VSI_sessionworker             replicated   1/1        loginvsi/sessionworker:3.1.66
lz2vqdgcslwa   VSI_statisticworker           replicated   1/1        loginvsi/statisticworker:3.1.52
z7ecvsyn73jp   VSI_synchronizationworker     replicated   1/1        loginvsi/synchronizationworker:3.1.44
kkp1867ey5g6   VSI_systemconfiguration       replicated   1/1        loginvsi/systemconfiguration:4.8.32
c4dh2ikkzj4t   VSI_systemconfigurationDb     replicated   1/1        loginvsi/postgres:13.3-le
k666qunvoobw   VSI_systemworker              replicated   1/1        loginvsi/systemworker:3.1.36
um1113nk59i4   VSI_technicalevents           replicated   1/1        loginvsi/technicalevents:3.1.38
j9pctdjvwctd   VSI_technicaleventsDb         replicated   1/1        loginvsi/postgres:13.3-le
pym6elnj458e   VSI_ui                        replicated   1/1        loginvsi/ui_loginpi:3.1.296
vvq3ke5cx5nq   VSI_userinterfaceapi          replicated   1/1        loginvsi/userinterfaceapi:3.1.67
y1zjl5xpok62   VSI_usersessionmetricsDb      replicated   1/1        loginvsi/postgres:13.3-le
```

Restart LoginVSI services

The Restart Services option allows you to restart services running on the Virtual Appliance. Reasons to restart services:

- **Recover from service crashes:** Restore functionality after a service failure.
- **Apply configuration changes:** Implement changes made to service settings.
- **Troubleshoot system issues:** Address problems by refreshing services.
- **Improve performance:** Enhance overall performance by restarting services.

This option provides a quick way to restore functionality with minimal downtime.

```
Are you sure you want to restart the Login Enterprise services? [y/N]: y
Stopping Login Enterprise services...[DONE]
Login Enterprise started

Press Enter to return to Pdmenu.
```

Open bash shell

The Open Bash Shell option provides access to a command prompt, allowing you to type commands directly. This interface is minimal and is primarily intended for users who know in advance what tasks they want to perform.

- **Direct command-line access:** Gain access for advanced troubleshooting, manual configurations, service management, or running scripts.
- **Flexibility:** Execute Linux commands as needed.

TIP: It may not be common for a Virtual Appliance administrator to perform tasks in this shell, so familiarity with command-line operations is required.

```
root@RC-app-pool-7:/home/admin#
```

To exit the shell, type `exit` and press the **Enter** key.

Getting Started with the Web Interface

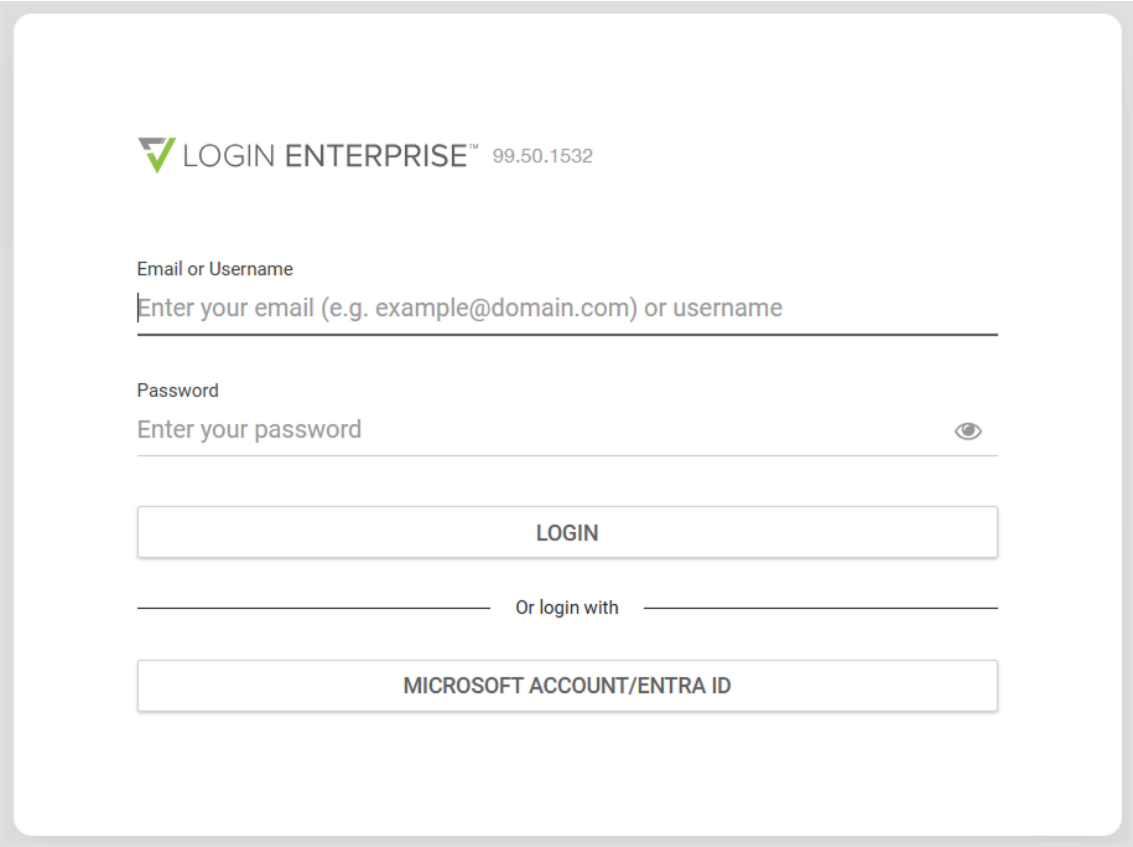
- Logging in
- Accepting the End User License Agreement (EULA)
- Viewing Release Notes
- Introduction and Guided Tour
- Updating your License

Logging in

Login Enterprise's web-based user interface is compatible with Google Chrome, Microsoft Edge, and Mozilla Firefox browsers, but not with Internet Explorer.

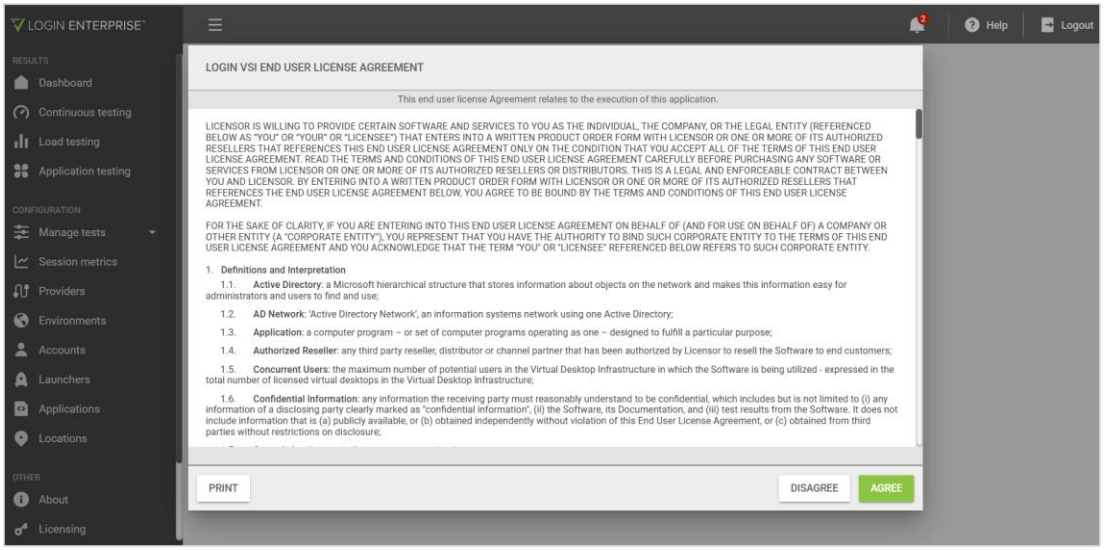
When you access the Virtual Appliance web interface, the login page appears.
Enter your login credentials based on your account type:

- **LDAP users** must log in using the **User Principal Name (UPN)** format, typically an email address such as username@domain.
    - As of v6.0, the previously accepted domain\username format (UNC) is no longer supported for LDAP users.
- **Appliance admin users** should continue using their regular username (not in email format).



Accepting the End User License Agreement (EULA)

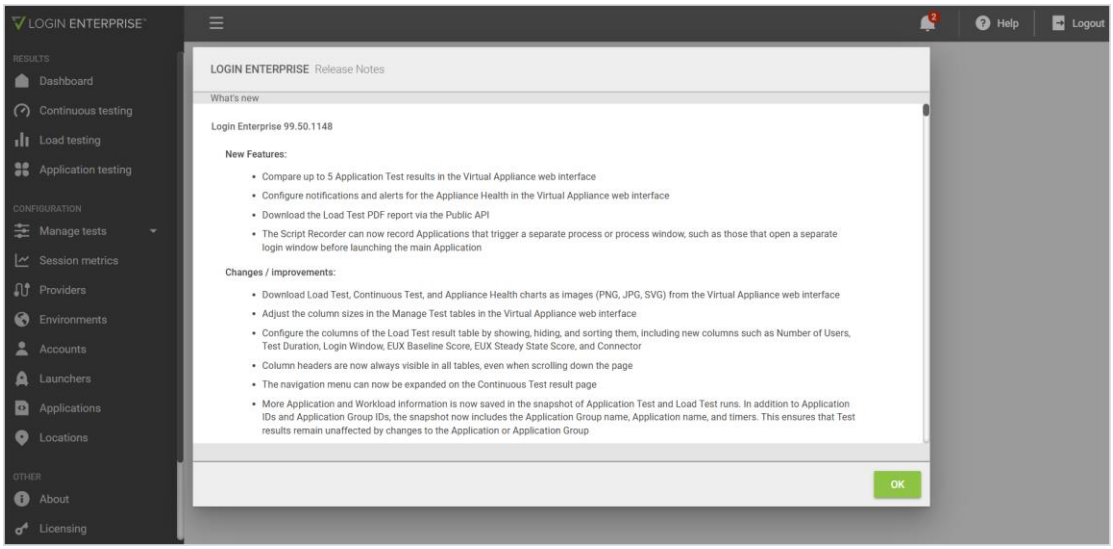
After you log in, the End User License Agreement (EULA) will appear.



When you accept the EULA, the popup will no longer display.

Viewing Release Notes

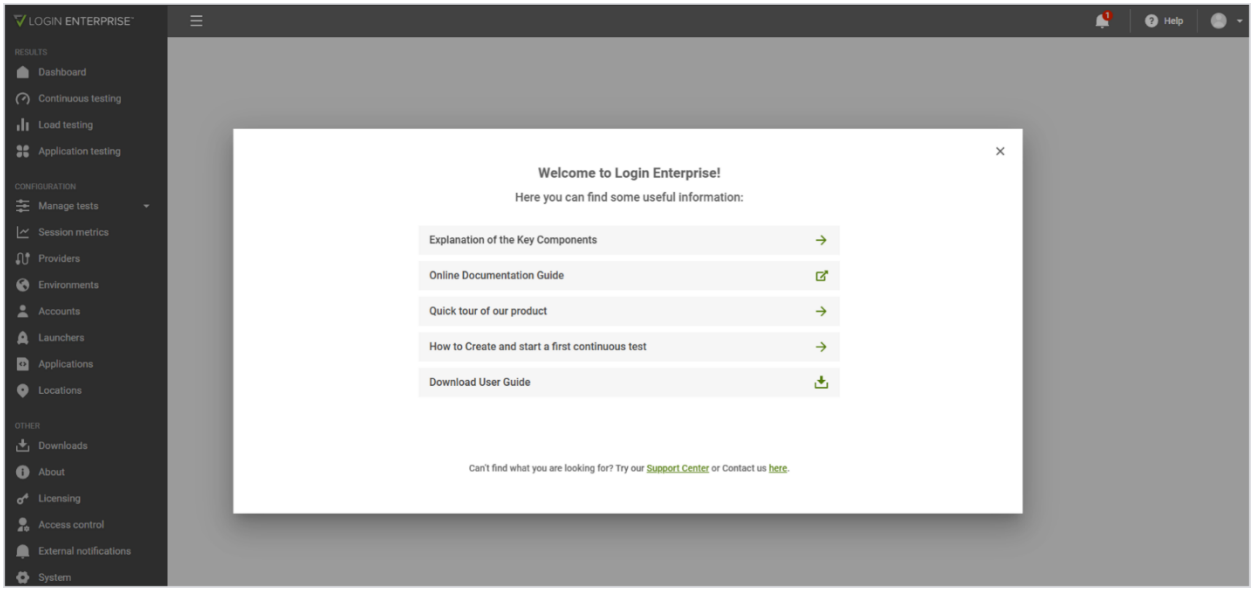
When you update your Appliance, release notes will appear the next time you access the Web UI.



Select **OK** to dismiss the release notes until the next upgrade.

Introduction and Guided Tour

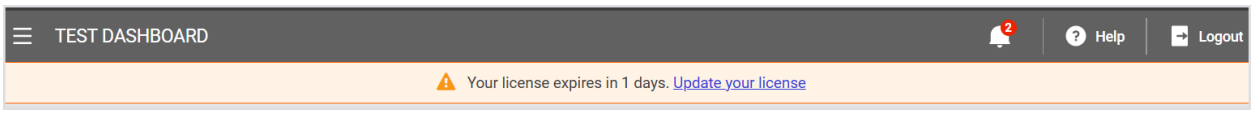
When you log in for the first time, the web interface will show an introduction and guided tour.



Use the guided tour to learn about the Login Enterprise key components and architecture, explore the online documentation, take a quick tour of the product, find out how to create and start your first test, for example, a Continuous Test, and download the user guide.

Updating your License

Once you've taken the guided tour, the system will show a notification asking you to update your license.



For more information on Licensing and how to update your License, see Managing Licenses.

Configuring Database Data Retention

- Overview
- Configuring data retention
- Additional resources

Overview

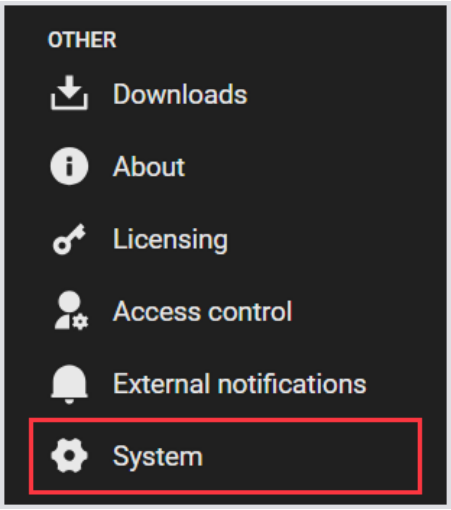
To make sure the database does not grow too rapidly, you can configure data retention for your database. The data retention functionality will effectively delete the raw data older than a specified number of days. This does not mean all graphs will become empty as they use statistical and not raw data, but certain views, for example, an Hourly view, will disappear when falling outside of the retention window.

IMPORTANT:

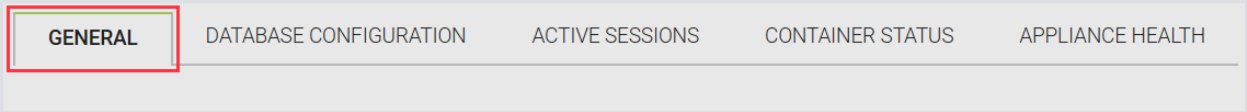
- When configured, it will eliminate the possibility of viewing the Hourly view of data older than the configured number of days.
- When configured, it will eliminate the possibility of viewing the events of data older than the configured number of days.
- A minimum of 45 days of data retention is enforced.

Configuring data retention

In the **Login Enterprise sidebar menu**, navigate to **Other > System**.

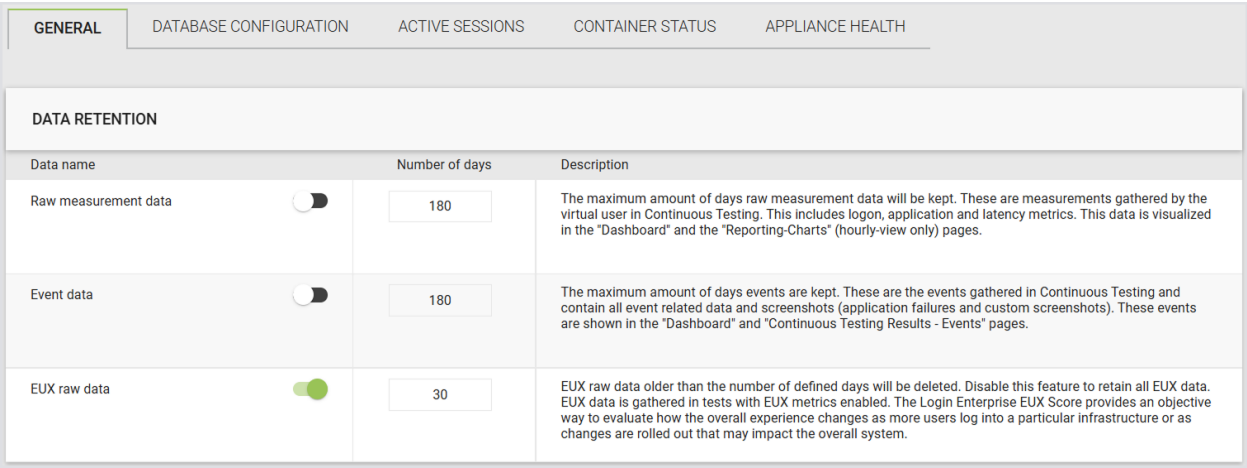


Ensure you're in the **General** section of the tab menu:



Data retention is disabled by default. To enable it, switch the toggle next to the item you wish to enable and configure the preferred option. There are 3 options you can configure:

Data name	Definition
Raw measurement data	When enabled, it deletes the Hourly view of data older than the configured number of days. However, longer statistical views, such as daily or weekly averages, are preserved. For example, if the retention period is set to 30 days, you can access minute-by-minute EUX scores for the past month, and after 30 days, only daily averages remain available. Weekly averages are also retained if configured.
Event data	When enabled, it deletes the events of data older than the configured number of days.
EUX raw data	When enabled, it deletes the EUX raw data older than the configured number of days. **NOTE:** This process only removes the underlying raw data and does not affect the EUX summary results.



Additional resources

- For information on configuring an external SQL database, see External SQL database.
- To learn more about the EUX score, see EUX score and VSImax.

Configuring External SQL Database

- Overview
- Why use an external SQL database?
- External SQL database requirements
- Configuring external SQL database

Overview

By default, Login Enterprise is delivered using an internal database. However, you can add an external SQL database.

IMPORTANT: Before you choose to migrate to an external SQL database in Login Enterprise, there are several things you need to consider:

- Migrating to an external SQL database is a one-way process. If you choose to go back to the internal database, your data will not be transferred back into the internal database and you will have data loss.
- Migrating to an external SQL database will result in deleting all existing data collected up to that point.

If you choose to migrate to an external SQL database, please create a snapshot to eliminate the possibility of losing data permanently.

Why use an external SQL database?

Benefit	Explanation
Improved scalability	Enables Login Enterprise to handle increased data volumes and user loads more efficiently.
Centralized management	Allows database administrators to monitor and optimize the database independently for better control and maintenance.
High availability and redundancy	Provides built-in features that ensure the system remains operational and resilient to failures.
Enhanced security	Offers advanced security measures to better protect data and maintain integrity.
Better performance	Improves overall system efficiency compared to internal database setups.
Integration flexibility	Supports seamless integration with various platforms and applications.
Easier compliance	Simplifies meeting regulatory and compliance requirements.

Long-term efficiency	Contributes to sustainable performance and reliability as Login Enterprise grows.

External SQL database requirements

For the successful external SQL database implementation in the Login Enterprise user interface, there are several requirements:

1. Login Enterprise is compatible with Microsoft SQL Server 2016, 2017, 2019, and 2022.
2. A database must exist on the SQL Server instance.
3. The SQL database connection requires an SQL user, not a Domain User.
4. Ensure that Named Pipes are enabled for the SQL Server service
5. The connection can be configured to utilize SSL when communicating with the SQL Server.

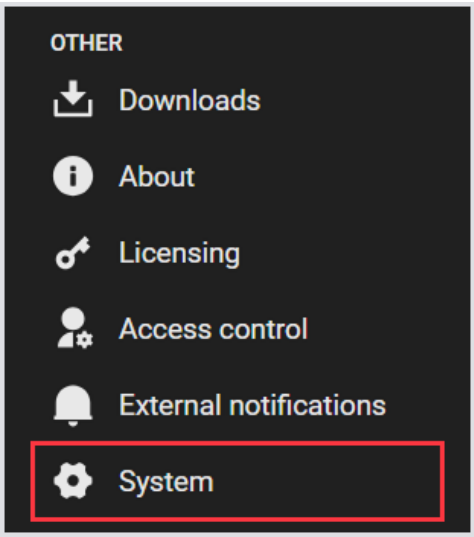
The minimum requirement for external SQL is the same as the default size of the appliance, or the minimum requirement for the version of the SQL you choose, whichever is greater.

Appliance default:

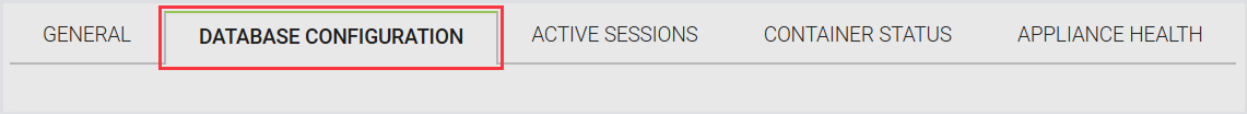
Configuration	CPU	Memory	Disk
Recommended	4(v)CPU	8GB RAM	100GB

Configuring external SQL database

1. In the **Login Enterprise idebar menu**, navigate to **Other > System**.

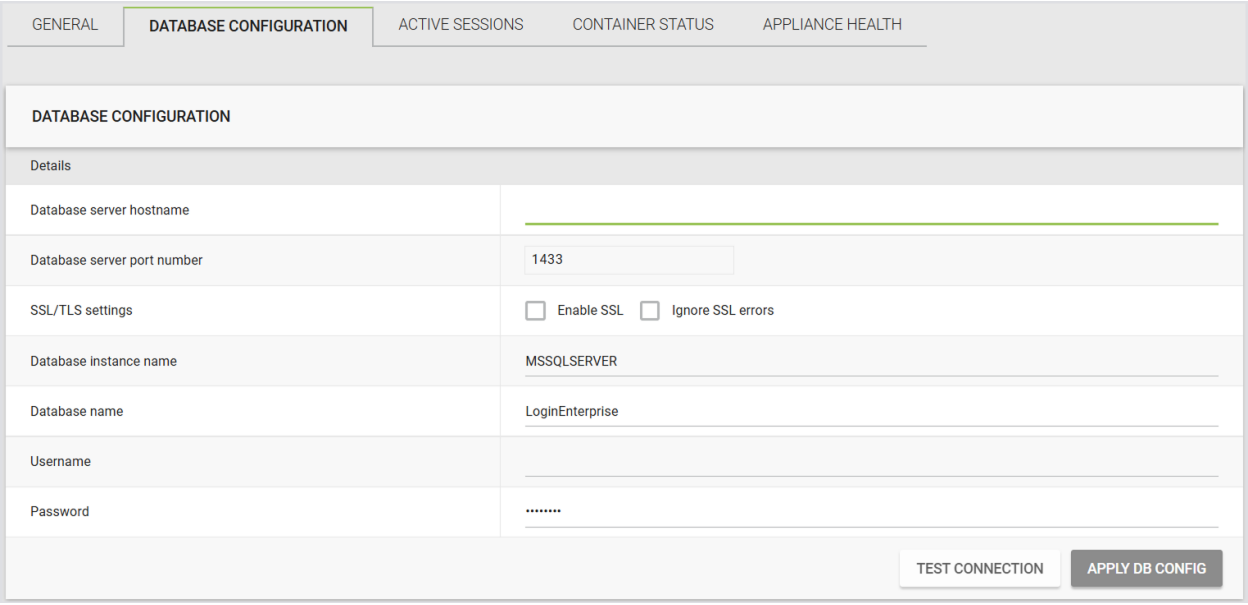


2. In **System**, select **Database configuration** from the tab menu.



3. In the **Database configuration** window, fill out the following details:

Field	Value
Database server hostname	Hostname / IP / FQDN of SQL server
Database server port number	The port number used to communicate to the SQL server
SSL/TLS settings	• Select **Enable SSL** to enhance the security of your connection by encrypting the data exchanged between the appliance and the SQL server. This is the recommended choice to protect sensitive information during transmission. • Select **Ignore SSL** errors to allow the connection to proceed even if there are SSL-related issues.
Database instance name	The name of the preferred instance for the database configuration
Database name	The name of the created database
Username	The username of the preferred SQL user
Password	The password of the SQL user



5. Click **Test connection** to ensure the database is correctly configured.

6. Click **Apply DB config** to save the database configurations.

Congratulations, your external SQL database has now been configured.

Expanding Appliance Disk Space

- Using the script
  - Prerequisites
  - Executing the script
- Manually (Only if the script fails)
  - Preparation
  - Moving the swap space to the end of the new partition
  - Expanding the primary disk into its new space
  - Finishing up

Using the script

The recommended method for expanding the Appliance disk space is to use the provided script, which automates the process. You can download the script from here.

This script is designed to quickly and easily expand the disk size on your Virtual Appliance, helping to prevent potential crashes due to a full disk. The script file is included in all new versions of the Appliance (starting with version 5.13), located in the `/loginvsi/bin` folder, and is named `expand-disk.sh`

NOTE: If you're running an earlier version of Login Enterprise (before version 5.13), the script will still work to expand disk space, even though it was not included in that version.

The manual method is more complex and should **only** be used as a last resort if the script fails.

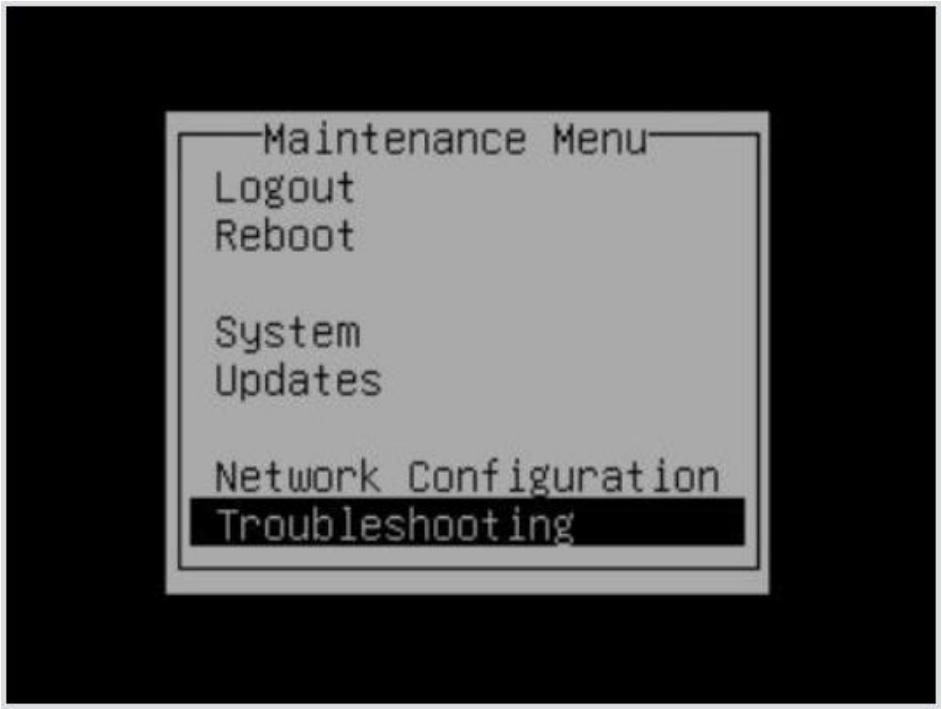
Prerequisites

This script expands the Appliance's disk using space that has already been added. Before executing the script, ensure that you have added the required disk space to the Appliance in your virtualization environment.

Executing the script

In the example provided, the Appliance has 100 GB of space, with an additional 100 GB added. The values shown in the screenshots correspond to these amounts. Be sure to refer to your own values based on the amount of space you add.

1. Log in to your Virtual Appliance console and open the bash shell by selecting **Troubleshooting > Open Bash Shell**.



2. The disk expansion script is named `expand-disk.sh` and is located in the `/loginvsi/bin` folder. Move to the aforementioned folder by running the cd `/loginvsi/bin` command.

3. Run the script with the `./expand-disk.sh` command. If the script can't find the added unpartitioned space, or if the added space is less than 1 GB, it will prompt you to reboot.

```
root@login-ent-regr511:/loginvsi/bin# ./expand-disk.sh
*** Available unused space on disk
Unpartitioned space /dev/sda: 0 B, 0 bytes, 0 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes

There is only 0 bytes of free space. Please reboot to detect the disk expansion.
root@login-ent-regr511:/loginvsi/bin#
```

4. After rebooting, run `./expand-disk.sh` again (remember to relocate to `/loginvsi/bin/` first).
5. The script will detect the unpartitioned space, display the existing partition table, and prompt you to confirm the expansion (in this example, by 100 GB). Press `Y` to proceed.

```
root@login-ent-regr511:/loginvsi/bin# ./expand-disk.sh
*** Available unused space on disk
Unpartitioned space /dev/sda: 100 GiB, 107375230976 bytes, 209717248 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes

    Start       End   Sectors  Size
209713152 419430399 209717248  100G

*** Existing partition table
Disk /dev/sda: 200 GiB, 214748364800 bytes, 419430400 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x2e628d0b

Device     Boot     Start       End   Sectors Size Id Type
/dev/sda1  *         2048 201328639 201326592  96G 83 Linux
/dev/sda2       201330686 209713151   8382466   4G  5 Extended
/dev/sda5       201330688 209713151   8382464   4G 82 Linux swap / Solaris

*** Existing /dev/sda1 partition is 96G, filesystem is 94G
*** This script will expand the partition and filesystem by 100GB
Are you sure you want to proceed? (y/N)_
```

6. The script will indicate the partition increased by 100 GB (you can see the difference compared to the previous message) and will prompt you to reboot. After rebooting, run `./expand-disk.sh continue` to continue.

```
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x2e628d0b

Device     Boot     Start       End   Sectors Size Id Type
/dev/sda1  *         2048 201328639 201326592  96G 83 Linux
/dev/sda2       201330686 209713151   8382466   4G  5 Extended
/dev/sda5       201330688 209713151   8382464   4G 82 Linux swap / Solaris

*** Existing /dev/sda1 partition is 96G, filesystem is 94G
*** This script will expand the partition and filesystem by 100GB
Are you sure you want to proceed? (y/N)y
*** Moving swap partition at /dev/sda2 out by 209715200 sectors.

*** Recreating swap partition /dev/sda5

*** Expanding partition /dev/sda1 into free space

*** /dev/sda1 partition is now 196G
*** Final partition table:
Disk /dev/sda: 200 GiB, 214748364800 bytes, 419430400 sectors
Disk model: Virtual disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x2e628d0b

Device     Boot     Start       End   Sectors  Size Id Type
/dev/sda1  *         2048 411047936 411045889  196G 83 Linux
/dev/sda2       411045886 419428351   8382466    4G  5 Extended
/dev/sda5       411047936 419428351   8380416    4G 82 Linux swap / Solaris

*** Please reboot and run the following command:
./expand-disk.sh continue
root@login-ent-regr511:/loginvsi/bin#
```

7. The final prompt will confirm that the partition and filesystem have been successfully increased.

```
root@login-ent-regr511:/loginvsi/bin# ./expand-disk.sh continue

*** Recreating swap file on /dev/sda5 with UUID=2c92b47e-2658-462b-814e-bef0f7824b38
Setting up swapspace version 1, size = 4 GiB (4290768896 bytes)
no label, UUID=2c92b47e-2658-462b-814e-bef0f7824b38

*** Enabling swap

*** /dev/sda1 partition is now 196G
*** Expanding filesystem on /dev/sda1
resize2fs 1.47.0 (5-Feb-2023)
The filesystem is already 51380224 (4k) blocks long.  Nothing to do!


*** /dev/sda1 filesystem is now 193G
root@login-ent-regr511:/loginvsi/bin#
```

Manually (Only if the script fails)

Reviewing disk utilization of the Login Enterprise Virtual Appliance is essential for maintaining performance and availability, especially as the number of Tests increases and the volume of performance data grows.

You can manage this growth using the data retention features of the Virtual Appliance for raw data and events. For more information, see Configuring Database for Data Retention. If disk utilization is not monitored closely, you may encounter issues during upgrades, which can lead to failures.

NOTE: These steps are intended for the default Debian Virtual Appliance implementation. Results may vary slightly with other Linux operating systems, such as Red Hat. This process typically takes 2-3 hours and does not require a reboot.

To add more disk space to the Virtual Appliance's primary partition, for example, /dev/sda1, follow these steps:

Preparation

1. **Backup your Virtual Appliance**: Take a snapshot of the Virtual Machine or the VM's hard disk using the hypervisor tools. This allows you to restore the Virtual Appliance to its previous configuration if any issues arise.
2. **Grow the Logical Disk**: Expand the logical disk that has run out of space. This process is generally straightforward in most hypervisors, including Azure.
3. **Access the Virtual Appliance Bash Shell**: Use the console or SSH tools like PuTTY to access the Bash shell.

NOTE: Some operations may disconnect your SSH session, so using the Login Enterprise Virtual Appliance Console is recommended.

4. **Check Disk Utilization**: Run the following command to assess disk utilization:

```
df -h
```

```
root@bpleva01:/home/admin# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.9G     0  7.9G   0% /dev
tmpfs           1.6G   12M  1.6G   1% /run
/dev/sda1        26G   23G  1.9G  93% /
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sdb1        32G   49M   30G   1% /mnt
```

5. **Check the Disk Partition Table**: Gather the necessary information for the next steps by running the following command:

```
sfdisk -l
```

A physical sector boundary occurs every 4096 bytes. If the partition does not start on a physical sector boundary, it may split I/Os across the boundary, potentially affecting disk performance. However, this should not be an issue for the virtual appliance swap space, as /dev/sda5 is aligned.

```
root@bpleva01:/home/admin# sfdisk -l
Disk /dev/sdb: 32 GiB, 34359738368 bytes, 67108864 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xcb05f06c

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 67106815 67104768  32G  7 HPFS/NTFS/exFAT


Disk /dev/sda: 128 GiB, 137438953472 bytes, 268435456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xc72a40d4

Device     Boot    Start      End  Sectors Size Id Type
/dev/sda1  *        2048 54527999 54525952  26G 83 Linux
/dev/sda2       54530046 62912511  8382466   4G  5 Extended
/dev/sda5       54530048 62912511  8382464   4G 82 Linux swap / Solaris

Partition 2 does not start on physical sector boundary.
root@bpleva01:/home/admin#
```

Moving the swap space to the end of the new partition

1. Calculate where the new starting sector for /dev/sda2 will be (/dev/sda5 is in /dev/sda2, so it will move when you move /dev/sda2). Use the following formula to determine the new starting sector:

(Total # sectors on disk /dev/sda) - (# Sectors in partition /dev/sda2) - (Starting Sector of /dev/sda2) = New Starting Sector

For illustration, let's consider a 128 GB disk as the new size (you can verify this by running sdlist -l, which will display the number of sectors for "Disk /dev/sda").

(268,435,456) - (8,382,466) - (54,530,048) = 205,522,944

The best practice is to align the starting sector to a physical sector boundary for optimal performance. To do this, divide the new starting sector by 4096, round down to the nearest integer, and then multiply by 4096. For example, the calculation would look like this:

(205,522,944 / 4,096) = 50,176.5, round down to 50,176 and multiply by 4096 equals ***205,520,896***
This will keep the swap partition /dev/sda5 aligned, while /dev/sda2 is not aligned

2. Before you make the move, you need to turn memory swapping off because your swap space is on /dev/sda5.

```
swapoff -a
```

3. Now, use sfdisk to move the /dev/sda2 partition to its new home:

```
echo '+205520896' | sfdisk /dev/sda -N 2 --move-data --no-reread
```

```
/home/admin# echo '+205520896' | sfdisk /dev/sda -N 2 --move-data --no-reread
Disk /dev/sda: 128 GiB, 137438953472 bytes, 268435456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xc72a40d4

Old situation:

Device     Boot    Start      End  Sectors Size Id Type
/dev/sda1  *        2048 54527999 54525952  26G 83 Linux
/dev/sda2       54530046 62912511  8382466   4G  5 Extended
/dev/sda5       54530048 62912511  8382464   4G 82 Linux swap / Solaris

Partition 2 does not start on physical sector boundary.

/dev/sda2:
New situation:

Device     Boot     Start       End  Sectors Size Id Type
/dev/sda1  *         2048  54527999 54525952  26G 83 Linux
/dev/sda2       260050942 268433407  8382466   4G  5 Extended
/dev/sda5        54530048  62912511  8382464   4G 82 Linux swap / Solaris

Partition 2 does not start on physical sector boundary.
Partition table entries are not in disk order.

Data move:
 typescript file: /root/sfdisk-sda2.move
 old start: 54530046, new start: 260050942 (move 8382466 sectors)

The partition table has been altered.
Calling ioctl() to re-read partition table.
Re-reading the partition table failed.: Device or resource busy
The kernel still uses the old table. The new table will be used at the next
reboot or after you run partprobe(8) or kpartx(8).
Syncing disks.
```

NOTE: The partition remapping process may take some time. In the example above, it typically took about 2-3 hours. During this period, monitor the disk activity to observe the machine remapping the partition.

To check the status of the remapping, you can run the following commands:

```
tail /root/sfdisk-sda2.move
```

and

```
cat /root/sfdisk-sda2.move | more
```

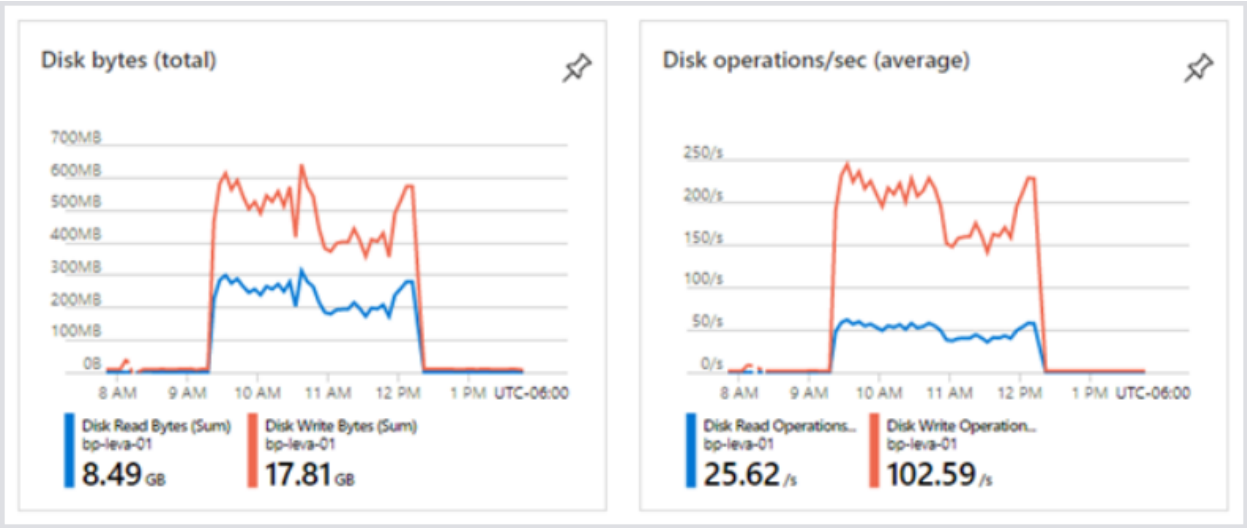
Once the move is complete, you will notice that /dev/sda5 has also been relocated.

Below is an example of what the partition layout looks like after the move is finished. Pay special attention to the new starting sectors for both /dev/sda2 and /dev/sda5:

```
root@bpleva01:/home/admin# sfdisk -l
Disk /dev/sdb: 32 GiB, 34359738368 bytes, 67108864 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xcb05f06c

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 67106815 67104768  32G  7 HPFS/NTFS/exFAT


Disk /dev/sda: 128 GiB, 137438953472 bytes, 268435456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xc72a40d4

Device     Boot     Start       End  Sectors Size Id Type
/dev/sda1  *         2048  54527999 54525952  26G 83 Linux
/dev/sda2       260050942 268433407  8382466   4G  5 Extended
/dev/sda5       260050944 268433407  8382464   4G 82 Linux swap / Solaris

Partition 2 does not start on physical sector boundary.
root@bpleva01:/home/admin#
```

During the 3-hour remapping process, you can monitor the disk I/O activity. Here's what the disk I/O might look like:

Expanding the primary disk into its new space



1. To utilize the newly allocated space, expand the /dev/sda1 partition:

```
echo ',+' | sfdisk /dev/sda -N 1 --no-reread
```

After expanding the /dev/sda1 partition, it should reflect the new size. The partition should now be 124 GB.

```
sfdisk -l
```

```
root@bpleva01:/home/admin# sfdisk -l
Disk /dev/sdb: 32 GiB, 34359738368 bytes, 67108864 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xcb05f06c

Device     Boot Start      End  Sectors Size Id Type
/dev/sdb1        2048 67106815 67104768  32G  7 HPFS/NTFS/exFAT


Disk /dev/sda: 128 GiB, 137438953472 bytes, 268435456 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xc72a40d4

Device     Boot     Start       End   Sectors  Size Id Type
/dev/sda1  *         2048 260050941 260048894  124G 83 Linux
/dev/sda2       260050942 268433407   8382466    4G  5 Extended
/dev/sda5       260050944 268433407   8382464    4G 82 Linux swap / Solaris

Partition 2 does not start on physical sector boundary.
```

2. At this point, you have successfully expanded your disk partition; however, the filesystem on the partition has not yet been expanded. To expand the filesystem capacity, you need to instruct the OS to rescan the partitions using the 'partprobe' command.

First, you'll need to install 'parted' with the following command, then run 'partprobe'.

```
apt-get install parted
partprobe
```

Now that you have rescanned the partitions, you can expand the filesystem on /dev/sda1 to utilize the new space. Run the following command to expand the filesystem:

```
root@bpleva01:/home/admin# apt-get install parted
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libparted2
Suggested packages:
  libparted-dev libparted-i18n parted-doc
The following NEW packages will be installed:
  libparted2 parted
0 upgraded, 2 newly installed, 0 to remove and 3 not upgraded.
Need to get 471 kB of archives.
After this operation, 801 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 https://snapshot.debian.org/archive/debian/20210414 stretch/main amd64 libparted2 amd64 3.2-17 [276 kB]
Get:2 https://snapshot.debian.org/archive/debian/20210414 stretch/main amd64 parted amd64 3.2-17 [194 kB]
Fetched 471 kB in 1s (290 kB/s)
Selecting previously unselected package libparted2:amd64.
(Reading database ... 42111 files and directories currently installed.)
Preparing to unpack .../libparted2_3.2-17_amd64.deb ...
Unpacking libparted2:amd64 (3.2-17) ...
Selecting previously unselected package parted.
Preparing to unpack .../parted_3.2-17_amd64.deb ...
Unpacking parted (3.2-17) ...
Setting up libparted2:amd64 (3.2-17) ...
Setting up parted (3.2-17) ...
Processing triggers for libc-bin (2.24-11+deb9u4) ...
Processing triggers for man-db (2.7.6.1-2) ...
root@bpleva01:/home/admin# partprobe
```

```
resize2fs /dev/sda1
```

After executing this command, your Login Enterprise Virtual Appliance primary disk will have 122 GB of usable disk space - /dev/sda1

```
root@bpleva01:/home/admin# resize2fs /dev/sda1
resize2fs 1.43.4 (31-Jan-2017)
Filesystem at /dev/sda1 is mounted on /; on-line resizing required
old_desc_blocks = 4, new_desc_blocks = 16
The filesystem on /dev/sda1 is now 32506111 (4k) blocks long.
```

```
root@bpleva01:/home/admin# df -h
Filesystem      Size  Used Avail Use% Mounted on
udev            7.9G     0  7.9G   0% /dev
tmpfs           1.6G   12M  1.6G   1% /run
/dev/sda1       122G   23G   95G  20% /
tmpfs           7.9G     0  7.9G   0% /dev/shm
tmpfs           5.0M     0  5.0M   0% /run/lock
tmpfs           7.9G     0  7.9G   0% /sys/fs/cgroup
/dev/sdb1        32G   49M   30G   1% /mnt
```

Finishing up

Turn Swapping back on

1. Once the filesystem expansion is complete, re-enable swapping by using the following command:

```
swapon -a
```

```
root@bpleva01:/home/admin# swapon -a
root@bpleva01:/home/admin# swapon
NAME      TYPE      SIZE USED PRIO
/dev/sda5 partition   4G   0B   -2
```

2. You have successfully expanded the Virtual Appliance disk and filesystem. The additional disk space is now available for use.

Performing Appliance Backup and Recovery

- Overview
- External database
- Internal database

Overview

You can migrate the database from a malfunctioning Login Enterprise instance to a functional one, preserving your data and configuration. Please note that encrypted data, such as user passwords and application secrets, may not be transferred, but your overall configuration will remain intact.

This section outlines the process for creating a backup of the internal database to facilitate restoration on a rebuilt Appliance. It also provides instructions on configuring a new Appliance to connect to an existing external database and recover configurations.

External database

If you still have access to the original Appliance, the best way to reconnect to your external database is to back up and restore your internal database. The database connection information is stored on the Appliance itself, so restoring the Appliance data will also restore access to the external database. If possible, use the Internal Database instructions below.

Otherwise, you can simply set up a new Appliance and configure it to connect to the existing database.

IMPORTANT: Avoid directing two Appliances to the same database simultaneously.

If you need to replace an Appliance, it's straightforward to create a new one and configure it to use the existing database. For additional information on cleaning up encrypted data, contact us.

Internal database

For the internal database, it's easiest if you can obtain a copy of the database before your Appliance VM becomes inaccessible. Otherwise, it may be possible to get the data files from the boot disk by attaching it to another VM.

IMPORTANT: You must ensure that all the Login Enterprise services are stopped before proceeding with the steps below.

1. Log in to the Appliance with the admin account, go to **Troubleshooting**, and open the Bash shell.
2. `loginvsid stop`
3. `systemctl stop docker`

4. `cd /home/admin`
5. `tar -zcvf data.tgz /loginvsi/data /var/lib/docker/swarm /loginvsi/.env /certificates /usr/local/share/ca-certificates /home/admin/LE_key`
6. `loginvsid start`
7. Download `/home/admin/data.tgz` to some other machine.

We use tar to ensure proper preservation of file permissions and ownership. This tar file can be restored to the original Appliance or a new Appliance if you want to clone it.

To restore the data to the same Appliance or install the database on a new Appliance, follow these steps:

1. Upload `data.tgz` to `/home/admin` on the new Appliance
2. `loginvsid stop`
3. `systemctl stop docker`
4. `cd /`
5. `rm -rf /loginvsi/data /var/lib/docker/swarm`
6. `tar -zxvf /home/admin/data.tgz`
7. `nano /loginvsi/.env`
8. Stop here and update the ALLOWED_CLIENT_URLS field in the `.env` file to reflect your current IP address and/or hostname. You will be unable to start the services if ALLOWED_CLIENT_URLS is wrong. Save and exit the editor. Then:
9. `update-ca-certificates`
10. `loginvsid restart`

After logging back into Login Enterprise, your data and configurations will be automatically restored.

NOTE: You can clone an Appliance in this way, too. If you are using the internal database, you can restore an Appliance to multiple VMs, and they will not conflict.

Launchers and Test Users need to be restricted to a single Appliance. Multiple Appliances cannot share Launchers or users.

Monitoring Active Sessions

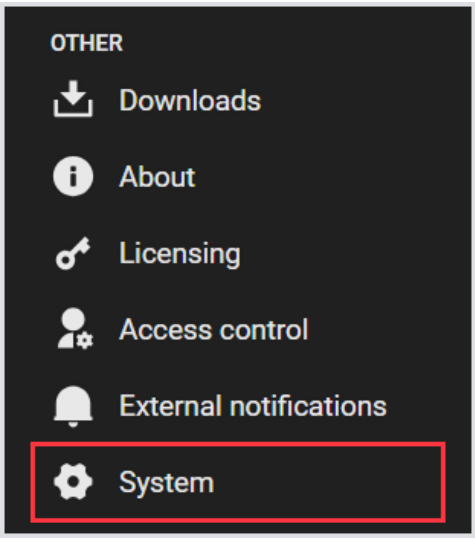
- Introduction
- Accessing Active Sessions
- Active Session Page Overview
- Filtering and Sorting Active Sessions

Introduction

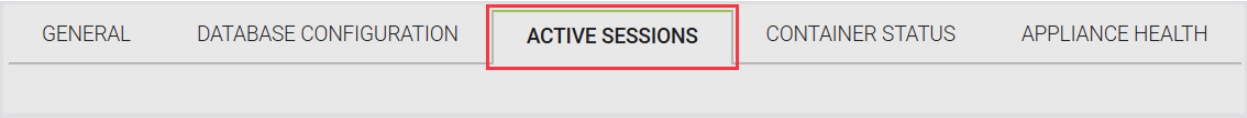
This section of the Troubleshooting page provides an overview of currently active sessions, the ability to filter and sort active sessions, and the ability to abort them.

Accessing Active Sessions

1. In the **Login Enterprise sidebar menu**, navigate to **Other > System**.



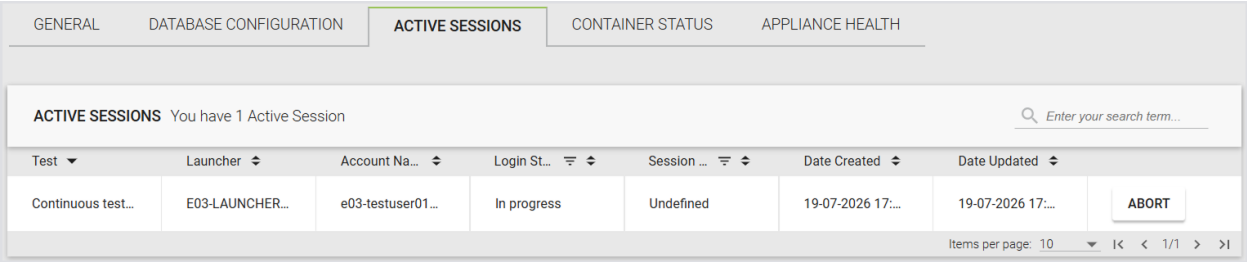
2. In **System**, select **Active sessions** from the tab bar menu, and wait for the page to load.



Active Sessions Page Overview

The Active Sessions page displays a table with the following information:

Test	Launcher	Account Name	Login State	Session State	Date Created
Name of the Test.	Name of the Launcher.	Name of the Launcher.	State of the login: succeeded or failed.	State of the session, e.g., Undefined.	Date when the active session was created.



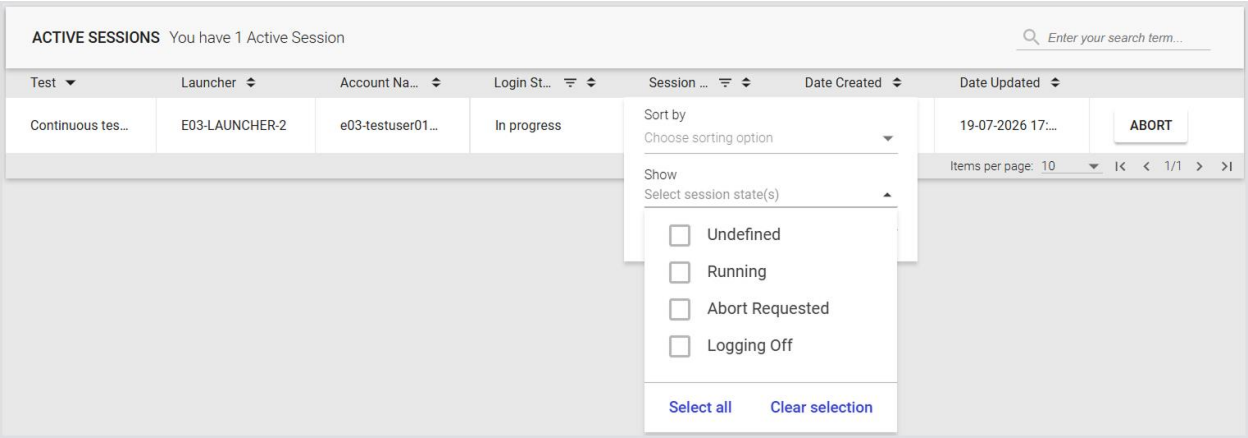
If you click **Abort**, the selected session will be terminated within one minute.

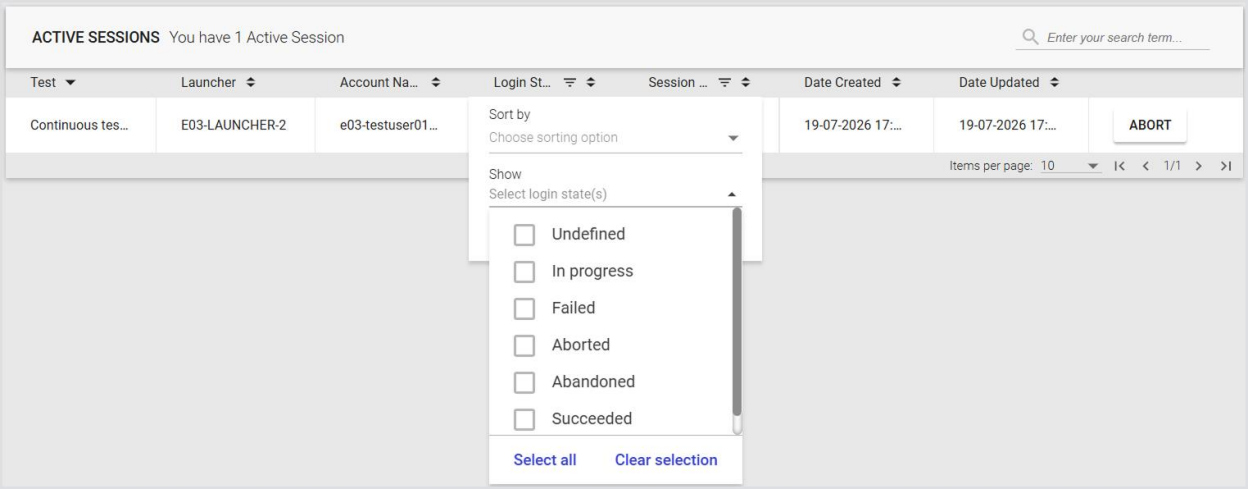
Filtering and Sorting Active Sessions

Use the search box to filter active sessions by:

- Test
- Launcher
- Account Name

Filter the **Login State** and **Session State** columns by selecting one or more values from their column filters.





Select any column heading to sort the table by that column. Select the heading again to reverse the sort order.

Monitoring Container Status

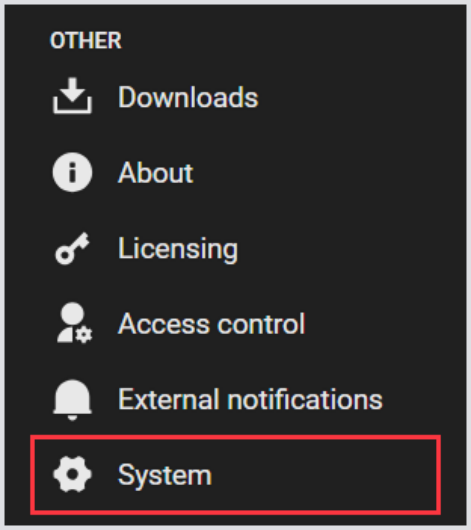
- Introduction
- Accessing the Container status
- Container Status Page Overview
- Troubleshooting
  - Container Last Status
  - Downloading Log Files

Introduction

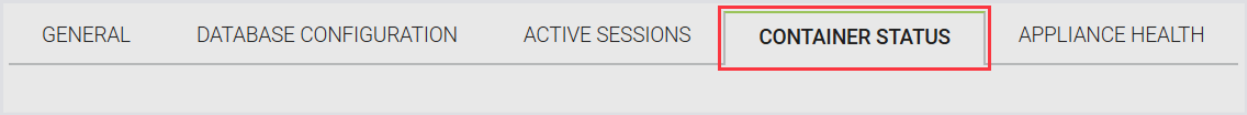
To simplify troubleshooting, the Container Status page provides an overview of the status of each container used in the Virtual Appliance. Since the Virtual Appliance is built on the Docker technology, each component is organized within specific containers, similar to Windows services. Each container has its own role and purpose. If you encounter unexpected behavior in Virtual Appliance, check this page to ensure all containers are functioning correctly.

Accessing the Container status

1. In the **Login Enterprise sidebar menu**, navigate to **Other > System**.

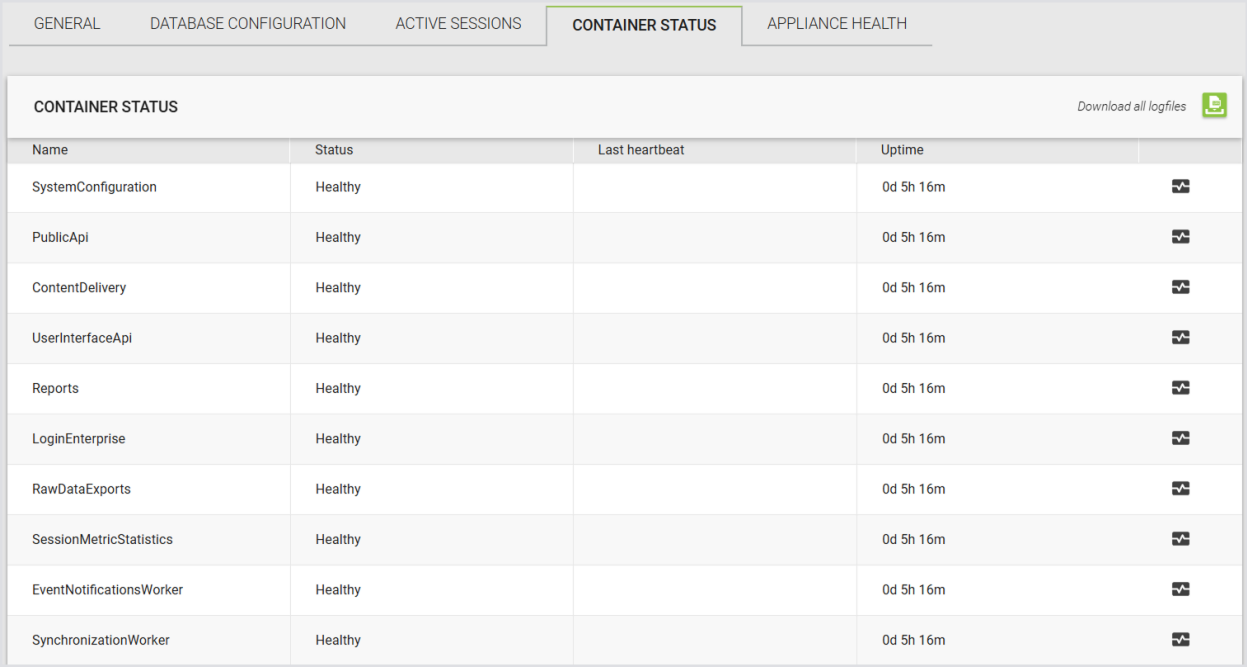


2. In **System**, select **Container status** from the tab bar menu and wait for the page to load.



Container Status Page Overview

This page displays various containers and their statuses. The status may change; ideally, all statuses should be set to Healthy.



Troubleshooting

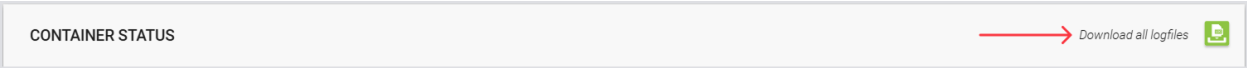
Container Last Status

Clicking the **Heart** icon will show you the latest status of each container:

```
{
  "status": "Healthy",
  "results": {
    "bootstrapping": {
      "status": "Healthy",
      "description": "Finished bootstrapping",
      "data": {
        "Status": "Running",
        "StartTime": "2026-03-20T05:57:49.2151287+00:00",
        "Running time": "05:27:03.2351062"
      }
    },
    "database": {
      "status": "Healthy",
      "description": "Database is ready",
      "data": {}
    },
    "exchanges": {
      "status": "Healthy",
      "description": "All message exchanges are healthy",
      "data": {
        "system-configuration.test-email": "OK. No messages sent yet",
        "system-configuration.email-settings": "OK. No messages sent yet",
        "system-configuration.data-retention-settings": "OK. No messages sent yet",
        "system-configuration.database-status": "OK. No messages sent yet",
        "system-configuration.notification-settings": "OK. No messages sent yet",
        "system-configuration.container-status": "OK. No messages sent yet",
        "system-configuration.license-states": "OK. Last Message sent = 2026-03-20T05:58:13.1375488+00:00 (-05:26:39.3126875)"
      }
    }
  }
}
```


Downloading Log Files

If a container's status deviates from Healthy and does not resolve, click **Download All Logfiles**. This action will download a ZIP file.



Please email support@loginvsi.com, including the zipped file and a description of the behavior you are experiencing.

TIP: If you prefer to review the logs yourself, you can use the SEQ application to make the files readable. For more information about this application, see the Seq documentation.

Monitoring Appliance Health

- Overview
- Benefits of Appliance Health Monitoring
- Key metrics monitored
- Technical details on metrics monitoring
  - Metric gathering methods
- Accessing Appliance Health Monitoring
- Using Appliance Health Monitoring
  - Downloading charts as an image
- Notifications
  - Notifications in the user interface
  - Email notifications
- Goals and takeaways

Overview

Appliance Health Monitoring helps you proactively manage and troubleshoot the performance of your Login Enterprise Virtual Appliance. By monitoring key metrics, you can ensure optimal operation and resource utilization, preventing minor issues from becoming major problems.

Benefits of Appliance Health Monitoring

Benefit	Explanation
Real-time health monitoring	Allows you to continuously monitor the health of your Appliance in real time.
Early issue detection	Helps identify and address potential problems before they escalate into major issues.
Disk space visibility	Provides clear insight into available disk space, helping prevent critical shortages.
Performance impact awareness	Enables observation of how heavy Tests affect Appliance performance.
Usage trend tracking	Tracks usage patterns over time to support better planning and analysis.
Informed resource allocation	Helps you make decisions about allocating resources based on collected data.

Key metrics monitored

Appliance Health Monitoring provides insights into the following metrics:

1. **Disk utilization**

- **Description**: Displays the percentage of total disk space currently in use.
- **Importance**: Helps you keep track of available disk space and avoid running out, which can lead to critical issues.

2. **Disk utilization (Inodes)**

- **Description**: Shows the percentage of inodes used versus the total available. An inode represents a file or directory, so running out of inodes means you can't create new files or directories even if there is free disk space.
- **Importance**: Useful for managing systems with a large number of small files, preventing inode exhaustion.

3. **CPU utilization**

- **Description**: Indicates the percentage of CPU resources currently in use.
- **Importance**: Helps you monitor how efficiently your Appliance's CPU is being used and identify potential performance bottlenecks.

4. **Memory utilization**

- **Description**: Shows the percentage of RAM currently in use.
- **Importance**: Provides insight into memory usage, helping you understand if your Appliance needs more resources or if there are inefficiencies.

Technical details on metrics monitoring

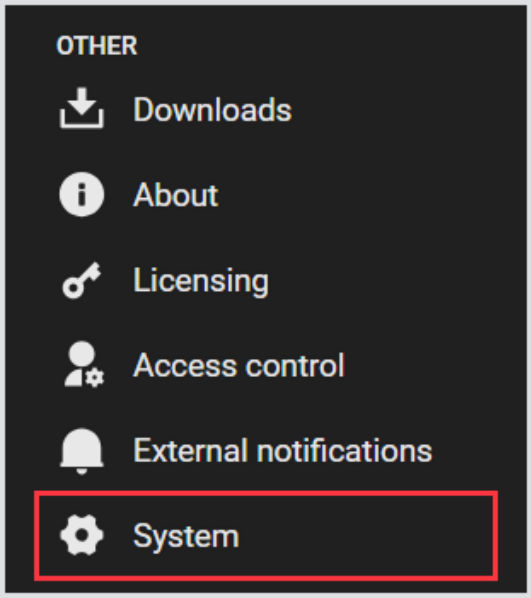
- **Disk space and inodes check**: We check for disk space and inodes every 20 minutes.
- **Disk and Memory utilization**: We monitor disk and RAM utilization every 20 seconds. The results are averaged per minute, which is then displayed in the daily graph view.

Metric gathering methods

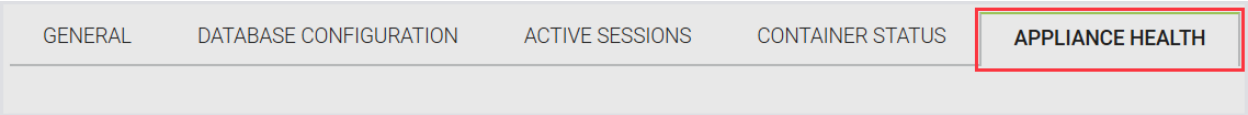
- **CPU metrics**: We read the `/proc/stat` file at a 20-second interval and calculate the delta using the algorithm in c#. For details, see Gathering CPU utilization from /proc/stat.
- **Memory metrics**: We gather memory metrics from /proc/meminfo, where the calculation is performed as follows: `UsedMemory = MemTotal - MemFree`.
- **Disk metrics**: Disk usage is monitored using the `df` command. For details, see the df Command documentation.
- **Inodes metrics**: Inodes are monitored using the `df` command with the `-i` option.

Accessing Appliance Health Monitoring

1. In the **Login Enterprise sidebar menu**, navigate to **Other > System**.

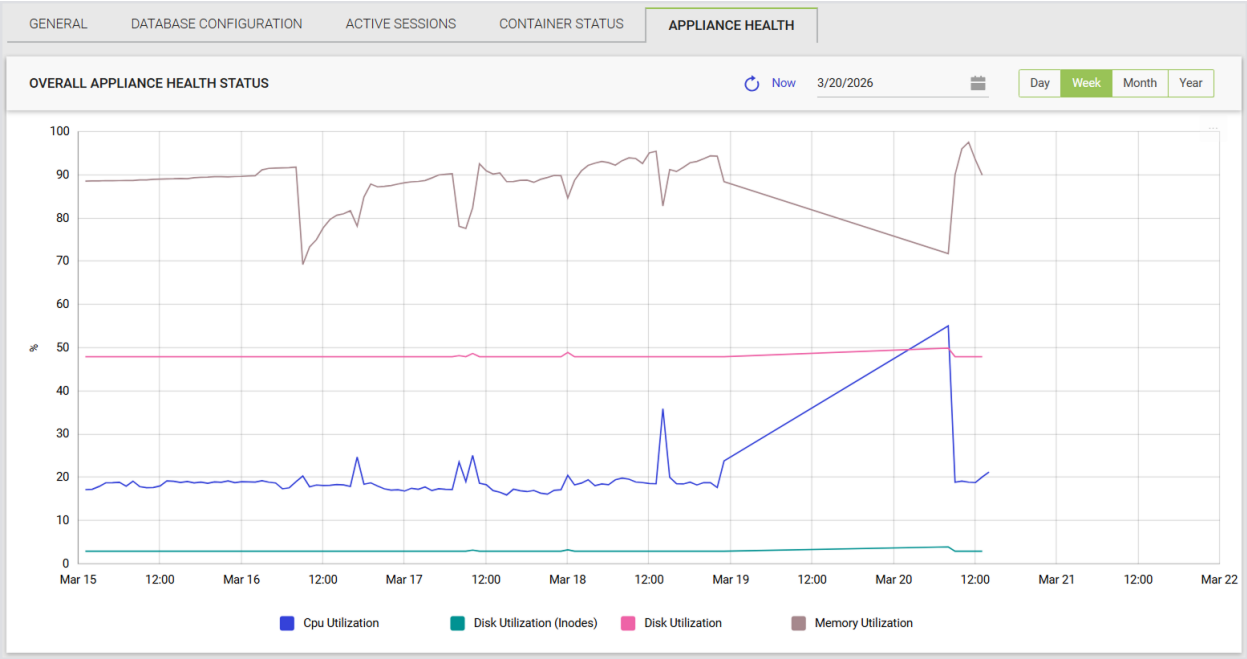


2. In **System**, select **Appliance Health** from the tab bar menu.

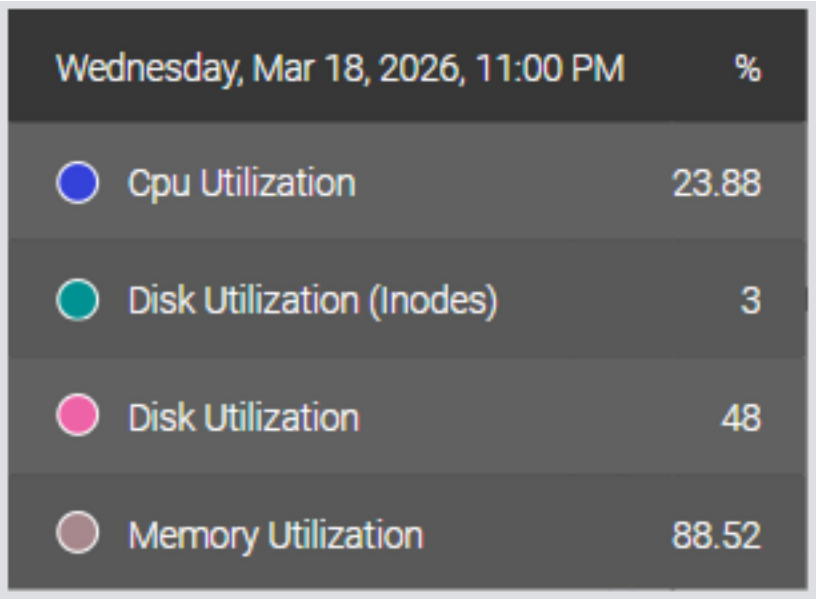


Using Appliance Health Monitoring

1. **Navigate to the feature**: In the **Sidebar menu**, go to **System > Appliance Health**.
2. **View metrics**: The main screen displays the 4 key metrics:
    1. CPU utilization
    2. Disk utilization (Inodes)
    3. Disk utilization
    4. Memory utilization.



TIP: Hover over the trend to see the metric details.

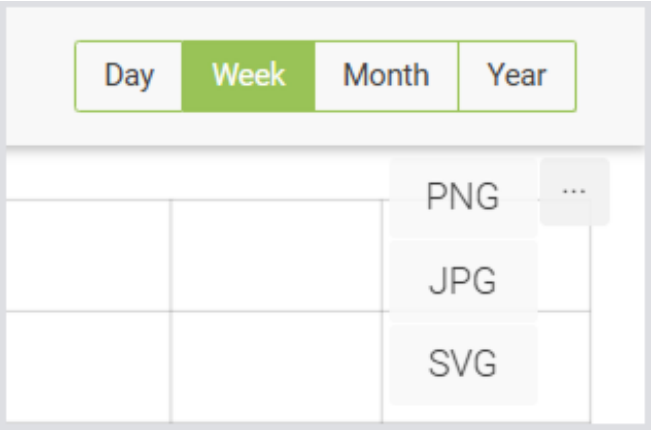


3. **Analyze data**: Review the metrics to understand your Appliance's current health status. Check for any unusual trends or values that might indicate potential issues.
4. **Take action**: Based on the data, you can take appropriate action, such as freeing up disk space, managing inode usage, or upgrading resources if necessary.

Downloading charts as an image

You can download charts as images for sharing or use in custom reports and presentations. Downloading a chart as an image applies to all charts: Load Test, Continuous Test, and Appliance Health. To download a chart:

1. Locate the Download icon in the top right corner of the chart.
2. Select image format: PNG, SVG, or JPG.
3. Download the image. It will be saved to your Downloads folder, or the directory you've configured as your default Downloads location.



Notifications

User interface

The Appliance Health Monitoring interface includes persistent notifications that provide real-time alerts about system health. Currently, these notifications cannot be disabled and will always appear when certain thresholds are exceeded.

Key features:

- **Threshold alert**: A notification appears when disk usage exceeds 90%.
- **Dismissal**: You can dismiss the notification; however, this action is saved only for the current session. If the issue is not addressed and you relaunch the browser, the notification will reappear. It will not reappear for the same session after dismissal.
- **Display timing**: Notifications are triggered in 20-minute intervals. If you log in shortly after the check, you may not see the alert immediately, leading to a potential wait of up to 20 minutes for the notification to appear.

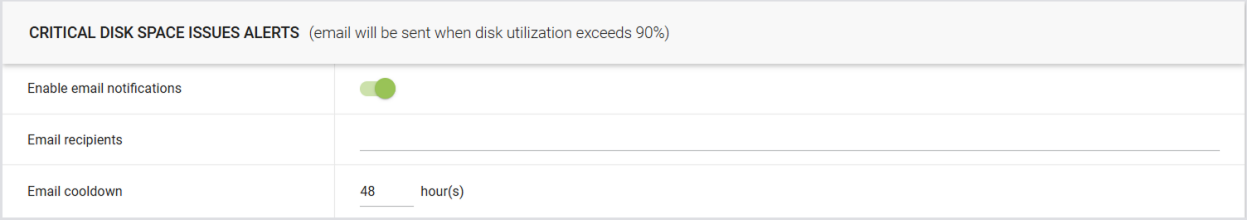
Email notifications

The Appliance Health Monitoring system allows for customizable email notifications to keep you informed about critical issues.

Configuration options:

- **Enable/Disable notifications**: You can easily enable or disable email notifications based on your preferences.
- **Recipient settings**: Specify recipients to receive alerts when disk usage exceeds the 90% threshold.
- **Cooldown period**: The default cooldown period is set to 48 hours. This means that if a notification is sent due to high disk usage, the system will check again after 48 hours before sending another email. If the issue is resolved within that timeframe, no further emails will be sent.

NOTE: Email notifications are currently only available for disk usage alerts, triggered by the same threshold of over 90% disk capacity.



Goals and takeaways

- **Proactive monitoring**: Stay ahead of potential issues by keeping an eye on critical metrics.
- **Informed decisions**: Use the data to make decisions about resource allocation and maintenance.
- **Efficient operation**: Ensure that your Appliance runs smoothly and efficiently by addressing issues before they impact performance.

NVIDIA nVector Integration

- Overview
- Requirements
- Quick start
  - What you'll get
  - What to check after the first run
  - Enabling browser viewing of latency metrics
  - How to view and export results
  - Important operational notes
  - Troubleshooting: quick checks
- Contact and support
- Appendix: File reference

Overview

Login VSI, in collaboration with NVIDIA, has developed an integration between Login Enterprise and NVIDIA nVector that leverages GPU and virtual GPU (vGPU) performance testing and analysis technology. This integration enables you to collect round-trip graphics latency data and image quality scores from your VDI environments, upload them to Login Enterprise as platform metrics, and use them to optimize GPU-based configurations for performance and cost-efficiency.

Latency measurement is a production feature of Login Enterprise. SSIM-based image quality scoring is available as a usable preview - functional and customer-usable, but not yet a built-in Login Enterprise feature. For full setup and usage details, refer to the installation guide PDF in the GitHub repo.

Requirements

- Login Enterprise Appliance v5.13.6 or higher (which has Platform Metrics enablement).
- Supported VDI stacks: Citrix, VMware/Omnissa (Horizon), RDS, Azure Virtual Desktop.
- **NVIDIA GPU (or vGPU) present on the Launcher machine** - nVector workloads require GPU-enabled Launchers.
- `nvector-agent.exe` on the Appliance and each Launcher. v1.x is bundled in Login Enterprise 6.4 and newer. v2.0.0 is required for SSIM preview functionality and is available from the repository or from Blair Parkhill (b.parkhill@loginvsi.com).
- `ssim-tool.exe` on the Launcher (v1.1.0, required for SSIM preview mode only). Available from Blair Parkhill (b.parkhill@loginvsi.com).
- Launcher access to the Login Enterprise API (for metric upload).
- Following the Quick Start section below.

Login Enterprise 6.4 update (bundled agent and workload)

Login Enterprise 6.4 and newer include the NVIDIA nVector agent and the "NVIDIA nVector Desktop Prepare" workload inside the Appliance by default. These files are present only if the Appliance was originally deployed as version 6.4 or newer.

If your Appliance was deployed as 6.4 or newer:

- No manual upload of `nvector-agent.exe` is required
- The agent is already located at `loginvsi/content/scriptcontent/nvector-agent.exe`
- The Desktop Prepare workload is already listed under Applications
- You can skip the manual upload steps shown elsewhere in this section

NOTE: The agent and workload bundled with Login Enterprise 6.4 support latency measurement only. For SSIM preview functionality, you will need to replace the bundled agent with `nvector-agent.exe v2.0.0` and replace the Desktop Prepare workload with the updated version from the repository. `ssim-tool.exe v1.1.0` is also required. Contact Blair Parkhill (b.parkhill@loginvsi.com) to obtain the binaries.

If your Appliance was upgraded from an older version, the bundled agent and workload will not appear. Follow the full setup steps, including manual upload.

SHA256 checksum of the bundled agent (v1.0.0): `2D6F61B5A67F72F8466E14CBCF625985C449898F867A0785A48909E8802E677B`

Quick Start

1. `git clone` https://github.com/LoginVSI/nVector
2. **Upload the agent to the Appliance**
    1. Via a tool like WinSCP, place `nvector-agent.exe` v2.0.0 in the Appliance script content folder at `loginvsi/content/scriptcontent/nvector-agent.exe`. This replaces the bundled v1.x agent - use the same path and filename.
3. **Create an Environment** in Login Enterprise and note the `EnvironmentId` (UUID). To learn more, see Creating an Environment.
4. **Upload the workload**
    1. Add `NVIDIA_nVector_Desktop_Prepare.cs` from the repository as a custom application/workload in Login Enterprise. Remove the existing NVIDIA nVector Desktop Prepare workload first and import the updated version. For details, refer to Importing Application Scripts.
5. **Prepare Launchers** (per Launcher you will use for latency measurements):
    1. Copy the following to the Launcher (recommended path is C:\ProgramData\NVIDIA Corporation\nVector): `nVector_Client_Prepare.ps1`, `nVector_SSIM_Daemon.ps1`, `nVector_Startup_Script.cmd`, `nvector-agent.exe (v2.0.0)`, `ssim-tool.exe` (required for SSIM mode).

        1. **NOTE**: `nVector_Client_Prepare.ps1` and `nVector_SSIM_Daemon.ps1` must be in the same folder. Do not invoke `nVector_SSIM_Daemon.ps1` directly - it is spawned automatically by `nVector_Client_Prepare.ps1`.
    2. Edit `nVector_Client_Prepare.ps1` and set these values in the User Config section, or pass them on the command line:
        1. BaseUrl > your Login Enterprise appliance base URL.
        2. ConfigurationAccessToken > API token with Configuration access level. For more information, consult Adding a System Access Token.
        3. EnvironmentId > the UUID from step 3 (for latency metrics).
        4. TargetHost > hostname of the target/desktop machine.
        5. SsimEnvironmentId > UUID of a second LE environment for SSIM metrics (recommended - see installation guide for details on single vs. two environment setup).
        6. Mode > both (default), latency, or ssim.
            1. For the full parameter reference, see the installation guide PDF in the repository.
    1. Place `nVector_Startup_Script.cmd` in the user Startup folder (like "*shell:common startup*" or create a scheduled task to run at logon, such as:

        ```
        1. powershell.exe -ExecutionPolicy Bypass -File
           "C:\temp\nvidia\nVector_Client_Prepare.ps1"
        ```

6. **Create a Test** in Login Enterprise (Load Test or Continuous):
    1. Use the Environment created earlier.
    2. Add the `nVector_Desktop_Prepare` workload to a session.
    3. **IMPORTANT:** Set the connector/session to run **full-screen** on the Target VM (nVector captures the top-most full-screen session - see note below).
7. **Run a short smoke test** (60–120s) to confirm datapoints are uploaded to Platform Metrics and the CSV uploader writes files.

Now you're ready to scale up and run further actual testing.

What You'll Get

- Scatter/time charts of endpoint graphical latency (ms) in Login Enterprise Platform Metrics.
- SSIM-based image quality scores in Login Enterprise Platform Metrics (preview feature - requires `nvector-agent.exe v2.0.0` and `ssim-tool.exe v1.1.0`).
- Per-launcher CSV/JSON exports of latency and SSIM data via the included `get_nVectorMetrics.ps1`.
- Ability to compare GPU vs no-GPU scenarios and correlate with other Login Enterprise measurements (EUX, VSImax, session metrics, etc.).

What to Check After the First Run

- Platform Metrics in the Appliance's UI show latency datapoints (scatter/time).

- If running in SSIM mode: Platform Metrics in the SSIM environment show SSIM score data points.
- The Launcher folder contains nvector logs and the SSIM daemon log.
- `get_nVectorMetrics.ps1` can pull data for your EnvironmentId and time window. To retrieve latency and SSIM data together, use -EnvironmentIds "latency-env-uuid", "ssim-env-uuid".

Enabling Browser Viewing of Latency Metrics

1. Open the Login Enterprise UI and sign in with an account that can view the Appliance UI.
2. Open the browser **Developer Tools**:
    - Windows: press **F12** or **Ctrl + Shift + I**
    - macOS: **Cmd + Option + I**
3. Select the **Console** tab in Developer Tools.
4. Paste the following command at the console prompt and press **Enter**:

    1. `leSetFeatureFlag('platformMetrics', 1)`
    2. This will enable the Platform Metrics view, so nVector metrics (latency, SSIM, etc.) become visible in the UI. This feature flag is currently required on all Login Enterprise versions.

5. Refresh the page (if the metrics view does not appear immediately) and navigate to the **Platform Metrics** / Results area to view the nVector charts.

How to View and Export Results

- **In the UI:**
    - For Continuous Test: select Platform Metrics > the environment/time window > view Endpoint Latency scatter and timeseries.
    - For Load Test: the measurements will automatically display in the test results
- **Export**: Run `get_nVectorMetrics.ps1` with `-LEApiToken`, `-EnvironmentId` (or `-EnvironmentIds` for multiple), and either `-LastHours` or `-StartTime`/`-EndTime` to export CSV/JSON. Use `-EnvironmentIds` "latency-env-uuid", "ssim-env-uuid" to pull latency and SSIM data together in one run. (Optional: `-ImportServerCert` and `-KeepCert` for appliances using self-signed/private CA certs.)
    - **If you don't see metrics:** Confirm the API token, `EnvironmentId`, and that the Launcher uploaded CSVs to the Appliance. If you hit SSL/certificate errors, try `-ImportServerCert` (and ensure `-BaseUrl` matches the server's cert hostname).

Important Operational Notes

- **Launcher must be GPU-enabled.** nVector requires the Launcher to have an NVIDIA GPU/vGPU — CPU-only Launchers will not generate latency telemetry.

- **Latency is collected only from the top-most full-screen session on a Launcher.** If multiple desktop sessions run on a single Launcher, only the active full-screen session will emit latency measurements.
    - **Best practice:** if possible, run one Target session per Launcher when you require latency metrics.

Troubleshooting: Quick Checks

- **No latency datapoints:**
    - Confirm `nvector-agent.exe` exists and `nVector_Client_Prepare.ps1` ran successfully at logon.
    - Verify the test session is full-screen on the Target VM (only full-screen sessions produce latency).
    - Confirm Launcher is GPU-enabled and drivers are up to date.
- **CSV upload failed/missing exports:**
    - Inspect local Launcher logs (path set in `nVector_Client_Prepare.ps1`).
    - Run `get_nVectorMetrics.ps1` manually with the `EnvironmentId`/API token and check for HTTP errors.
- **Platform Metrics UI not visible:**
    - Enable the platform metrics UI flag as documented in the Enabling Browser Viewing section above. The flag is currently required on all Login Enterprise versions. Confirm the Appliance is v5.13.6 or newer.

(See the PDF in the repo for log examples, feature-flag commands, and advanced diagnostics.)

Contacts and Support

- **Repo and primary docs:** https://github.com/LoginVSI/nVector
- **Agent / binary requests and escalation:** Blair Parkhill — b.parkhill@loginvsi.com

Appendix: File Reference

- `nVector_Client_Prepare.ps1` - main launcher-side orchestration script. Manages the nVector agent lifecycle, collects latency data, uploads platform metrics to Login Enterprise, and coordinates SSIM processing via the daemon.
- `nVector_SSIM_Daemon.ps1` - SSIM processing daemon. Spawned automatically by `nVector_Client_Prepare.ps1` when running in SSIM or both modes. Do not invoke directly. Must be in the same folder as `nVector_Client_Prepare.ps1`.
- `nVector_Startup_Script.cmd` - startup wrapper that invokes `nVector_Client_Prepare.ps1` automatically at logon on the Launcher.
- `NVIDIA_nVector_Desktop_Prepare.cs` - Login Enterprise workload for the target/desktop machine. Downloads and starts `nvector-agent.exe` in the desktop role during a test session.

- `get_nVectorMetrics.ps1` - export helper that queries the Login Enterprise API and outputs timestamped CSV and JSON. Supports single or multiple environment IDs in one run.
- `Create-NvidiaGpuSessionMetricGroup.ps1` - one-time setup script that creates a pre-configured NVIDIA GPU session metrics group in Login Enterprise via the API.
- `Add-SessionMetrics.ps1` - interactive script for discovering and registering additional PerfMon and WMI session metrics in Login Enterprise.
- `nVector_Agent_Installation_x.pdf` - full installation guide, parameter reference, and troubleshooting guide (in repo).

Nutanix Platform Metrics Integration

- Overview
- What You Can Do With It
- Metrics Collected
- Requirements
- Login Enterprise Setup
- Quick Start
- How Long to Run the Collector
- Collector Script Parameters
- Retrieval Script
- Where to Run the Scripts
- Running as a Windows Scheduled Task
- Finding Your Cluster ExtId
- Certificate Handling
- Power BI
- Caveats and Known Behaviors
- Troubleshooting
- Security
- Files
- What's Coming
- Additional Resources

Overview

Login Enterprise can pull infrastructure metrics directly from Nutanix Prism and display them alongside your test results in Platform Metrics. During a Load Test or Continuous Test, you see EUX scores, session measurements, and Nutanix cluster, host, and VM metrics on the same timeline, without switching tools.

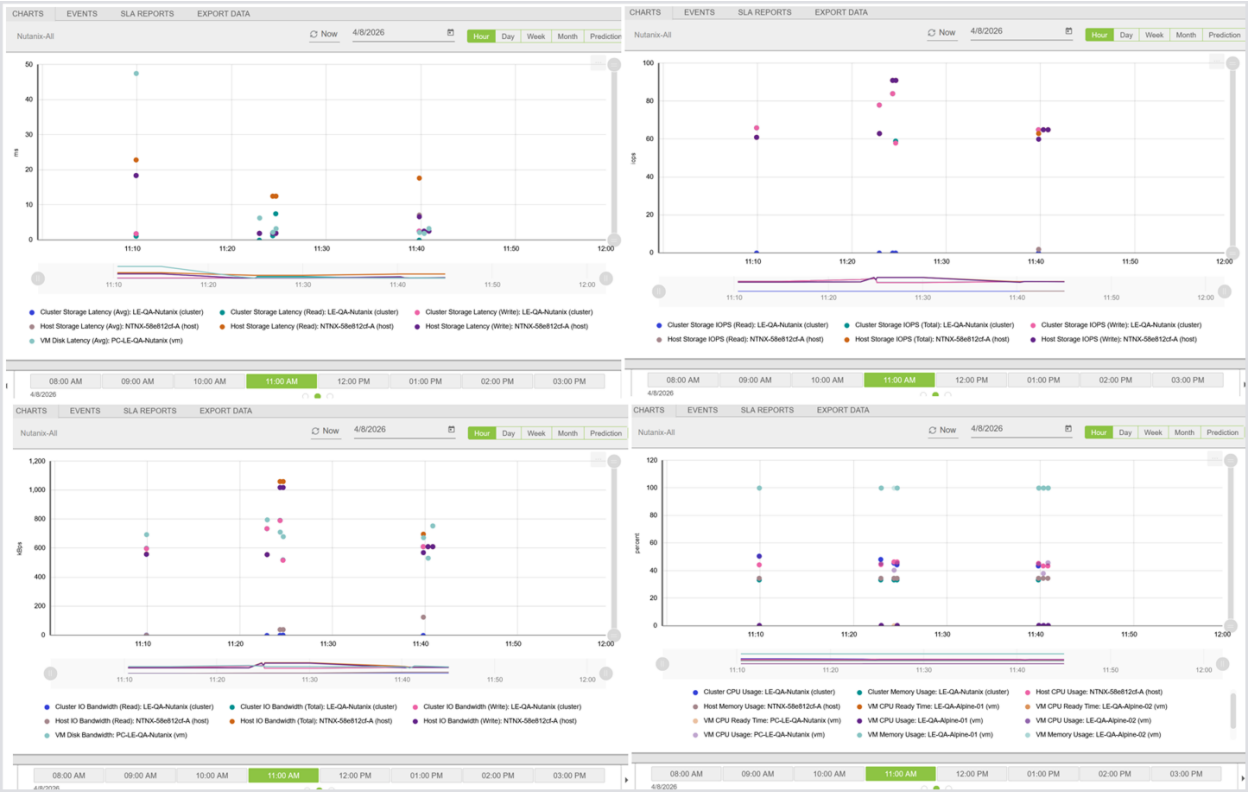
The solution is available in the Login VSI GitHub repo.

What You Can Do With It

Three use cases drive most of the value here:

1. **Continuous Testing** keeps Nutanix infrastructure metrics flowing 24/7 alongside user experience scores. If something changes overnight, you see it in Platform Metrics the next morning, next to the EUX data, not in a separate tool.
2. **Load Testing** lets you correlate infrastructure saturation with EUC performance degradation as user count climbs. When EUX scores drop, you can check immediately whether the host CPU was already at 90%, whether storage latency climbed during the ramp, or whether a specific VM's disk bandwidth maxed out.

3. **Migration and bake-off** scenarios let you run the same workload across two configurations, or two EUC stacks on the same Nutanix infrastructure, and compare Platform Metrics side by side. CPU behavior, IOPS, and VM-level latency across identical workloads. The data makes platform decisions easier to justify.



Metrics Collected

Cluster-Level (via Prism Central v4 API)

Metric	Unit
Cluster CPU Usage	percent
Cluster Memory Usage	percent
Cluster Storage IOPS (Total, Read, Write)	iops
Cluster Storage Latency (Avg, Read, Write)	ms
Cluster IO Bandwidth (Total, Read, Write)	kBps

Host-Level (via Prism Central v4 API)

Metric	Unit

Host CPU Usage	percent
Host Memory Usage	percent
Host Storage IOPS (Total, Read, Write)	iops
Host Storage Latency (Avg, Read, Write)	ms
Host IO Bandwidth (Total, Read, Write)	kBps

VM-Level (via Prism Central v4 API)

Metric	Unit
VM CPU Usage	percent
VM CPU Ready Time	percent
VM Memory Usage	percent
VM Disk Latency (Avg)	ms
VM Disk Bandwidth	kBps
VM Network RX	bytesPerSec
VM Network TX	bytesPerSec

Requirements

Nutanix

Component	Minimum Version	Notes
AOS	7	Required for all collection
Prism Central	7.5	Required for all collection
AHV	10.x	VM stats tested on AHV only

The integration has been tested against AOS 7.5 and Prism Central 7.5.1.1.

On Prism Central 7.5 and newer, the script negotiates the API version automatically. On older PC versions (any pc.YYYY.x format), it defaults to v4.0.

Login Enterprise

- Login Enterprise appliance with Platform Metrics enabled
- API token with **Configuration** access level

- Network access from the machine running the script to the Nutanix port 9440 and the Login Enterprise port 443

PowerShell

- Windows PowerShell 5.1 (tested and validated)
- PowerShell 7.x (compatible)
- No admin privileges required

Login Enterprise Setup

1. **Enable Platform Metrics**
    1. Platform Metrics is currently a preview feature. Enable it in your browser's DevTools console (F12 in most browsers):
       `leSetFeatureFlag('platformMetrics', 1)`
    2. Refresh the page after running.
2. **Create an Environment**
    1. In Configuration > Environments, create a single environment named, for example, Nutanix Platform Metrics. This environment receives all metric types from the collector.
    2. Copy the UUID of the environment from the browser URL bar when viewing it. You will use this as the -LEEnvironmentId parameter when running the script.
3. **Create a Test**: Link the environment to a continuous test or a load test. The Platform Metrics tab appears on test result pages once an environment is linked. For more on creating tests, see Configuring Continuous Testing and Configuring Load Testing.
4. **Create an API Token**: Configuration > System Access Tokens > create a token with **Configuration** access level.

Quick Start

If you omit `-NutanixPassword` or `-LEApiToken`, the script prompts for them securely at runtime. You can also pass them on the command line if running unattended.

Cluster + host + VM stats (requires Prism Central)

```
.\Nutanix-LE-PlatformMetrics.ps1 `

  -PrismCentralHost "your-prism-central.example.com" `

  -ClusterExtId "your-cluster-ext-id" `

  -LEApplianceUrl "https://your-le-appliance.example.com" `

  -LEEnvironmentId "uuid-nutanix-all" `

  -CollectHostStats `
```

```
    -CollectVmStats `
    -RunOnce
```

Continuous polling (run until Ctrl+C)

```
.\Nutanix-LE-PlatformMetrics.ps1 `
    -PrismCentralHost "your-prism-central.example.com" `
    -ClusterExtId "your-cluster-ext-id" `
    -LEApplianceUrl "https://your-le-appliance.example.com" `
    -LEEnvironmentId "uuid-nutanix-all" `
    -CollectHostStats `
    -CollectVmStats
```

Filter to specific VMs

```
.\Nutanix-LE-PlatformMetrics.ps1 `
    -PrismCentralHost "your-prism-central.example.com" `
    -ClusterExtId "your-cluster-ext-id" `
    -LEApplianceUrl "https://your-le-appliance.example.com" `
    -LEEnvironmentId "uuid-nutanix-all" `
    -CollectVmStats `
    -VmFilter "MyLauncher-01,MyLauncher-02,MyTarget-01" `
    -RunOnce
```

Dry run (verify connectivity without uploading)

```
.\Nutanix-LE-PlatformMetrics.ps1 `
    -PrismCentralHost "your-prism-central.example.com" `
    -ClusterExtId "your-cluster-ext-id" `
```

-LEApplianceUrl "https://your-le-appliance.example.com" `

-LEEnvironmentId "uuid-nutanix-all" `

-CollectHostStats `

-CollectVmStats `

-DryRun -RunOnce

How Long to Run the Collector

You have flexibility here. Two common approaches:

1. **Run it continuously** alongside a continuous test or ongoing monitoring window. The script loops on its own polling interval until you stop it with Ctrl+C, or you can set it up as a scheduled task or startup item on the data collector machine (see Running as a Windows Scheduled Task). This is the right approach for continuous testing or overnight validation runs.
2. **Start and stop it around specific tests.** If you only care about infrastructure data during a particular load test, just launch the script before the test starts and stop it when the test ends. Use -RunOnce for a single collection iteration, which is handy for quick spot-checks or verifying connectivity before committing to a longer run.

Collector Script Parameters

Nutanix Connection

Both `-NutanixPassword` and `-LEApiToken` are optional on the command line. If either is omitted, the script prompts for it securely at runtime; the input is masked and never echoed. Neither is ever written to the script's log files or stored anywhere by the script. If your environment uses a secrets vault or credential manager, you can modify the script to retrieve credentials from that source.

Parameter	Required	Description
-NutanixPassword	No	Prism Central admin password
-NutanixUser	No	Prism Central username (default: admin)
-PrismCentralHost	Yes	Prism Central IP or hostname. Required for all collection.
-ClusterExtId	Conditional	Required for all collection

Login Enterprise Connection

Parameter	Required	Description
-LEApiToken	No	Login Enterprise API token (Configuration access level)
-LEApplianceUrl	Yes	Full URL of the Login Enterprise appliance
-LEApiVersion	No	API version segment (default: v8-preview)

Environment ID

Parameter	Description
-LEEnvironmentId	UUID of the Nutanix-All environment you created during setup

Collection Scope

Parameter	Default	Description
-CollectHostStats	off	Enable host-level metrics (requires PC)
-CollectVmStats	off	Enable VM-level metrics (requires PC)
-VmFilter	(all)	Comma-separated VM names to collect. If omitted, it collects all powered-on VMs. Names are case-sensitive and must match exactly as they appear in Prism Central.

Run Control

Parameter	Default	Description
-RunOnce	off	Run one iteration and exit
-Iterations	0 (unlimited)	Run N iterations and exit
-PollingIntervalSec	30	Seconds between polls
-DryRun	off	Collect metrics but do not upload to LE
-SkipTimeSync	off	Skip LE time sync

Output and Diagnostics

Parameter	Description
-LogDir	Directory for log files (default: script directory)
-Verbose	Enable detailed per-metric logging

Certificate Handling Parameters

Parameter	Description
-IgnoreCertificateErrors	Bypass TLS certificate validation for the Login Enterprise appliance connection. See Certificate Handling.

Retrieval Script

Get-LEPlatformMetrics.ps1 queries the Login Enterprise API and exports metrics to CSV and JSON. Use it to verify data after a run, pull results for analysis, or feed Power BI.

Examples

```
# Last 1 hour

.\Get-LEPlatformMetrics.ps1 `

    -LEApiToken "YourToken" `

    -BaseUrl "https://your-le-appliance.example.com" `

    -EnvironmentId "uuid-nutanix-all" `

    -LastHours 1

# Specific time range

.\Get-LEPlatformMetrics.ps1 `

    -LEApiToken "YourToken" `

    -BaseUrl "https://your-le-appliance.example.com" `

    -EnvironmentId "uuid-nutanix-all" `

    -StartTime "2026-04-08T10:00:00.000Z" `

    -EndTime "2026-04-08T11:00:00.000Z"
```

Output Files

File	Description
Get-LEPlatformMetrics_YYYYMMDD_HHMMSS.csv	Flat data, one row per data point

Get-LEPlatformMetrics_YYYYMMDD_HHMMSS.json	Raw API response
Get-LEPlatformMetrics_Log_YYYYMMDD_HHMMSS.txt	Run log

CSV columns: timestamp, value, metricId, environmentKey, displayName, unit, instance, componentType, group, customTags

Parameters

Parameter	Required	Description
-LEApiToken	Yes	Login Enterprise API token
-BaseUrl	Yes	Login Enterprise appliance URL
-EnvironmentId	Yes	Environment UUID
-LastHours	No	Hours of data to retrieve (default: 1)
-StartTime	No	ISO 8601 start time (overrides -LastHours)
-EndTime	No	ISO 8601 end time
-ApiVersion	No	Login Enterprise API version (default: v8-preview)
-OutputDir	No	Output directory (default: script directory)
-MetricGroups	No	Filter by metric group name
-IgnoreCertificateErrors	No	Bypass TLS certificate validation for the Login Enterprise appliance connection. See Certificate Handling.

Where to Run the Scripts

The scripts run from any Windows machine with network access to Nutanix (port 9440) and the Login Enterprise appliance (port 443). Common setups include a dedicated management or jump box, a Login Enterprise Launcher VM on the same network as the Nutanix cluster, or a workstation with VPN access to both environments. There is no agent, no service installation, and no requirement to run on the Nutanix host itself.

Running as a Windows Scheduled Task

One way to keep the collector running continuously is to set it up as a Windows Scheduled Task. This is just one example. How you manage startup and persistence on your data collector host is entirely up to you. A login startup script, a service wrapper, or just launching it manually before a test window all work fine.

For a scheduled task setup:

1. Open T**ask Scheduler** and select **Create Task**.
2. On the **General** tab, give the task a name (e.g., "Nutanix LE Metrics"), select **Run whether user is logged on or not**, and check **Run with highest privileges**.
3. On the **Triggers** tab, create a new trigger on a schedule. Daily with a repeat interval works well if you want the OS to restart the script on failure. The script handles its own polling loop internally.
4. On the **Actions** tab, create a new action: Start a program. Set the program to powershell.exe and the arguments to:
   -NonInteractive -ExecutionPolicy Bypass -File "C:\Scripts\Nutanix-LE-PlatformMetrics.ps1" -NutanixPassword "YourPassword" -LEApiToken "YourToken" -LEApplianceUrl "https://your-le-appliance.example.com" -LEEnvironmentId "your-env-uuid" -CollectHostStats -CollectVmStats -PrismCentralHost "your-prism-central.example.com" -ClusterExtId "your-cluster-ext-id"

NOTE: Because scheduled tasks run unattended, `-NutanixPassword` and `-LEApiToken` must be passed explicitly. The secure prompt only works when running the script manually.

5. On the **Settings** tab, check **If the task is already running, do not start a new instance**.

TIP: Avoid hardcoding passwords in task arguments visible to other users on shared machines. Use a dedicated service account with read-only Nutanix access where possible.

Finding Your Cluster ExtId

The Cluster ExtId is required for host and VM stats. Retrieve it from Prism Central using your browser or a REST client:

https://your-prism-central.example.com:9440/api/clustermgmt/v4.0/config/clusters

Authenticate with your Prism Central credentials and look for the extId field in the response for your cluster.

Certificate Handling

Most environments won't need this. If the script connects to your Login Enterprise appliance without any SSL errors, skip this section.

If you receive an SSL/TLS error such as "The underlying connection was closed: Could not establish trust relationship for the SSL/TLS secure channel," the machine running the scripts does not trust the certificate presented by the Login Enterprise appliance.

Nutanix (Prism Central)

Prism uses self-signed certificates by default. The scripts handle this automatically: no configuration needed on the Nutanix side. If your organization has replaced the default Prism

certificate with a CA-issued cert, ensure that CA's root certificate is trusted on the machine running the scripts. The scripts do not manage Nutanix certificates.

Login Enterprise appliance: if using a certificate from your organization's CA

Export the root CA certificate (and any intermediate CA certificates) and install them into Trusted Root Certification Authorities on the machine running the scripts. For guidance on managing certificates in the Windows certificate store, see the Microsoft documentation on Trusted Root Certification Authorities. For Login Enterprise appliance certificate management, see Managing Certificates on the Appliance.

Once the certificate is trusted on the machine, run the scripts normally without any additional flags.

Login Enterprise appliance: if using the default LE self-signed certificate

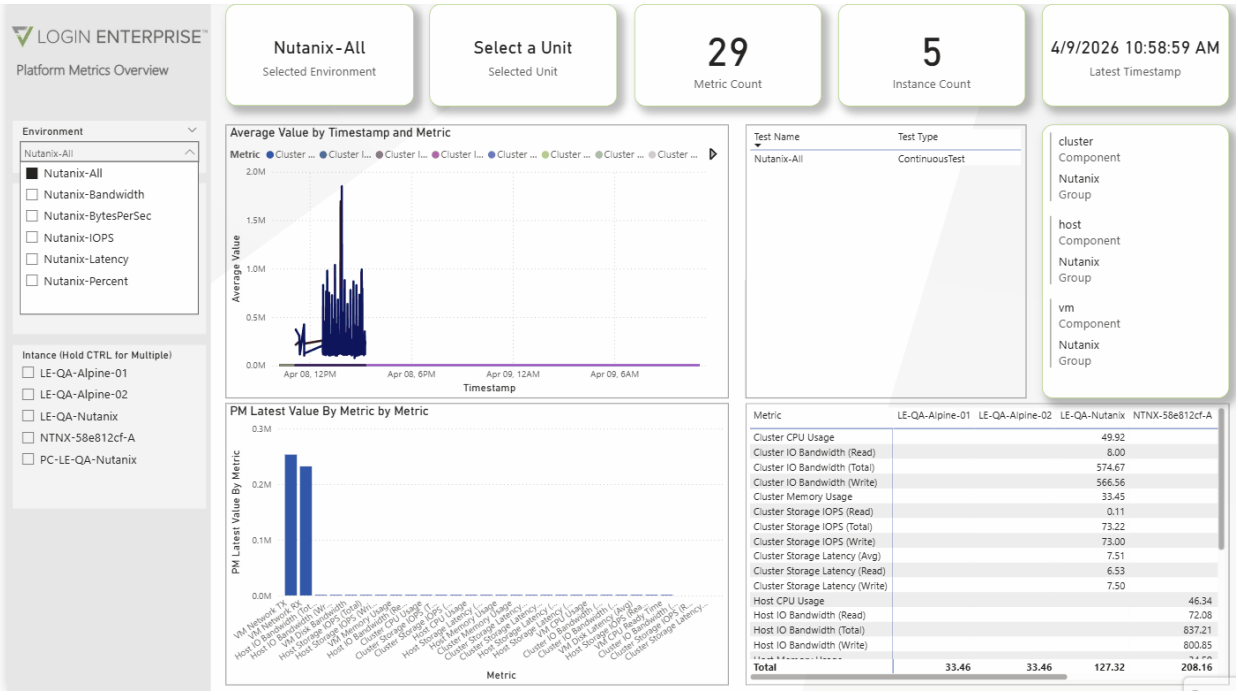
Use `-IgnoreCertificateErrors` to bypass certificate validation for the Login Enterprise appliance connection:

```
.\Nutanix-LE-PlatformMetrics.ps1 `
    -LEApplianceUrl "https://your-le-appliance.example.com" `
    -LEEnvironmentId "uuid-nutanix-all" `
    -IgnoreCertificateErrors `
    -RunOnce
```

`-IgnoreCertificateErrors` bypasses TLS certificate validation for the Login Enterprise appliance connection, which means the connection does not verify the identity of the appliance. Review your organization's security policies before using this flag in your environment. Some endpoint security solutions may also generate alerts when certificate validation is bypassed at the application level.

Power BI

A Power BI connector and report template for Login Enterprise (which includes Platform Metrics) is available in the Login VSI GitHub repo. The Login Enterprise Environment setup described in this doc works directly with the connector; all metrics land in one environment, keeping the connector configuration straightforward.



Caveats and Known Behaviors

- **VM stats require a warm-up period**: After a VM is created or powered on, Prism Central needs 5-10 minutes before the stats endpoint returns data. The script handles this gracefully.
- **VM Memory Usage at 100% for the Prism Central VM is expected**: PC pre-allocates its full memory allocation at startup.
- **Host metric count varies by run (typically 8-11 metrics)**: Some host metrics return null when there is no activity for that stat in the sampling window, e.g., no read IOPS means no read latency value. The script skips nulls and uploads only what it receives. This is correct behavior.
- **Prism Central v4 API rate limits VM stats calls**: The script adds a 500ms delay between VM calls to reduce this. On very large clusters (50+ VMs), the collection loop may exceed the 30-second polling interval. The script detects this and starts the next iteration immediately rather than compounding the delay.
- **Prism Central connectivity**: If PC is unreachable, the script detects this at startup and skips all PC-dependent work. Cluster stats continue normally.
- **Prism Central v4 API response shape varies by PC version**: The script handles differences in how Prism Central returns cluster stats data across versions automatically. No configuration is needed.

Troubleshooting

- **Host stats return 400 "startTime or endTime refer to time in future"**: Minor clock skew between the machine running the script and the Nutanix cluster. The script

automatically offsets the endTime by 30 seconds to compensate. If you see this occasionally, it is harmless.

- **Pre-flight fails with 'Prism Central: connection failed'**: All metric collection requires Prism Central. Check the PC IP or hostname and verify port 9440 access.
- **Upload timeout errors**: Usually transient. The retry logic (3 attempts, exponential backoff) handles these. If persistent, check the Login Enterprise appliance connectivity and API token validity. Uploading to a non-existent environment ID will cause the server to close the connection.
- **VM stats show 404 errors**: PC may not have collected a stats window yet for newly created VMs. Wait 5-10 minutes and try again. You can also verify the VM ExtId is valid in PC at /api/vmm/v4.0/ahv/config/vms/{extId}.
- **429 Too Many Requests on VM stats**: PC is rate-limiting consecutive VM stats calls. The script logs a warning and skips that VM for that iteration. Use -VmFilter to reduce the number of VMs being monitored.
- **VmFilter returns 0 matches**: VM names are case-sensitive and must match exactly as they appear in Prism Central. Verify the VM name in the Prism Central UI.
- **SSL/TLS or certificate errors**: The Login Enterprise appliance certificate is not trusted on the machine running the script. If the appliance uses a certificate from your organization's CA, install the root CA certificate on this machine. If the appliance uses its default self-signed certificate, re-run with `-IgnoreCertificateErrors`. See Certificate Handling.
- **Enable verbose logging for more detail**: .\Nutanix-LE-PlatformMetrics.ps1 -NutanixPassword "pass" -LEApiToken "token" ... -Verbose -RunOnce

Security

Passwords and API tokens are passed as command-line parameters at runtime and are never written to the script's log files or stored anywhere by the script. If your organization uses a secrets vault or credential manager, you can modify the script to pull credentials from that source rather than accepting them directly on the command line.

Files

File	Description
Nutanix-LE-PlatformMetrics.ps1	Collector script
Get-LEPlatformMetrics.ps1	Retrieval and export script

What's Coming

This integration today runs as a standalone PowerShell script outside of Login Enterprise. The plan is to bring it natively into the product as a supported, built-in integration, no external scripts required. When that happens, the setup experience will change, and this documentation will be updated accordingly. Keep an eye on the Login Enterprise release notes for updates.

Additional Resources

- For the Nutanix v4 API reference, see the Nutanix v4 API reference.
- For details on Nutanix PC v4 API version negotiation, refer to Nutanix v4 SDK: API Version Negotiation in Prism Central 7.5.

Launchers

The Login Enterprise Launcher acts as an endpoint that facilitates communication between the Virtual Appliance and Target environments, such as VMware and Citrix.

The Appliance supports both Linux (including IGEL) and Windows platforms. During the Test, the Launcher connects to designated Targets using specified Test credentials and manages up to 30 concurrent Test sessions. It displays a console window that mirrors the activities of the Target desktop or Application.

Once the Tests are complete and users log off, the Launcher automatically closes the console windows, terminating the connections.

- Launcher Overview and Best Practices
- Configuring the Windows Launcher
- Configuring the Linux Launcher
- Configuring the IGEL 12 Launcher
    - Launcher Connections in IGEL UMS 12
    - Deploying the Launcher through the IGEL App Portal

Launchers Overview and Best Practices

- The role of the Launcher in Login Enterprise
- What Launchers do and how they do it
- Why does Login Enterprise support different Launchers?
- What does the Launcher represent?
- Launcher groups and locations
  - Creating a Launcher group
  - Adding a Launcher to a Test
  - Adding Launchers to Locations
  - Observing Launchers and sessions in map view
- Initiating a session
- Launcher performance across different locations
- Monitoring latency
  - Continuous Testing scenario
  - Application Testing and Load Testing scenarios
- Lifecycle of the Launcher
- Handling account credentials on the Launcher
- Best practices
  - Important considerations
  - Tips and tricks
  - Restarting the Launcher
- Related terms and concepts
- Additional resources

The role of the Launcher in Login Enterprise

The Login Enterprise Launcher is a host used as an endpoint to communicate between the Login Enterprise virtual appliance running a Test and the targets (VMware, Citrix, etc.). We support Linux (including IGEL, which runs Linux) and Windows.

To learn where the Launchers architecturally reside in the Login Enterprise setup, see System Architecture and Components.

What Launchers do and how they do it

During a Login Enterprise Test, the Test configuration directs the Launcher to establish a connection to a designated target environment. The Launcher receives instructions on what to invoke, including which credentials (Test user/password) to employ for connecting to the specified resource through a broker. Then, the Launcher displays a window, commonly referred to as a console, depicting the appearance and activities of the target desktop or published application throughout the Test. Launchers can initiate and manage up to 30 concurrent target Test sessions. Upon completion of their respective defined Tests and subsequent logoff by the Test Users, the console windows associated with the targets will automatically close, terminating the connection.

Why does Login Enterprise support different Launchers?

Login Enterprise performs equally well regardless of whether the Launcher software is installed on Windows or Linux machines; there's no discernible advantage between the two platforms.

While Microsoft Windows remains the preferred choice among our customers, it typically entails higher licensing costs compared to Linux. Linux often provides free or more affordable licensing options, and some of our clients either have existing infrastructure tailored for Linux or simply prefer Linux endpoint hosts due to experience, contractual agreements, or security policy.

Additionally, we provide support for IGEL endpoint clients through our partnership with them. For more information, see IGEL 12 Launcher.

What does the Launcher represent?

The Launcher functions as the physical access point for virtualized clients, acting as an endpoint. Therefore, it's crucial to consider the Launcher's location based on testing objectives to ensure accurate round-trip latency between the Launcher and the target (EUC client). Additionally, verifying if the Launcher endpoint has access to the target is essential, as this can highlight potential production downtime requiring attention.

An endpoint host refers to a specific network address or server enabling connectivity for a software application, service, or device to communicate with a target. This target could be another application, service, or device that the endpoint host needs to interact with.

For example, within a client-server architecture, the endpoint host typically represents the server where clients connect to access resources or services. In a peer-to-peer network, an endpoint host could be any participating node that communicates directly with other nodes.

Launcher groups and locations

To launch Test sessions, you need to add Launchers to the Launcher Group.

Launcher Groups specify which Launchers should be used for a particular Test. Creating a Test requires at least one Launcher Group, so having at least one Launcher Group is a requirement. This is helpful when trying to logically divide Launchers, for example, based on geographic location.

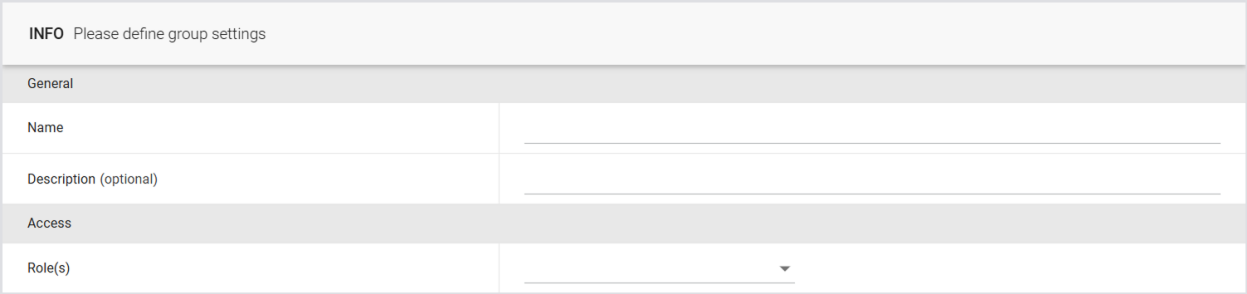
For large environments, creating or managing a large number of Launchers can be a manually intensive process. To automate many operations, such as creating Launcher groups, adding Launchers to a Launcher Group, or assigning Launcher Groups to a Test, use the API along with PowerShell.

Creating a Launcher group

1. In **Launchers > Groups**, click the green "+" on the right to add a group.

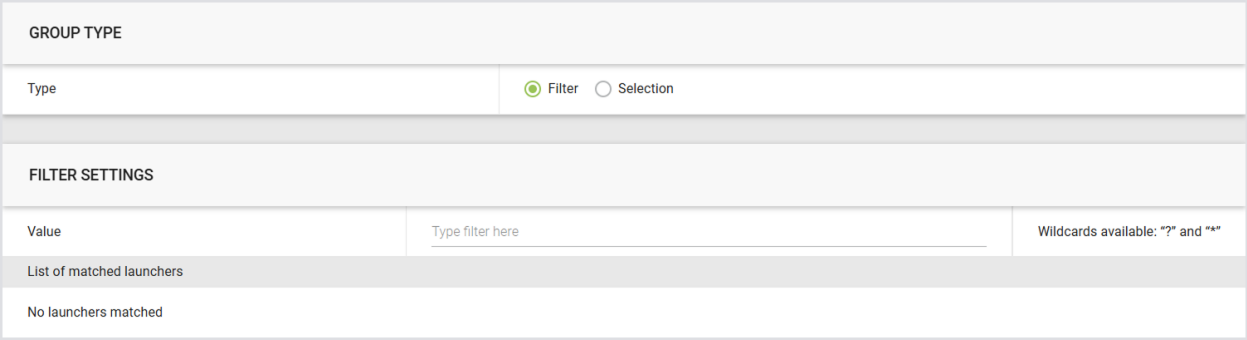
2. Define the Group settings:

General	
Name	A designated name for the Group.
Description (optional)	A description specifying the Group's purpose.
Access	
Roles	The roles you'd like to assign to this Group. For details, see Role-Based Access Control (RBAC).

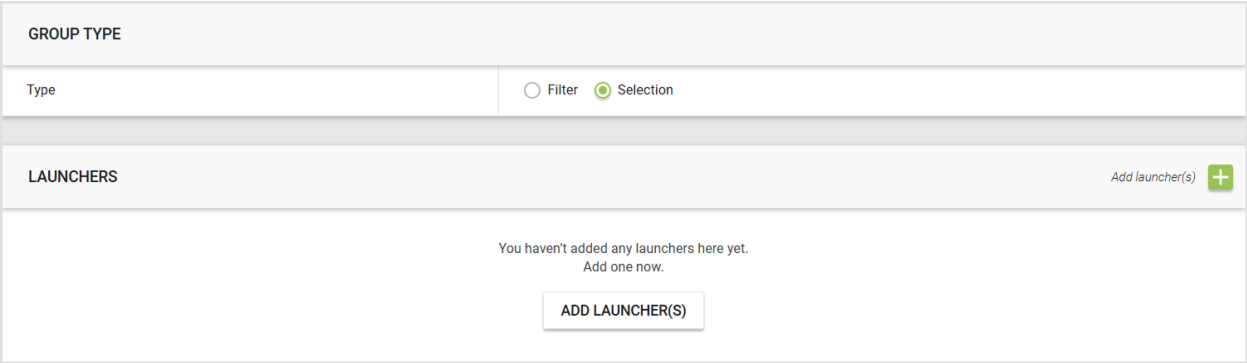


3. In the **Group Type**, select **Filter** or **Selection**.

- **Filter** lets you fill out the details and the Launchers you want to add to the group. Here, you can use two wildcards:
  - **? (Question mark)**: Represents a single-character wildcard.
  - *** (Asterisk)**: Represents a multi-character wildcard.



- **Selection** lets you select Launchers to add to the new group manually.



NOTE: If you add a Launcher to a Launcher Group by Selection, i.e., you choose it specifically and not by wildcard, and that Launcher goes offline, it will still be part of the Launcher Group. It will not show up in the Launchers list, and it will not be available for Tests, but if the Launcher comes back online, it will immediately be available in that Group and any Tests that include that Launcher Group.

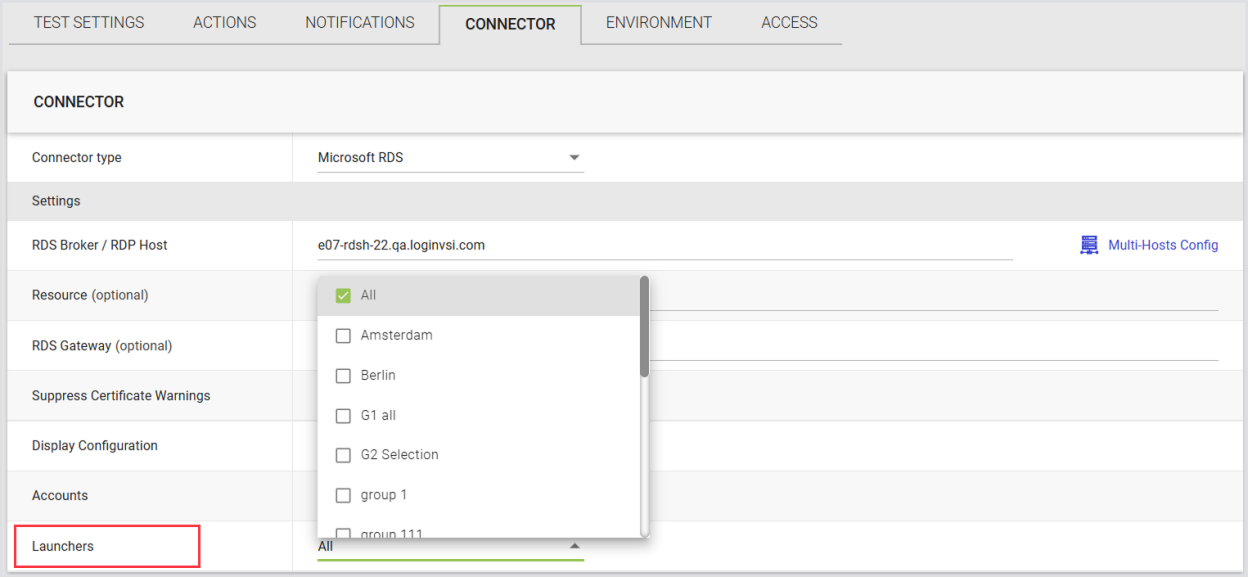
You have successfully created the Launcher Group. These Launchers are now able to successfully receive and execute new Test connections to the Target EUC stacks and platforms. The next step is to add Launchers to a Test.

Adding a Launcher to a Test

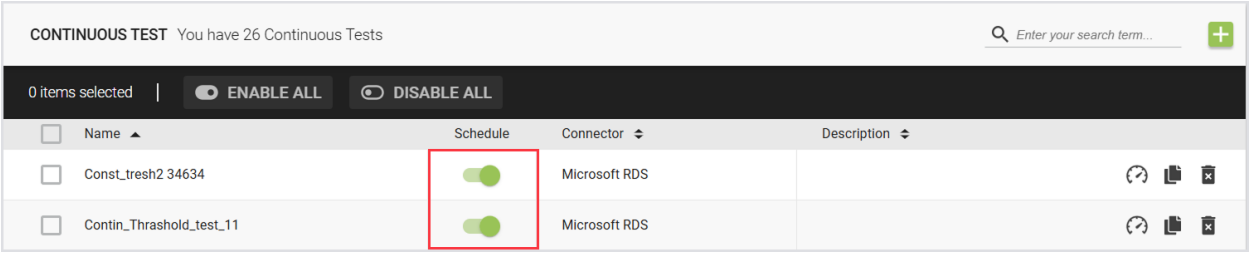
Launchers write a log file at %TEMP%\LoginEnterprise\Launcher\Logs. The log will be created as soon as the Launcher starts and will record successes and errors in its operations. The Launcher also displays a table in its window showing the Tests that have been assigned to it since it started this current session. This list is kept in memory, so it is lost on every program restart.

To add a Launcher to a Test:

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Manage tests**, and select the Test, e.g., Continuous Test.
2. Click on the Test instance, and then select **Connectors** from the tab menu.
3. In **Connectors**, select the checkbox with the Launcher you want to add to the Test, and click **Save** to apply the changes.



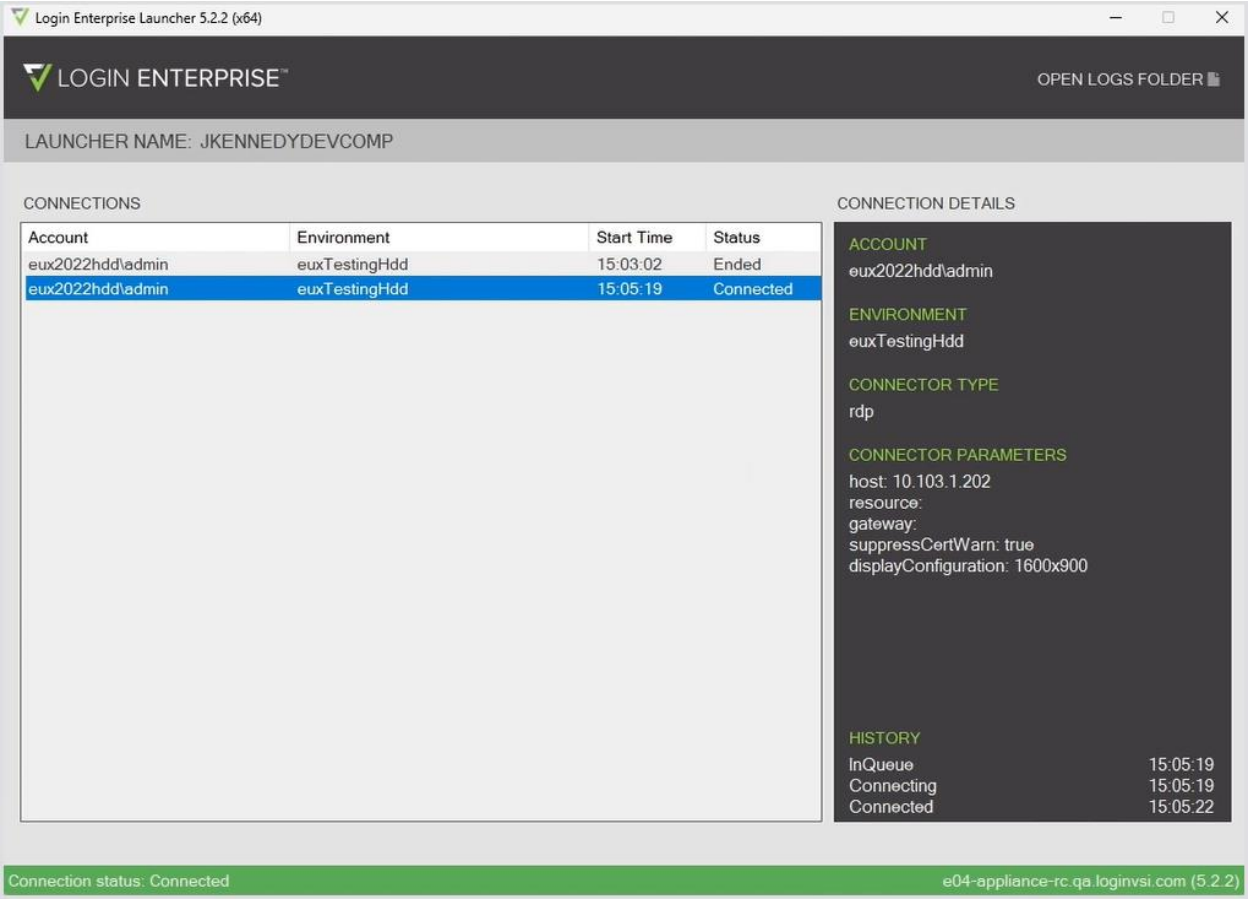
5. Go back to the **Continuous Tests** page.
6. Next to each Test instance, in **Schedule**, select the toggle so that it's turned on.



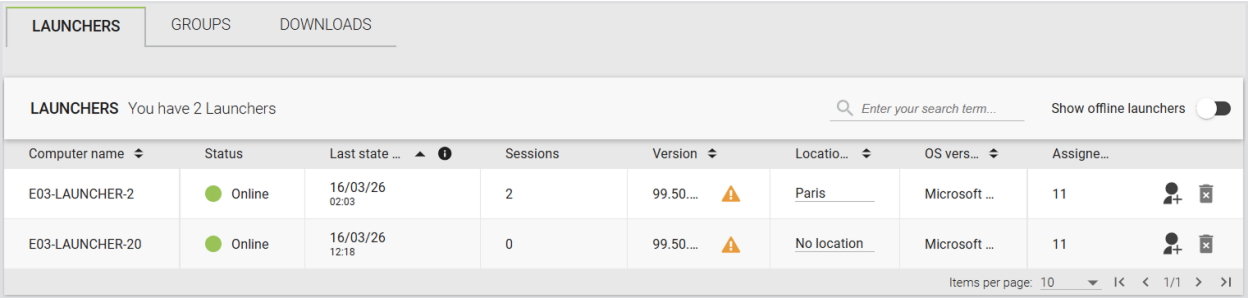
When the Tests are turned on, you can now:

a. See the target session is being connected to in a console window:

b. The Launcher application shows the connection was made, and the connection information in the Launcher is visible.



c. You can see the connection sessions in the **Launchers > Sessions**.



When this particular Test ends, it will start a new Test via the Launcher application. The Launcher operation is autonomous and needs no human intervention.

Adding Launchers to Locations

The Launchers can be added to defined locations. This is to more closely represent successful connections and latency that real-life users would experience in, for example, an office branch. Having a launcher set up in each location needing to be monitored is a best practice. However, if you narrow down your selection to one location that's not connecting to targets successfully, targeted troubleshooting can occur.

TIP: In a Load Test scenario, the best practice ratio for simultaneous Test connections per Launcher is 30 to 1. So, if running a 100-user Load Test, you need 4 Launcher hosts.

If Launchers are attached to configured locations when a Continuous Test type is in progress, these Launchers will be visible in the map view (see the section below).

Observing Launchers and sessions in map view

For details, see the Location View section in the Test Dashboard.

Initiating a session

When the Login Enterprise appliance wants to initiate a session, it checks the Test for the Launcher Group and Account Group, picks an available one from each, and sends a message to that Launcher. The Launcher receives the session information and Connector configuration and begins the process of trying to launch a login for the user through the configured Connector.

It determines whether the connection was successful or not by looking for specific events and return codes, and then alerts the Login Enterprise appliance. It will also report if the session terminates unexpectedly. The Target is responsible for reporting back to the Login Enterprise appliance that the session login is completed. All the Launcher can say is whether the process is still running. If the Target has not checked in with the appliance within 5 minutes (configurable), the Launcher will terminate the session and the appliance will mark it as failed.

Once the Target checks in, the appliance tells the Launcher and the Launcher turns off that 5-minute timer and switches to simply monitoring the status of the remote session client's process ID. It will report to the appliance if the session terminates itself. The appliance will tell the Launcher if the Target has reported completion and thus that the client is expected to terminate itself. So, between the appliance and the Launcher and the Engine, they know when a process starts, if it terminates unexpectedly, and when the termination is expected.

Launcher performance across different locations

Monitoring latency

Login Enterprise Launchers play a crucial role in capturing round-trip latency metrics between the Launcher endpoint host and the virtualized target. The target represents the location where the actual workload for Login Enterprise performance testing occurs. This latency metric quantifies the delay experienced by a user when accessing the virtualized target. For example, they measure the duration it takes for a user's action, such as clicking on an item in the virtualized desktop, to be registered. By collecting these metrics, Login Enterprise offers insights into potential latency fluctuations throughout a workday. Moreover, administrators can set a latency threshold. If this latency is exceeded, Login Enterprise can generate alerts, enabling prompt action to address any performance issues.

Continuous Testing scenario

For Continuous Testing, which typically involves testing against production infrastructure, it's crucial to deploy a Launcher in each physical location to emulate real users accessing virtualized resources. When two locations access two different published resources, this creates a multiplication effect, necessitating the configuration of four distinct Test scenarios.

Application Testing and Load Testing scenarios

While Continuous Testing typically focuses on production infrastructure, the positioning of Launcher endpoint hosts is equally crucial in Application and Load Testing, depending on the testing objectives and setup.

For example, during a Load Test conducted outside of regular work hours, when actual users aren't accessing the target EUC infrastructure (which remains in a production state), strategically placing the Launchers to mimic real user access can help identify any potential issues with the Launchers themselves before they impact production, an outcome best avoided.

Conversely, if the primary goal of the Load Test is to stress the target environment and evaluate its performance, with less emphasis on stressing the endpoints, precise representation of where actual users access the targets may be less critical.

In Application Testing, selecting real user locations can offer benefits, despite being a single-user Test. For instance, when validating an image before scaling up for a Load Test, utilizing a Launcher in the desired location representing an endpoint accessing a target resource can facilitate early detection and resolution of any connection issues.

Lifecycle of the Launcher

1. **Setup Launcher host**: Install the Launcher software on all Launcher hosts, regardless of whether they run on Windows or Linux. Each Launcher, configured with a minimum of 2 vCPUs and 4GB of memory, is capable of concurrently initiating up to 30 target sessions.
2. **Install Launcher software**:
    - Maintain multiple Launchers to distribute workload and ensure redundancy from a capacity perspective. For more information, see Launcher Groups and Locations.
    - Use descriptive naming conventions for Tests, Launchers, Groups, and Accounts to enhance traceability and visibility. This ensures they accurately represent the physical locations where actual users access the endpoints connecting to the target infrastructure. Subsequently, the Launchers will be viewable and plotted on a world map within Login Enterprise UI, providing comprehensive information about connection success and performance for each location. For more information, see Locations.
    - Install endpoint software, such as VMware's or Citrix's, on the Launchers.
    - Avoid domain joining for Launchers to maintain isolation, unless necessary due to organizational security policies.

- Implement least privilege principles by avoiding the use of administrative accounts. Instead, use standard user accounts for domain users, ensuring membership in the local machine's user group. However, ensure compatibility with the Login Enterprise Launcher software, allowing seamless operation and connection establishment via the connection clients. Maintain updated endpoint client software on the Launchers, ensuring alignment with the protocols and standards used by actual endpoints accessed by users.

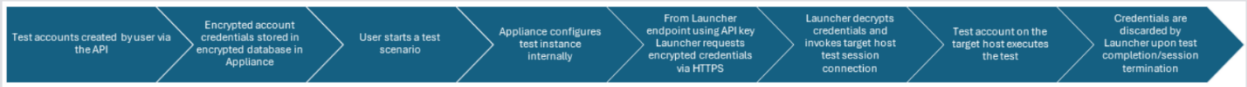
3. **Run the software and keep running**:
    - Disable power-saving modes to ensure continuous availability and performance consistency.
    - The Launcher software on the Launcher hosts can be configured as a startup item. This ensures that upon reboot, the Launcher will automatically commence running once logged into the system.
    - Before initiating significant or large-scale Tests, we recommend conducting dry runs with a few users per Launcher. This step ensures that all settings are configured correctly and validates the Launcher's capability to handle concurrent sessions successfully.
4. **Periodically restart the Launcher host for optimal performance**:
    - Monitor system resource utilization, ensuring CPU and memory stay within acceptable thresholds (e.g., 85% CPU utilization, 80% memory utilization).
    - To mitigate performance degradation caused by prolonged uptime, especially during heavy utilization like repeated Load Tests, it's beneficial to periodically reboot Launchers. Setting up a Windows scheduled task or implementing automation for rebooting Launchers can alleviate this issue. Since not all Launchers share identical configurations or run the same policies, there's no fixed requirement for the reboot schedule. However, based on our observations with customers, a weekly reboot schedule has proven effective. You can restart the Launcher in the UI. For details, see Restarting the Launcher.
    - To prevent false positive alerts regarding Launcher unavailability, ensure that in Continuous Testing configurations, testing is disabled during the scheduled reboot. This ensures that any downtime associated with the reboot doesn't trigger unnecessary alerts. For more information, see the Daily and weekly schedules.
5. **Update the Launcher upon new Login Enterprise releases**: When setting up a new version of Login Enterprise, you should install the new Launcher version software over the existing version. While this is the recommended best practice, it is not mandatory unless the version difference is significant (a major update). In such cases, failing to update may prevent the Launchers from initiating sessions until they are updated to the latest version.

NOTE: There is no standalone upgrade process for the Launcher. Running the new version installer on a system with an existing Launcher will behave like a fresh installation. The user interface and installation flow will not differ, so be prepared for the installation process to appear as if you are installing the Launcher for the first time.

6. After the last step, go back to run the Launcher software and keep running; Rinse and repeat.

Handling account credentials on the Launcher

1. **API key and credential transmission**:
    - Login Enterprise hosts a specific and secured API endpoint for the Launcher. The Launcher uses an API key to authenticate against this endpoint. When a Test is running, the Launcher uses the API key to request encrypted Test account credentials. Login Enterprise then sends these encrypted credentials to the Launcher via HTTPS. The Launcher proceeds with using and discarding the credentials as outlined below.
    - Login Enterprise encodes and sends account credentials through this secure channel to the Launcher.
    - Encrypted account credentials are stored in an encrypted database in Login Enterprise. Passwords are stored utilizing AES-256-bit encryption and are FIPS-compliant. For more information, see Encryption.
2. **Credential management**:
    - When a Login Enterprise Test starts, it determines the defined Test account(s) to use for the Test. These credentials, which were previously configured in the Login Enterprise database by the user or administrator (either through the Public API or Login Enterprise web interface), are encrypted and secured, ensuring that none of the content is visible. For more information, see Accounts.
    - The stored Test account credentials are only accessible via a specific endpoint, managed by a Launcher-specific API. This endpoint is authorized solely for the Launcher; nothing else is allowed to interact with it.
3. **Credential handling by the Launcher**:
    - The Launcher receives an encrypted payload with the credentials from Login Enterprise and performs the following actions:
        - **Usage**: The Launcher uses the credentials to invoke Test sessions based on the defined Test scenarios that Login Enterprise is running. This involves using the Test account(s) credentials to perform the necessary actions as specified in the Test scenarios.
        - **Discarding**: After the authentication phase (including retries), the Launcher discards the credentials.
    - The Launcher does not save or cache credentials. After the Test is completed, the Launcher will terminate and dispose of any information that was only in memory.
4. **Session management**:
    - For each new session, from the Launcher endpoint using the API key, the Launcher requests and receives the encrypted credentials from Login Enterprise via HTTPS. Then, the Launcher uses and discards the credentials.
    - Any time a Launcher is started, it will be a new instance of the Launcher, with previous instances having already been disposed of.
5. **Additional considerations**:
    - In the Microsoft RDP scenario, Microsoft RDP may save a .rdp file with connection details that can include an encrypted password string using Microsoft's logic.
6. No caching or persistence of credentials occurs in the Launcher beyond the duration of the session.



Best practices

1. Before running a Test on a Launcher, verify that you can manually invoke a connection to the target. To do this:
    1. **Access the target gateway**: On the Launcher, navigate to the web front end of your target gateway. Depending on your setup, this may involve accessing a web front end or using a different method such as the RDS client (mstsc) for Remote Desktop Services. Ensure you are familiar with the specific access method required for your environment to verify a successful connection.
    2. **Log in**: Use a Test user account to log in.
    3. **Open the published resource**: Attempt to manually open the published resource.
    4. **Evaluate the result**: Check if the connection is successful.

If the connection works manually, proceed with using the Launcher. If not, troubleshoot the issue. Common problems might include insufficient Citrix licenses, downed bare-metal servers hosting the target desktop, etc.

NOTE: This manual verification is crucial for both initial setups and diagnosing sudden connection failures. It helps identify underlying issues that could be affecting the Launcher's performance. This best practice applies to all Launchers, regardless of the operating system (Windows, Linux, or IGEL).

2. Utilize thin clients over laptops for optimal performance and manageability. However, note that you might want to use a laptop/desktop in an office as a Launcher to simulate an endpoint connecting to a target. The idea is that in the office you're using laptops/desktops to connect to remote targets. So this Launcher "model" would be more accurate with overall uptime representation.
3. During a Load Test, Login Enterprise will restrict the usage of the Launchers designated for the Test from participating in other testing scenarios. This implies that for the duration of the Load Test, which may extend to several hours if instructed, the designated Launchers will be unavailable for launching Continuous or Application Tests, even if these Tests remain enabled or initiated during the Load Test. Upon completion of the Load Test, the Launchers will automatically resume other assigned testing tasks. Therefore, it is advantageous to allocate dedicated Launchers for Load, Application, and Continuous testing, and organize them within Login Enterprise Launcher Groups. For more information, see Launcher Groups and Locations.
4. Automate Launcher deployment and configuration to ensure consistency and efficiency.
5. Standardize the process of installing the Launcher through proceduralization and automation.

Important considerations

1. **Dedicated use for Load Tests**: Launchers running Load Tests cannot launch other sessions during the Test. This is by design.

TIP: Avoid using Launchers for mixed testing scenarios to ensure accurate Load Testing results.

2. **Session launch rate**
    1. A Launcher will only initiate a new session every 20 seconds by default.
    2. If multiple Continuous Tests are enabled simultaneously and they attempt to launch sessions on the same Launcher, the Launcher will still adhere to the 20-second interval.
    3. Additional sessions will enter a scheduling queue, waiting to be launched sequentially.

3. **Load Test scenario ratio**: In a Load Test scenario, the best practice ratio for simultaneous Test connections per Launcher is 30 to 1. So, if running a 100-user Load Test, you need 4 Launcher hosts.

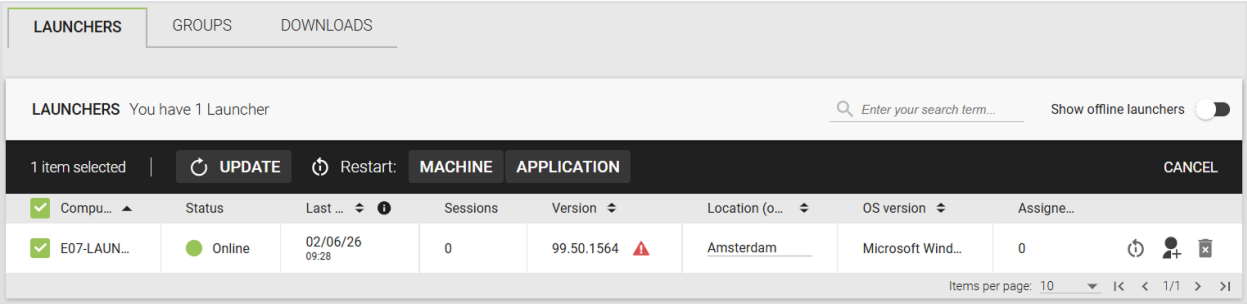
Tips and tricks

1. Putting the Launcher shortcut in the Launcher host's startup folder can make getting the Launcher starting easier in case the Launcher needs to be restarted.
2. Pinning the Launcher to the Windows taskbar can make it easier to start the Launcher application if it's ever closed.
3. If you're using the Login Enterprise Launcher (Windows-based), the Launcher can be installed over a network (see Silent installation). This lets you deploy multiple Launchers easily.

Restarting the Launcher

You can restart the Launcher in the **Login Enterprise sidebar menu** > **Configuration** > **Launchers** or using the Public API. There are two restart options available:

Option	Explanation	Details
Restart Launcher machine	Restarts the machine where the Launcher is running.	The Launcher will stay offline after machine restart until a Windows user signs in. Make sure that autologon is configured on the Launcher machine to avoid manual recovery. To learn more, see the Microsoft documentation.
Restart Launcher application	Restarts the Launcher application.	-



NOTE: A restart cannot be done when:

- Launcher is offline.
- Launcher has active sessions.
- Launchers are on versions older than 6.7.

Related terms and concepts

Some other related terms and concepts that you'll find in this article include:

- **Account**: a username/password pair used to trigger connections and identify connections once they start (usually an AD account).
- **Appliance**: a Linux-based Login Enterprise control center.
- **Connector**: a Login Enterprise software used to interact with a Connection Broker to launch a remote connection to a target desktop. Also, a Connector may relate to VMware connection and custom connections that Login Enterprise supports. For more information on Connectors and connection options, see Connectors.
- **Target**: an individual end-user machine where a Test will run.
- **Test**: a collection of Application scripts that will perform a series of actions and take measurements as defined in those Application scripts. Tests also include taking other metrics outside the application scripts (workloads), such as Protocol latency, Logon times, EUX measurements, and Session Metrics.

Configuring the Windows Launcher

- Introduction
- Downloading and installing the Launcher
  - Installation directory
  - Silent installation
- Running, operating, and starting the Launcher
  - Ensuring proper connection
  - Accessing the Launcher
  - Enabling remote connectivity to the Launcher
  - Enabling verbose logging for the Launcher
- Maintenance and updates
- Updating the Launcher
- Starting the Launcher
  - Manually
  - Automatically
    - Option 1: Using the Startup Folder
    - Option 2: Using a Workspace Management tool
    - Option 3: Configuring automatic logon in Windows
    - Option 4: Creating an executable for automatic logon
  - Launcher user interface
  - Running the Launcher in console mode
- Managing access to Launcher and Launcher Groups
- Troubleshooting and best practices
  - Certificate update
  - Connection status
    - Connection failure overview
    - Diagnosing connection failure
  - Versioning notice
  - Launcher process issues
  - Best practices
- Deployment and Compatibility
  - Physical Launcher setup
  - IT environment Launcher implementation
  - Launcher requirements
- Additional resources

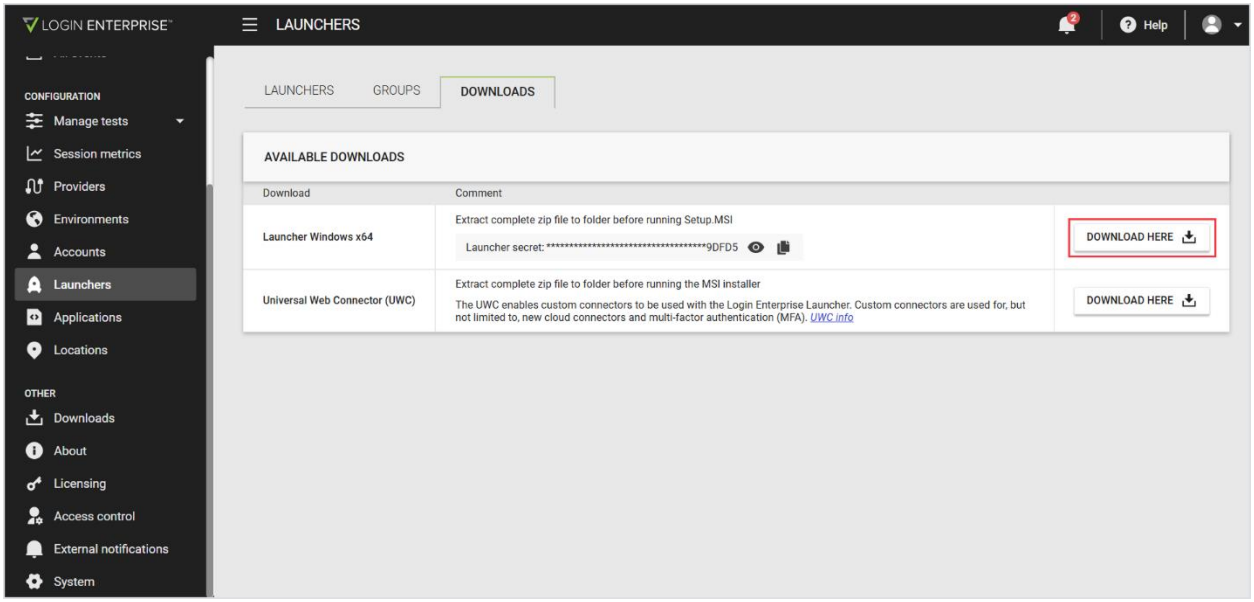
Introduction

To learn about the Launcher's role in Login Enterprise, what it does, how it works, and more, see Launcher Overview and Best Practices.

For details on where the Launchers architecturally reside in the Login Enterprise setup, see the System Architecture and Components.

Downloading and installing the Login Enterprise Windows Launcher

1. Log in to your Login Enterprise with the Launcher intended to be the host.
2. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Launchers**.
3. In **Launchers**, select **Downloads** from the tab bar menu and download the file.



TIP: Alternatively, the Launcher can be downloaded from the sidebar menu under **Other > Downloads > Launchers**.

4. Once downloaded, extract the Launcher zip file.
5. Start the Launcher installation by running the Setup.msi file, following the prompts.

The Launcher will launch by default at the end of the installation.

Installation directory

You can save the extracted Launcher installation directory on a shared network location. This way, you can access it quickly without downloading and extracting it each time you set up a new Launcher.

Silent installation

You can use command-line functions to simplify the deployment and update processes for the Launcher installation.

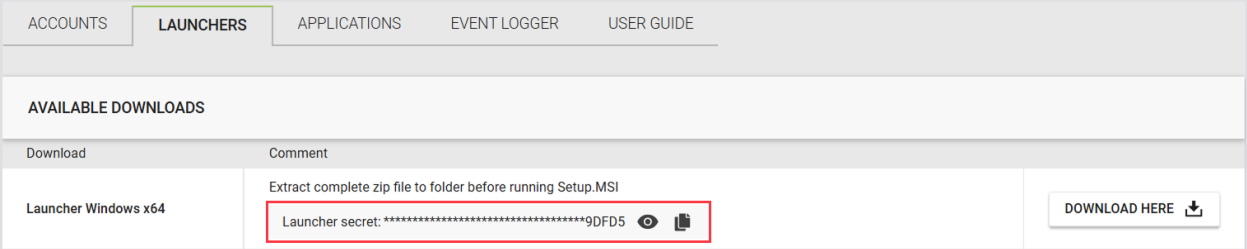
To run the MSI file silently, use the following parameters. Be sure to replace `<URL>` and `<secret>` with your specific values:

```
msiexec /I "<full path to Setup.msi>"/qb! serverurl=<FQDN of appliance>
secret=<secret>
```

For example:

```
msiexec /I "\\fileserver\share\Setup.msi"/qb!
serverurl=https://loginvsi.demo.com
secret=E0733F0B555035C7E377CA8797530855770141C1
```

Copy the secret from the **Downloads > Launcher secret**:



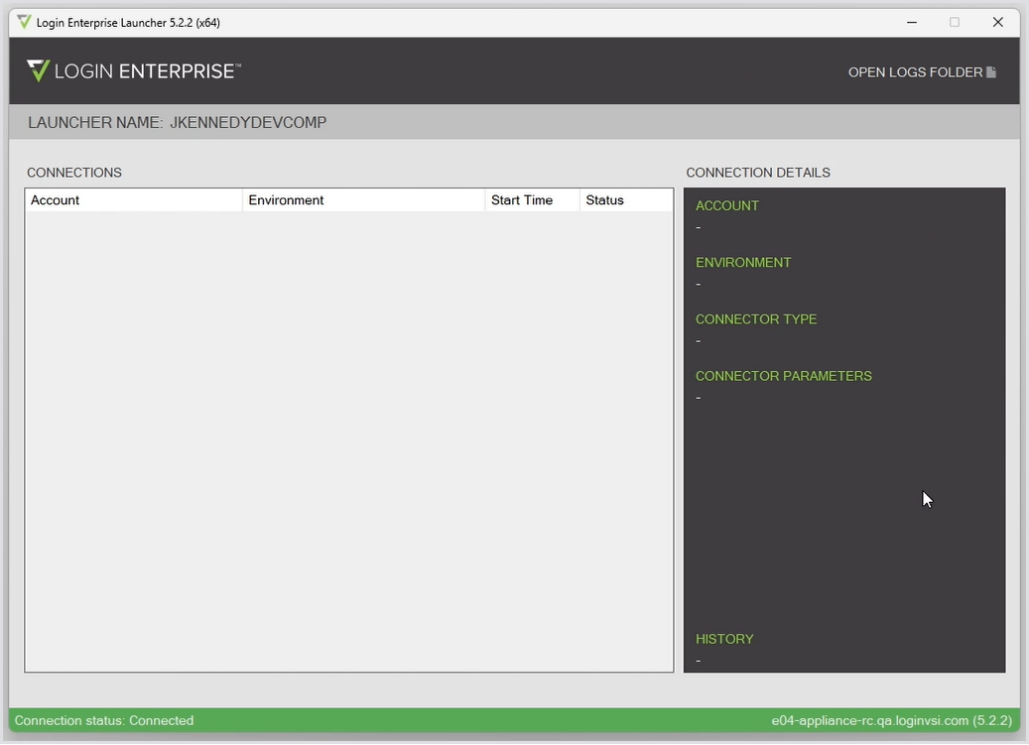
Using silent installation for the Launchers allows administrators to streamline the deployment process, reduce potential human error, and ensure consistent configurations across all systems.

TIP: Silent installation supports automated, programmatic deployment.

Running, operating, and starting the Login Enterprise Windows Launcher

Ensuring proper connection

When you run the Launcher, check the bottom toolbar of the UI. It will indicate if the Launcher is successfully connected to the appliance from which the installer was originally downloaded.



Accessing the Login Enterprise Windows Launcher

Enabling remote connectivity to the Launcher

If you are using a physical Launcher machine, we recommend enabling remote connectivity. Options include Remote Desktop Protocol (RDP), TeamViewer, Virtual Network Computing (VNC), or other remote access applications. Remote access simplifies troubleshooting when necessary.

Enabling verbose logging for the Launcher

To enable more detailed logging for the Launcher, modify the `appsettings.json` file located in the installation directory of the Launcher. By default, this path is `C:\Program Files\Login VSI\Login PI 3 Launcher\appsettings.json`

In the `appsettings.json` file, change the value of "Default" from **Information** to **Verbose**.

Maintenance and updates

For the optimal performance of the Launcher, follow these best practices

1. **Restart the Launcher periodically**.
2. **Have the Launcher run automatically upon startup**.
3. **Update the Launcher with each Login Enterprise release**.

- To ensure compatibility with each Login Enterprise release, you must manually install the new version of the Launcher. For more information, see Updating the Launcher.
- After installing a new version of the Virtual Appliance, navigate to **Launchers** in the **Login Enterprise sidebar menu**. Download the updated version and install it over the existing Launcher.

The installation of the new version will overwrite the previous version without requiring any additional steps. For details, see Launcher life cycle and best practices.

Updating the Launcher

You can update Launchers from the Login Enterprise UI or by using the API. This feature is useful when managing multiple Launchers.

IMPORTANT: The new UI and Public API Launcher upgrade feature in v6.6 is only compatible with Windows-based Launchers already running v6.6 or higher.

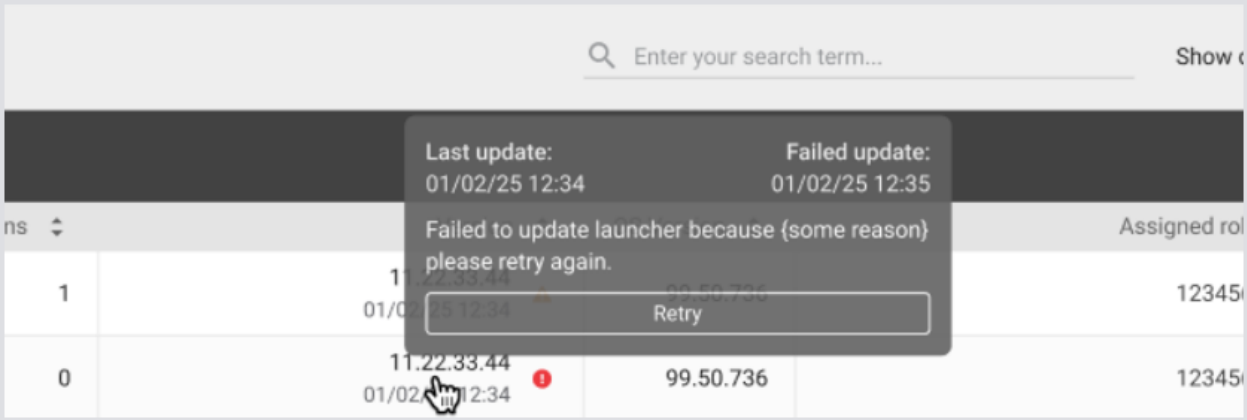
To use the new update method, you must update each Launcher one last time using the old method. After that, the new update functionality becomes available.

Update scenarios

- Launcher is up to date.
- A Launcher update is available.
- A Launcher update is available, and the previous update attempt failed.
- A Launcher update is available, but the update is blocked due to one of the following:
  - The Launcher is offline.
  - Active sessions are in progress.
  - The Launcher version is v6.5 or lower.

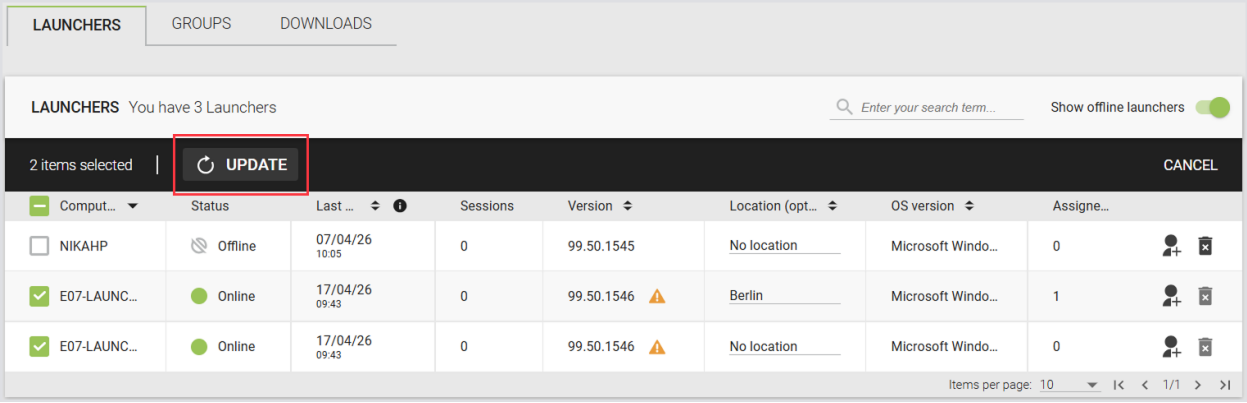
Example: Failed update attempt

The following example shows a Launcher with an available update where the previous attempt failed:

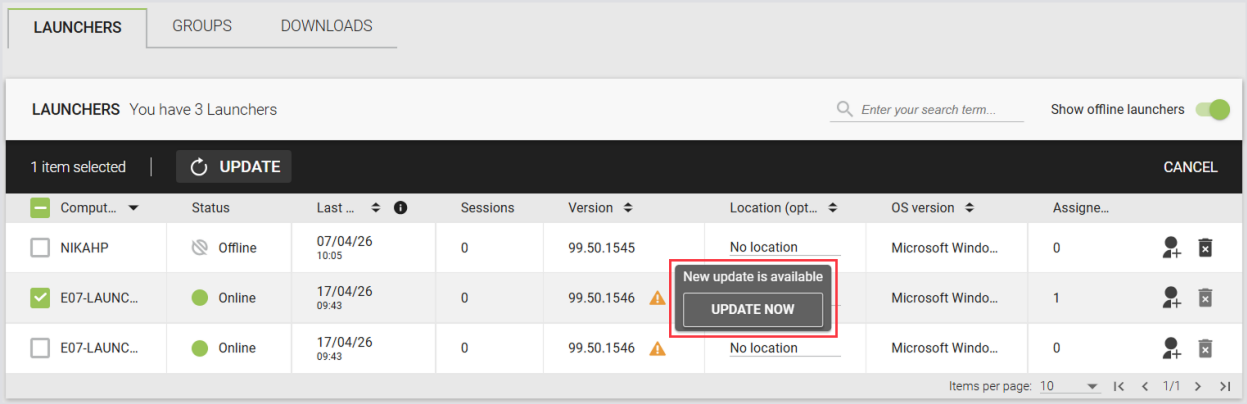


You can update the Launcher in 2 ways:

1. **Using multi-select in the table**
    1. Select one or more Launchers.
    2. If none of the selected Launchers can be updated, the **Update** button is disabled. A tooltip explains why.
    3. If at least one selected Launcher can be updated, the **Update** button is enabled.
    4. When you select **Update**, only eligible Launchers are updated. Launchers that cannot be updated are skipped.



2. **Using the status icon**
    1. Hover over the warning icon for a Launcher.
    2. Click **Update now** from the tooltip.



NOTE: The old update method remains available. It is required for Launchers running on older v6.5 or lower.

Starting the Login Enterprise Windows Launcher

Manually starting the Launcher

By default, the installation creates a desktop shortcut for the launcher. To manually run the Launcher, double-click the shortcut on your desktop. The shortcut points to the installation path: "C:\Program Files\Login VSI\Login Enterprise Launcher\LoginEnterprise.Launcher.UI.exe".

Automatically starting the Launcher

You can configure the Launcher to start automatically when Windows starts. There are several methods to achieve this:

Option 1: Using the Startup Folder

1. **Copy the Launcher shortcut**: After installation, the shortcut is located on the desktop.
2. **Open the Startup folder**:
   - Press `Win + R` to open the Run dialog.
   - Type `shell:startup` (for the current user) or `shell:common startup` (for all users) and press Enter.
3. **Paste the Shortcut**: Paste the copied launcher shortcut into the Startup folder. This ensures the launcher starts automatically when a user logs in.

Option 2: Using a Workspace Management tool

For domain users or domain-attached launchers, consider using workspace management tools such as VMware UEM or Ivanti:

1. **Integrate with Workspace Management tool**:
   - Attach a login action to the launcher device or user.

- Configure the tool to execute the Launcher executable found in the installation directory (by default located at "C:\Program Files\Login VSI\Login Enterprise Launcher\LoginEnterprise.Launcher.UI.exe").

Option 3: Configuring automatic logon in Windows

To streamline operations and reduce manual intervention, you can enable automatic login after scheduled restarts by configuring specific registry keys:

- **Registry Location**: HKLM:\Software\Microsoft\Windows NT\CurrentVersion\Winlogon
- **Registry Keys**:
  - **AutoAdminLogon**
    - Type: REG_SZ
    - Value: 1 (enabled) or 0 (disabled)
    - Enables automatic administrator logon functionality.
  - **DefaultDomainName**
    - Type: REG_SZ
    - Value: Your domain name (if applicable)
    - Specifies the domain name for the user.
  - **DefaultPassword**
    - Type: REG_SZ
    - Value: The user's password
    - Sets the password for auto-login.
  - **DefaultUserName**
    - Type: REG_SZ
    - Value: Username
    - Specifies the username for auto-login.
  - **AutoLogonCount**
    - Type: REG_SZ
    - Value: 1 (once), 2 (twice), etc.
    - Configures the number of times the machine stores the configuration.

Ensure the following registry keys are disabled for the above settings to work:

- **Registry Location**: HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon
- **DontDisplayLastUsername**
  - Type: REG_SZ
  - Value: 0 (disabled) or 1 (enabled)
  - Controls the display of the username on the logon screen.
- **Registry Location**:
  - HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
- **DisableAutomaticRestartSignOn**
  - Type: REG_DWORD
  - Value: 0 (disabled) or 1 (enabled)
  - Enables or disables automatic sign-on for users.

- **Using Sysinternals "Autologon" tool**
    - The Sysinternals section of Microsoft support offers a tool called "Autologon" which simplifies the process of automating the autologon feature on a machine. To utilize the tool, follow these steps:
    - Visit the Autologon v.3.10 provided by Microsoft support to access detailed instructions on downloading and using the "Autologon" tool.
    - Download the "Autologon" tool from the provided link.
    - Follow the instructions provided in the guide to configure autologon for your machine.

Option 4: Creating an executable for automatic logon

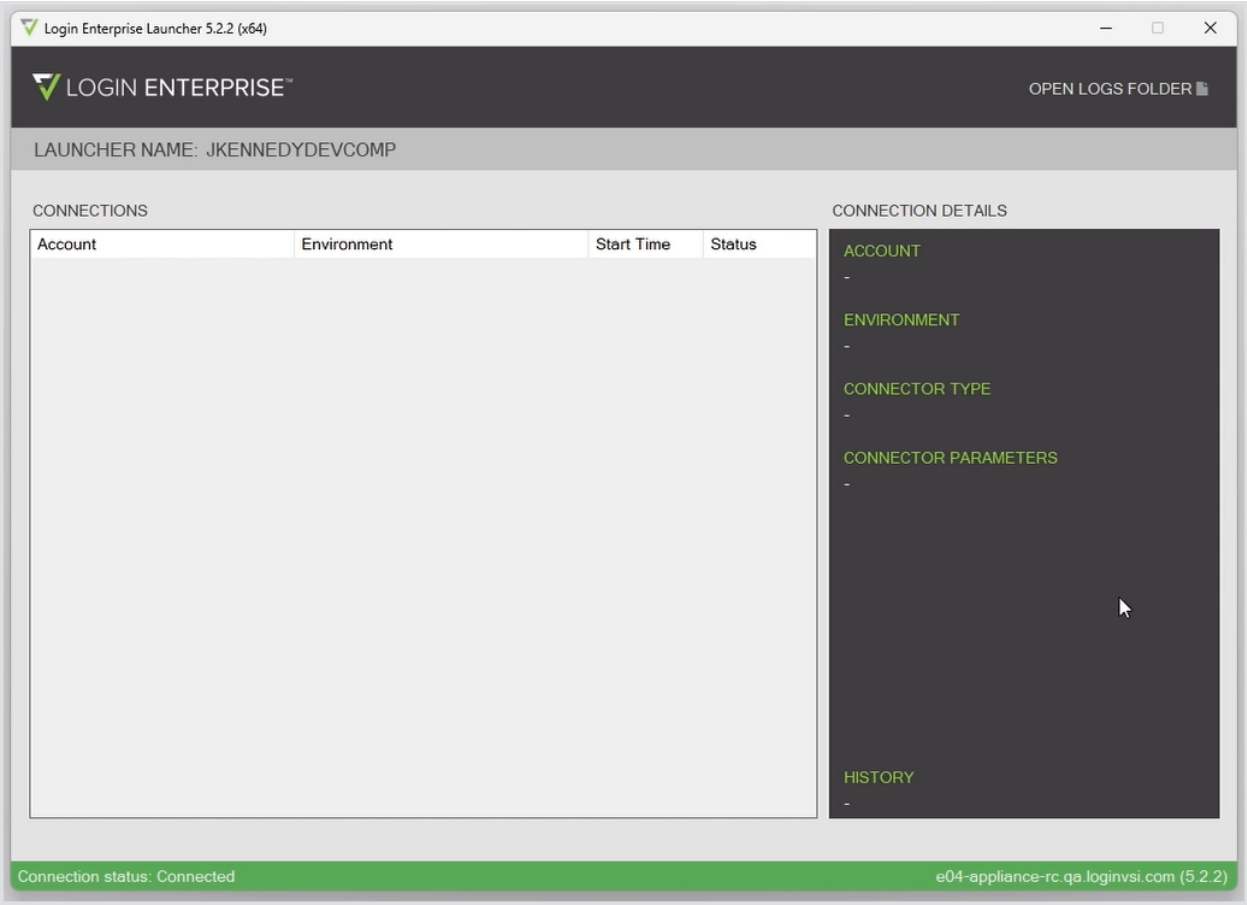
For a successful and secure implementation, you can create an executable using a tool like AutoIT. This executable will be executed before the automatic reboot of a machine and will contain the implementation of the registry keys described earlier. Ensure that the "*AutoLogonCount*" registry value is set to 1.

The rationale behind using an executable is twofold:

1. The executable compiles and encrypts the content, ensuring that all values, including passwords, are securely stored. For example, PowerShell scripts store values as plain text, which can pose a security risk.
2. By default, the "*DefaultPassword*" value stores the password as plain text in the registry. This means that if a person with malicious intent gains access to the device, they can see the password stored in the registry. Using an executable in conjunction with the "*AutoLogonCount*" registry key ensures that after the reboot, the registry entries are cleared, eliminating any security issues.

Login Enterprise Windows Launcher user interface

When the Launcher loads, pay attention to the Login Enterprise Launcher menu window:



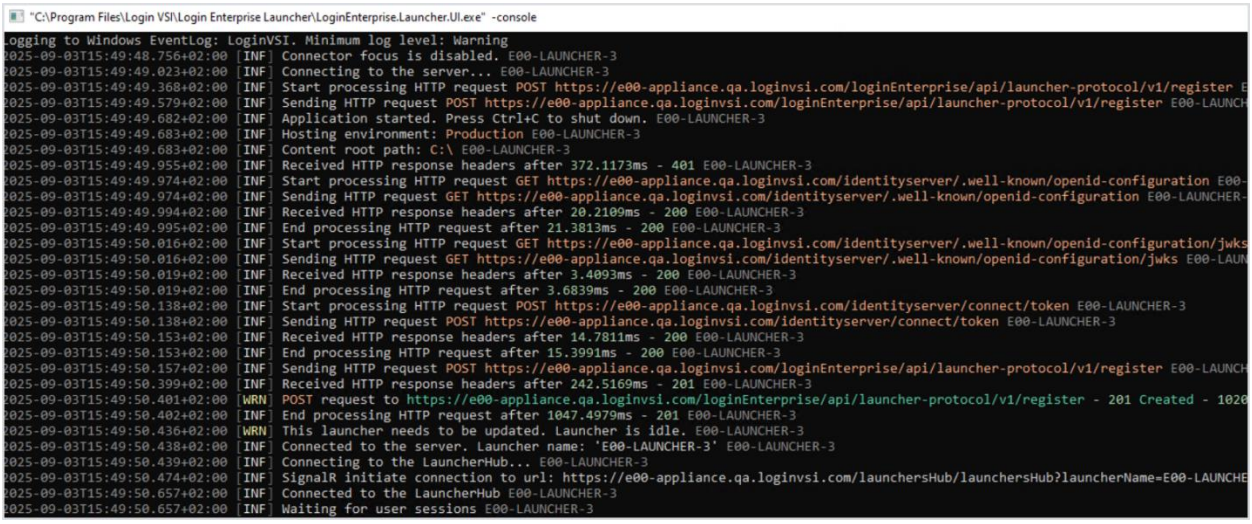
Menu option	Definition
Top bar	Shows the Launcher version
Launcher name	The Launcher name defaults to the machine name where the Launcher software is installed. If a specific name is required for a Launcher, it should match the name of the machine.
Connections pane	Shows active and historical Test session connections and status
Right-side menu	Shows connection information for selected connections
Open logs folder	Shows deeper troubleshooting information, if needed
Bottom launcher connection status banner	If the banner is green, the connection to the Login Enterprise Virtual Appliance is successful. The Appliance hostname and the Appliance version are at the right of the connection status banner.

Running the Launcher in Console Mode

LoginEnterprise.Launcher.UI is the current Launcher GUI. It can run in GUI or console mode. By default, it runs in a GUI.

To use console mode, run the Launcher executable with the `-console` switch, such as:

```
"C:\Program Files\Login VSI\Login Enterprise
Launcher\LoginEnterprise.Launcher.UI.exe" -console
```



Managing Access to Launcher and Launcher Groups

After a Launcher is registered, you can control access by assigning roles. For information, see [Managing Access to Launcher and Launcher Groups](#).

Troubleshooting and best practices

Certificate update

If a new certificate is uploaded to the appliance, we recommend reinstalling the launcher software.

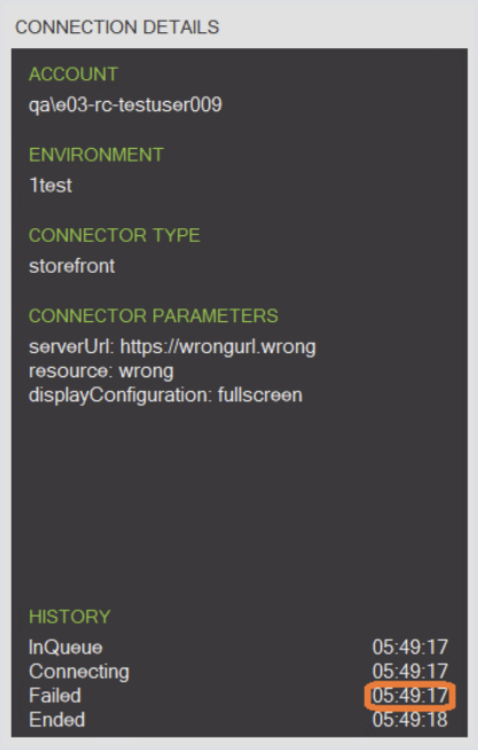
Connection status

1. Pay attention to the Launcher's UI bottom toolbar.
    1. If there's a connection error, the toolbar won't be green.
2. Check the Launcher page in the Login Enterprise web interface for the Launcher's hostname;
    1. If it doesn't exist, a connection error may be occurring.

2. Sometimes, the log files in the *%appdata%* path for the logged-in Launcher user can provide helpful information, so remember to check them as well. For instructions on how to do this, see the procedure below.
3. Launcher debug level logging can be enabled by adding a Launcher to a Test. For more information, see Adding a Launcher to a Test.

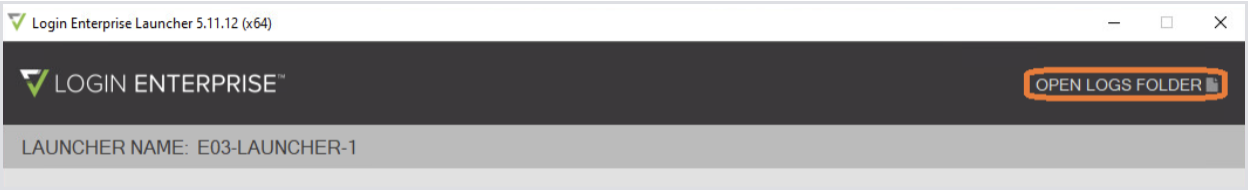
Connection failure overview

When running the Login Enterprise Launcher, whether for the first time or after extended use, you may encounter issues where a session fails to log on. This is typically indicated by a "Connection failed" message in the **Login Enterprise Launcher menu > Connection details**.

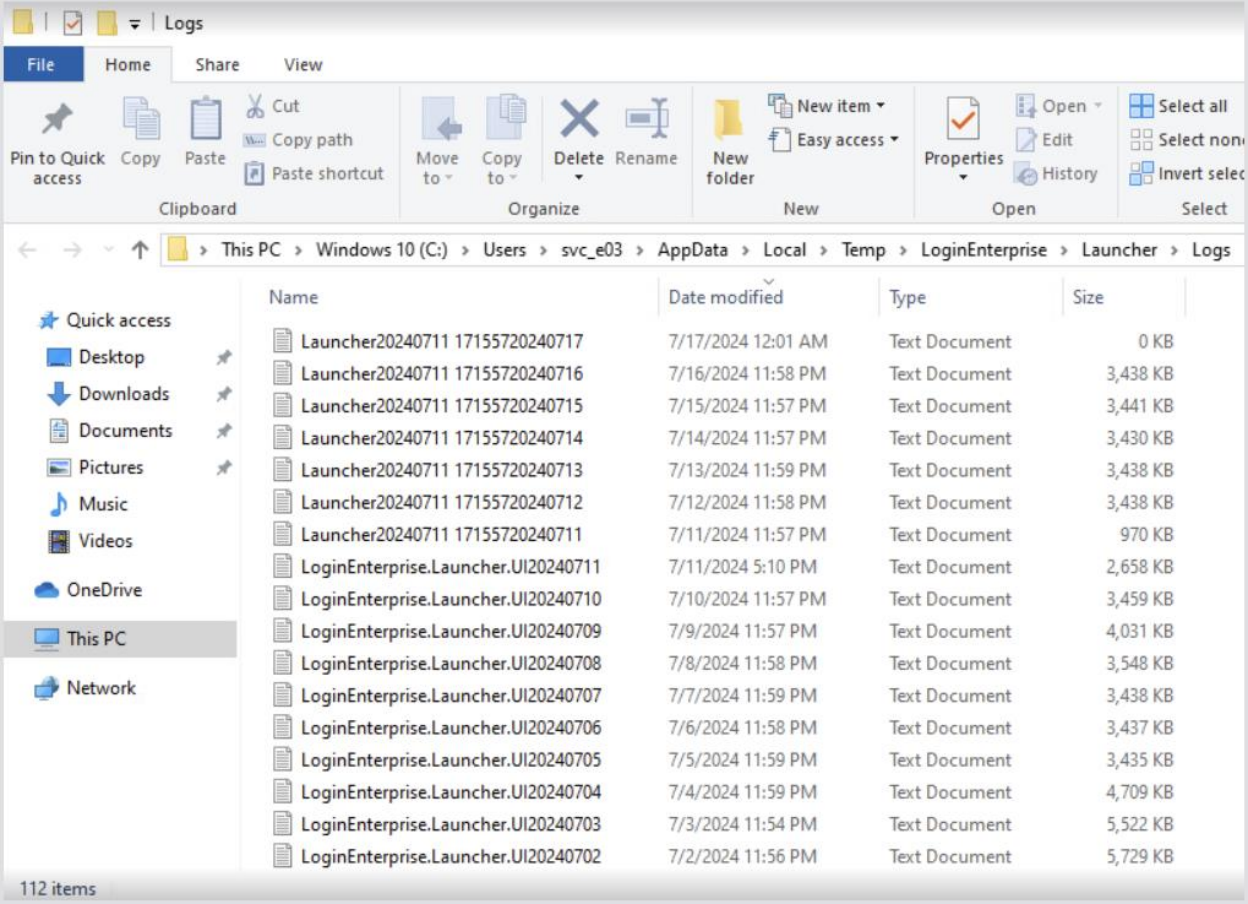


Diagnosing connection failure

Connection issues can vary based on the connector type used. To diagnose the issue, you need to examine the log files of the Launcher.



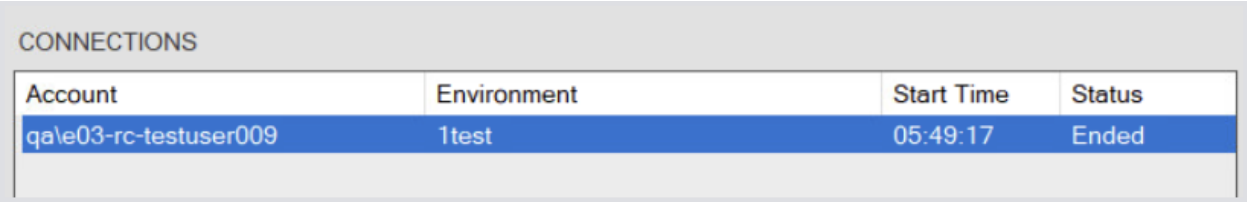
1. Locate the log files in the **%tmp%\LoginEnterprise\Launcher\Logs** folder on your machine.



These logs contain detailed activity records of the Launcher. Each file presents a comprehensive list of events.

```
Launcher20240717 23151920240717 - Notepad
File  Edit  Format  View  Help
2024-07-17T23:15:26.122+02:00 [INF] Connector focus is disabled. E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.HostedService.LauncherService
2024-07-17T23:15:26.455+02:00 [INF] Connecting to the server... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.Registration
2024-07-17T23:15:26.977+02:00 [INF] Start processing HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.LogicalHandler
2024-07-17T23:15:27.188+02:00 [INF] Sending HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:27.262+02:00 [INF] Application started. Press Ctrl+C to shut down. E03-LAUNCHER-1 LoginEnterprise.Launcher.UI Microsoft.Hosting.Lifetime
2024-07-17T23:15:27.262+02:00 [INF] Hosting environment: Production E03-LAUNCHER-1 LoginEnterprise.Launcher.UI Microsoft.Hosting.Lifetime
2024-07-17T23:15:27.262+02:00 [INF] Content root path: C:\Program Files\Login VSI\Login Enterprise Launcher E03-LAUNCHER-1 LoginEnterprise.Launcher.UI Microsoft.Hosting.Lifetime
2024-07-17T23:15:27.767+02:00 [INF] Received HTTP response headers after 574.1037ms - 401 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:27.822+02:00 [INF] Start processing HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:27.822+02:00 [INF] Sending HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration" E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.836+02:00 [INF] Received HTTP response headers after 13.9405ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.837+02:00 [INF] End processing HTTP request after 15.1071ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:27.869+02:00 [INF] Start processing HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration/jwks" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:27.869+02:00 [INF] Sending HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration/jwks" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.874+02:00 [INF] Received HTTP response headers after 4.0693ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.874+02:00 [INF] End processing HTTP request after 4.3531ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:28.198+02:00 [INF] Start processing HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/identityserver/connect/token" E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:28.198+02:00 [INF] Sending HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/identityserver/connect/token" E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:28.210+02:00 [INF] Received HTTP response headers after 11.7556ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:28.210+02:00 [INF] End processing HTTP request after 12.3193ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:28.213+02:00 [INF] Sending HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:28.243+02:00 [INF] Received HTTP response headers after 30.2567ms - 201 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:28.245+02:00 [WRN] POST request to "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" - 201 "Created" - 1260ms (Threshold is 1000ms)
E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginVsi.HttpClients.HttpClientLogger
2024-07-17T23:15:28.246+02:00 [INF] End processing HTTP request after 1287.8008ms - 201 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.LogicalHandler
2024-07-17T23:15:28.310+02:00 [INF] Connected to the server. Launcher name: 'E03-LAUNCHER-1' E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.Registration
2024-07-17T23:15:28.311+02:00 [INF] Connecting to the LauncherHub... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
2024-07-17T23:15:28.329+02:00 [INF] SignalR initiate connection to url: https://e03-appliance-rc.qa.loginvsi.com/launchersHub/launchersHub?launcherName=E03-LAUNCHER-1&runId=cf7d17a6-ec1e-4168-
912f-ae0ad65c7127 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
2024-07-17T23:15:28.810+02:00 [INF] Connected to the LauncherHub E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
2024-07-17T23:15:28.811+02:00 [INF] Waiting for user sessions E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
```

2. Identify the timestamp of the failed session from the Launcher overview.



CONNECTIONS

Account	Environment	Start Time	Status
qa\e03-rc-testuser009	1test	05:49:17	Ended

3. Open the corresponding log file and search for the timestamp to pinpoint the events during that session.

```
Launcher20240717 23151920240717 - Notepad
File  Edit  Format  View  Help
2024-07-17T23:15:26.122+02:00 [INF] Connector focus is disabled. E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.HostedService.LauncherService
2024-07-17T23:15:26.455+02:00 [INF] Connecting to the server... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.Registration
2024-07-17T23:15:26.977+02:00 [INF] Start processing HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.LogicalHandler
2024-07-17T23:15:27.188+02:00 [INF] Sending HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:27.262+02:00 [INF] Application started. Press Ctrl+C to shut down. E03-LAUNCHER-1 LoginEnterprise.Launcher.UI Microsoft.Hosting.Lifetime
2024-07-17T23:15:27.262+02:00 [INF] Hosting environment: Production E03-LAUNCHER-1 LoginEnterprise.Launcher.UI Microsoft.Hosting.Lifetime
2024-07-17T23:15:27.262+02:00 [INF] Content root path: C:\Program Files\Login VSI\Login Enterprise Launcher E03-LAUNCHER-1 LoginEnterprise.Launcher.UI Microsoft.Hosting.Lifetime
2024-07-17T23:15:27.767+02:00 [INF] Received HTTP response headers after 574.1037ms - 401 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:27.822+02:00 [INF] Start processing HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:27.822+02:00 [INF] Sending HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration" E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.836+02:00 [INF] Received HTTP response headers after 13.9405ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.837+02:00 [INF] End processing HTTP request after 15.1071ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:27.869+02:00 [INF] Start processing HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration/jwks" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:27.869+02:00 [INF] Sending HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/identityserver/.well-known/openid-configuration/jwks" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.874+02:00 [INF] Received HTTP response headers after 4.0693ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:27.874+02:00 [INF] End processing HTTP request after 4.3531ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:28.198+02:00 [INF] Start processing HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/identityserver/connect/token" E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:28.198+02:00 [INF] Sending HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/identityserver/connect/token" E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:28.210+02:00 [INF] Received HTTP response headers after 11.7556ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.ClientHandler
2024-07-17T23:15:28.210+02:00 [INF] End processing HTTP request after 12.3193ms - 200 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IOpenIdConnectClient.LogicalHandler
2024-07-17T23:15:28.213+02:00 [INF] Sending HTTP request POST "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" E03-LAUNCHER-1
LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:28.243+02:00 [INF] Received HTTP response headers after 30.2567ms - 201 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.ClientHandler
2024-07-17T23:15:28.245+02:00 [WRN] POST request to "https://e03-appliance-rc.qa.loginvsi.com/loginEnterprise/api/launcher-protocol/v1/register" - 201 "Created" - 1260ms (Threshold is 1000ms)
E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginVsi.HttpClients.HttpClientLogger
2024-07-17T23:15:28.246+02:00 [INF] End processing HTTP request after 1287.8008ms - 201 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI System.Net.Http.HttpClient.IMicroServiceClient.LogicalHandler
2024-07-17T23:15:28.310+02:00 [INF] Connected to the server. Launcher name: 'E03-LAUNCHER-1' E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.Registration
2024-07-17T23:15:28.311+02:00 [INF] Connecting to the LauncherHub... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
2024-07-17T23:15:28.329+02:00 [INF] SignalR initiate connection to url: https://e03-appliance-rc.qa.loginvsi.com/launchersHub/launchersHub?launcherName=E03-LAUNCHER-1&runId=cf7d17a6-ec1e-4168-
912f-ae0ad65c7127 E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
2024-07-17T23:15:28.810+02:00 [INF] Connected to the LauncherHub E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
2024-07-17T23:15:28.811+02:00 [INF] Waiting for user sessions E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.LauncherProtocol.LauncherProtocolSignalR
```

4. Look for the event type "**ConnectionFailed**," which is automatically recorded whenever a connection attempt fails.

```
Launcher20240717 23214120240717 - Notepad
File  Edit  Format  View  Help
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.287+02:00 [INF] [StoreFrontConnector]: Authentication OK, getting data E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.288+02:00 [INF] [StoreFrontConnector]: 3524 symbols were read from response E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.291+02:00 [INF] [StoreFrontConnector]: Token is created successfully E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Got 'Unauthorized' status trying to get the token E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Temporary token is successfully created E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Sending POST request to url 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Authentication/auth/v1/token'. The message's
length is 384 bytes. E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Request to 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Authentication/auth/v1/token' was sent. Getting response...
E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.362+02:00 [INF] [StoreFrontConnector]: Response from 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Authentication/auth/v1/token' is received E03-LAUNCHER-1
LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.363+02:00 [INF] [StoreFrontConnector]: 3226 symbols were read from response E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.364+02:00 [INF] [StoreFrontConnector]: Token is created successfully E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.364+02:00 [INF] [StoreFrontConnector]: Getting resources again with the authentication token H4sIAAAAAA... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.364+02:00 [INF] [StoreFrontConnector]: Sending GET request to 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Store/resources/v2' E03-LAUNCHER-1
LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.564+02:00 [INF] [StoreFrontConnector]: Response from 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Store/resources/v2' is received E03-LAUNCHER-1
LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.564+02:00 [INF] [StoreFrontConnector]: Reading resource data... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.637+02:00 [INF] [StoreFrontConnector]: 3935 symbols were read from response E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.637+02:00 [INF] [StoreFrontConnector]: Parsing resource data... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.678+02:00 [INF] [StoreFrontConnector]: Finished parsing received XML data. 2 resources were processed E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.681+02:00 [INF] [Handler @11:21:49 PM]: UserSession aa1f10fa-c54c-4493-924d-3dec650a137d Connection Event Received, Event Type ConnectionFailed.  E03-
LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
```

5. Trace back through the log entries to determine the cause of the failure. In the example below, the issue was related to the unavailability of a resource called "**Test**".

```
Launcher20240717 23214120240717 - Notepad
File  Edit  Format  View  Help
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.287+02:00 [INF] [StoreFrontConnector]: Authentication OK, getting data E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.288+02:00 [INF] [StoreFrontConnector]: 3524 symbols were read from response E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.291+02:00 [INF] [StoreFrontConnector]: Token is created successfully E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Got 'Unauthorized' status trying to get the token E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Temporary token is successfully created E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Sending POST request to url 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Authentication/auth/v1/token'. The message's
length is 384 bytes. E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.292+02:00 [INF] [StoreFrontConnector]: Request to 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Authentication/auth/v1/token' was sent. Getting response...
E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.362+02:00 [INF] [StoreFrontConnector]: Response from 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Authentication/auth/v1/token' is received E03-LAUNCHER-1
LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.363+02:00 [INF] [StoreFrontConnector]: 3226 symbols were read from response E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.364+02:00 [INF] [StoreFrontConnector]: Token is created successfully E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.364+02:00 [INF] [StoreFrontConnector]: Getting resources again with the authentication token H4sIAAAAAA... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:48.364+02:00 [INF] [StoreFrontConnector]: Sending GET request to 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Store/resources/v2' E03-LAUNCHER-1
LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.564+02:00 [INF] [StoreFrontConnector]: Response from 'https://e03-xd7infra.qa.loginvsi.com/Citrix/Store/resources/v2' is received E03-LAUNCHER-1
LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.564+02:00 [INF] [StoreFrontConnector]: Reading resource data... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.637+02:00 [INF] [StoreFrontConnector]: 3935 symbols were read from response E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.637+02:00 [INF] [StoreFrontConnector]: Parsing resource data... E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.678+02:00 [INF] [StoreFrontConnector]: Finished parsing received XML data. 2 resources were processed E03-LAUNCHER-1 LoginEnterprise.Launcher.UI
LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.681+02:00 [INF] [Handler @11:21:49 PM]: UserSession aa1f10fa-c54c-4493-924d-3dec650a137d Connection Event Received, Event Type ConnectionFailed.  E03-
LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.Connectors.StoreFront.StoreFrontConnector
2024-07-17T23:21:49.682+02:00 [WRN] Connector connection failed. User session Id: 'aa1f10fa-c54c-4493-924d-3dec650a137d', extra information: 'There was no available resource with
name 'Test'' E03-LAUNCHER-1 LoginEnterprise.Launcher.UI LoginEnterprise.Launcher.HostedService.UserSessionsManager
2024-07-17T23:21:49.686+02:00 [INF] Start processing HTTP request GET "https://e03-appliance-rc.qa.loginvsi.com/loginenterprise/api/user-sessions/aa1f10fa-c54c-4493-924d-
```

TIP: You can also send the log files along with your inquiry to Login VSI Support at support@loginvsi.com.

Versioning notice

The Launcher app won't connect to a virtual appliance that's one version ahead of it. For example, if the Launcher app is version 5.9.99 and the virtual appliance (VA) is version 6.0.0, they will still communicate, but the Launcher won't attempt to launch Target connections.

Launcher process issues

In some environments, the Launcher process may freeze or become unresponsive. This typically indicates an environmental issue and may occur inconsistently. To address this in the short term while investigating the root cause, you can use our API to check whether the Launcher is active from the Virtual Appliance's perspective.

The sample code, which can be downloaded from here, can be scheduled to run as a task that validates whether the Launcher is running and can restart it if necessary. This script is designed to initiate the Launcher when it is not running.

Setup instructions

1. **API key and FQDN**: Obtain a read-only or higher API key and the Fully Qualified Domain Name (FQDN) of the Virtual Appliance. This FQDN is the same as the address used to access the Virtual Appliance interface. You can generate a new API key in the **External notifications** section in the **Login Enterprise sidebar menu**.
2. **Edit the script**:
    - Place the FQDN within the quotes on line 3.
    - Insert the generated API token in the single quotes on line 4. The single quotes on line 4 are intentional to accommodate special characters in the API token.

```
3   $global:fqdn = "" #Example fqdn = "demo.loginvsi.com"
4   $global:token = '' #Example token = 'f_Yl7m-9XncLQNA6BmPAY21lPOBCMzm7FzHyH2t4WQg'
```

Best practices

1. **Recommended deployment: two Launchers per environment**:
    - One Launcher within the data center: used to gauge connectivity and establish baseline measurements.
    - One Launcher outside the data center: used to assess real-world variations from the baseline.

This comparison helps identify potential performance degradation from either the connection to the data center or the data center itself. It also assesses environment accessibility from both internal and external perspectives.

2. **Configuring the Launcher (Windows)**: To ensure the successful implementation of the Login Enterprise Launcher on Windows, perform several configurations on the machine:
    - Automatic reboots for the Launcher

- Scheduled reboots for stability
    - To mitigate caching issues and other potential instabilities arising from prolonged uptime, schedule regular reboots for the Launcher machine. Set up a scheduled task within Windows to automatically reboot the machine on a daily or weekly basis.
    - For detailed instructions on setting up a scheduled task in Windows, see Windows Server – Schedule a Reboot.

3. **Power, Lock Screen, and Screensaver settings** - To ensure continuous operation of the Launcher, it's essential to prevent interruptions caused by power-saving modes, lock screens, and screensavers. Follow these steps to configure the necessary settings:
    - **Power Settings**:
        - Set the screen setting "On battery power, Turn off after" to **Never**.
        - Set the screen setting "When Plugged, Turn off after" to **Never**.
        - Set the power setting "On battery power, PC goes to sleep after" to **Never**.
        - Set the power setting "Sleep when plugged in, PC Goes to sleep after" to **Never**.
        - Set the "Lid close action" for both "On battery" and "Plugged In" scenarios to **Do Nothing**.
        - Set the "Power button action" for both "On battery" and "Plugged In" scenarios to **Do Nothing**.
        - Set the "Sleep button action" for both "On battery" and "Plugged In" scenarios to **Do Nothing**.
    - **Other Settings**:
        - Disable the screensaver on the machine.
        - Disable the lock screen on the machine.

Deployment and compatibility

Physical Launcher setup

When configuring the Launcher as a physical device, prioritize security and minimize the risk of unauthorized access by following these best practices:

1. **Secure placement**: Place the Launcher machine in a secure location, such as a locked storage room or server room. Limit access to the room as much as possible to minimize the risk of unauthorized entry.
2. **Kensington lock**: Ensure that the device supports the use of a Kensington lock to prevent easy removal from the secure location.
3. **Device selection for security**: To mitigate potential security risks, use a physical device without an attached screen. Avoid using laptops and instead opt for devices such as Intel NUCs, desktop machines, Windows Thin clients, IGEL Thin clients, or similar alternatives. If a laptop is used, ensure that the lid is closed at all times.
4. **Stable connection**: To maintain a stable connection, connect the Launcher to the internet using a wired LAN cable. Use wireless connections only when necessary.

IT environment Launcher implementation

Extend security considerations for the Launcher beyond the physical realm by following these recommendations:

1. **Avoid adding a Launcher device to the domain**: For management or remote connectivity, refrain from adding the Launcher machine to the Domain unless necessary. This reduces the risk of unauthorized access to the domain if the device is compromised.
2. **Updates and upgrades**: Ensure that the Launcher device is consistently updated with the latest security patches recommended by the software supplier.
3. **Add Launcher to automatic software distribution method**: Enhance manageability and streamline software upgrades for the Login Enterprise by adding the Launcher machine to your preferred software distribution setup, such as SCCM or similar tools. This enables easy distribution of new Launcher software updates as soon as they are released.
4. **Virus scanner configuration**: Configure and maintain a virus scanner on the Launcher machine at all times. Ensure the virus scanner is updated regularly to provide optimal protection against threats.

Login Enterprise Launcher requirements

For detailed information on the Launcher requirements, see the related section in the System Requirements.

Configuring the Linux Launcher

- Introduction
  - Notes and considerations
- Launcher requirements
  - Azure SKU sizes
- Downloading the Launcher
  - Network share deployment for the Launcher
- Installing the Launcher
  - Running the Launcher
  - Automatically starting the Launcher at logon
  - Ensuring proper connection
- Configuring Connectors
  - RDP Connector
  - Citrix Connector
  - Horizon Connector
  - Citrix Workspace
- Maintenance and updates
  - Updating the Launcher
- Troubleshooting and best practices
  - Certificate update
  - Connection status
  - Versioning notice

Introduction

To learn about the Launcher's role in Login Enterprise, what it does, how it works, and more, see Launcher Overview and Best Practices.

For details on where the Launchers architecturally reside in the Login Enterprise setup, see System Architecture and Components.

Notes and considerations

- The Linux Launchers, including IGEL, do not work with UWC (Universal Web Connector). UWC is a Windows-only tool now.
- All testing was conducted using Debian 10, Debian 12, Ubuntu 20.04 LTS, and Fedora 33.
- Support is provided on a best-effort basis.
- Restarting the Launcher machine on failure is not supported.
- VMware Horizon custom command line: `/usr/bin/vmware-view --serverURL={serverurl} --userName="{username}" --password="{password}" --domainName="{domain}" --desktopName="{resource}" --nonInteractive`

- PCOIP and ICA latency do not work out of the box. The root cause of this likely lies in the clients. Even without our Launcher, the vendors do not report latency. This issue is still being investigated.

Linux Launcher requirements

For detailed information on the Launcher requirements, see the related section in the System Requirements.

Azure SKU sizes

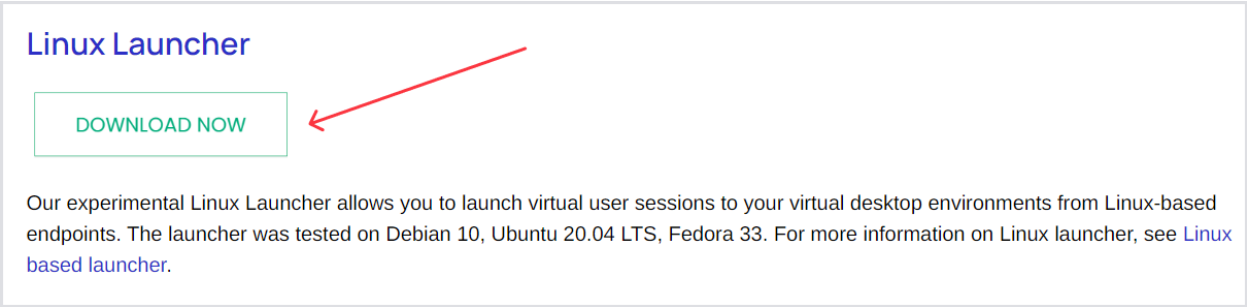
When deploying in Azure, consider using general-purpose D-series machines, such as the D2 v3 or D2 v4 instances. These machine instances offer 2 vCPUs and 8 GB of memory, providing sufficient resources for running the Launcher effectively.

For more information on Azure virtual machine SKU sizes, see the Azure documentation.

Downloading Linux Launcher

Since we don't supply the downloads or binaries for the Linux Launcher inside the Login Enterprise management console, you need to download the package externally. For this:

Navigate to Downloads, scroll down to find **Linux Launcher**, and click **Download now**.
NOTE: To access the Downloads page, you need to be signed in.



Network share deployment for Linux Launcher

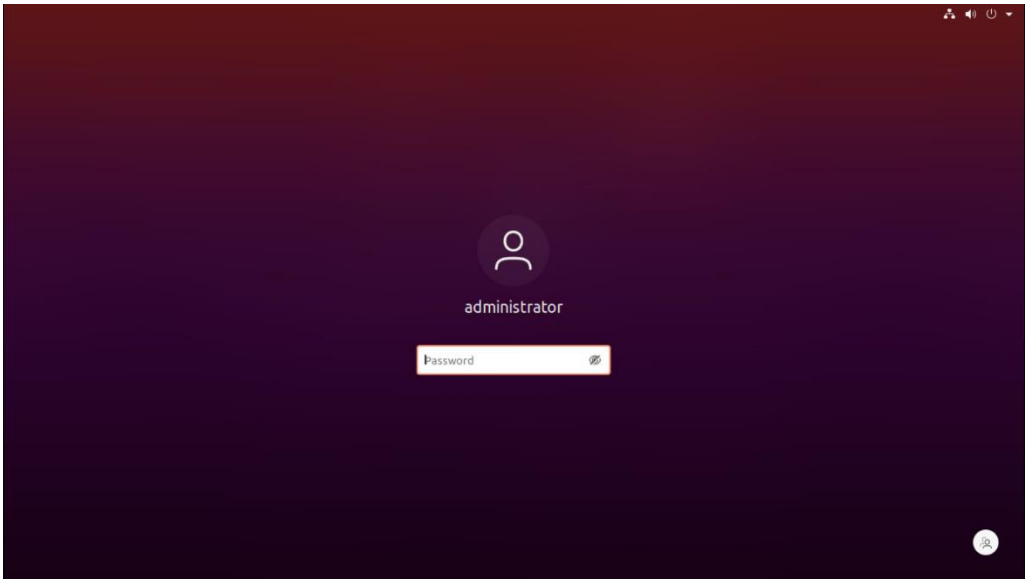
To streamline the deployment of the Linux Launcher across multiple systems, you can store the installer file on a network share. This approach eliminates the need for repeated downloads and ensures centralized management of the Launcher.

NOTE: The installation steps for the Linux Launcher remain the same regardless of where the installer is hosted. Ensure that the system on which you are installing the Launcher has access to the network share.

Installing Linux Launcher

NOTE: Ensure you have already downloaded the Linux Launcher onto your intended Linux machine.

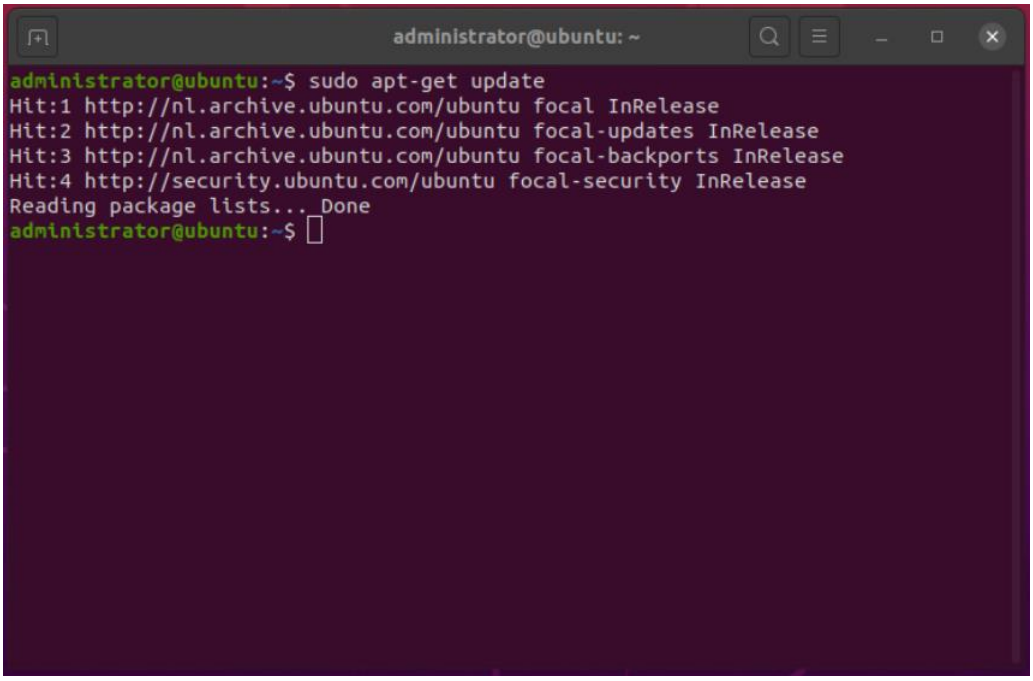
1. Log in to the Linux Machine. Open your terminal application and log into your Linux machine using your user credentials.



2. Open the terminal application on your machine, and run the following command:

```
sudo apt-get update
```

This command updates the package list to ensure that you have the latest information on the newest package versions. This helps ensure that when you install packages later, you are installing the latest versions available.



NOTE: If the Software Updater window appears with available updates, click **Install Now** to install the updates, and then continue with the instructions.

3. In the same terminal window, run the following command to install Curl:

```
sudo apt-get install curl
```

or

```
yum install curl
```

When prompted, type `Y` and press **Enter** to continue.

```
administrator@ubuntu: ~
administrator@ubuntu:~$ sudo apt-get install curl
[sudo] password for administrator:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libfprint-2-tod1 libllvm10 linux-headers-5.4.0-42
  linux-headers-5.4.0-42-generic linux-image-5.4.0-42-generic
  linux-modules-5.4.0-42-generic linux-modules-extra-5.4.0-42-generic
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl libcurl4
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 396 kB of archives.
After this operation, 1.115 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

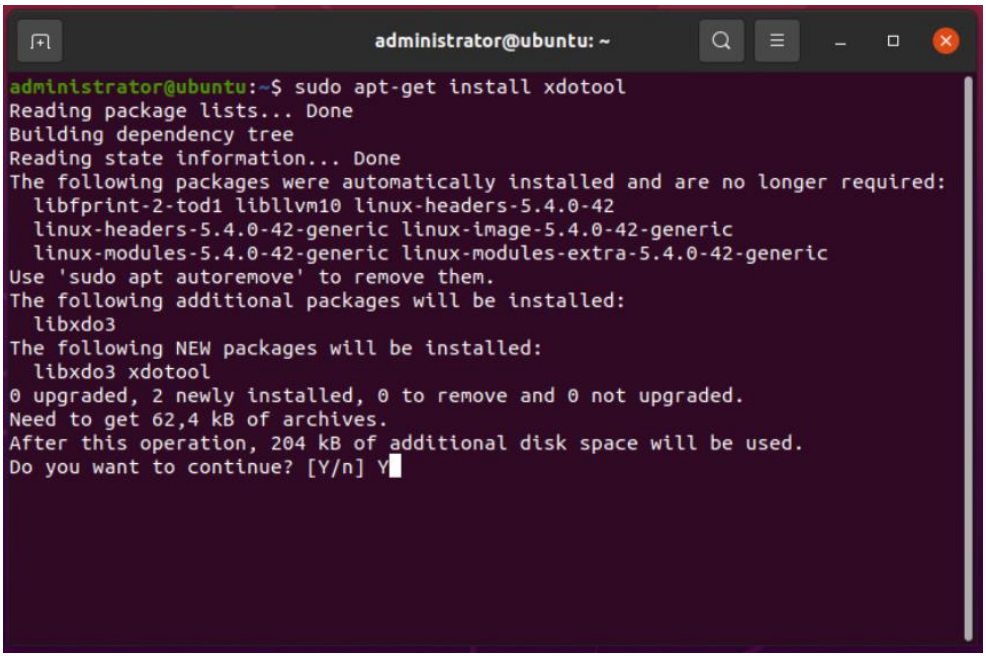
4. In the same terminal window, run the following command to install XDOTool:

```
sudo apt-get install xdotool
```

or

```
yum install xdotool
```

When prompted, type `Y` and press **Enter** to continue.



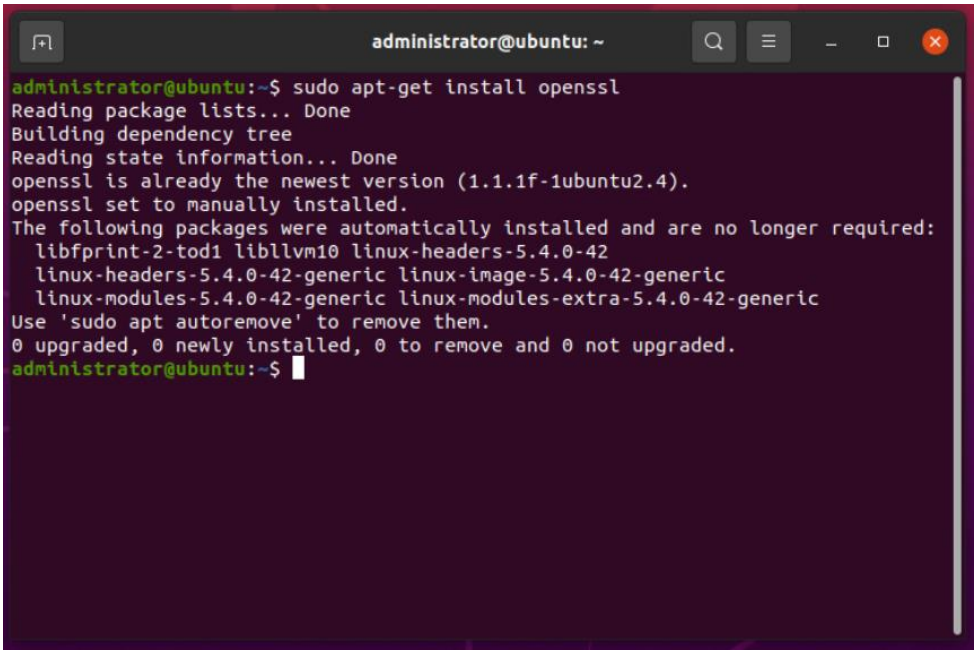
5. In the same terminal window, run the following command to install OpenSSL:

```
sudo apt-get install openssl
```

or

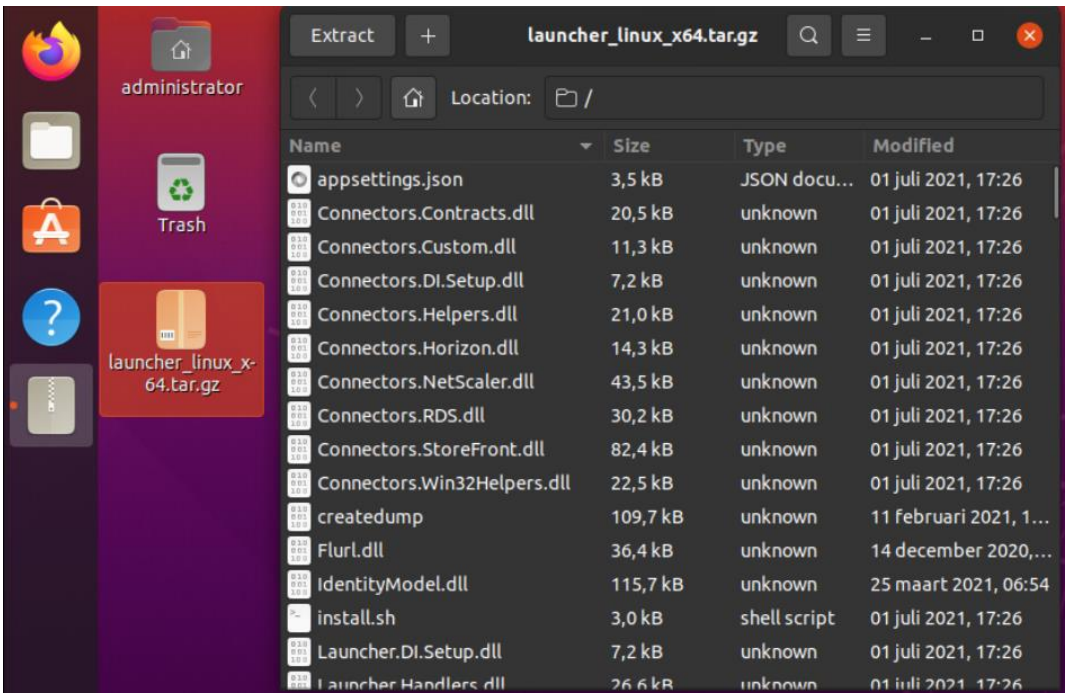
```
yum install openssl
```

When prompted, type `Y` and press **Enter** to continue.

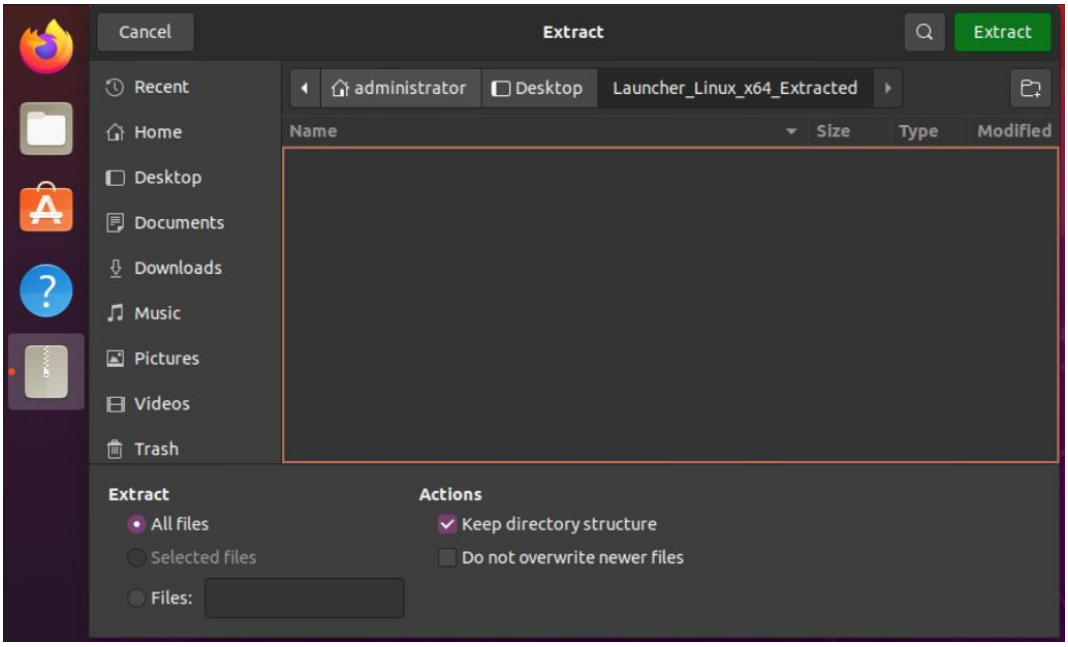


6. Unpack the Launcher files on the Ubuntu machine:

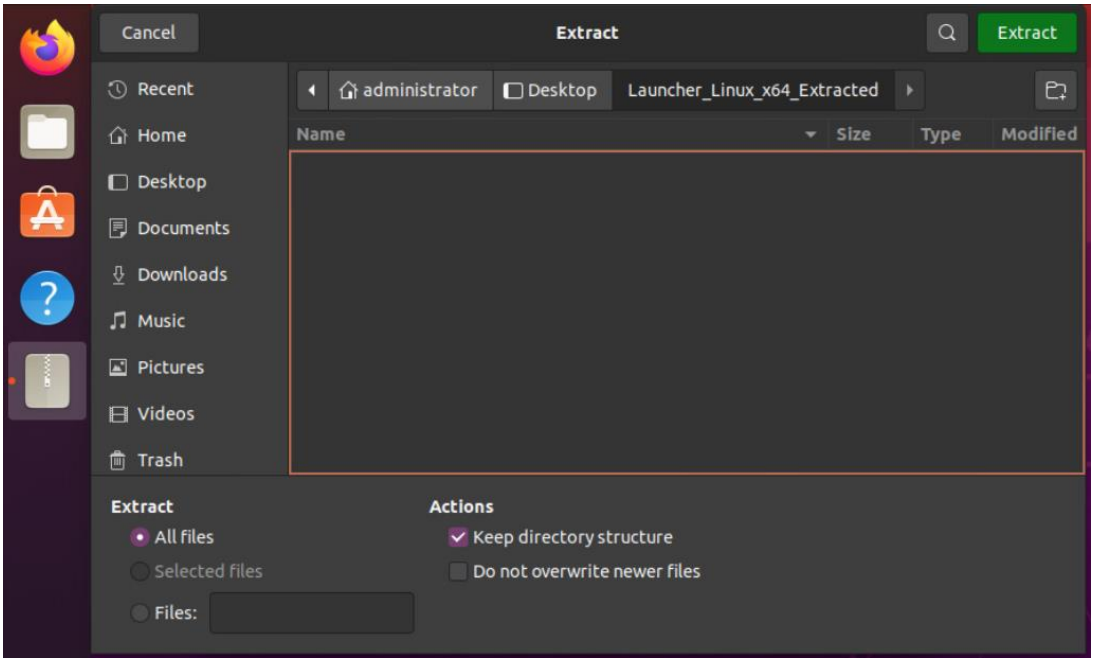
Double-click on the `launcher_linux_x64.tar.bz2` file to open it.



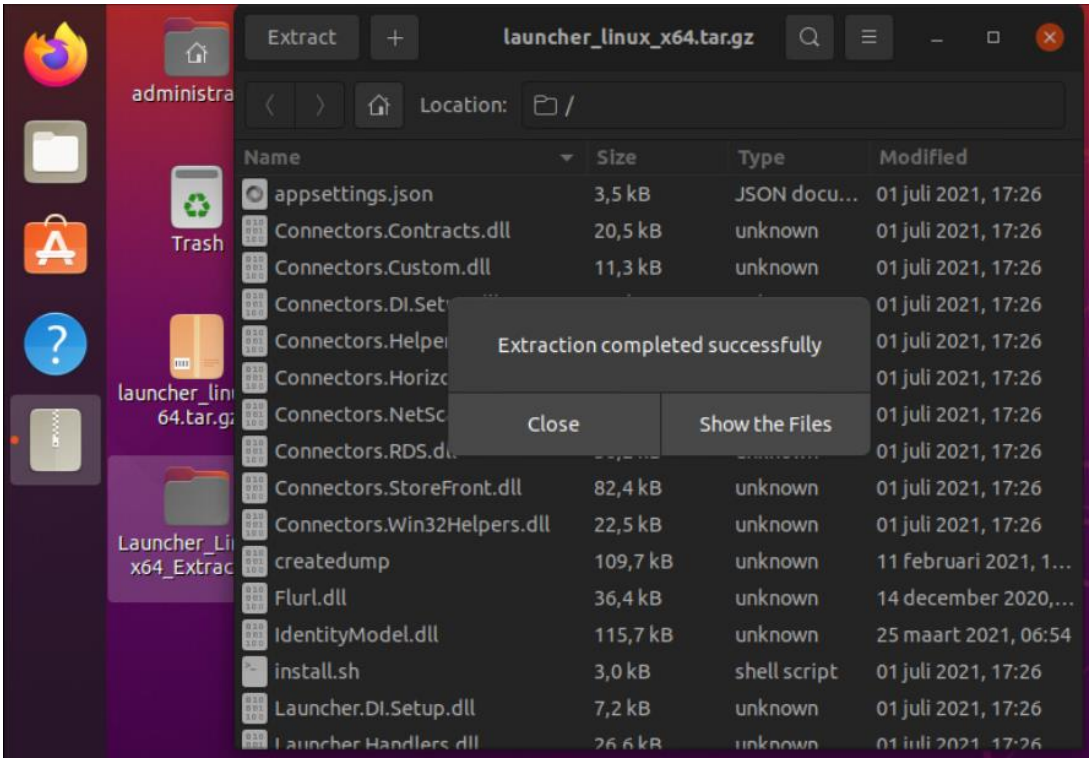
In the upper left corner, click **Extract**.



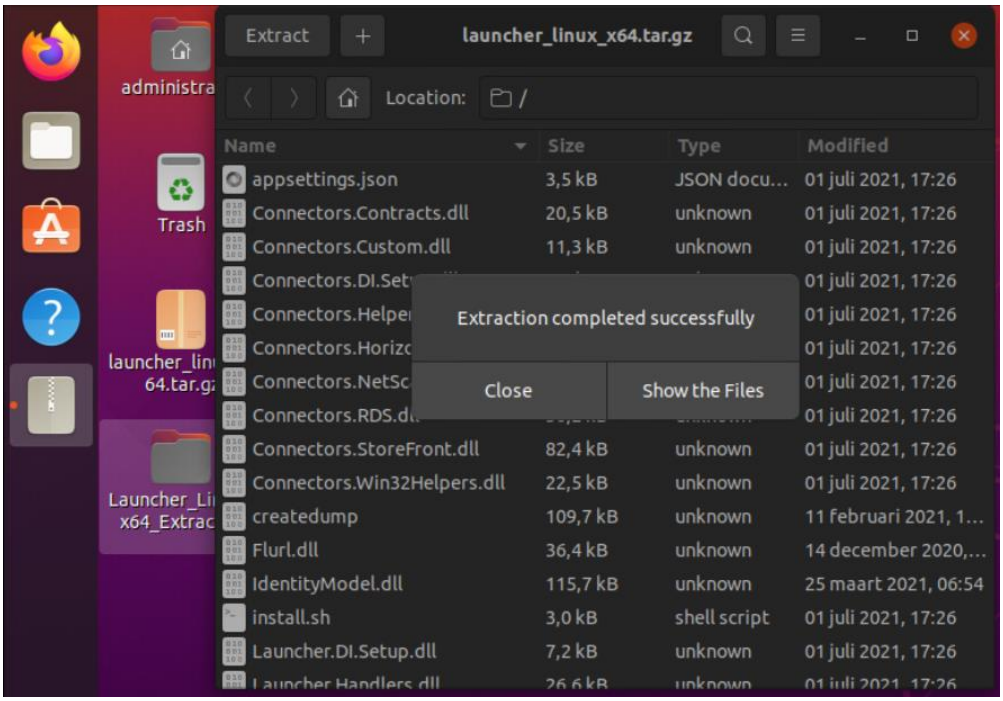
In the upper right corner, click the new folder icon to create a new folder in the desired location.



In the upper right corner, click **Extract** to extract the file into the new folder.



Click **Show the Files** to check the files.

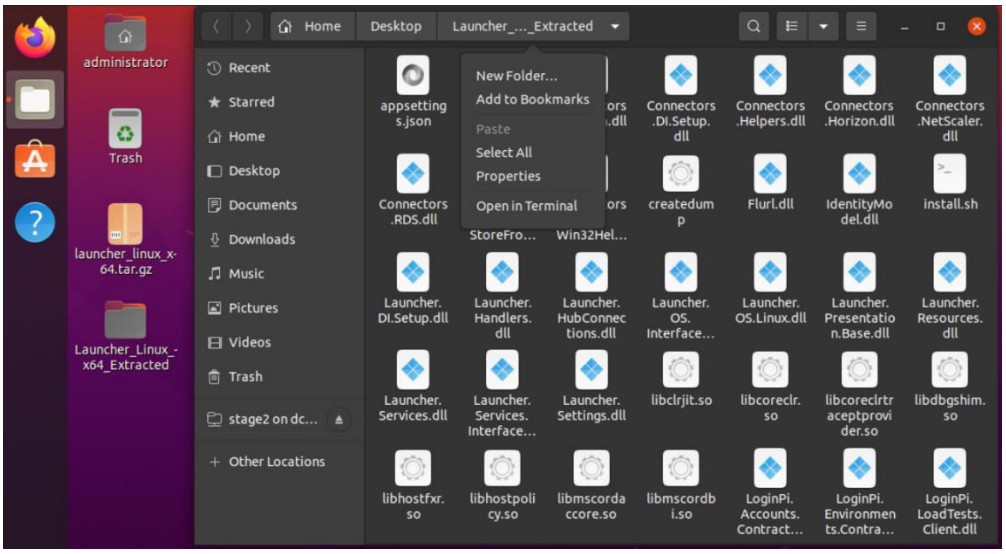


7. Browse to the extracted file folder in the terminal application:

a. Browse to the Extracted folder.

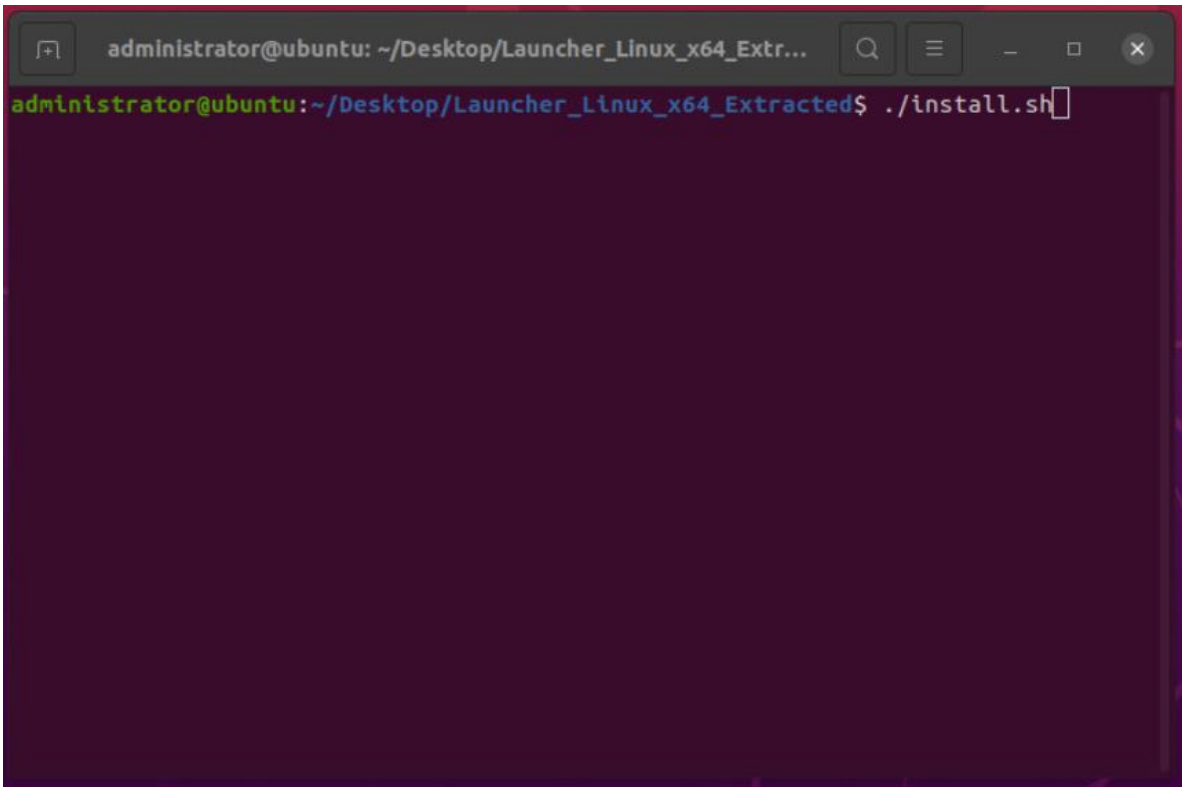
b. In the top menu, click on the downward-facing arrow.

c. Click **Open in Terminal**.

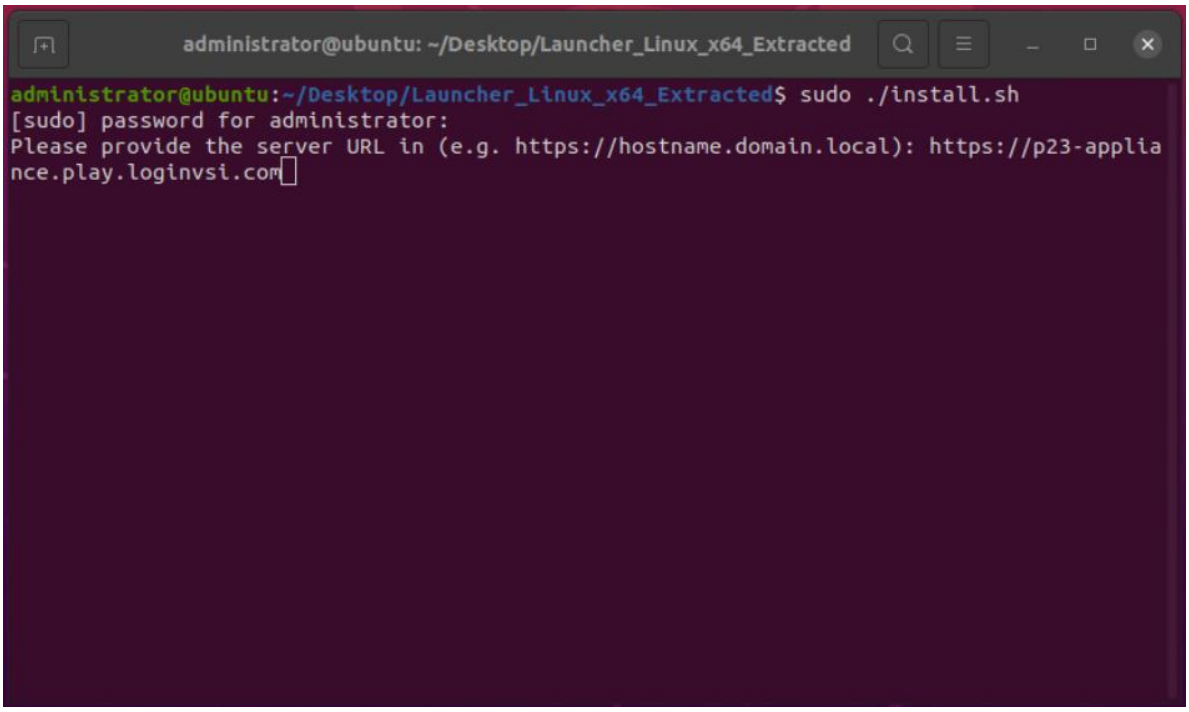


8. Execute the `install.sh` from the Terminal application. Use the following command:

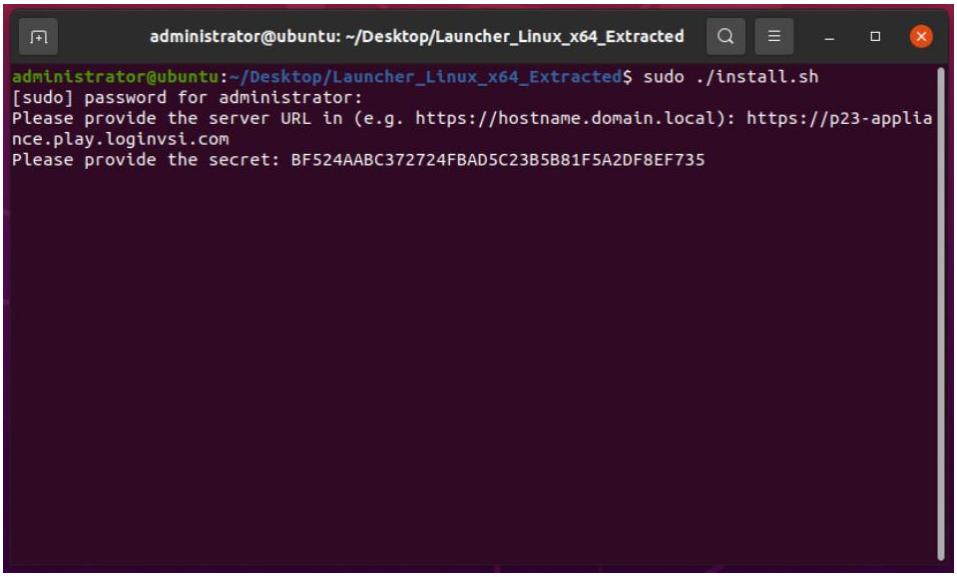
```
sudo ./install.sh
```



Insert the URL of the Appliance. For example, https://MyAppliance.MyDomain.com.



Insert the secret key of the Launcher. For example, BF524AABC372724FBAD5C23B5B81F5A2DF8EF735.

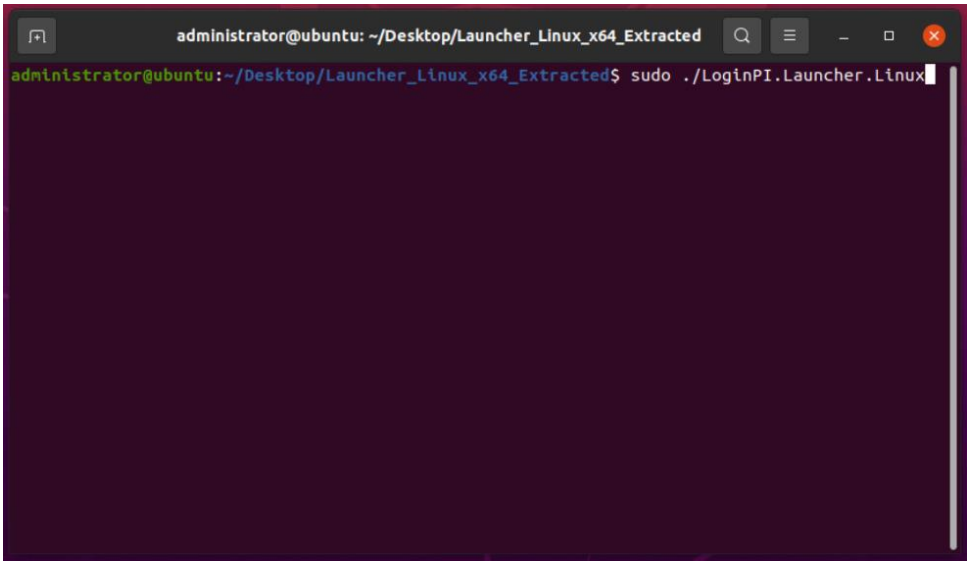


Running the Linux Launcher

Run the `LoginPI.Launcher.Linux` from the same terminal window. Use the following command:

```
./LoginPI.Launcher.Linux
```

Do not use `sudo` for this installation, as the Citrix connector will not work if `sudo` is used to execute `LoginPI.Launcher.Linux`. This is because the logged-in user to X11 must be the administrator.



Automatically starting the Linux Launcher at logon

To ensure the Launcher starts automatically when the launcher user logs on, refer to the documentation of your specific Linux distribution. Different Linux distributions may have different methods for configuring executable files to run at logon.

For more information, consult the relevant section in your distribution's documentation or support resources.

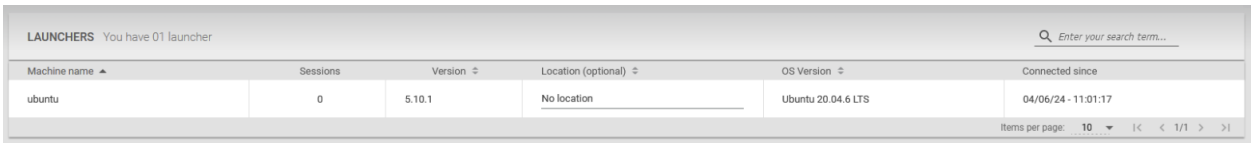
Ensuring proper connection

When the Linux Launcher is run, it will attempt to establish a connection to the appliance. A successful connection indicates that the Launcher is operating correctly:

```
Reading Configuration
 18:00:56 INF  Waiting for appliance to become healthy with timeout of 20 minutes
 18:00:59 VRB  Attempting to retrieve discovery information from 'https://e05-appliance-rc.qa.loginvsi.com/identityserve
r/.well-known/openid-configuration'.
 18:00:59 VRB  Attempting to request token using client credentials (Url: 'https://e05-appliance-rc.qa.loginvsi.com/iden
tityserver/connect/token', ClientId: 'Launcher', Scope: 'microservice')


 18:01:10 INF  Appliance health check response code: OK, status: Healthy
 18:01:10 VRB  Attempting to retrieve discovery information from 'https://e05-appliance-rc.qa.loginvsi.com/identityserve
r/.well-known/openid-configuration'.
 18:01:10 INF  Start processing HTTP request GET https://e05-appliance-rc.qa.loginvsi.com/identityserver/.well-known/ope
nid-configuration
 18:01:10 VRB  Request Headers:
Accept: application/json
```

To verify the connection:

1. Open the Login Enterprise web interface.
2. In the **Sidebar menu**, navigate to **Configuration > Launchers**.
3. Look for the Machine name on the list. The Launcher machine name should appear in this list, e.g. "ubuntu" for an Ubuntu machine.



NOTE: This verification process is the same for all Launcher operating system types.

Additionally, when the Launcher attempts to launch a new session, you may see a console similar to the following example for an RDP connection:

```
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: 'Connector Settings:'.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      host: E05-RDSH-22.qa.loginvsi.com'.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      resource: '.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      gateway: '.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      suppressCertWarn: true'.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      displayConfiguration: fullscreen'.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: 'Credentials:'.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      Username : 'e05-rc-testuser015''.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      Password: '********''.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      Domain: 'qa.loginvsi.com''.
18:01:27 VRB  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: '      Email: '''.
18:01:27 INF  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: 'Obtaining RDP templates...'.
18:01:27 INF  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: 'Writing to '/tmp/LoginPI/RDS/82c4fa59-9f4f-427
0-af53-4aaeb3cea181-E05-RDSH-22.qa.loginvsi.com.rdp' file...'.
18:01:27 INF  [RDSLinuxConnector 82c4fa59-9f4f-4270-af53-4aaeb3cea181]: ''/tmp/LoginPI/RDS/82c4fa59-9f4f-4270-af53-4aae
b3cea181-E05-RDSH-22.qa.loginvsi.com.rdp' file was created'.
```

Configuring Connectors

Configuring connectors is an essential step in setting up your Linux Launcher for seamless remote access. Connectors serve as bridges between your local environment and remote resources, enabling you to access desktops, applications, and virtualized environments with ease. Follow the instructions below to set up RDP, Citrix, Horizon, and Citrix Workspace connectors and enhance your remote workflow.

RDP Connector

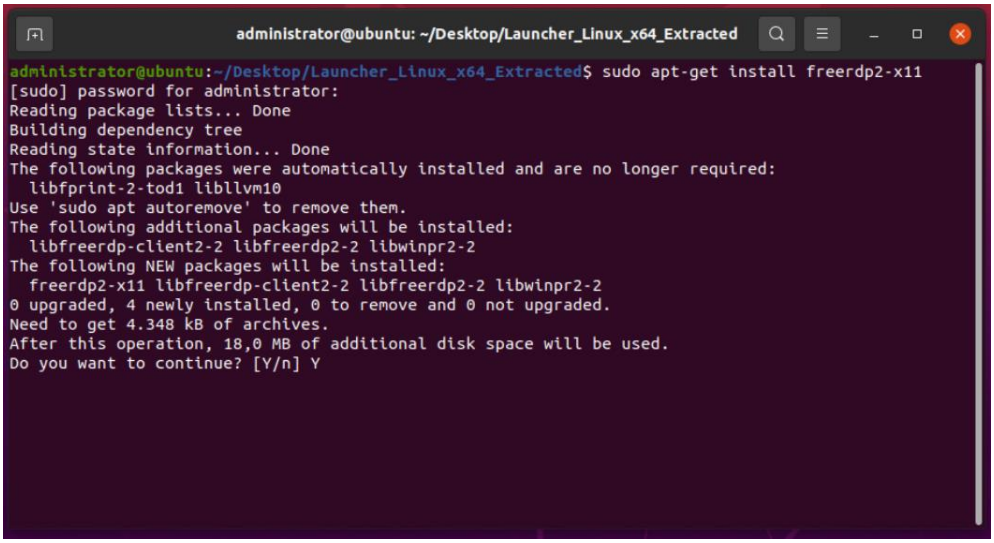
1. Open the Terminal application and run the following command to install FreeRDP2:

```
sudo apt-get install freerdp2-x11
```

or

```
yum install freerdp
```

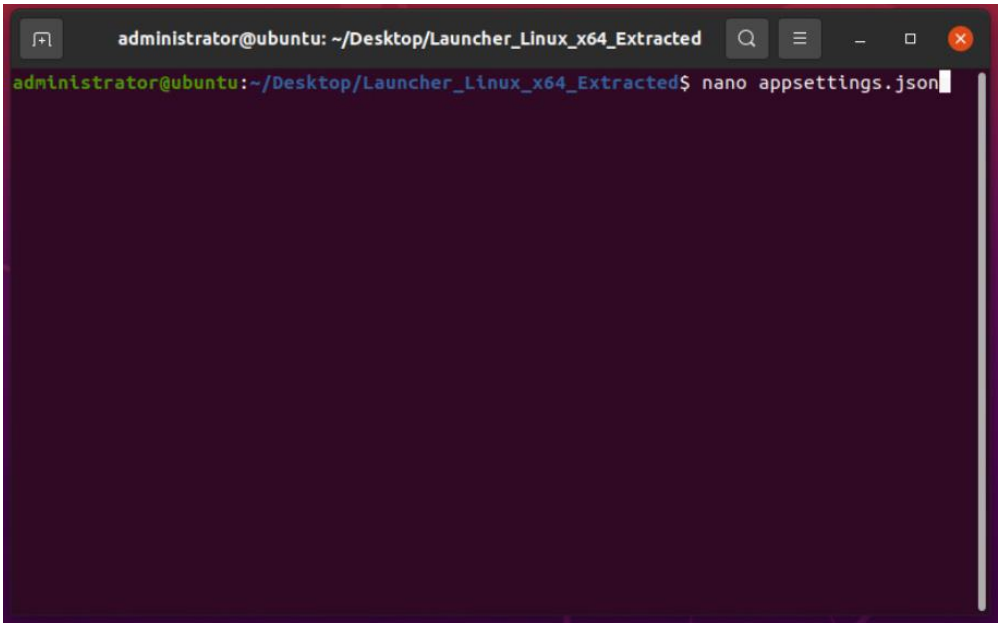
When prompted, type `Y` and press **Enter** to continue.



2. Configure the freeRDP2 path in appsettings.json (/usr/bin/xfreerdp):

Browse to the Launcher folder in the terminal window and run the following command:

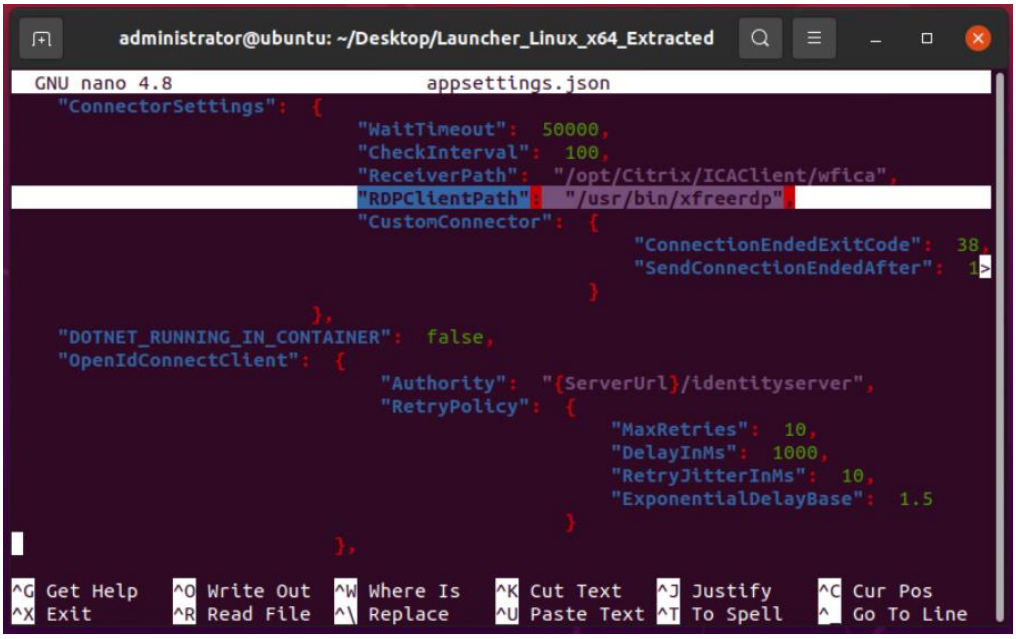
```
nano appsettings.json
```



Change the following line in the file:

From: `RDPClientPath: /services/rrdp/bin/igelrdp2`

To: `RDPClientPath: /usr/bin/xfreerdp`



Save and close the terminal.

Citrix Connector

1. Download the latest workspace client.

You can simply download the full package for Debian.

2. Install the package by running the following command:

```
sudo dpkg -i icaclient*.deb
```

The binary will be installed in the `/opt/Citrix/ICAClient/` folder, and the path to wfica will already be in the appsettings.json file, so it does not need to be edited.

Horizon Connector

a. Download the latest Horizon View client

b. Set execute permission for the downloaded file by running the following command:

```
chmod +x VMware*bundle
```

c. Install Horizon View client by running the following command:

```
sudo ./VMware*bundle --console --required --eulas-agreed
```

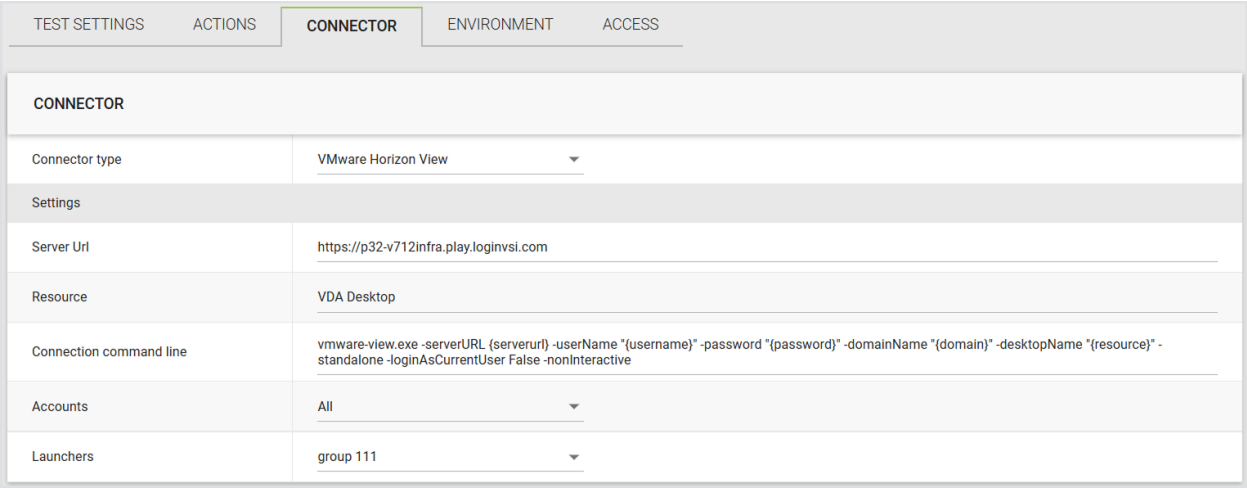
```
administrator@ubuntu:~/Downloads$ sudo ./VM*bundle --console --required --eulas-agreed
Extracting VMware Installer...done.
Installing VMware Horizon Client 8.4.0
    Configuring...
[####################################################################] 100%
Installation was successful.
Register and start installed services(s) after installation(Select
yes, the Installer will create necessary entries in your system
autostart or generate a launching script, so that the installed
service(s) can be ready before the Horizon Client starts)[yes/no]: no

Do you want to check your system compatibilities for Horizon Client,
this Scan will NOT collect any of your data?[yes/no]: no

administrator@ubuntu:~/Downloads$
```


```
administrator@ubuntu:~/Downloads$ sudo ./VM*bundle --console --required --eulas-agreed
Extracting VMware Installer...done.
Installing VMware Horizon Client 8.4.0
    Configuring...
[####################################################################] 100%
Installation was successful.
Register and start installed services(s) after installation(Select
yes, the Installer will create necessary entries in your system
autostart or generate a launching script, so that the installed
service(s) can be ready before the Horizon Client starts)[yes/no]: no

Do you want to check your system compatibilities for Horizon Client,
this Scan will NOT collect any of your data?[yes/no]: no

administrator@ubuntu:~/Downloads$
```


The binary path to the Horizon client is `/usr/bin/vmware-view` and the parameters may be different from the Windows client.

```
administrator@ubuntu:~/Downloads$ which vmware-view
/usr/bin/vmware-view
administrator@ubuntu:~/Downloads$ vmware-view --help | more
Using log file /tmp/vmware-administrator/vmware-horizon-client-22571.log
Usage:
  vmware-view [OPTION…] - connect to VMware Horizon desktops

Help Options:
  -h, --help                                        Show help options
  --help-all                                        Show all help options
  --help-gtk                                        Show GTK+ Options

Application Options:
  -K, --keep-wm-bindings                            Keep window manager key
  -s, --serverURL=<broker URL>                      Specify connection broke
  -u, --userName=<user name>                        Specify user name for pa
  -p, --password=<password>                         Specify password for pas
  -t, --tokenUserName=<token user name>             Specify user name for RS
  -c, --passcode=<passcode>                         Specify passcode for RSA
  -d, --domainName=<domain name>                    Specify domain for passw
  -a, --applicationName=<application name>          Specify application by r
  -n, --desktopName=<desktop name>                  Specify desktop by name.
  -q, --nonInteractive                              Connect automatically if
  --fullscreen                                      Enable full screen mode
```

Modify the Login Enterprise connector setting for Horizon connector to use Linux path and parameters:



Citrix Workspace

The Launcher is already set up for Citrix Workspace. If the Workspace app has been installed in a different directory, you will need to modify the `appsettings.json` file to include the correct path, similar to the FreeRDP configuration. You can find the `appsettings.json` file in the

installation directory. For more information on connectors and connection configurations, see Connectors.

Maintenance and updates

To ensure the smooth operation of your Linux Launcher, follow these maintenance and update recommendations:

1. Perform periodic OS updates.
2. Regularly update connection clients (e.g., VMware).
3. Restart the Linux Launcher host periodically.
4. Ensure the system remains logged in. If the screen gets locked or the user logs off, the Launcher will not be able to launch target sessions.

Updating Linux Launcher

To update the Linux Launcher, simply delete the Launcher directory and download/reinstall it from Login Enterprise.

Troubleshooting and best practices

Certificate update

We recommend reinstalling the Launcher software if a new certificate is uploaded to the appliance. For more information on certificates, see Managing Certificates.

Connection status

If the Login Enterprise logging/dashboard indicates a connection error in the event logs, or if you notice that the Launcher application isn't launching a console to the target when attempting to invoke a connection, follow these steps:

1. **Check the console output**: The best place to investigate is the default verbose console output for the Launcher process. This serves as the GUI for the Linux Launcher.
2. **Analyze the logs**: The Launcher logs are available in the console output of the Linux Launcher. Look for any messages that might indicate the nature of the problem.

By examining the console output, you can often identify the issue or get clues about what might be causing the problem.

Versioning notice

If the Launcher is a major version behind, it won't launch sessions anymore, as it's not supposed to. As a general recommendation, we advise updating all Launchers after updating Login Enterprise (manually or through a custom script; we do not provide a solution for this).

Configuring the Launcher in IGEL UMS 12

- Overview
- Requirements
- Getting the Login Enterprise certificate into UMS and onto Device
  - Confirming the Certificate is Applied to the Device (Optional)
- Installing the SDK certificates on the IGEL device
- Manually downloading and getting the Launcher package on the IGEL UMS web
- Getting the Launcher onto IGEL devices
- Running the Login Enterprise Launcher
- Updating the Login Enterprise Launcher

Overview

The Login Enterprise Launcher for IGEL UMS (Universal Management Suite) enables seamless deployment and management of virtual appliances on IGEL devices. This section walks you through setting it up and ensuring that all necessary certificates and packages are correctly installed and configured.

In the following sections, you will learn how to:

- Install and configure the required certificates on both the IGEL UMS and IGEL devices.
- Import the necessary SDK certificates to ensure secure communication.
- Deploy the Login Enterprise Launcher package to the IGEL UMS web interface and the IGEL device.
- Run the Login Enterprise Launcher to manage virtual appliance instances effectively.

By the end of this section, you will have successfully deployed and configured the Login Enterprise Launcher in your IGEL UMS environment, enabling you to manage virtual appliance deployments across your IGEL devices.

Requirements

These steps assume the following has been done already:

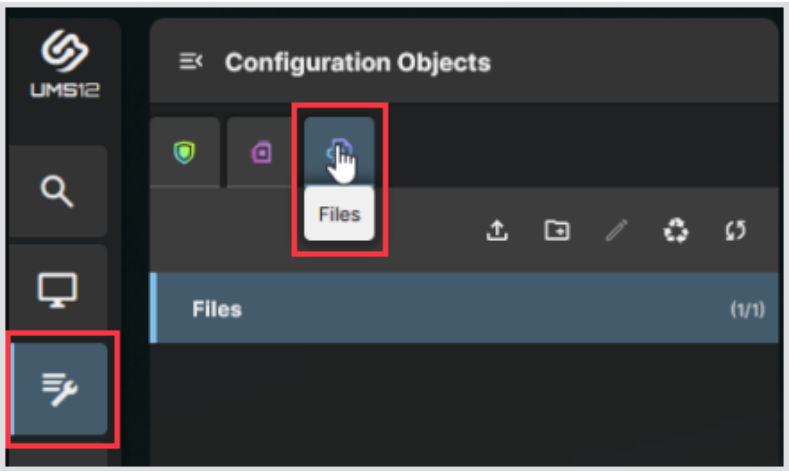
1. Login Enterprise is configured.
   1. Contact Login VSI Sales if not configured.
2. IGEL UMS 12 Desktop and UMS 12 Web are installed. For details, see Installing / Upgrading to IGEL UMS 12.
   1. The IGEL UMS is licensed. For details, see UMS Licenses.
   2. The IGEL UMS is Registered to IGEL. For details, see Registering the UMS.
   3. The IGEL UMS is configured as an Update proxy instead of the IGEL App Portal. For details, see Configuring Global Settings for the Update of IGEL OS Apps.

        1. This enables the Login Enterprise launcher to be installed without the App Portal.
3. An IGEL 12 OS Device is already set up to be used as a Login Enterprise Launcher. For details, see the Installation Procedure.
    1. The IGEL OS minimum version is 12.6 -- for the IGEL SDK certificates to work; certificates will be set up later in this article.
    2. The IGEL OS is licensed. For details, see Deploying Licenses through the IGEL UMS.
    3. The IGEL OS is registered to UMS. For details, see Registering IGEL OS Devices on the UMS Server.

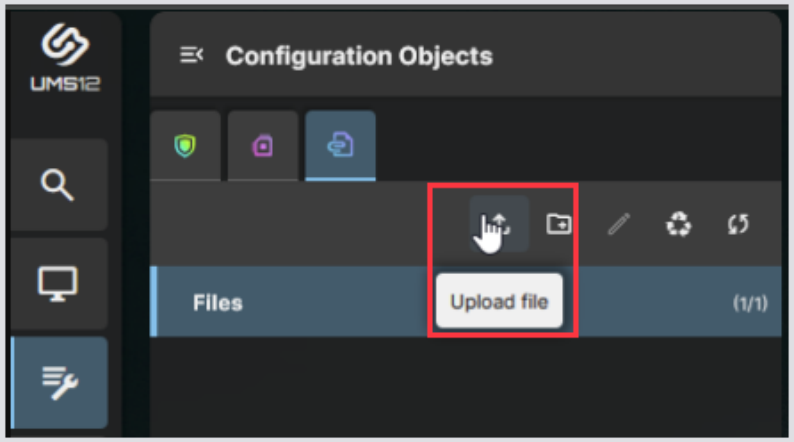
Getting the Login Enterprise certificate into UMS and onto Device

NOTE: The process below is only for self-signed certificates on the appliance. If your appliance has a CA-issued certificate, deploy the root and issuing CA certificates to the IGEL instead.

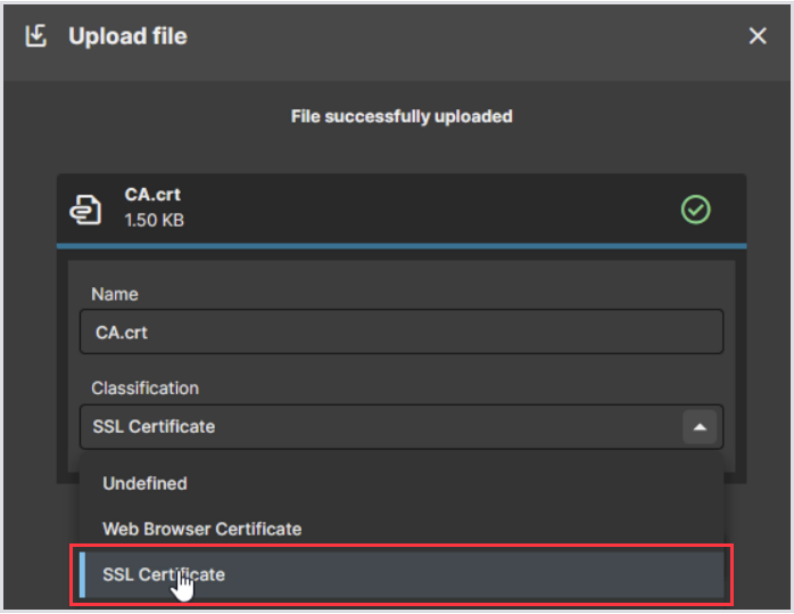
1. Download the Login Enterprise `CA.crt` in a web browser by navigating to `https://<FQDN Login Enterprise virtual appliance hostname>/contentDelivery/content/CA.crt`
2. Log in to the UMS web console.
3. In the **UMS sidebar menu**, go to **Configuration** and click **Files**.



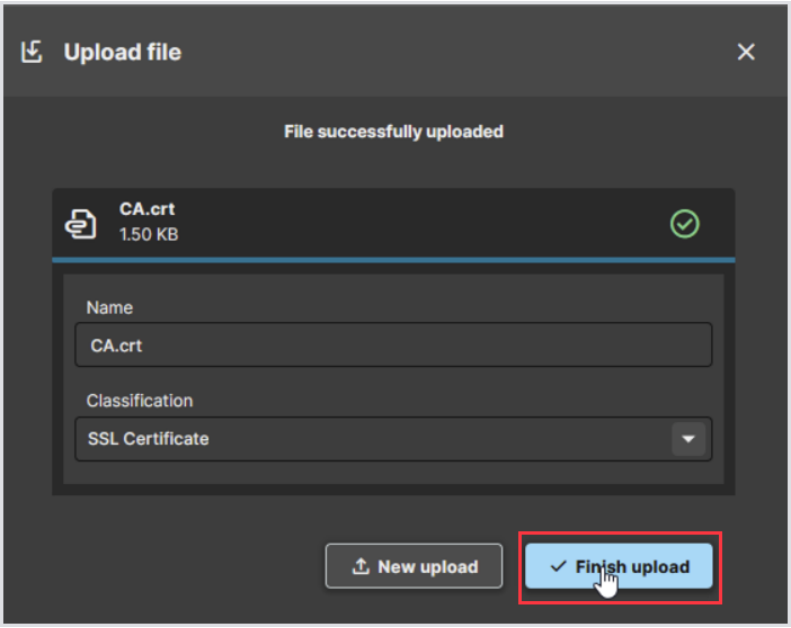
4. In **Files**, click **Upload File**. Navigate to the downloaded `CA.crt` and upload it.



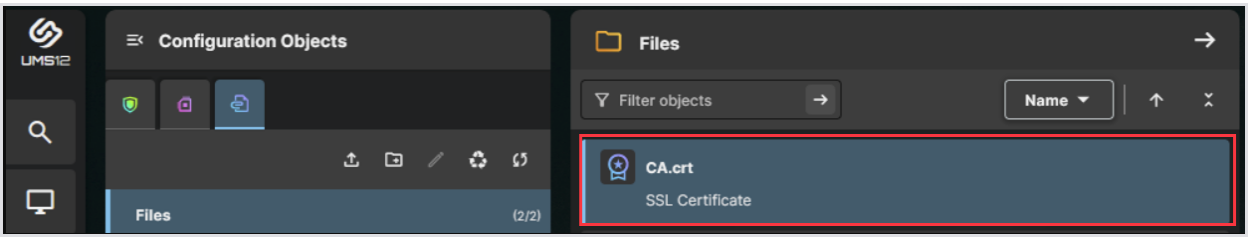
5. In **Classification**, select **SSL Certificate**.



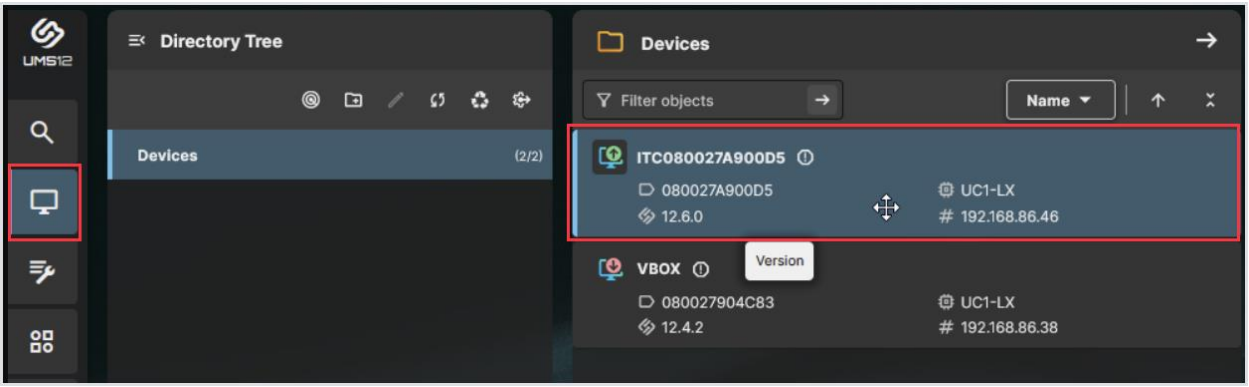
6. Click **Finish upload**.



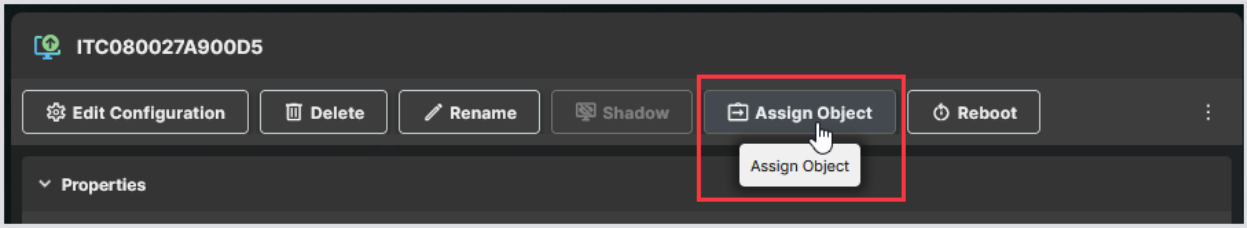
7. The `CA.crt` is now visible in the UMS web console.



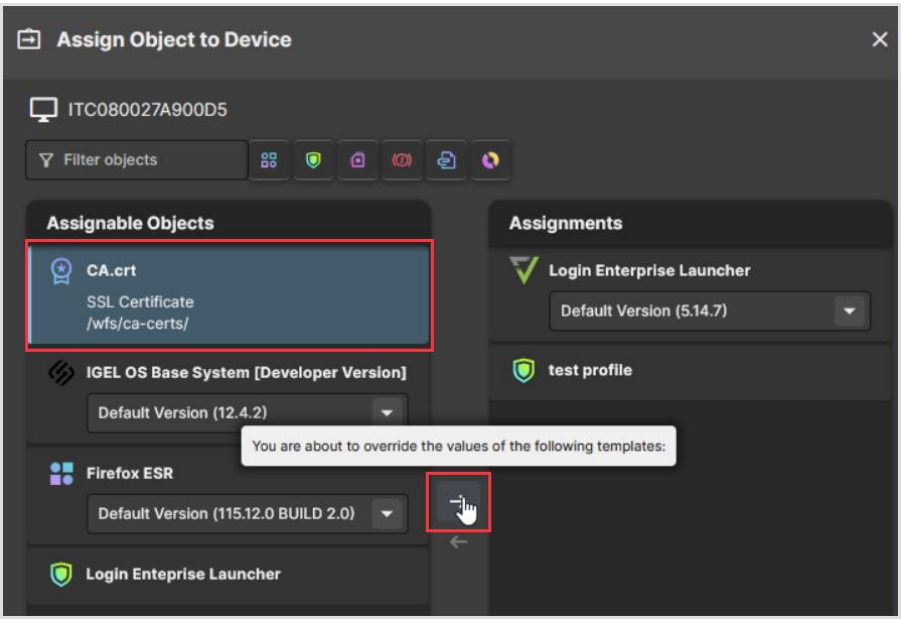
8. In the **UMS sidebar menu**, go to **Devices**, and select the IGEL device you want to be a Login Enterprise Launcher.



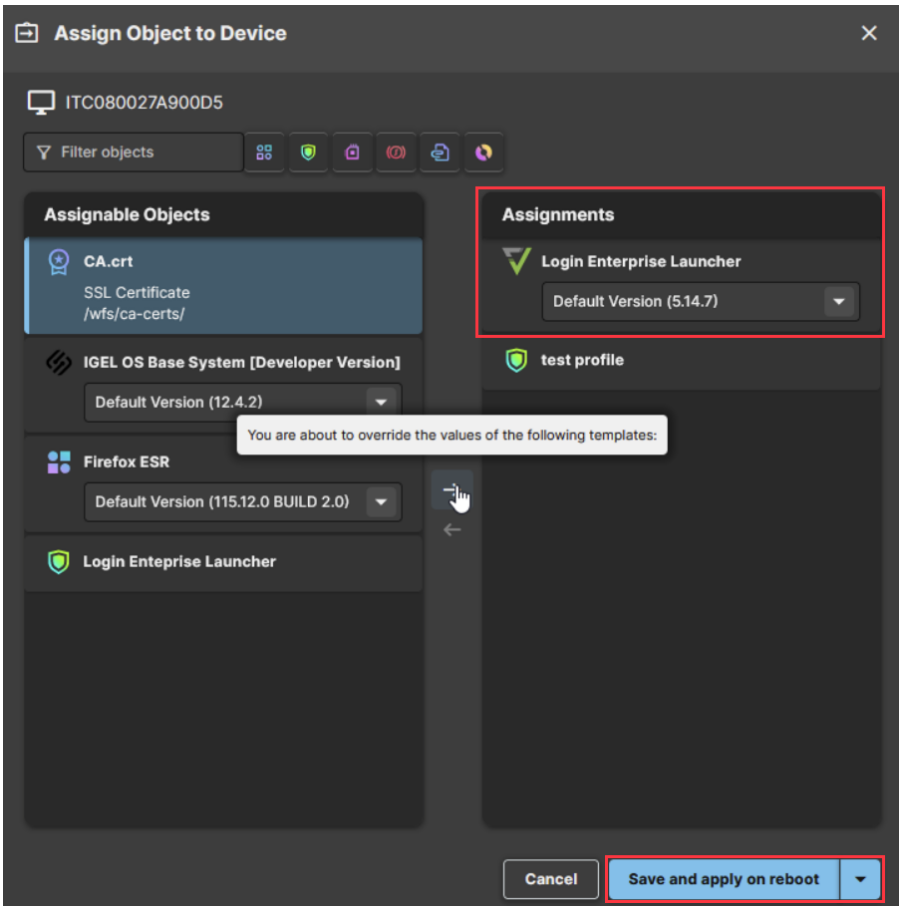
9. In the **top bar menu**, click **Assign Object**.



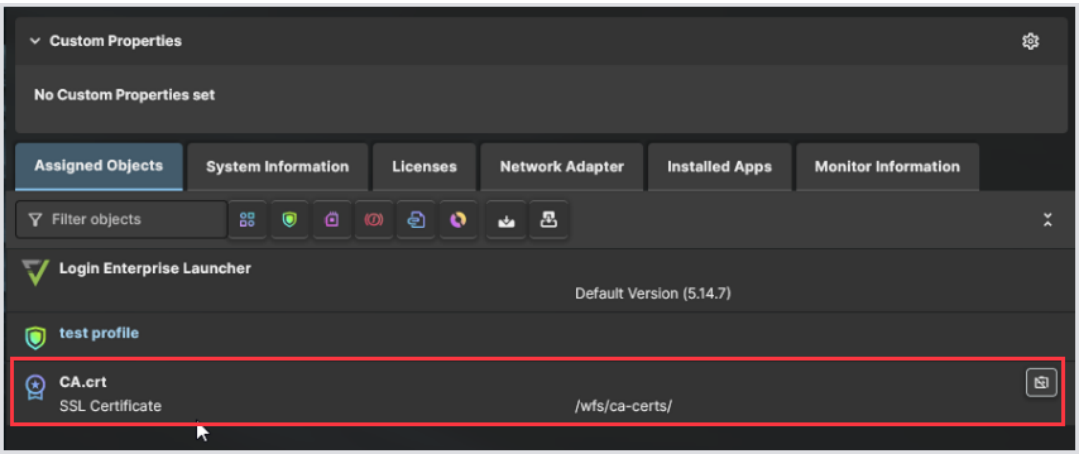
10. In **Assign Object to Device** > **Assignable Objects**, click on the `CA.crt`, then click on the right arrow.



11. Ensure the `CA.crt` is now in **Assignments**. Then, click **Save and apply on reboot**.



12. Click **Reboot**.
13. Once the device has rebooted, refresh the page to confirm that the license is now applied to the Device.



Confirming the License is Applied to the Device (Optional)

To verify that the license is correctly applied to the device:

1. Console into the device.
2. Access the root shell by running the following command: `openssl x509 -in /wfs/ca-certs/CA.crt -text -noout`
3. Review the output in the terminal shell. You should see details about your Virtual Appliance certificate, such as the hostname.

For more information, see Terminals in IGEL OS.

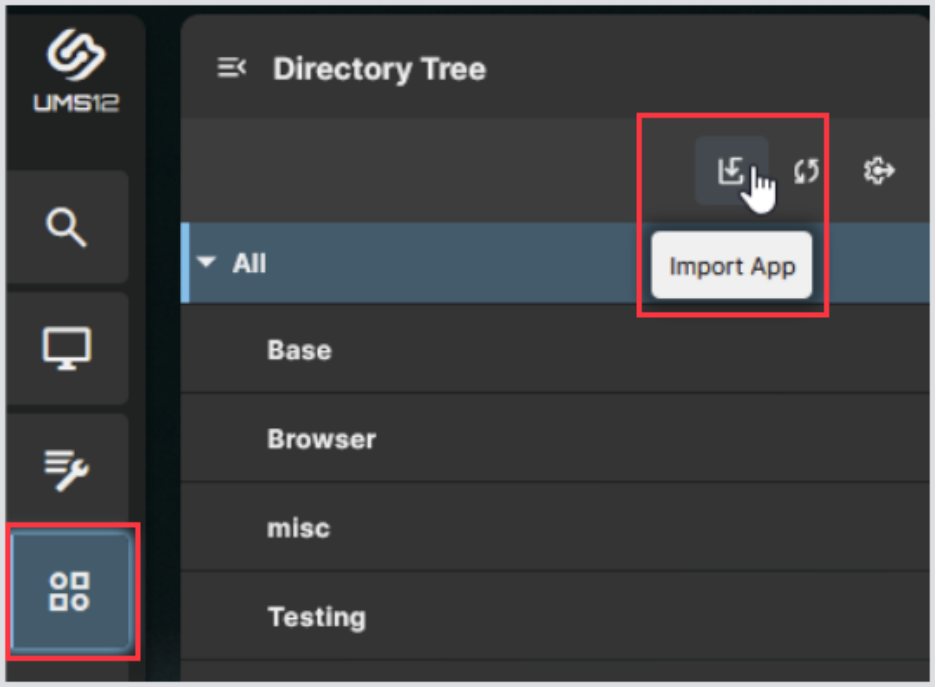
Installing the SDK certificates on the IGEL device

1. Download the following files: **`IGEL_OS_12_SDK-intermediate.crt`** and **`IGEL_OS_12_SDK-leaf.crt`**
2. Copy or transfer these files to the IGEL device that will act as the Login Enterprise Launcher. You can use one of the following two methods to get the files onto the device:
    1. Drive configuration for IGEL Device. For details, see Windows Drive Configuration in IGEL OS 12.
    2. Getting files into an IGEL Device through UMS Web. For details, see Upload and Assign Files in the IGEL UMS Web App.
3. Via a terminal in the IGEL Device create the directories for the certs. Run these commands:
   ```
   mkdir /wfs/cmty/im-certs
   mkdir /wfs/cmty/certs/
   ```
4. Copy the `IGEL_OS_12_SDK-intermediate.crt` into `/wfs/cmty/im-certs/`.
   For example, if you have `IGEL_OS_12_SDK-intermediate.crt` in `/tmp/` you would run the following command: `cp -r /tmp/IGEL_OS_12_SDK-intermediate.crt /wfs/cmty/im-certs/`
   Copy the `i` into `/wfs/cmty/certs/`. For example, `cp -r /tmp/IGEL_OS_12_SDK-leaf.crt /wfs/cmty/certs/`
5. Reboot the IGEL device and log in again once it has fully restarted.
6. Open a terminal again and run the following commands:
   ```
   cd /wfs/cmty/im-certs/
   dir
   ```
   If the `IGEL_OS_12_SDK-intermediate.crt` file appears in the terminal console output, the IGEL device is successfully trusting the certificate.
7. Run run the following commands:
   ```
   cd /wfs/cmty/certs/
   dir
   ```
   If the `IGEL_OS_12_SDK-leaf.crt` file appears in the terminal console output, the IGEL Device is successfully trusting this certificate.
8. Now the IGEL Device is ready to have the Login Enterprise Launcher installed on it via the IGEL UMS.

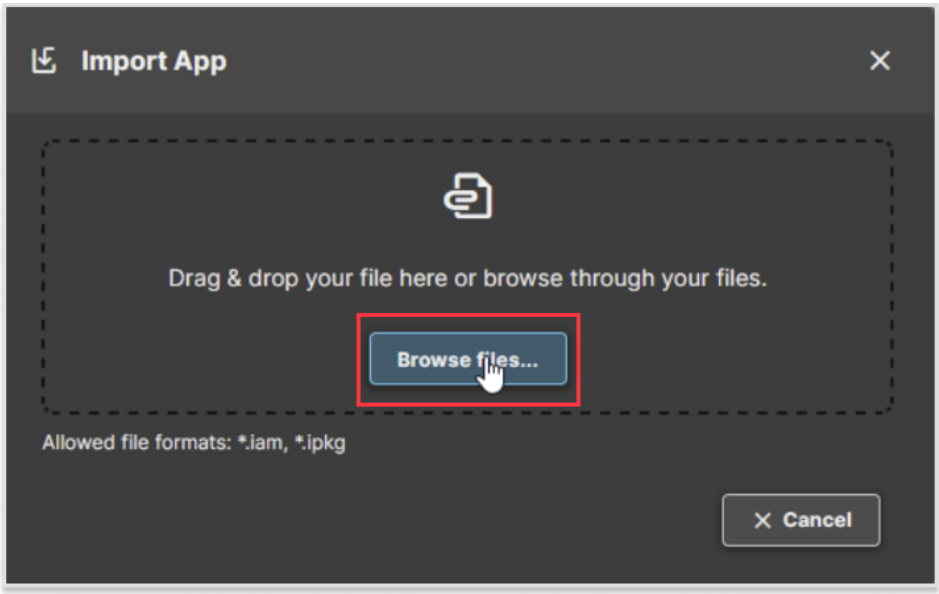
Manually downloading and getting the Launcher package on the IGEL UMS web

1. Download the Login Enterprise Launcher package according to your Login Enterprise virtual appliance version. Then, extract this file to get the `.ipkg` file inside
2. Open and log in to the IGEL UMS Web console.

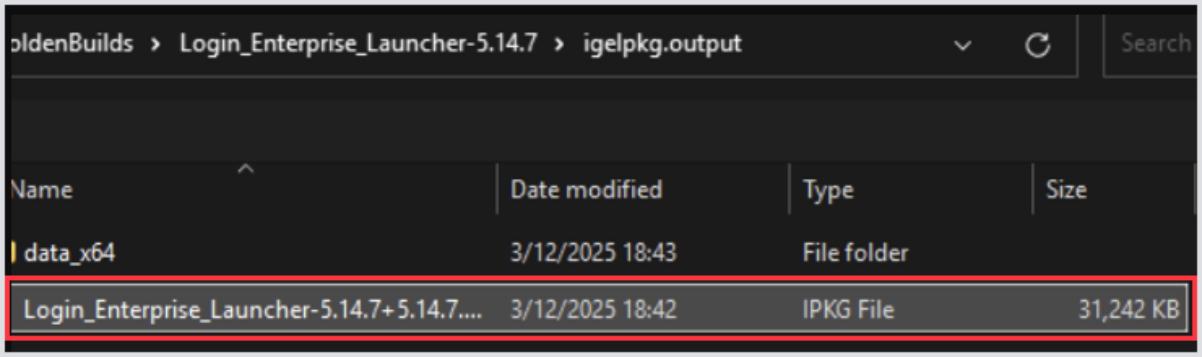
3. In the **UMS sidebar menu**, navigate to **Apps** and then click **Import App**.



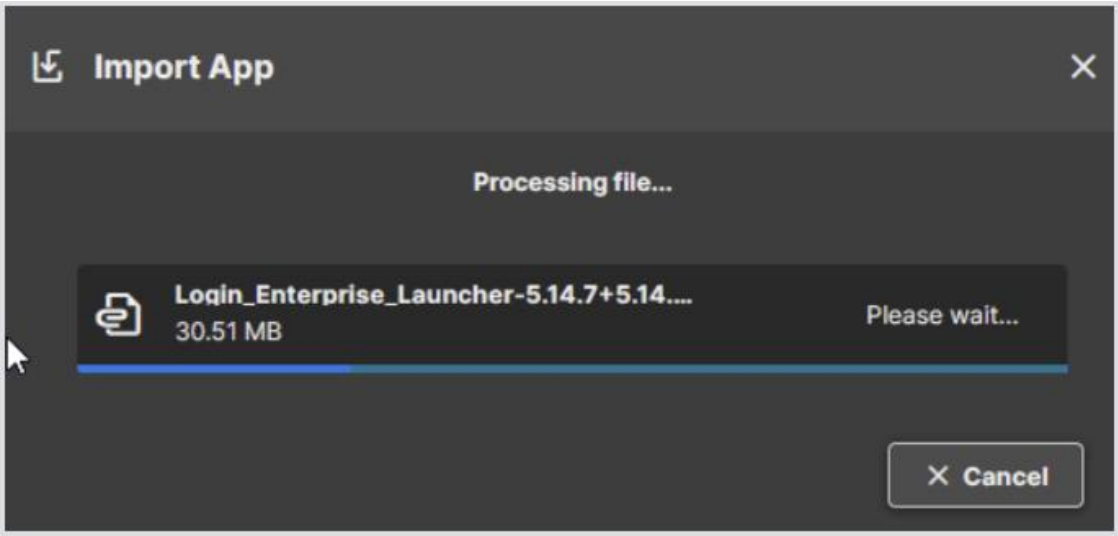
4. In **Import App**, click **Browse files**.



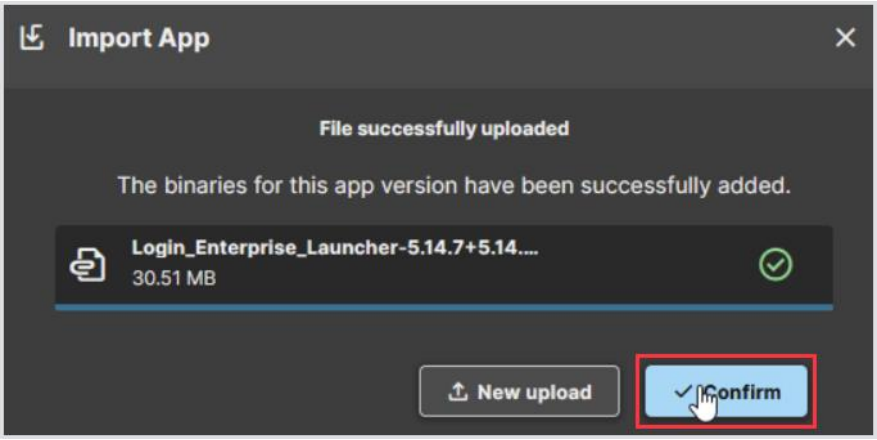
5. Browse and select the Login Enterprise Launcher `.ipkg` file.



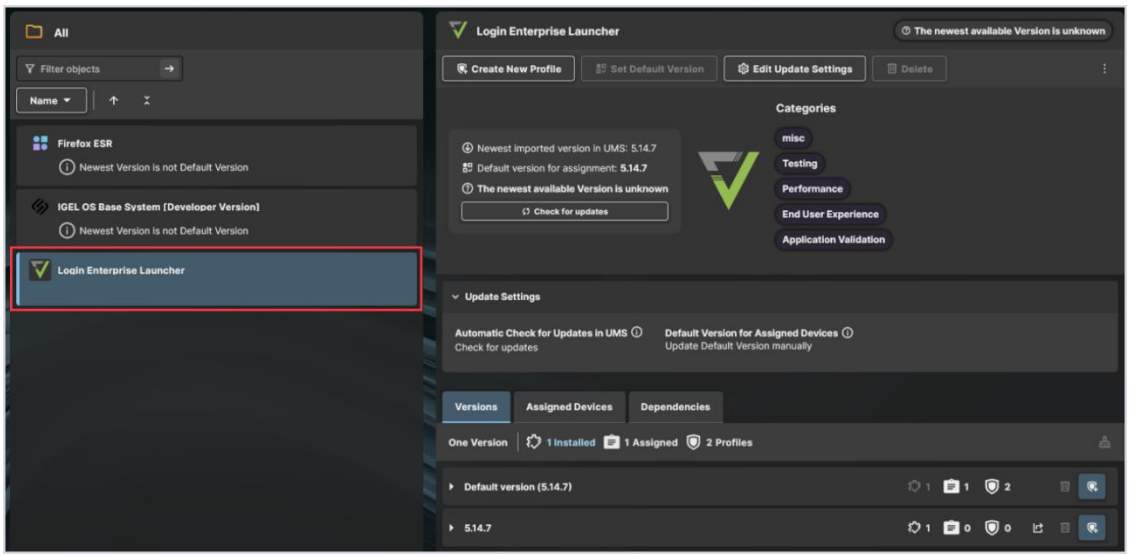
6. The **Import App** popup will show that the file is being uploaded.



7. After uploading, click **Confirm**.



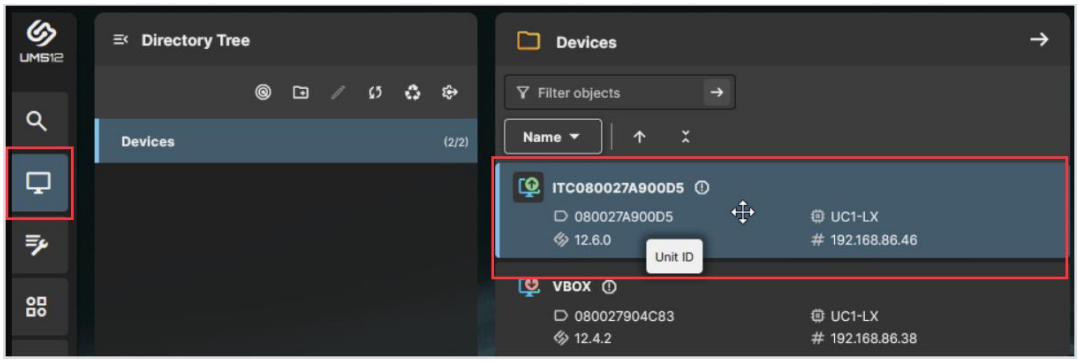
8. The Login Enterprise Launcher application should now be visible in the IGEL UMS Web interface.



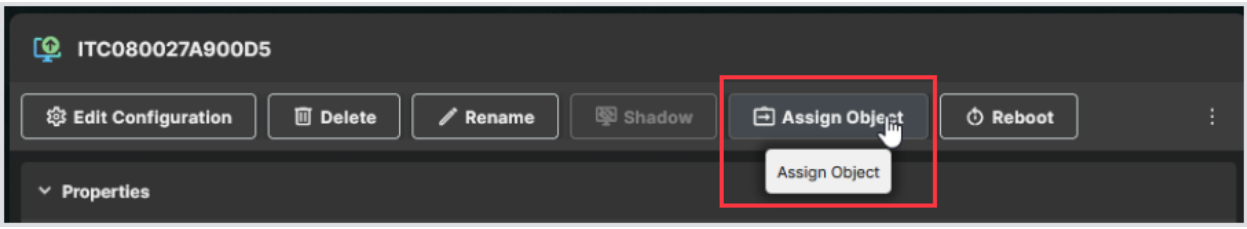
Getting the Launcher onto IGEL devices

Follow these steps regardless of whether you got the Login Enterprise Launcher into the UMS manually or via the App Portal.

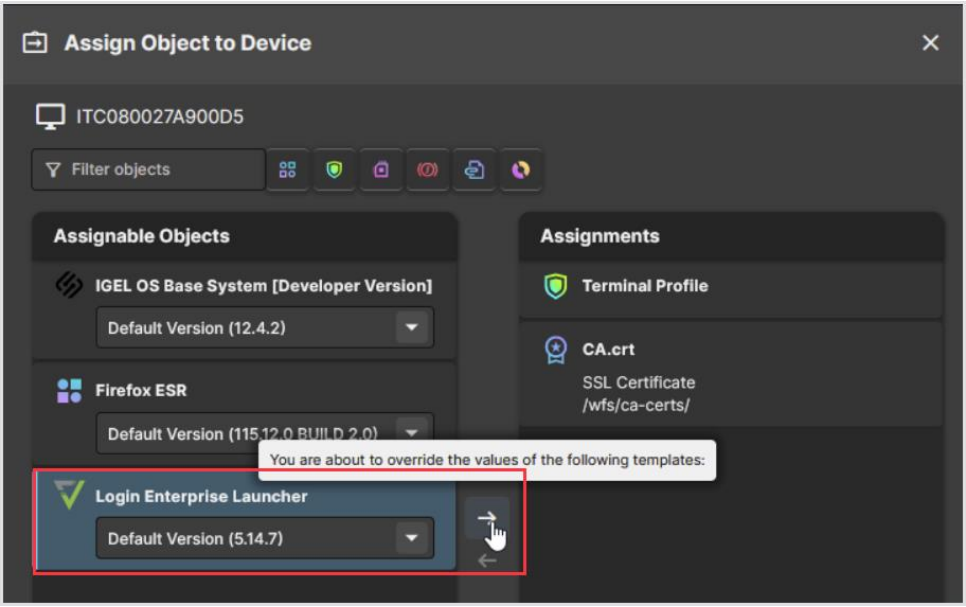
1. In the **UMS sidebar menu**, go to **Devices**, and select the IGEL Device you want to be a Login Enterprise Launcher.



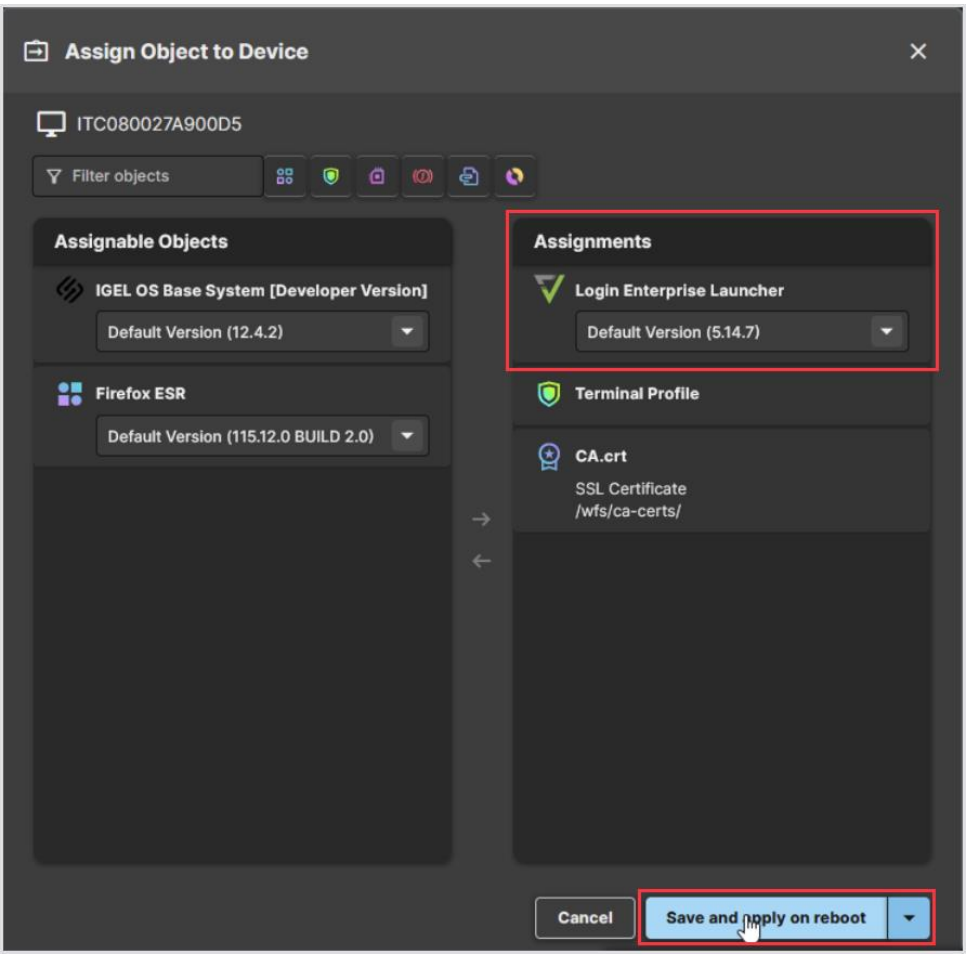
2. In the **top bar menu**, click **Assign Object**.



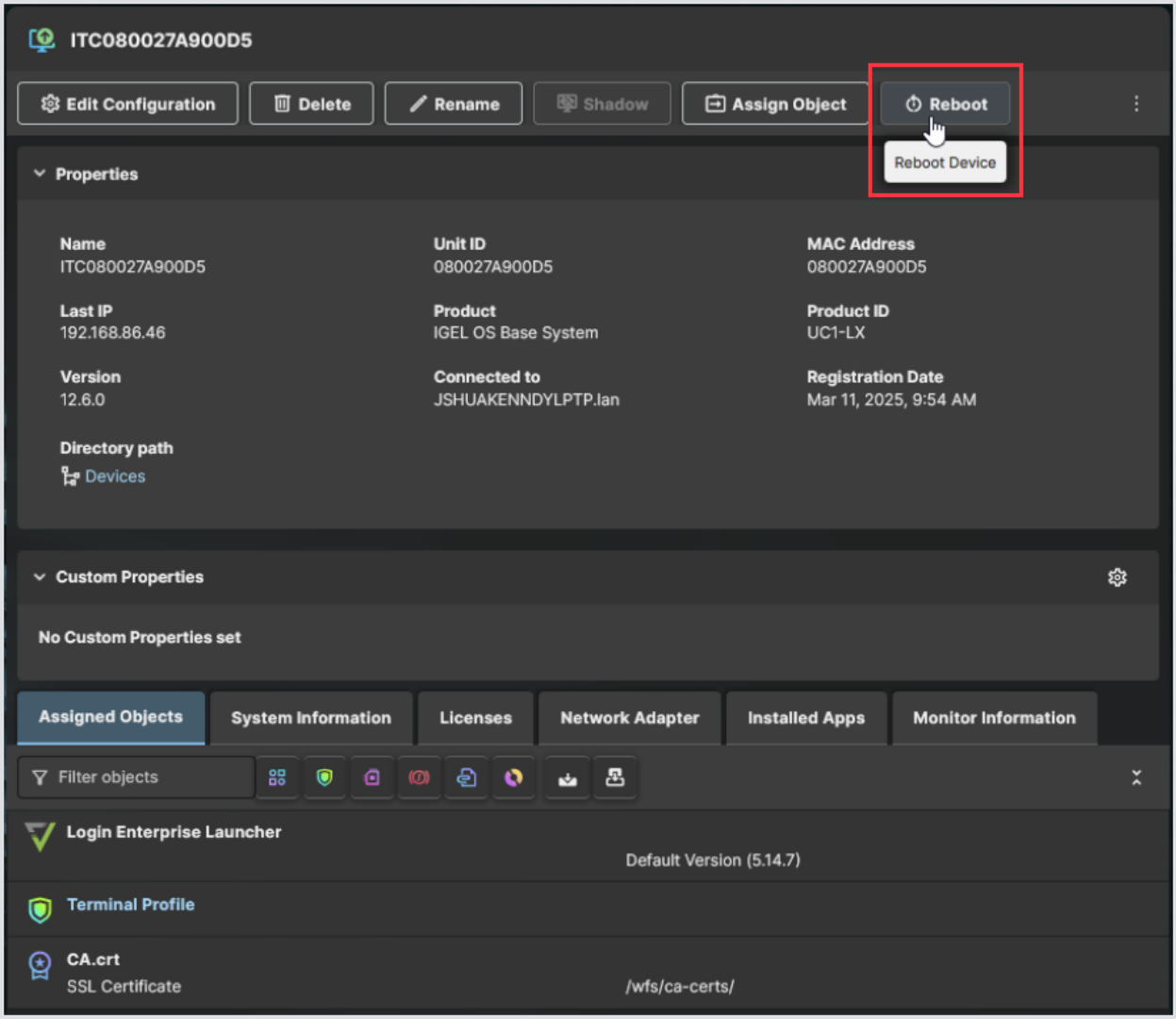
3. In **Assign Object to Device** > **Assignable Objects**, click on the Login Enterprise Launcher, and then click on the right arrow.



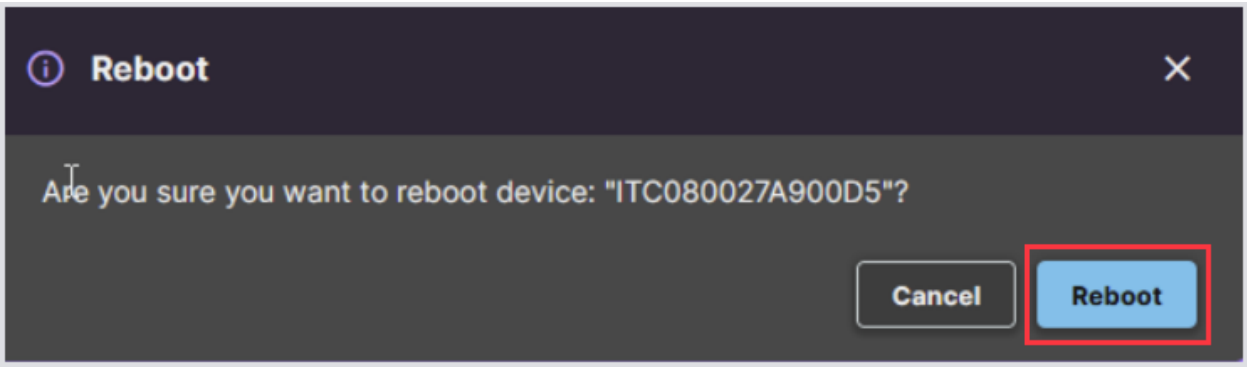
4. Ensure the Login Enterprise Launcher is in **Assignments**. Then, click **Save and apply on reboot**.



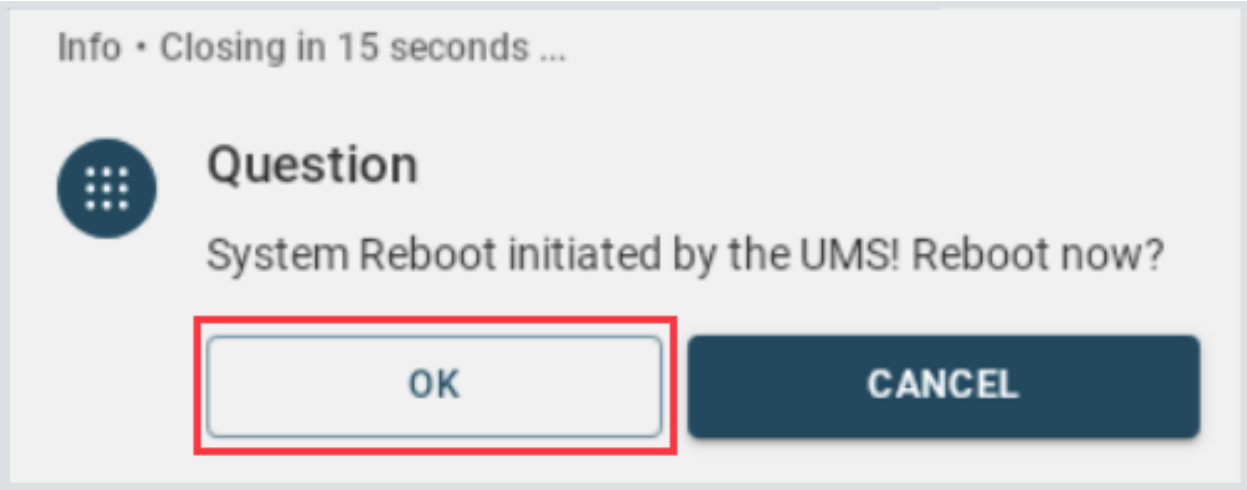
5. Click **Reboot**.



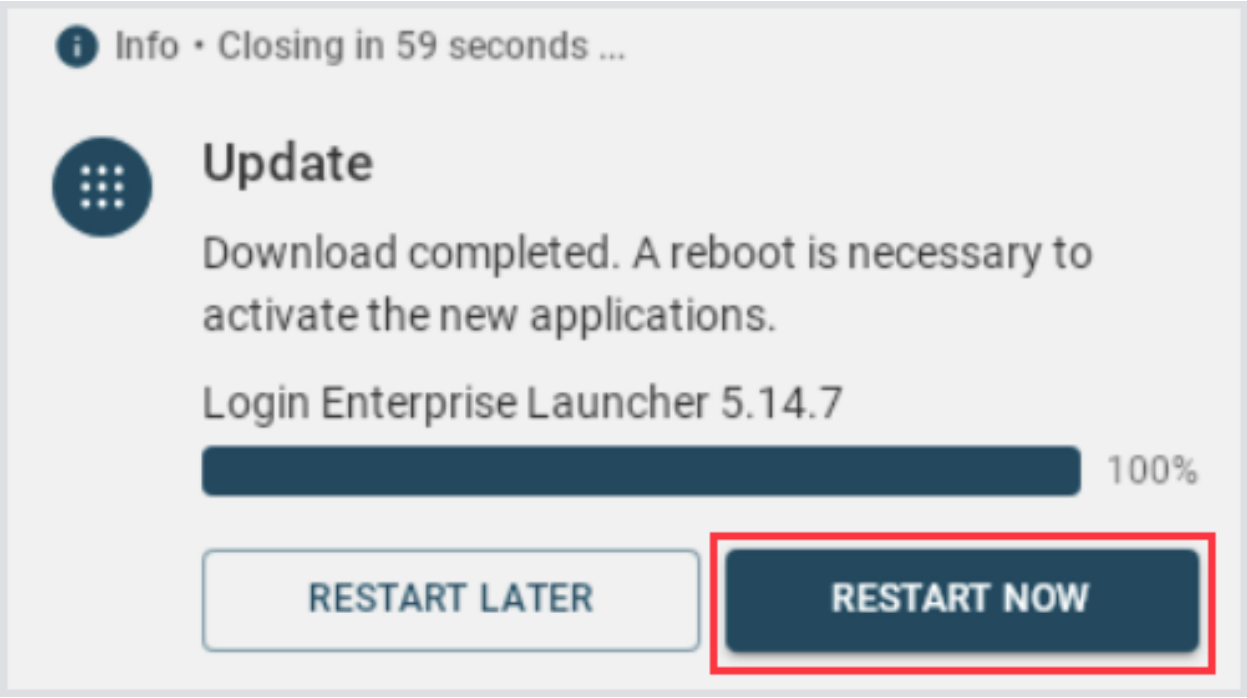
6. Confirm the **Reboot** action.



7. On the IGEL device where the Login Enterprise Launcher is being installed, click **OK** on the System Reboot pop-up initiated by UMS. This will speed up the installation process.



8. After the IGEL device reboots, a pop-up will appear, indicating that the download is complete. Click **Restart Now** to speed up the installation process.



9. After the IGEL device reboots, open a terminal and type the following command:
   `/services/Login_`
   Press the **Tab** on your keyboard to automatically complete the path.
   If `/services/Login_Enterprise_Launcher/` appears in the terminal, the Login Enterprise Launcher has been installed.

```
Local Terminal
login as "user" or "root": user
user@ITC080027A900D5:~$ /services/Login_Enterprise_Launcher/
```

10. Download the `appsettings.json` file.

In a text editor, edit the `appsettings.json` file

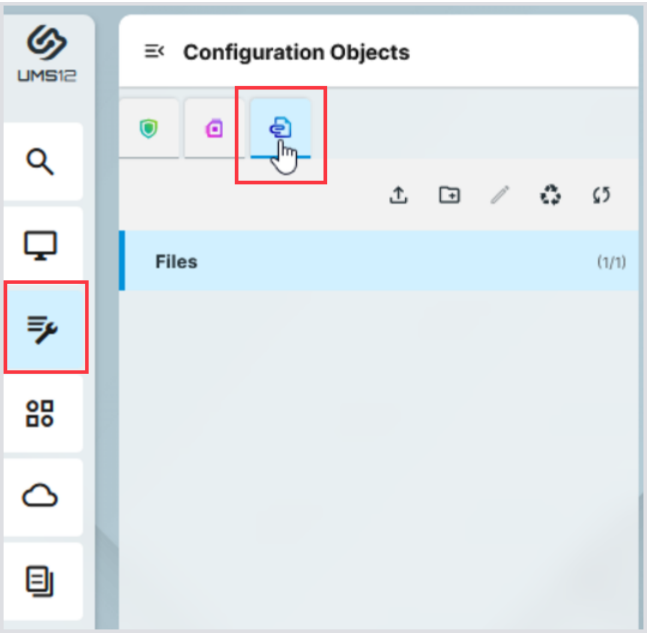
1. Find *ServerUrl* and enter the value for your Login Enterprise appliance's web URL, such as https://myDomain.myAppliance.com – note without the slash at the end, like how it is here:

```
"ServerUrl":  "https://myLoginEnterpriseAppliance.myDomain.com",
```

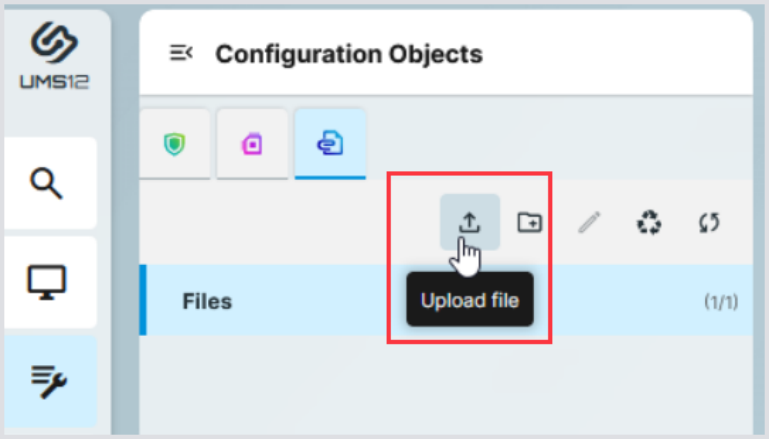
2. Find the **ClientSecret** and place it in the Launcher secret value. To copy the **ClientSecret**, in the **Login Enterprise sidebar menu**, navigate to **Launchers > Downloads**.
3. Below is how the **ClientSecret** will appear in the `appsettings.json` file once you've added your actual **ClientSecret**. The placeholder value is shown here:

```
"ClientSecret":  "39Q3R54C4PI0Y17ZA45B55PCFPA9PMW4OKILLK1E",
```

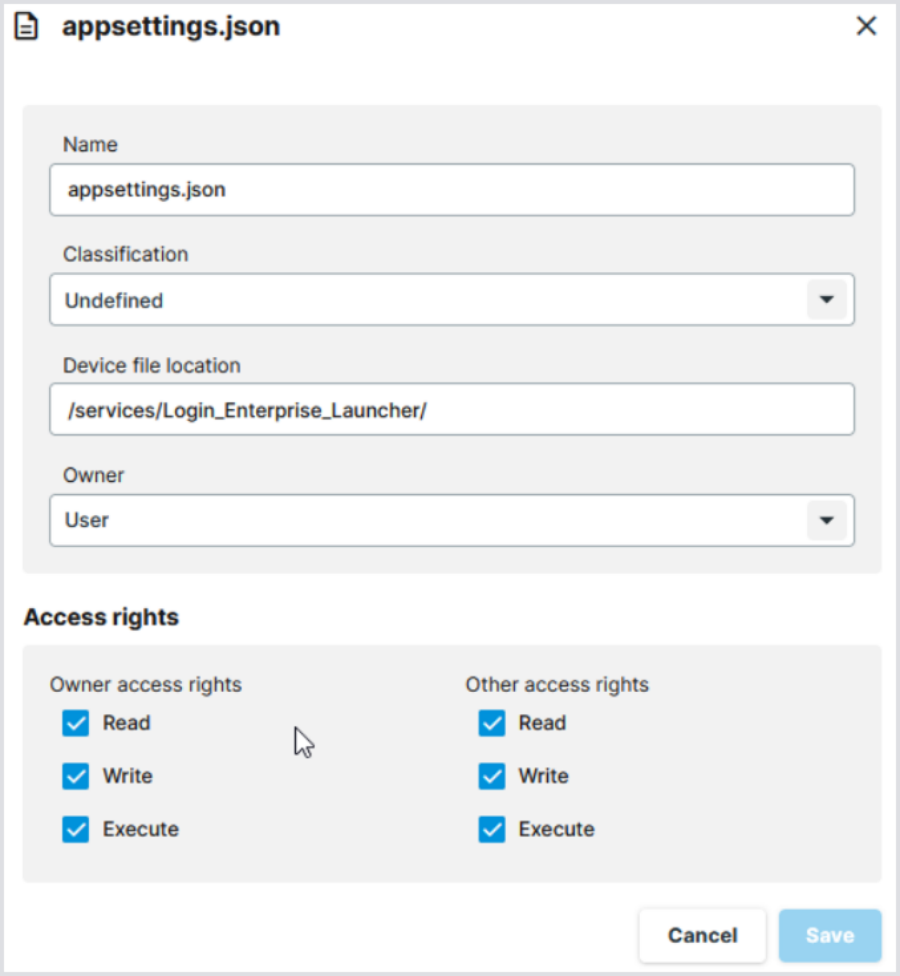
11. Save the **appsettings.json** file
12. In the **UMS Web Console**, navigate to **Configuration**, and then click **Files**.



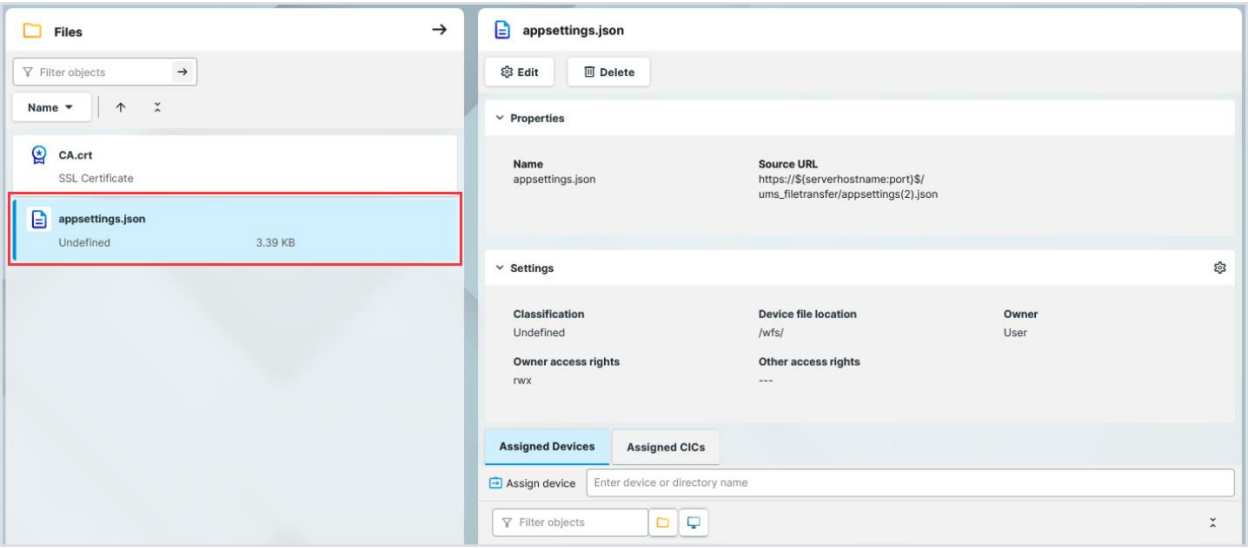
13. In **Files**, click **Upload file**.



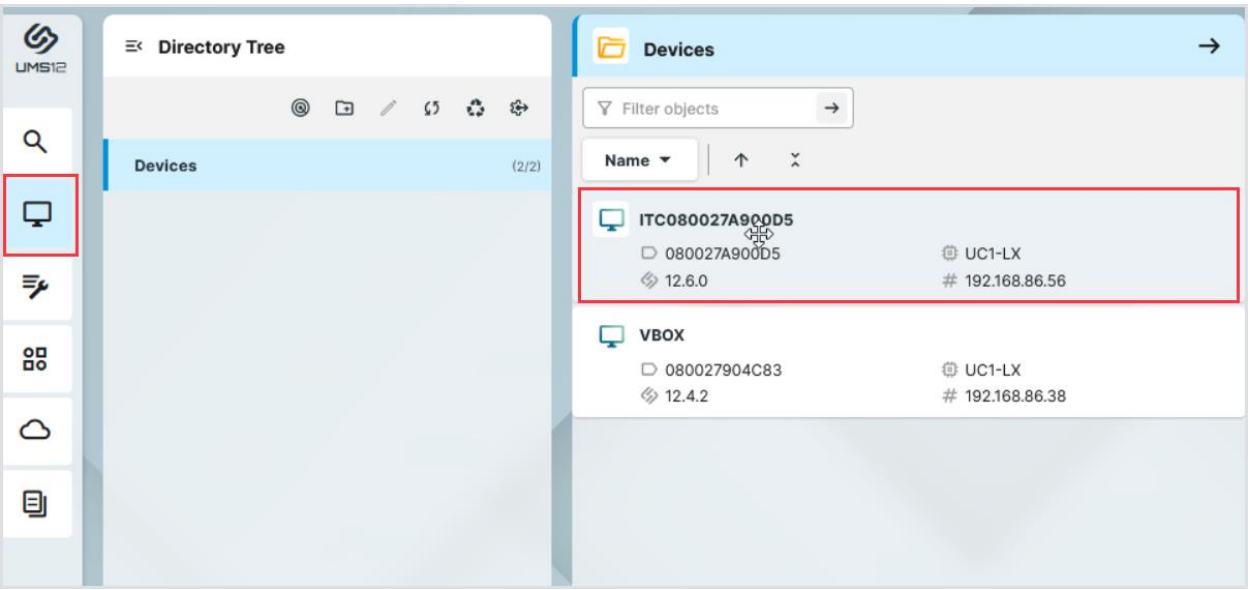
14. In the **Upload File** popup, the **Name** will remain as `appsettings.json`. Set the **Classification** to **Undefined**, the **Device File Location** to **/services/Login_Enterprise_Launcher**, and the **Owner** to **User**. Ensure to enable all the Access rights via the checkboxes, then click **Save**.



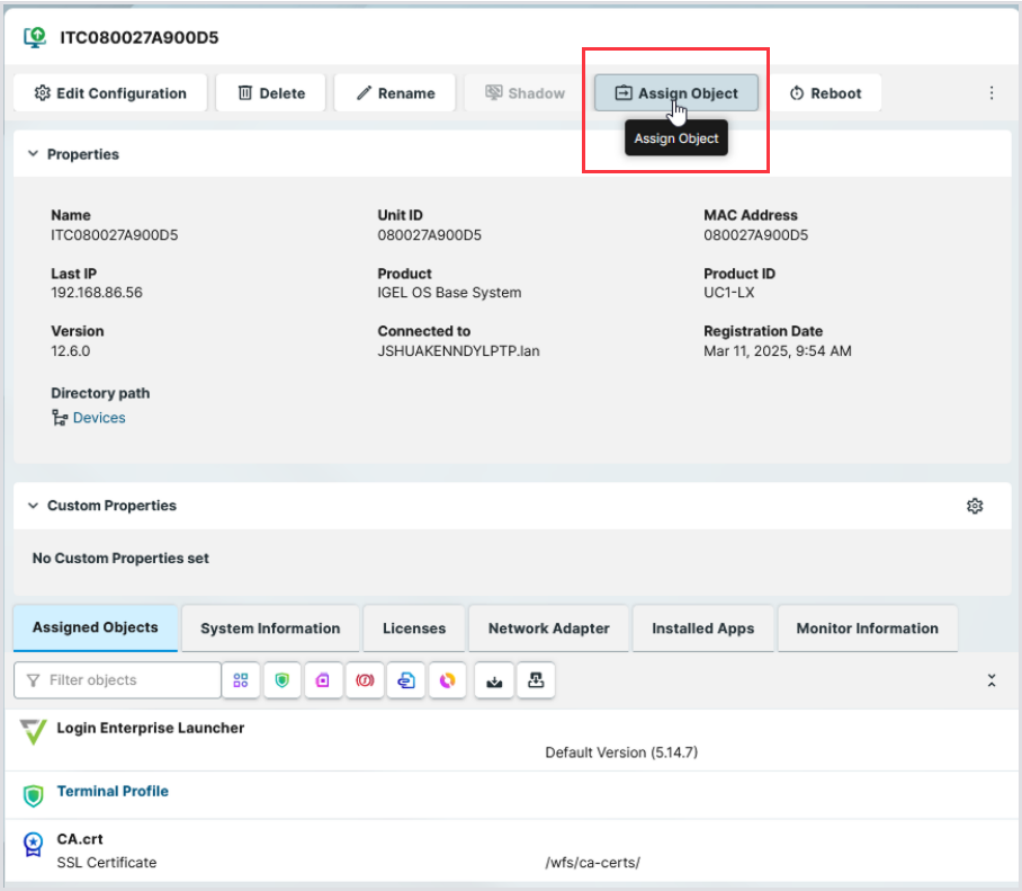
15. The `appsettings.json` will be visible now in the UMS Web interface.



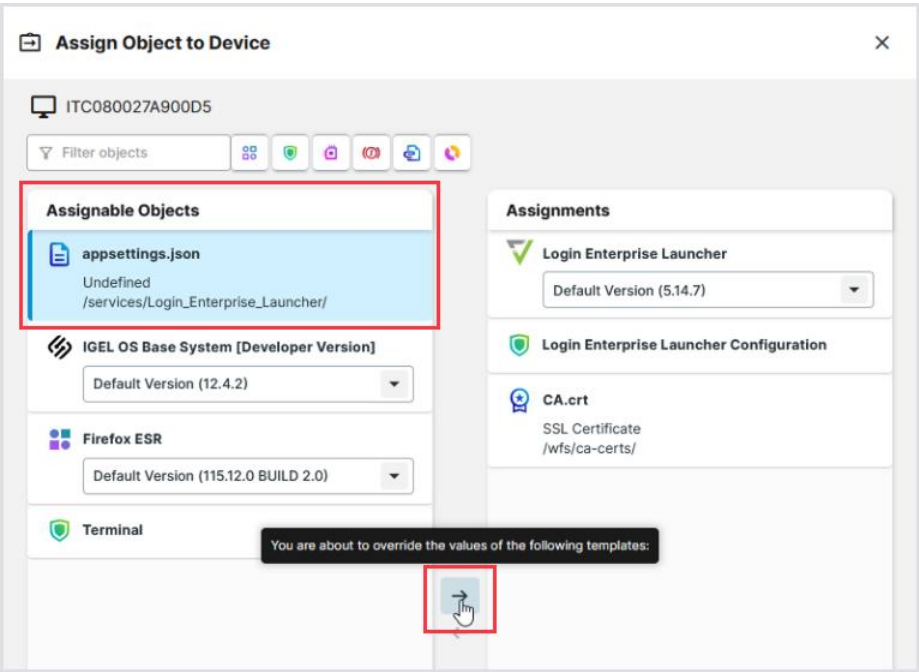
16. In the **UMS sidebar menu**, go to **Devices** and select the IGEL device.



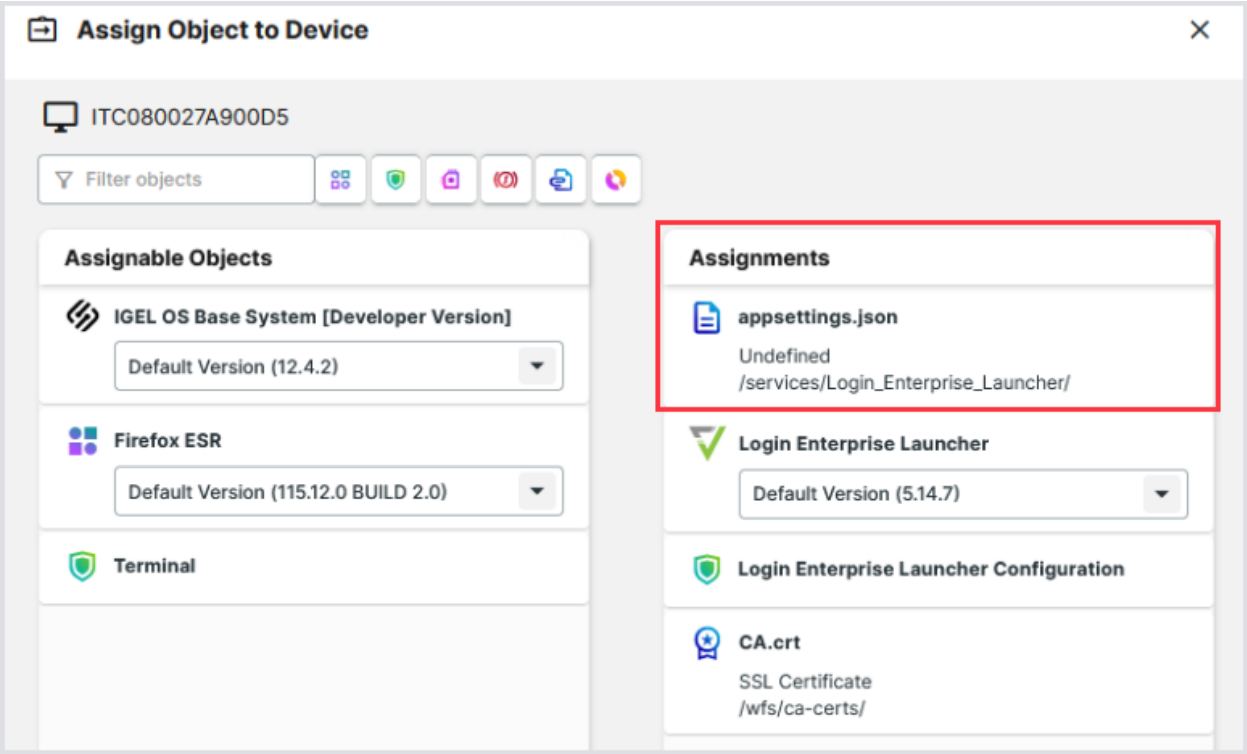
17. In the **top bar menu**, click **Assign Object**.



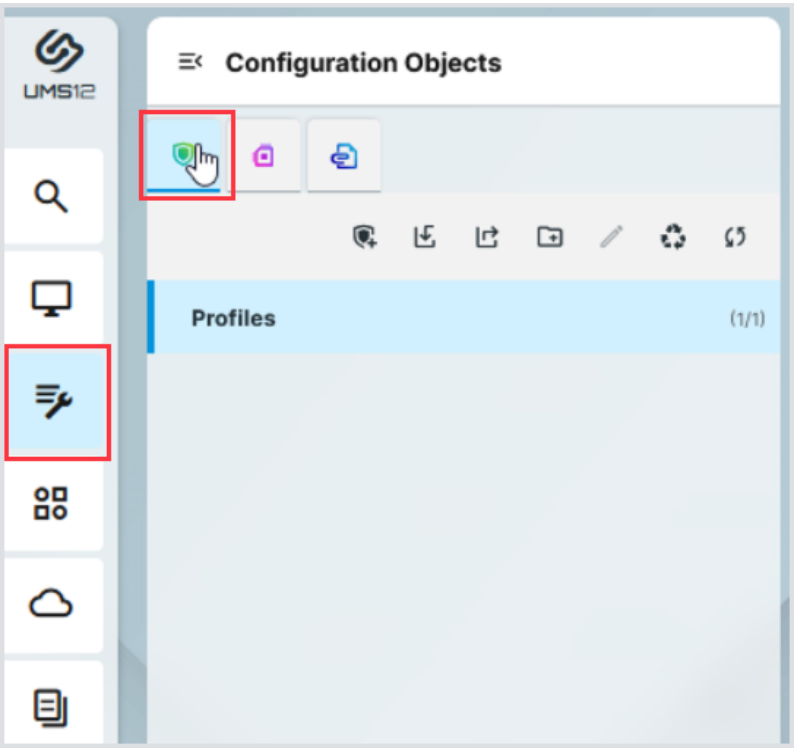
18. In **Assign Object to Device** > **Assignable Objects**, click on `appsettings.json`, and then click the right arrow.



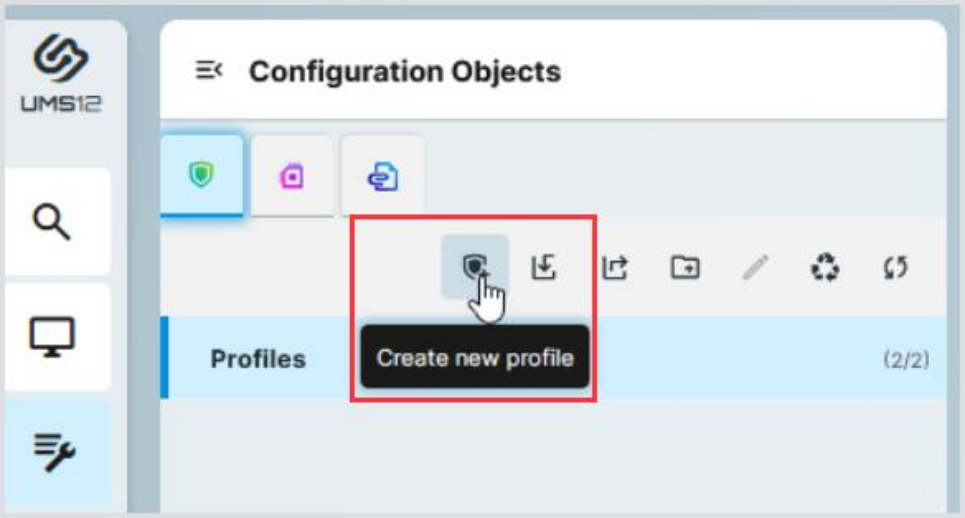
19. Ensure the `appsettings.json` is in **Assignments**.



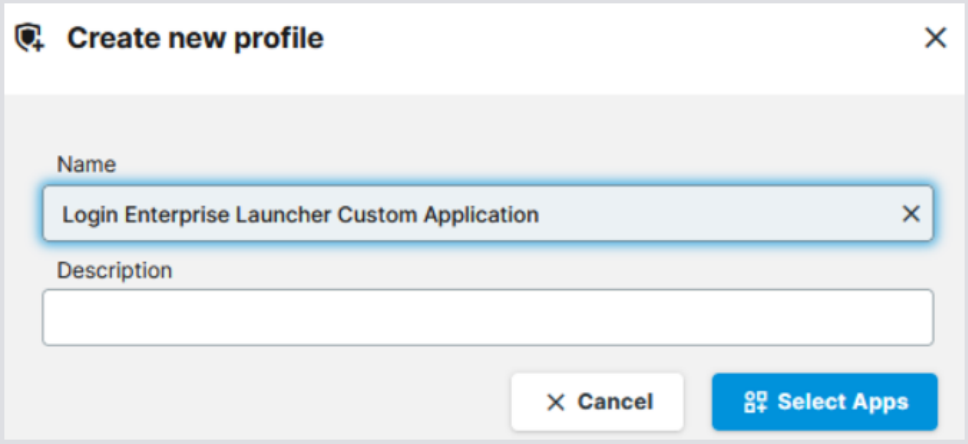
20. In the **UMS sidebar menu**, go to **Configuration > Profiles**.



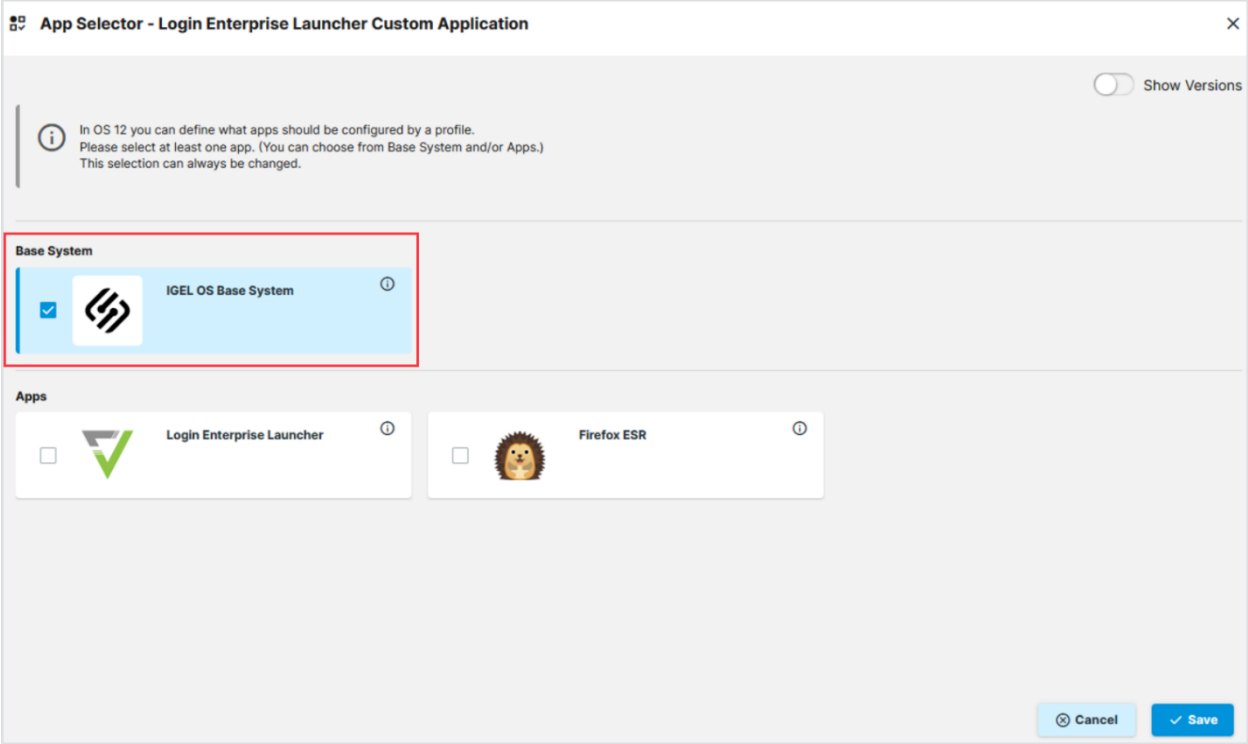
21. In **Profiles**, click **Create new profile**.



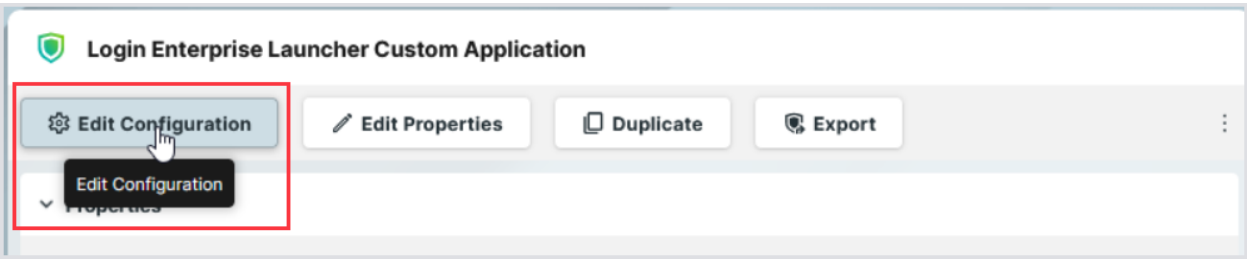
22. In **Create New Profile**, give the new profile the name **Login Enterprise Launcher Custom Application**, and then click **Select Apps**.



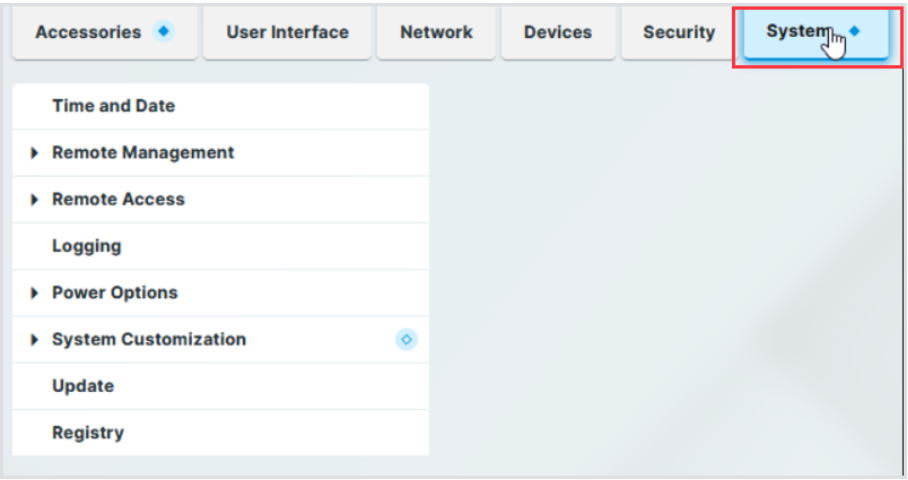
23. In the **App Selector**, click on **IGEL OS Base System** and then **Save**.



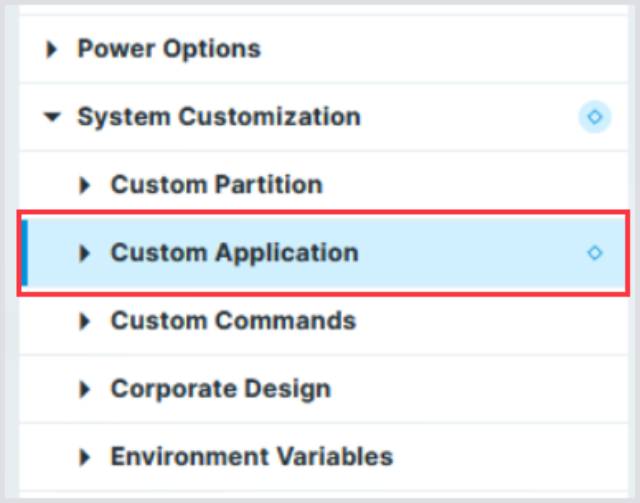
24. In **Profiles**, click **Edit Configuration**.



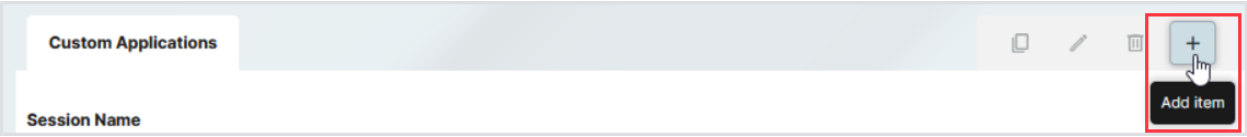
25. In **Profile Configurator**, click **System**.



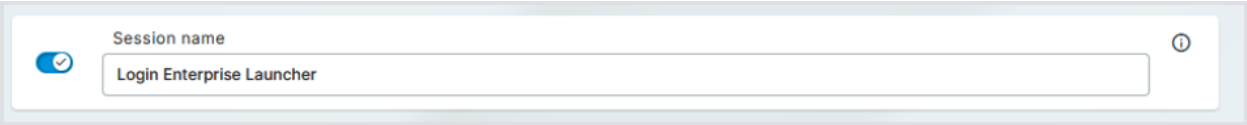
26. Expand **System Customization** and click on **Custom Application**



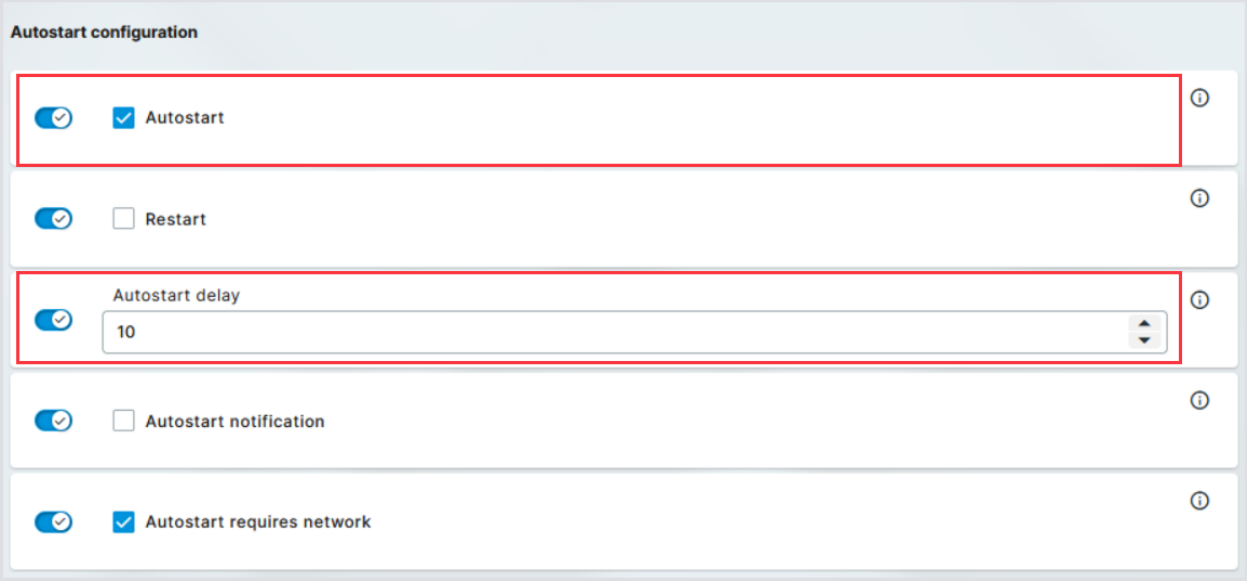
27. In **Custom Applications**, click **Add Item**.



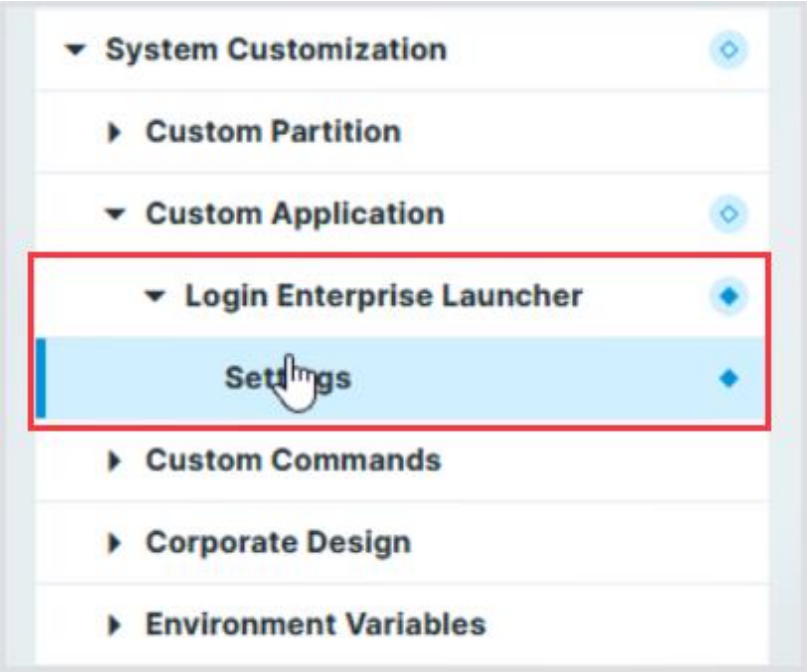
28. In **Session name**, fill in the field with **Login Enterprise Launcher**.



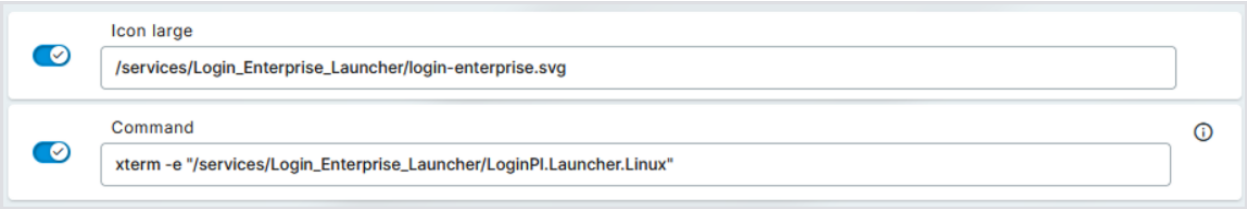
29. Scroll down to find the **Autostart** configuration. Enable **Autostart** to ensure the Login Enterprise Launcher starts automatically when the IGEL device boots. Set the **Autostart delay** to 10 seconds. This is the recommended configuration.



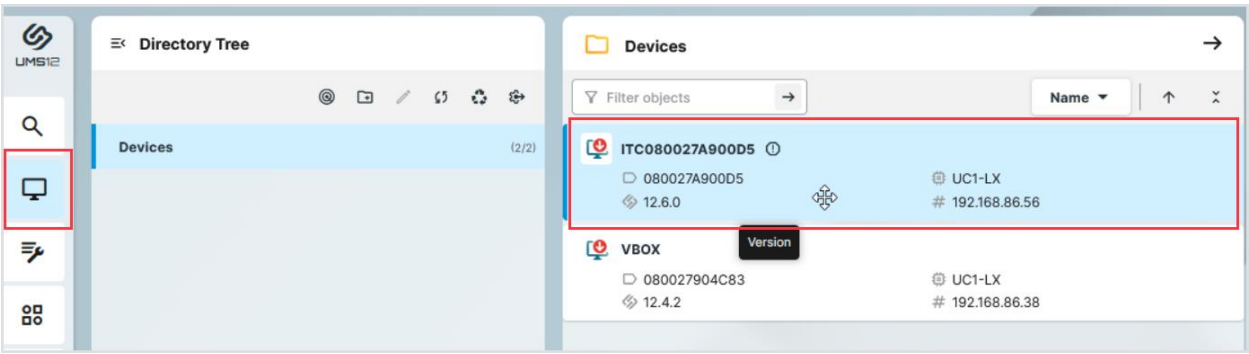
30. Expand the **Login Enterprise Launcher** in the left pane and click **Settings**.



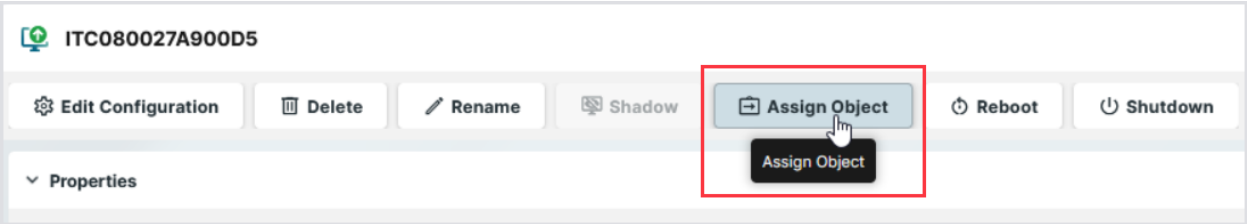
31. Ensure that **Icon large** and **Command** are both enabled.
    1. In **Icon large**, input: `/services/Login_Enterprise_Launcher/login-enterprise.svg`
    2. In **Command**, input: `xterm -e "/services/Login_Enterprise_Launcher/LoginPI.Launcher.Linux"`



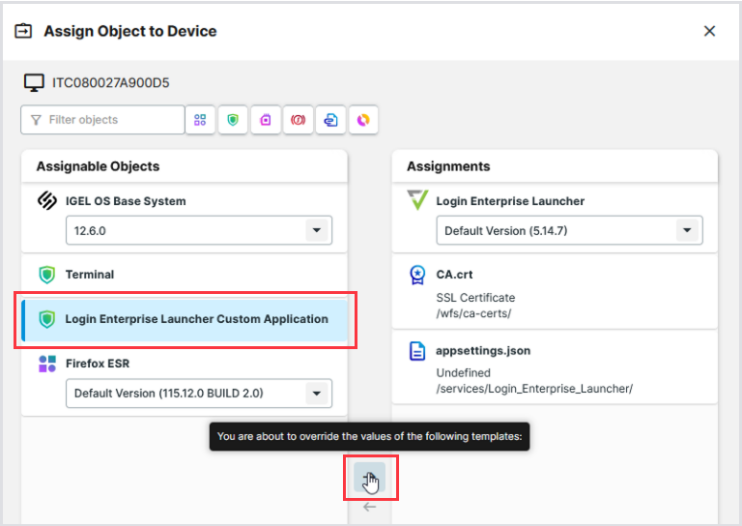
32. Click **Save and Close**.
33. In the **UMS Web sidebar menu**, navigate to **Devices** and select the IGEL Device.



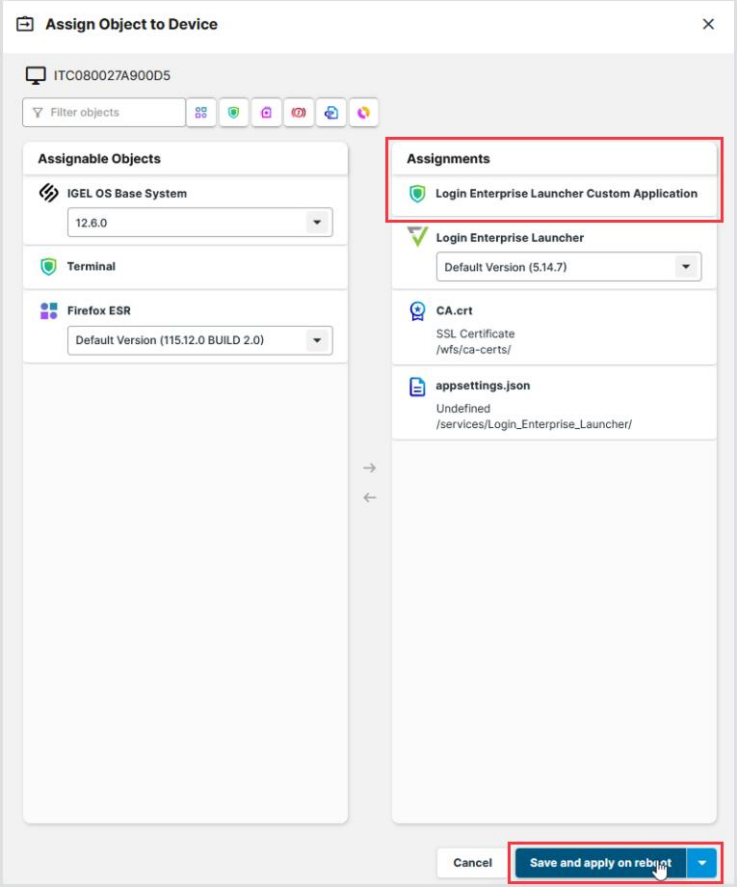
34. In the **top bar menu**, click **Assign Object**.



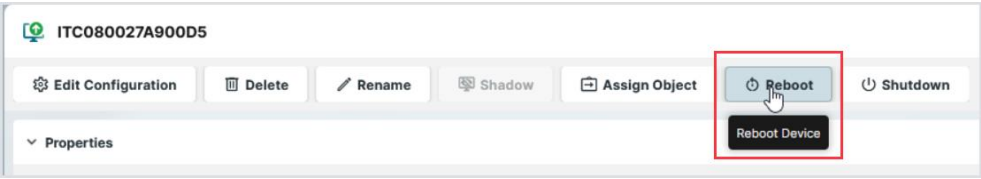
35. In **Assign Object to Device** > **Assignable Objects**, click **Login Enterprise Launcher Custom Application**, and then click the right arrow.



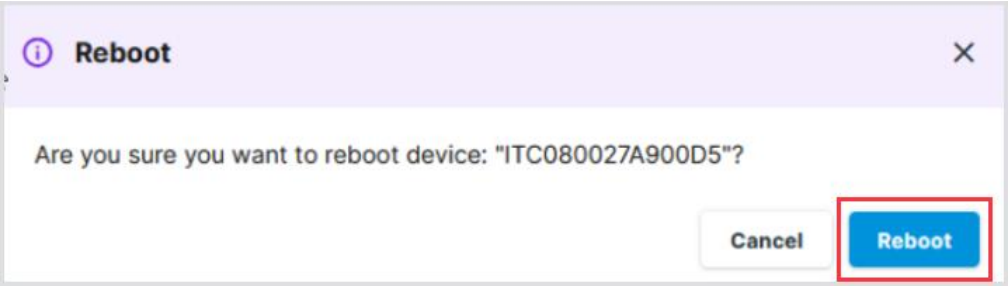
36. Ensure the **Login Enterprise Launcher Custom Application** is in the **Assignments**. Then, click **Save and apply on reboot**.



37. Click **Reboot**.



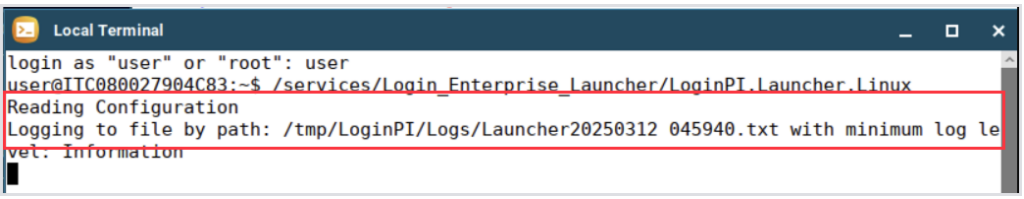
38. Confirm the reboot, then reboot the IGEL device as before to speed up the process.



After rebooting, the IGEL device may display a secondary reboot confirmation, indicating that the device has changed. If this happens, click **Reboot**.

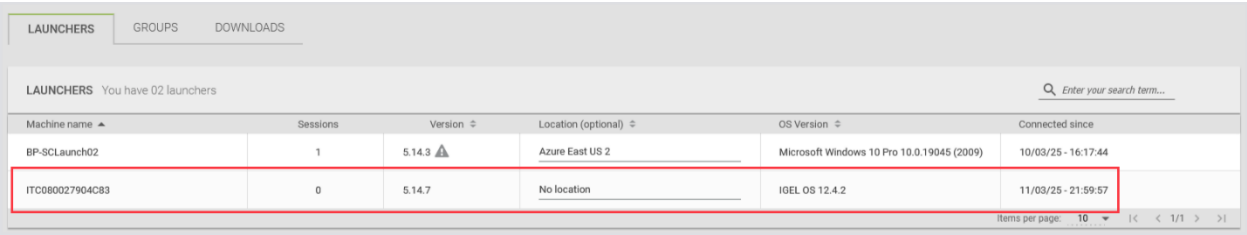
Running the Launcher

1. If Autostart was configured in the previous steps, the Virtual Appliance Launcher can be invoked by logging into the IGEL device.
2. When the Login Enterprise Launcher runs and successfully connects to the Virtual Appliance, it will display a message indicating it's waiting for a Test to run. You will see the status "Reading Configuration" and a log path being displayed.



3. Alternatively, you can start the Virtual Appliance Launcher by clicking the desktop icon or selecting it from the Application Launcher on the IGEL device.

4. To verify that the Login Enterprise Launcher is successfully connected to the Virtual Appliance, in the **Login Enterprise sidebar menu**, go to the **Launchers**, and check that the IGEL device's machine name is listed.



The Login Enterprise Launcher is now ready to be used as a Launcher for the Test!

Updating the Login Enterprise Launcher

This is necessary when updating the Login Enterprise Virtual Appliance, for example, after updating Login Enterprise 5.14 to 6.0. If the Login Enterprise Launcher is version 5.14 and the

Login Enterprise virtual appliance is version 6.0, the Login Enterprise Launcher won't launch Test sessions. There must be parity. Below are the steps to update the Login Enterprise Launcher.

1. Open in a text editor the *appsettings.json* file you set up in step #10 in the [Getting the Launcher onto IGEL devices] section.
2. Update the *LauncherVersion* key to reflect the Login Enterprise version update and save the file. For example:

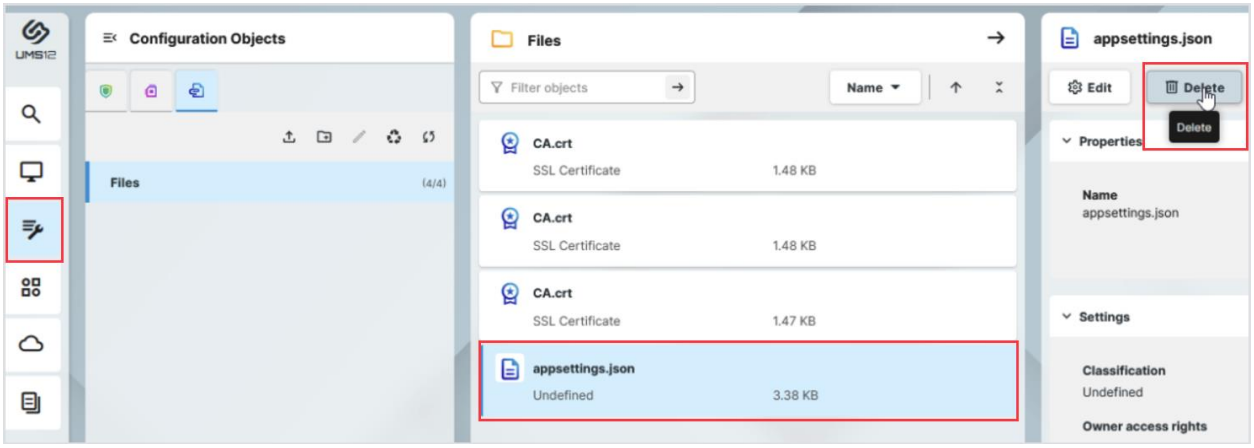
- Before the update:

```
"LauncherVersion":  "5.14.7",
```

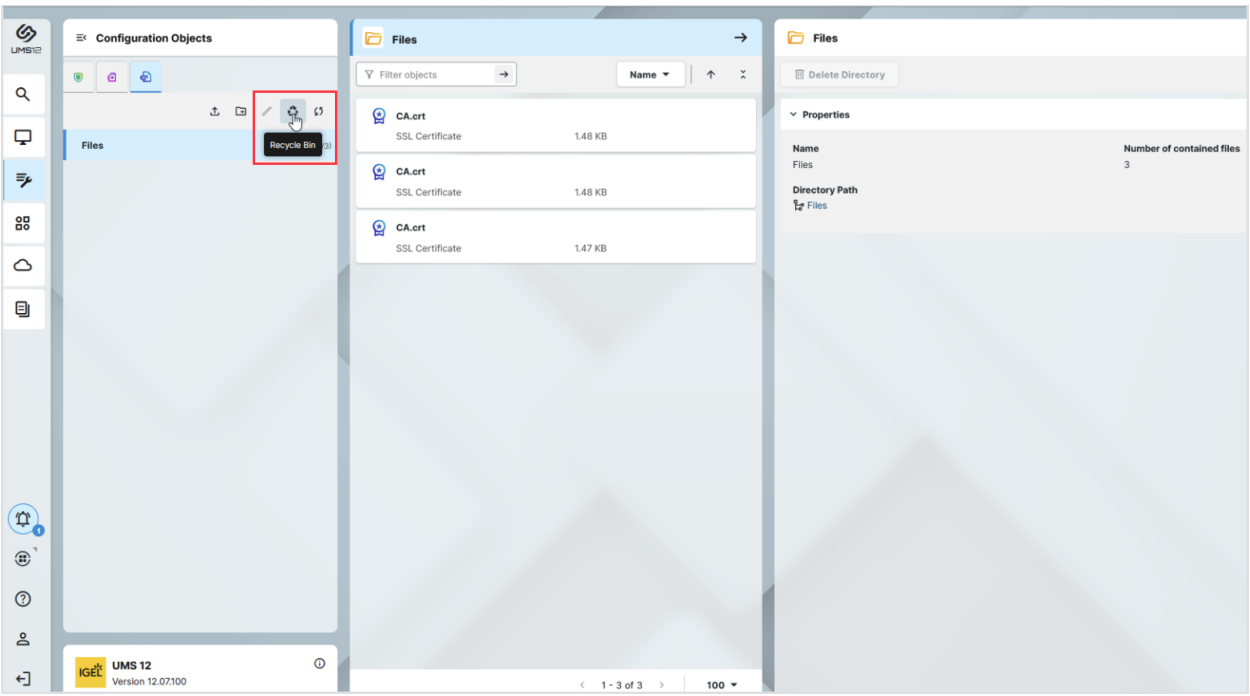
- After the update:

```
"LauncherVersion":  "6.0.14",
```

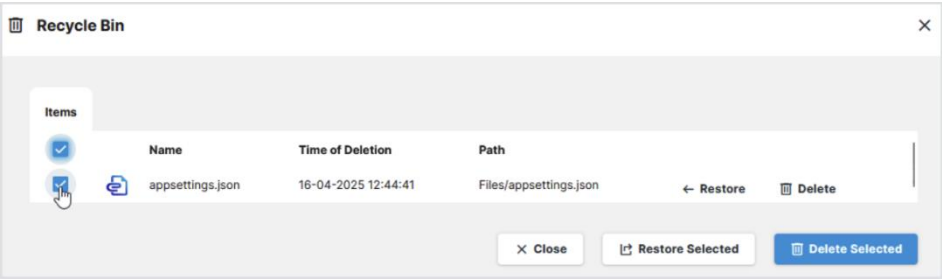
3. In the **UMS Web interface**, navigate to **Configuration Objects > Files** > Select the existing *appsettings.json* file, and click **Delete**.



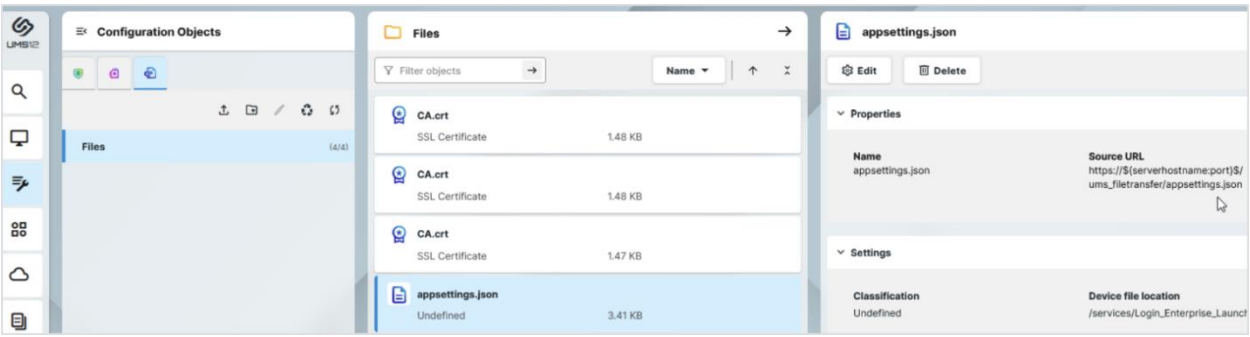
4. Observe that the appsettings.json file is no longer in the **Files** pane, then click the Recycle Bin.



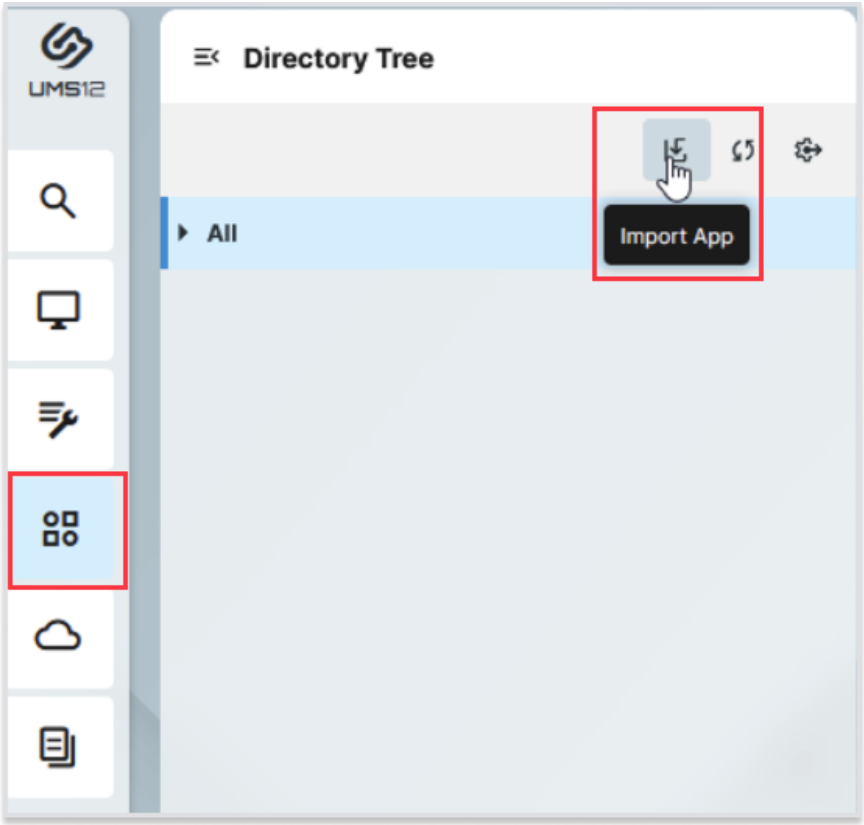
5. In the **Recycle Bin**, click on *appsettings.json* and then **Delete Selected**.



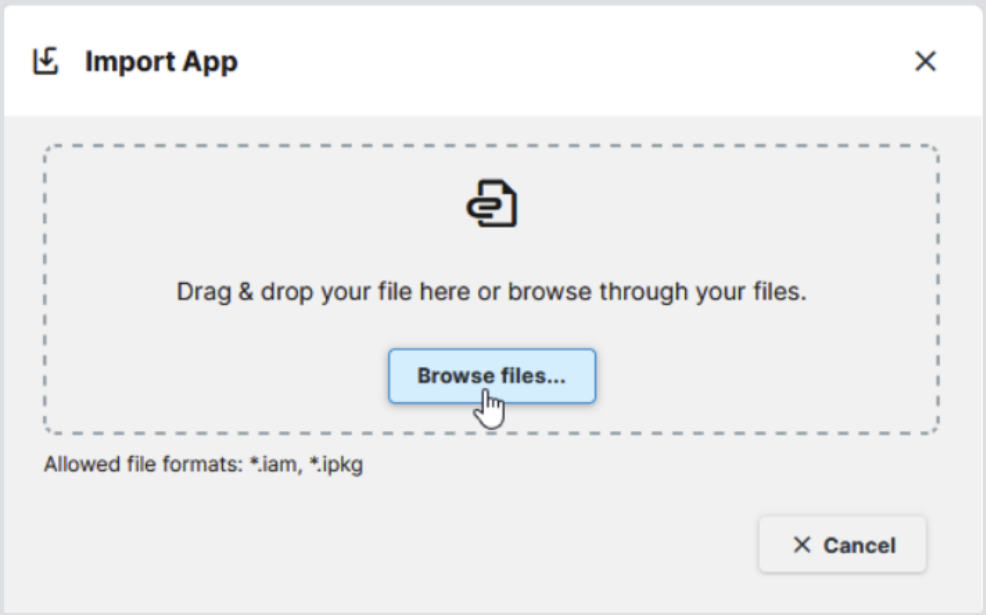
6. To upload the *appsettings.json* file modified with the new Login Enterprise version and add it to the IGEL OS device(s) using the Login Enterprise Launcher, perform the steps starting with step #12 in Getting the Launcher onto IGEL devices.
7. When the updated *appsettings.json* file is re-uploaded, ensure in the UMS Web that the *appsettings.json* Source URL ends with plain appsettings.json, and not *appsettings(1).json.*



8. Download the update Login Enterprise Launcher package for IGEL according to your Login Enterprise virtual appliance version.
9. In **UMS Web**, navigate to **Apps**, and then click **Import App**.

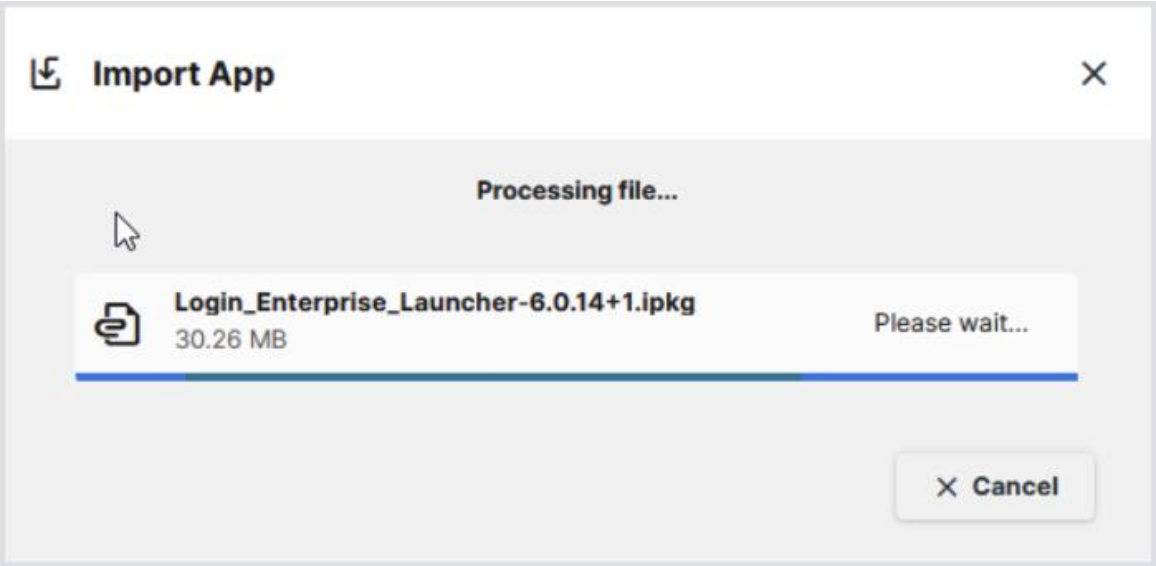


10. In **Import App**, click **Browse files**.

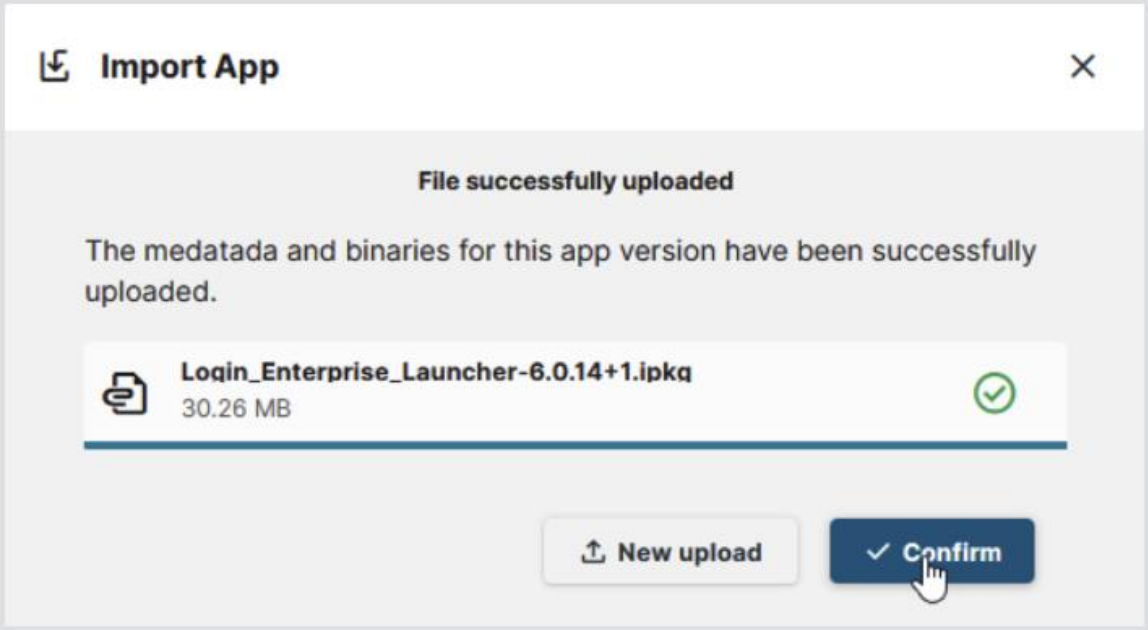


11. Select the newly downloaded package in the file browser.

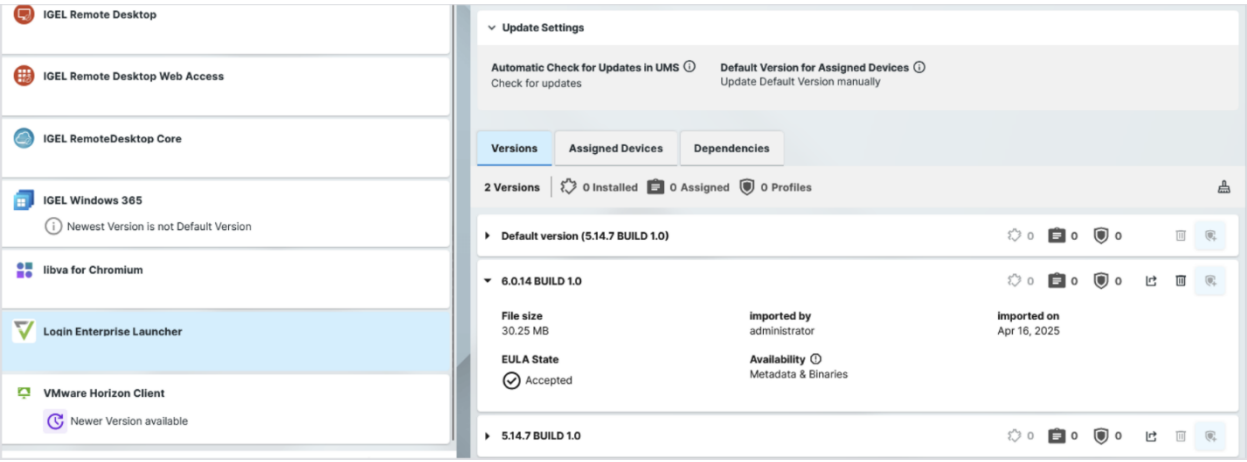
12. In **Import App**, observe the package getting uploaded.



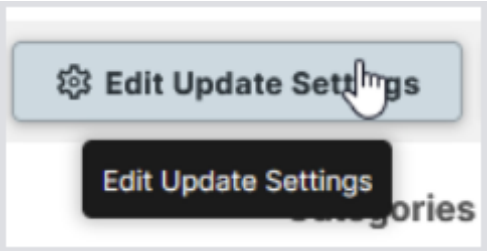
13. In **Import App**, click **Confirm** once the package has been uploaded.



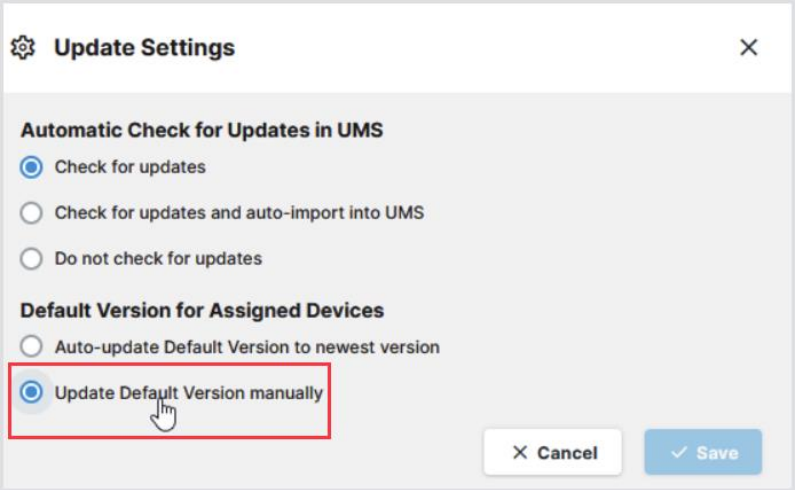
14. In the **UMS Web interface**, observe that the new Login Enterprise Launcher version is shown in the Versions tab of the App.



15. In the **UMS Web interface for the Login Enterprise Launcher App**, click **Edit Update Settings**.

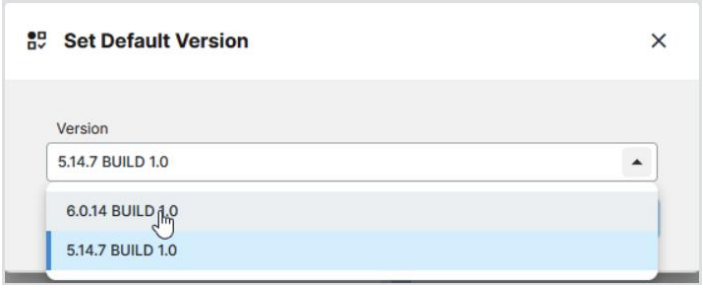


16. In **Update Settings**, if not already selected, click **Update Default Version Manually**, then **Save**.

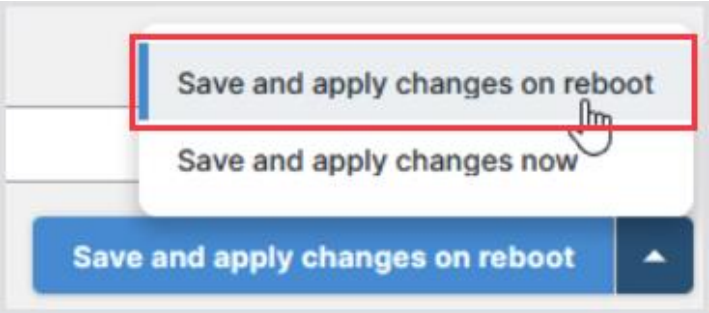


17. In the **UMS Web interface for the Login Enterprise Launcher App**, click **Set Default Version**.

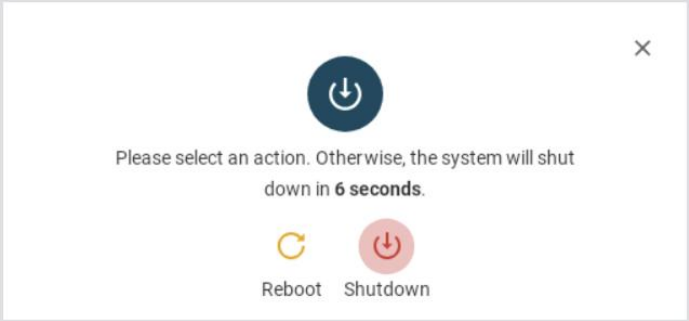
18. In **Set Default Version**, select the new version of the Login Enterprise Launcher from the drop-down.



19. In **Set Default Version**, click **Save and apply changes on reboot**.



20. Initiate a reboot on the affected IGEL device(s) that have the Login Enterprise Launcher installed.



Once the reboot has completed, the Launcher can be used again for Login Enterprise testing!

Launcher Connections in IGEL UMS 12

- Overview
- Prerequisites for Successful Target Connections
- Connection Setups
  - Microsoft Connections
  - Citrix Connections
  - Omnissa Connections

Overview

This section guides you through setting up the Login Enterprise Launcher on IGEL OS 12 so that it connects to a Login Enterprise Windows target desktop. Whether you run Tests or prepare your environment for live deployments, these steps and best practices will ensure your configuration supports successful testing. By the end of this document, you'll be equipped to optimize your IGEL 12 environment for reliable connectivity to target desktops such as Citrix, Omnissa, or RDS, ultimately empowering you to achieve a smooth testing experience.

Prerequisites for Successful Target Connections

1. **Existing Target Desktops**: Ensure the target desktops are set up so the Launcher can run connections.
2. **Broker Configuration**: Brokers for the published target resources must be properly configured.
3. **Launcher Setup**: Complete the Login Enterprise Launcher setup on IGEL OS 12. For instructions, see Deploying Login Enterprise Launcher in IGEL UMS 12.
4. **Citrix Connections**: Install the "Citrix Workspace App" on the IGEL device if connecting to Citrix-published resources.
5. **RDS Connections**: Install "IGEL Remote Desktop" on the IGEL device if connecting to RDS resources.
6. **Omnissa Connections**: Install "VMware Horizon Client" on the IGEL device if connecting to Omnissa resources.
7. **Connectivity Verification**: Confirm that the Launcher has verified connectivity to the intended target desktops/resources.
8. **Accounts Setup**: Ensure the accounts for Login Enterprise Tests are already configured. For details, see Accounts.
9. **Launcher Group Assignment**: Verify that the Login Enterprise Launcher(s) running on IGEL OS or other operating systems are added to a launcher group. For guidance, see Launcher Groups and Locations.

Connection Setups

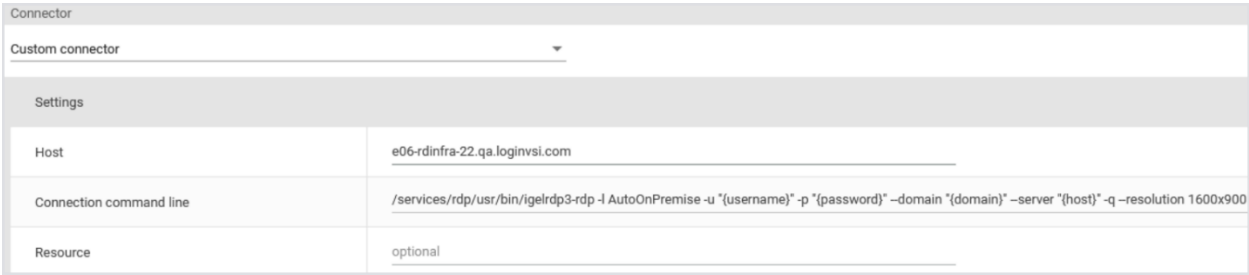
For setting up your Test connections, refer to the following:

- For Continuous Test setups, refer to Managing Continuous Testing.

- For the Load Tests, see Managing Load Testing.
- For Application-specific Tests, consult Managing Application Testing.

Microsoft Connections

Setup the Connection: Configure your Microsoft connection as described below, as a Custom connector.

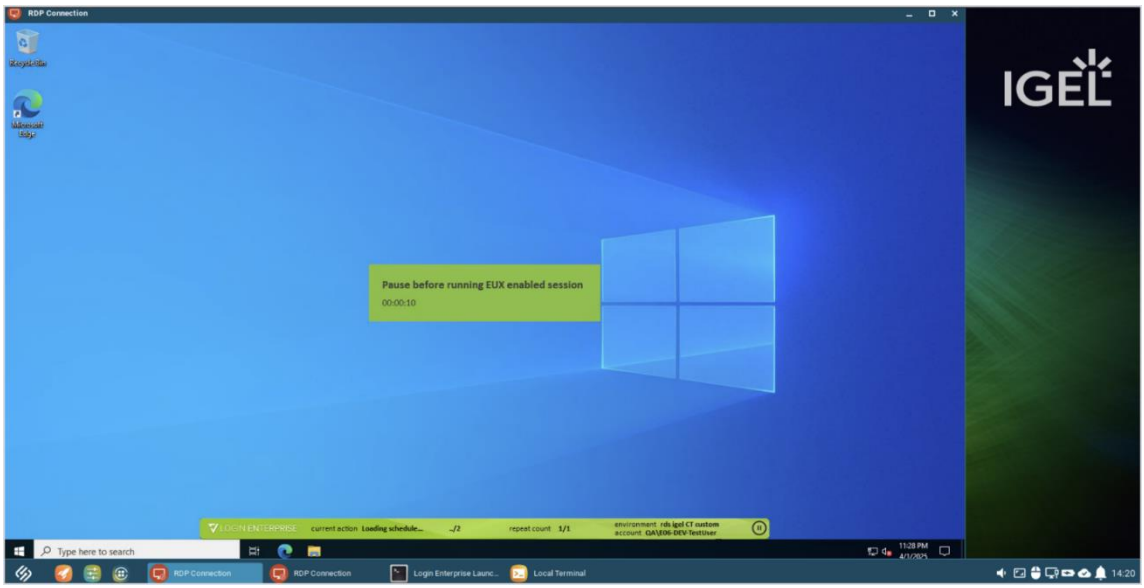


A base Connection command line would be:
`/services/rdp/usr/bin/igelrdp3-rdp -l AutoOnPremise -u "{username}" -p "{password}" --domain "{domain}" --server "{host}" -q`

You can use other connection command line options and arguments, such as `Resource`. Please refer to the `IGEL Remote Desktop` command line reference for more options by running `/services/rdp/usr/bin/igelrdp3-rdp --help`. Here's a snippet of the command output:

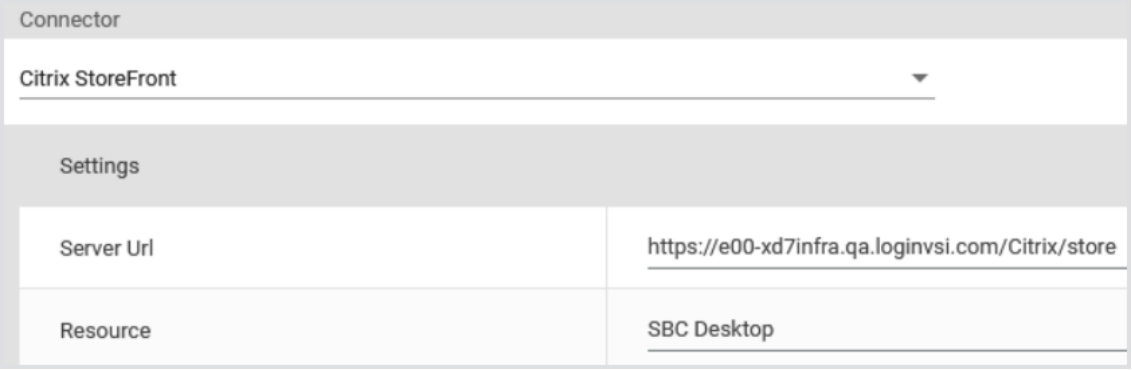
```
root@VBOX:~# /services/rdp/usr/bin/igelrdp3-rdp --help
QStandardPaths: runtime directory '/run/user/777' is not owned by UID 0, but a directory p
ermissions 0700 owned by UID 777 GID 100
Usage: /services/rdp/usr/bin/igelrdp3-rdp [options]
IGEL rdcore application

Options:
  -h, --help                                        Displays help on
                                                    commandline options.
  --help-all                                        Displays help including Qt
                                                    specific options.
  -v, --version                                     Displays version
                                                    information.
  -c, --configfile <configfile>                     Use <configfile> for
                                                    session configuration (set
                                                    to "-" to read from stdin).
  -l, --login <mode>                                Use <mode> as the login
                                                    mode:
                                                    None,DefaultOnPremise,Defau
                                                    ltAVD,AutoOnPremise,AutoAVD
  --msal, --MSAL                                    Force MSAL.
  --nomsal, --noMSAL                                Disable MSAL.
  --w365, --windows365, --cloudpc, --cpc            Enable Windows 365 Cloud
```

Verification: You'll know the connection is successful when the RDS target desktop initiates the login process and runs Login Enterprise Tests.

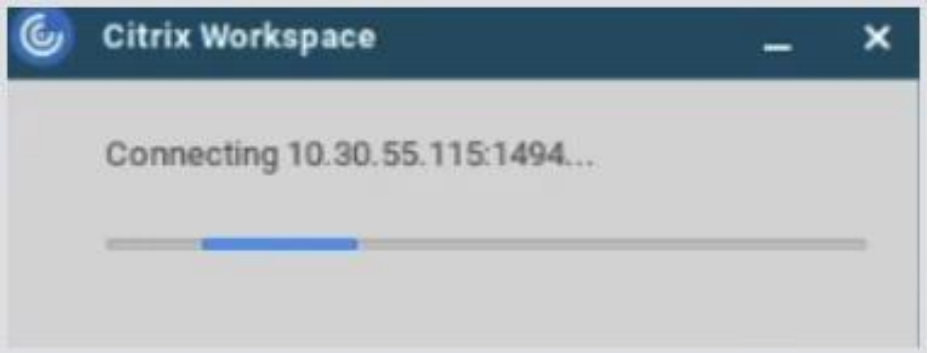


Citrix Connections

Setup the Connection: Configure your Citrix connection as shown below. Be sure to use the correct `Server Url` and `Resource` for the testing.



Verification: A successful connection is indicated by the Citrix target desktop launching and displaying the active Login Enterprise Test session.



Omnissa Connections

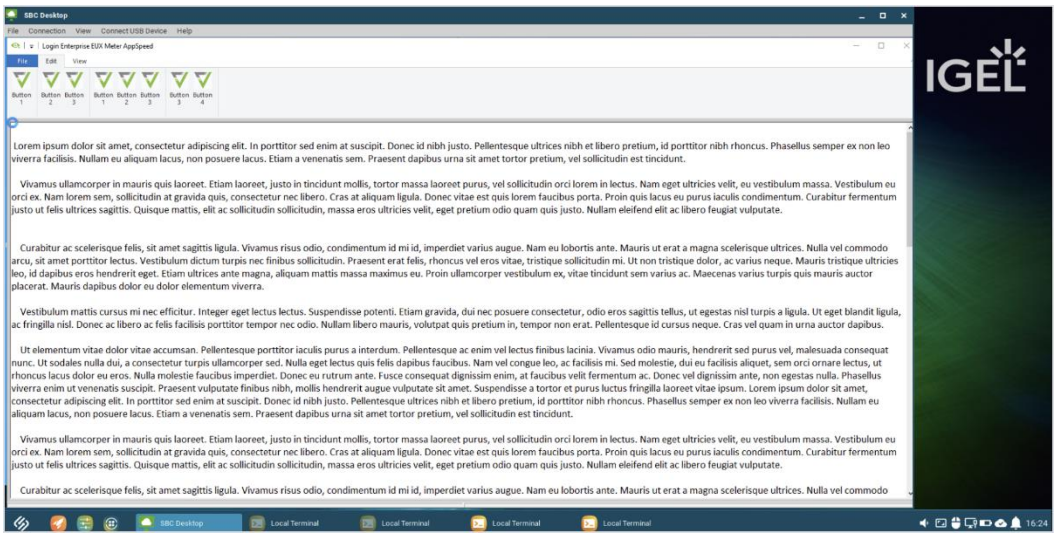
Setup the Connection: Set up your Omnissa connection as outlined below as a VMware Horizon View connection type.

A base Connection command line would be:
`/services/horizon/bin/vmware-view --serverURL "{serverurl}" --password "{password}" --userName "{username}" --desktopName "{resource}" --domainName {domain} --nonInteractive`

You can use other connection command line options and arguments. Please refer to the `vmware-view` command line reference for more options by running `/services/horizon/bin/vmware-view --help`. Here's a snippet of the command output:

```
root@VBOX:~# /services/horizon/bin/vmware-view --help
Using log file /tmp/vmware-root/vmware-horizon-client-13408.log
Usage:
  vmware-view [OPTION…] - connect to VMware Horizon desktops

Help Options:
  -h, --help                                          Show help options
  --help-all                                          Show all help options
  --help-gtk                                          Show GTK+ Options

Application Options:
  -K, --keep-wm-bindings                              Keep window manager key binding
s (ignored by some remoting protocols).
  -s, --serverURL=<broker URL>                        Specify connection broker.
  -u, --userName=<user name>                          Specify user name for password
authentication.
  -p, --password=<password>                           Specify password for password a
uthentication.
  -t, --tokenUserName=<token user name>               Specify user name for RSA or Ra
dius authentication.
  -c, --passcode=<passcode>                           Specify passcode for RSA or Rad
ius authentication.
  -d, --domainName=<domain name>                      Specify domain for password aut
hentication.
  -a, --applicationName=<application name>            Specify application by name.
  -n, --desktopName=<desktop name>                    Specify desktop by name.
  -q, --nonInteractive                                Connect automatically if enough
 values are given on the command line.
  --fullscreen                                        Enable full screen mode in sing
le monitor mode.
  --allmonitors                                       Enable full screen mode accross
```

Verification: You can confirm the connection when the Omnissa target desktop is visible and the Login Enterprise testing session is actively running.



Congratulations on completing the setup of the Login Enterprise Launcher on IGEL OS 12. Now you're ready to run testing with Login Enterprise and your IGEL Launcher device. To learn how to run Tests, see the links in the Connection Setups section.

Deploying the Launcher through the IGEL App Portal

- Overview
- Prerequisites
- Steps
- Alternative: Local On-Device Installation
- Next steps

Overview

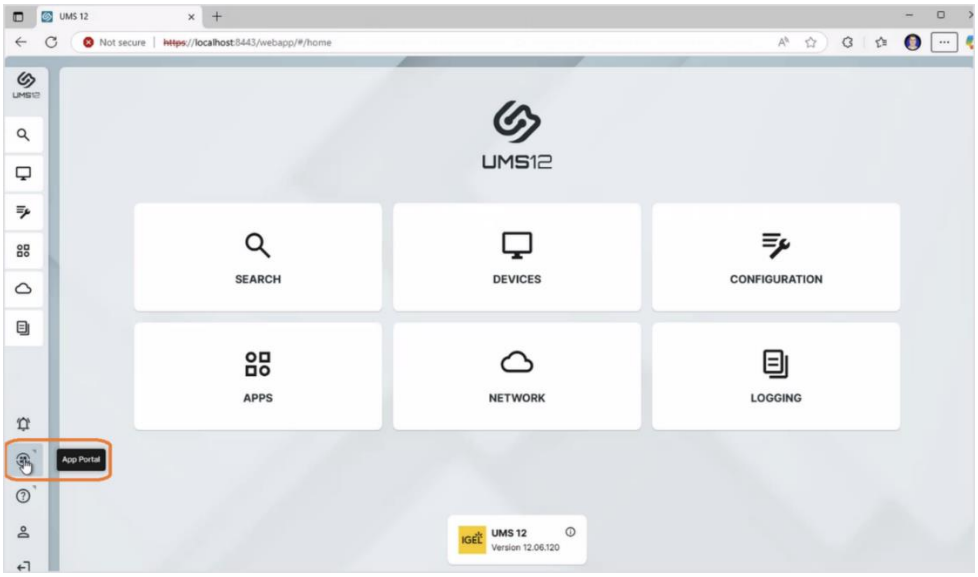
With Login Enterprise Launcher now available on the IGEL App Portal, you can deploy performance-testing endpoints on IGEL OS devices in minutes. This guide shows you how to import the Launcher app into your IGEL UMS so it's ready for centralized distribution and management.

Prerequisites

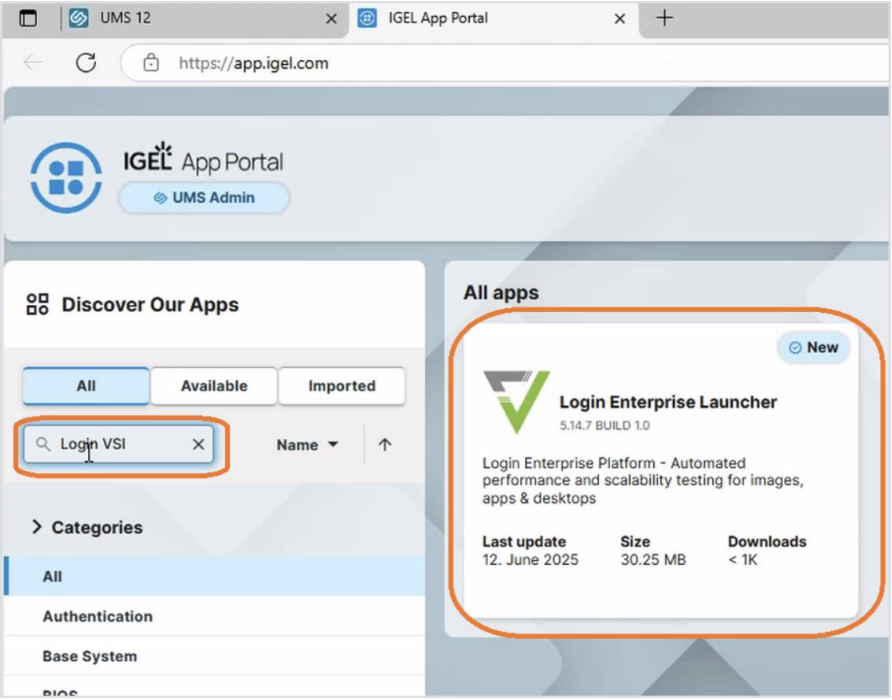
- **IGEL OS device(s)**: Already set up and registered in UMS, and intended to run as Login Enterprise Launchers
- **IGEL UMS 12**: Fully installed, licensed, registered, and able to access the IGEL App Portal

Steps

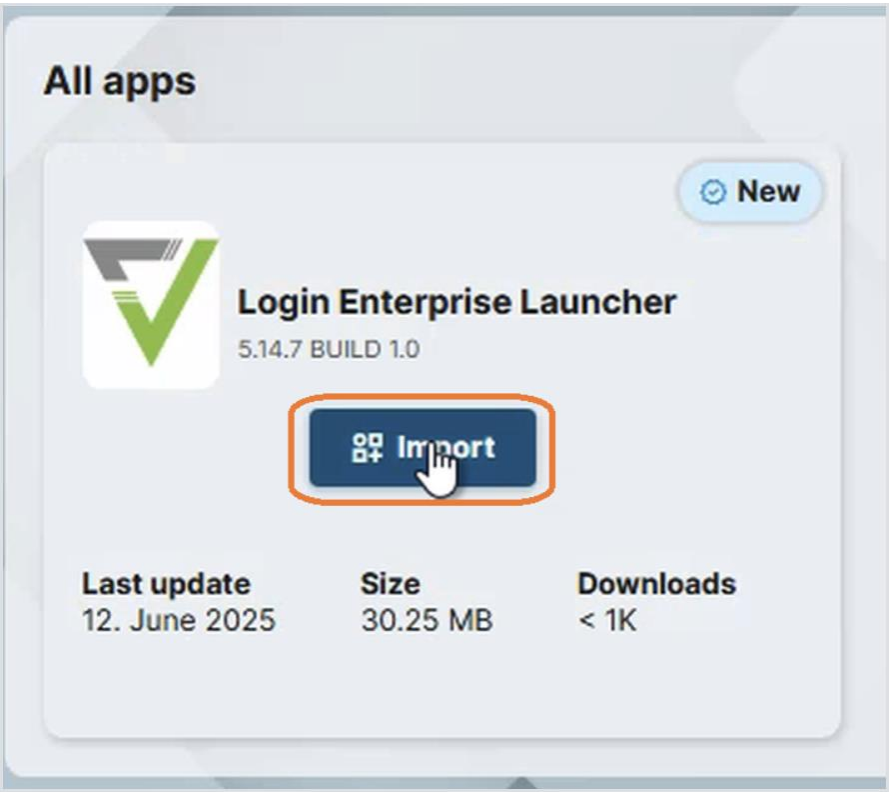
1. Log in to UMS Web with an account that has Admin or Super-Admin rights.
2. Click **App Portal** in the left sidebar.



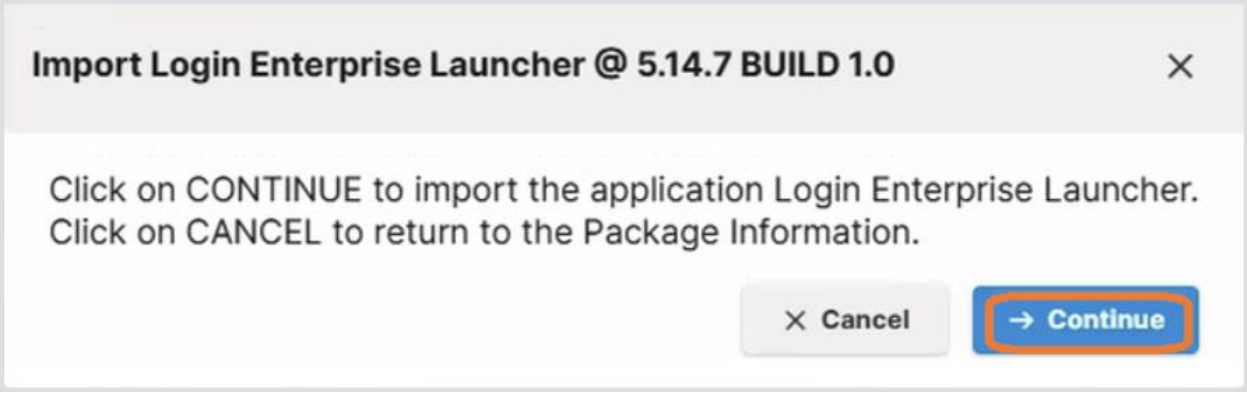
3. In **App Portal**, enter **Login Enterprise** or **Login VSI** in the search box.



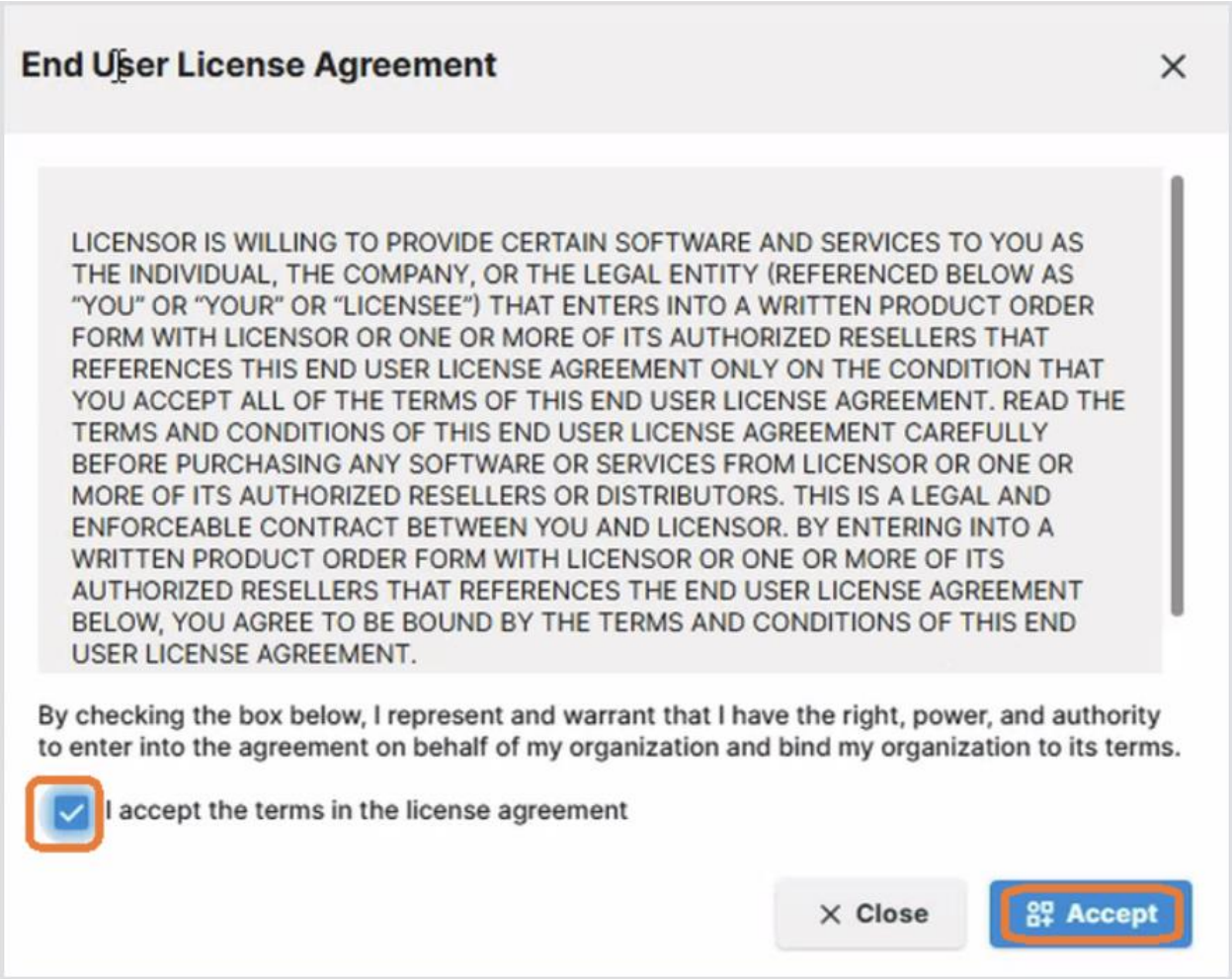
4. Hover over Login Enterprise Launcher and click **Import.**



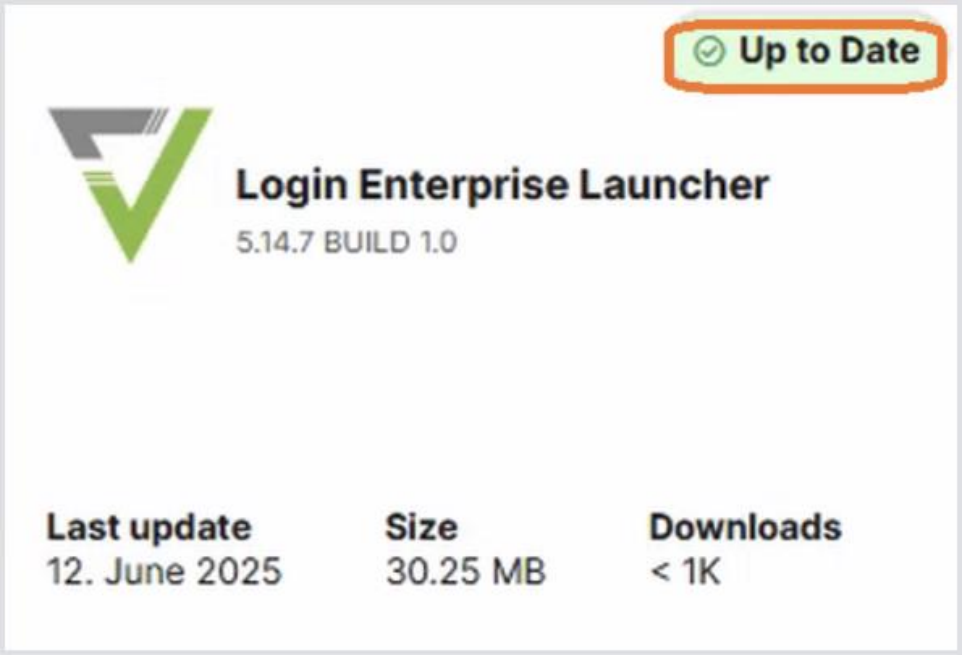
5. Click **Continue** to begin the import process.



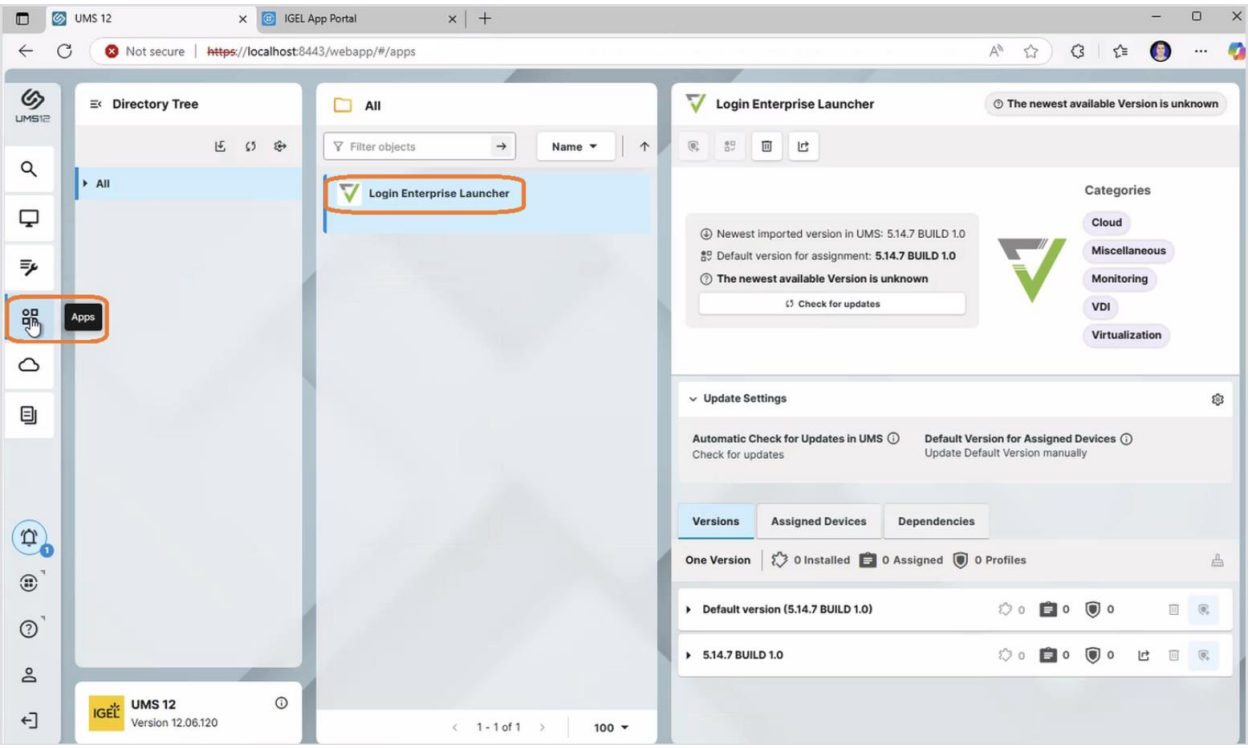
6. Check **I accept the terms in the license agreement**, then click **Accept**.



Once complete, the app's status will read **Up to date**, indicating the current version is available in UMS.



7. Return to the UMS Web home and click **Apps**. You'll see **Login Enterprise Launcher** listed and ready for assignment.



Now you're ready to assign Login Enterprise Launcher to your IGEL OS devices and begin performance testing at scale! To deploy the Login Enterprise Launcher to the IGEL OS Device(s), refer to [Getting the Login Enterprise Launcher onto IGEL Devices].

Alternative: Local On-Device Installation

If you prefer not to use UMS, you can install directly on an IGEL OS 12 device:

1. Open the **App Portal** application on the IGEL device
2. Search for **Login Enterprise Launcher.**
3. Click **Install.**

NOTE: Direct device installs bypass UMS but lack centralized control. UMS import is the recommended approach.

Next steps

1. For information on setting up automatic or manual updates for the Login Enterprise Launcher app in IGEL UMS (when released onto the IGEL App Portal), refer to How to Configure Update Settings for Apps in the IGEL UMS Web App.
    1. When the Login Enterprise Launcher update occurs, the Launcher's appsettings file needs to be updated. For details, see steps 1 through 7 in Updating the Launcher.
2. To get the Virtual Appliance certificate set up in IGEL, see Getting the Login Enterprise certificate into UMS and onto the Device.
3. For documentation on the IGEL App Portal usage, see IGEL App Portal.

Virtual Users

This subchapter guides managing Virtual User Accounts, also referred to as Test Users, within the Virtual Appliance. You will explore the logon components necessary for user authentication and learn how to create and manage Virtual User accounts, including handling email fields.

Additionally, the section covers bulk account creation, including the process for creating users in Azure AD in bulk and streamlining user management for your environment, as well as understanding how Login Enterprise Engine works.

Test Users are responsible for logging into the Windows test desktops, executing workloads, and gathering measurements during testing. This ensures a comprehensive evaluation of application performance and user experience.

- Configuring Logon Components
  - Configuring Logon Processor as a Logon Script
- Managing Virtual User Accounts
- Understanding Login Enterprise Engine

Configuring Logon Components

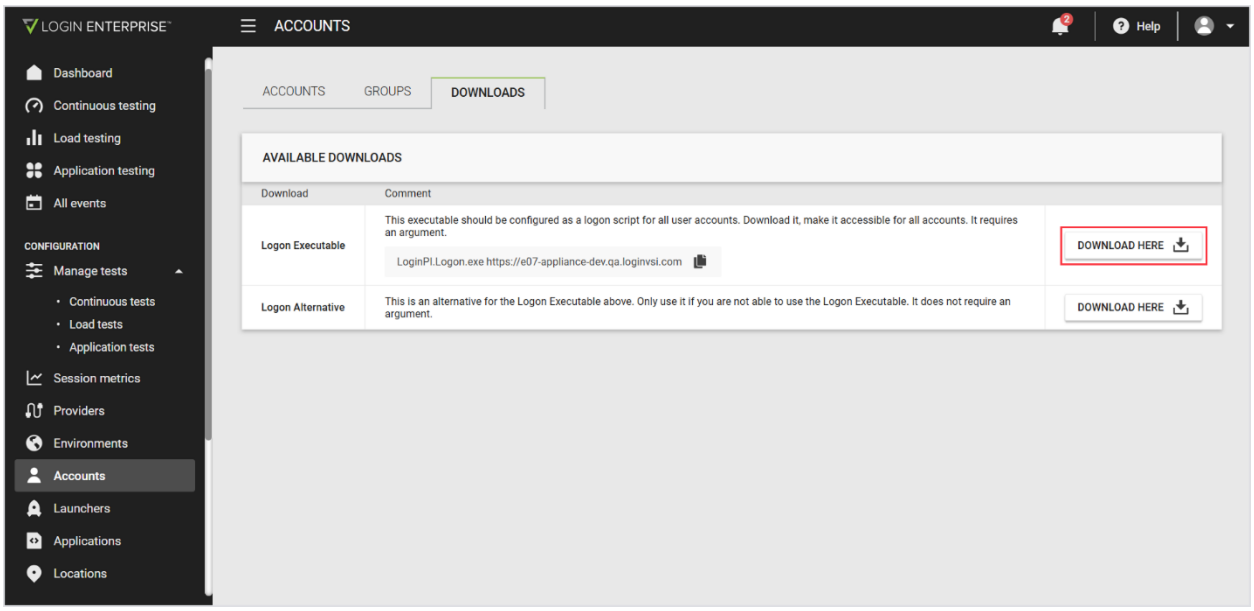
- Overview
- Downloading logon components
- Logon Executable
  - Options for configuration
- Desktop connection
- Logon script

Overview

The logon components are crucial to the functionality of Login Enterprise. This part ensures the software starts as soon as the users log in.

Downloading logon components

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Accounts**.
2. In **Accounts**, select **Downloads** from the tab menu and then download the **Logon Executable** file.



TIP: Alternatively, the logon components can be downloaded from the sidebar menu under **Other > Downloads > Accounts**.

The Logon component must run when a Login Enterprise virtual user logs into the desktop. Therefore, it must be located somewhere accessible by the virtual user when it logs on to the system. There are multiple ways to do this:

- You can either run the component in the user properties for the AD User Object or in the AD Group Policy.
- You can run it via UEM tooling, such as RES Workspace Manager or AppSense.

Login Enterprise has two options to start the Virtual User within the Target session. The recommended option is Login Executable:

- Logon Executable
- Logon Script

Logon Executable

IMPORTANT: The Logon Executable does not support Server 2008 R2, Windows 7, or older versions of Windows.

Once you've downloaded and extracted the files from the Logon Executable, you need to run it. All we require is that the LoginPI.Logon.exe is executed with a single command-line parameter, the URL of your Login Enterprise environment. See the example below. Note the lack of a forward slash character after the base URL string:

```
.\LoginPI.Logon.exe https://p05-appliance.play.loginvsi.com
```

You can implement this as the logon script command and place the executable in a location that is accessible by all Login Enterprise users. Or, you can create your logon script with this command that is then executed by your preferred setup.

Options for configuration

When configuring the Logon Executable, several common scenarios may suit various organizational needs. They are as follows:

1. **AD user profile login script**: You can specify a login script for each Virtual User account within their profile settings. In legacy environments, this approach may already be in use for organizational login scripts, ensuring consistent user experience.
2. **Login script with GPO filtering**: You can create a Group Policy Object (GPO) to execute LoginPI.Logon.exe during user logon. By applying security filtering, you can ensure that this policy only affects Virtual Users or the Organizational Unit (OU) containing them. This method allows for targeted execution based on user roles.
3. **Environment manager policies**: If your organization utilizes an Environment Manager, for example, WEM or Ivanti, you can configure policy-based login scripts that execute for a defined scope of users, such as an OU for Virtual Users. This flexibility helps tailor the logon process to your specific environment.
4. **Integration with existing login scripts**: If your organization already employs a standard login script, you can enhance it by adding a command to invoke LoginPI.Logon.exe after all other startup activities have completed. This ensures that the new command runs seamlessly alongside existing processes.

Desktop connection

If you are using the Desktop Connector for Application or Continuous tests, there is one exception on the command-line parameter. You might need to add the test name, so the command will look like the following:

```
.\LoginPI.Logon.exe https://p05-appliance.play.loginvsi.com "Test Desktop"
```

If you are on a version below 4.4.X, you will need to add the GUID at the end of the command line parameters as shown in the example below:

```
.\LoginPI.Logon.exe https://p05-appliance.play.loginvsi.com ebdfa18b-772b-
4423-8069-5211c7a62688
```

The GUID at the end of this command is the unique ID of the environment you have configured with the desktop connector. The easiest way of finding out the GUID of the environment is by opening the environment configuration page and selecting the environment. In the URL bar of your browser, you can see the ID at the end of the URL.

p20-appliance.play.loginvsi.com/configuration/environments/dbacb332-d9a8-4171-8985-0dd5cc523ef7

If your company uses Windows 10, there will be a short delay before the Logon script starts running. To speed up the process, Microsoft has released a fix. For more details, see the Microsoft documentation.

Logon script

The Logon script is an alternative to the Logon Executable. It is meant to be used in older environments that the Logon Executable does not support. To use Logon Script, download it and attach it to the user. There are multiple ways to do this:

- You can either run the component in the user properties for the AD User Object or in the AD Group Policy.
- You can run it via UEM tooling, such as RES Workspace Manager or AppSense.

The Logon Script uses PowerShell. Your test users must be allowed to run this in PowerShell's full language mode. For more information on the Logon Script, see Antivirus and app blocker exclusion.

For instructions on configuring LoginPI.Logon.exe as a Logon Script, see Configuring Logon Processor as a Logon Script.

Configuring Logon Processor as Logon Script

- Introduction
- Logon Processor Location
  - NETLOGON
  - Standard File Server Folder
  - On the VDI Image
  - Pros and Cons
- Configuring a Logon Script
  - Domain GPO
  - Domain User Profile
  - Startup Folder
  - Registry
  - Scheduled Task
  - Profile Manager
- Configuration Considerations

Introduction

To run `LoginPI.Logon.exe` (the Logon Processor) when a test user logs in, you need to configure it as a login process. For this:

1. Place the executable in a location where it can be accessed.
2. Configure it to run at login.

IMPORTANT: Each user account should be dedicated to a single installation of Login Enterprise. The Logon Script must point to the specific Appliance URL.

Although it is technically possible to create a system where an account selects which appliance to connect to at login, you cannot have two remote sessions to the same account at the same time. Even if an account can detect which appliance to use, multiple appliances cannot coordinate access to ensure only one connects at a time. Configuring multiple appliances to avoid overlap introduces unnecessary complexity. Make sure to use separate accounts for each appliance.

`LoginPI.Logon.exe` must run as a process in an interactive logon session. It cannot run as a service or any other non-interactive process. It requires a real desktop environment with a window and a window manager.

Logon Processor Location

The Logon Processor must be in a location that test users can access automatically, without requiring an additional interactive logon. Common locations include:

1. The **NETLOGON** share in your domain
2. A folder on a standard file server
3. A directory on the VDI image

NOTE: Place the executable in a location accessible to all test users using the same path. Avoid storing it in individual users' personal shares.

Using a common location makes it easier to assign a single logon process to all users, rather than maintaining separate processes for each user.

NETLOGON

Domains often include a network share for logon resources, usually configured as the **NETLOGON** share under the domain name. For example:

```
\\MyDomain.com\NETLOGON
```

This share is normally accessible to any user who has successfully authenticated to the domain, making it a simple location for the executable. Drag and drop `LoginPI.Logon.exe` to the **NETLOGON** share, and configure your logon script to point to:

`\\MyDomain.com\NETLOGON\LoginPI.Logon.exe ...` (where `...` would be additional parameters, like the URL of the Login Enterprise Appliance)

NOTE: You may need to access the domain controller to place `LoginPI.Logon.exe` in the **NETLOGON** folder. The folder is located at:

```
%SystemRoot%\SYSVOL\DomainName\Scripts
```

The folder is shared out under the name **NETLOGON**.

Standard File Server Folder

Create a share on your file server that is accessible to all test users. You can configure an Active Directory user group containing your test users and grant read access to the share or to a folder within an existing share. Ensure that the permissions do not require additional authentication.

Configure your logon script to point to the executable using the UNC path, for example:

```
\\MyDomain.com\Shares\TestUsers\LoginPI.Logon.exe ...
```

On the VDI Image

You can place `LoginPI.Logon.exe` in a directory on the boot image or an image layer if you are using layering technology. For example, create the folder:

```
C:\Program Files\LoginEnterprise\Logon Processor\
```

and place `LoginPI.Logon.exe` in that folder when building the image used for test desktops. Configure your logon script to point to:

```
C:\Program Files\LoginEnterprise\Logon Processor\LoginPI.Logon.exe ...
```

You can also use lifecycle automation to install `LoginPI.Logon.exe` automatically at boot or during a refresh.

Pros and Cons

Boot image

- **Pros**: Simple to deploy. No need to manage network resources, permissions, or routing.
- **Cons**: Harder to update. We provide a new `LoginPI.Logon.exe` with every appliance update. Although the system is designed to be backward-compatible, you should update the program whenever you update the appliance, just as you would upgrade all Launchers.

NETLOGON share

- **Pros**: Simple to use in a controlled environment.
- **Cons**: May require additional permissions. You might need another team to place the program and update it when necessary.

File server folder

- **Pros**: Easy to configure, as long as test users can access it. Updating is as simple as copying the new file in place.
- **Cons**: Requires network access to the file server.

TIP: Login VSI training resources generally recommend the NETLOGON share as the simplest option, since a training lab usually provides full control over the testing domain.

Configuring a Logon Script

Typically, you configure `LoginPI.Logon.exe` as a logon script for specific users.

IMPORTANT: Avoid configuring a machine-level logon script. It will run for all users and may log other test users out unexpectedly. This is especially important when running Login Enterprise on a shared VDI pool. Even on a closed testing system, avoid this practice.

You can configure a process to run at logon using one of the following methods:

- **Domain GPO**
- **Domain user profile properties**
- **Local policy**
- **Startup folder in the Start Menu**
- **Registry**
- **Scheduled task**
- **Profile Manager**

Domain GPO

You can configure a Group Policy Object (GPO) to run a script at logon. A full GPO tutorial is beyond the scope of this section. To configure a logon script:

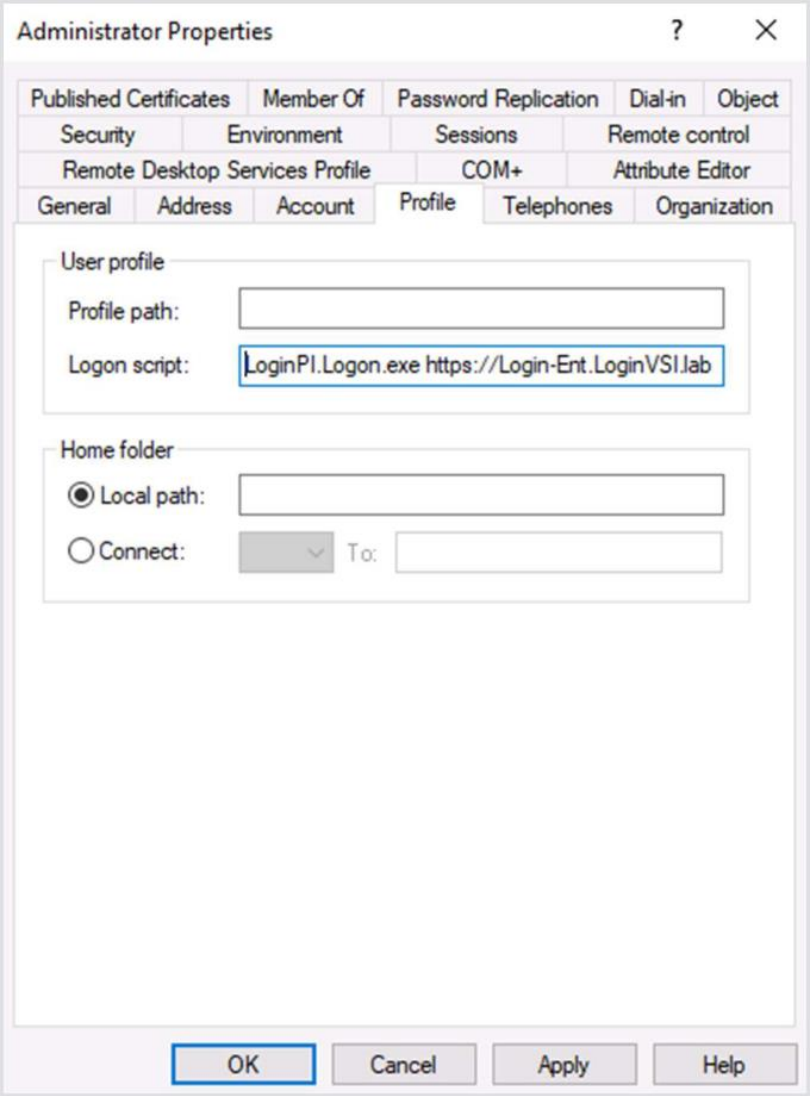
1. Create a new GPO.
2. Configure the GPO to include a logon script that points to your `LoginPI.Logon.exe`, including any required parameters.
3. Apply the GPO to your test users or to a user group containing them.

For detailed instructions on assigning logon scripts, see the Microsoft documentation.

Domain User Profile

To assign a logon script to a user:

1. Open **Active Directory Users and Groups** for your domain.
2. In the console tree, select **Users**.
3. Double-click the user to whom you want to assign a logon script.
4. Click the **Profile** tab.
5. In the **Logon script** field, enter the full path and any parameters for the script, and then select **OK**.
6. Repeat these steps for each user.



Startup Folder

If you place a program or shortcut in the **Startup** folder, it will be executed when Explorer starts up. That is, an interactive session starts when a user logs in.

- **Current user:** `C:\Users\USERNAME\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup`
- **All users:** `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup`

You can also open these folders using the **Run** dialog: enter `shell:startup` for the current user or `shell:common startup` for all users.

To configure a startup process:

1. Place a script, a shortcut to a script, or a shortcut to the Logon Processor (with the required command-line parameters) into the appropriate folder.
2. The current-user folder affects only the logged-in user.
3. The all-users folder applies to every user on the machine, which dedicates the machine to testing.

Registry

To define a startup script through the registry, create one of these keys. You can configure a startup script by creating one of the following keys:

- **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run** – Runs programs for all users.
- **HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run** – Runs programs for the current user.

IMPORTANT: HKEY_CURRENT_USER applies only to the account you're signed in with. You must configure the key for every user who needs the script.

To configure the startup script:

1. Under the appropriate **Run** key, create a new value of type **REG_SZ**.
2. Enter any name for the value.
3. Set the value data to the full command line of the program you want to run, including parameters and any quotation marks required if you were entering the command in a Command Prompt window.

Scheduled Task

You can use the Task Scheduler, a built-in Windows tool, to create a task that runs when a user signs in.

1. Open **Start** > **Administrative Tools** > **Task Scheduler**.
2. In the tree, select **Task Scheduler Library**.
3. In the **Actions** pane, select **Create Task**.
4. In the **Create Task** dialog, configure the following tabs:
    - **General**: Enter a name for the task.
    - **Triggers**: Select **New**, then set **Begin the task** to **At log on**.
    - **Actions**: Select **New**, then select **Browse** to locate your script.

IMPORTANT: Processes started from Task Manager run at a lower priority than standard user processes by default. If you use the Task Manager to start the login process, ensure that you update the task settings after creating it. Set **Priority** to **4, 5, or 6 (Normal)** instead of the default **7 (Below Normal)**.

If you do not change the priority, test results and EUX information may be artificially low. For more information about Task Priority settings, see the Microsoft documentation.

The Task Scheduler UI does not support changing task priority. To update the priority, run the following PowerShell command:

```
$task = Get-ScheduledTask -TaskName 'Your task's path and name'
$settings = $task.Settings
$settings.Priority = 4
Set-ScheduledTask -TaskName $taskName -Settings $settings
```

Profile Manager

If you use a Windows profile manager such as FSLogix, Ivanti, or Citrix Profile Manager, you may be able to configure it to run this program at startup in user sessions. For configuration steps, see your profile manager's documentation.

Configuration Considerations

The **Scheduled task**, **Registry**, and **Startup folder** options require changes to the base image or a roaming profile that follows the user. Configure them per user, not per machine, to avoid committing the entire machine to testing. These methods are useful for lab testing, but not recommended for production.

We recommend using a **domain Group Policy Object (GPO)** instead of a domain user profile. A GPO is easier to modify and can be applied to all users immediately. If each user profile has a logon script, you must update every profile to make a change.

Avoid configurations directly on the VDI machine. They don't apply in streamed app environments, where you may not control the local configuration of the app server. Domain-level policies follow the user, even to the app server.

Your profile manager may have the configuration flexibility you need to target specific test users or hosts. Because we don't provide documentation for profile managers, check your product's options.

Managing Virtual User Accounts

- Overview
- Creating an Account
  - Email field
    - Syntax
    - Usage with Custom Connectors
  - Creating Multiple Accounts
    - Email template format
- Custom Fields (Optional)
- Secure Custom Fields (Optional)
  - Create flow
  - Edit flow
  - Behavior and security
- Handling Account Credentials on the Launcher
- Creating Account Groups
  - Adding a Group to an Environment
- Logon Components
- Account Status

Overview

Users must be created in the Active Directory by an AD administrator. Login Enterprise does not create users automatically.

You need account groups for all types of Tests (Continuous, Load, and Application test environments). For details, see Creating Account Groups.

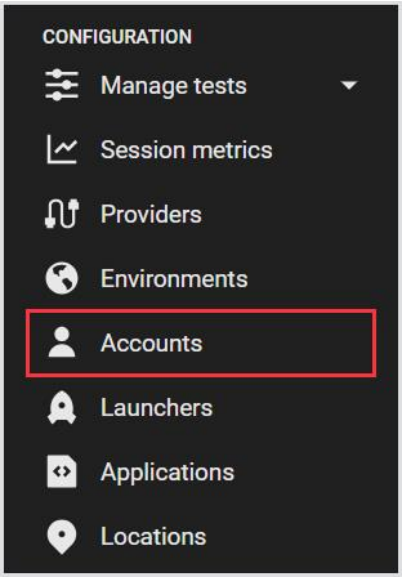
All the Login Enterprise users need to exist in Active Directory (AD) before configuring Virtual Users in Login Enterprise. We recommend creating users with a name followed by four digits, such as:

- LoginEnt0001
- LoginEnt0002
- LoginEnt0003
- LoginEnt0004
- etc.

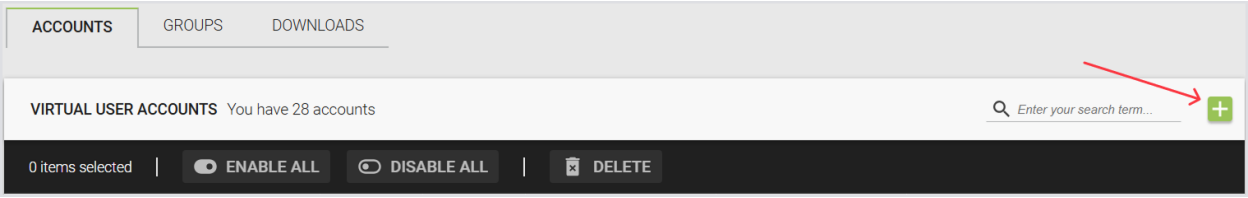
NOTE: Users should be created using the principle of least privilege. Test accounts should be given the minimum access and permissions required to perform their tasks.

Creating an Account

1. In the **Login Enterprise sidebar**, navigate to **Configuration** > **Accounts**.



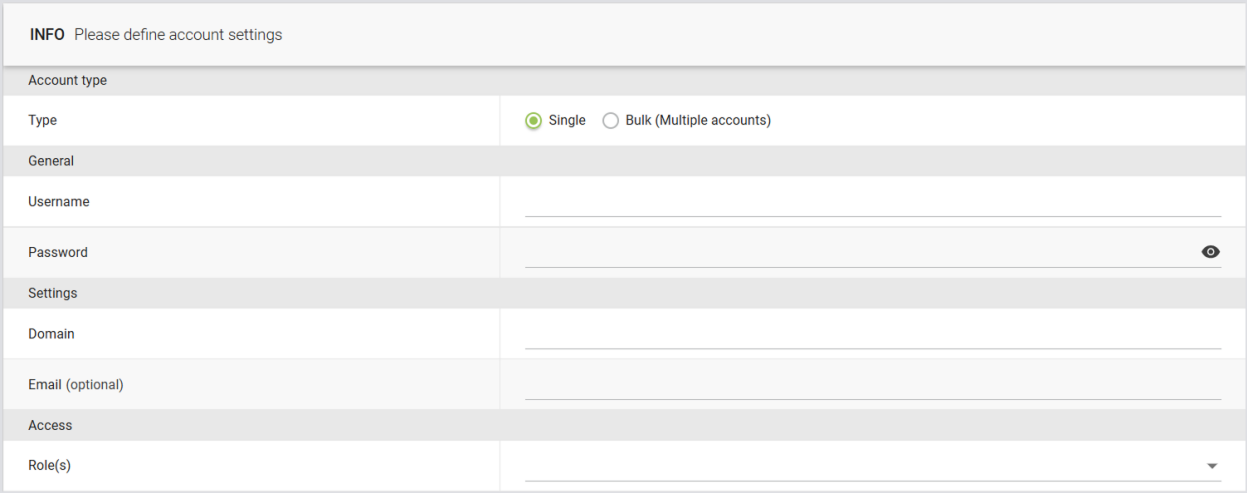
2. In **Accounts**, click the green "+" on the top right to add a new account.



3. Define the account settings:

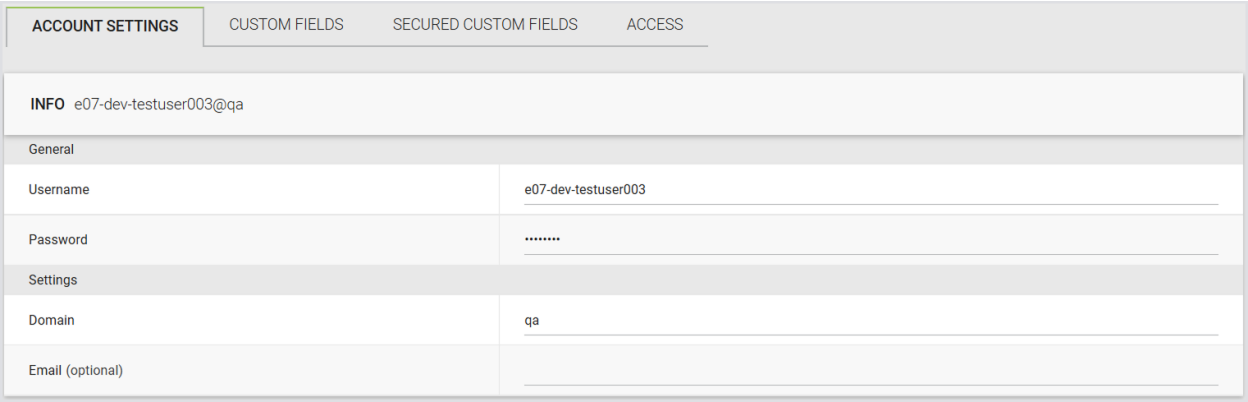
Account Type	
Type	Single or bulk (Multiple accounts)
General	
Username	The username of your choice.
Password	The password of your choice.
Settings	
Domain	The domain of the user account.
Email (optional)	The email address of the user account. For details, see the Email field.

Type	Select the way you'd like to create your account: **Single** or **Bulk** (multiple accounts). For details, see Creating Multiple Accounts.
Access	
Roless	The roles you'd like to assign to this account. For details, see Role-Based Access Control (RBAC).



Email field

You can save email addresses related to your accounts and use them later with custom connectors. When filling out individual account information in the account configuration form, you can enter the email address in a dedicated additional field. This field is optional, so you can choose to leave it empty. You can view and edit email information directly in the Accounts table:



Syntax

The following are valid examples of email entries:

- example@email

- example@email.com
- example.example@email.com
- example@email.email.com
- example.example@email.email.com

The following are invalid examples of email entries:

- example
- @email
- example.@email
- example@.email
- .example@email
- example@email.
- example@email@email

Usage with Custom Connectors

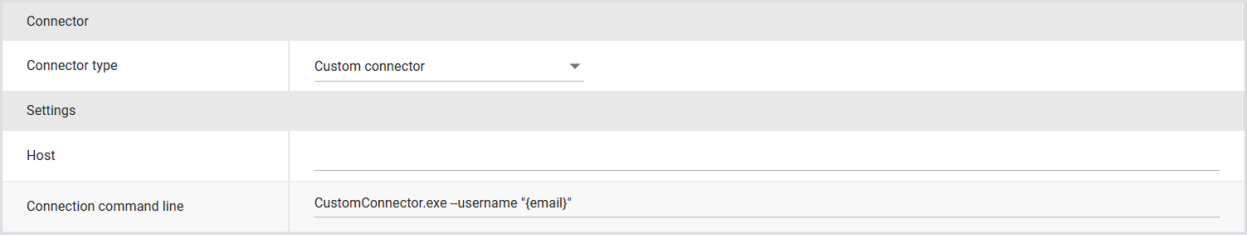
You can add an email as an argument that will be passed when starting a new custom connector process during a Test. The email placeholder in the script will be replaced with the actual email value from the accounts added to the Accounts Group used in the Test.

For example:

```
CustomConnector.exe --email "{email}"
```

The email value can also be used in place of the username:

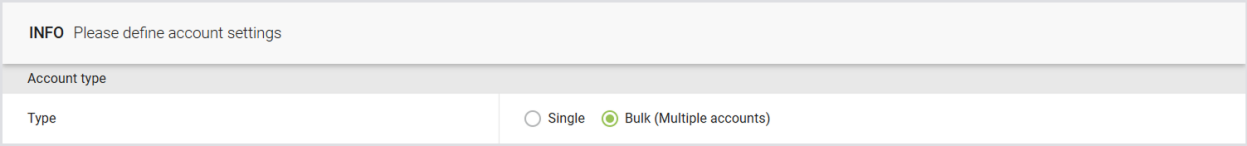
```
CustomConnector.exe --username "{email}"
```



You can also add email addresses when creating accounts using the Public API. Since this is optional, you can choose to omit the property.

Creating Multiple Accounts

You can create multiple accounts at once using the **Bulk (Multiple accounts)** option:



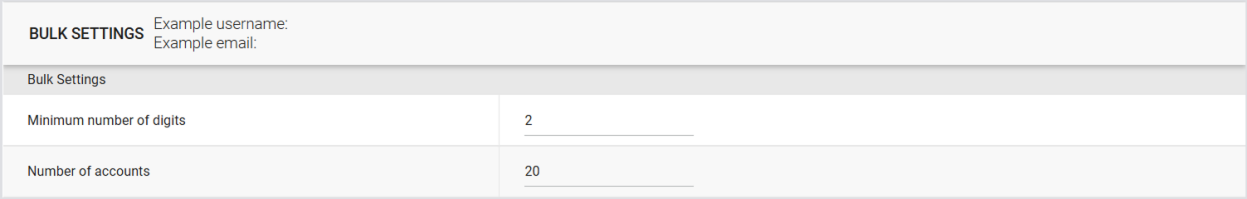
1. **Number of digits**: Specify the incremental number format (e.g., 01, 001).
2. **Number of accounts**: Indicate how many accounts you want to create.

When using this option, an example username and, if specified, an email will be generated based on your input. You can preview this information in the bottom left corner of the configuration screen.

The "Email" field is optional. If left empty, all generated accounts will have no email associated.

NOTE: We do not verify the existence or accessibility of email templates or generated emails. The system does not attempt to create actual emails in any form. Email uniqueness is not enforced, so multiple accounts can have the same email address associated.

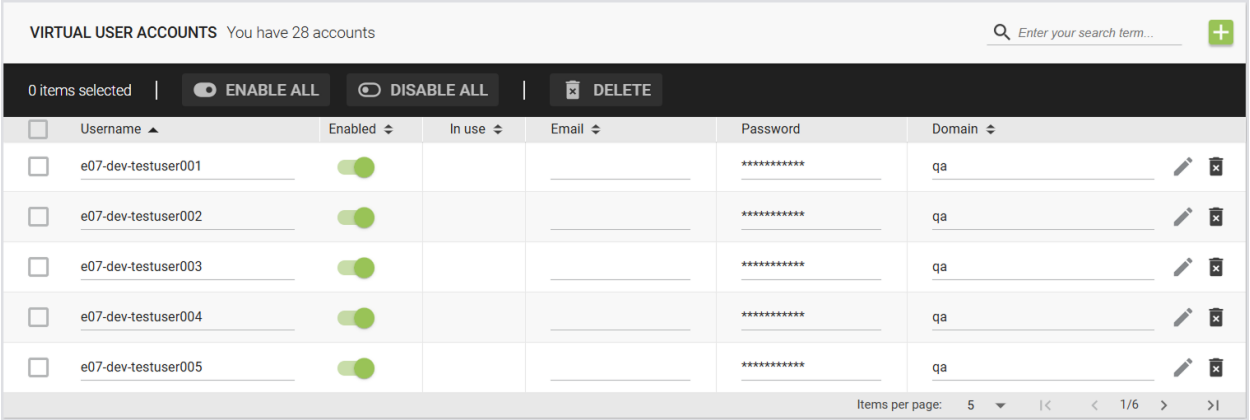
You can bulk-generate accounts with auto-generated emails using a specific email template. This template must be provided in the Email field on the **Create Bulk Accounts** screen.



Email template format

Each account will be created with an email based on the specified template. The logic mirrors username generation, where a number is added to the local part. An example email appears in the bottom-left corner of the screen next to the example username.

The "Email" field is optional. If left empty, all generated accounts will have no email associated.



IMPORTANT: We do not verify the existence or accessibility of email templates or generated emails. The system does not attempt to create actual emails in any form. Email uniqueness is not enforced, so multiple accounts can have the same email address associated.

Custom Fields (Optional)

Custom fields can enhance account configuration and are used with custom connectors, similar to email, username, domain, and password fields. Here are key points to consider:

- You can add up to five custom fields with editable values during account creation. These fields have immutable names.
- Once added, custom fields cannot be edited directly in the table view. Instead, click the "pencil" icon in each row to open the account editing form with pre-populated fields, including any existing custom fields.
- Custom fields can store various types of information, such as single digits, random text, or authentication tokens, but they are limited to 256 characters.

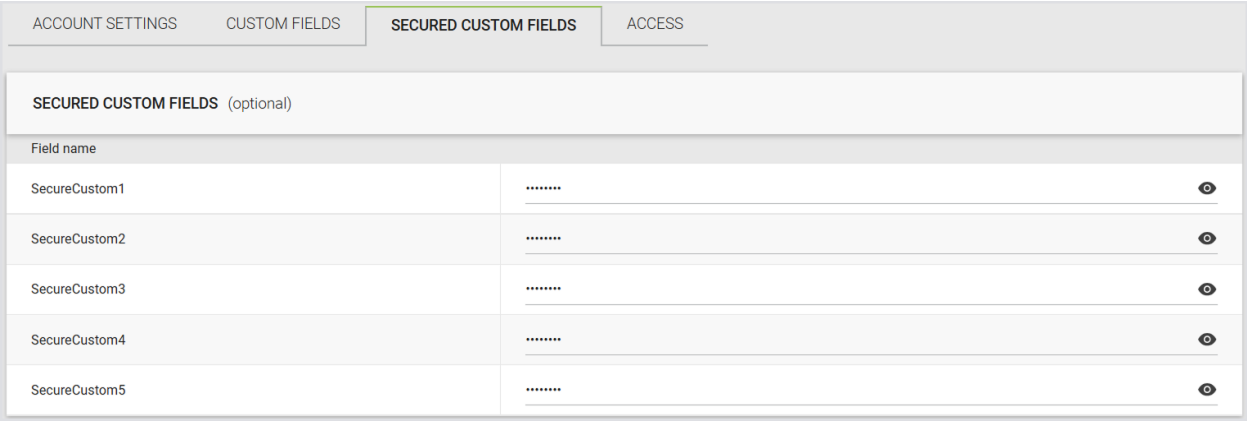
For more information on using custom fields with custom connectors, see Custom Fields.

Secure Custom Fields (Optional)

Secure custom fields are used to store sensitive values in your Virtual User accounts. You can add up to 5 secure custom fields:

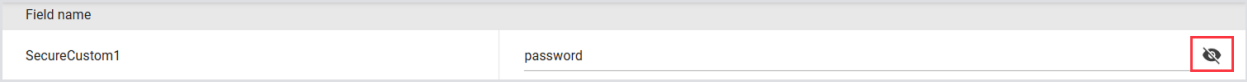
- **SecureCustom1**
- **SecureCustom2**
- **SecureCustom3**
- **SecureCustom4**
- **SecureCustom5**

Each field accepts a string value of up to 255 characters. Leading and trailing spaces are trimmed on save.



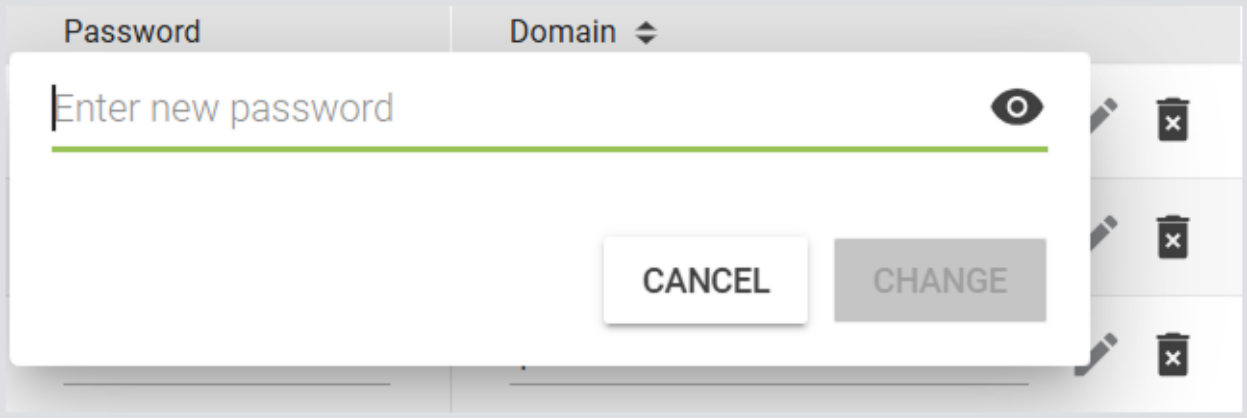
Create flow

- When you enter a value, the field masks the text with dots.
- You can select the **eye** icon to reveal the text you're typing before saving.



Edit flow

- Previously saved values appear as dots.
- When you start typing in a field, the placeholder text **Enter new password** appears.



- The **eye** icon only applies to the new text you enter before saving.

Behavior and security

- Values are masked in the UI and can't be retrieved through the UI or API after they're saved.
- Values are encrypted at rest using the same mechanism as the password field.
- Trimming on save (no leading/trailing spaces are kept).

- No role, including admin, can view saved values.

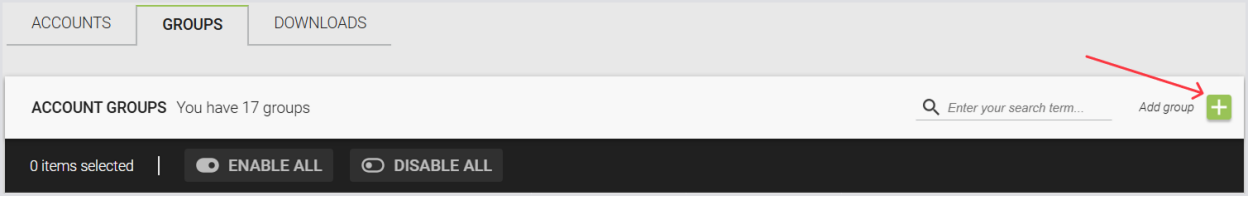
Handling Account Credentials on the Launcher

For details, see the section of the same name in Launcher Overview and Best Practices.

Creating Account Groups

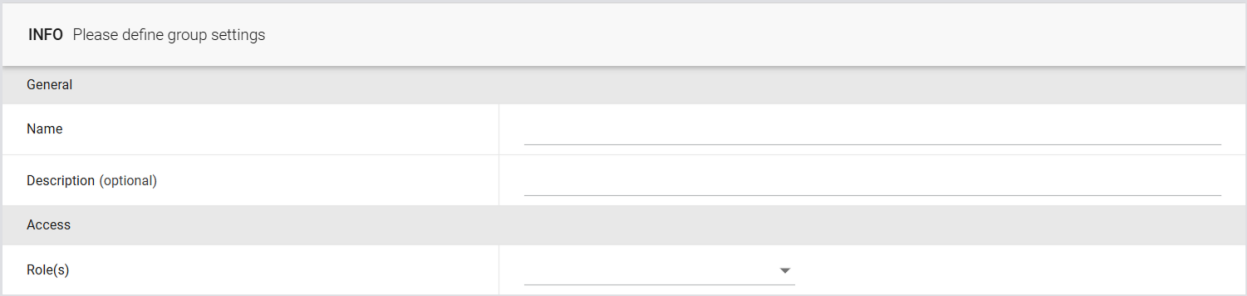
Account Groups allow you to grant specific users access to particular environments. To create an Account Group:

1. In **Accounts** > **Groups**, click the green "+" on the top right to add a new group.



2. Define the Group settings.

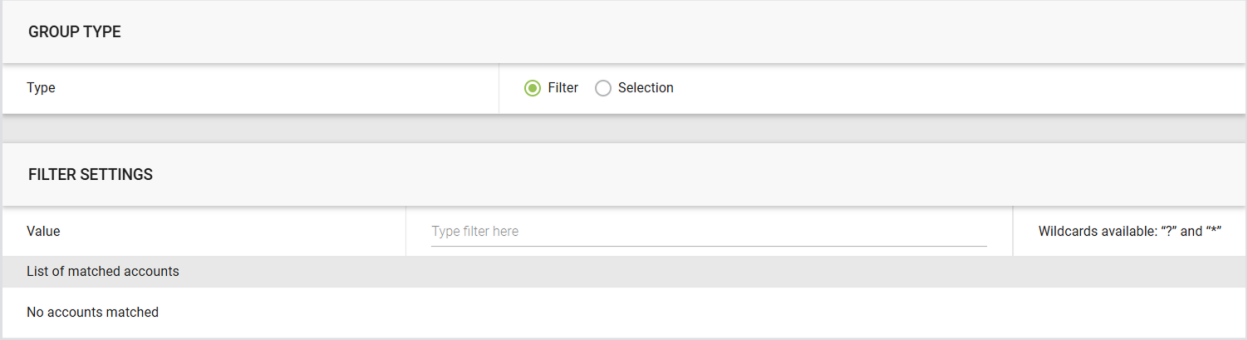
General	
Name	A Group name of your choice.
Description (optional)	A description specifying the Group's purpose.
Access	
Roles	Select the roles you'd like to assign to this Account Group. For details, see Role-Based Access Control (RBAC).



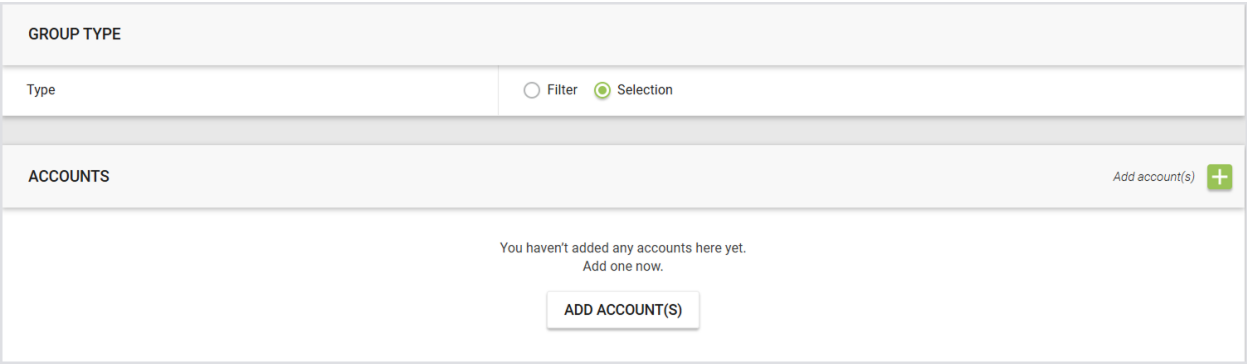
5. In **Group Type**, select **Filter** or **Selection**.

- **Filter**: Automatically include all accounts that match specific criteria. Here, you can use two wildcards.

- **? (Question mark)**: Represents a single-character wildcard.
- *** (Asterisk)**: Represents a multi-character wildcard.



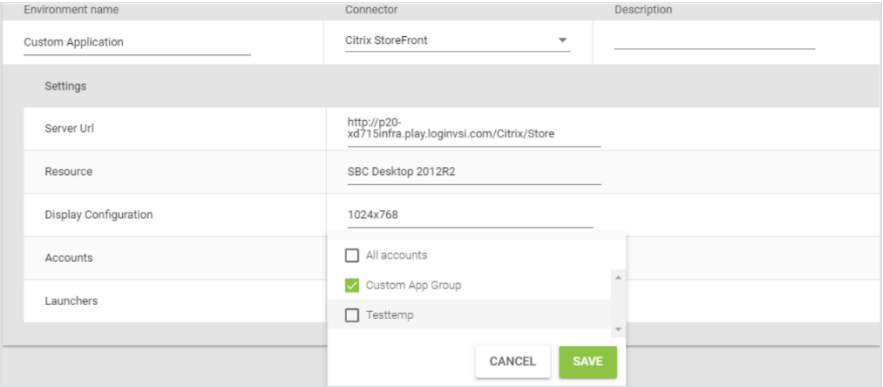
- **Selection**: Manually choose individual accounts to include in the group.



6. Click **Save** to apply the changes.

Adding a Group to an Environment

1. In the **Login Enterprise sidebar**, navigate to **Configuration** > **Environments**.
2. In **Environments**, edit the environment where you want to attach the user group.
3. Click on "Account" to view the groups you've created.
4. Select the desired group.
5. Click **Save**.



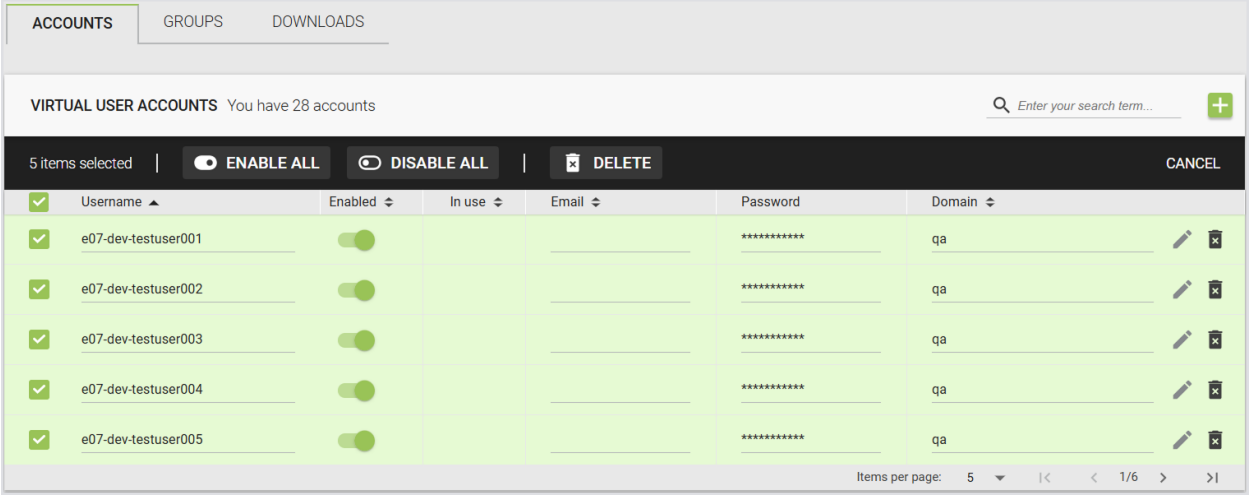
Logon Components

The Logon Components are crucial to the functionality of Login Enterprise. This part makes sure the software starts as soon as the user logs in. To learn more about the logon components, see Logon Components.

Account Status

By default, Login Enterprise accounts are enabled. If an account experiences three consecutive errors, it will be automatically disabled. You can manage account statuses in bulk starting from the accounts table. For this:

1. Select multiple accounts by ticking their checkboxes.
2. Use the buttons in the action panel to enable or disable the selected accounts.
    - **Enable** will be disabled (grayed out) if all selected accounts are already enabled.
    - **Disable** will be disabled if all selected accounts are already disabled.



Login Enterprise Engine

- Overview
- Functionality
- Engine User Interface
  - Appearance and Behavior
  - Key Information Displayed
  - Pause/Play Functionality
  - Additional Wait Messages
- Engine Processes
  - Core Components
  - Engine Process Errors
  - Log File Locations
- Key Benefits
- Value

Overview

The Login Enterprise Engine, also referred to as the Target Engine, is a critical executable within the Login Enterprise platform. It is responsible for facilitating Virtual User interactions, performance measurement collection, and seamless communication with the Login Enterprise Virtual Appliance.

Functionality

Upon login by a Virtual User, the Logon Component deploys the Login Enterprise Engine to the target digital workspace environment. The Engine determines if a Test is scheduled, retrieves Test parameters, and reports Test initiation to the Login Enterprise Virtual Appliance. It then executes the following processes:

- Runs Virtual User actions specified by the assigned workloads.
- Loops through workloads based on the configured Test duration or iteration count.
- Collects performance events and metrics, reporting them back to the Appliance over a secure HTTPS (port 443) connection.
- Logs off the Virtual User after the Test session, confirming the session's completion to the Appliance.

Engine User Interface

The Engine User Interface (UI) is a minimalistic toolbar that appears at the bottom of the desktop when the Engine process is running.

This UI is not intended for direct user interaction but provides useful status information during Test execution.

Appearance and Behavior

- **Location**: The Engine UI appears as a green bar at the bottom of the desktop.
- **Initial Display**: When the Engine begins running, the UI displays the message "Reading configuration…" as it communicates with the Virtual Appliance to retrieve the Test configuration.
- **Interactivity**: The green Engine UI toolbar is draggable, allowing you to move it around the screen.
- **Status Indicators**: The Engine UI shows different status messages, such as:
    - **Reading configuration**: Communicating with the Virtual Appliance.
    - **Workload name**: The current workload being executed.
    - **Logging off / Ending the Test**: Indicates the Engine is concluding the Test session.

Key Information Displayed

The following details are shown in the Engine UI to keep users informed about the Test status:

- **Repeat Count**: Displays the current iteration in a Load Test or Continuous Test. It indicates either the loop number or how many workload loops are configured to run.
- **Environment**: Shows the name of the Test configuration being used for the current Test.
- **Account**: Displays the Test user/Account that is logged in and running the Test on the Target desktop.

Pause/Play Functionality

A **Pause/Play** button is located on the right side of the Engine UI. By default, the **Pause** button is visible.

- **Pause**: When clicked, the Engine pauses indefinitely and waits for the user to click **Play** to resume the Test. This allows for troubleshooting or manual interaction with the desktop without the Virtual User interfering.
- **Resume**: Clicking **Play** resumes the Test, allowing the Virtual User to continue with the next steps in the Test plan.

NOTE: When the Engine is paused, it will finish the current workload before pausing and will not move on to the next action until resumed.

Additional Wait Messages

The Engine UI may also display a green wait message in the center of the screen, which is generated by the wait workload function. This message provides more specific details about the wait condition.

Engine Processes

This section highlights additional aspects of how the Engine functions during the Test execution, including its role in running executable files, collecting metrics, and executing workloads.

- **Test Discovery**: At startup, the Engine opens a connection to the Appliance and requests information about the Test pending for the current domain/user in this login session. If one exists, the Engine will download the Test configuration and begin to perform the actions. Otherwise, it will report an error and log out.
- **Data Collection and Upload**: The Engine gathers various performance metrics and uploads them to the Virtual Appliance for further analysis. These include:
    - **Logon Process**: Timing and details of the logon process.
    - **App Timers**: Timers for Application startup and interaction times.
    - **Session Metrics**: Session metrics, such as user activity and Application behavior.
- **Workload Execution**: The Engine runs Virtual User actions. These actions are typically implemented as C# workload scripts, which can either be:
    - **Out-Of-the-Box Workloads**: Pre-configured scripts for common testing scenarios.
    - **Custom Workloads**: You can create your own workloads using tools like the Script Recorder, App X-Ray, and the Script Editor to design custom Tests.

For more detailed information on the Virtual User actions, see the Scripting Functions Overview.

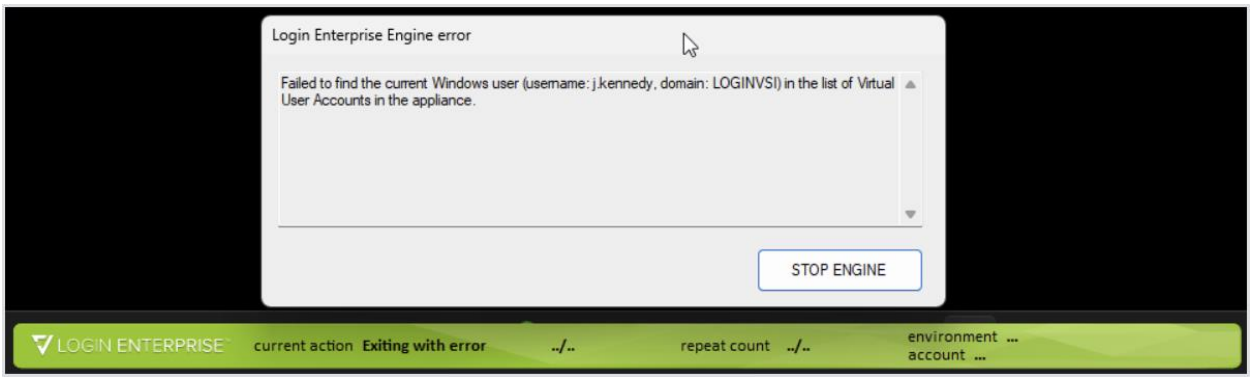
Core Components

The main components of this feature include:

1. **Logon Component**: Acts as a bootstrapper, downloading and unzipping the Login Enterprise Engine from the Appliance. For more information, see Configuring Logon Components.
2. **Engine Executable**: Facilitates the execution of Test workloads and the collection of performance metrics.

Engine Process Errors

- **User Configuration Error:** After the logon executable downloads and extracts the Engine folder (which contains the engine.exe), and the engine.exe is invoked, the Engine will check if the user running the Engine is configured as a Test user in the Virtual Appliance web interface. If the user is not found, an error is generated, as shown in the screenshot below. If this error occurs, the system will automatically log you off unless you've clicked the **Stop Engine** to prevent the logout.



- **Test Assignment Error:** If the user exists in the system but no Test is assigned to the user at the time of invocation, a different error occurs. This indicates that while the user is configured, no corresponding Test is available for execution at that point. In this case, the system will also log you off unless you've clicked the **Stop Engine** button to prevent the logout.

Log File Locations

Log files generated by the Login Enterprise Engine are located in:
`%temp%\LoginPI\Logs\Engine*.txt`

NOTE: Make sure to verify additional or updated log file paths if applicable.

Key Benefits

Benefit	Explanation
Agentless Deployment	The Login Enterprise Engine ensures target environments remain agentless. This approach avoids the performance impact and security hurdles associated with traditional agent-based solutions.
Hands-Off Operation	The Engine operates without requiring user intervention, making it inherently functional as soon as a synthetic user logs in.
Seamless Integration	Hosted on the Login Enterprise Virtual Appliance, the Engine is downloaded and executed per session, ensuring consistent operation across diverse environments.
Secure Communication	All communication between the Engine and the Virtual Appliance occurs over HTTPS (port 443), ensuring secure data transfer.

Value

The Login Enterprise Engine is indispensable for delivering actionable performance insights. By enabling Virtual User testing without agents, it saves time, simplifies deployment, and avoids the need for complex security approvals. As a result, you benefit from a lightweight, efficient, and secure solution for monitoring and optimizing system and application performance.

Managing Notifications

- Overview
- Global Email Notifications
  - Default notifications
  - Application notifications
    - Unique email notifications for each threshold
  - Email notifications
  - Global notifications
  - Email cool-down
- Continuous Test Notifications
  - Configuring email notifications

Overview

This section discusses the optional email notification features within Login Enterprise, divided into two main categories: Global Email Notifications and Continuous Test Notifications.

Type	Description
Global Email Notifications	Covers the configuration of global email settings, which are essential for sending alerts regarding system status. You will learn how to set up notifications for events such as when the appliance's external database is offline or if a Launcher is unavailable. This global configuration must be successfully established for the Continuous Test alert emails to function.
Continuous Test Notifications	Focuses on configuring email alerts specific to Continuous Tests. You will learn how to set notifications that alert you when measurement thresholds are exceeded, including the option to include recent measurement data. Additionally, you can manage the frequency of these notifications to ensure you stay informed without being overwhelmed by alerts.

By understanding both types of email notifications, you can effectively monitor the health of your Virtual Appliance and receive timely updates on Test performance.

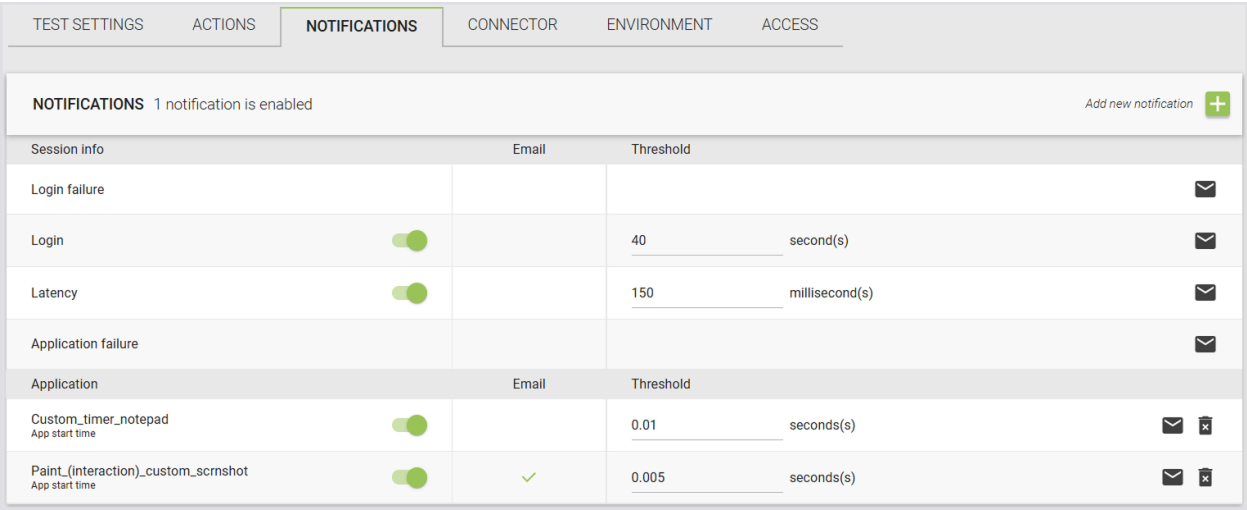
Global Email Notifications

Default notifications

When configuring Continuous Testing, you can define notifications for the conducted measurements. These notifications serve to alert you if actions or measurements deviate from expected behavior. There are 4 default notifications:

1. **Login Failure**: This notification triggers when a Virtual User is unable to log in to the configured environment (not applicable to the "desktop" connector).

2. **Login Time**: This notification checks if the login time exceeds the configured threshold. If this occurs, an event is generated, and depending on the configuration, an email may be sent.
3. **Latency**: This notification checks if the measured latency exceeds the configured threshold. If so, an event is generated, and based on configurations, an email may be dispatched.
4. **Application Failure**: This notification triggers an event whenever an application fails to start.



NOTE: Every timer defined in any Application script you have included in this Test is available for Notification. You must add the Application script first before adding the Notification.

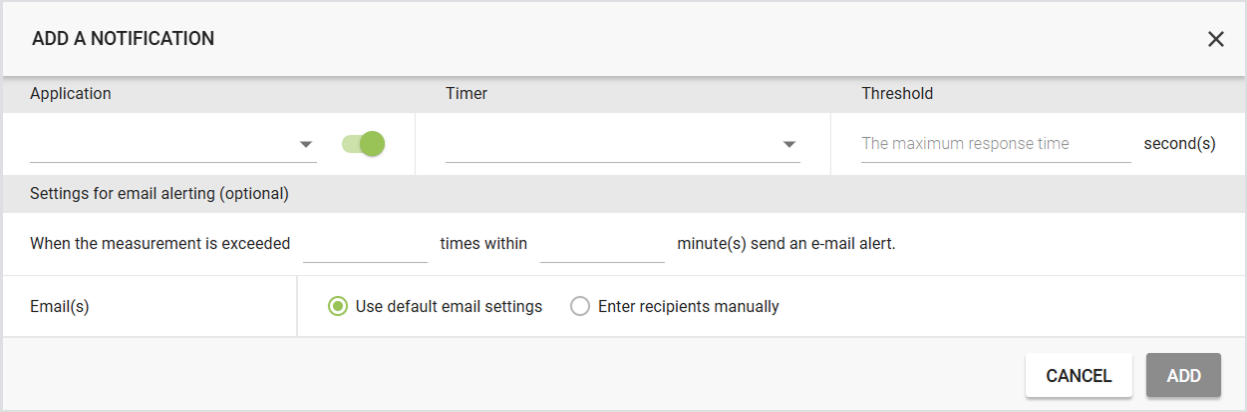
Application notifications

You can manually add application notifications and select the specific applications or measurements to monitor. You can then define the threshold manually. Whenever a measurement exceeds the defined threshold, an event is generated. As illustrated in the screenshot, you can configure email notifications to be sent when a measurement exceeds a specific number of occurrences within a specified time frame.

To add a new Notification:

1. Click the green "+" on the right to add a notification.

2. In the **Add a Notification**, specify the following:

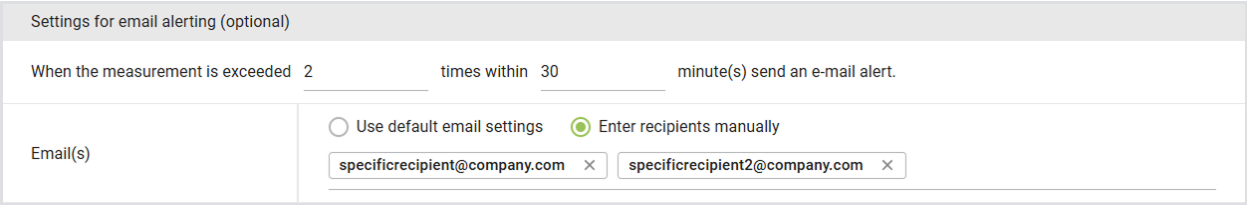


Unique Email Notifications for Each Threshold

Configuring unique email notifications for different thresholds allows you to specify distinct email recipients based on which threshold is breached. This is particularly useful for alerting specific stakeholders relevant to particular thresholds, providing more granular control over notifications compared to the previous global email notification settings.

To configure a unique email notification for each threshold:

1. Identify the threshold for which alerts should be sent to a specific subset of stakeholders.
2. In **Settings for Email Alerting** (Optional), use the radio buttons to select the desired email options.
    1. **Use default email settings**: The threshold will inherit email settings from the Global Email Notifications settings.
    2. **Enter recipients manually**: Input semi-colon-separated email addresses for those who should be notified when the conditions for alerting are met. This option overrides the global settings on a threshold-by-threshold basis.



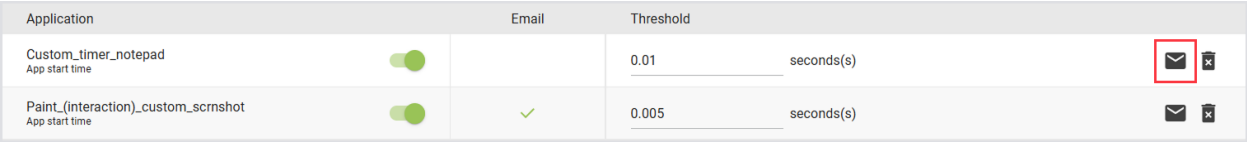
3. Click **Add** to enable the threshold.

After saving your settings, the new threshold configuration will be active, and email notifications will be sent according to the specified criteria. Ensure you review the alerting setup periodically to confirm it aligns with your stakeholders' needs and threshold requirements.

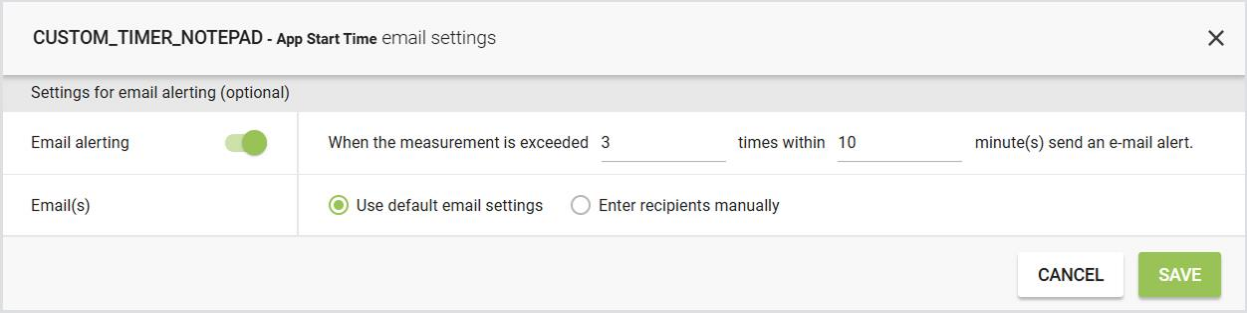
Email notifications

Email notifications for Continuous Tests include the last 5 measurements when thresholds are exceeded. If there are fewer than 5 measurements available, the email will indicate the actual number present. If there are no previous measurements (for example, because the Test Run has only started), this section will not be displayed.

To configure email notifications, you need to enable the email option for the measurements you want to stay up to date with. You can add email notifications to each measurement individually by selecting the 'letter' icon.



This will present you with two text fields and a switch.



You can define when you want to receive emails. This feature is designed to prevent you from being spammed whenever a spike occurs, giving you the ability to decide whether behavior is a fluke or a consistent issue. In the example provided, it is stated that when the measurement crosses the defined thresholds more than five times in one hour, it will send you an email.

An email typically looks like this:

Login time went over the threshold 1 times within 1 minutes

Test: 'Virtual Concurrent'
Threshold: 1.9 seconds

Last measurement:
[03/19/2024 17:03:31 UTC]: 2.92 seconds (54% over threshold)
Launcher: DESKTOP-8OR7AG2_
Account: 0b2a6ba6-4b5c-4dcf-872f-9eea62e9c4e1

Previous (5) measurements:

[03/19/2024 16:57:02 UTC]: 2.504 seconds (32% over threshold)
Launcher: DESKTOP-8OR7AG2_
Account: eab50724-42e5-4e35-96fc-579e688d6d8a

[03/19/2024 16:56:46 UTC]: 1.968 seconds (4% over threshold)
Launcher: DESKTOP-8OR7AG2_
Account: 95328a2e-2a06-4aed-992e-389de0ced315

[03/19/2024 16:55:59 UTC]: 1.999 seconds (5% over threshold)
Launcher: DESKTOP-8OR7AG2_
Account: ec34adb8-20d7-4f13-8a4b-34d3a2c4318a

[03/19/2024 16:55:42 UTC]: 2.126 seconds (12% over threshold)
Launcher: DESKTOP-8OR7AG2_
Account: c6534b72-660d-4a0e-9f4b-6786a61b3017

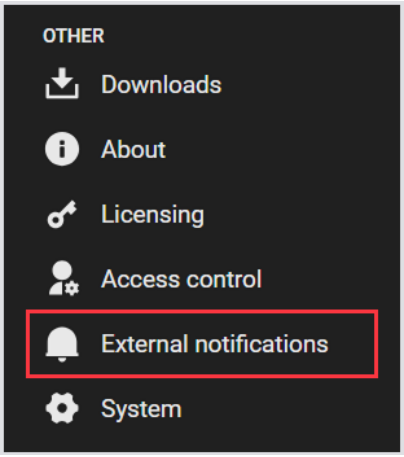
[03/19/2024 16:55:19 UTC]: 1.894 seconds
Launcher: DESKTOP-8OR7AG2_
Account: 6555a820-4811-4f04-b80a-a52bc2b16843

In the title, it states which Environment, and which Measurement or Application went over the threshold. In the email, you can find details about the threshold it exceeded (as configured). It also includes information about the environment, the threshold time measured, and the last five measurements, along with a remark indicating the percentage over the threshold.

Global notifications

Global notifications can notify you of such scenarios as disabled accounts or launchers going offline. To configure global notifications:

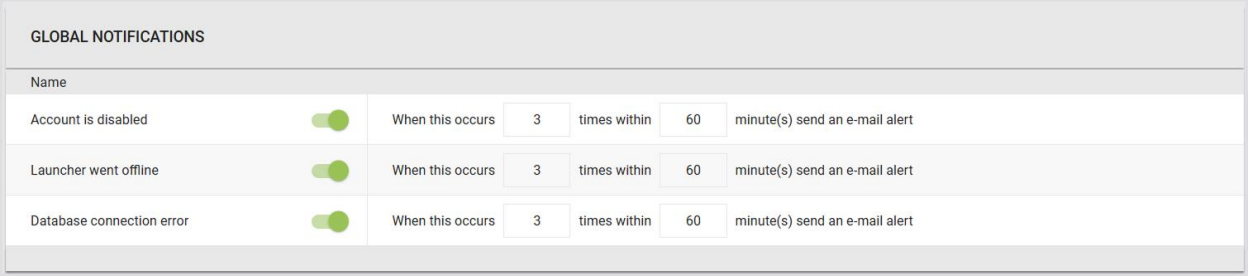
1. In the **Login Enterprise sidebar menu**, navigate to **Other > External notifications**.



2. In **External notifications**, scroll down to find **Global notifications**.

In **Global notifications**, you can configure the following settings:

Setting	Description
Account is disabled	Triggers when a user account is used but fails to become active. If this occurs a specified number of times within a specified timeframe, an email notification will be sent containing the name(s) of the affected user account(s).
Launcher went offline	Activates when a launcher fails to report and is marked as offline. If this happens more than a specified number of times within a specified timeframe, an email notification will be sent containing the name(s) of the affected launcher(s).
Database connection error	Activates when the database connection fails a specified number of times within a specified timeframe. In this case, an email notification will be sent.



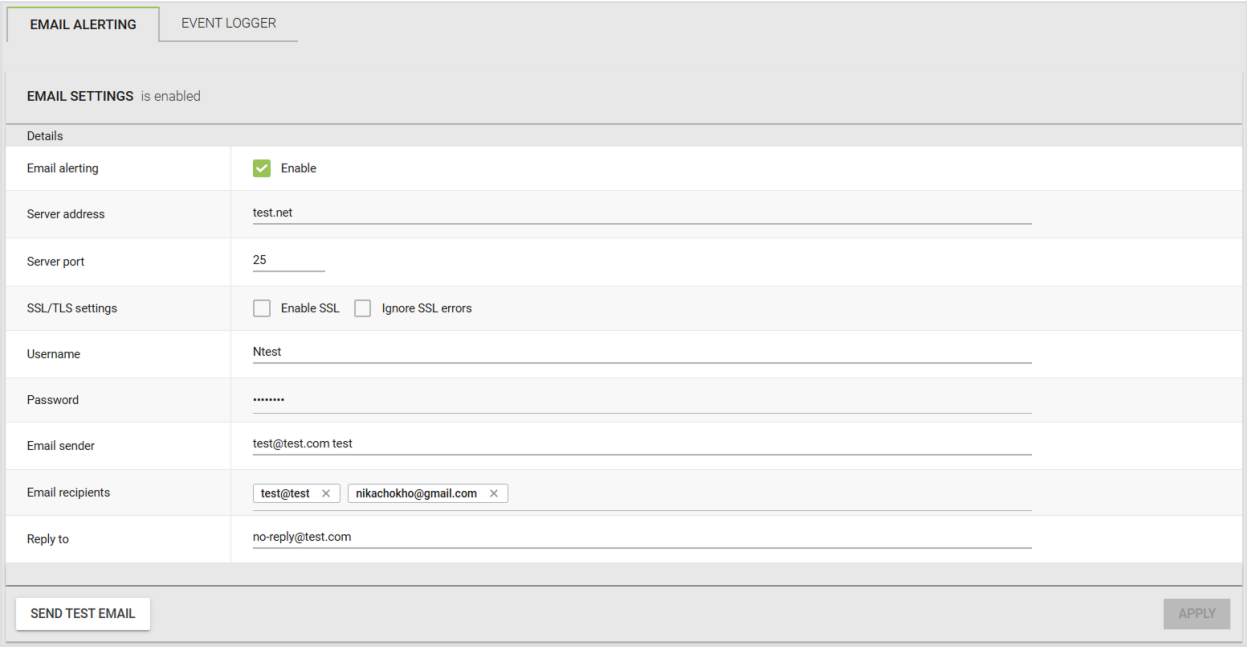
Email cool-down

After sending each email, there is a cool-down period in place to prevent excessive email notifications. The cool-down period is identical to the timeframe configured, as demonstrated by the 60-minute example in the screenshot above.

Continuous Test Notifications

Configuring email notifications

1. In the **Login Enterprise sidebar menu**, navigate to **Other > External notifications**.
2. In **Email altering**, provide your mail server details (To send email notifications to multiple recipients, enter multiple email addresses separated by commas).



3. Click **Send test email** to make sure your configuration is working.

When you have verified that your mail notifications are working, enable the email alerting and click **Done**.

You can now configure email alerts in **Continuous Tests > Test instance > Notifications**. For more information, see Continuous Testing.

Configuring the Event Logger

- Overview
- Possible events
- Integrating Login Enterprise with the Windows Event Log Service
  - Benefits of using the Login Enterprise Event Logging
  - Steps for integration

Overview

The Login Enterprise Event Logger records events generated by the Virtual Appliance on a Windows host with the Event Logger service installed.

By establishing a connection between the Virtual Appliance and the Windows host's Event Logger service, relevant events are seamlessly integrated into the Windows Event Logs.

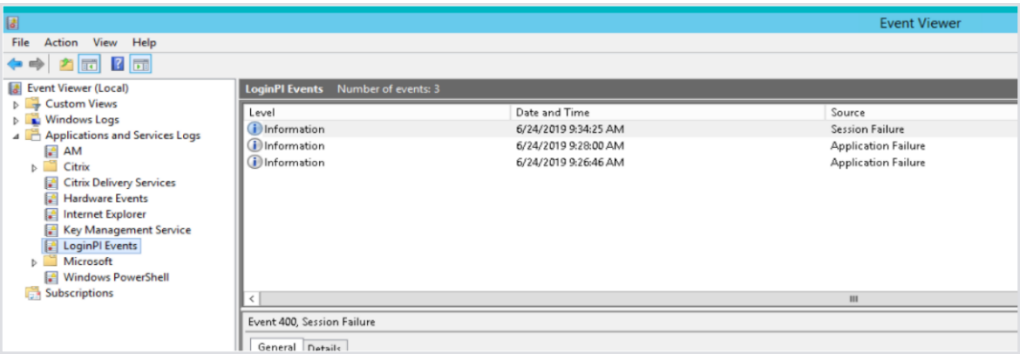
This integration offers several benefits:

Benefit	Description
ITSM Integration	Your ITSM tool can scrape the Windows Event Logs to automatically create tickets or trigger configured external automation.
Enhanced Analytics	The Event Logger on the Windows host can be used for further event analysis and data analytics, providing deeper insights into system performance and activities.

For information on installing and configuring the Login Enterprise Event logger, see Integrating Login Enterprise with the Windows Event Log Service.

You can find the logs in the Event Viewer of the system you have configured. Check the Applications and Services Logs folder for the LoginPI Events log.

NOTE: The log may be empty if your environment is running without events.



Possible events

- `accountCapacityExceeded`
- `accountDisabled`
- `appExecutionAbandoned`
- `applicationFailure`
- `applicationThresholdExceeded`
- `connectionInitializationTimeout`
- `customUserSessionEvent`
- `dataRetentionFailed`
- `dataRetentionFinished`
- `dataRetentionStarted`
- `databaseFailure`
- `emailFailure`
- `emailRequest`
- `emailSent`
- `engineLogs`
- `enginePaused`
- `engineResumed`
- `euxExecutionFailure`
- `euxInitializationFailure`
- `latencyMeasurementFailed`
- `latencyThresholdExceeded`
- `launcherCapacityExceeded`
- `launcherLogs`
- `launcherOffline`
- `licenseSessionLimit`
- `loginFailure`
- `loginTimeMeasurementFailed`
- `loginTimeThresholdExceeded`
- `remoteSessionDisconnected`
- `screenshotFailure`
- `scriptEvent`
- `scriptScreenshot`
- `sessionDiscoveryError`
- `sessionFailure`
- `sessionRequestEndedBeforeEngineBecameOnline`
- `testRunCancelled`
- `testRunFailed`
- `testRunFinished`
- `testRunStarted`

Integrating Login Enterprise with the Windows Event Log Service

With Login Enterprise, you can utilize the Windows Event Log Service to enhance your monitoring capabilities. This integration actively sends all events generated in Login Enterprise to a Windows machine's Event Viewer, allowing you to leverage this information in other monitoring tools like Microsoft SCOM.

The Login Enterprise Event Logger is a service connected to the Login Enterprise Appliance. Each event is pushed from the appliance to the Event Logger machine in real-time, ensuring there is no delay in event logging.

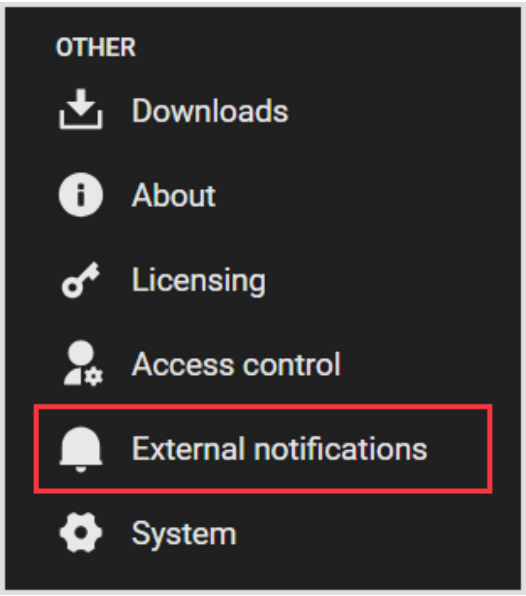
Benefits of using the Login Enterprise Event Logging

Benefit	Description
Real-time event logging	Events generated in Login Enterprise are immediately sent to the Event Logger machine, ensuring timely updates.
Enhanced monitoring	By integrating with the Windows Event Log Service, you can utilize the Event Viewer and other tools like Microsoft SCOM for comprehensive monitoring.
Seamless integration	The pushing mechanism is built into the event process, providing seamless and efficient event logging.

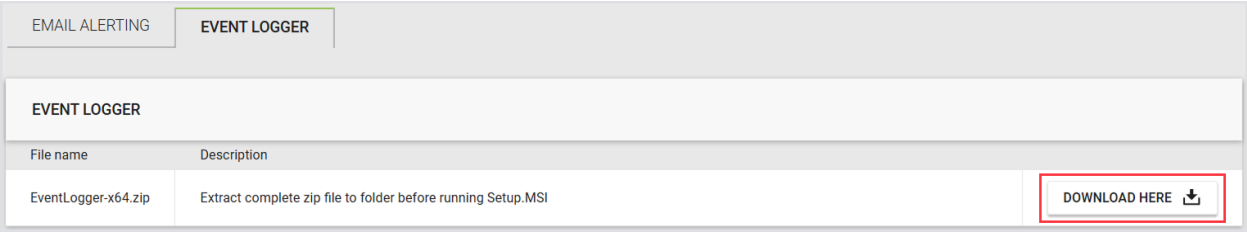
This integration streamlines event management and monitoring, offering a robust solution for maintaining system health and performance.

Steps for integration

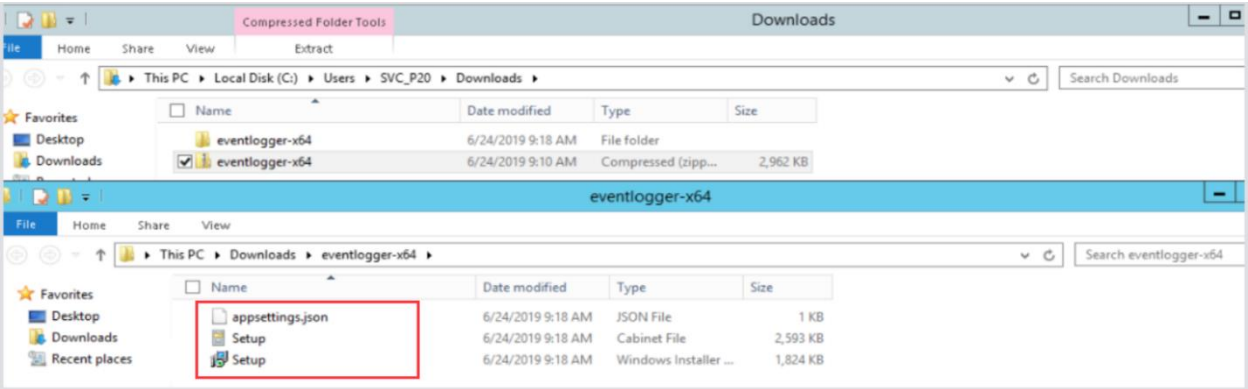
1. In the **Login Enterprise sidebar menu**, navigate to **Other > External notifications**.



2. In **External notifications**, select **Event Logger** from the tab menu.
3. Download the **EventLogger-x64.zip** and extract the contents.



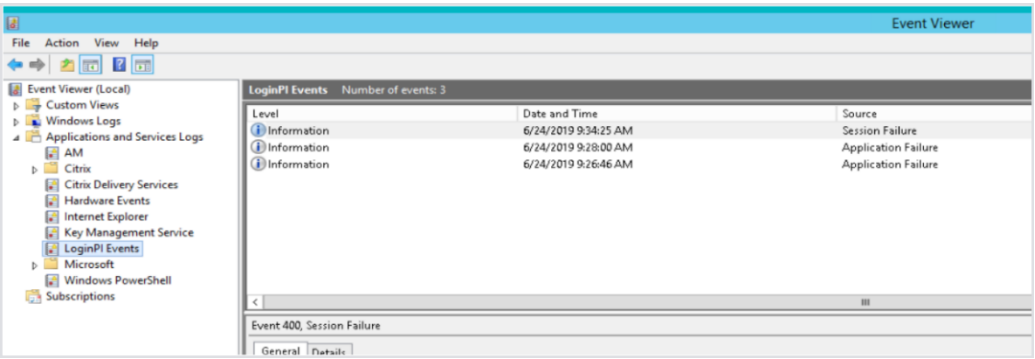
4. You should see 3 files inside the folder.



5. Execute the Setup.MSI and follow the setup wizard to install the application. Once completed, the service will be installed. To verify this, open the Services window on your Windows machine and check that it is running.

6. You can also open the Event Viewer and check the Applications and Services Logs folder to see if the LoginPI Events log is present and if there are any entries.

NOTE: The log may be empty if your environment is running without events.



Updating Browser Drivers

- Overview
  - Preferred method: Automatic updates
  - Manual update method (fallback)
- Determining the current browser version
  - Downloading the Edge Driver
  - Downloading the Chrome Driver
  - Downloading the Firefox Driver
- Importing the Driver
  - Updating the Login Enterprise Appliance
    - Opening the console session
    - Uploading the driver
  - Updating Script Editor and X-Ray on the Appliance
    - Version file
  - Updating Application Script Editor and X-Ray locally

Overview

To maintain the smooth operation of browser automation scripts on your Virtual Appliance, it's crucial to keep the Chrome, Edge, and Firefox drivers updated. Chrome, Edge, and Firefox browsers regularly release updates, and outdated drivers can lead to script failures in Applications like the Script Editor and X-Ray.

Preferred method: Automatic updates

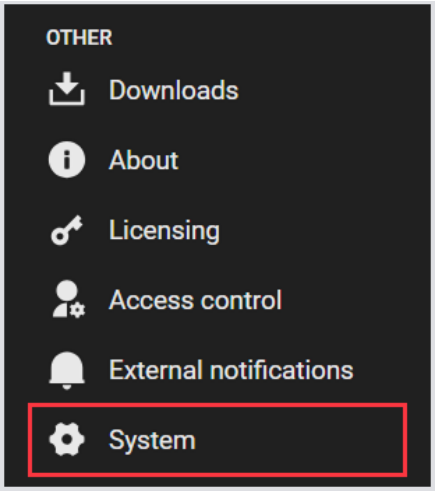
For most users, the preferred method is to configure your Virtual Appliance to perform automatic updates of the browser drivers. This ensures that your drivers are always up to date, providing compatibility and reliable performance across your automation environment. See the steps below for more information:

In Login Enterprise, you can update Selenium web automation drivers, ensuring you always have the latest driver versions for optimal performance.

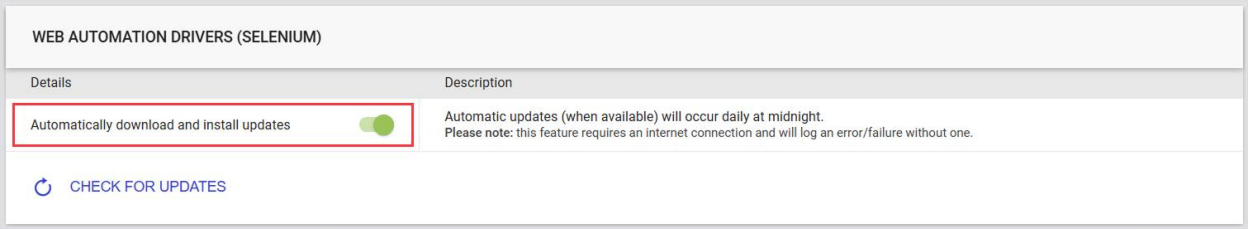
NOTE: This feature requires an internet connection. If you do not have internet access, you must update drivers manually.

Enabling automatic updates

1. In the **Login Enterprise sidebar menu**, navigate to **Other > System**.



2. In **System**, look for **Web Automation Drivers (Selenium)**.
3. In **Automatically download and install updates**, toggle the switch to green. The Appliance will check for driver updates daily at midnight and download any new versions of available drivers.

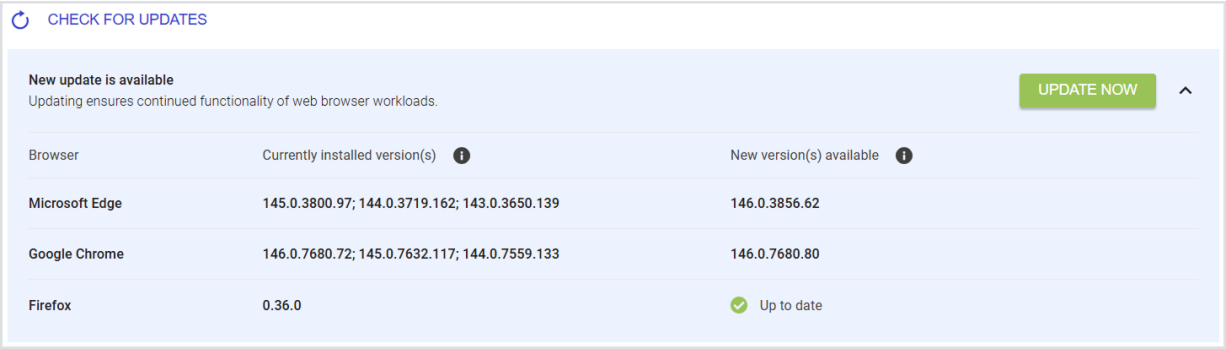


Manually checking for updates

To check for newer drivers, click **Check for updates**. If no updated drivers are available, you will receive an "**up to date**" notification, listing each major browser (Chrome, Edge, Firefox) along with their respective web automation driver (Selenium) versions.

Updating drivers

When updates are available, you will see a "**New update is available**" message indicating which drivers need to be updated. Click **Update Now** to perform the update. A loading spinner will appear while the download and update are in progress.



NOTE:

- By default, automatic downloads are disabled.
- Driver updates are only available if the Appliance is connected to the internet.
- For Chrome and Edge, updates for minor versions are performed across the latest 3 major versions.
    - Updating to a newer minor version overwrites the previous minor version, ensuring only 1 minor version is stored per major version.
- For Firefox, only the latest ESR version is supported.
    - A maximum of 1 minor version of the latest major version is stored.
- Updating to a new major version will not delete previous major versions, but the UI will only list the latest 3 installed versions.

Manual update method (fallback)

These are step-by-step instructions on how to manually update the drivers for Chrome, Edge, and Firefox.

NOTE: This method is intended as a fallback option for customers who deliberately do not have the Appliance connected to the internet. Use it only if automatic updates are not possible in your environment.

Determining the current browser version

Before proceeding with driver updates, it's essential to verify the current version of your browser on the target machine. Follow these steps to check:

1. Open a connection to the target machine (where the Login Enterprise will run or where you are creating your script).
2. Check the version of your browser using the following links:

- Edge
- Chrome
- Firefox

Downloading the Edge Driver

1. Open the Edge Driver download link.
2. Scroll down to see the latest drivers.
3. Click the **x64** download link next to the version you are running internally.

Downloading the Chrome Driver

1. Open the ChromeDriver download link.
2. You will see a folder overview. Locate the version you are using from the list and open that link.
3. Download the file named **chromedriver_win32.zip**.

Downloading the Firefox Driver

1. Open the FirefoxDriver download link from GitHub.
2. Scroll down to find the Assets section. You might need to click Show All Assets to find the file you need.
3. Download the file ending with **win64.zip**.

Importing the Driver

You need to update the following items with the newly downloaded driver:

- Login Enterprise Appliance
- Application Script Editor and X-Ray

Updating the Login Enterprise Appliance

Opening the console session

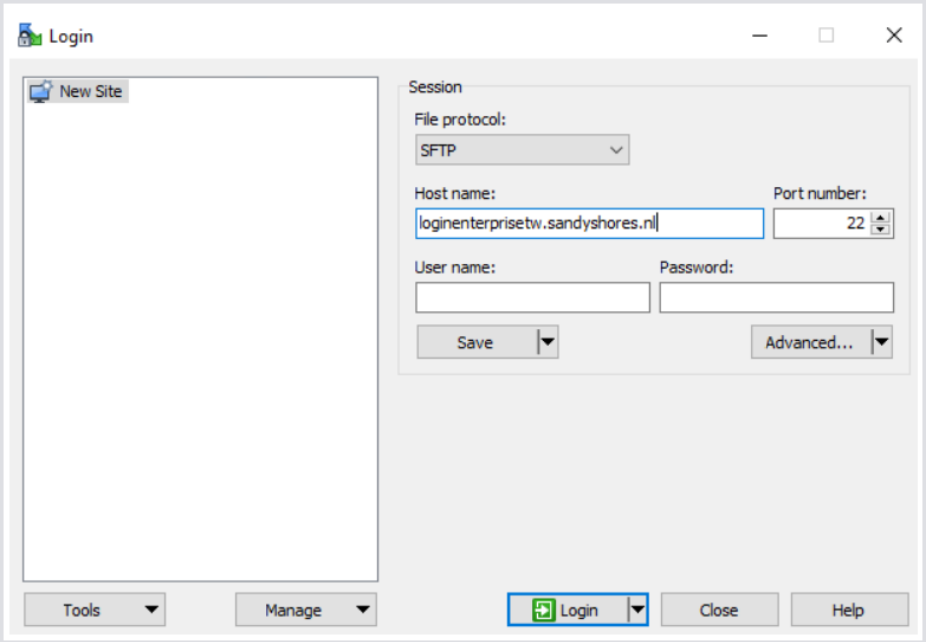
1. Open a console session to the Login Enterprise Appliance.
2. Navigate to **Troubleshooting > Open Bash Shell**.
3. Execute the following command to allow writing to the folder (use lowercase letters):

```
chmod 707 /loginvsi/content/selenium
```

Uploading the driver

1. **Connect with WinSCP**:

- Download and install WinSCP.
- Open WinSCP and set up an SFTP connection to the Login Enterprise Appliance.
- Use the admin password for the connection.



2. **Upload the Driver**:

- Browse to the `./loginvsi/content/selenium directory`.
- Upload the newly created driver folder, e.g. `Chrome91`, `EdgeChromium91, or Gecko34` to this location.

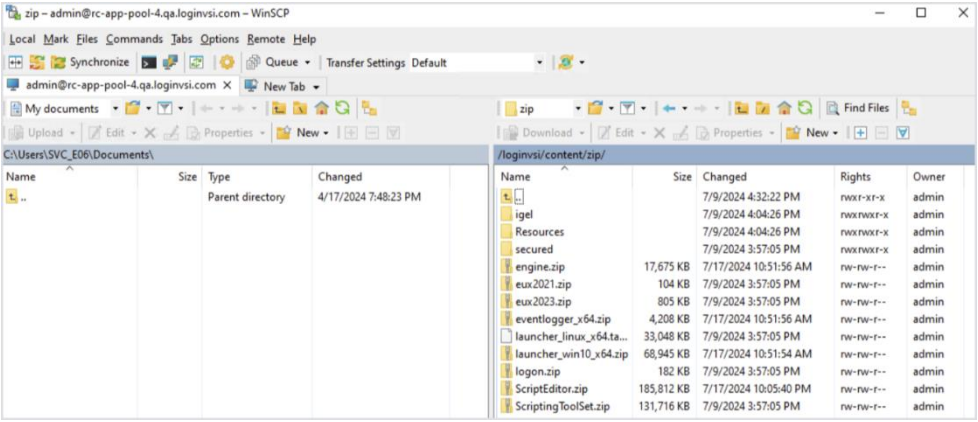
Example folder structure:

```
/loginvsi/content/selenium/Chrome91/chromedriver.exe

/loginvsi/content/selenium/EdgeChromium91/msedgedriver.exe

/loginvsi/content/selenium/Gecko34/geckodriver.exe
```

Updating Script Editor and X-Ray on the Appliance

The following steps will guide you through updating the ScriptEditor and ScriptingToolset on the Login Enterprise Appliance, ensuring any subsequent downloads from the Appliance include the updated browser drivers.

1. **Connect to the Appliance**: Use a tool like WinSCP on MyHost to establish a connection with the virtual Appliance. This opens a file browser interface for accessing the Appliance's file system.
2. **Download ScriptEditor.zip**:
    1. Navigate within WinSCP to the Appliance path: `/loginvsi/content/zip/`.
    2. Download the `ScriptEditor.zip` file from the Appliance to MyHost.



3. **Update Browser Driver**:
    1. On **MyHost**, extract the `ScriptEditor.zip` file.
    2. Locate the `Drivers` subdirectory within the extracted `ScriptEditor` folder.
4. **Add new Driver**: Insert or replace the outdated browser driver files (`chromedriver.exe`, `msedgedriver.exe, or geckodriver.exe` with the new driver files into the `Drivers` subdirectory.
5. **Replace** `ScriptEditor.zip` **on the Appliance**: Upload the updated `ScriptEditor.zip` from MyHost back to `/loginvsi/content/zip/` on the Appliance, replacing the existing `ScriptEditor.zip`.
6. **Verify updates**: Confirm that downloads of `ScriptEditor.zip` and `ScriptingToolset.zip` through the Login Enterprise web interface now include the updated browser driver.

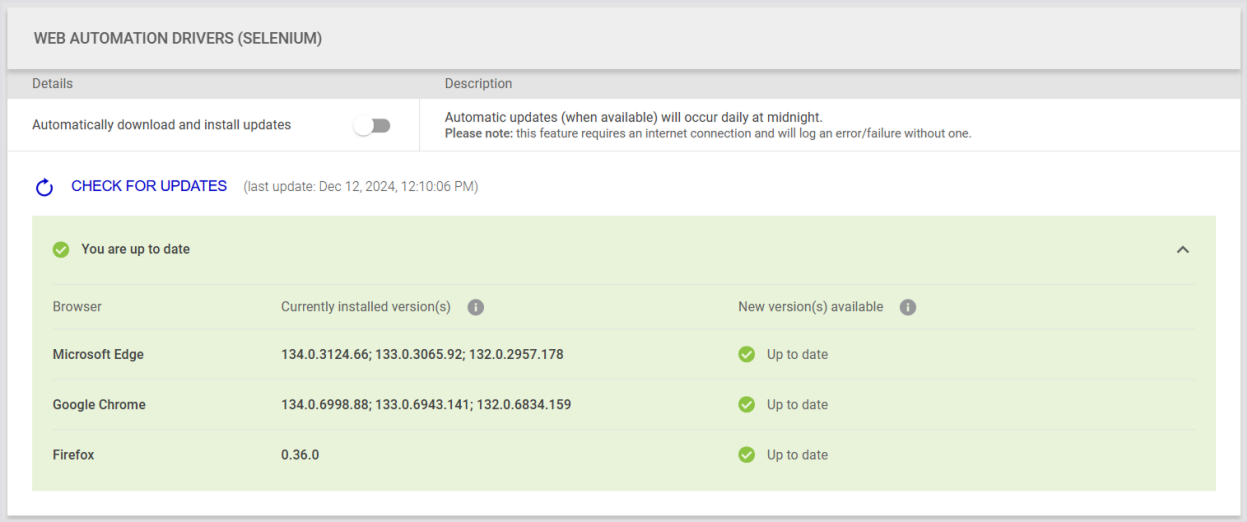
By following these instructions, you should be able to successfully update the Chrome and Edge drivers on your Virtual Appliance.

Version file

The version file is used solely to verify if Login Enterprise is using the latest version of drivers. If you do not manually add the version file during the update process, the **Currently installed version(s)** column in the **Login Enterprise sidebar menu > System > General** might not display the manually updated driver. So, when you check for updates, you might see available updates even if you already have the latest drivers installed.

Login Enterprise automatically utilizes the latest drivers regardless of the version file. If you want the Login Enterprise UI to display the driver that was manually uploaded, add a version file named "**version**" (without any extension) in the same folder as the driver, e.g., Edge Driver or Chrome Driver. This "version" file is essential for the system to verify the latest driver version.

You can download the version file example here.

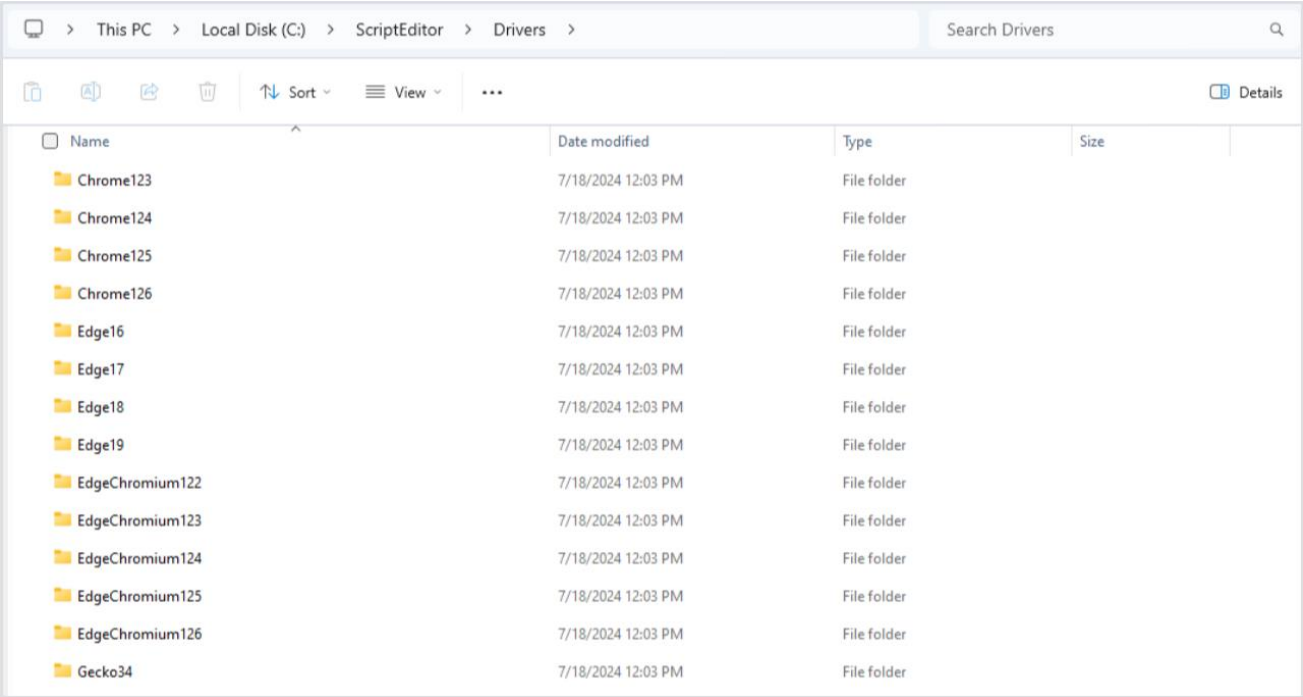


Updating Application Script Editor and X-Ray locally

For existing Script Editor installations on other hosts where scripting occurs, follow the steps below to manually update the drivers if necessary.

NOTE: Application X-Ray and Script Editor both use the exact same folder to access drivers. Therefore, updating the drivers in this folder will apply to both applications.

1. **Unzip the Driver**: Unzip the downloaded driver file. You will get either `chromedriver.exe` , `msedgedriver.exe`, or `geckodriver.exe`.
2. **Create a New Folder**:
    1. Navigate to the `ScriptEditor\Drivers` directory.
    2. Create a new folder named `Chrome(your.chrome.version)`, `EdgeChromium(your.edge.version)`, or `Gecko(your.firefox.version)`.
    3. Example:
        1. For Chrome version 91, create `Chrome91`.
        2. For Edge version 91, create `EdgeChromium91`.
        3. For Firefox version 0.34.0, create `Gecko34`.
3. **Move the Driver**: Copy the unzipped `chromedriver.exe`, `msedgedriver.exe`, or `geckodriver.exe` to the newly created folder.
    1. Example folder structure:
        1. `ScriptEditor\Drivers\Chrome91\chromedriver.exe`
        2. `ScriptEditor\Drivers\EdgeChromium91\msedgedriver.exe`
        3. `ScriptEditor\Drivers\Gecko34\msedgedriver.exe`


This PC
Local Disk (C:)
ScriptEditor
Drivers
Search Drivers
Sort
View
Details
Name
Date modified
Type
Size
Chrome123 7/18/2024 12:03 PM File folder
Chrome124 7/18/2024 12:03 PM File folder
Chrome125 7/18/2024 12:03 PM File folder
Chrome126 7/18/2024 12:03 PM File folder
Edge16 7/18/2024 12:03 PM File folder
Edge17 7/18/2024 12:03 PM File folder
Edge18 7/18/2024 12:03 PM File folder
Edge19 7/18/2024 12:03 PM File folder
EdgeChromium122 7/18/2024 12:03 PM File folder
EdgeChromium123 7/18/2024 12:03 PM File folder
EdgeChromium124 7/18/2024 12:03 PM File folder
EdgeChromium125 7/18/2024 12:03 PM File folder
EdgeChromium126 7/18/2024 12:03 PM File folder
Gecko34 7/18/2024 12:03 PM File folder

Using the Public API

- Overview
- Use cases
- Accessing Public API/Swagger UI
- System access tokens
  - Adding a system access token
  - Legacy system access tokens
- Common examples
  - Creating a Test
  - Starting a Test
  - Retrieving a list of environments
  - Retrieving dashboard data

Overview

Public APIs' primary use case is importing data into other applications, such as Splunk. This way, you can merge Login Enterprise data in your preferred dashboard or data collector to have all data centralized. The Public API of Login Enterprise is built on the REST Web API standard. You can use the APIs to pull data from the Login Enterprise instance. This data can then be used for your specific goal.

NOTE:

- Applies to Appliance v6.0 and higher: v4 and v5 of the Public API have been removed.
- Applies to Appliance v4.6 and higher: v3 of the Public API has been removed.
- Applies to Appliance v4.5 and higher: v1 and v2 of the Public API have been removed.

Use cases

Bulk Administration

Using the API makes managing your Login Enterprise environment at scale much easier. Instead of manually handling one-off tasks in the web interface, you can create, modify, and maintain multiple Tests or user accounts in a fraction of the time. For example, you might rely on tools like Postman to send automated requests that create or update a range of Tests in a single go.

Data Retrieval and External Analysis

The API provides quick, programmatic access to your Login Enterprise instance's Test results and performance metrics. This data can be fed directly into external analytics or reporting platforms, such as Microsoft Power BI, to generate custom dashboards, track long-term trends, and compare different Test sets. This approach helps ensure you’re getting the most out of your data, whether you’re looking for insights or just need a better way to visualize results.

Scheduling, Starting, and Stopping Tests

By tapping into the API's capabilities, you can set up Test runs according to your own schedule, kick off Tests on demand, or pause them as needed, all without clicking through the web interface. This gives you the flexibility to integrate testing into your existing workflows and processes, making it easier to keep Tests running smoothly at all times.

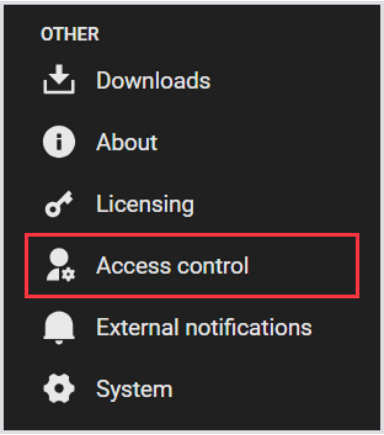
Integration with Third-Party Systems for Alerts and Maintenance

The API also simplifies integration with platforms like ServiceNow, making it possible to automatically create and manage ITSM tickets when critical issues arise. If a published desktop resource becomes unavailable, for example, you can have a ticket created automatically for quick follow-up. Beyond alerting, it's easy to connect to third-party automation platforms to perform maintenance tasks, such as restarting a problematic desktop, so you can address problems before they affect users.

Accessing Public API/Swagger UI

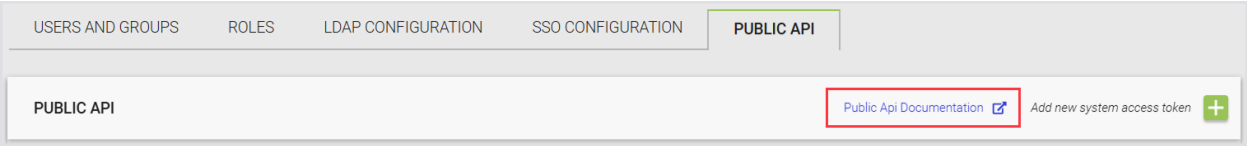
Login Enterprise provides a web-based mechanism to access the API for testing and validation, using the open-source Swagger suite of tools. Using the built-in Swagger page, you can explore and test using all of the API functions in Login Enterprise. You do not need to generate an API token to use Swagger; it will inherit the permissions of your web user.

1. In the **Login Enterprise sidebar menu**, navigate to **Other > Access control**.

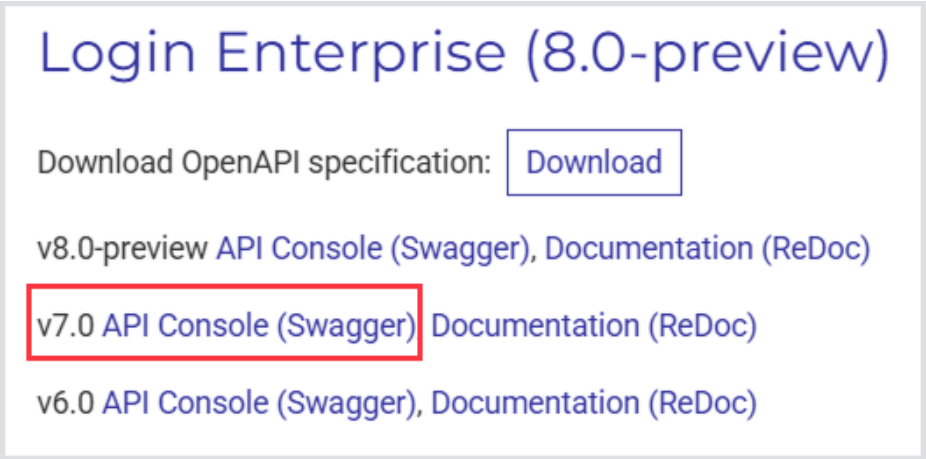


NOTE: In Login Enterprise v5 and earlier, the Public API is available under External notifications.

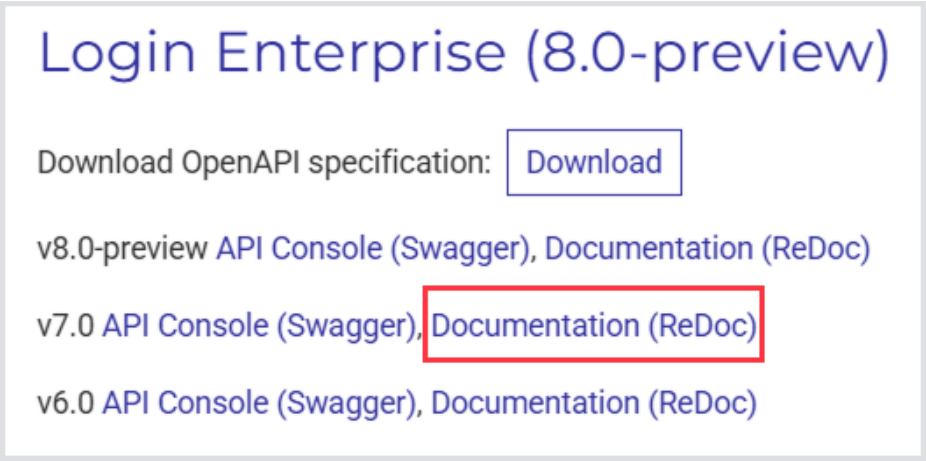
2. Select **Public API** from the tab menu and click **Public API documentation** (this will redirect you to the API Reference).



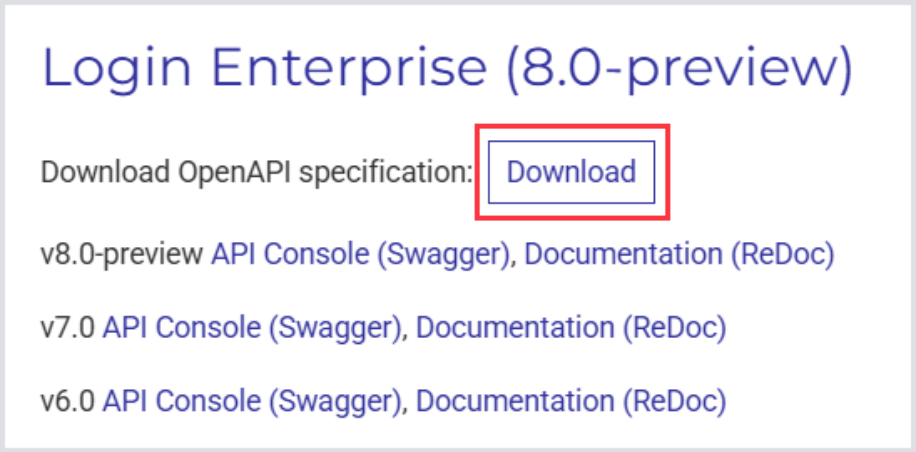
3. To interact with the API, choose a version (**v7.0** or **v8.0-preview**), and click on the **API Console (Swagger)** link. This will open the Swagger interface.



4. To see the API documentation in the ReDoc interface, click on the **Documentation (ReDoc)** link for the version you want.



5. To download our full OpenAPI specification, click **Download**:



System Access Tokens

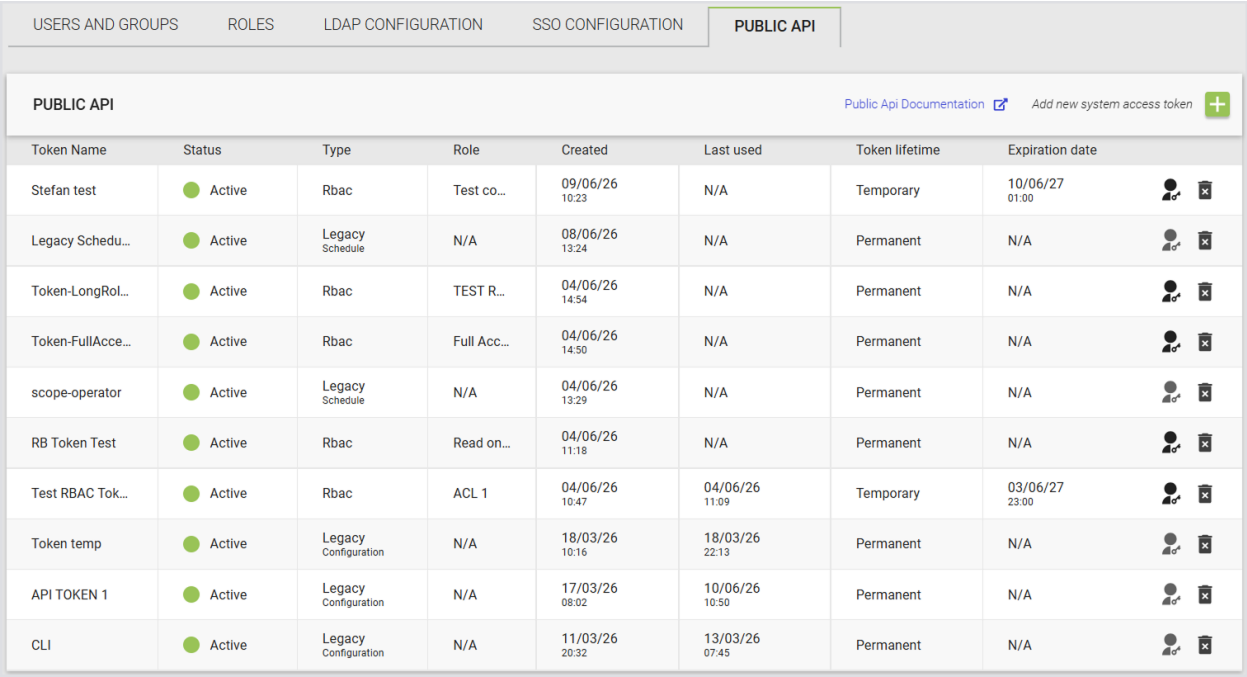
API tokens use RBAC to access the API using the permissions of a specific role. RBAC tokens are assigned to one role, and API calls made with the token use the permissions granted by that role.

Tokens created before version 6.7 are called Legacy tokens. They use one of the legacy access levels: Read-only, Configuration, or Schedule.

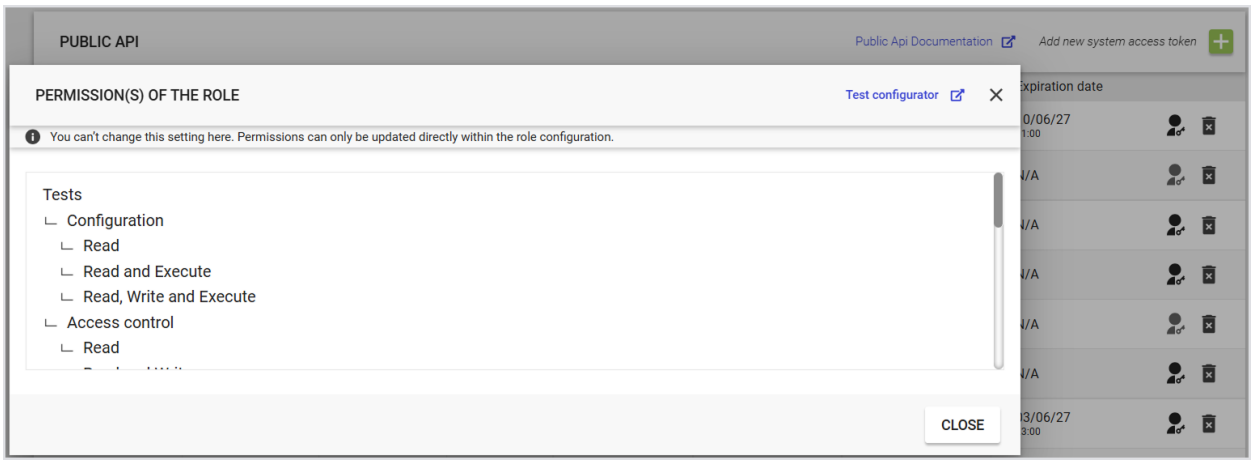
The System tokens page shows the available API tokens. Use this page to review token details, check token status, view role permissions, and delete tokens that are no longer needed.

Column	Description
Token Name	The name of the token.
Status	The current status of the token, such as Active.
Type	The token type: 1. **Legacy**: A token created before RBAC token support. Legacy tokens aren't assigned to an RBAC role. Instead, they use one of the static legacy access levels: Read-only, Configuration, or Schedule. The access level is shown under the Legacy token type. 2. **RBAC**: A token assigned to one RBAC role. API calls made with the token use the permissions granted by that role.
Role	The role assigned to the token. **NOTE**: Legacy tokens show N/A because they are not assigned to an RBAC role. For information on RBAC, see Role-Based Access Control (RBAC).

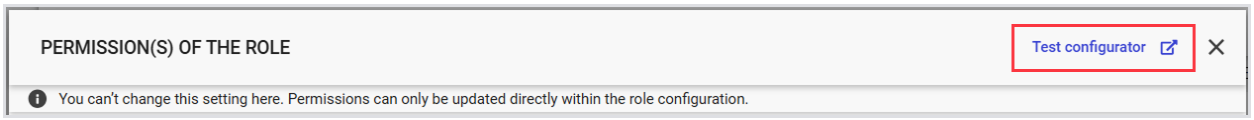
Created	The date when the token was created.
Last used	The date when the token was last used.
Token lifetime	Indicates the token lifetime: • **Permanent**: Doesn't expire. • **Temporary**: Expires on the specified date.
Expiration date	The date when the token expires, if applicable.



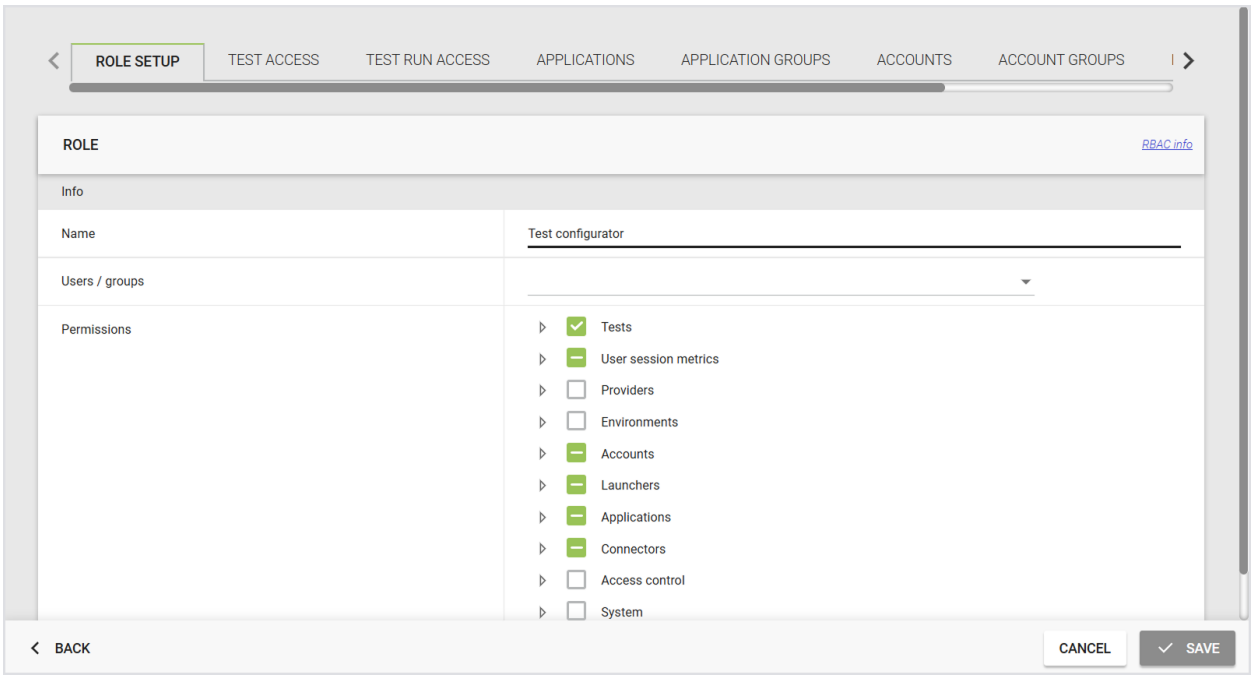
To view the permissions for a role, select the human icon next to the expiration date. The permissions list shows the permissions included in the assigned role, such as Test Configurator.



To review or change the permissions for the assigned role, click on the role name in the top right corner of the Permissions of the role window.



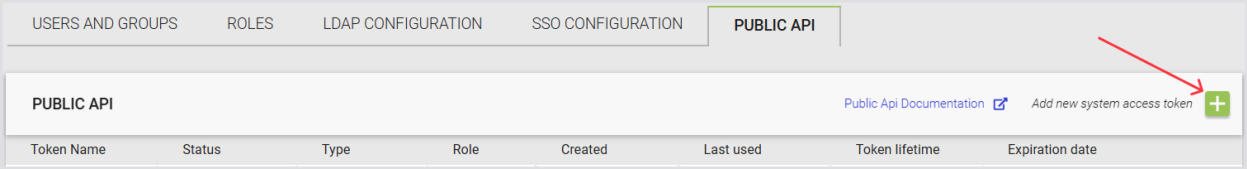
The Role Setup page opens, where you can review or change the role configuration.



To delete a token, select the Delete icon next to the token.

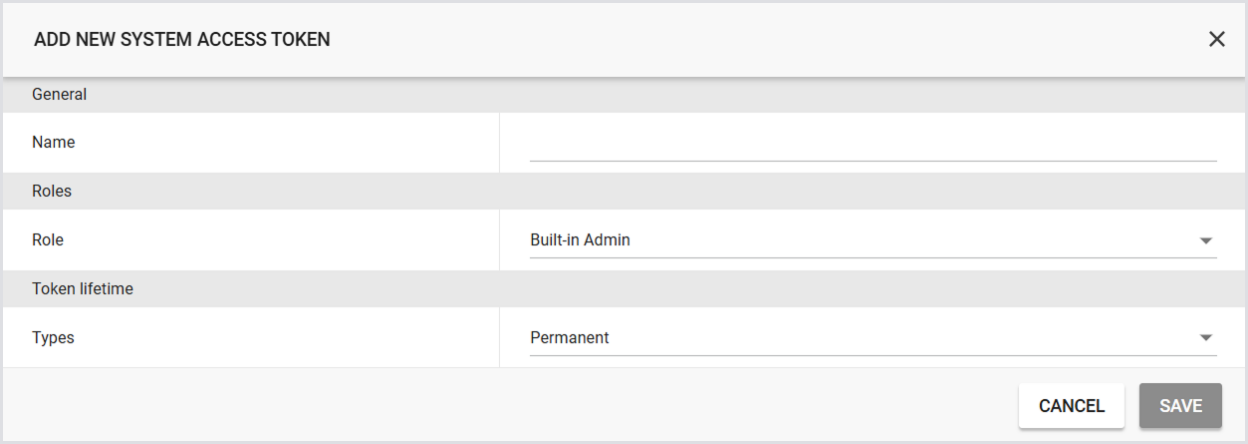
Adding a System Access Token

1. In **Public API**, click the "+" on the top right to add a new token.



2. In the **Adding a system access token** window, provide the following:

General	
Name	The name of the token.
Roles	
Role	Select a role. This field is required. Built-In Admin is selected by default. For details on roles and permissions, see Role-Based Access Control (RBAC).
Token lifetime	
Types	The token lifetime: Permanent or Temporary. If you select Temporary, specify an expiration date.



3. Click **Save**.

NOTE: System access tokens can't be updated after they're created. To change the permissions used by a token, update the permissions for the assigned role or create a new token with a different role.

Legacy System Access Tokens

Legacy system access tokens were created before RBAC tokens were introduced. These tokens use one of the legacy access levels: Read-only, Configuration, or Schedule.

Existing legacy tokens remain valid and unchanged. However, new system access tokens are created as RBAC tokens.

Common examples

Creating a Test

You can use a PowerShell script to create a Continuous Test in your environment using the RDP connector with the multiple-host function. The script also attaches an Account Group and Launcher Group, as well as an RBAC Role, if they are configured. To learn how to do this, see Creating Account Groups, Creating Launcher Groups, and Role-Based Access Control (RBAC).

TIP: You can download the script from here.

Requirements

To use this script, you will need the following information:

- **Virtual Appliance URL:** ($BaseURL)
- **API Token:** ($ConfigToken). The role assigned to the token must include the required permission. For details, see Managing Roles and Permissions.
- **Account Group Name:** ($AccountGroupName)
- **Launcher Group Name:** ($LauncherGroupName)
- **Role Name**: ($RoleName)

These values should be specified at the top of the script:

```
## Script Variables
$baseUrl = "loginenterprise.myenvironment.nl"
$ConfigToken = "BifSUhkFfEKkLb7tqnYUtYiNBkaJA4OgrdemwoLM71Q"
$AccountGroupName = "Account Group Name"
$LauncherGroupName = "Launcher Group Name"
$RoleName = "Role Name"
```

The section following the script variables sets up SSL connection certificates. By default, the script ignores certificate errors, allowing it to work with both self-signed and official certificates. For more information on certificates, see Managing Certificates.

Script functions

The script is built using three functions:

1. **get-AccountGroupID**: Checks if the configured Account Group name exists in the Virtual Appliance and returns its ID if found.
2. **get-LauncherGroupID**: Checks if the configured Launcher Group name exists in the Virtual Appliance and returns its ID if found.
3. **get-RoleID**: Checks if the configured Role name exists in the virtual appliance and returns its ID if found. You can also specify the UUID of the Role instead.
4. **Create-Test**: Sends the API request to create the Test entry based on its own `$CreateBody` and the retrieved Account, Launcher Group, and Role IDs.

To modify the Test type, connector, and other parameters of a Test case, adjust the `$CreateBody` variable:

```
$CreateBody = @"
{
    "type": "ContinuousTest",
    "name": "Your Test Name",
    "description": "My New Test",
    "connector": {
        "type": "Rdp",
        "hostList": [
            {"enabled": true, "endpoint": "Host1"},
            {"enabled": true, "endpoint": "Host2"},
        ],
        "suppressCertWarn": "True",
    },
    "accountGroups": [$Global:AccountGroupId],
    "launcherGroups": [$Global:LauncherGroupId]
}
"@
```

Additionally, there are two global definitions: `$header` and `$body`:

1. `$header` is used for all API requests and defines the authorization method.
2. `$body` is used to find the Account Group ID and Launcher Group ID.

```
## Script Variables
$baseUrl = "loginenterprise.domain.lab"
$ConfigToken = "PublicAPIToken"
$AccountGroupName = "Account Group Name"
$LauncherGroupName = "Launcher Group Name"
# Optional, can also specify Role UUID instead
$RoleName = "LDAP Admins"

# WARNING: ignoring SSL/TLS certificate errors is a security risk
$code = @"
public class SSLHandler
{public static System.Net.Security.RemoteCertificateValidationCallback
GetSSLHandler()
    {return new
System.Net.Security.RemoteCertificateValidationCallback((sender, certificate,
chain, policyErrors) => { return true; });}
}
"@
Add-Type -TypeDefinition $code

# WARNING: ignoring SSL/TLS certificate errors is a security risk
[System.Net.ServicePointManager]::ServerCertificateValidationCallback =
[SSLHandler]::GetSSLHandler()

# this is only required for older version of PowerShell/.NET
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls
-bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls12
```

```
# Define global authorization header for the script
$Header = @{
    "Accept"        = "application/json"
    "Authorization" = "Bearer $ConfigToken"
}

#Define global Body statements for finding the Launcher and Account group
name
$Body = @{
        "orderBy" = "name"
        "direction" = "desc"
        "count" = "100"
        }

## The Get-AccountGroupID function grabs all of the existing account groups
and compares the configured name ($AccountGroupName) with the list
## If the configured name is found in the list it returns the ID of that
account group ($global:AccountGroupID)
function get-AccountGroupID {

    #Parameters of the API command that combines all values and selects the
target API
    $Parameters     = @{
        Uri         = 'https://' + $baseUrl + '/publicApi/v8-preview/account-
groups/'
        Headers     = $Header
        Method      = 'GET'
        body        = $Body
        ContentType = 'application/json'
    }

    #Execute the command using the parameters and placing them in the
$request object.
    $request = Invoke-RestMethod @Parameters

    #setting counter to 0
    $i=0

    #Go through the array of items generated by the rest command
    ForEach ($item in $request.items)
    {
        $i++
        # Check each entry in array to check if the configured name exists in
the array, if not it does nothing.
        If($item.name -like $AccountGroupName){
           $global:AccountGroupID = '"' + $item.groupId + '"'
           Write-Host 'Found Account group match:'$item.groupId
           Break
        }

        # if the end of the array is reached and no match is found, report
and continue
        ElseIf($i -ge $request.items.count){
            Write-Host 'No Account Group name match found
for'$AccountGroupName ', Continuing script...'
            # set value to null so test is still created without an account
group
```

```
            $global:AccountGroupID = $null
        }

    }

}

## The Get-LauncherGroupID function grabs all of the existing launcher groups
and compares the configured name ($LauncherGroupName) with the list
## If the configured name is found in the list it returns the ID of that
launcher group ($global:LauncherGroupID)
function get-LauncherGroupID {

    #Parameters of the API command that combines all values and selects the
target API
    $Parameters     = @{
        Uri         = 'https://' + $baseUrl + '/publicApi/v8-
preview/launcher-groups/'
        Headers     = $Header
        Method      = 'GET'
        body        = $Body
        ContentType = 'application/json'
    }

    #Execute the command using the parameters and placing them in the
$request object.
    $request = Invoke-RestMethod @Parameters

    #setting counter to 0
    $i=0

    #Go through the list of items generated by the rest command
    ForEach ($item in $request.items)
    {
        $i++
        # Check each entry in array to check if the configured name exists in
the array, if not it does nothing.
        If($item.name -like $LauncherGroupName){
           [string]$global:LauncherGroupID = '"' + $item.id + '"'
           Write-Host 'Found Launcher group match:'$item.id
           Break
        }
        # if the end of the array is reached and no match is found, report
and continue
        ElseIf($i -ge $request.items.count){
            Write-Host 'No Launcher Group name match found
for'$LauncherGroupName ', Continuing script...'
            # set value to null so test is still created without an account
group
            $global:LauncherGroupID = $null
        }

    }

}
```

```
## The Get-RoleID function grabs all of the existing roles and compares the
configured name ($RoleName) with the list
## If the configured name is found in the list it returns the ID of that
account group ($global:RoleID)
function get-RoleID {

    if (-not $RoleName) {
        $global:RoleID = $null
        return
    }
    try {
        $throwaway = [guid]$RoleName
        Write-Host "Using $($RoleName) as RoleID"
        $global:RoleID = $RoleName
        return
    } catch {
        # Otherwise proceed with the lookup
    }
    #Parameters of the API command that combines all values and selects the
target API
    $Parameters     = @{
        Uri         = 'https://' + $baseUrl + '/publicApi/v8-
preview/auth/roles/'
        Headers     = $Header
        Method      = 'GET'
        body        = $Body
        ContentType = 'application/json'
    }

    #Execute the command using the parameters and placing them in the
$request object.
    $request = Invoke-RestMethod @Parameters

    #setting counter to 0
    $i=0

    #Go through the array of items generated by the rest command
    ForEach ($item in $request.items)
    {
        $i++
        # Check each entry in array to check if the configured name exists in
the array, if not it does nothing.
        If($item.name -like $RoleName){
           $global:RoleID = '"' + $item.id + '"'
           Write-Host 'Found Role match:'$item.id
           Break
        }

        # if the end of the array is reached and no match is found, report
and continue
        ElseIf($i -ge $request.items.count){
            Write-Host 'No Role name match found for'$RoleName ', Continuing
script...'
            # set value to null so test is still created without an account
group
            $global:RoleID = $null
        }
```

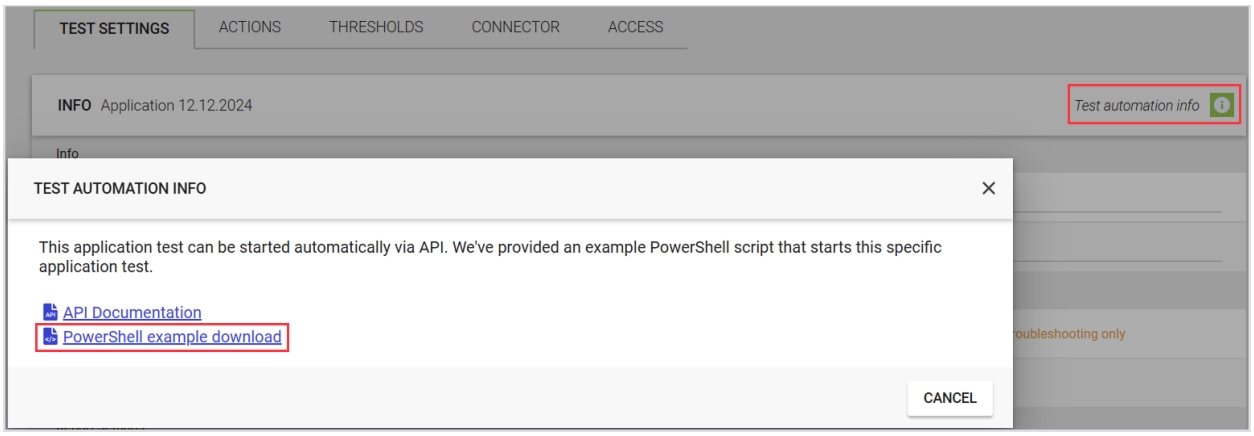
```
    }

}

function create-test {
## Create Body of API to create the new test, contains all required fields to
create a new test.
## This example makes use of the host function of the RDP connector
## for more info check the API documentation on the External Notifications >
Public API Page.

$CreateBody = @"
{
    "type": "ContinuousTest",
    "name": "Your Test Name",
    "description": "My New Test",
    "connector": {
        "type": "Rdp",
        "hostList": [
            {"enabled": true, "endpoint": "Host1"},
            {"enabled": true, "endpoint": "Host2"},
        ],
        "suppressCertWarn": "True",
    },
    "accountGroups": [$Global:AccountGroupId],
    "launcherGroups": [$Global:LauncherGroupId],
    "roles": [$Global:RoleID]
}
"@

    #Parameters of the API command that combines all values and selects the
target API
    $Parameters     = @{
        Uri         = 'https://' + $baseUrl + '/publicApi/v8-preview/tests/'
        Headers     = $Header
        Method      = 'POST'
        body        = $CreateBody
        ContentType = 'application/json'
    }

    #Execute the rest command, if a test with the same name already exists it
will generate an error. This command does not need an AccountGroupId or
LauncherGroupId to work
    Invoke-RestMethod @Parameters
}

#execute each function
get-AccountGroupID
get-LauncherGroupID
get-RoleID
Create-test
```

Starting a Test

A custom PowerShell script to start a Test is included with every Virtual Appliance release starting with v4.1. In v4.3 and later, specific example scripts are available for Continuous Testing, Load Testing, and Application Testing.

When you create an Application Testing scenario, you will find relevant information in the top right corner by clicking on **Test automation info**. A window with two links opens:

1. One for the Public API of your Appliance.
2. Another is for downloading the example PowerShell (.ps1) file.



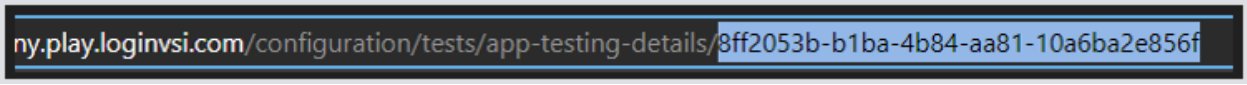
TIP: You can also download the script from here.

NOTE:

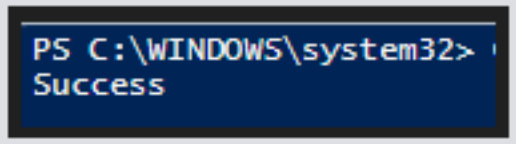
- If you download the script **from your own Appliance**, it will be pre-configured for your environment and Virtual Appliance.
- If you **download it from here**, you will need to configure the following 3 values:

```
[string]$accessToken = "U49-UKj5rtzP5cuL-PcsQsAQxtbGhwVT3OZesdLlUAE"
[string]$baseUrl = 'https://MyLoginEnterpriseApplianceURL/publicApi'
# Test id
$testId = '3a9f24a1-232f-4849-8a37-56117ba82734'
```

1. **$accessToken:** The role assigned to the token must include the required permission. For details, see Managing Roles and Permissions.
2. **$baseUrl:** This is the URL of your Virtual Appliance.
3. **$testId:** This is the GUID of your test, which can be found in the URL bar of your browser after opening an environment.



Once you have configured these values, you can execute the Test. If everything is set up correctly, you should see "Success" as the return value.



Example script

You can download example scripts for each use case by clicking the green exclamation mark button mentioned earlier. This section uses the Continuous Test script as an example. For more information on Continuous Testing, see Continuous Testing.

```
# basic information
# The access-token needs to be a System Access Token with the appropriate
access
# For more information see https://MyLoginEnterpriseAppliance/external-
notifications/public-api
[string]$accessToken = "U49-UKj5rtzP5cuL-PcsQsAQxtbGhwVT3OZesdLlUAE"
[string]$baseUrl = 'MyLoginEnterpriseAppliance/publicApi'

# Test id
$testId = '3a9f24a1-232f-4849-8a37-56117ba82734'
# Start-request data
$requestObject = @{
    # Comment
    'comment' = '...'
}

# if something goes wrong we stop processing this script
$ErrorActionPreference = 'Stop'

# this is only required for older version of PowerShell/.NET
[Net.ServicePointManager]::SecurityProtocol =
[Net.SecurityProtocolType]::Tls12 -bor [Net.SecurityProtocolType]::Tls11


# WARNING: ignoring SSL/TLS certificate errors is a security risk
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = {
return $true; }

# set the content-type and add the authorization http header with our access
token
$requestHeaders = @{
    'Content-Type' = 'application/json'
    'Authorization' = "Bearer $accessToken"
}

$requestBody = ConvertTo-Json -InputObject $requestObject
try {
```

```
    $response = Invoke-RestMethod -Method Put -Uri
"${baseUrl}/v6/tests/${testId}/start" -Headers $requestHeaders -Body
$requestBody
    Write-Host -Object 'Success'

} catch {
    [int]$statusCode = $_.Exception.Response.StatusCode;

    switch ($statusCode) {
        404 { Write-Host -Object 'Not Found' }
        409 { Write-Host -Object 'Conflict' }
        400 { Write-Host -Object 'Bad Request' }
        401 { Write-Host -Object 'Unauthorized' }
        default { throw }
    }
}
```

Retrieving a list of environments

This is a practical example of how to use API calls through a PowerShell script. The logic in this script can also be adapted for use in other scripting or programming languages, such as Python.

The script retrieves a list of available and configured environments. To learn more about the environments, see Environments. It's configured to ignore certificate errors if you are using a self-signed certificate. For more information on certificates, see Managing Certificates. You can utilize the authentication mechanisms within the script for any API endpoint. Depending on the specific API call, you can modify or retrieve data.

TIP: You can download the script from here.

To run this script, you will need the following:

1. **Virtual Appliance URL:** `https://myvirtualappliance.loginvsi.com`
2. **API token:** The role assigned to the token must include the required permission. For details, see Managing Roles and Permissions.

```
# basic information
# role based access API
[string]$publicApiSecret = 'PUBLICAPISECRET'

#url without https
$baseUrl = 'loginenterprise.my.url'

# if something goes wrong (e.g. authentication) we stop processing this
script
$ErrorActionPreference = 'Stop'

# WARNING: ignoring SSL/TLS certificate errors is a security risk
$code = @"
public class SSLHandler
{public static System.Net.Security.RemoteCertificateValidationCallback
GetSSLHandler()
```

```
    {return new
System.Net.Security.RemoteCertificateValidationCallback((sender, certificate,
chain, policyErrors) => { return true; });}
}
"@
Add-Type -TypeDefinition $code

# WARNING: ignoring SSL/TLS certificate errors is a security risk
[System.Net.ServicePointManager]::ServerCertificateValidationCallback =
[SSLHandler]::GetSSLHandler()


# this is only required for older version of PowerShell/.NET
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls
-bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls12

# Define authorization header
$Header = @{
    "Accept"        = "application/json"
    "Authorization" = "Bearer $publicApiSecret"
}

# Define body contents
$Body = @{
        "orderBy" = "Name"
        "direction" = "desc"
        "count" = "20"
}

#Parameters of the API command
$Parameters     = @{
    Uri         = 'https://' + $baseUrl + '/publicApi/v6/tests/'
    Headers     = $Header
    Method      = 'GET'
    body        = $Body
    ContentType = 'application/json'
}

#request data with call to the public api environments URL
$Results = Invoke-RestMethod @Parameters

#display each environment in the results
ForEach ($item in $Results.items)
{
Write-Host $item
}
```

Retrieving dashboard data

In some scenarios, you do not wish to view the Login Enterprise dashboard at all times, but you do want to use the data that is presented in your own custom dashboard. In these cases, you want to know how to retrieve this data. This section shows an example of a script based on PowerShell.

TIP: You can download the script from here.

Requirements

To ensure the script functions correctly, you will need the following items:

1. **Virtual Appliance URL:** ($WebURL)
2. **API Token:** ($ConfigToken). The role assigned to the token must include the required permission. For details, see Managing Roles and Permissions.
3. **Test GUID:** ($TestGuid)

These values should be specified at the beginning of the script:

```
## Script Variables
$WebURL = "https://Loginenterprise.my.domain"
$ConfigToken = "BifSUhkFfEKkLb7tqnYUtYiNBkaJA4OgrdemwoLM71Q"
$TestGUID = "3a9f24a1-232f-4849-8a37-56117ba82734"
```

The section following the script variables sets up SSL connection certificates. By default, the script ignores certificate errors, allowing it to work with both self-signed and official certificates. For more information on certificates, see Managing Certificates.

Script actions

The script will perform the following actions:

1. Retrieve the top two launcher locations.
2. Retrieve diagnostic tile data for all Continuous Tests (for the last hour) as found on the dashboard overview.
3. Retrieve diagnostic tile data for a specific Continuous Test (for the last 15 minutes).
4. Retrieve diagnostic tile data for a specific Continuous Test and Launcher combination (for the last hour).
5. Retrieve application statistics for a specific Continuous Test (for the last hour).
6. Retrieve application statistics for a specific Continuous Test and a specific launcher location (for the last hour).

```
## Script Variables
$WebURL = "https://Loginenterprise.my.domain"
$ConfigToken = "BifSUhkFfEKkLb7tqnYUtYiNBkaJA4OgrdemwoLM71Q"
$TestGUID = "3a9f24a1-232f-4849-8a37-56117ba82734"

# WARNING: ignoring SSL/TLS certificate errors is a security risk
$code = @"
public class SSLHandler
{public static System.Net.Security.RemoteCertificateValidationCallback
GetSSLHandler()
    {return new
System.Net.Security.RemoteCertificateValidationCallback((sender, certificate,
chain, policyErrors) => { return true; });}
```

```
}
"@
Add-Type -TypeDefinition $code

# WARNING: ignoring SSL/TLS certificate errors is a security risk
[System.Net.ServicePointManager]::ServerCertificateValidationCallback =
[SSLHandler]::GetSSLHandler()

# this is only required for older version of PowerShell/.NET
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls
-bor [Net.SecurityProtocolType]::Tls11 -bor [Net.SecurityProtocolType]::Tls12

# Define global authorization header for the script
$Header = @{
    "Accept"        = "application/json"
    "Authorization" = "Bearer $ConfigToken"
}

## Retrieve location site 1 and 2 in the list. To add more simply copy paste
and make changes to the numbers in the []
$Location1 = (Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/locations" -
Headers $Header -Method Get).Items[0].id
$Location2 = (Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/locations" -
Headers $Header -Method Get).Items[1].id

## Retrieve the diagnostic tile data on the dashboard for all continuous
tests for the last hour
Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/test-
diagnostics?timeRange=lasthour" -Headers $Header -Method Get

## Retrieve the diagnostic tile data on the dashboard for a specific
continuous test, for the last 15 minutes
Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/tests/$TestGUID/test-
diagnostics?timeRange=last15minutes" -Headers $Header -Method Get

## Retrieve diagnostic tile data based on the location dashboard per site for
the last hour, to expand duplicate the entries.
Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/tests/$TestGUID/test-
diagnostics?timeRange=lasthour&locationId=$Location1" -Headers $Header -
Method Get
Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/tests/$TestGUID/test-
diagnostics?timeRange=lasthour&locationId=$Location2" -Headers $Header -
Method Get

## Retrieve application statistics for a specific continuous test for the
last hour
Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/tests/$TestGUID/application-
diagnostics?timeRange=lasthour" -Headers $Header -Method Get

## Retrieve application statistics for a specific continuous test and a
specific launcher location, for the last hour
Invoke-RestMethod -Uri "$WebUrl/publicApi/v6/tests/$TestGUID/application-
diagnostics?timeRange=lasthour&locationId=$Location2" -Headers $Header -
Method Get
```

Back to Configuration

Managing Tests

Login Enterprise supports 3 Test types (Continuous, Load, and Application), each utilizing the same Application scripts and Engine but offering distinct scheduling options and purposes to meet diverse testing needs. In addition to the Login Enterprise UI, you can manage Tests using the Public API.

This chapter covers the following topics:

- Configuring Continuous Testing
- Configuring Load Testing
- Configuring Application Testing
- Connectors
  - Citrix Netscaler 12.1 and 13.0
  - Citrix Storefront
  - Microsoft RDS
  - VMware Horizon View
  - Windows 365 (Preview)
  - Desktop Connector
  - Custom Connector
    - Configuring the Universal Web Connector
    - Deploying the AVD Connector
    - Custom Events for Connectors and Launchers
    - Named Pipes for Custom Connectors
      - Custom Connector SDK (Named Pipe)
- Managing Session Metrics (Optional)
  - Managing Session Metrics Using the Public API
  - Session Metrics Library
  - Session Metrics Import Script (Optional)
- Configuring Providers (Optional)
- Configuring Environments (Optional)
- Configuring Locations (Optional)
- Configuring Applications for Testing
  - Troubleshooting Application Errors
- Setting the Engine Start Timeout on a Test

Configuring Continuous Testing

- Overview
- How it works
- Creating a Continuous Test
- Configuring the Continuous Test
  - Test Settings
    - Schedule Settings
    - Settings
    - Session Metrics
  - Actions
  - Notifications
  - Citrix Published Apps
  - Access
- Adjusting the Engine start timeout
- Starting a Continuous Test
  - Enabling Continuous Tests in bulk
- Locations
- Best practices
- Continuous Test results and comparison

Overview

Continuous Tests, also known as Continuous Monitoring, refer to a series of Tests that run consecutively to continuously monitor end-user experience. These Tests aim to promptly alert you to any issues in your infrastructure, preempting the need for users to report problems.

How it works

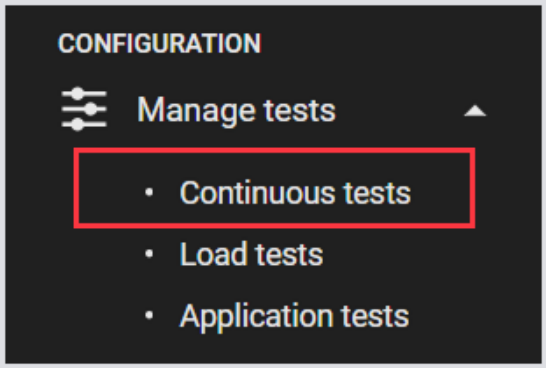
A Continuous Test records all timers defined in your Application scripts, along with Session Metrics from Windows and your connection broker. It can also execute our EUX Applications to gauge user experience on your hardware. Each metric can be charted over extended periods within your Test.

Continuous Tests initiate a session to log in as a specific user, execute the designated Test, and log out upon completion. The results from all executions are displayed as a single, continuous set, allowing you to monitor performance graphs over extended periods. You can configure email alerts to notify you when specific counters exceed their assigned threshold values within a specified interval, helping you identify performance degradation.

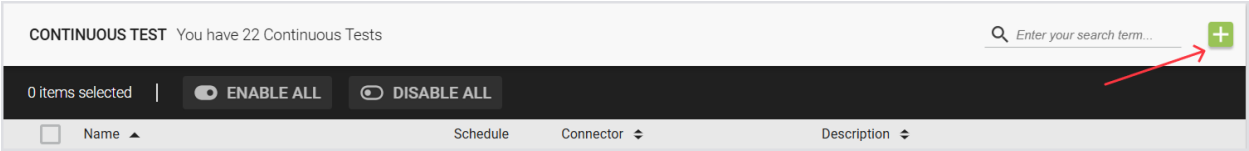
When viewing a Continuous Test in the **Login Enterprise sidebar menu > Results > Continuous Testing**, you'll initially encounter a chart displaying your metrics. This enables you to promptly assess the success of your Test and the consistency of your results. For more information, see Viewing Continuous Testing Results.

Creating a Continuous Test

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Manage Tests**, and select **Continuous Tests**.



2. In **Continuous Tests**, click "+" on the top-right.



3. Define the connection settings:

General	
Name	The display name for the Test.
Description (optional)	An informative description of the Test.
Environments (optional)	
Name	The target environment where the Test will be conducted. For more information, see Configuring Environments.
Connector	
Connector type	The type of Connector to be utilized by the Launcher. **NOTE:** Each Connector has its own settings. Once you define your Connector, you need to specify the necessary information in the Settings section. For details, see Connectors.
Access	
Roles	Select roles to control access to Test configurations and results. For more information, see Role-Based Access Control (RBAC).

Configuring the Continuous Test

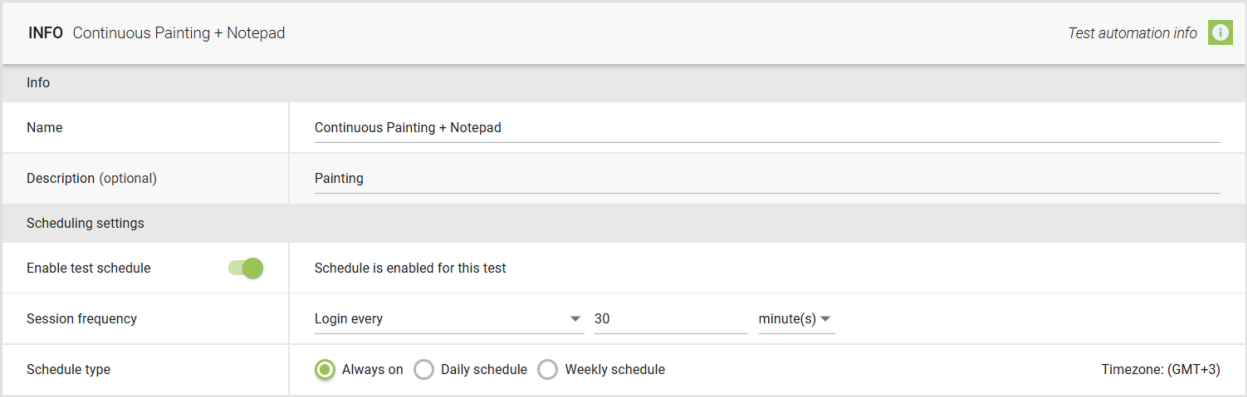
Test Settings

Once you've created the Test, you can configure Schedule, Settings, and Session Metrics.

Schedule Settings

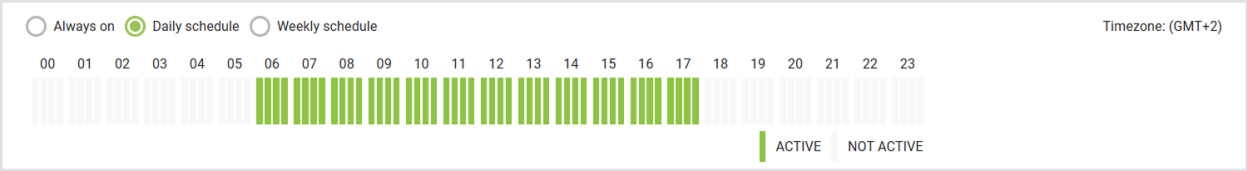
Scheduling settings		
Enable test schedule	Enable or disable the Test scheduling of the Test.	
Session frequency		
	Login every	• Launch a session every X minutes, utilizing Launchers in a round-robin pattern. ○ This option initiates a virtual user to log in, execute the Test, log off, and then repeat the cycle after waiting for the configured duration.
	Run a session from each Launcher every	• Launch a session from each Launcher every X minutes, utilizing all Launchers in the specified Launcher Group. ○ This option launches one session from every configured launcher for the environment within the specified time frame.

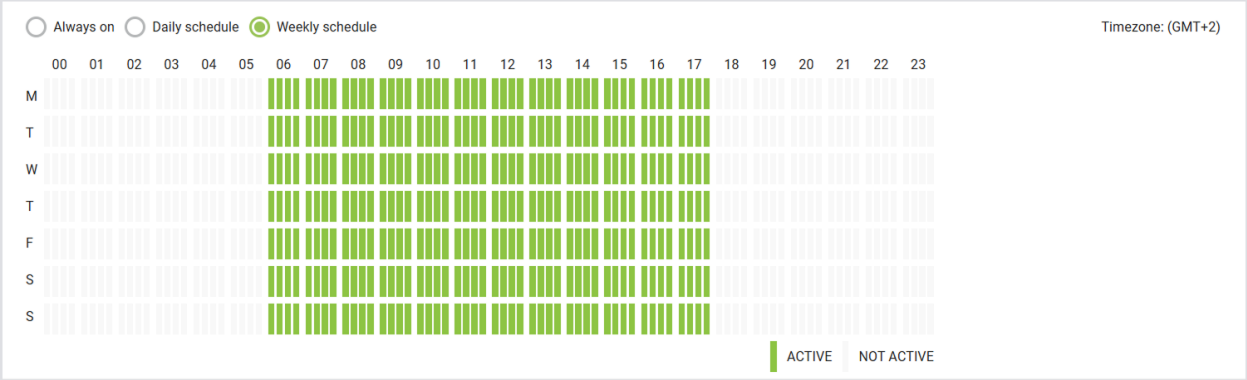
	Run concurrent sessions	• Launch X concurrent sessions, utilizing Launchers in a round-robin pattern. ○ With this option, users continuously log in, perform defined actions, log out, and immediately restart the process without waiting for a set time interval.	
Schedule type			
	Always on	Daily schedule	Weekly schedule
	Runs the Test continuously.	Runs the Test during specified hours each day.	Runs the Test during specified hours on selected days of the week.



The following examples show a schedule configured to run from 6:00 AM to 5:00 PM on a daily and weekly basis. This lets you exclude planned downtime from testing.

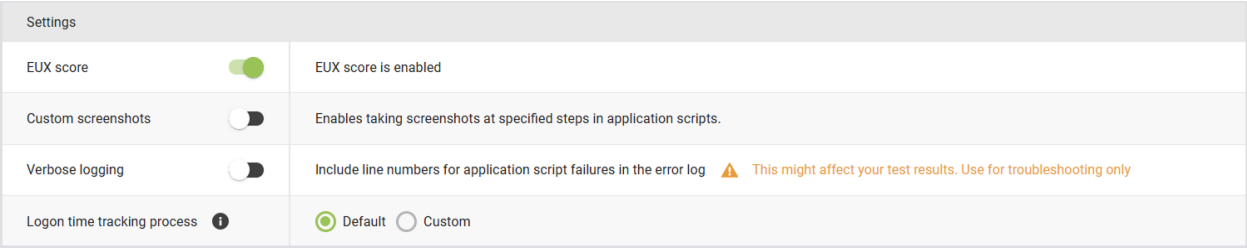
TIP: Use the scheduling interface to select time ranges. Each bar represents one hour, and each block represents 15 minutes. Save your changes to apply the schedule.





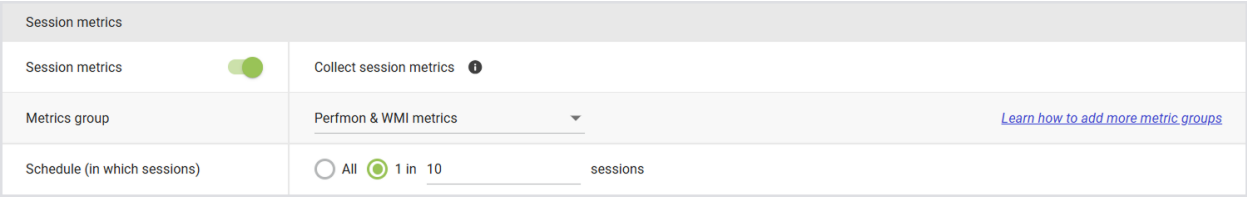
Settings

Settings	
EUX Score	Enable or disable the calculation of the End User Experience Score during Test Runs. For details, see EUX score and VSImax.
Custom screenshots	Enable or disable custom screenshots within Application scripts. For more information, see Taking Screenshots.
Verbose logging	Display the command line number that triggered an error during Application execution failures. **NOTE**: This may affect Test results as the line number is returned during execution.
Logon time tracking process	• Track the complete duration from when the Connector initiates the login process for the user session until the final user-session executable (usually Windows Explorer) is up and running. • You can also overwrite the default process of time tracking with a process of your choice. For more information, see Measuring total login time.



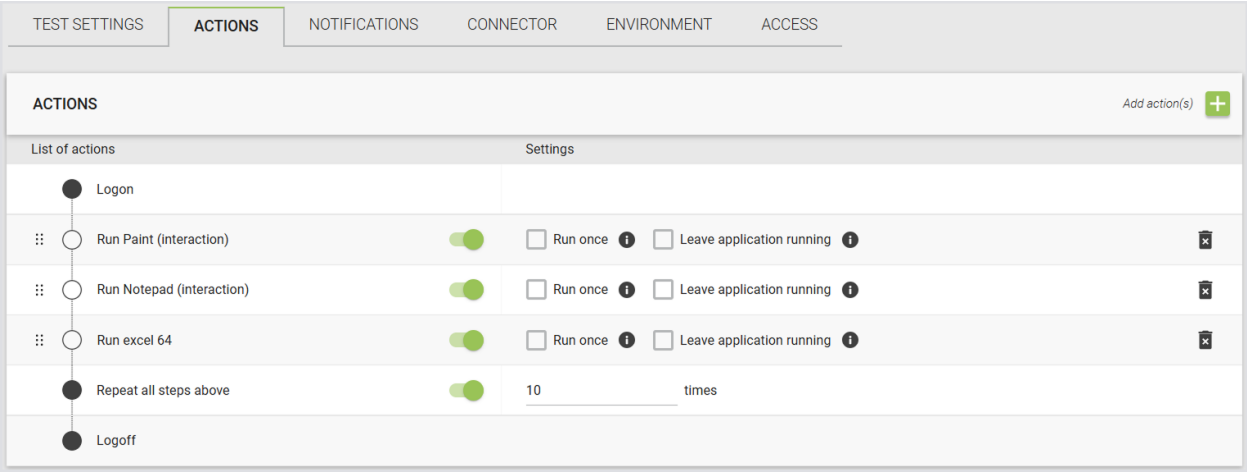
Session Metrics

Session Metrics	
Session metrics	Enable or disable the gathering of Session Metrics. System performance metrics from each Virtual User session will be collected periodically. The metrics collected are determined by the metric group you select below. For more information on Session Metrics, see Session Metrics.
Metrics group	The desired Session Metrics Group specifies what performance counters should be collected.
Schedule (in which session)	The schedule for sessions. If you don't want to collect Session Metrics in every session, specify how often it can run, e.g., 1 out of every X sessions.



Actions

You can add the preferred Applications for the Virtual User to test. You can also add Application Groups, and "Waits." For details, see Applications.



Notifications

One of the major advantages of Continuous Testing is that Virtual Users operate independently of any active real-user sessions. This allows them to effortlessly execute workloads, detect failures or performance degradation, and promptly and proactively alert IT Operations teams. For more information, see Notifications.

Citrix Published Apps

You can connect to a Published Application using either the StoreFront or NetScaler Connector. The key difference between connecting to a Virtual Desktop and a Published Application is that the resource configured is an Application rather than a desktop. For details, see Citrix Netscaler 12.1 and 13.0 and Citrix Storefront.

Schedule Configuration

Define the time schedule during which the virtual user is active.

Actions List Configuration

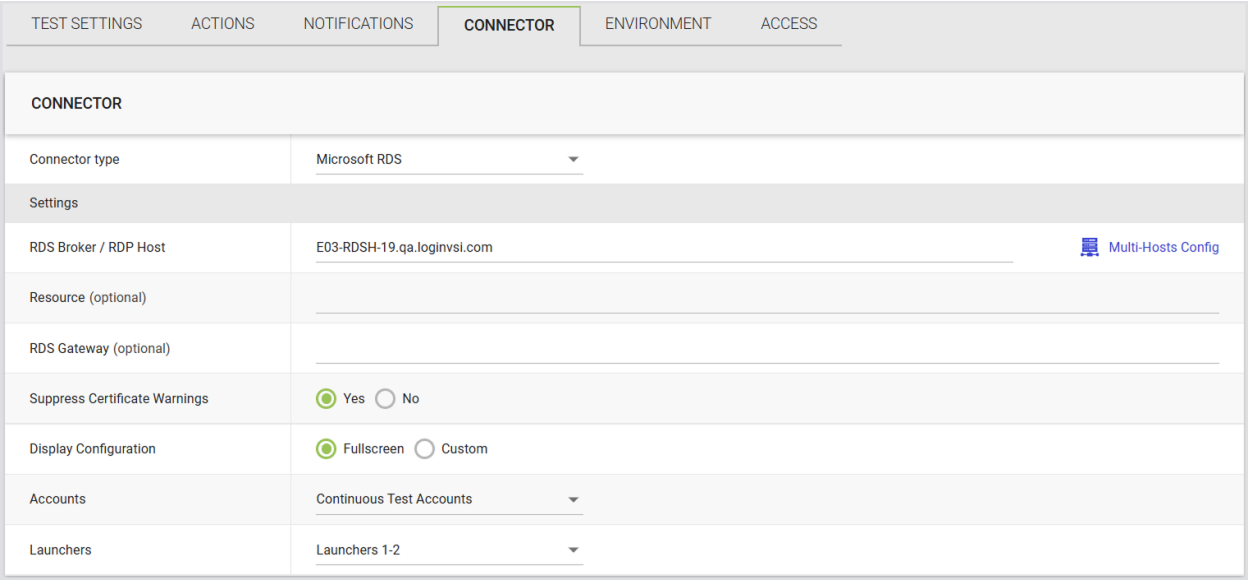
You can configure Applications in the Actions list. It is recommended to set up Application scripts that correspond with the published apps. While you can add other Application scripts, they may not function correctly if they are not intended for the published Applications defined for that user. Ensure that each Application script includes `"mainWindowTitle: *ApplicationTitle*"` in the START parameter.

NOTE: We support only one Continuous Testing environment with a single Published Application resource in the configuration.

How It Works

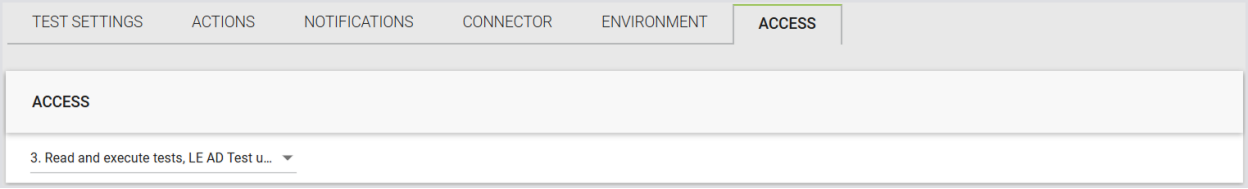
When the Launcher initiates the session, it receives information from StoreFront or NetScaler indicating that the configured resource is a Published App. This information instructs the Virtual Appliance script executor not to start the Application as it would in a virtual machine (VM) or desktop environment. As a result, you can use existing scripts for published apps, such as the default Microsoft Word scripts.

NOTE: Since the start command is skipped for Published Apps, the "App start time" value is not collected, as it no longer applies.



Access

You can assign roles to Tests to control access to the Test configuration and results. By assigning roles, only users with the designated roles will be able to view or modify the Test configuration and results, depending on their permissions. For detailed instructions on creating and managing roles, permissions, and dependencies, see the Role-Based Access Control.



Adjusting Engine start timeout

NOTE: This setting is not available through the Login Enterprise UI. To adjust it, you need to use the API.

The timeout for the Engine-start setting controls how long the Virtual Appliance will wait for the Engine on the Target to come online after a session is launched, and before the connection is considered a failure (login failure). Since we cannot see what is happening during the login phase, all we can do is wait for the Engine to communicate with the Appliance. Or give up and abort the session.

The default timeout is 300 seconds (5 minutes).

You can modify this timeout on an individual Test by using the Public API to reconfigure the Test. A PowerShell script is available to make this process easier. For specific directions on

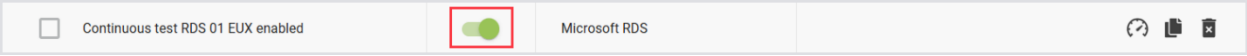
setting the appropriate parameter for an individual Test, see [Setting the Engine Start Timeout on a Test](#).

Key considerations:

- **Default timeout**: If no value is specified, or the value is set to 0, the timeout defaults to 300 seconds (5 minutes).
- **Web interface limitation**: This option is not available through the Appliance web interface. If you create a Test via the web interface, the timeout will be left empty and default to 300 seconds.
- **Update via API**: To modify the timeout, first create a Test, then use the update endpoint to set the timeout. The timeout cannot be set during Test creation.

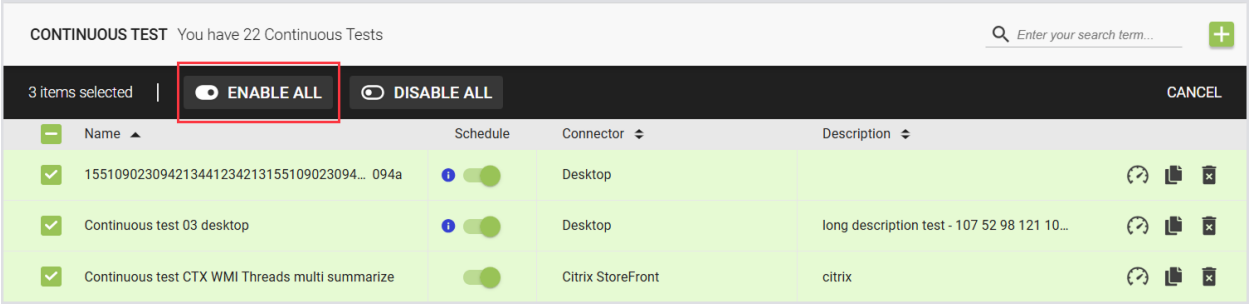
Starting a Continuous Test

In the **Continuous Tests**, switch the toggle next to the Test you'd like to start.



Enabling Continuous Tests in bulk

If you have multiple Continuous Test runs, you can enable or disable them in bulk. This way, you'll provide increased ease of management for larger Login Enterprise deployments. To perform bulk start/stop operations, select the checkboxes of all Tests that should be considered, and click **Enable all**.



Locations

If Launchers are attached to configured locations when a Continuous Test type is in progress, these Launchers will be visible in the map view. For more details, see [Locations](#).

Best Practices

Continuous Tests provide flexible scheduling options that allow administrators to automate test execution and monitor environments over time. The following tips help ensure reliable scheduling and avoid common issues.

Choosing the Right Connector for Scheduling

Scheduling Continuous Tests requires a Connector capable of initiating user sessions.

The Desktop Connector cannot initiate sessions, so it cannot run scheduled tests based on specific times or intervals. If you need to schedule tests automatically, configure a real Connector that supports user logon.

Using Scheduling Options to Match Testing Needs

Continuous Tests can be configured in several ways:

- Run continuously
- Run during specific hours of the day
- Run on specific days of the week
- Run at fixed intervals
- Run with a defined number of concurrent test instances

These options allow you to simulate ongoing workloads or periodic validation checks depending on your testing goals.

Time Zone Considerations

When scheduling a test, the appliance converts the configured time from your browser to Coordinated Universal Time (UTC) based on your current time zone.

However, the system does not store the original time zone or automatically adjust for daylight saving time (DST).

If your test schedule aligns closely with working hours and your region observes DST, you may need to manually adjust the scheduled time after the seasonal change.

Using Continuous Tests for Automation

Although the appliance is primarily designed as a measurement platform, Continuous Tests can also be used for other automation tasks, such as scheduling interactive actions or operational checks.

Because of their flexibility, Continuous Tests can function similarly to a task scheduler for executing scripted workflows.

Manually Triggering Tests with Desktop Connector

The Desktop Connector provides an alternative method for running tests manually.

You can trigger a Test Run directly from a machine by launching the connector `.exe` file with the appropriate parameters. This approach is useful when you need manual control over when tests start.

NOTE: Desktop Connector is not recommended for long-term production use when triggering Continuous Tests. After a Continuous Test iteration completes, the session remains logged in.

Continuous Test results and comparison

- For information on the Continuous Test results, see Viewing Continuous Testing Results.
- For general guidelines on how to analyze Continuous Testing Session Metrics, see Viewing Session Metrics.

Configuring Load Testing

- Overview
- How it works
- Creating a Load Test
- Configuring the Load Test
  - Test Settings
    - Settings
    - Session Metrics
  - Actions
  - Access
- Adjusting the Engine start timeout
- Starting a Load Test
  - Starting the Load Test from the Test config
- EUX score and VSImax
- Load Test results and comparison

Overview

Load Tests are designed to simulate multiple active users all running concurrently, to determine the point at which an additional user would impact existing users noticeably. Login Enterprise will launch multiple sessions on Target machines in parallel, ramping up to a specific number of sessions. Each of those sessions will run the defined Application scripts in a repeating loop for the duration of the Test. By default, it includes the EUX data collection phase and tries to calculate a VSImax number for your Test configuration.

Load Tests are not schedulable. They must be manually started, either through the **Login Enterprise sidebar menu > Manage Tests > Load Tests** or by making an API call. However, they require a real Connector and do not support the manual Desktop Connector. Load Tests need to be in control of session initiation to control and measure multiple sessions properly.

How it works

A Load Test performs the following sequence of actions:

- Ramp up sessions based on your total number of desired logins and the number of minutes you allot to launch them. For instance, 5 users in 8 minutes would launch a new user every 2 minutes (because you always launch one at 0 minutes).
- Once all the sessions have been launched, start counting towards the Test Duration configured in the Test. (If a session fails, it is not replaced).
- Within each session, repeat the Application script test sequence in a continuous loop until the Test duration has been met.
- If configured, periodically run our EUX applications to generate and measure an approximation of what a user will experience. Also, capture any Application script timers and other session metrics.

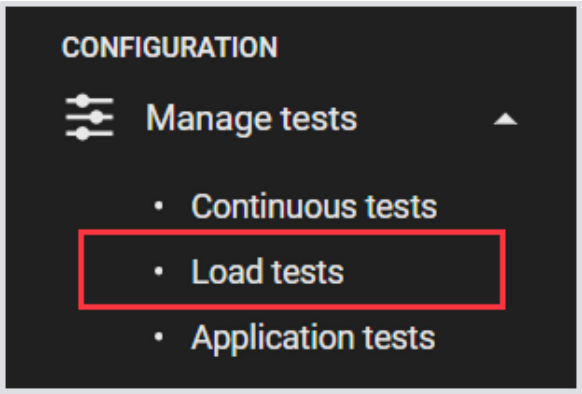
- When the overall Test Duration has been met, signal all the sessions to terminate. They will terminate after the current Application script.

When you look at a Load Test, you will see a chart that shows your login sessions ramping up, and how your recorded metrics behave as your number of sessions increases. What you're looking for are points in the curve where your metrics start to drop off as your sessions ramp up. For more information, see Viewing Load Testing Results.

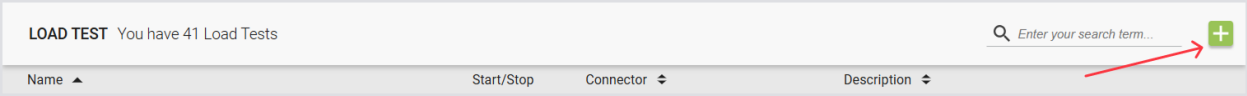
NOTE: Login Enterprise doesn't enable alert thresholds or emails for Load Tests, since these types of Tests are intended to be run manually, not as a Continuous background Test.

Creating a Load Test

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Manage Tests**, and select **Load Tests**.



2. In **Load Tests**, click "+" on the top-right.



3. Define the connection settings:

General	
Name	The display name for the Test.
Description (optional)	An informative description of the Test.
Environments (optional)	
Name	The target environment where the Test will be conducted. For more information, see Environments.

Connector	
Connector type	The type of Connector to be utilized by the Launcher. **NOTE**: Each Connector has its own settings. Once you define your Connector, you need to specify the necessary information in the Settings section. For details, see Connectors.
Access	
Roles	Select roles to control access to Test configurations and results. For more information, see Role-Based Access Control (RBAC).

4. Click **Save** to apply the changes.

NOTE: When you delete a Load Test, only the Test definition itself is removed. Your existing historical Load Test runs and their associated data will continue to be visible on your Results page. However, if you delete a Load Test run, you will remove all associated data, including measurements, screenshots, logs, and events.

Configuring Load Test settings

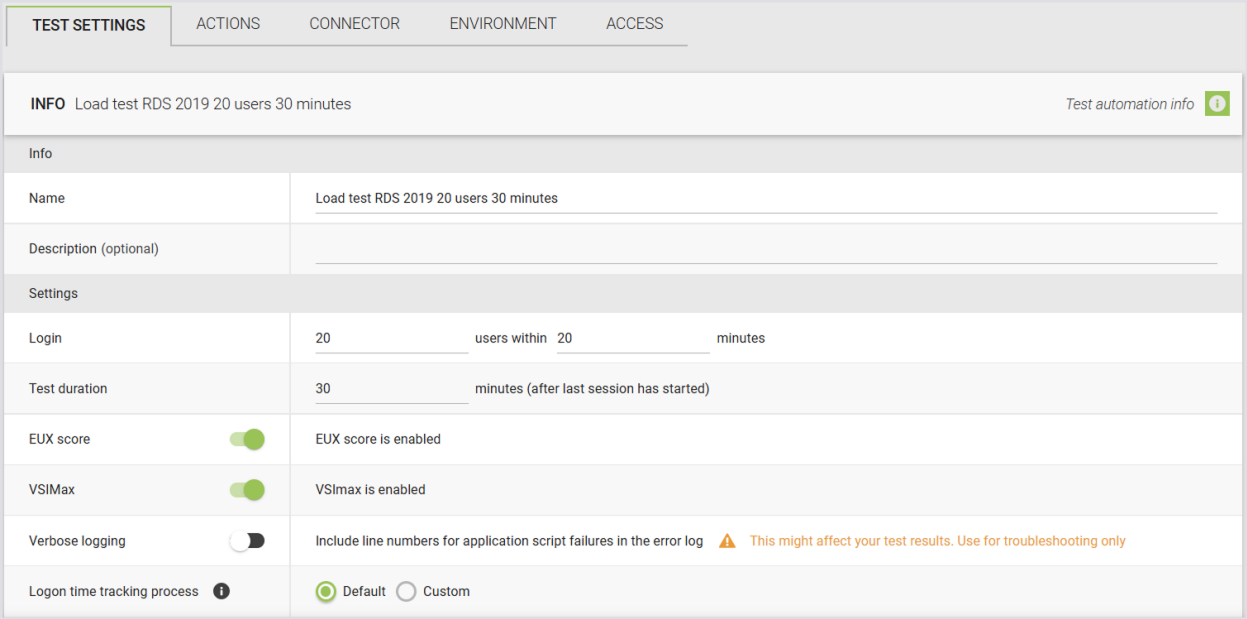
Test Settings

Once you've created the Test, you can configure Settings and Session Metrics.

Settings

Settings

Login X users within Y minutes	The number of users you want to be logging in and how long they should take. • The logon window is user-defined. We will evenly space your desired sessions across the number of minutes you specify. Failed sessions won't be retried and won't affect the speed at which sessions log on. That is, the configured logon rate of sessions will remain the same for the logon window regardless of logon failures. • Logoff timing cannot be configured. Login Enterprise will log off sessions at the rate of 10 per minute, with a maximum of 15 minutes. So, for example, if you have a Test with 140 sessions, the logoff window will take 14 minutes. Anything above 150 sessions will still take 15 minutes.
Test duration	How long the Test takes in minutes (after the last session starts). After the Test duration ends, all Engines will stop executing the script and log off.
EUX score / VSImax	Control how Login Enterprise measures performance responsiveness and scalability. For details, see the EUX score and VSImax.
Verbose logging	Display the command line number that triggered an error during application execution failures. **NOTE**: This may affect Test results as the line number is returned during execution.
Logon time tracking process	• Track the complete duration from when the Connector initiates the login process for the user session until the final user-session executable (usually Windows Explorer) is up and running. • You can also overwrite the default process of time tracking with a process of your choice. For more information, see Measuring total login time.

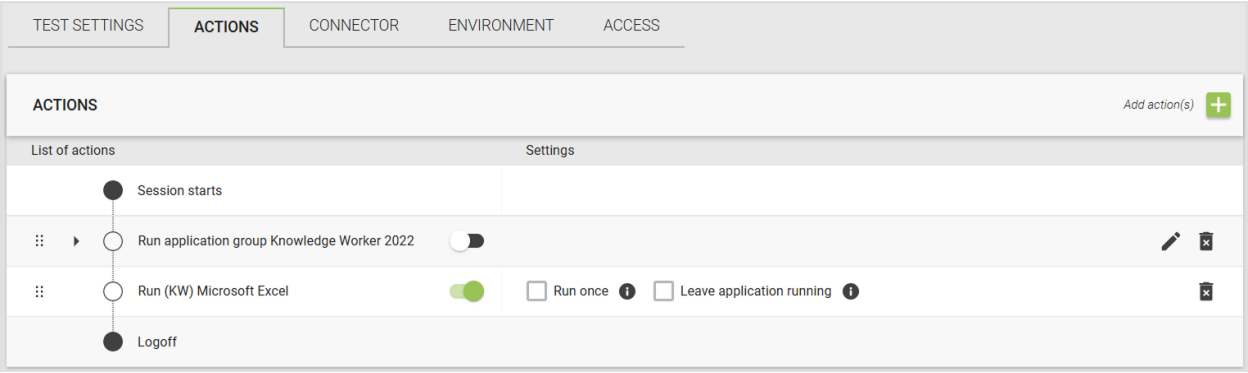


Session Metrics

Session Metrics	
Session Metrics	Enable or disable the gathering of Session Metrics. System performance metrics from each virtual user session will be collected periodically. The metrics collected are determined by the metric group you select below. For more information on Session Metrics, see Session Metrics.
Metrics group	Choose the desired Session Metrics Group to specify which performance counters should be collected.
Schedule (in which session)	Specify the schedule for sessions. If you don't want to collect Session Metrics in every session, specify how often it can run, e.g., 1 out of every X sessions.

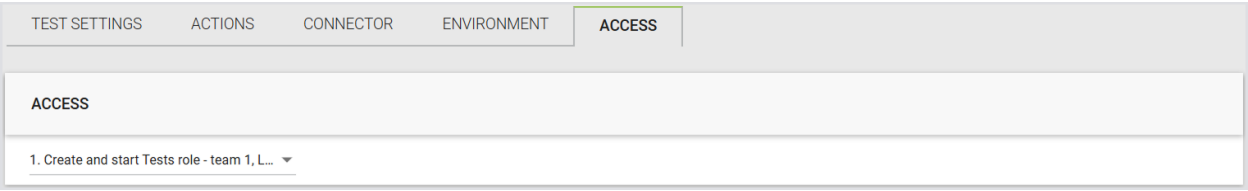
Actions

You can add the preferred Applications for the Virtual User to test. You can also add Application groups and an option “to wait”. For details, see Applications.



Access

You can assign roles to Tests to control access to the Test configuration and results. By assigning roles, only users with the designated roles will be able to view or modify the Test configuration and results, depending on their permissions. For detailed instructions on creating and managing roles, permissions, and dependencies, see the Role-Based Access Control.

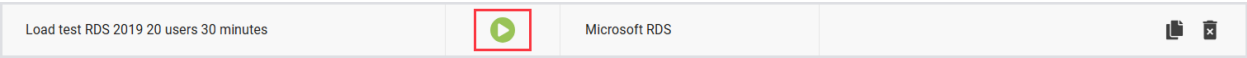


Adjusting the Engine start timeout

This setting works in the same way for all Test types. For details, see Adjusting the Engine start timeout.

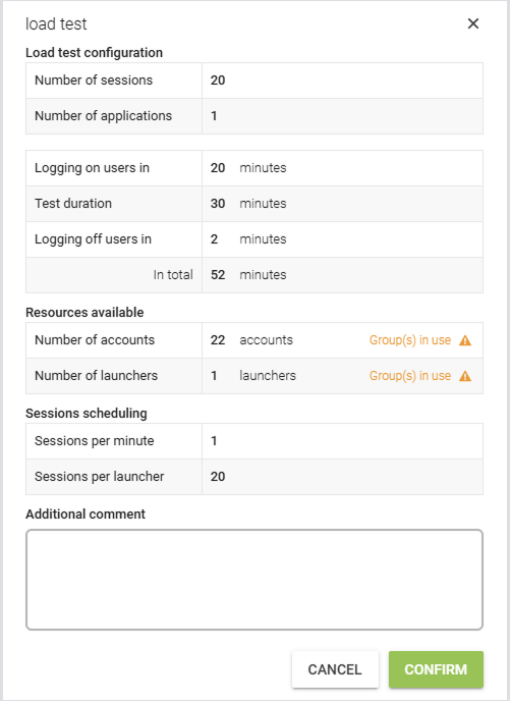
Starting a Load Test

1. In the **Load Test**, click "play" next to the Test you’d like to start.



When you start the Load Test, Login Enterprise will attempt to reserve the necessary accounts. It will also verify the availability of the selected Launchers. Accounts cannot be shared, but Launchers always will be, if multiple Tests are set to use them. Accounts will be exclusively allocated for use by Load Testing during its execution.

2. After you click **Play**, a confirmation dialog window will appear, displaying the current Application configuration. If the configuration is correct, click **Confirm** to begin the test.

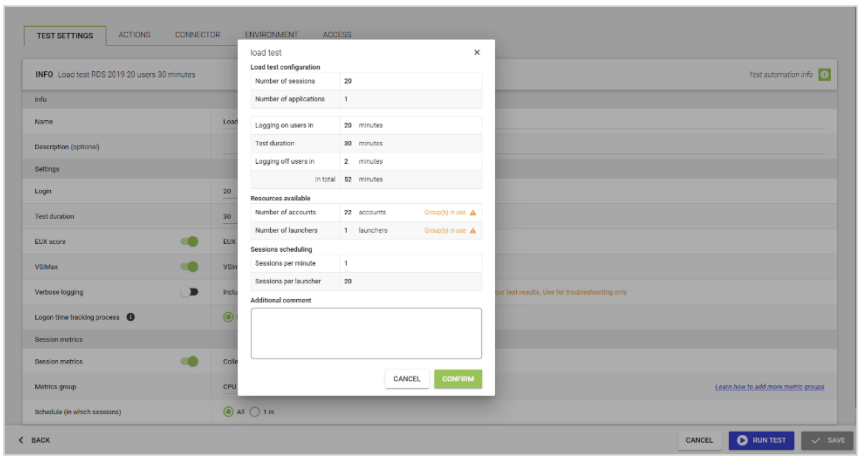


3. Once you click **Confirm**, you will be redirected to the Test Dashboard. Here, you can see the Load Test and its progression. To view the early results of your Load Test, in the Test Dashboard, click on the Load Test that you started. For more information, see the Test Dashboard.

Starting the Load Test from the Test Config page

You can run a Load Test directly from the Test configuration page. This allows you to run the Test immediately after reviewing or editing its settings, without returning to the Overview page. If there are unsaved changes, you'll be prompted to save before the test starts.

NOTE: You can only run one Load Test at a time; if another Test is already running, the Start button will be disabled.



EUX score and VSImax

Dependency	Behavior	Calculation
EUX score represents a responsiveness score (0–10) that reflects perceived system performance.	Controlled by its own toggle in *Create/Edit Load Test* settings (enabled by default).	Calculated directly from system responsiveness metrics.
VSImax estimates the maximum number of concurrent sessions a resource can support before responsiveness noticeably degrades.	• Controlled by a separate toggle (enabled by default). • Depends on the EUX score. ○ When the EUX score is disabled, VSImax becomes disabled and grayed out. • Turning VSImax off displays "–" in Test result views (single test, multi-test comparisons, PDF reports). • Changes affect only new Test runs; past runs remain unchanged. • No run-level override; toggle available only in Test settings. • Existing Tests migrate with VSImax = On (if EUX score is enabled).	Defined as the point where the EUX score drops below 5.5, indicating responsiveness degradation.

To learn more about the EUX Score and VSImax, see [Calculating the EUX score and VSImax](#).

Load Test results and comparison

- For information on the Load Test results and comparison, see Viewing Load Testing Results.
- For general guidelines on how to analyze Load Testing Session Metrics, see Viewing Session Metrics.

Configuring Application Testing

- Overview
- How it works
- Creating an Application Test
- Configuring the Application Test
  - Test Settings
    - Settings
    - Report settings
  - Actions
  - Thresholds
  - Access
- Adjusting the Engine start timeout
- Starting an Application Test
  - Starting the Application Test from the Test config
- Best practices: handling partial failures in scripts
- Application Test results

Overview

Application Tests verify that your user software functions correctly and remains within established operational parameters. Similar to Load Tests, these are one-time, on-demand tests that return data and events specific to the current test. They are not scheduled to run periodically.

How it works

An Application Test launches a single session. It only requires one session because it does not gather long-term tracking data, unlike Continuous Tests, and it does not measure aggregate performance, unlike load tests. Its sole purpose is to select a single virtual desktop infrastructure (VDI) and run a set of Application scripts to verify that your software is functional. EUX testing cannot be enabled for these tests.

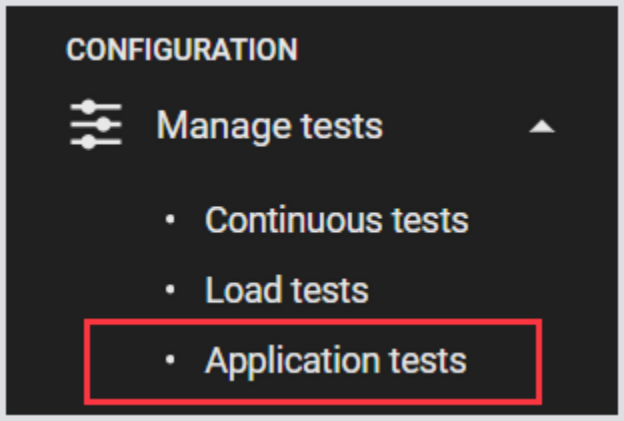
Application scripts for Application Tests focus on functional areas rather than metrics. While a Continuous Test might benefit from timing every operation with measurable durations, an Application Test aims to verify that each part of the program functions correctly. The scripts should check various aspects of the program, aborting if any errors occur. If timers are used, they are intended to identify significant issues, such as tasks that never complete, rather than small changes in speed.

Unlike Load Tests, Application Tests can trigger alert e-mails for threshold violations.

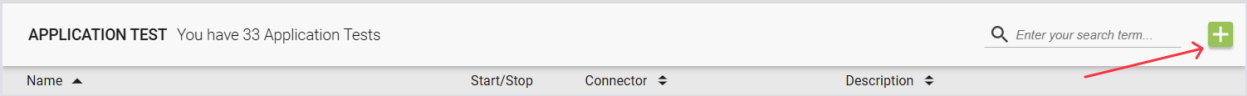
NOTE: EUX and VSImax, designed as aggregate statistical measures, are unavailable in Application Tests. However, as Application Tests utilize only one session, the Desktop Connector remains accessible. For more information, see Desktop connection.

Creating an Application Test

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Manage Tests**, and select **Application Tests**.



2. In **Application Tests**, click "+" on the top-right.



3. Define the connection settings:

General	
Name	The display name for the Application Test.
Description (optional)	An informative description of the Application Test.
Connector	
Connector type	The type of Connector to be utilized by the Launcher. **NOTE**: Each Connector has its own settings. Once you define your Connector, you need to specify the necessary information in the Settings section. For details, see Connectors.
Access	
Roles	Select roles to control access to Test configurations and results. For more information, see Role-Based Access Control (RBAC).

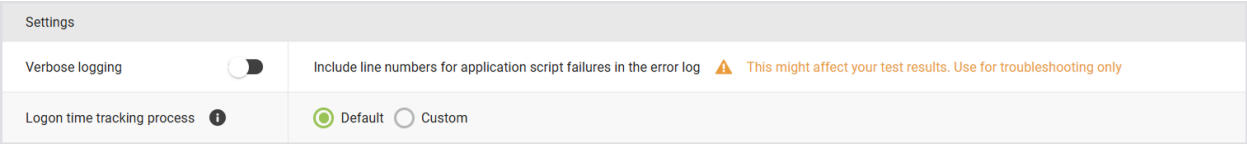
Configuring the Application Test

Test Settings

Once you've created the Test, you can configure Settings and Report Settings:

Settings

Settings	
Verbose logging	Display the command line number that triggered an error during Application execution failures. **NOTE**: This may affect test results as the line number is returned during execution.
Logon time tracking process	• Track the complete duration from when the Connector initiates the login process for the user session until the final user-session executable (usually Windows Explorer) is up and running. • You can also overwrite the default process of time tracking with a process of your choice. For details, see Measuring total login time.



Report Settings

Report settings

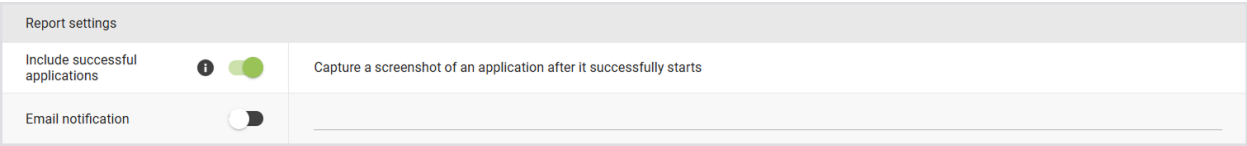
Include successful applications	• Choose to capture a screenshot of successful applications when the script starts the program without errors. • Disabling this option skips the screenshot, but the results of the successful applications (such as measurements) will still be included in the report. • This option only produces screenshots if your script uses the built-in START() command to start your program. Otherwise, you can use the TakeScreenshot() command in your script after you start the program.
Email notification	• Choose to be notified when Applications pass or fail. • If you do not enable notifications, the report will only indicate failures, which is usually the most relevant information.

You can also configure the system to send an email once the Application Test is complete. If the email notification switch is enabled but no email address is specified, the default email address from the email configuration will be used. You can enter multiple email addresses in the Email Notification field, separated by commas. For more information, see Managing Notifications.

TIP: You can personalize the report by adding your company logo.

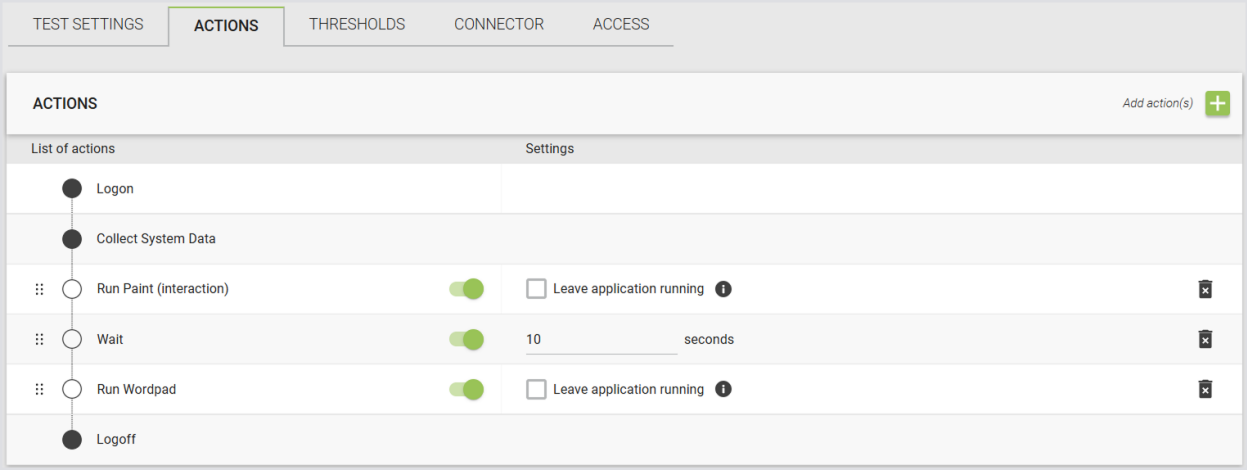
NOTE: There is no strict limit on the number of screenshots in a generated PDF report. However, note that including a large number of screenshots (for example, up to 1,000 screenshots) may cause browser downtime and increase the PDF file size.

For optimal performance and compatibility, we recommend keeping the number of screenshots below 250. At around 250 screenshots, the PDF size is typically under 20 MB, which aligns with email systems that limit attachments to 20 MB.



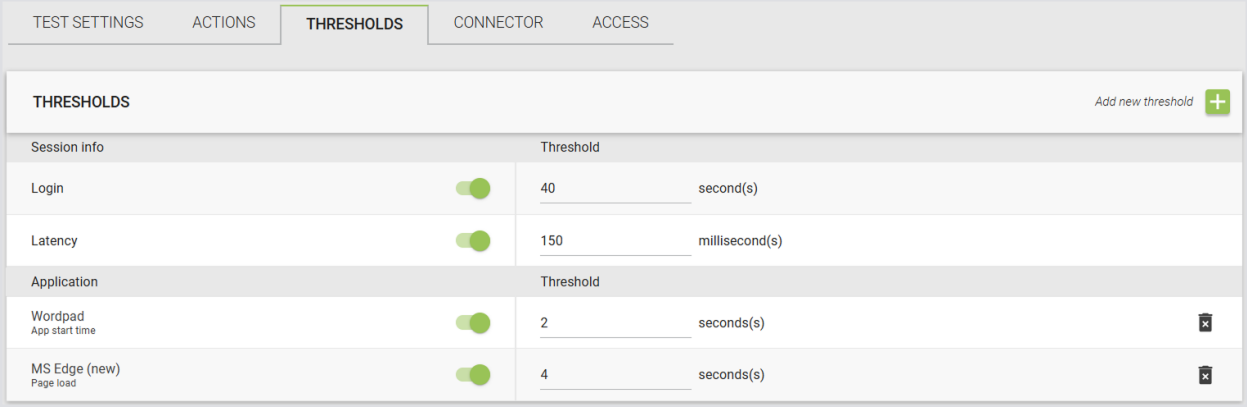
Actions

You can add the preferred Applications for the Virtual User to test. You can also add Application Groups and “Waits.” For details, see Applications.



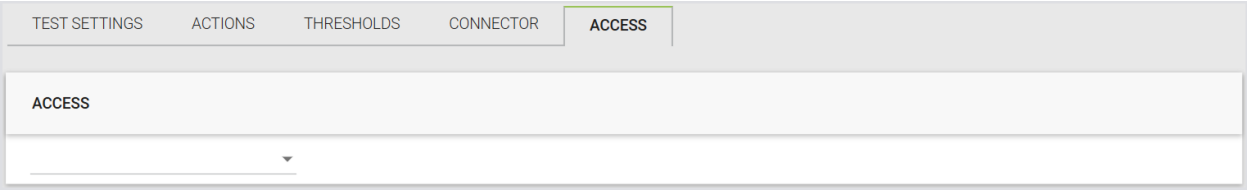
Thresholds

You can also define thresholds for all actions, similar to the Continuous Testing setup. These thresholds ensure that the Application not only functions correctly but also maintains its performance across different configurations. If performance thresholds are not met, you will be notified in the Application Testing report.



Access

You can assign roles to Tests to control access to the Test configuration and results. By assigning roles, only users with the designated roles will be able to view or modify the Test configuration and results, depending on their permissions. For detailed instructions on creating and managing roles, permissions, and dependencies, see the Role-Based Access Control.

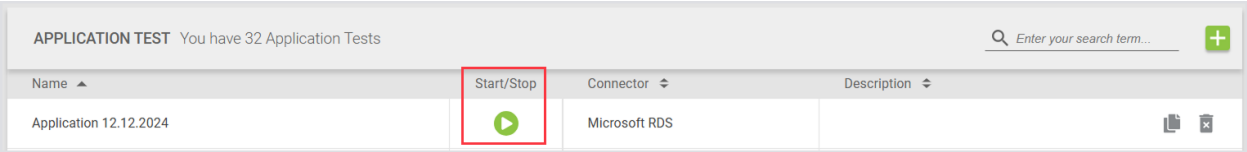


Adjusting Engine start timeout

This setting works in the same way for all Test types. For details, see [Adjusting the Engine start timeout](#).

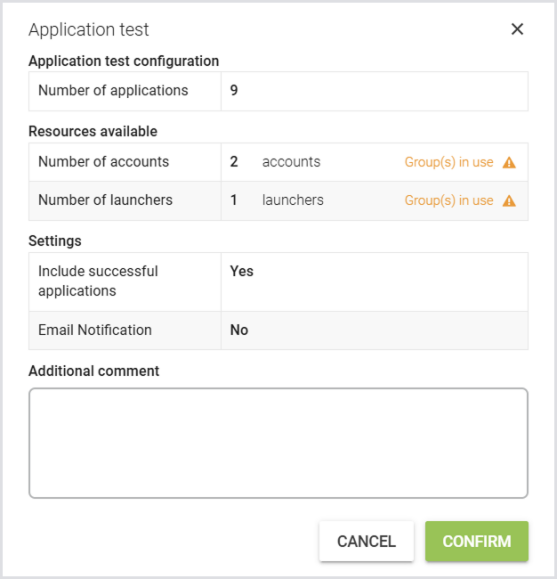
Starting an Application Test

1. In the **Application Tests**, click the **Play** button next to the Test you'd like to start.



NOTE: Even if you configured a user group that contains multiple users, the Application Test scenario only uses one user.

2. After you click **Play**, a confirmation dialog window will appear, displaying the current Application configuration. If the configuration is correct, click **Confirm** to begin the test.

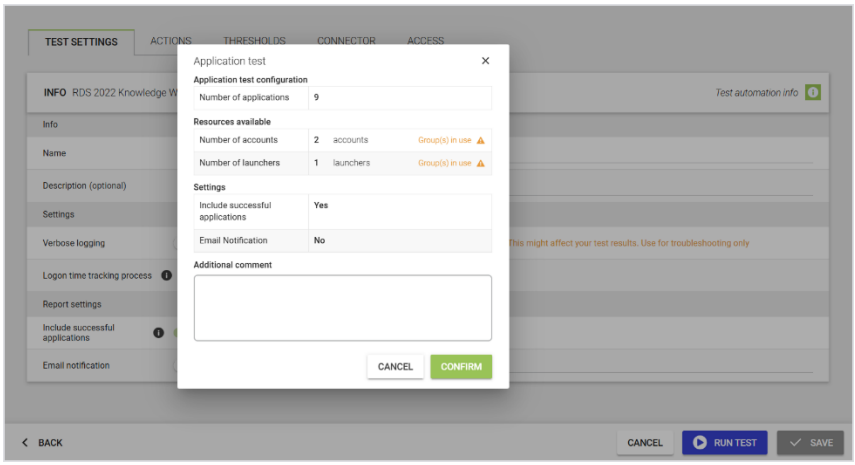


3. Once you click **Confirm**, you will be redirected to the Test Dashboard. Here, you can view the current status of your recently started Application Test. The dashboard displays how many Applications have started and if any have failed. Performance measurements are also shown. For more information, see the [Test Dashboard](#).

Starting the Application Test from the Test Config

You can run an Application Test directly from the Test configuration page. This allows you to run the Test immediately after reviewing or editing its settings, without returning to the Overview page.

Unlike Load Tests, multiple Application Tests can run at the same time. The **Run Test** button remains available even if other Application Tests are currently running. If there are unsaved changes, clicking **Run Test** will trigger a confirmation pop-up:



TIP: Click **Save and Run** to save your changes and start the Test. Click **Cancel** to return to the page without starting the Test or saving your changes.

For Desktop Connector Tests, clicking **Run Test** opens the same pop-up used on the Overview page.

Best Practices: Handling Partial Failures in Scripts

Script failures are treated as an all-or-nothing outcome. If any function fails, the entire script is marked as failed. This can make it difficult to evaluate multiple functions within a single test. If you need more granular results and want the script to continue running even after a subtask fails, consider the following approaches.

Log Subtask Failures with Events

Use `CreateEvent` in your script to log when a specific subtask fails. This allows the script to:

- Record the failure in the Login Enterprise events for the Test Run.
- Continue executing the remaining subtasks.

This approach is useful when you want to capture multiple issues in a single Test Run instead of stopping at the first failure.

Use Timers to Represent Functional Results

Timers can also be used to report success or failure conditions. Instead of measuring a duration with `StartTimer` and `StopTimer`, use `SetTimer` to report a predefined value.

Example approach:

- **Success value:** `1000` (represents 1 second)
- **Failure value:** `2000`

Configure an alert threshold (for example, **1500**) so that:

- Values below the threshold indicate success
- Values above the threshold indicate failure

Using clearly distinguishable values (such as `1000` and `2000`) makes the script easier to read and maintain compared to smaller values like `1` and `2`.

Application Test results

For information on the Application Test results, see Viewing Application Testing Results.

Connectors

Login Enterprise connectors are used for remotely initiating sessions to published applications and desktops across a variety of technologies, including Citrix Virtual Apps and Desktops (CVAD), Citrix Cloud, Microsoft Remote Desktop Services, Microsoft Azure Virtual Desktop, Microsoft Windows 365, and VMware Horizon.

Connectors are selected and configured as part of a Test configuration and come in 3 distinct types: Standard Connectors, Desktop Connectors, and Custom Connectors. Login Enterprise supports the following Connector types:

- Citrix Netscaler 12.1 and 13.0
- Citrix Storefront
- Microsoft RDS
- VMware Horizon View
- Windows 365 (Preview)
- Desktop Connector
- Custom Connector

Netscaler, Storefront, RDS, and Horizon connectors will actively authenticate as the designated user account, select a resource, and launch the remote session client to establish your connection.

The Desktop Connector allows you to manually start a test from an existing login session, instead of having the Launcher establish the session for you.

The Custom Connector allows the Launcher to call an external program to perform the connection, rather than using a built-in Connector. You can call executables or scripts, and the Launcher will place fields on the command line as you direct it to. Username, domain, password, remote server URL, resource name, and so on can all be added as command-line parameters. We can provide some custom connector programs, like our AVD connector and our Universal Web Connector, but you can call any external program, including building your own automation.

Review the parameters you must specify to configure a connector for your connection server, e.g., “username”, “domain”, “password”, “broker URL”, and “resource name”. The parameters supply the details of how to interact with the target environment. These required parameters will be the same for a Connector regardless of Test type, so once you have configured a particular Connector for one type of Test, the configuration settings can be used for any other Test or Test type.

Benefits of Using Connectors

Benefit	Description

Seamless integration	Login Enterprise offers pre-configured Connectors for leading platforms, such as Citrix Netscaler 12.1, Citrix Netscaler 13.0, Citrix Storefront, Microsoft RDS, and VMware Horizon View. This ensures seamless integration, simplifying remote access setup.
Customization options	Organizations can tailor configurations with the Custom Connector, adapting settings to specific needs while maintaining simplicity.
Centralized management	Administrators can efficiently manage and monitor Connectors from a unified interface, simplifying remote access management and troubleshooting processes.
Scalability and compatibility	With support for diverse Connectors, Login Enterprise scales easily to accommodate changing IT environments while ensuring compatibility with evolving technology landscapes.
User-like operation	Connectors simulate an actual user connecting to the environment, so the availability and connection performance to the target digital workspace will be indicated in the connection pass/fail and timing results.

Citrix Netscaler 12.1 and 13.0

- Overview
- Configuring the Citrix Netscaler 12.1 and 13.0 Connector

Overview

For the Netscaler Connector, we interact with the web interface to identify and select the desired resource from the supplied URL. That will result in an ICA file, which we then pass to Citrix\ICA Client\wfcrun32.exe to initiate the actual remote session.

The Citrix NetScaler connector exclusively supports nFactor authentication and is currently compatible only with forms-based authentication factors, such as LDAP. For configuring nFactor and understanding the requirements, refer to the following links:

- nFactor authentication with NetScaler Unified Gateway
- nFactor Authentication – NetScaler Gateway 12 / Citrix Gateway 12.1
- nFactor Authentication – Citrix Gateway 13

Configuring the Citrix Netscaler 12.1 and 13.0 Connector

In Login Enterprise, the Citrix Netscaler connector supports the following parameters:

Field	Description
Netscaler URL	The Netscaler URL.
Resource	The resource name visible to end users when they log in to StoreFront.
Display Configuration	Full-screen display or custom resolution settings.
Seamless Mode	When enabled, Citrix published apps open as seamless windows instead of in a remote desktop window. For details, see Seamless Mode for Citrix Published Apps.
Accounts	Select previously configured Account Groups or all Accounts.
Launchers	Choose from previously configured Launcher Groups or all Launchers.

TEST SETTINGS | ACTIONS | **CONNECTOR** | ENVIRONMENT | ACCESS

CONNECTOR

Connector type	Citrix Netscaler 12.1 and 13.0
Settings	
Netscaler Url	https://e03-xd7-ns.qa.loginvsi.com
Resource	SBC Desktop
Display Configuration	Fullscreen / Custom Width: 1680 px Height: 1050 px
Seamless Mode	Seamless mode requires Launcher version 6.2 or later. If an older Launcher is used, this setting will be ignored.
Accounts	Load Test Accounts
Launchers	Launchers 1-2

Citrix Storefront

- Overview
- Configuring the Citrix Storefront Connector
- Seamless Mode for Citrix Published Apps
  - Benefits of Seamless Mode
  - How to Enable Seamless Mode
  - Behavior and Compatibility
  - Best Practices

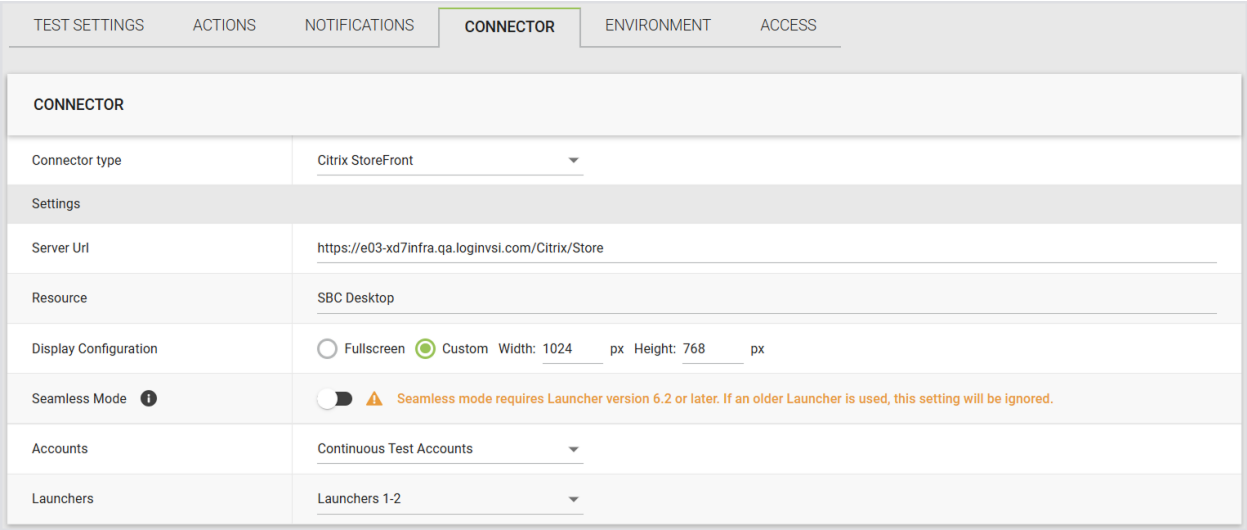
Overview

For the Storefront Connector, we interact with the web interface to identify and select the desired resource from the supplied URL. That will result in an ICA file, which we then pass to Citrix\ICA Client\wfcrun32.exe to initiate the actual remote session.

Configuring the Citrix Storefront Connector

In Login Enterprise, the Citrix Storefront connector supports the following parameters:

Field	Description
Server URL	The Store URL (not the StoreWeb URL).
Resource	The resource name visible to end users when they log in to StoreFront.
Display Configuration	Full-screen display or custom resolution settings.
Seamless Mode	When enabled, Citrix published apps open as seamless windows instead of in a remote desktop window. For details, see Seamless Mode for Citrix Published Apps.
Accounts	Select from previously configured Account Groups or all Accounts.
Launchers	Choose from previously configured Launcher Groups or all Launchers.



Seamless Mode for Citrix Published Apps

Seamless Mode supports published applications delivered through Citrix virtualization platforms. This mode allows for more accurate performance testing that aligns with Citrix's best practices. When enabled, it launches apps in seamless mode instead of the Login Enterprise default windowed mode (`TWIMode=Off`).

This configuration is especially useful for organizations running newer Citrix LTSR versions where ShellBridge or other seamless mode mechanisms affect application behavior and performance. Running in windowed mode may no longer reflect the typical user experience in these environments.

IMPORTANT: Seamless Mode is only available when using Launcher version 6.2 or later. Ensure your Launcher is up to date before enabling this setting.

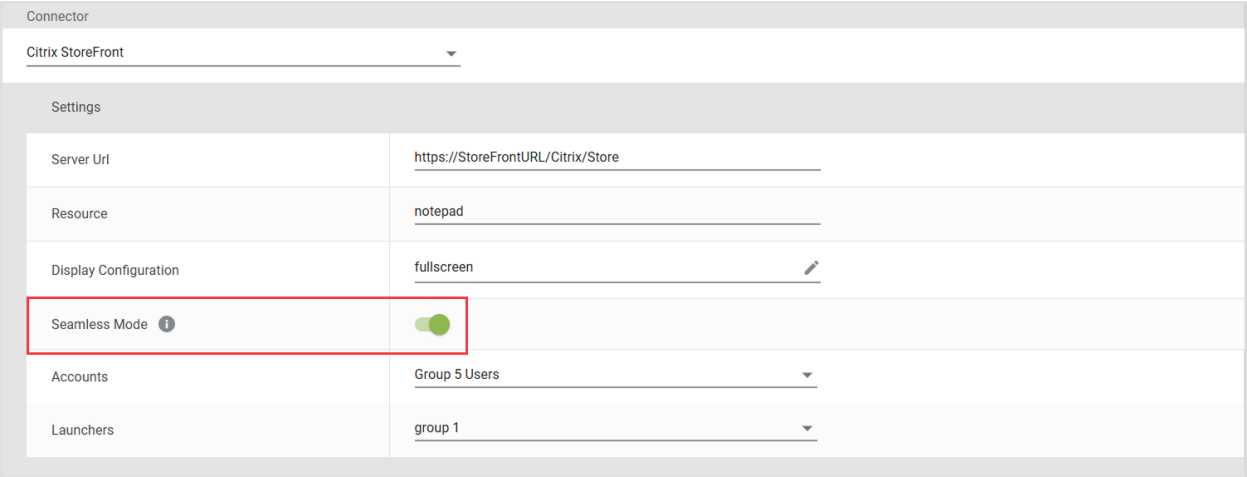
If the Launcher version is not supported, this option will be ignored, and the Test will run in windowed mode.

Benefits of Seamless Mode

- Reflects Citrix's recommended configuration for published apps.
- Improves the accuracy of performance metrics.
- Enhances the realism of Test scenarios for end-user experience.

How to Enable Seamless Mode

1. Open the connector configuration for StoreFront and/or NetScaler.
2. Locate the **Seamless Mode** toggle.
3. Enable **Seamless Mode** to allow published apps to launch outside a fixed window.
4. Save the configuration.



NOTE: If Seamless Mode is not enabled, Login Enterprise defaults to windowed mode (TWIMode=Off) for compatibility.

Behavior and Compatibility

- Enabling Seamless Mode allows multiple published app sessions to be tracked reliably using ProcessId (instead of relying on window titles).
- Session lifecycle behavior (start, monitor, terminate) is consistent in both seamless and windowed modes.
- If Seamless Mode is disabled, apps will behave as they did in previous versions.
- Upgrading to this version does not change existing connector behavior unless Seamless Mode is explicitly enabled.
- Enabling Seamless Mode may affect testing results, such as changes in the EUX score.
- Seamless Mode requires the app window to stay in focus. When multiple published app sessions are running simultaneously, they may compete for focus. In some cases, this can cause interference between sessions, and some applications may fail, depending on the application itself and how the script is written. You may need to restrict your test to have only one Seamless Mode session per Launcher.

Best Practices

- Use Seamless Mode when testing Citrix-published apps.
- Refer to in-product tooltips for guidance on when to enable Seamless Mode.
- Always confirm that your Launcher version supports this feature before use.

Microsoft RDS

- Overview
- Configuring the Microsoft RDS Connector
- Multi-Host Configurations
  - Adding Multi-Host Configurations

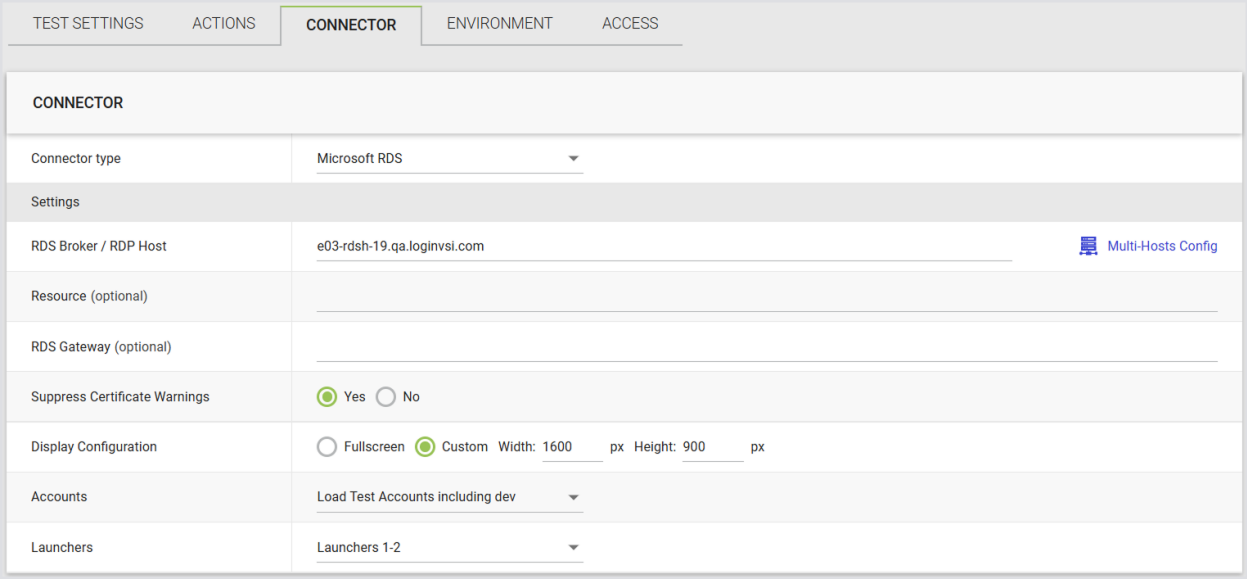
Overview

For the Microsoft Remote Desktop Session (RDS), we construct an RDS file that contains the target server, resource information, and authentication information, then we execute MSTSC.exe to launch it and monitor the resulting process.

Configuring the Microsoft RDS Connector

In Login Enterprise, the Microsoft RDS connector supports the following parameters:

Field	Description
RDS Broker / RDP Host	Allows connection to an RDS farm or a single desktop. For adding mult-host configurations, see Multi-Host Configurations.
Resource (optional)	Specify the name of the desktop pool when the RDS broker is configured.
RDS Gateway (optional)	Optionally configures RDS Gateway information if applicable.
Suppress Certificate Warnings	Enables ignoring certificate warnings for untrusted hosts.
Display Configuration	Select between full-screen or custom resolution settings.
Accounts	Choose from previously configured Account Groups or all Accounts.
Launchers	Select from previously configured Launcher Groups or all Launchers.



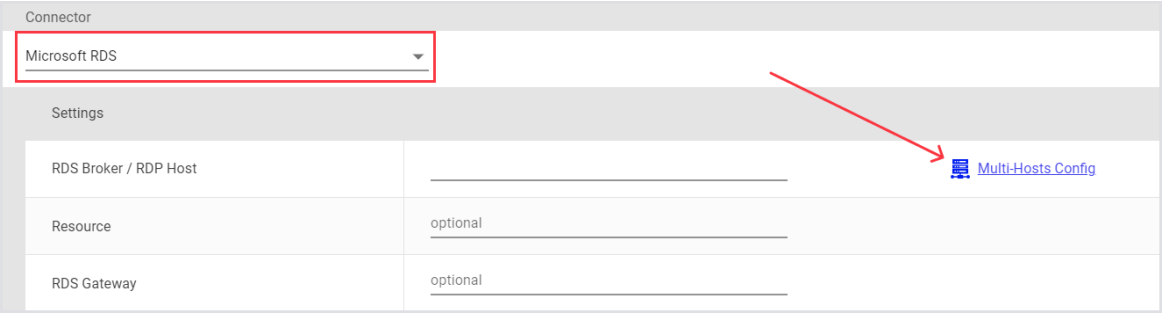
Multi-Host Configurations

Multi-host configurations in the Microsoft RDS are crucial for achieving various objectives such as high availability, load balancing, scalability, and fault tolerance. Unlike using the RDS Broker for load balancing, the multi-host configuration allows organizations to specify hosts or IP addresses directly, bypassing load balancing. This means that virtual users can be directed to specific hosts, such as host1, host2, and so on, without going through the Broker's load-balancing mechanism. When multiple hosts are configured, the Test will direct connections to them in a round-robin sequence, cycling through the entire list in order.

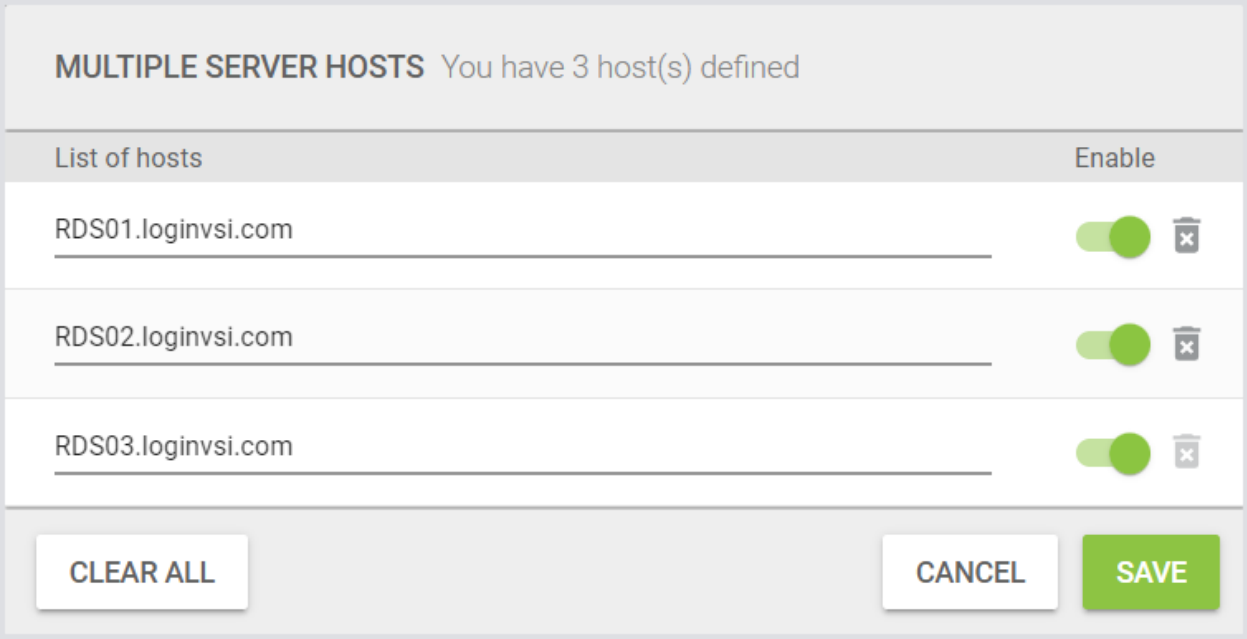
This approach offers flexibility, as you can choose to test only a subset of servers within the RDS Farm. For instance, if you want to test only a few servers (e.g., host1 to host5) instead of all servers on the farm, you can utilize the Multi-host option. By doing so, you can ensure that virtual users are directed only to the specified hosts, without undergoing load balancing by the Broker. This strategy helps optimize testing scenarios and ensures efficient resource utilization.

Adding Multi-Host Configurations

1. When creating or modifying your Microsoft RDS environment, click **Multi-Host Configs** on the right (The Multiple-server hosts window will open).



2. In the **Multiple server hosts** table, provide the details of the RDS servers you wish to have tested. You can also copy and paste a list of machines here for easy entry. Please note that we will connect to your machines in a round-robin style. That is, we will connect to machine 1 first, then proceed to machine 2, and so on in a sequential fashion.
3. Click **Save** to apply the changes.



VMware Horizon View

- Overview
- Configuring the VMware Horizon View Connector
  - Connection Command Line

Overview

For the VMware Horizon View connector, we call the installed vmware-view.exe executable with the appropriate resource, server, and authentication parameters, and monitor the resulting process. Below is a breakdown of the parameters used:

- **Resource**: Specifies the resource (virtual desktop or application) that the user wants to access.
- **Server**: Specifies the server address or hostname of the VMware Horizon View server.
- **Authentication Parameters**: Include credentials or authentication tokens required to authenticate the user and establish a secure connection.

If your VMware View client is not included in your System Path variable, you need to provide the explicit path. Typically, the path is as follows:

- **For 64-bit versions of VMware View client**: C:\Program Files\VMware\VMware Horizon View Client\vmware-view.exe
- **For 32-bit versions of VMware View client**: C:\Program Files (x86)\VMware\VMware Horizon View Client\vmware-view.exe

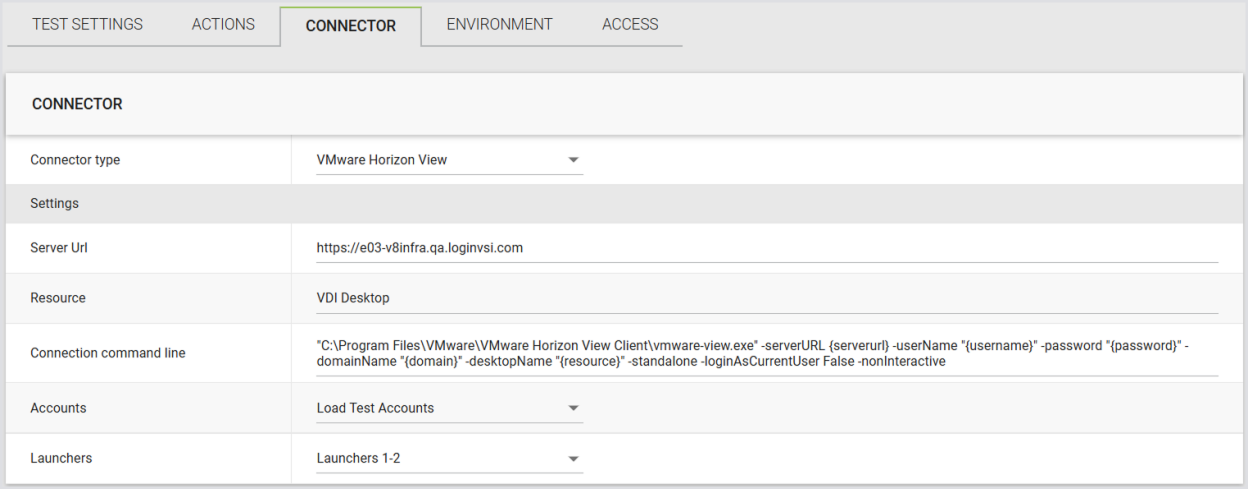
Ensure to verify the installation directory on your system and adjust the path accordingly when specifying it.

Configuring the VMware Horizon View Connector

In Login Enterprise, the VMware Horizon View connector supports the following parameters:

Field	Description
Server URL	The View Server for the connection.
Resource	The resource name visible to end users.
Connection command line	The path and parameters used by the VMware View executable. See Connection Command Line for more information. **NOTE:** A default command line is generated automatically.
Accounts	Select from previously configured Account Groups or all Accounts.

Launchers	Select from previously configured Launcher Groups or all Launchers.



Additionally, you can incorporate all accepted parameters for the Horizon View Client. For further details, see the VMware View Connector Reference.

Connection Command Line

The following variables are commonly used in the connection command line for the VMware Horizon View connector. The available connection command-line options are controlled by the VMware client itself. If you need to set additional options, see the Omnissa documentation.

Field	Description
serverurl	The URL address of the server to connect to
username	The username field from the selected account
password	The plain-text password from the selected account
domain	The domain field from the selected account
resource	The “Resource” field from the custom connector configuration
custom*X*	The Custom1 to Custom5 fields configured within the Account record
SecureCustom*X*	The Secure Custom1 to Secure Custom5 fields configured within the Account record

Connector

VMware Horizon View

Settings

Server Url	https://horizon.zandbak.com
Resource	Finance Desktop
Connection command line	vmware-view.exe -serverURL {serverurl} -userName "{username}" -password "{password}" -domainName "{domain}" -desktopName "{resource}" -standalone -loginAsCurrentUser False -nonInteractive
Accounts	All
Launchers	All

Windows 365 (Preview)

- Overview
- Windows 365 Script Version and Updates
- Windows 365 Connector Behavior
- Important Considerations
- How the Windows 365 Connector Works
- Prerequisites
- Preparing the Virtual User Account
- Configuring the Windows 365 Connector
- Running a Test with the Windows 365 Connector
- Expected Behavior
- Windows 365 Connector Script Management
  - In the UI
    - Uploading a Custom Script
    - Downloading the Script
  - Using the Public API
    - ConnectorScript Model
    - Creating a Script
    - Script File Requirements

Overview

The Windows 365 Connector enables automated testing of Windows 365 Cloud PCs by launching and interacting with a remote desktop session through the Windows App. The connector is typically used for Application and Continuous Tests, where a single user connects to a dedicated Cloud PC.

Windows 365 Script Version and Updates

Due to the nature of Windows 365 application and platform updates, the connector script may require updates. The latest version of the W365 script is maintained in the Login VSI GitHub repository. In the repository, you will find:

- Script versions
- Corresponding Windows App versions
- Tested script and app version pairings

Windows 365 Connector Behavior

The Windows 365 connector processes only one session at a time during the connection/login phase on a given Launcher. If multiple Windows 365 sessions are started on the same Launcher, their connection phases are queued and handled sequentially in FIFO (First-In, First-Out) order.

This limitation exists because Windows 365 automates the Windows App, which cannot be automated concurrently in multiple ways on the same machine.

The queuing behavior applies only to the connection phase. Once a session is fully connected, additional sessions may proceed. Sessions running on different Launchers can connect in parallel.

Important Considerations

The Windows 365 connection phase may take significantly longer than other connector types. Windows 365 sessions are exempt from the standard in-queue timeout and can safely wait for their turn.

However, if Windows 365 sessions share a Launcher with other connector types, non-Windows 365 sessions may time out and fail while Windows 365 is holding the queue (default in-queue timeout is 20 seconds). The most common scenario is:

"The Launcher did not respond to a session request in a timely fashion."

TIP: For optimal reliability, use dedicated Launchers for Windows 365 sessions.

How the Windows 365 Connector Works

1. Launches the Windows App and waits (up to the configured timeout) until the main **Windows App** window appears, then focuses it
2. Checks whether the app is already on the **Sign in** screen (i.e., the **Sign in** button is visible)
3. If it isn't on the **Sign in** screen (i.e., a user is likely already signed in or a dialog is open), it repeatedly attempts to reset the app back to a clean sign-in state until timeout by:
    1. Opening the Account menu (if present) and selecting **Sign out**
    2. Closing the account picker dialog (if it appears)
    3. Re-checking until the **Sign in** button is visible
4. Clicks the **Sign in** button in the Windows App
5. Closes any leftover Microsoft authentication windows that may already be open (to avoid conflicting sign-in dialogs)
6. If the account picker offers it, clicks **Use another account** (to force a fresh login rather than reusing a cached account)
7. Waits (up to the configured timeout) for the Microsoft sign-in window to appear
8. In the Microsoft sign-in window:
    - Enters the username/email
    - Clicks **Next**
    - Enters the password
    - Clicks **Sign in**
9. Switches MFA method (if the option is available) by selecting **Sign in another way**, then chooses **Use a verification code**.

10. Generates a TOTP verification code (from the configured TOTP secret/settings), enters it, and clicks **Verify**.
11. Waits for sign-in to complete and for the Windows App UI to reach the post-login state, where the **Devices** navigation is available. If a first-run/onboarding **Skip** prompt is shown, it clicks **Skip**.
12. Opens **Devices**.
13. Uses the **Search** field to search for the configured Cloud PC name/title.
14. Before connecting, it closes any Cloud PC authentication web prompts/windows that might already be open in the background (cleanup step).
15. Clicks **Connect to** in order to initiate the Cloud PC (Windows 365) connection.
16. Waits briefly, then checks whether additional authentication is required for the Cloud PC connection. It loops (up to the configured timeout) looking for either:
    1. A new remote desktop connection window to appear (indicating the connection window has launched), or
    2. A Cloud PC sign-in web prompt
17. If a Cloud PC sign-in web prompt appears, it attempts to complete it:
    1. Enters the password, and clicks **Sign in**
    2. Switches to **Use a verification code** and enters a newly generated TOTP code
    3. Clicks **Verify**
18. If a confirmation/consent prompt appears with a **Yes** button (e.g., an approval/confirmation step), it clicks **Yes**
19. After initiating the connection and handling any prompts it detects, it waits a short period, then attempts to reset the Windows App back to a clean **Sign in** state again (signs out if still signed in).
20. Stops the Windows App (ends the automation run)

Prerequisites

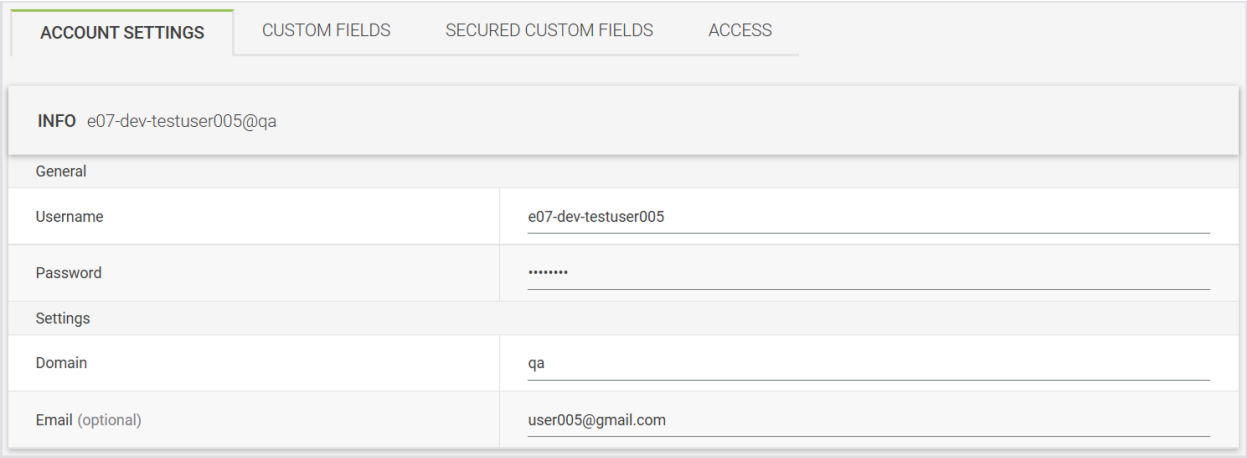
1. **Launcher installed**: Install the required Launcher version (the version that includes the Windows 365 Connector support).
2. **Windows app installed**: The Windows App must be installed on the Launcher machine.
3. **Warm (pre-used) Windows account** (recommended): The Windows account used on the Launcher machine should have previously signed in to the Windows App installed on that machine.

NOTE: This is a best practice. First-time sign-in flows in the Windows App might display one-time confirmation or welcome dialogs that can't be reliably reproduced or automated. This prerequisite is evaluated per Windows App installation. Deploying a new Launcher machine, installing a new instance of the Windows App, or reinstalling the Windows App causes the account to be treated as new for this prerequisite.

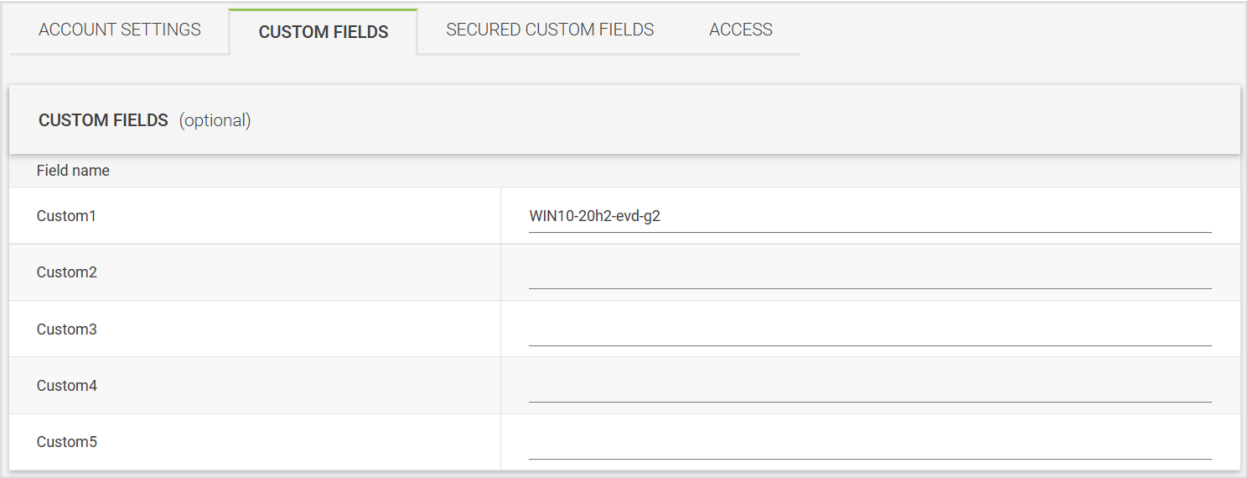
Preparing the Virtual User Account

1. In the **Login Enterprise sidebar** > navigate to **Accounts**.
2. In **Accounts Settings** > **Email**, enter the user's login information. You can use either the UPN (User Principal Name) or the email address associated with the account.

IMPORTANT: While the Email field appears as optional, it is required for the Windows 365 connector.



3. Select **Custom Fields** from the tab menu.
4. In **Custom Field 1** (recommended): Enter the Cloud PC or Virtual Machine name that the test should connect to.
    1. **Getting a Virtual Machine or Device Name for Testing**:
        1. Sign in to the **Windows 365 portal** at windows365.microsoft.com.
        2. Go to **Cloud PCs**.
        3. Look under the **Device Name** column: this is the value you should use in **Login Enterprise** > **Accounts** > **Custom Field 1**.



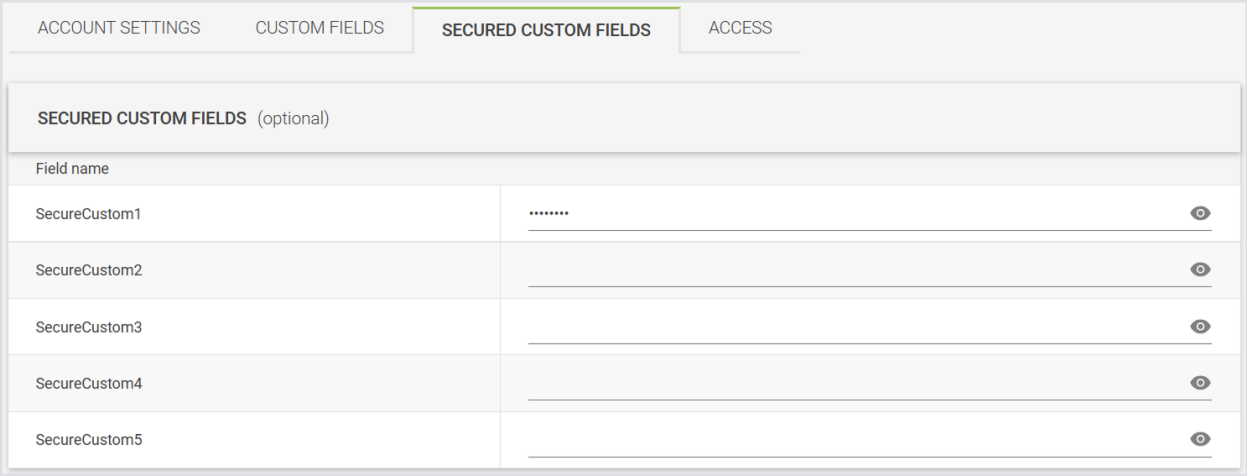
5. Select **Secured Custom Fields** from the tab menu.
6. In Secured Custom field 1 (recommended): Enter the Time-based One-Time Password (TOTP) secret for multi-factor authentication.
    1. **Getting a TOTP Secret**:
        1. **If you haven't set up the authenticator app yet**:
            1. Sign in to your account at account.microsoft.com.
            2. Go to **Security** > **Advanced security options**.

3. Under **Additional security options**, select **Add a new sign-in method** > **Authenticator app**.
4. During setup, a QR code is displayed. Select the option to manually enter the code (for example, “Can't scan the QR code”).
5. Copy the displayed secret key (TOTP seed).

Save this secret key. Enter this value in Secured Custom field 1 in your Windows 365 configuration.

2. **If the authenticator app is already set up**:
   1. The secret key is shown only during the authenticator app setup and can’t be viewed later in the Microsoft account security settings. If you didn’t save the secret key during setup, you must remove and reconfigure the authenticator app to generate a new secret.

7. Click **Save** to apply the changes.



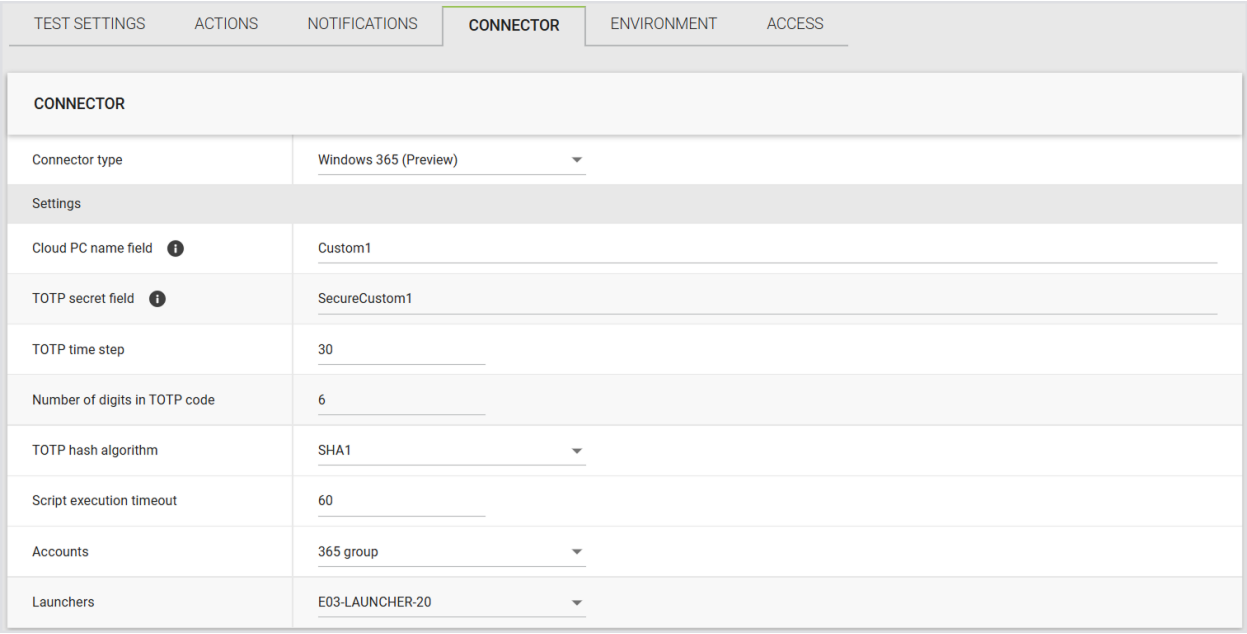
NOTE: You can use different secure fields and custom fields if needed, but you must then update the connector settings accordingly. Using the first custom field avoids the need for additional configuration.

Configuring the Windows 365 Connector

1. In the **Login Enterprise sidebar** > navigate to **Configuration** > **Manage tests**, and create an Application or Continuous Test.
2. In **Connector** settings, select **Windows 365 (Preview)** as a connector type.
3. Configure the following settings:

Field	Description
Cloud PC name field	The secure custom field in the account configuration that stores the Cloud PC name. The default value: **Custom1**

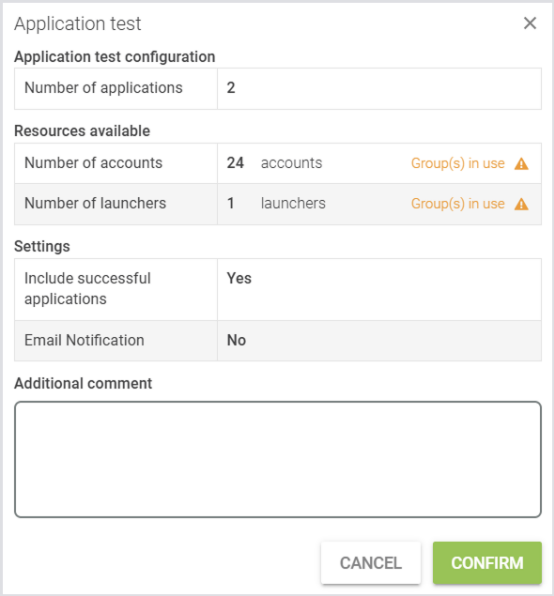
TOTP secret field	The secure custom field in the account configuration that stores the TOTP secret used to generate the one-time passcode. The default value: **SecureCustom1**
TOTP time step	Leave this field at its default value unless troubleshooting.
Number of digits in TOTP code	Leave this field at its default value unless troubleshooting.
TOTP hash algorithm	Leave this field at its default value unless troubleshooting.
Script execution timeout	Leave this field at its default value unless troubleshooting.
Accounts	Select from previously configured Account Groups or all Accounts. **NOTE:** All required authentication secrets must be configured.
Launchers	Select from previously configured Launcher Groups or all Launchers.



NOTE: If you store required values in different custom fields, make sure to update the connector settings to reference those fields.

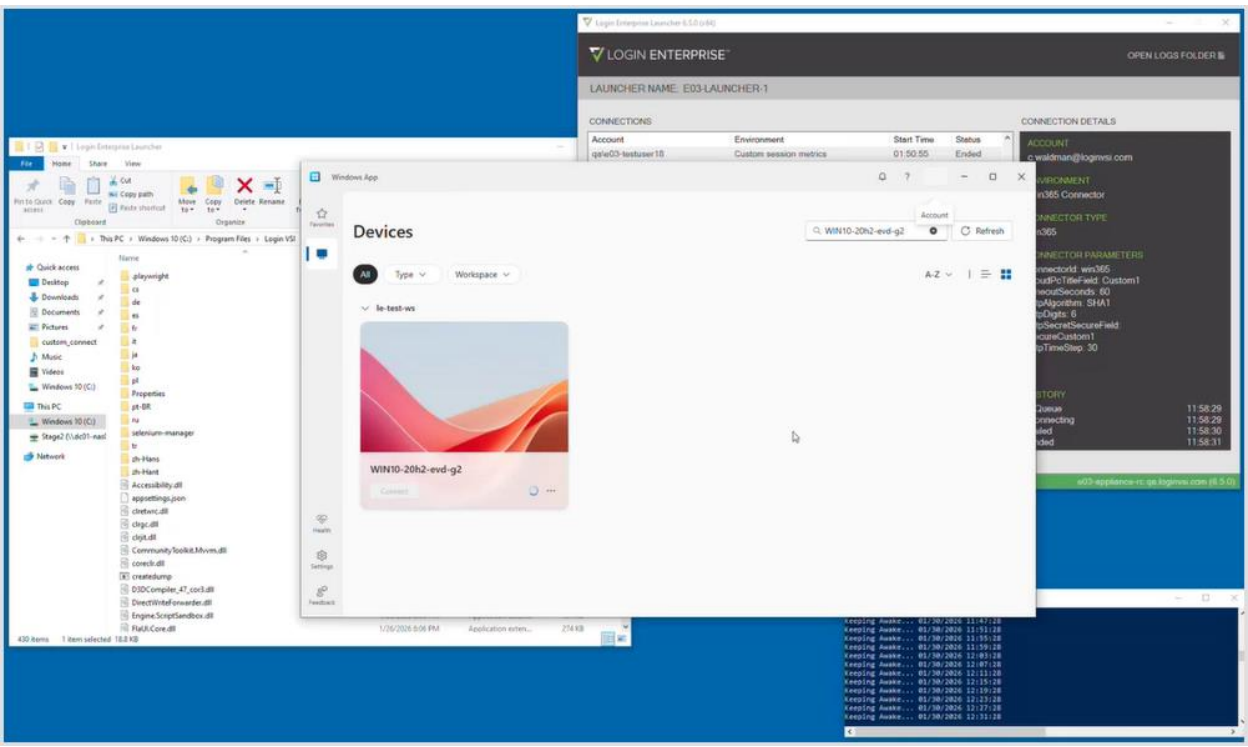
Running a Test with the Windows 365 Connector

1. In your newly configured test, click **Run Test**.
2. Review the test settings and verify they match your intended configuration.
3. Click **Confirm**.



Expected Behavior

During execution, the Windows 365 Connector behavior should match the steps described in [How the Windows 365 Connector Works](#) section. The result should look as follows:



Windows 365 Connector Script Management

Windows 365 Connector scripts are centrally managed within Login Enterprise, removing the need for per-launcher updates. The system includes multiple built-in script versions and automatically selects the best match for the Appliance version. If no suitable match is found, the latest available built-in script is used as a fallback.

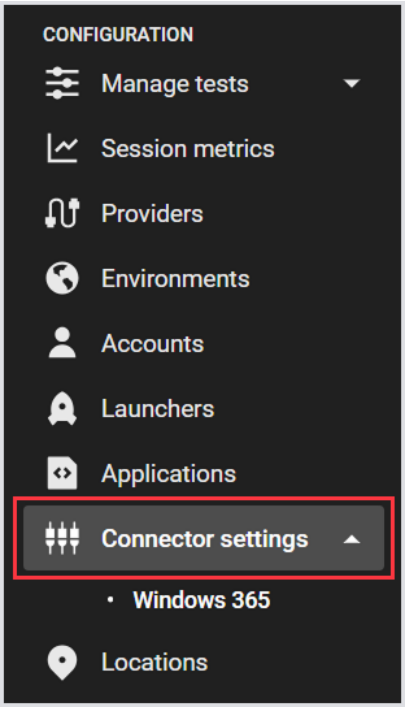
Alternatively, you can provide a custom script. If a custom script is provided, only that single script is always used, and no automatic matching is performed. You need to choose between using the system's matching logic or supplying a custom script.

You can manage Windows 365 connector scripts in the Login Enterprise UI and/or by using the Public API.

IMPORTANT: To be able to perform Windows 365 Connector script upload and management, you need to upgrade both the Appliance and the Launcher to v6.6. For more information, see Updating the Appliance and Updating the Launcher.

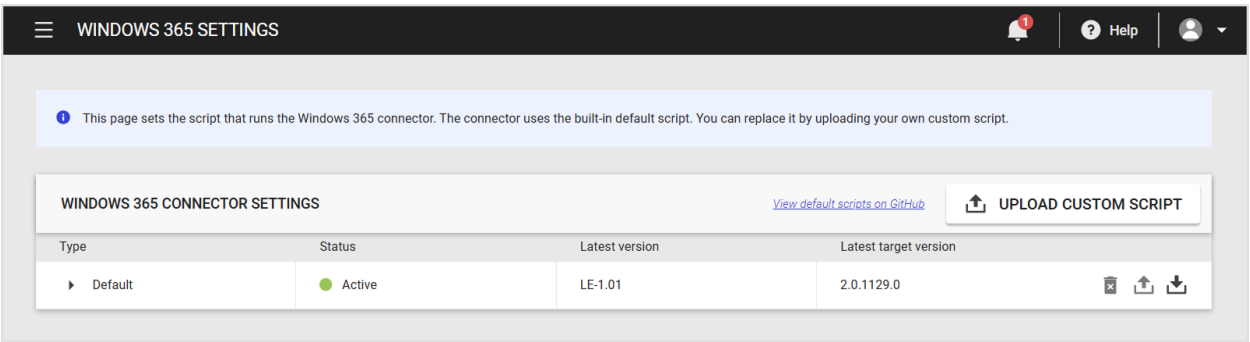
In the UI

In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Connector settings** > **Windows 365**.



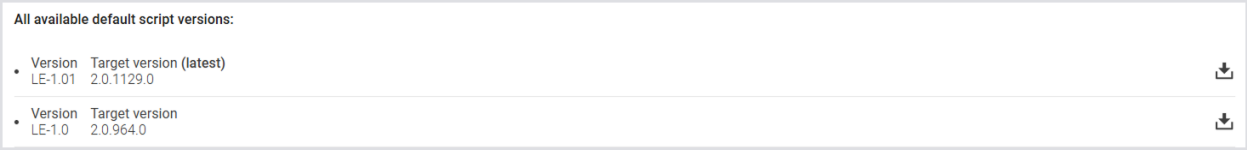
Use the **Windows 365 settings** page to manage the script that runs the Windows 365 connector. You can view the available scripts, upload a custom script, download scripts, or delete a custom script.

Column	Description
Type	The script type, e.g., **Default** or **Custom**.
Status	Indicates which script is active. If no custom script exists, the default script is active. If a custom script exists, the custom script is active, and the default script is inactive.
Latest version	The latest available version of the script.
Latest target version	The Windows App version targeted by the latest script version.



Expand the **Default** row to view all available default script versions. Each version shows its target version and includes an option to download the script.

When a Windows 365 connector test runs, the connector checks the Windows App version and uses the default script version that matches the target version. If no matching target version is available, the connector uses the default script with the latest target version.



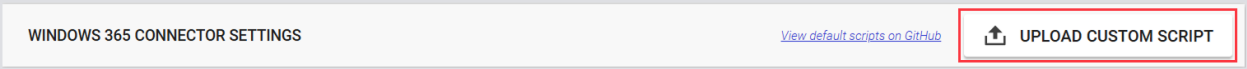
Uploading a Custom Script

By default, the Windows 365 connector uses the built-in default script. To use your own script, upload a custom script. The custom script applies to all Windows 365 Connector Tests. It can't be configured per Test or per Launcher.

When you upload a custom script, the custom script overrides the built-in default script. If you delete the custom script, the connector reverts to the default script.

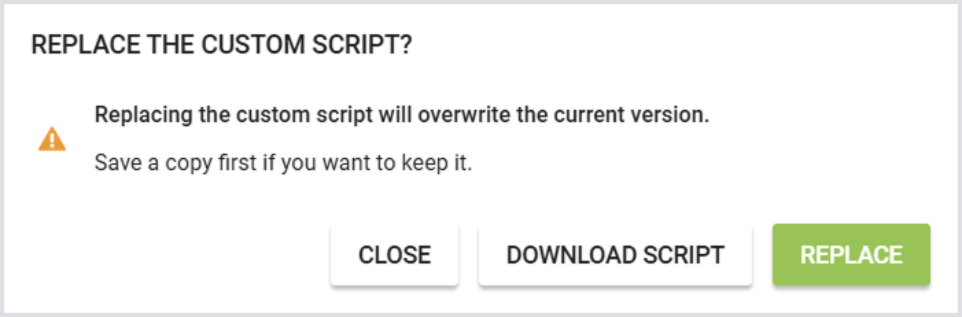
To upload a custom script:

1. Click **Upload custom script**.



2. Select a `.cs` file. Only one custom script can be uploaded.
3. Complete the upload.

You can also upload a custom script from the **Custom** row. If a custom script already exists, confirm that you want to replace it. Before you replace the script, you can download the current script from the confirmation dialog.

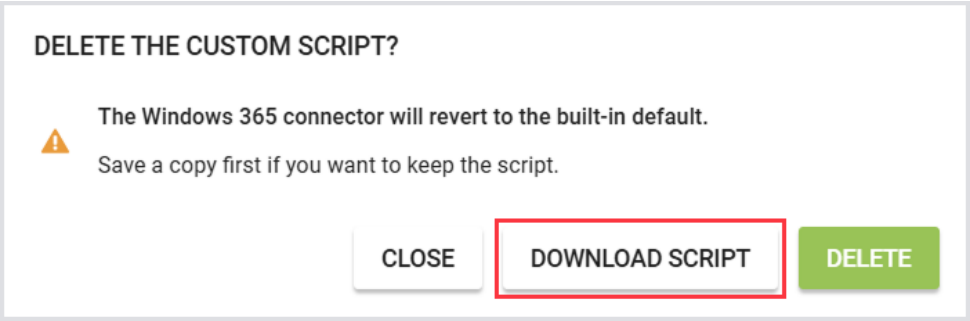


NOTE: You can't upload a script for the **Default** row.

Downloading the Script

You can download default and custom scripts:

- For default scripts, expand the **Default** row, and then select the download option for the version you want to download.
- To delete a custom script, select the delete option for the **Custom** row, and then confirm the deletion. Before you delete the script, you can download it from the confirmation dialog.



NOTE: You can't delete the default script.

Using the Public API

This section describes how to retrieve and create Windows 365 Connector scripts via Public API, as well as the required structure of script files.

ConnectorScript Model

Endpoint responses return a `ConnectorScript` model with the following fields:

Name	Value
Version	A version number that identifies the script
Target	The platform this script is for (e.g., Windows App)
Target Version	A version of the platform that the script supports
Script Content	The actual script code
Is Custom	Whether this is your custom script or a system default
Created Time	When the script was created

Creating a Script

When creating a script via the API, provide the following fields:

Name	Value
Target Platform	Specifies the target platform for connector scripts. Currently, only Windows App (1) is supported.
Script Content	The script code you want to use.

Script File Requirements

Connector scripts must include metadata comments at the top of the script file. These comments are parsed by Login Enterprise to validate and manage scripts.

IMPORTANT: Header field names and values are case-insensitive.

Each header must follow this format:

Name	Value
Version	The version identifier of the script (e.g., LE-1.0, Custom-2.0). Used to track script versions. Max 64 characters.
Target	The target platform/application this script is designed for. Currently supported: Windows App
Target Version	The version of the target application this script supports (e.g., 2.0.964.0). Used to match scripts to installed application versions. **NOTE:** This field is mandatory only for default scripts, and can be skipped for custom scripts.

The header prefix must include a colon followed by a space, as this format is required for proper parsing.

Example:

```
// Version: LE-1.0
// Target: Windows App
// Target version: 2.0.964.0
```

Desktop Connector

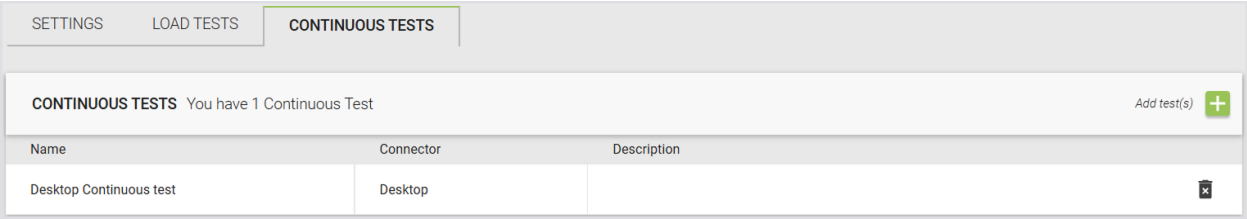
- Overview
- Continuous Testing Scenario
- Load Testing Scenario
- Application Testing Scenario
- Logon App

Overview

The Desktop Connector lets you trigger a Test run without needing a remote-access system. Instead of using a Launcher that triggers a login for a properly configured account, you, as a user, run the appropriate program by hand. When you select a Desktop Connector, there are no configuration parameters. The Test definition lets you simply copy a command (LoginPI.Login.exe <URL> <TestID>) and paste it into your existing session to download and execute the Test.

The Desktop connection method is highly beneficial for initial testing and can also prove invaluable in unique situations where creating a remote session is not feasible. Instead, you can use existing machines with active interactive sessions to trigger the Tests. To set the Connector in the environment (Continuous, Load, or Application Testing), you need 3 values:

Name	Connector	Description
The designated title or identifier for the environment.	Desktop	A summary that outlines the purpose and key characteristics of the environment.



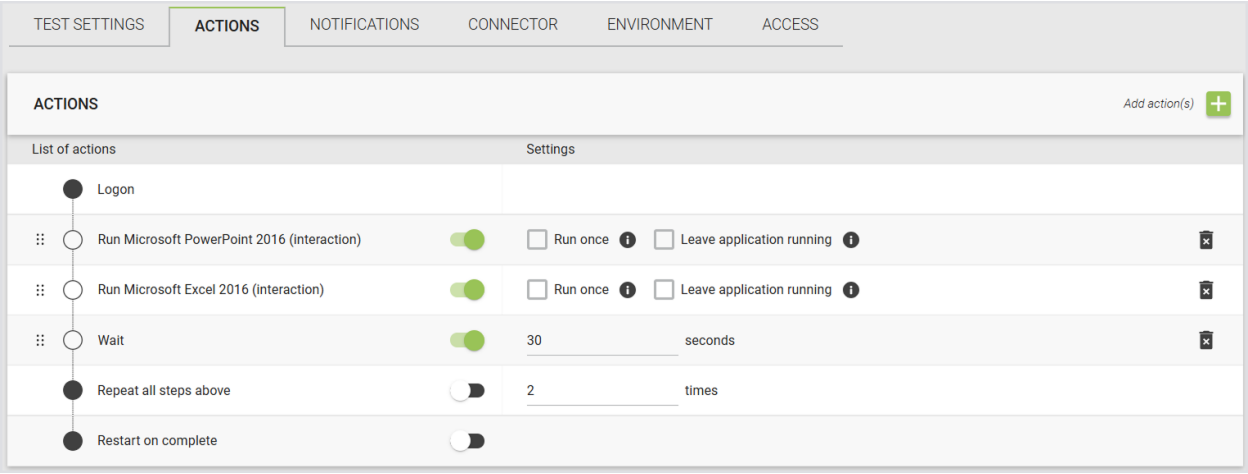
NOTE: Once configured, you might notice the lack of other configurations as seen in the different connectors. This is expected behavior as we start the process from the machine; there is no remote connection required. You can now configure the environment as usual.

Continuous Testing Scenario

There are other differences in the connection setup of the Desktop Connection with other remoting configurations. With Continuous Testing, enabling the schedule will allow the Desktop Connector to start. If it is disabled, no sessions can be initiated.

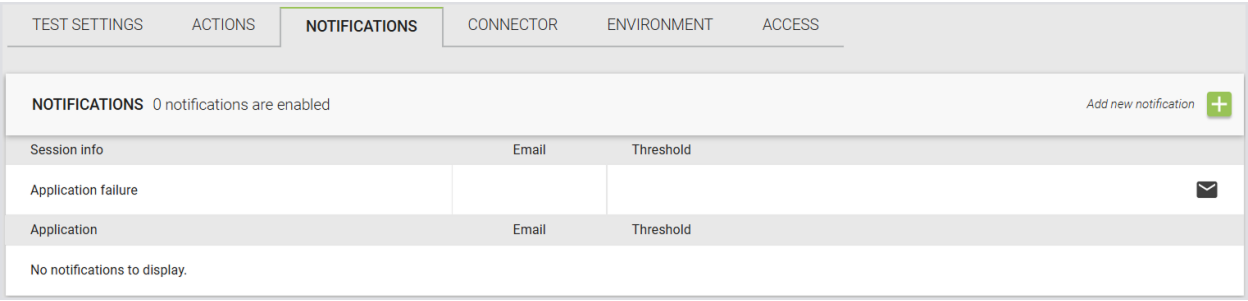
The **Actions** list is almost the same as the Continuous Testing that uses a remote connection. There is a single difference. Instead of **Logging off** at the end, there's **Restart on complete**. Restart on complete automatically restarts the machine. Depending on the configuration, the machine will auto-start / login and repeat the same process.

TIP: If you do not want the machine to reboot after each run, you can decide if you want to run the process once or repeat it x number of times. We have seen situations where customers want to repeat the process 9999 times using **Repeat all steps above**.



Notifications are also different, as you can see, there are no results concerning the remote connection to the environment. There are no Logon Failures, Logon Measurements, and Latency measurements done. Only application failures are measured by default.

TIP: You have the option, similar to other situations, to add specific thresholds you wish to be notified of when they are crossed.



For more information on configuring Continuous Testing, see [Configuring Continuous Testing](#).

Load Testing Scenario

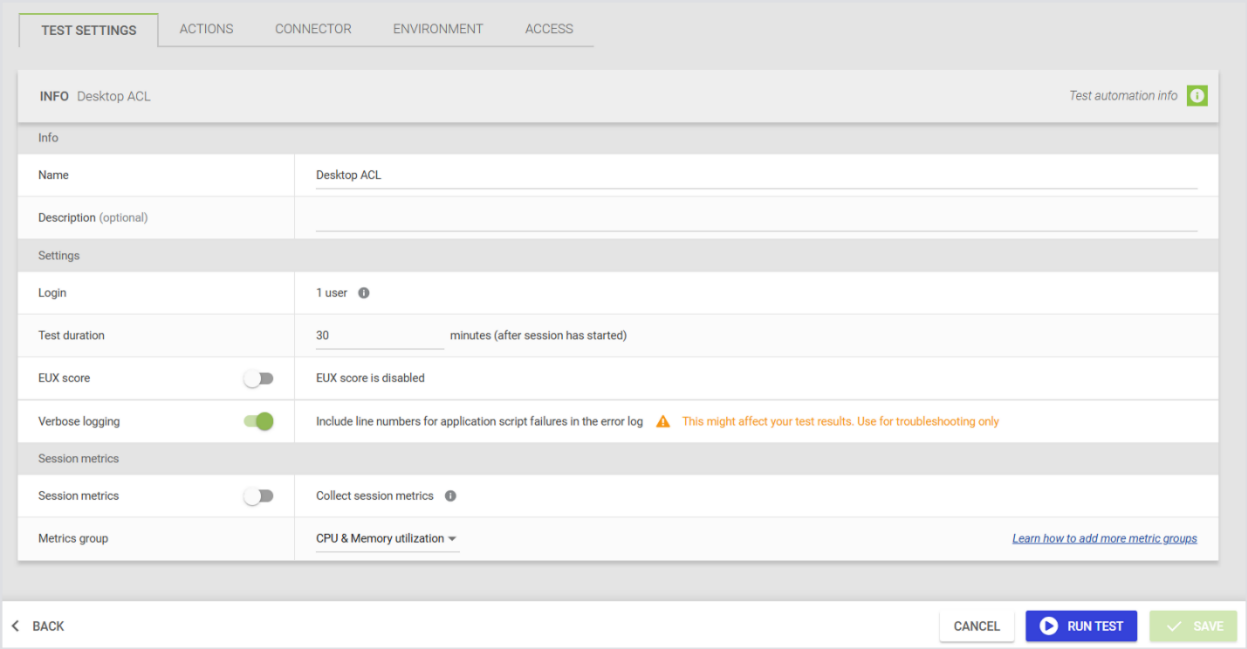
You can run a Load Test by using a Desktop Connector. When you create a new Load Test, the Connector dropdown includes Desktop as an additional option. Selecting a Desktop Connector gives you access to capabilities that aren't available in Continuous Testing or Application Testing, such as:

- Session metrics
- EUX calculation
- PDF reporting

Desktop Connectors behave differently from other connectors:

- Desktop Tests don't include a login process.
- Tests always run with one user.
- The VSImax users setting is hidden.
- Even though this is a one-user Test, the limitation that only one Load Test can run at a time still applies.
- The "Schedule (in which sessions)" row in the Session metrics configuration is not shown. Session metrics are collected for the single session by default.

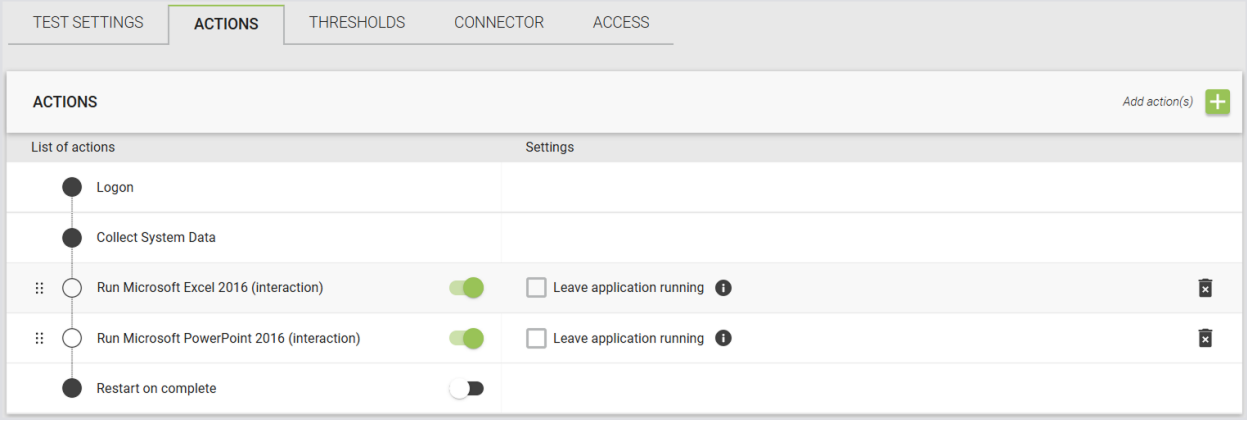
Here's how the Load Test settings appear when you select a Desktop connector:



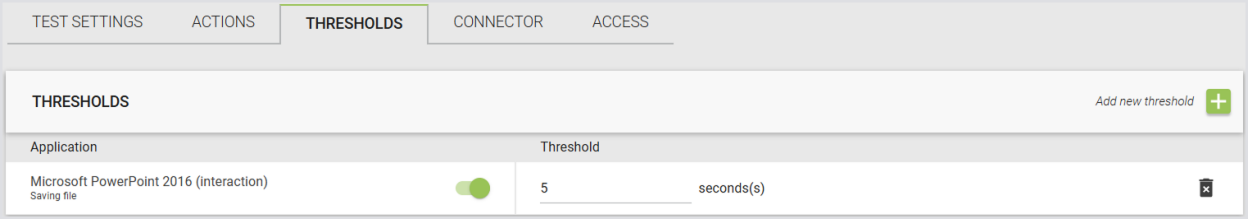
For details on configuring Load Testing, see [Configuring Load Testing](#).

Application Testing Scenario

Application testing configuration differs from Continuous Testing and Application Testing done with a remoting protocol. Again, **Restart on complete** is added to the **Actions** list. You can decide to reboot the machine when it is done.



The thresholds are configurable as normal, but without the Login and Latency measurements.



For more information on configuring Application Testing, see Configuring Application Testing.

Logon App

To configure the Logon App for Desktop situations, see Configuring Logon Components.

Custom Connector

- Overview
- Building a Custom Connector
- Configuring the Custom Connector
  - Connection Command Line
  - Custom Fields
    - Benefits of Using Custom Fields
    - Setting Custom Field Values
    - Adding Custom Fields as Arguments
    - Modifying Command Line Values
    - Other Customization Options
- Custom Events for Connectors and Launchers
- Secure Custom Fields in Connector Configurations
- Process Tracking (Optional)
  - Configuring Process Tracking
  - Behavior Overview
  - Best Practices
- Named Pipes for Custom Connector

Overview

The Custom Connector provides a flexible solution for defining commands with parameter substitution, allowing the Launcher to execute various remote connection software. This versatility enables you to trigger connections using executable clients on the Launcher machine. The Custom Connector allows the Launcher to call an external program to perform the connection, rather than using a built-in Connector.

When none of the predefined connector options meet your requirements, the Custom Connector offers a tailored approach. If your connection system provides a command-line client that accepts parameters on the command line, you can simply call it. You can also build automation scripts using the Long Enterprise Script Editor to manage your interactive connections. We also provide a general web-interface automation framework called the Universal Web Connector (UWC). For guidance on using the UWC with the Custom Connector, see Configuring the Universal Web Connector.

Building a Custom Connector

If your connection system provides a command-line interface where you can specify credentials, addresses, and resources on the command line, that is the simplest option. Simply make sure the client software is installed on the Launcher, and then construct your command-line in the test configuration Connectors page.

If your connection system requires some kind of automation scripting, you can build a Login Enterprise Engine Script using our script Editor. There are stand-alone Engine executables available in the Script Editor in "ScriptEditor\engine\LoginEnterprise.Engine.Standalone.exe" or as part of the Universal Web Connector in "C:\Program Files\Login VSI\Universal Web Connector\engine\LoginEnterprise.Engine.Standalone.exe".

For an example of using an engine Script as a Custom Connector, see Deploying the AVD Connector.

If your connection system's authentication and connection interface are implemented as a web server, you may be able to leverage the Universal Web Connector instead. It provides a framework where customer-generated JavaScript code can interact directly with the web page to fill in fields, select controls, and eventually launch the rete client. For guidance on using the UWC with the Custom Connector, see Configuring the Universal Web Connector.

Configuring the Custom Connector

In Login Enterprise, the Custom Connector supports the following parameters:

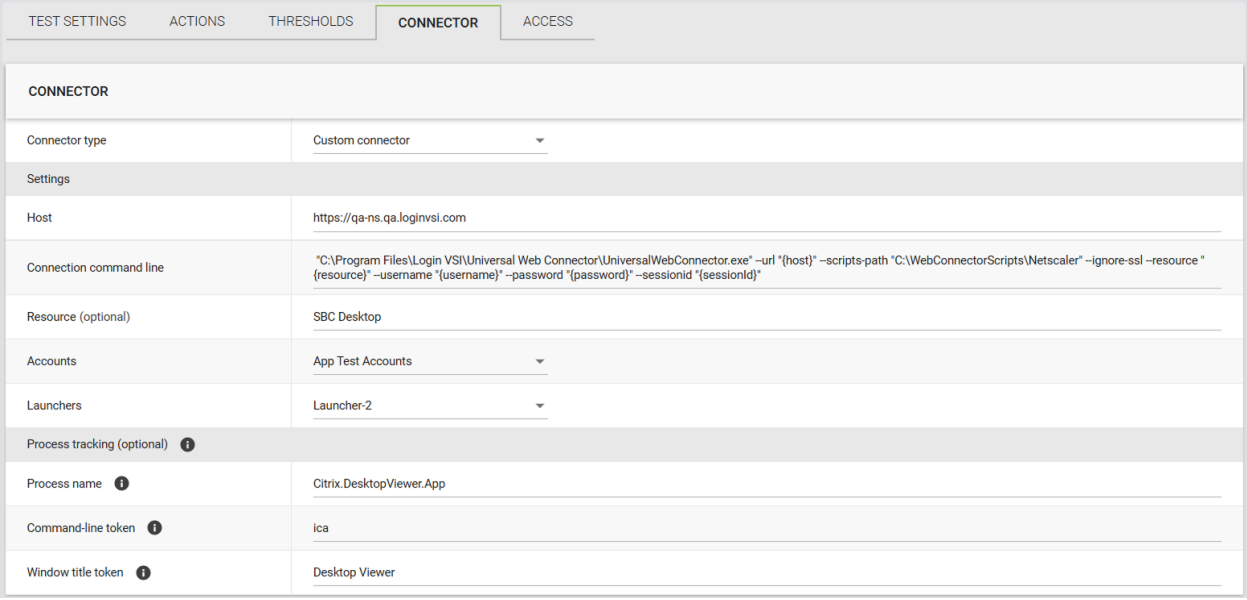
Field	Description
Host	The connection server.
Resource	The resource name visible to end users.
Connection command line	The command line executed on the Launcher machine to initiate a session. See Connection Command Line for details.
Accounts	Select from previously configured Account Groups or all Accounts.
Launchers	Select from previously configured Launcher Groups or all Launchers.

Connection Command Line

The following variables are used in the connection command line for the Custom Connector.

Field	Description
host	The "Host" field from the Custom Connector configuration
username	The username field from the selected account
password	The plain-text password from the selected account
domain	The domain field from the selected account
resource	The "Resource" field from the Custom Connector configuration
email	The e-mail field from the selected account

sessionId	The current session ID provided by the Launcher for event tagging
custom*X*	The Custom1 to Custom5 fields configured within the Account record
SecureCustom*X*	The Secure Custom1 to Secure Custom5 fields configured within the Account recordSecureCustomX



Custom Fields

You can add custom fields during account configuration and use them with Custom Connectors, just like you would with email, username, domain, password, and other standard fields. You can define up to five custom fields with editable values, but their names remain fixed.

Benefits of Using Custom Fields

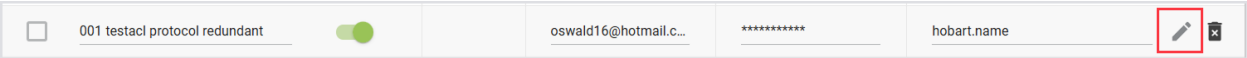
Custom fields offer several practical advantages, especially when used with the Universal Web Connector:

Benefit	Description
Enhanced customization	You can store extra information, like additional usernames or authentication tokens, making your setup more specific to your needs.
Improved flexibility	Custom fields let you pass values in command lines, so you can easily adapt to different configurations without changing core settings.
Streamlined testing	You can quickly adjust field values for testing. This makes your testing process more efficient and thorough.

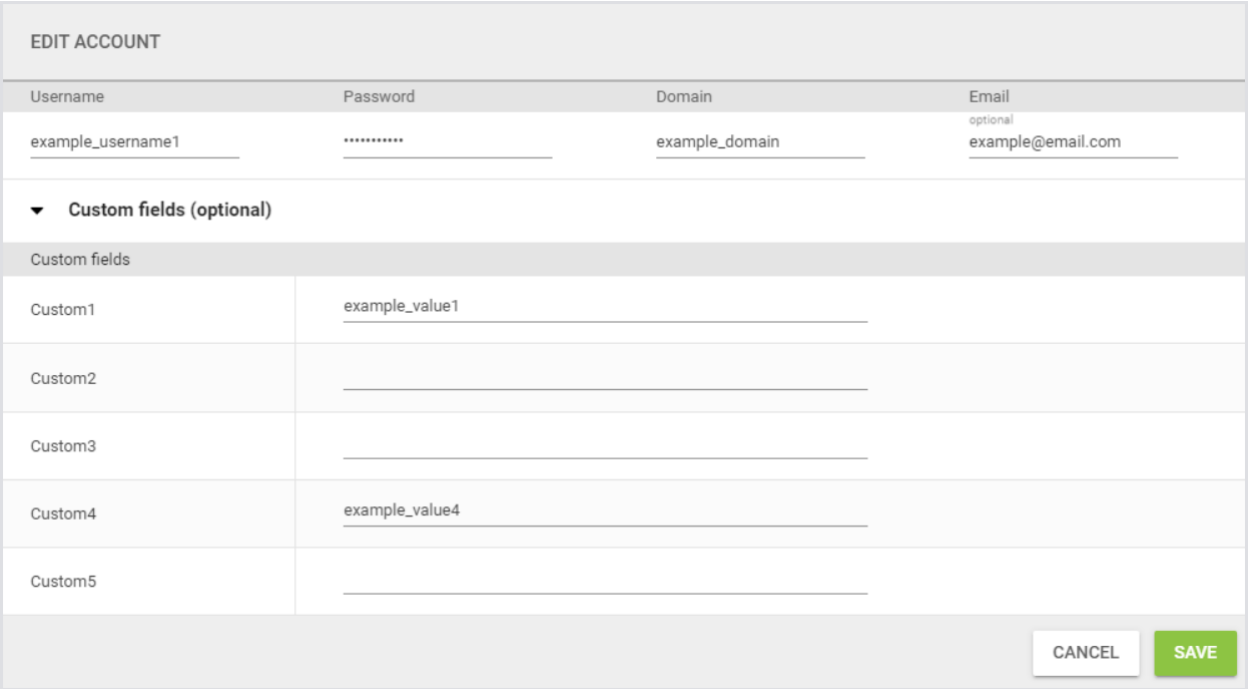
Seamless integration	Custom fields work well with Custom Connectors and Swagger, helping you manage extra data smoothly across different tools.
Increased efficiency	Updating custom field values through Launchers and command lines reduces manual errors and saves time.

Setting Custom Field Values

Custom field values can be set when creating a new Account. To edit these values later, use the account editing form. Click the "pencil" icon in each row to open this form, which will display the fields with their current values pre-populated.



If any custom fields have been saved, the panel automatically expands to show their values. Custom fields can hold various types of values, including a single digit, random text, or an authentication token. They accept any string up to 1,024 characters in length.



NOTE: To use saved custom field values with Custom Connectors, you need to update your Launchers. This ensures that the latest custom field values are correctly integrated and utilized. For more information, see Launchers.

Adding Custom Fields as Arguments

Once the Launchers are updated, you can include custom fields as arguments when starting a new Custom Connector process. For example, you might pass a custom field value like this:

```
CustomConnector.exe --custom1 "{custom1}"
```

Modifying Command Line Values

You can also modify the command line to source values for username, email, domain, and other fields from custom fields instead of their dedicated fields. For example:

```
CustomConnector.exe --username "{username}" --password "{password}" --domain
"{domain}"--resource "{resource}" --email "{email}"
```

Can be updated to:

```
CustomConnector.exe --username "{custom1}"--password "{custom2}" --domain
"{custom3}" --resource "{custom4}" --email "{custom5}"
```

This flexibility allows you to use custom field values in various configurations and testing scenarios.

Other Customization Options

In some cases, you might need to test with an additional username, password, or authentication token. Custom fields provide this additional level of customization.

Custom fields can also be added when creating accounts in Swagger (API v5 and v6). Since this is optional, the property can be set to `null` or omitted entirely.

Custom Events for Connectors and Launchers

You can emit custom events directly from connector scripts and view them alongside standard Launcher events in the Events feed. This helps you trace and diagnose Custom Connector behavior without relying on external logs. For implementation information and example scripts, see Custom Events for Connectors and Launchers.

Secure Custom Fields in Connector Configurations

Secure custom fields can be passed to Launchers and Connectors at runtime, the same way password fields are handled.

Runtime Behavior

- Launchers and connectors receive secured values at runtime in the same way as passwords.
- Values aren't retrievable after the test finishes.

UWC/Custom Connectors

When you use secure custom fields with the Custom Connector, pass them as command-line parameters. Use these parameter names in the same way you use `{custom1}`, `{custom2}`, and so on for standard custom fields.

Field	Parameter name
SecureCustom1	`{securecustom1}`
SecureCustom2	`{securecustom2}`
SecureCustom3	`{securecustom3}`
SecureCustom4	`{securecustom4}`
SecureCustom5	`{securecustom5}`

For details on configuring secure custom fields in accounts, see Secure Custom Fields (Optional).

Process Tracking (Optional)

By default, the Launcher will track the specific process ID it gets when it starts the Custom Connector. Login Enterprise will track the main window associated with that process ID, and report when that process ID terminates. If your Custom Connector creates additional processes, and if the initial process terminates before the remote session, the Launcher can find itself watching the wrong process.

Process tracking lets you monitor your Custom Connector and any processes it creates. It helps you:

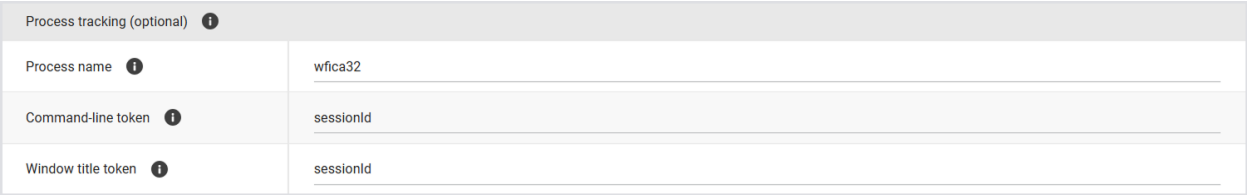
- See exactly when connections start and stop.
- Identify the single process associated with the connector for the session (this may be the connector process itself or one of its child processes) that is configured for tracking.
- Troubleshoot issues faster because you know which processes are running.

This ensures your Custom Connector runs smoothly and gives you accurate insight into its lifecycle.

Configuring Process Tracking

In your **Custom Connector settings** > **Process Tracking**, configure the following 3 fields:

Field	Description
Process name	The name of the process to monitor. Process names don't include file extensions. For example, enter `mstsc`, not `mstsc.exe`.
Command-line token	A substring to match in the process command line.
Window title token	A substring to match in the main window title.



So that the fields are filled accurately, follow these rules:

- Maximum 255 characters per field.
- Leading and trailing spaces are removed automatically.
- Token examples: `{sessionid}`, `{username}`, `{password}`, `{domain}`, `{custom1}`.
- Functionality is enabled only if a value is present. Blank fields are ignored. The value is treated as a substring, so you don't need to enter wildcards. For example, enter `{sessionid}` instead of `*{sessionid}*`.

NOTE: The system only checks that fields aren't empty or too long. Other errors, like typos in the process name, won't be detected until you run the process.

Behavior Overview

- If you leave all fields blank, process tracking is turned off, and the system defaults to the legacy behavior. If you enter one or more values, process tracking is turned on.
- The system checks all the fields you enter and tracks a process **only when every field matches**.
    - For example, if you enter both the **Process name** and the **Window title token**, the system will track the process **only if both match**.
- Matching happens automatically on the backend, i.e., you do not need to add scripts to your Custom Connector.
- If no match is found, the system defaults to the legacy behavior.
- A session is considered ended when the tracked process stops or the window closes. The engine timeout still applies, but is used only as a fallback safety measure.

Best Practices

- Use unique identifiers in the **Window title** or **Command-line token** to avoid conflicts in multi-session scenarios.
- Configure **at least one field** to ensure accurate session tracking.
- Avoid relying solely on the **Process name** unless you are sure it will be unique.
- Keep fields concise, under 255 characters, and avoid unnecessary spaces.
- Track the actual remote session process, not the Custom Connector that initiates it. For example, if your connector is PowerShell-based, track the Windows remote session process (`mstsc`) instead of the PowerShell process.

Named Pipes for Custom Connector

Custom Connectors communicate with the Launcher using Named Pipes, a lightweight and efficient Windows inter-process communication (IPC) mechanism. Named Pipes support bilateral communication, allowing both the Launcher and the Custom Connector to exchange structured messages with each other without relying on network services, sockets, or external dependencies.

Using Named Pipes improves reliability and bridges the gap between Custom Connectors and Built-in Connectors. It also replaces command-line arguments, which can expose sensitive data in logs or the Event Viewer. For implementation information and sample code for this system, see Named Pipes for Custom Connector.

Configuring the Universal Web Connector

- Overview
- When to use UWC
- Downloading and installing UWC
- Connector templates
- Arguments
  - Managing static URL scenarios
- Configuring a Test to use UWC
  - Configuring UWC in Login Enterprise
  - Running UWC as a Custom Connector
- Logging
- Troubleshooting

Overview

The Universal Web Connector (UWC) is designed to seamlessly integrate with technologies offering virtual access to resources. It simplifies tasks such as automating login processes, selecting resources, and launching retrieved connection files (in scenarios where a downloaded file is used, e.g. Citrix and AWS), or executing applications (in scenarios where the relevant app is invoked directly, e.g. VMware Horizon). This is achieved through customizable connection scripts, which can be easily adjusted to accommodate evolving UI designs on web pages over time.

The UWC operates as an independent process, accessible via Login Enterprise's Custom Connector functionality. It is self-contained, bundling all necessary runtimes, frameworks, and the Login Enterprise engine within the connector itself. The only requirement is the installation of UWC on the local machine for launcher detection and utilization.

Launchers running UWC can facilitate concurrent user sessions, similar to the functionality found in standard built-in Connectors. However, it's important to note that certain engine scripts may restrict the number of engines that can run concurrently. This limitation is implemented to prevent conflicts arising from simultaneous interactions, particularly in cases where application windows are not uniquely identifiable for each user session.

By default, the UWC has a timeout period of 5 minutes, adjustable using the --timeout argument. In the event of a timeout, crash, or exception, UWC performs automatic cleanup, ensuring no leftover artifacts or lingering child processes remain.

Since the UWC relies on a Custom Connector, it may not be able to access all the information available in native connectors. In the event of a failed connection, the only indication will be the engine not establishing a connection after 5 minutes.

When to use UWC

The UWC is primarily intended for connection scenarios not supported by built-in connectors, such as NetScaler, Storefront, VMware Horizon, AVD, and RDS.

In certain instances, customizations made to platforms such as NetScaler, StoreFront, VMware Horizon, and others may require the use of the UWC. These customizations, including but not limited to configurations related to Time-Based One-Time Password (TOTP) authentication, may necessitate the utilization of the UWC for seamless integration. To learn about support for specific scenarios, contact us.

NOTE: The script files for the UWC may differ slightly from the script files used for the preview version of this connector, also known as the Browser Connector. For assistance in converting your preview version scripts to work with the UWC, contact us.

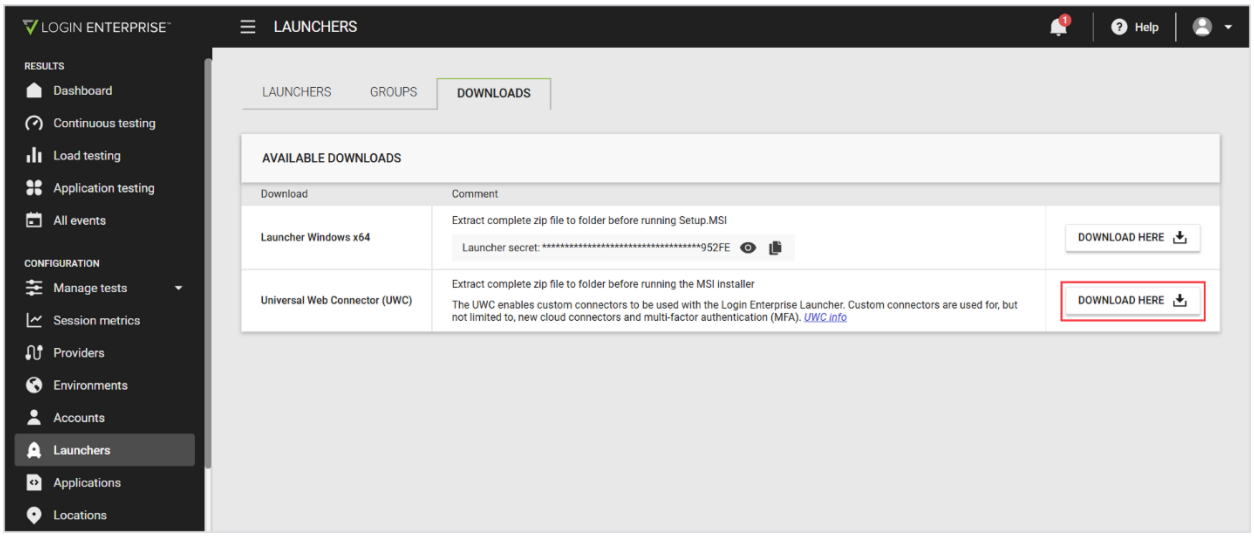
Downloading and installing UWC

The UWC can be easily installed on Windows operating systems using an MSI installer. This installer allows you to install the UWC application or update an existing installation.

NOTE: The MSI installer can update the UWC to newer versions. However, if you need to revert to an older version, you need to uninstall the current version before using the MSI installer to install the older version.

To download and install the UWC:

1. In your **Login Enterprise sidebar menu**, navigate to **Configuration > Launchers**.
2. In **Launchers**, select **Downloads**, and then download the **Universal Web Connector (UWC)** file.



TIP: Alternatively, UWC can be downloaded from the sidebar menu under **Other > Downloads > Launchers**.

3. Once downloaded, extract the complete zip file to the folder.
4. Start the UWC installation by running the MSI installer file, and following the prompts.

By default, the UWC will install to:

```
C:\Program Files\Login VSI\Universal Web Connector\
```

Once installed, make modifications as necessary in the resulting Program Files directory, and then configure the Custom Connector.

Connector templates

To avoid potential difficulties with editing or updating scripts used by the UWC due to permissions restrictions, it's recommended to place the scripts in a location other than the protected 'C:\Program Files' folder.

For a structured and compliant folder organization, we recommend the following format:

```
C:\ProgramData\LoginVSI\UWC\Scripts
```

Each connection template should be stored within its dedicated subfolder within the 'Scripts' directory.

For example, if you have different connection templates for launching 'desktops' and 'apps,' each template and its associated scripts should be stored in its respective subfolder within 'Scripts.' This approach ensures clear organization and minimizes the risk of unintended interactions between scripts customized for different environments.

Connection scripts are executed by the UWC to automate actions related to accessing, selecting, and launching virtualized resources. These actions may include clicking buttons and entering text. Connection scripts are primarily written in JavaScript, with the capability to execute C# Engine scripts and PowerShell commands.

To use connection scripts with the UWC, you need to download, unzip, and place them in a designated folder. This folder path is then passed to the connector via the '--scripts-path' argument.

Each connection script begins with a wildcarded path, such as:

```
//Path=/logon/*
```

This path acts as a trigger for script execution. For example, the above path corresponds to a web page accessed during the 'Logon' phase of the connection. When the UWC encounters a web page URL containing '/logon/', the associated script will execute.

Arguments

The UWC only requires two parameters on the command line:

- **--url**: Enter the web page address that UWC will open in its WebView window.
- **--scripts-path**: Provide the location of the Connection Script folder (it's recommended to use an absolute path).

There are 4 optional parameters that will affect the UWC operations:

- **--timeout**: Default 300 seconds, if the UWC has not completed by then, the program will quit. Specify a timeout of 0 to disable closing the UWC.
- **--ignore-ssl**: Toggle this option to bypass errors for self-signed, expired, or untrusted SSL certificates on websites.
- **--maximized**: Maximize the UWC browser window at startup.
- **--open-dev-tools**: Open the Developer Tools dialog in your browser at startup.

You can add any other command-line parameters, and they will become variables available in the JavaScript code. These must be lower-case and cannot include punctuation, dashes, or underscores. There are three parameters we have standardized within our example scripts, so it’s good to continue to use those names for consistency:

- **--username**: If needed, input the username of your account.
- **--password**: If required, input the password of your account.
- **--resource**: Specify the name of the Desktop or Application from the Resource Catalog.

Feel free to include additional custom arguments as needed to create personalized scripts. Each argument you include will become a JavaScript variable. If you just specify the command-line parameter and do not include a value, it will be treated as a Boolean flag. For instance, if you add `--customvar "{custom1}"`, it will become the variable customvar. For instance, `await SetInputValue(TryGetEmailInput, '${customvar}');`

Managing static URL scenarios

When using the UWC, you may encounter challenges when the URL of the web page remains constant throughout the interaction. In such cases, all steps of the interaction must be contained within the same script file. This may require significant customization to ensure seamless execution of the desired actions.

To address this challenge, you should carefully plan and organize your script files to accommodate all necessary steps within a single file. Consider breaking down complex

interactions into smaller, manageable tasks, and use conditional logic or loops where appropriate to control the flow of execution.

Additionally, leverage the capabilities of the UWC scripting language, such as JavaScript and PowerShell, to create dynamic and flexible scripts that can adapt to various scenarios. Regular testing and iteration may be necessary to refine and optimize the scripts for optimal performance.

Configuring a Test to use UWC

Before configuring the UWC Connector in Login Enterprise, we recommend testing the configuration from the Launcher. This can be completed with the following PowerShell code:

```
$url = 'https://myworkspace.com/'
$scriptsPath = 'C:\ProgramData\LoginVSI\UWC\Scripts'
# Either of the following options can be used
& "C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url $url --scripts-path $scriptsPath
```

To allow the UWC to run through a set of scripts, you can use the blank test harness with the following arguments:

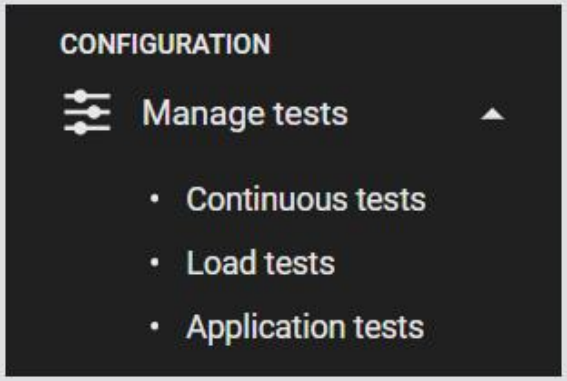
```
$username = ''
$password = ''
$resource = ''
$url = ""
$scriptPath = ""
$TeeFilepath = "c:\temp\Test.log"
& "C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "$url" --scripts-path
"$scriptpath" --resource "$resource" --username "$username" --password
"$password" --timeout 0 | Tee-Object -FilePath $TeeFilepath
```

Once the connection has been successfully established, you can configure the UWC in Login Enterprise.

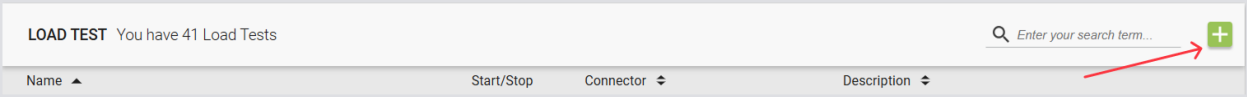
Configuring UWC in Login Enterprise

UWC can be used as a Custom Connector when configuring a Continuous, Load and Application test. To do this:

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Manage Tests**, and select the needed test from the dropdown.



2. On the top right, click the green "+" to create a new Test.



3. Give your Test a name and, optionally, a description.
4. In **Connector**, select **Custom Connector**, and specify connection settings. For details, see Configuring the Custom Connector.

Running UWC as a Custom Connector

To run the UWC as a Custom Connector, use the following commands within the 'Connection command line' field.

Without placeholders:

```
"C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "https://myworkspace.com/" --
scripts-path "C:\ProgramData\Login VSI\Universal Web Connector\Scripts" --
resource "SBC Desktop" --username "loginent0001" --password "pass**word"
```

With placeholders (for standard use within Virtual Appliance):

```
"C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "{host}" --scripts-path
"C:\ProgramData\Login VSI\Universal Web Connector\Scripts" --resource
"{resource}" --username "{username}" --password "{password}"
```

With account custom fields:

```
"C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "{host}" --scripts-path
"C:\ProgramData\Login VSI\Universal Web Connector\Scripts" --resource
"{resource}" --username "{custom1}" --password "{custom2}" --custom3
"{custom3}"
```

With custom arguments:

```
"C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "{host}" --scripts-path
"C:\ProgramData\Login VSI\Universal Web Connector\Scripts" --resource
"{resource}" --username "{username}" --password "{password}" --somenewarg
"some constant value" --somenewarg2 "{custom1}"
```

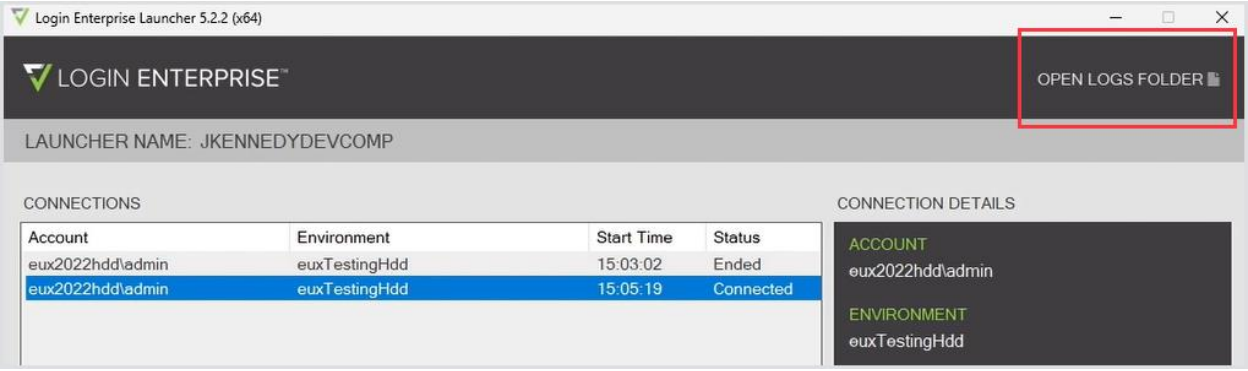
In this example, the new variables `somenewarg` and `somenewarg2` will be available in JavaScript and set to the values from the command line.

NOTE: For script examples, contact us.

Logging

If the Launcher encounters issues during the setup of the connection, such as being unable to locate the script location or access the website, all error messages are logged in the Launcher Logs, and the process will silently exit.

To access these logs, open the logs folder from the Launcher menu window:



Or navigate to the following path:

```
C:\Users<Launcher User>\AppData\Local\Temp\Login Enterprise\Launcher\Logs
```

Troubleshooting

For streamlined troubleshooting, the UWC generates error, warning, and informational messages, which are displayed in its dedicated console window. These messages include the sources of the messages, such as Dotnet, JavaScript, PowerShell, Login Enterprise Engine, and HTTP, providing valuable context for diagnosing issues.

Deploying the AVD Connector

- Overview
- Infrastructure requirements
- Software prerequisites installed on Launcher
- Building a UWC Launcher for AVD
  - Setting up the AVD Remote Desktop Client Custom Connector for Load Tests
    - Creating a Load Test
    - Configuring Custom Connector settings
- Best practices
  - Creating a Test with the AVD UWC Connector
  - AVD Remote Desktop Client (msrdc.exe) Web Connector
- Additional resources

Overview

Microsoft Azure Virtual Desktop (AVD) is a cloud-based VDI platform hosted by Microsoft Azure. With AVD, organizations can empower their workforce with remote access to essential resources through a centralized and efficient platform. Login Enterprise and all of its components can be deployed in Microsoft Azure, including the Virtual Appliance and Launchers in support of testing AVD.

This document will guide you in using Login Enterprise to test AVD via a Login Enterprise Custom Connector that lets the virtual user connect through the AVD brokering service. This testing approach is beneficial for Continuous Testing, Load Testing, and App Compatibility Testing with Login Enterprise.

Infrastructure requirements

- **AVD Host Pool**: Ensure you have configured an AVD Host Pool with at least one session host.
- **Virtual Appliance**: You will need a Login Enterprise Virtual Appliance. You can opt for the Azure deployment. For more information, see Deploying VHD to Azure.
- **Launcher**: You will need at least one Login Enterprise Launcher. These instructions are tailored for a Windows 10 Azure Marketplace image, specifically Win10 Pro 21H2. Please note that utilizing other Windows images for the launcher may necessitate additional configuration.
- **Test User Accounts**: Prepare test user accounts in Azure Active Directory (Azure AD) or Azure Active Directory Domain Services (ADDS) with appropriate access permissions to the AVD resources.

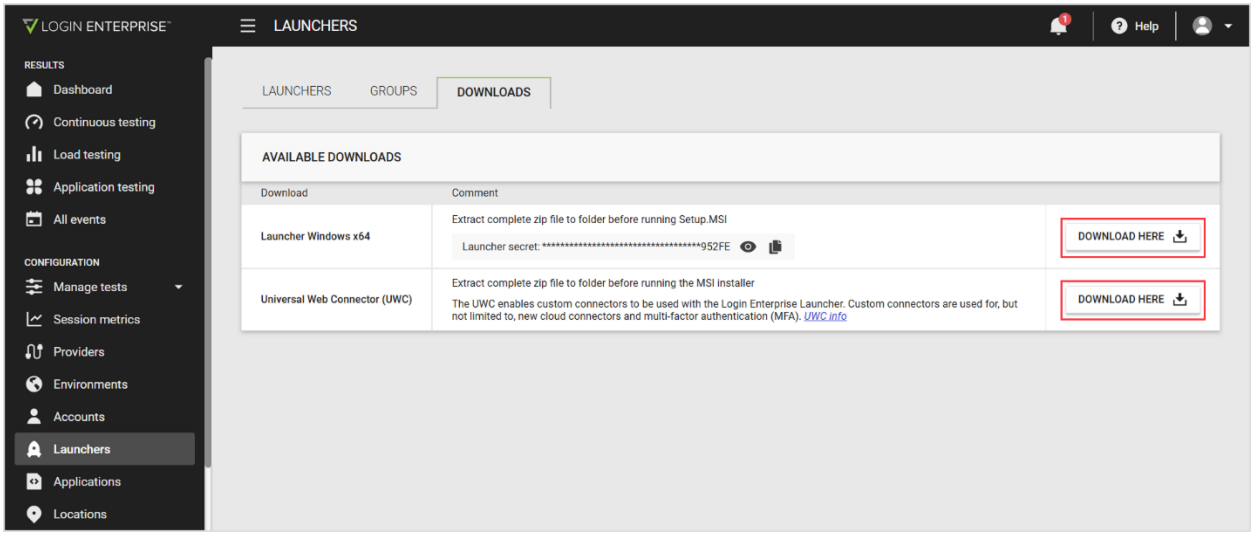
Software prerequisites installed on Launcher

- **Login Enterprise Launcher Software**: You will need to download it from your virtual appliance.

- **Login Enterprise Universal Web Connector (UWC) Software**: You will need to download it from your virtual appliance.
- **Login Enterprise UWC Connection Scripts**: For example, AVD, AVD with MFA, Windows 365, NetScaler, Citrix Cloud, etc. If you would like access to these scripts, contact us.
- **AVD Remote Desktop Client**: To download one, see Connect to Azure Virtual Desktop with the Remote Desktop client for Windows.

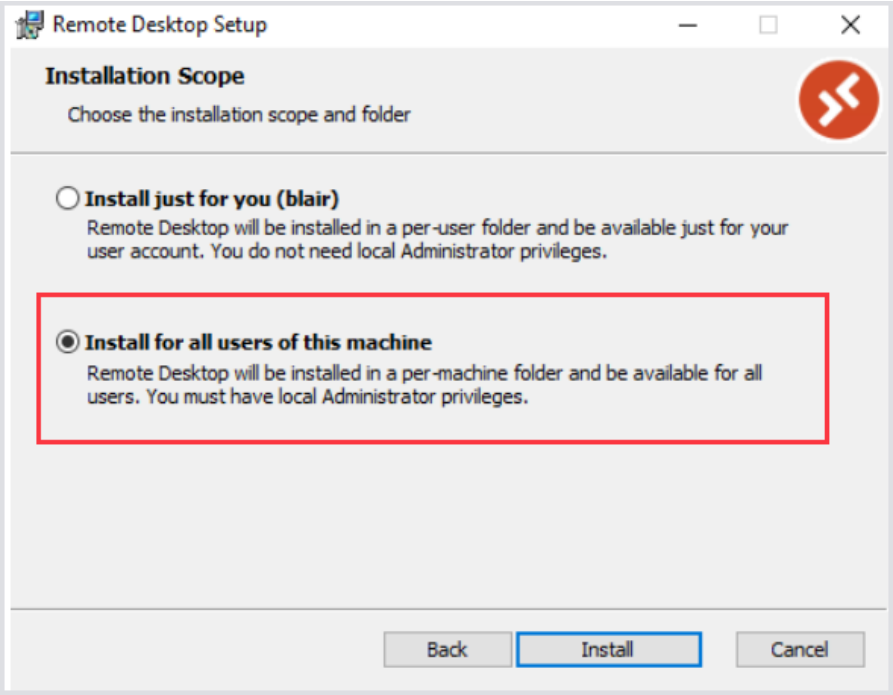
Building a UWC Launcher for AVD

1. Download the UWC and Launcher packages. For this:
    1. Log in to your Login Enterprise Virtual Appliance with the Launcher machine.
    2. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Launchers**.
    3. In **Launcher**s, select **Downloads** from the tab bar menu and download both files next to the Launcher Universal Web Connector (UWC).



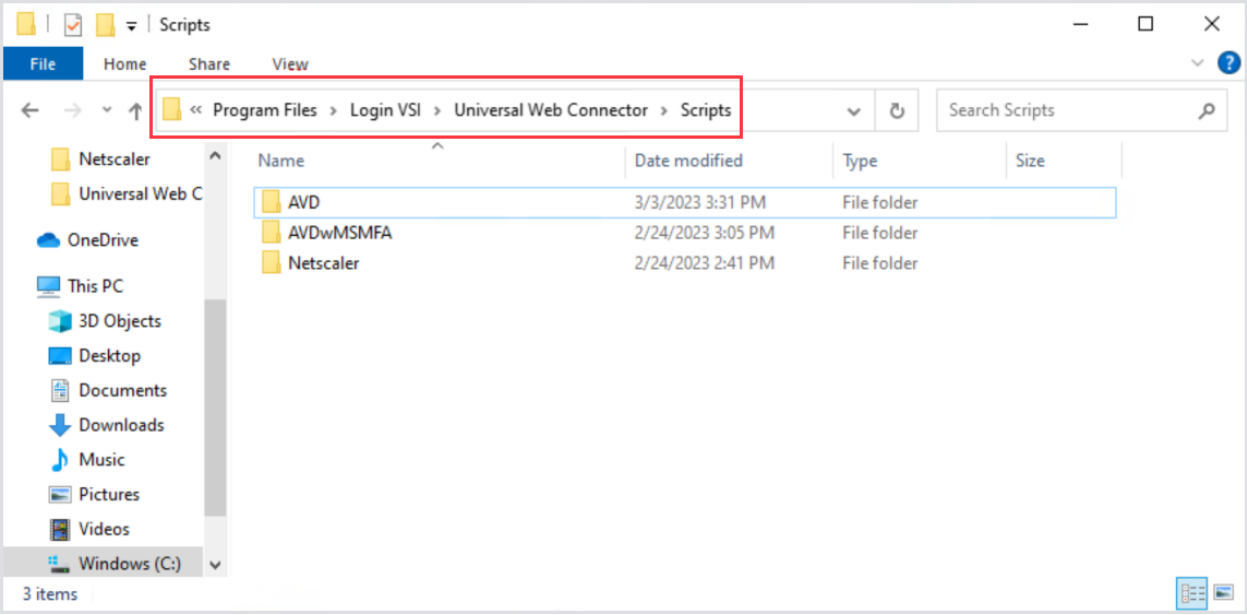
TIP: Alternatively, the Launcher and UWC can be downloaded from the sidebar menu under **Other > Downloads**.

2. Once downloaded, extract the zip files to separate folders.
3. Start the installation by running the Setup.msi file, following the prompts. Make sure to install both the Launcher and the UWC on the machine designated to be the Launcher (this can be Windows Server or Windows 10+).
4. Install the Microsoft Remote Desktop Client and ensure to select the option **Install for all users of this machine** during the installation process. For more details, see Connect to Azure Virtual Desktop with the Remote Desktop client for Windows.
5. After installing the Remote Desktop Client program, ensure that you can manually connect to a Desktop Session from the Client using a virtual user account. Take note of the resource name you wish to connect to, as it will be used in the custom command line. In the example below, you can see a resource named W10NVDesktop.



6. Locate the UWC connection scripts provided by the support team, and if necessary, unblock them (found by right-clicking on the zip file and selecting Properties).
7. Next, unzip the files and place them in a directory named 'Scripts' within the Universal Web Connector directory. For example, “C:\Program Files\Login VSI\Universal Web Connector\Scripts”.

Setting up the AVD Remote Desktop Client Custom Connector for Load Tests



The instructions below will also work for Continuous Tests and Application Tests.

Creating a Load Test

In the **Login Enterprise sidebar menu**, navigate to **Configuration > Manage tests**, and select **Load tests**. For more information, see Creating a Load Test.

Configuring Custom Connector settings

1. In **Connector**, select Custom Connector. For more information, see Custom Connector.
2. In **Host**, specify the connection server. This should be the standard Microsoft AVD gateway URL or https://rdweb.wvd.microsoft.com/arm/webclient/index.html.
3. In **Resource**, define the resource name visible to end users. This should be the resource you intend to connect to (e.g., W10NVDesktop) and it will be used by the script.
4. In the **Connection command line**, provide the command line executed on the Launcher machine to initiate a session. For example:

```
"C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "{host}" --scripts-path
"C:\Program Files\Login VSI\Universal Web Connector\Scripts\AVD" --resource
"{resource}" --username "{username}@{domain}" --password "{password}"
```

You can use one of the following methods to specify the username parameter. Modify the command line accordingly.

a. Using the {username}@{domain} variables:

b. Using the {email} variable:

The variable values you will be able to pass from the Login Enterprise Virtual Appliance to the customer connection command line are **{host}** and **{resource}** from the test configuration, and **{username}**, **{password}**, **{domain}**, **{email}**. **{custom1}** from the user account configuration. For more information on accounts and account groups, see Accounts. For more information about Load Testing, see Load Testing.

5. In **Accounts** and **Launchers**, select the account and launcher groups you want to test with. This will specify the user accounts and launchers available for this specific testing configuration.

Now you're ready to run your first Test. To learn how to start, e.g., a Load Test, see Starting a Load Test.

Best practices

Creating a Test with the AVD UWC Connector

We recommend initially setting up the AVD UWC Connector on a single Login Enterprise Launcher machine to ensure compatibility with your solution configuration. Once the functionality is validated, you can replicate this deployment by cloning the launchers, manually building them, or automating the process with PowerShell scripts.

This AVD web connector script will be executed by the Launcher service for each virtual user session.

The scalability of the Launcher depends on the workload. Typically, you can launch approximately 20 sessions on a 4-vCPU Launcher instance with no graphical or GPU-related workload, equating to around 5 sessions per vCPU. However, if the workload involves video or HD video, scalability may decrease, allowing for approximately 2 sessions per vCPU.

Ensure you're aware of the time required to establish a session, which typically ranges from 45 to 60 seconds in a standard configuration. If faster session launches are required, consider adding more launchers to the environment.

- For information on Launchers overview and best practices, see Launcher Overview and Best Practices.
- To learn about the Login Enterprise Windows Launcher setup, configuration, maintenance, and more, see Login Enterprise Windows Launcher.

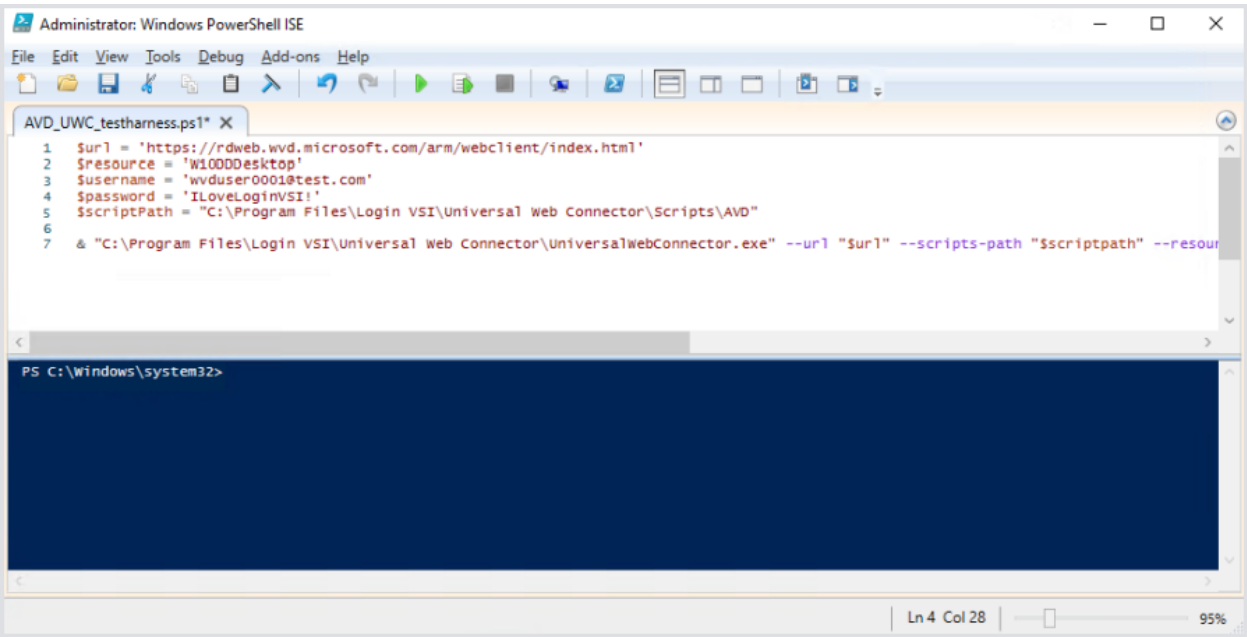
AVD Remote Desktop Client (msrdc.exe) Web Connector

Before running your first test from Login Enterprise, we recommend testing your connector using PowerShell ISE on the Launcher machine. For this:

1. On the Launcher machine, open a PowerShell ISE window. Copy the custom command script from the blue box below, paste it into the PowerShell window, and replace the following variables with their literal values: $resource,

$username, $password, and $scriptpath. In most configurations, the $url variable below is standard and does not need to be changed.

```
$url = 'https://rdweb.wvd.microsoft.com/arm/webclient/index.html'
$resource = 'MyDesktop'
$username = 'username@mydomain.com'
$password = 'mypassword'
$scriptPath = "C:\Program Files\Login VSI\Universal Web
Connector\Scripts\AVD"
& "C:\Program Files\Login VSI\Universal Web
Connector\UniversalWebConnector.exe" --url "$url" --scripts-path
"$scriptpath" --resource "$resource" --username "$username" --password
"$password"
```



2. Test the custom script to ensure its functionality by clicking the play button or pressing [F5] in PowerShell ISE.
3. If the script doesn't work as expected, you can make adjustments as necessary. Be sure to update the custom connector command line in Login Enterprise accordingly.
4. Configure your Launcher to start automatically upon login by placing it in the common Startup folder.
5. Configure the Launcher to automatically log in as an administrator, simplifying the process of resetting Launchers by rebooting. You can use the Autologon Sysinternals tool for this.

Now that you know that your command line string works, and you have created your test, run the Load Test. For this, in the L**ogin Enterprise sidebar menu**, navigate to **Configuration > Manage Tests > Load Tests**. For more information, see Starting a Load Test.

Custom Events for Connectors and Launchers

- Overview
- When to Use Custom Events
- Viewing Custom Events
- Enabling Custom Events
- Implementation
- Notes and Considerations
- Passing Session ID to the Custom Connector
  - How It Works
  - Example Connector Script

Overview

Custom events are user-defined messages sent from your connector implementation during a Test. They appear in the Events section of the Login Enterprise UI and include:

- A **timestamp**
- A **session ID**
- A **message** (custom, free-form text)

This functionality lets you track key steps, issues, or decisions made during the Test session, especially helpful when using custom logic or workflows in your connector.

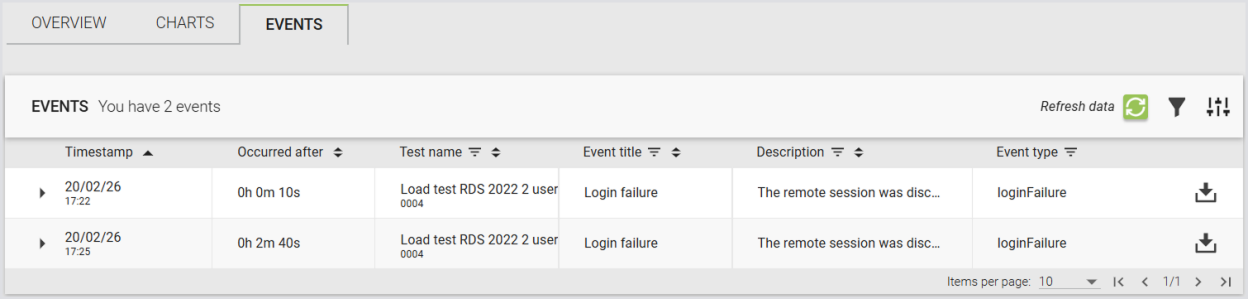
When to Use Custom Events

- Report internal states, checkpoints, or branching logic
- Flag warnings or known edge conditions in real time
- Log custom errors tied to a specific test session

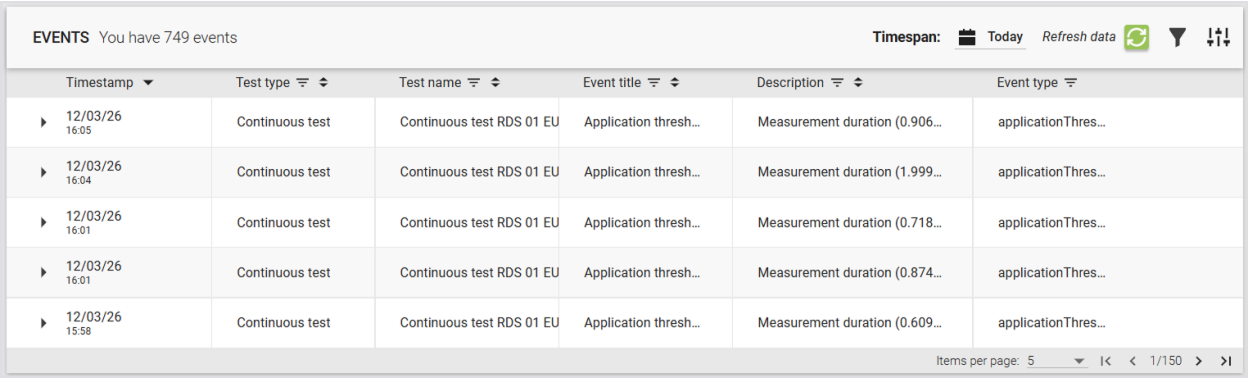
Custom events are purely informational and do not affect test execution or scoring.

Viewing Custom Events

1. The Events tab of a test run, alongside standard events like `SessionStart` and `SessionEnd`



2. The Events section of the Appliance UI, where they can be identified by their message text.



3. Launcher logs, for local debugging.

These events follow the same retry and buffering logic as standard events, so they won't be lost during temporary connection issues.

Enabling Custom Events

1. Ensure you are using Login Enterprise v6.2, which supports this feature.
2. Generate a Configuration-level Public API system access token (see Adding a System Access Token).
3. Include the Session ID in your Custom Connector invocation (refer to Passing Session ID to the Custom Connector).

No additional configuration is required on the Appliance or Launcher side.

Implementation

Custom Events are implemented as an API call (see Using the Public API). You must provide the User Session ID of the session you want to attach your event to, and a “description” field as a JSON payload.

The API is a REST method at `/v8-preview/user-sessions/{sessionId}/events`. It accepts POST invocations only, and expects a simple payload:

```
{
"description": "Event message content"
}
```

For instance, to post a Custom Event for a specific User Session from PowerShell, this command will work (as will the corresponding Invoke-RestMethod):

```
Invoke-WebRequest -Method POST `
-uri https://login-ent.loginvsi.com/publicApi/v8-preview/user-
sessions/$sessionId/events `
-H @{"accept" = "*/*"; "Content-Type" = "application/json"
"Authorization" = "Bearer zRZXRvJoAHM3tGiAiYpecKzTfcQN0lwZGr-GJ4XagzE"} `
-body '{ "description": "This is a custom event" }'
```

NOTE: You may need to tell PowerShell to ignore certificate errors if you are using a self-signed certificate (see this Public API example for how to do that).

To make this call from a Login Enterprise Stand-Alone Engine script, use this code:

```
using System.Net;
string appliance = "appliance-hostname";
string apiKey = "generated-API-key";
var client = new WebClient();
client.Headers.Add("accept", "*/*");
client.Headers.Add("Content-Type", "application/json");
client.Headers.Add("Authorization", $"Bearer {apiKey}");
var response = client.UploadString($"https://{appliance}/publicApi/v8-
preview/user-sessions/{sessionId}/events", "{ \"description\": \"This is a
custom event\" }");
}
}
```

To make this call from a Linux server or Linux-like environment, you can use this curl:

```
curl -X POST https://login-ent.loginvsi.com/publicApi/v8-preview/user-
sessions/$sessionid/events -H 'accept: */*' -H 'Content-Type:
application/json' -H "Authorization: Bearer $key" -d '{ "description": "This
is a custom event" }'
```

TIP: REST calls are standard, so you can implement this in any language you need to.

Notes and Considerations

- Custom events do not support attachments such as screenshots or logs.
- The feature is optional and backward compatible. If your environment doesn't use a supported Launcher version, the functionality will not be available.

- Use descriptive but concise messages. Overuse of verbose or excessive events may clutter logs.
- Your Custom Connector must already include the address for the Login Enterprise Appliance and the API Secret Key in the code, or you will need to add them as Custom Fields for all of your accounts and include the Custom Fields in your command-line invocation.

How Custom Events for Connectors and Launchers and Passing Session ID to the Custom Connector work together:

Pass `{sessionId}` in your Custom Connector command line; the Launcher substitutes the real session ID at runtime, allowing the connector to POST `{"description":"..."}` to `https://<appliance-FQDN>/publicApi/v8-preview/user-sessions/{sessionId}/events` so events appear attached to that test session in the Appliance Events feed.

Passing Session ID to the Custom Connector

Before Login Enterprise v.6.2, the session ID was only available inside the Launcher process. This enhancement ensures that the session ID is passed to the connector during its initialization.

How It Works

When you define a Custom Connector, there is a new variable available for command-line substitution: `{sessionid}`. Add it to your custom connector invocation, and you can use that parameter however you need to. You must make sure your custom connector is expecting and will accept this new parameter.

When the Launcher initiates a custom connector on the Appliance, it supplies the current session ID. This allows the connector to tag any custom events or log messages with the correct session so that they appear in the appropriate place in the Appliance UI.

Key Details

- **Accuracy**: The session ID is accurate and deterministic, ensuring that any custom events posted by the connector are matched to the correct session in the event list.
- **Non-breaking**: Existing custom connectors that do not use the session ID continue to function without any changes.

Example Connector Script

Here is a simple example script showing how to pass the session ID to a PowerShell script and use that script to post Custom Events. To use this script as a Custom Connector, set your Connection Command Line field to this:

```
powershell.exe -c C:\LoginVSI\Scripts\connector.ps1 {host} "{username}"
"{password}" "{domain}" "{resource}" "{email}" {sessionId}
```

The script simply receives these parameters, posts a custom event reporting the parameters passed, sleeps for 30 seconds, and posts a closing custom event. You would place your actual commands to perform the login in place of the `sleep 30`.

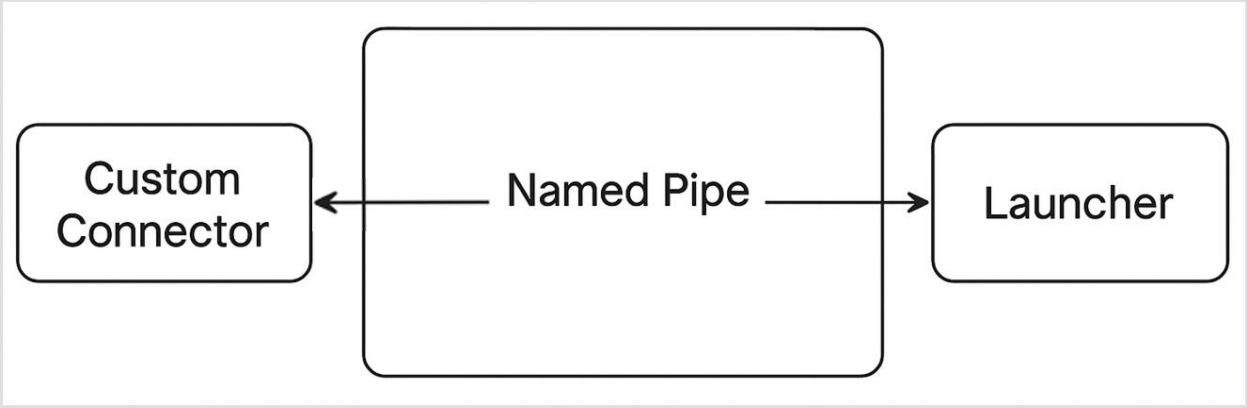
```
[CmdletBinding()]
param(
[string]$hostname,
[string]$username,
[string]$password,
[string]$domain,
[string]$resource,
[string]$email,
[string]$sessionid
)

$appliance = "your-appliance-FQDN"
$apikey = "your API secret key"

Invoke-WebRequest -Method POST `
-uri https://$appliance/publicApi/v8-preview/user-sessions/$sessionId/events
`
-H @{"accept" = "*/*"; "Content-Type" = "application/json"
"Authorization" = "Bearer $apikey"} `
-body "{ ""description"": ""Host $hostname; Username $username@$domain;
Resource $resource; Email $email; SessionID $sessionId"" }"

# perform the actual login process

sleep 30

Invoke-WebRequest -Method POST `
-uri https://$appliance/publicApi/v8-preview/user-sessions/$sessionId/events
`
-H @{"accept" = "*/*"; "Content-Type" = "application/json"
"Authorization" = "Bearer $apikey"} `
-body '{ "description": "This is a custom event injected at the end of the
test" }'
```

Named Pipes for Custom Connector

- Overview
- Pipe Channel Characteristics
- Benefits of Named Pipes
- Prerequisites
- How Named Pipes Work
- Custom Events and Session Information
- SDK Support
- From the Launcher Perspective (Named Pipe)
  - When Starting a New Session
  - When an Event Message is Received from the Connector Process
  - When the Custom Connector Doesn't Use the Pipe
- From the Connector Perspective (Named Pipe)
  - When the Launcher Starts the Connector Process
  - When the Custom Connector Starts
  - When the Remote Session Window is Created
  - When the Remote Window is Closed by the Connector
  - When a Connection Fails
  - Sending Custom Events
- From the User Session Perspective (Named Pipe)
  - Connector Sends RemoteConnectionWindowCreated
  - Connector Sends RemoteConnectionWindowClosed
  - Connector Sends RemoteConnectionFailure
  - Connector Sends a Custom Event
- Connector Named Pipe Schema
  - ConnectionConfiguration
  - CustomConnectorEvent
  - CustomConnectorEventType enum
- Minimal End-To-End Flow

Overview

This diagram illustrates the flow of messages between the Custom Connector and the Launcher using Named Pipes:



Pipe Channel Characteristics

- **Bidirectional**: Both the Launcher and the Connector can send and receive messages.
- **Isolated**: Each pipe is unique per user session.
- **Fast and reliable**: Memory-based transport ensures efficient data exchange.
- **Schema-driven**: Messages use JSON for consistency and validation.

Benefits of Named Pipes

Benefit	Description
Secure communication	Data is transmitted in memory (RAM) and scoped to the user session.
Bidirectional messaging	Both Launcher and Custom Connectors can send and receive messages.
Improved monitoring/logging	Admins can observe custom events without exposing sensitive data.
Faster reporting	Custom Connectors can report status/errors immediately.
Future expandability	Lays the foundation for advanced tracking/status reporting.

Prerequisites

- Windows environment
- A Launcher version that supports Named Pipes
- Custom Connector configured to leverage a data pipe

How Named Pipes Work

- Named Pipes are automatically created when a Custom Connector process starts.

- The Launcher stores the pipe name in an environment variable that's available to the entire user session, not to a specific process. Any process running in that session can use the pipe name if it has access to it, so the connection works on a first-come, first-served basis. After the connection is established, the two endpoints remain fixed and can't be changed.
- Messages transmitted through the pipe are structured (JSON) and handled internally by the Launcher.
- No command-line arguments are required, reducing exposure of sensitive information.

Custom Events and Session Information

Custom Connectors can send custom events to the Launcher via Named Pipes. These events allow the Launcher to track the state of a user session or report informational updates from the Connector.

Event Type	Description
`WindowCreated`	Signals that a new user session has begun.
`WindowClosed`	Signals that the user session has finished.
`ConnectionSuccess`	Reports that a connection attempt completed successfully.
`ConnectionFailure`	Reports that a connection attempt failed.
`InformationalEvents`	Optional messages such as login attempts, status updates, or session logs.

Key points:

- Events are informational and generally do not require action.
- Custom Connectors can report errors or session drops immediately without waiting for the Launcher or the Engine.
- Events are tied to the user session, allowing you to trace which account, test, or attempt raised each event.

SDK Support

A Custom Connector can implement Named Pipe communication manually or use the CustomConnector SDK included with the Launcher.

- The SDK is provided as a DLL in the extracted Launcher package.
- It exposes methods for sending Launcher events directly, without requiring custom pipe-handling code.
- The SDK can be used from PowerShell, C#, or other languages that can load the DLL.

For more information, see the Custom Connector SDK (Named Pipe).

From the Launcher Perspective (Named Pipe)

When Starting a New Session

When the Launcher starts a new session, it performs the following actions:

1. Starts `SessionTracker` (if configured) and `PipeServer`
2. Adds the `PipeName` to the **environment variables** of the connector process (environment variable name: `VSI_IPC_PIPE_NAME`)
3. Starts the custom connector process

The Launcher also starts background threads to manage the communication channel:

- `SessionTracker` thread starts looking for the session window
- Another thread waits until the client connects to the pipe
    - When the client connects, it sends `ConnectionConfiguration` to the client
- `PipeServer` thread waits for messages from the connector

When an Event Message is Received from the Connector Process

When the Launcher receives an event message from the connector process, it processes it based on the event type.

Event type	Description	Arguments	Launcher Behavior

`RemoteConnectionWindowCreated`	Signals the Launcher that the remote connection was successful and updates the internal state.	`ProcessId` (int): The ID of the remote session window.	When `ProcessId` is provided in the event: • If `SessionTracker` is disabled, the Launcher creates a new one and tracks the process with the given `ProcessId` • If `SessionTracker` didn't find the window yet, the Launcher replaces all configuration with `ProcessId` in the event and starts tracking that process. • If `SessionTracker` tracks the correct process, monitoring continues. • If `SessionTracker` tracks a different process, the Launcher overrides it using `ProcessId` from the event and starts tracking the new process.

`RemoteConnectionWindowClosed`	Sends a `ConnectionEnded` event to the appliance, but **only if** the event hasn't already been sent when `SessionTracker` detected that the window closed.	None	-
`RemoteConnectionFailure`	Sends a `ConnectionFailed` event to the Appliance, invalidating the session.	`Message` (string): The error message.	-
`Custom`	Sends a `CustomEvent` to the Appliance.	`Message` (string): Description of the custom event.	-

When the Custom Connector Doesn't Use the Pipe

Using the pipe is optional. If the custom connector doesn't interact with the pipe, the workflow continues to operate as it does.

- If `SessionTracker` is configured, the Launcher still attempts to track the session window using the existing mechanism.
- If the Launcher can't track the window, it falls back to the legacy behavior:
  - The Launcher waits for the configured timeout after the session ends.
  - The Launcher then sends the "connection closed" signal to the Appliance.

This ensures backward compatibility and prevents breaking existing custom connector implementations.

From the Connector Perspective (Named Pipe)

When the Launcher Starts the Connector Process

It adds `PipeName` to the connector's environment variables.

When the Custom Connector Starts

It should:

- Read the pipe name from the `VSI_IPC_PIPE_NAME` environment variable.
- Connect to the named pipe.
- (Optional) Read the connection configuration from the pipe.
    - Parameters can still be passed through the command line if preferred.

When the Remote Session Window is Created

The custom connector can report progress to the Launcher by sending a `RemoteConnectionWindowCreated` event. The event must include a string message with additional information about the event.

The event object also supports an optional `ProcessId` property, which represents the process ID of the created remote session window. The Launcher uses this process ID to track the window, even if the connector settings do not include session tracking configurations.

When the Remote Window is Closed by the Connector

It can send a `RemoteConnectionWindowClosed` event to the Launcher to indicate the end of the connection. The event must include a string message with additional information.

If the Launcher did not detect the window closure (i.e., it wasn't able to track it in the first place), this will generate a `ConnectionEnded` event in `LoginEnterprise` and move the session to its final state.

If the launcher is not tracking the window and the connector never sends this event, the system falls back to legacy behavior: it waits for the configured timeout and then sends the event.

When a Connection Fails

If something goes wrong during the connection process, the connector can send a `RemoteConnectionFailure` event with an error message. This notifies the Launcher that the connection has failed and generates a `ConnectionFailed` event in `LoginEnterprise`, moving the session to its final state.

If the failure occurs before the remote window is created and the connector does not send this event, the system falls back to legacy behavior: the user session will eventually be timed out by `LoginEnterprise`.

Sending Custom Events

At any time, the connector can send custom events to the launcher by sending a `CustomConnectorEvent` event with the type `Custom` and a string message. This generates a

custom user-session event in `LoginEnterprise`, which is visible on the All Events page, Dashboard, etc.

From the User Session Perspective (Named Pipe)

When the Launcher receives a new session request, it posts a `ConnectionStarted` event to the session. This moves the session login state to `InProgress`. Next, the Launcher initializes the `PipeServer` and `SessionTracker`, and then starts the custom connector process. At this point, it begins listening for event messages from the connector.

Connector Sends `RemoteConnectionWindowCreated`

This event does not produce any user-session events or state changes, but it signals the launcher that the window has been created and updates its internal state.

If the event includes a process ID for tracking, the Launcher starts a new, or restarts an existing, `SessionTracker` to monitor the remote window.

Connector Sends `RemoteConnectionWindowClosed`

This event signals the Launcher that the connection window has closed. If the Launcher was not already tracking the window, the event generates a `ConnectionEnded` user-session event. Based on the current session state, this moves the session to one of its final states:

State before the event		State after the event	
Login: `InProgress`	Session: `Undefined`	Login: `Failed`	Session: `Connected`
Login: `Succeded`	Session: `Running`	Login: `Succeded`	Session: `Faulted`
Login: `Succeded`	Session: `AbortRequested`	Login: `Succeded`	Session: `Aborted`
Login: `Succeded`	Session: `AbortConfirmed`	Login: `Succeded`	Session: `Aborted`
Login: `Succeded`	Session: `LoggingOff`	Login: `Succeded`	Session: `Completed`

If the connector does not send the event, but the Launcher is tracking the window, the Launcher raises the `ConnectionEnded` event when the tracked process closes.

If the connector does not send the event and the Launcher was unable to track the window, the `ConnectionEnded` event is sent only after the configurable timeout following the session's end, failure, or timeout.

Connector Sends `RemoteConnectionFailure`

This event signals the Launcher that a failure occurred during the connection process, so no remote window is created. The Launcher posts a `ConnectionFailed` event in `LoginEnterprise`, moving the session to its final state.

State before the event		State after the event	
Login: `InProgress`	Session: `Undefined`	Login: `Failed`	Session: `Abandoned`

If the connector does not send this event, the Launcher cannot detect the failure. In this case, `LoginEnterprise` times out the session (as it will stay too long in `LoginState.InProgress`). After the configured delay, the Launcher sends a `ConnectionEnded` event, which moves the session to the final faulted state.

Using this event allows a custom connector to notify the Launcher of a failure immediately, invalidating the session without waiting for a timeout.

Connector Sends a `Custom` Event

This event forwards a user-session custom event to `LoginEnterprise` without requiring the custom connector to post it manually through the Public API. The event is visible in Test results, Dashboards, Event pages, and the Event APIs.

Connector Named Pipe Schema

`ConnectionConfiguration`

After the Launcher detects a new client connection, it sends the following message containing all the information required to launch the remote connection:

```
{
  "sessionId": "<GUID>",
  "host": "<string>",
  "resource": "<string>",
  "username": "<string>",
  "password": "<string>",
  "domain": "<string>",
  "accountCustomFields": {
    "<string>": "<string>"
  },
  "accountSecureFields": {
    "<string>": "<string>"
  }
}
```

Field	Description

`SessionId`	Unique `LoginEnterprise` user session ID. Must be included in all events that the connector sends back.
`Host`	Target machine host (forwarded from the connector configurations).
`Resource`	Target resource (forwarded from the connector configurations).
`Username`	Username of the account selected for the session.
`Password`	Password of the account selected for the session.
`Domain`	Domain of the account selected for the session.
`AccountCustomFields`	Dictionary of custom fields of the account selected for the session.
`AccountSecureFields`	Dictionary of secure custom fields of the account selected for the session.

`CustomConnectorEvent`

The connector can send messages back to the Launcher through the named pipe.
Each message uses the following schema:

```
{
  "eventType": "<CustomConnectorEventType>",
  "message": "<string>",
  "processId": <int|null>
}
```

Field	Description
`EventType`	The type of the event being reported. Determines how the `Launcher` interprets the message and what `LoginEnterprise` event will be created.
`Message`	Human-readable description of the event.
`ProcessId`	(Optional) ID of the remote window process. Only used for `RemoteConnectionWindowCreated`, ignored for other types.

`CustomConnectorEventType` enum

The connector can send messages back to the Launcher through the named pipe.
Each message uses the following schema:

```
[
  "Custom",
  "RemoteConnectionWindowCreated",
  "RemoteConnectionFailure",
  "RemoteConnectionWindowClosed"
]
```

Event Type	Description	Expected Payload
`Custom`	Custom event.	`message`: required.
`RemoteConnectionWindowCreated`	Notifies Launcher that the connector successfully launched the remote client window.	`message`: optional details `processId`: optional ID of remote window process.
`RemoteConnectionFailure`	Reports a failure during connection and invalidates the session.	`message`: error reason, required.
`RemoteConnectionWindowClosed`	Notifies Launcher that the remote client process has exited and moves the session to the final state.	`message`: optional.

Minimal End-To-End Flow

Here's a simplified example demonstrating basic communication between the Launcher and the Connector using the named pipe.

A connector implemented in any language should roughly:

1. Read pipe name from environment variable `VSI_IPC_PIPE_NAME`
2. Connect to the named pipe as a client
3. Wrap the pipe in a UTF-8 text reader/writer (newline-based)
4. Read first line from the pipe > parse as `ConnectionConfiguration`
5. Use `ConnectionConfiguration` to start the remote client process
6. When the remote window is created, send:

```
CustomConnectorEvent {
  eventType: "RemoteConnectionWindowCreated",
  message: "...",
  processId: 1234
}
```

7. On connection failure, send:

```
CustomConnectorEvent {
  eventType: "RemoteConnectionFailure",
  message: "error details"
}
```

8. When the remote client process exits, send:

```
CustomConnectorEvent {
  eventType: "RemoteConnectionWindowClosed",
  message: "optional reason"
}
```

9. Close the pipe and exit.

Custom Connector SDK (Named Pipe)

- Overview
- SDK Contents
  - Typed Models
  - Utility Methods
- Example Usage (C#)
- Using the Pipe Without the SDK
  - 1. Getting the Pipe Name
  - 2. Event Messages (Newline-Delimited JSON)
  - 3. Launcher > Connector: ConnectionConfiguration
  - 4. Connector > Launcher: CustomConnectorEvent

Overview

The Launcher installer ships an optional Connector SDK (a .NET DLL) that you can import to simplify communication over the named pipe.

SDK Contents

Typed Models

For all messages exchanged with the Launcher (`ConnectionConfiguration`, `CustomConnectorEvent`).

`LauncherIpcClient` Client Functionality

- Pipe creation and connection
- Serialization and deserialization
- Message framing
- Error handling
- Automatic reading of `ConnectionConfiguration`
- Automatic event construction (`RemoteConnectionWindowCreated`, `RemoteConnectionWindowClosed`, `RemoteConnectionFailure`, `Custom`)

Utility Methods

Method	Description
`ConnectToLauncherAsync()`	Creates an instance of `LauncherIpcClient` connected to the named pipe.
`WaitForConnectionConfigurationAsync()`	Blocks execution until a `ConnectionConfiguration` message is received from the `Launcher`

`SendCustomEventAsync()`	Sends a custom user-session event to the `Launcher`
`SendRemoteConnectionWindowCreatedAsync()`	Notifies the `Launcher` that the remote connection window was created successfully (optionally sends the process ID for tracking)
`SendRemoteConnectionWindowClosedAsync()`	Notifies the `Launcher` that the remote connection window was closed
`SendRemoteConnectionFailureAsync()`	Notifies the `Launcher` that the connection failed

Example Usage (C#)

The following example demonstrates how to use the Connector SDK to communicate with the Launcher over a named pipe.

```
using Connectors;
using LoginEnterprise.Launcher.Connectors.Custom.Ipc.Client;
...
// create an instance of launcher client AND connects to the pipe
await using var launcherClient = await
LauncherIpcClient.ConnectToLauncherAsync();
try {
    // waits (blocks execution) until launcher sends the connection
configurations
        var config = await
launcherClient.WaitForConnectionConfigurationAsync();
        // send custom event
        await launcherClient.SendCustomEventAsync("starting new session");
        ...
        // starts remote connection
        ...
        // notifies the launcher the remote window is created
        await launcherClient.SendRemoteConnectionWindowCreatedAsync();
        ...
        // after the window is closed
        ...
    // notifies that the session is ended
    await launcherClient.SendRemoteConnectionWindowClosedAsync();
}
catch (Exception e){
    // notifies that something went wrong during the connection
    await launcherClient.SendRemoteConnectionFailureAsync(
            $"Connection failed: {e.Message}");
}
```

Using the Pipe Without the SDK

This section explains how to communicate with the Launcher over a named pipe without using the Connector SDK. It covers retrieving the pipe name, sending and receiving event messages, and implementing a minimal end-to-end flow.

1. Getting the Pipe Name

When the Launcher starts your connector, it exposes the pipe name through the `VSI_IPC_PIPE_NAME` environment variable. Your process should:

1. Read this environment variable.
2. Use the value to connect as a **named pipe client**.

Pseudo-code sample:

```
// get pipe name from env variables
pipeName = getenv("VSI_IPC_PIPE_NAME")
// use pipe name to connect to launcher
pipe = connect_to_named_pipe(pipeName)
// create write and read stream
reader = new TextReader(pipe, encoding=UTF8)
writer = new TextWriter(pipe, encoding=UTF8, autoFlush=true)
```

2. Event Messages (Newline-Delimited JSON)

Follow the format and structure of event messages exchanged between the Launcher and the Connector.

All messages over the pipe:

- are **UTF-8 encoded** text,
- contain a **single JSON object per line**, and
- are **terminated by a newline (`\n`)**.

Both the Launcher and the connector follow the same pattern:

- To **send** a message: serialize the JSON object to a string, then write it, followed by a newline.
- To **receive** a message: read a full line (until `\n`), then parse that line as JSON.

Pseudo-code sample:

```
// send message
jsonText = serialize_to_json(messageObject)
writer.write_line(jsonText) // appends '\n'
```

```
// receive message
line = reader.read_line() // reads until '\n'
messageObject = parse_json(line)
```

IMPORTANT: Do not send multiple JSON objects on the same line, and do not rely on length prefixes. Newlines are the only framing mechanism.

3. Launcher > Connector: `ConnectionConfiguration`

Once your connector is connected to the pipe, the **first** message the `Launcher` sends is always a `ConnectionConfiguration` JSON object:

```
{
  "sessionId": "<GUID>",
  "host": "<string>",
  "username": "<string>",
  "password": "<string>",
  "domain": "<string>",
  "accountCustomFields": {
    "<string>": "<string>"
  },
  "accountSecureFields": {
    "<string>": "<string>"
  }
}
```

Use this information to start your remote connection.

4. Connector > Launcher: `CustomConnectorEvent`

While handling the connection, you can report status back to the Launcher using `CustomConnectorEvent` messages:

```
{
  "eventType": "<CustomConnectorEventType>",
  "message": "<string>",
  "processId": <int|null>
}
```

Where `eventType` is one of:

```
[
  "Custom",
  "RemoteConnectionWindowCreated",
  "RemoteConnectionFailure",
  "RemoteConnectionWindowClosed"
]
```

Managing Session Metrics (Optional)

- Overview
- Why use Session Metrics?
- Tech tips
- Built-in Session Metrics
- Creating custom Session Metrics
  - Windows Performance Monitor (PerfMon)
    - Performance counter details
  - Windows Management Instrumentation (WMI)
    - Performance counter details
  - Creating a Session Metric
    - Creating a PerfMon Session Metric definition
    - Creating a WMI Session Metric definition
      - Adding a WMI measurement
  - Creating a Group for Session Metric definitions
  - Adding a Session Metric definition to a Group
    - Removing a Session Metric Definition from a Group
  - Linking a Group to a Test
- Viewing Session Metrics
- Additional resources

Overview

Session Metrics are Windows performance metric data that are collected by the Engine throughout a Test Run. They are user-defined, so you can collect any Windows counter that the Virtual User has access to. When the Test Run starts, the Engine collects all of these user-defined performance counters for the duration of the Test and reports them back to the Appliance as part of your Test results. Session Metrics are displayed in charts for Load Tests and Continuous Tests.

IMPORTANT: Charts only display metrics with the same unit (%, Ops/Sec, Bytes, etc.) simultaneously. This unit is set when you structure a Session Metric Definition for an API Request.

Administrators can collect additional performance metrics from Windows Performance Monitor (PerfMon) and Windows Management Instrumentation (WMI) to display with EUX Score and Application metrics. When enabled, Session Metrics are captured every 5 seconds by default.

An individual Session Metric is just the PerfMon or WMI counter you have defined. A Session Metric Group is an entity that organizes performance counters so they can be associated with a Test. You can link a Session Metric Group to a Test to specify performance counters for Load Tests and Continuous Tests. The Group can contain metrics sourced from both PerfMon and WMI if desired, allowing you to gather a unique dataset that meets your needs.

The system captures and displays the percentage of CPU and Memory Utilization by default. For details, see Built-In Session Metrics.

NOTE: You can use the Session Metric Library with examples of WMI and PerfMon Session Metric definitions. These are provided as examples only. Login VSI has limited ability to recommend particular metrics for data collection. The customer is responsible for understanding and defining metrics for their monitoring needs.

Why use Session Metrics?

Benefit	Description
Improved root cause analysis during Load Testing	Session Metrics allow administrators to collect performance data from target Windows VDI or PCs during load tests. This makes it easier to correlate degraded EUX Scores or reduced VSImax density with system metrics such as CPU, memory, disk utilization, and other performance counters.
Identify resource-heavy applications and processes	Session Metrics help pinpoint agents, applications, or processes that consume large amounts of system resources, allowing teams to identify factors impacting system performance and scalability.
Better visibility for optimization and capacity planning	With a complete view of the workspace under stress, teams can analyze detailed performance data, fine-tune optimization settings, and measure the impact of configuration or infrastructure changes.
Granular comparison between test results	When comparing two test runs, Session Metrics help identify detailed changes in key performance indicators, enabling teams to understand how updates or configuration changes affect resource consumption.
Continuous real-time monitoring	For Continuous Testing, Session Metrics can be collected in real time, 24/7, without requiring an additional agent, simplifying ongoing monitoring of infrastructure performance.
Early detection of infrastructure issues	By correlating performance counters such as CPU and memory utilization with EUX Score degradation, logon times, or application response timers, Session Metrics can serve as an early warning system for infrastructure problems.
Application performance tracking across versions	Session Metrics allow administrators to measure application-specific resource utilization across version updates and upgrades, helping evaluate performance changes introduced by new releases.
Visibility into GPU and graphics-intensive workloads	Using WMI metrics, Session Metrics provide visibility into GPU usage, supporting accurate testing and tuning of graphics-intensive workloads.

Monitoring of platform-specific protocols and video quality	Platform-specific metrics such as ICA/HDX and Blast protocol information can be correlated with collaboration workloads like Zoom or Teams to evaluate video quality metrics such as frames per second during capacity testing.

Tech tips

When you enable Session Metric collection using Custom Metrics Groups, note that adding additional metrics to a Group increases resource consumption. This may impact the expected EUX Score and VSImax results. For this reason, consider capturing a benchmark of your EUX and VSImax before enabling Session Metrics. This will help assess the impact of collecting the session metrics you're interested in. You can also change the way Session Metrics are gathered in the Test definition by setting the proportion of your end-user Test targets that will gather Session Metrics.

Before you create a WMI query, define the performance counters you want to collect. This will minimize query overhead. For better performance, choose PerfMon over WMI when a specific session metric is available in both sources.

Built-in Session Metrics

There are two built-in Session Metrics that Login Enterprise comes pre-configured with. These are "CPU utilization" and "Memory utilization". Both of these metrics are expressed as a percent.

- **CPU Utilization** is simply the PerfMon counter "Processor Information", "% Processor Utility", "_Total".
- **Memory Utilization** is a hybrid counter. There is no native PerfMon counter that expresses memory utilization as a percentage. We use "Memory", "Available MBytes" from PerfMon and use WMI to capture the query "SELECT Capacity FROM Win32_PhysicalMemory". The WMI query returns bytes, so we convert them to MB and calculate 100 * (capacity - available) / total to get the in-use percentage.

Creating custom Session Metrics

In evolving environments, the demand for customized session metrics becomes increasingly apparent. Customizing session metrics in Login Enterprise enables you to gather more detailed and personalized data, facilitating the generation of reports that are specific to your environment. For instance, in Continuous Testing scenarios, this capability provides real-time insights finely tuned to your unique setup. Leveraging these metrics lets you optimize testing processes and enhance the overall customer experience.

The first step you need to take is to define the performance counters you want to collect.

- To learn where to find performance counter details about PerfMon Session Metrics, see Performance counter details for a PerfMon Session Metric definition.

- To learn where to find performance counter details about WMI Session Metrics, see Performance counter details for a WMI Session Metric definition.

Windows Performance Monitor (PerfMon)

Performance counter details

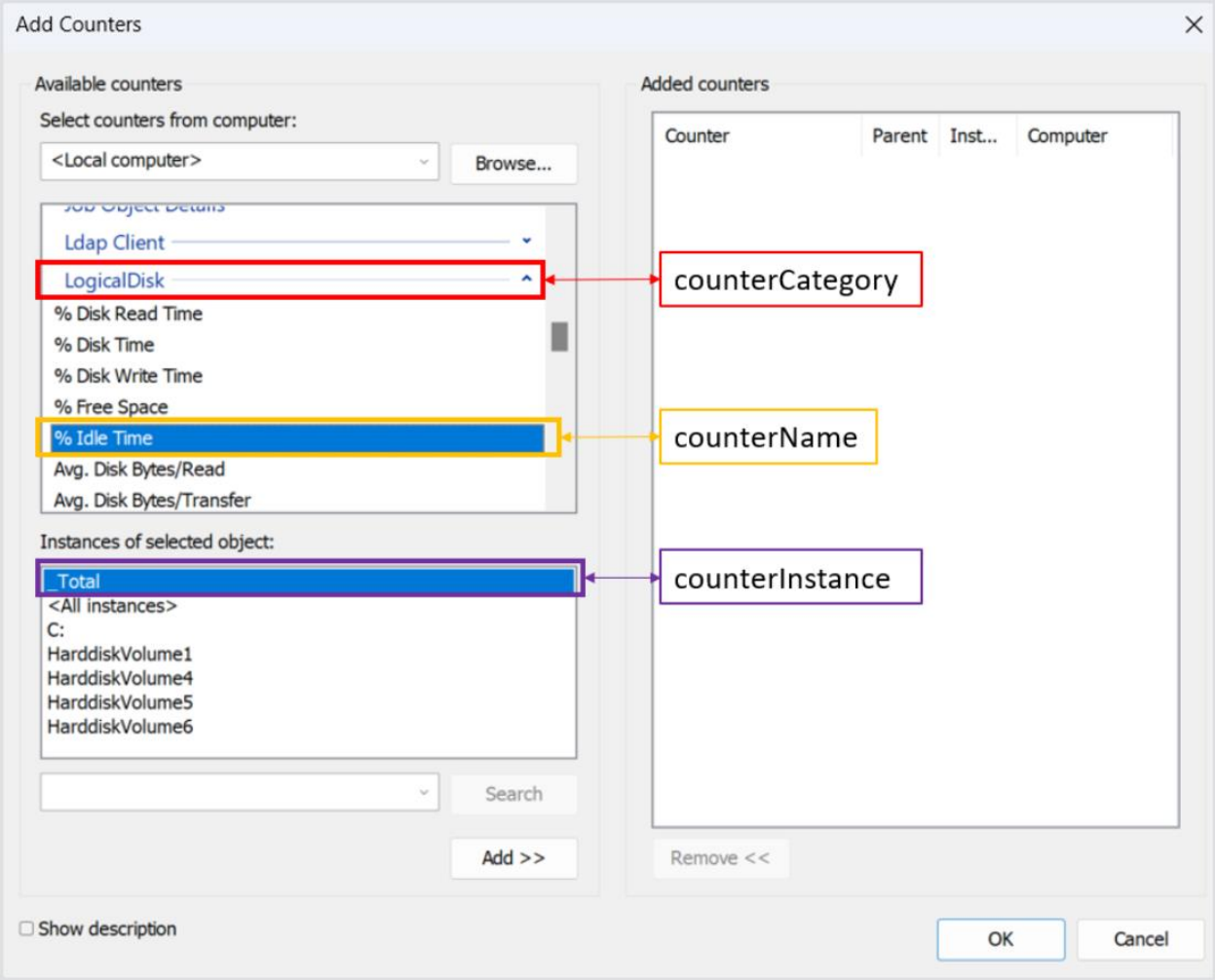
Windows Performance Monitor (PerfMon) is a Windows tool designed for analyzing the performance of applications and the overall system. It offers a variety of performance counters covering CPU usage, memory usage, disk activity, and network performance. Each counter will have details such as its name, description, and instance information.

The steps below explain how to find details about specific performance counters in PerfMon on your Windows system:

1. **Launch PerfMon**:
    - Open Performance Monitor on your Windows system. You can usually find it in the "Administrative Tools" or by searching for "perfmon" in the Start menu.
2. **Add the Desired Counter**:
    - In **PerfMon**, click "+" to add a performance counter.
    - Choose the performance object (category), counter (specific metric), and, if applicable, instance (e.g., for a specific process).

For example, suppose you want to monitor the "% Idle Time" performance counter for logical disks. This metric reports the percentage of time during the sample interval that the selected disk instance, or aggregate of all disk instances, was idle. In the following example, the red outline indicates the 'counterCategory,' the yellow outlines the 'counterName,' and the purple highlights the 'counterInstance'. Make sure you have copied the counterCategory, counterName, and counterInstance of your performance counter, for example, in Notepad as you will need it in further steps.

You can adapt the formula for other performance counters depending on your monitoring scenario.



Once you've identified the PerfMon performance counters you'd like to collect with Login Enterprise, you need to create and import your Session Metric definition.

To create a PerfMon Session Metric, see [Creating a PerfMon Session Metric definition](#).

Windows Management Instrumentation (WMI)

Performance counter details

Windows Management Instrumentation (WMI) is a powerful management framework in Windows for monitoring, configuration management, event handling, and remote administration of your system. WMI provides more detailed information than PerfMon, making it preferred in certain scenarios.

The steps below explain how to find details about WMI classes and their specific performance counters.

To identify available WMI performance counters, start with collecting available WMI classes. To do this, open the PowerShell console on your Windows system.

1. **Using the Start Menu:**
    - Click on the "**Start**" button.
    - Type "**PowerShell**" in the search bar.
    - Click on "**Windows PowerShell**" or "**PowerShell**" to open the PowerShell console.
2. **Using Run:**
    - Press **Win + R** to open the Run dialog.
    - Type "**powershell**" and press **Enter**.

To retrieve a list of all available WMI classes that can be queried, use the following command:

```
Get-WmiObject Win32_PerfFormattedData | Select-Object -Unique __Class |
Format-Table -AutoSize
```

To run the command, press **Enter**.

In this output, some examples of significant WMI classes related to performance monitoring include:

```
PS C:\Users\       > Get-WmiObject Win32_PerfFormattedData | Select-Object -Unique __CLASS | Format-Table -AutoSize

__CLASS
-------
Win32_PerfFormattedData_Counters_DNS64Global
Win32_PerfFormattedData_Counters_EventTracingforWindows
Win32_PerfFormattedData_Counters_HTTPService
Win32_PerfFormattedData_AFDCounters_MicrosoftWinsockBSP
Win32_PerfFormattedData_Counters_EventLog
Win32_PerfFormattedData_Counters_FileSystemDiskActivity
Win32_PerfFormattedData_Counters_GenericIKEv1AuthIPandIKEv2
Win32_PerfFormattedData_Counters_BluetoothDevice
Win32_PerfFormattedData_Counters_EventLogSubscriptions
Win32_PerfFormattedData_Counters_EventTracingforWindowsSession
Win32_PerfFormattedData_Counters_IPHTTPSGlobal
Win32_PerfFormattedData_Counters_HyperVVirtualMachineBus
Win32_PerfFormattedData_Counters_HTTPServiceRequestQueues
Win32_PerfFormattedData_Counters_BluetoothRadio
Win32_PerfFormattedData_Counters_HTTPServiceUrlGroups
Win32_PerfFormattedData_Counters_DCLocatorClient
Win32_PerfFormattedData_Counters_IPsecIKEv2IPv4
Win32_PerfFormattedData_Counters_IPsecIKEv2IPv6
Win32_PerfFormattedData_Counters_IPsecAuthIPIPv4
Win32_PerfFormattedData_Counters_IPsecAuthIPIPv6
Win32_PerfFormattedData_Counters_IPsecIKEv1IPv6
Win32_PerfFormattedData_Counters_IPsecIKEv1IPv4
Win32_PerfFormattedData_Counters_PhysicalNetworkInterfaceCardActivity
Win32_PerfFormattedData_Counters_PacerPipe
Win32_PerfFormattedData_Counters_SecurityPerProcessStatistics
Win32_PerfFormattedData_Counters_ProcessorInformation
Win32_PerfFormattedData_Counters_NtfsBucketizedPerformance
Win32_PerfFormattedData_Counters_PerProcessorNetworkActivityCycles
Win32_PerfFormattedData_Counters_PerProcessorNetworkInterfaceCardActivity
Win32_PerfFormattedData_Counters_StorportUnitQueue
Win32_PerfFormattedData_Counters_TeredoClient
Win32_PerfFormattedData_Counters_SecuritySystemWideStatistics
Win32_PerfFormattedData_Counters_SMBServer
Win32_PerfFormattedData_Counters_SMBServerShares
Win32_PerfFormattedData_Counters_TeredoServer
Win32_PerfFormattedData_GPUPerformanceCounters_GPUAdapterMemory
```

- **Win32_PerfFormattedData_PerfOS_Processor**
    - Provides processor performance information, including CPU usage, interrupts, and related statistics.
- **Win32_PerfFormattedData_PerfOS_Memory**
    - Offers performance data related to system memory usage, including available bytes, cache size, and memory-related rates.

- **Win32_PerfFormattedData_PerfDisk_LogicalDisk**
  - Provides performance data related to logical disks, including metrics for disk transfer rates, queue lengths, and average response times.
- **Win32_PerfFormattedData_GPUPerformanceCounters_GPUEngine**
  - Contains GPU engine performance data, offering metrics for GPU utilization, video decoding, and processing activity.
- **Win32_PerfFormattedData_PerfOS_System**
  - Offers performance data related to the overall system, including metrics such as context switches, system calls, and other counters.
- **Win32_PerfFormattedData_PerfProc_Process**
  - Provides performance data related to individual processes, including CPU and memory usage.
- **Win32_PerfFormattedData_Tcpip_NetworkAdapter**
  - Includes performance data related to network adapters, offering metrics such as bytes sent and received per second, packet rates, and error counts.
- **Win32_PerfFormattedData_Counters_VMwareBlastImagingCounters (specific to VMware Horizon only)**
  - Contains performance data related to VMware Blast Imaging, offering metrics for imaging operations and bandwidth usage in a VMware Horizon environment.

For example, suppose you want to query the PhysicalDisk class. This WMI class provides performance data specifically related to physical disk drives on a Windows system. It includes various performance counters that allow you to monitor the performance and activity of individual physical disk drives. You can collect such metrics as DiskReadsPerSec (the rate at which read operations are performed on the disk) and DiskWritesPerSec (the rate at which write operations are performed on the disk) among others.

To retrieve all counters for the WMI class (in this example, PhysicalDisk), run the following command:

```
Get-WmiObject -Query "SELECT * FROM Win32_PerfRawData_PerfDisk_PhysicalDisk"
```

The SELECT * part of the query specifies that you want to retrieve all properties or counters available for each instance of the PhysicalDisk class, so the output may be extensive.

```
PS C:\Users\       > Get-WmiObject -Query "SELECT * FROM Win32_PerfRawData_PerfDisk_PhysicalDisk"


__GENUS                      : 2
__CLASS                      : Win32_PerfRawData_PerfDisk_PhysicalDisk
__SUPERCLASS                 : Win32_PerfRawData
__DYNASTY                    : CIM_StatisticalInformation
__RELPATH                    : Win32_PerfRawData_PerfDisk_PhysicalDisk.Name="0 C:"
__PROPERTY_COUNT             : 40
__DERIVATION                 : {Win32_PerfRawData, Win32_Perf, CIM_StatisticalInformation}
__SERVER                     : DESKTOP-T1CC26R
__NAMESPACE                  : root\cimv2
__PATH                       : \\DESKTOP-T1CC26R\root\cimv2:Win32_PerfRawData_PerfDisk_PhysicalDisk.Name="0 C:"
AvgDiskBytesPerRead          : 701179298816
AvgDiskBytesPerRead_Base     : 22749568
AvgDiskBytesPerTransfer      : 916494615040
AvgDiskBytesPerTransfer_Base : 26415684
AvgDiskBytesPerWrite         : 215315316224
AvgDiskBytesPerWrite_Base    : 3666116
AvgDiskQueueLength           : 262489096260
AvgDiskReadQueueLength       : 227209894278
AvgDisksecPerRead            : 3871594886
AvgDisksecPerRead_Base       : 22749568
AvgDisksecPerTransfer        : 496091204
AvgDisksecPerTransfer_Base   : 26415684
AvgDisksecPerWrite           : 919463614
AvgDisksecPerWrite_Base      : 3666116
AvgDiskWriteQueueLength      : 35279201982
Caption                      :
CurrentDiskQueueLength       : 0
Description                  :
DiskBytesPersec              : 916494615040
DiskReadBytesPersec          : 701179298816
DiskReadsPersec              : 22749568
DiskTransfersPersec          : 26415684
DiskWriteBytesPersec         : 215315316224
DiskWritesPersec             : 3666116
Frequency_Object             : 0
Frequency_PerfTime           : 10000000
Frequency_Sys100NS           : 10000000
Name                         : 0 C:
PercentDiskReadTime          : 227209894278
PercentDiskReadTime_Base     : 133439353276946547
PercentDiskTime              : 262489096260
PercentDiskTime_Base         : 133439353276946547
PercentDiskWriteTime         : 35279201982
PercentDiskWriteTime_Base    : 133439353276946547
PercentIdleTime              : 825422009921
PercentIdleTime_Base         : 133439353276946547
SplitIOPerSec                : 258586
```

If you are looking for specific counters or properties, you can modify the query to include only the ones you are interested in. For example, suppose you want to collect the DiskWritesPerSec and DiskReadsPerSec metrics.

To retrieve specific counters (DiskWritesPerSec, DiskReadsPerSec), modify the command in the following way:

```
Get-WmiObject -Query "SELECT DiskWritesPerSec, DiskReadsPerSec FROM
Win32_PerfRawData_PerfDisk_PhysicalDisk"
```

```
PS C:\Users\       > Get-WmiObject -Query "SELECT DiskWritesPerSec, DiskReadsPerSec FROM Win32_PerfRawData_PerfDisk_PhysicalDisk"


__GENUS          : 2
__CLASS          : Win32_PerfRawData_PerfDisk_PhysicalDisk
__SUPERCLASS     :
__DYNASTY        :
__RELPATH        :
__PROPERTY_COUNT : 2
__DERIVATION     : {}
__SERVER         :
__NAMESPACE      :
__PATH           :
DiskReadsPersec  : 22806884
DiskWritesPersec : 3672495
PSComputerName   :

__GENUS          : 2
__CLASS          : Win32_PerfRawData_PerfDisk_PhysicalDisk
__SUPERCLASS     :
__DYNASTY        :
__RELPATH        :
__PROPERTY_COUNT : 2
__DERIVATION     : {}
__SERVER         :
__NAMESPACE      :
__PATH           :
DiskReadsPersec  : 22806884
DiskWritesPersec : 3672495
PSComputerName   :
```

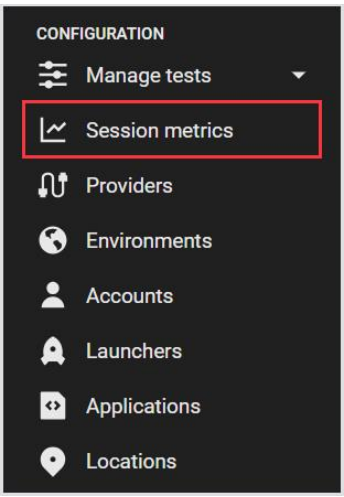
You can replace DiskWritesPerSec and DiskReadsPerSec with the specific counters you want to retrieve. By specifying only the desired properties in the SELECT statement, you focus the data returned and limit the performance overhead when polling the target system.

Once you've defined the performance counters for a WMI Session Metric definition, you need to create a WMI Session Metric definition.

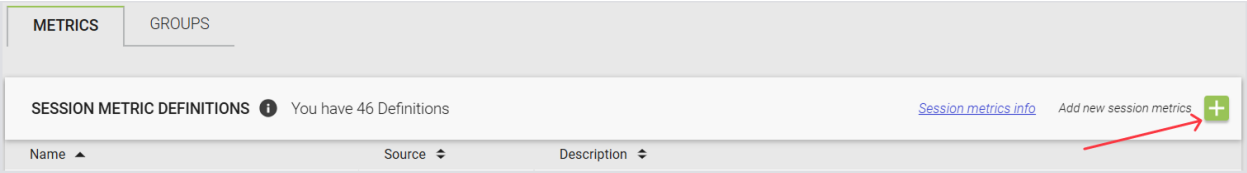
To create a WMI Session Metric, see Creating a WMI Session Metric definition.

Creating a Session Metric

1. In the **Login Enterpriseidebar menu**, go to **Configuration > Session metrics**.

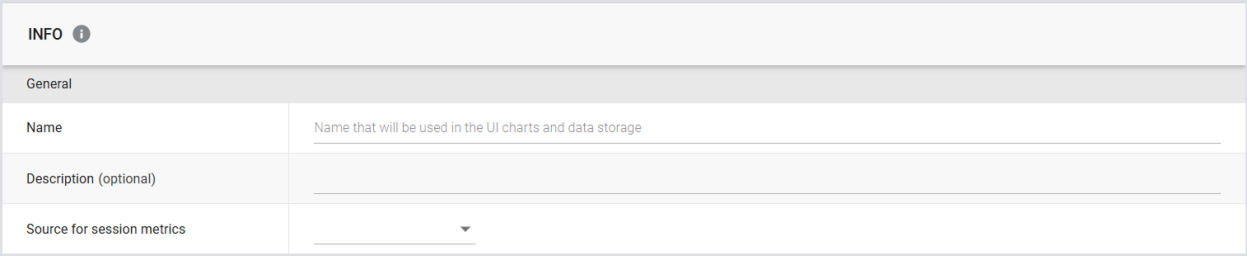


2. In Metrics, click the green "+" on the right to add a session metric. Note that the CPU and Memory utilization metrics are built-in (default ones) and cannot be deleted.



3. In the **Create new session metric** window, provide the general information for your session metric:

General	
Name	A unique descriptive name for the metric.
Description (Optional)	A brief description of the metric, if applicable.
Source for session metrics	The source you'd like to collect session metrics from: PerfMon or WMI.



Creating a PerfMon Session Metric definition

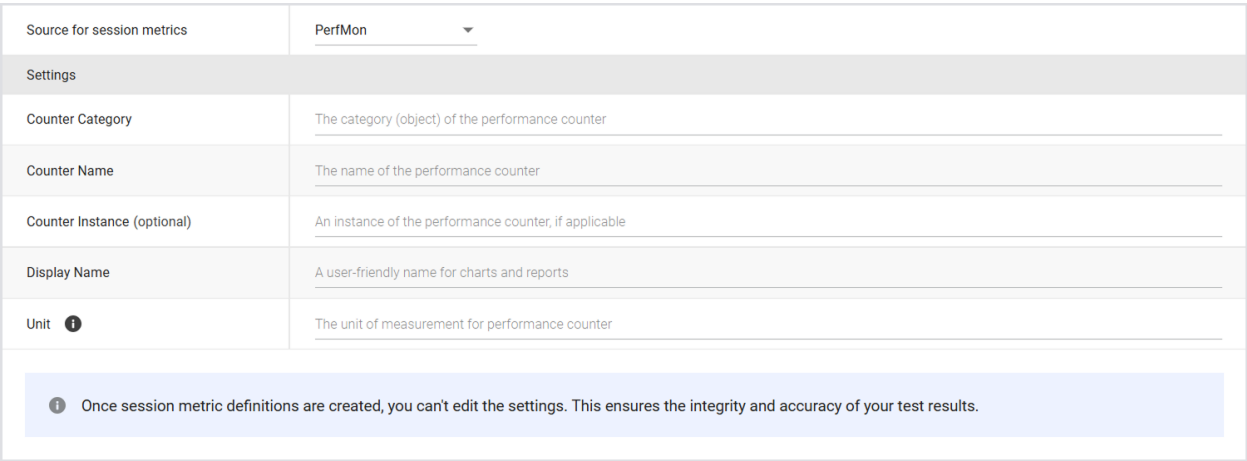
1. In **Settings**, provide the following information for a PerfMon session metric definition:

Settings	
Counter Category	The category (object) of the performance counter.
Counter Name	The name of the performance counter.
Counter Instance (optional)	An instance of the performance counter, if applicable.
Display Name	A user-friendly name for charts and reports.

Unit	The unit of measurement for the performance counter. This label appears on the y-axis for charts showing the metric. Note that charts only display metrics with the same unit simultaneously.

2. Save your settings to apply the changes. If you have more metrics to add, continue following the steps for each.

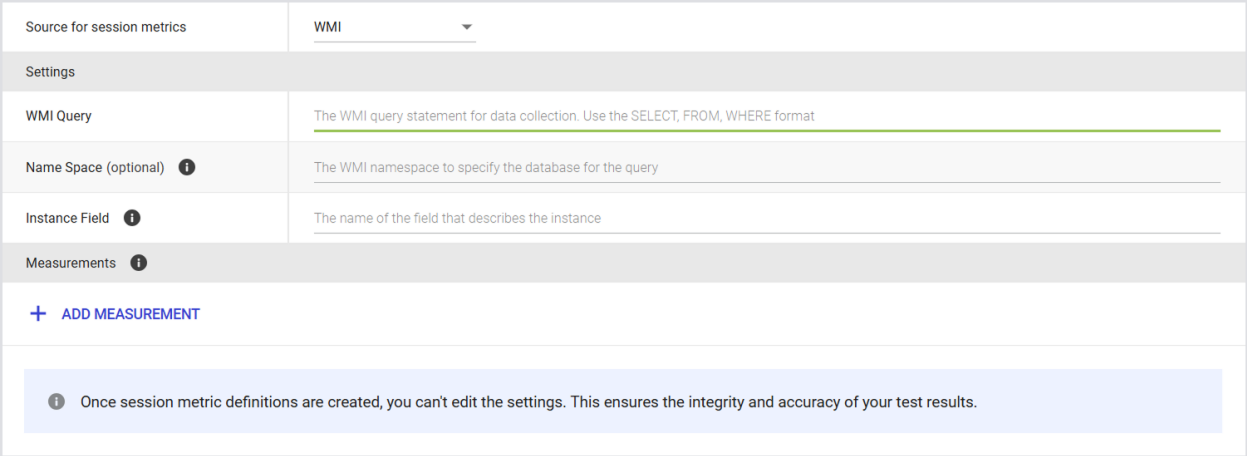
Once you've created a PerfMon Session Metric definition, you need to [create a Group for Session Metric definitions](#).



Creating a WMI Session Metric definition

In **Settings**, provide the following information for your WMI session metric definition:

Settings	
WMI Query	The WMI query statement for data collection. Use the SELECT, FROM, WHERE format.
Name Space	The namespace to specify the database for the query. When left blank, the default namespace, "root\cimv2" will be used.
Instance Field	The name of the field that describes the instance. Note this is NOT the actual instance name. For example, if you use '_Total' as an instance, you would reference the 'Name' column in the dataset listing the instances.



TIP: You can add measurements for monitoring system performance, troubleshooting, and establishing baselines. Measurements provide the data needed for performance analysis, capacity planning, and proactive issue identification. They enable alerts, trend analysis, and optimization, supporting effective system management and adherence to service level agreements (SLAs). For details, see Adding a Measurement.

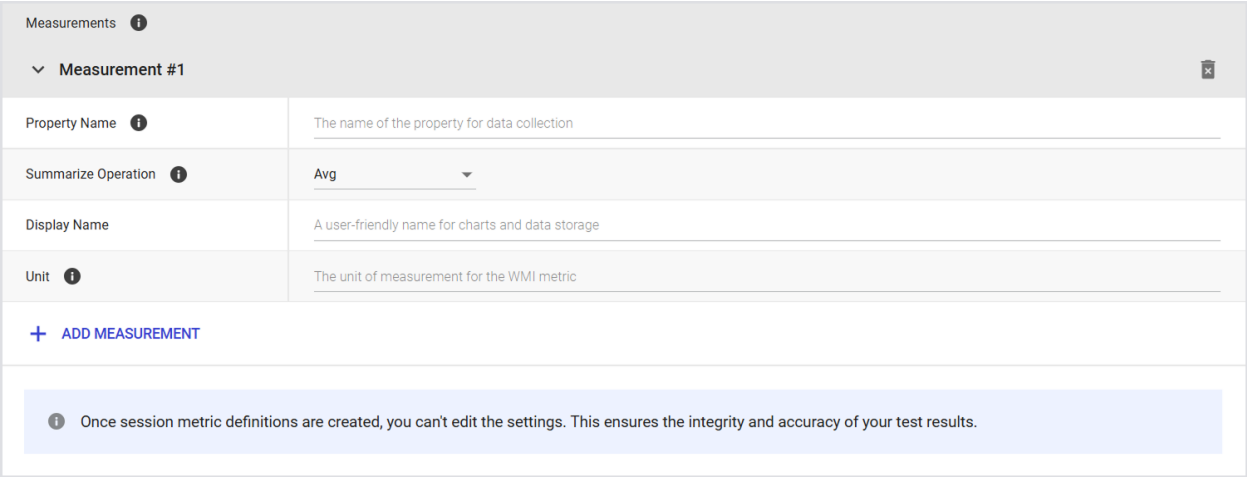
Adding a WMI measurement

1. To create a new measurement, click **Add Measurement** and provide the following information:

Measurements	
Property name	The name of the property for data collection.
Summarize operation	The summarization operation for the measurement, e.g., Min, Max, Avg, etc. If set to None, up to 16 instances will be returned in a random order.
Display name	A user-friendly name for charts and data storage.
Unit	The unit of measurement for the WMI metric. This label appears on the y-axis for charts showing the metric. Note that charts only display metrics with the same unit simultaneously.

2. Save your settings to apply the changes. If you have more metrics to add, continue following the steps for each.

Once you've created a WMI Session Metric definition, you need to create a Group for Session Metric definitions.

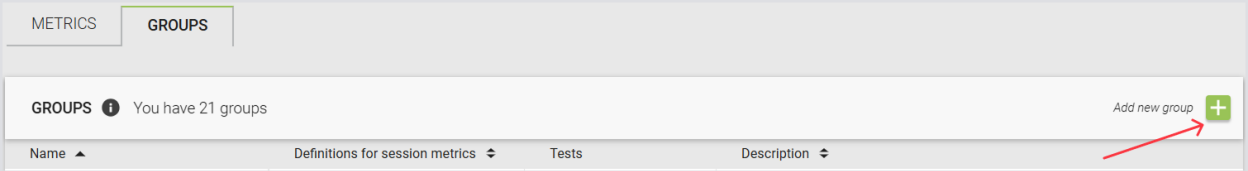


Creating a Group for Session Metric definitions

A Group is a collection of previously specified session metrics. Note that at the test level, you can connect only one Group. The Group can have as many session metric definitions as you would like, but the test can only contain one group.

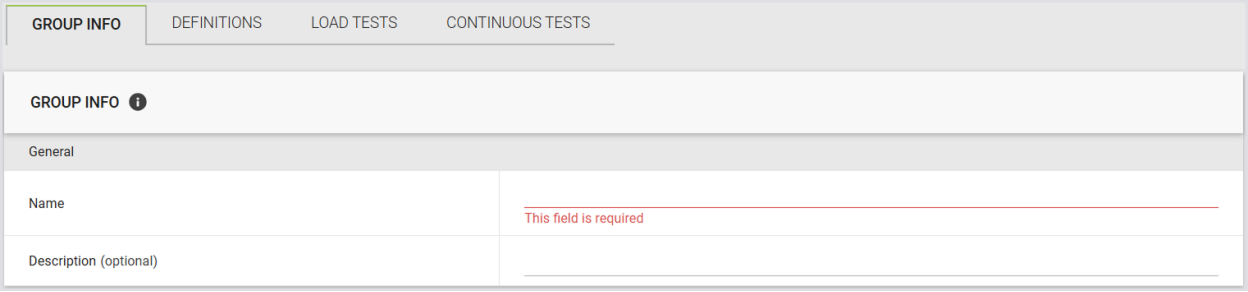
To create a new Group for Session Metric definitions:

1. In **Session metrics**, select **Groups** from the tab menu.
2. Click the green "+" on the right to add a Group.



3. In the **Add new group** window, provide the general information:

General	
Name	A name for the Group to easily identify and categorize related metrics.
Description (optional)	A brief description of the Group, if applicable.

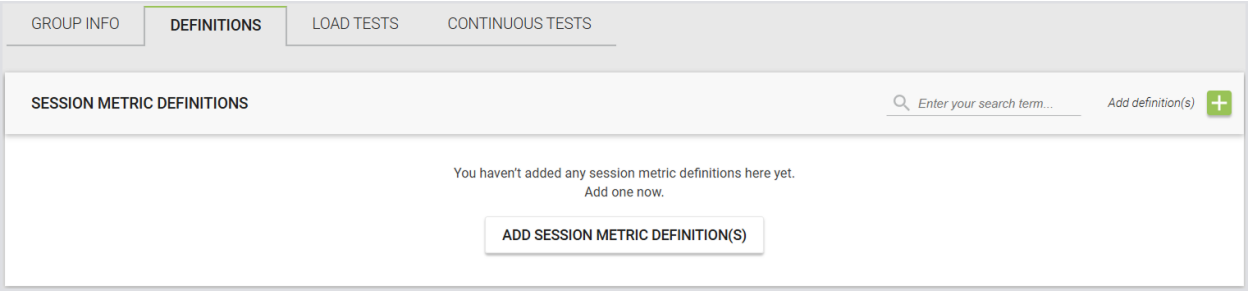


4. Save your settings to apply the changes. If you have more Groups to add, continue following the steps for each.

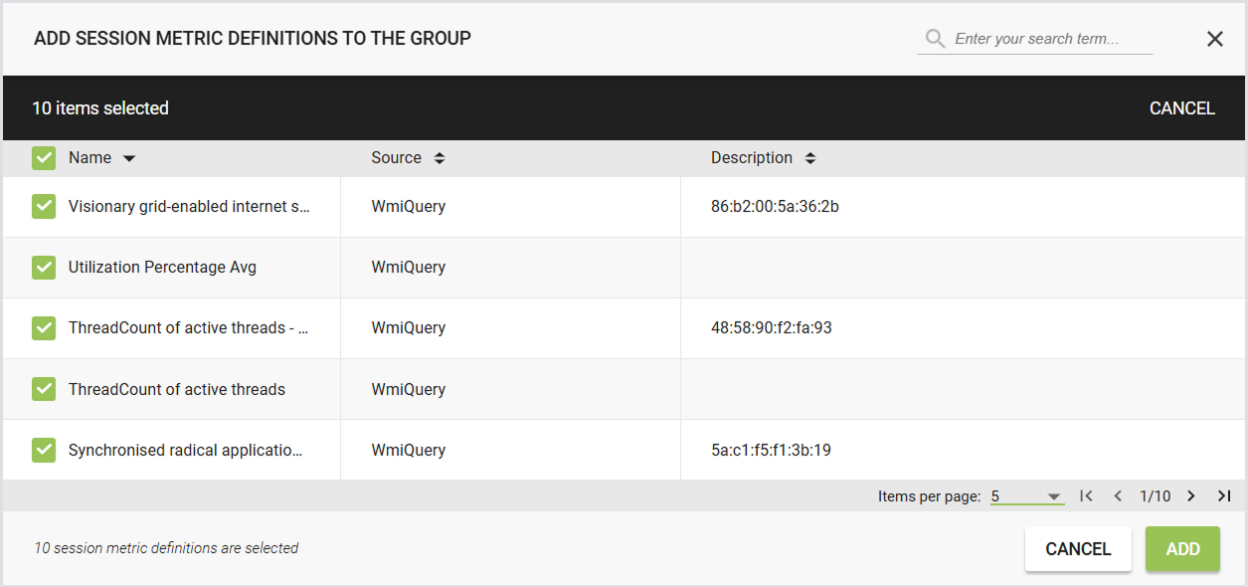
Once you've created a Group, you need to add Session Metric definitions to a Group.

Adding a Session Metric definition to a Group

1. In **Session Metric Groups**, select **Definitions** from the tab menu.
2. Click **Add session metric definitions**.



3. Select the checkboxes next to the Session Metric definitions to add them to a Group, and click **Add**. You can select individual definitions or select all by using the checkbox in the **Name** column.

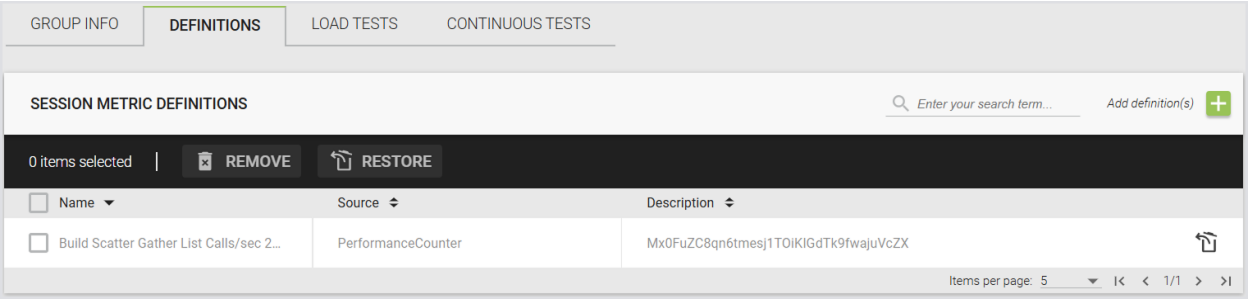


4. Click **Save**.

Once you've added Session Metric definitions to a Group, you need to link the Group to a Test. This way, the Test will capture the metrics specified in the Group.

Removing a Session Metric Definition from a Group

1. Go to your preferred Session Metric Group and select **Definitions** from the tab menu.
2. Select individual definitions or select all by using the checkbox in the **Name** column.
3. Click **Remove**. The removed definitions will remain in the list but appear greyed out, indicating they are marked for removal.



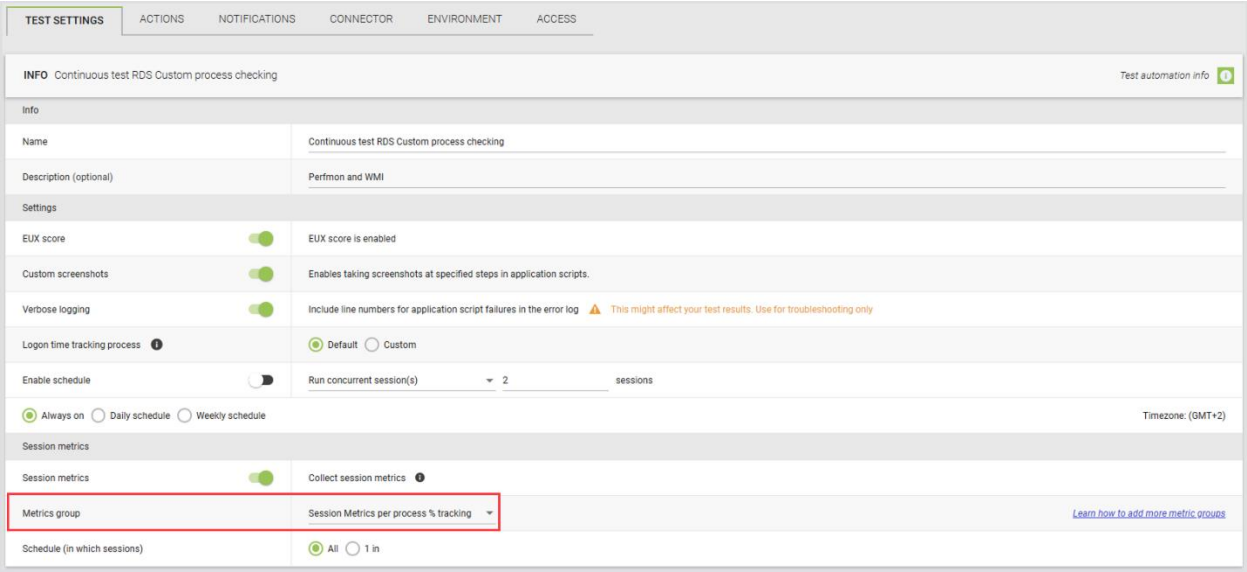
4. Click **Save** to apply the changes. The greyed-out rows will disappear once the changes are saved.

Linking a Group to a Test

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Manage tests**.
2. In **Manage tests**, select Continuous tests or Load tests from the dropdown.

NOTE: Session Metric Groups are supported only for Continuous tests and Load tests.

3. Click on the test.
4. In **Test settings** > **Session metrics > Metrics group**, select the Session Metric Group from the dropdown.



TIP: You can control the frequency of Session Metric polling to manage performance overhead. This feature allows you to specify whether metrics are collected from each session or only from 1 out of X sessions. It's a useful tool you can use to balance the need for metric gathering, particularly in medium to large-sized Session Metric Groups.

Viewing Session Metrics

For general guidelines on how to analyze your Continuous Testing and Load Testing Session Metrics, see Viewing Session Metrics.

Additional resources

- To learn how to export raw data, including Session Metrics, see, e.g., Exporting raw data for Continuous Tests.
- For more examples on Session Metrics, see Session Metrics Library.

Managing Session Metrics Using the Public API

- Structuring a Session Metric Definition for an API Request
  - PerfMon
  - WMI
- Creating a Session Metric Definition
- Creating a Group for Session Metric Definitions
- Adding a Session Metric Definition to a Group
- Adding Members to a Group

Structuring a Session Metric Definition for an API Request

PerfMon

To structure a PerfMon Session Metric definition, use a JSON schema. Start by defining a JSON schema that outlines the structure of a PerfMon Session Metric.

The following JSON schema consists of general properties like `type`, `name`, and `description`, along with a nested `measurement` object that specifies performance counter details.

```
{
  "type": "string",
  "name": "string",
  "description": "string",
  "tag": "string",
  "measurement": {
    "counterCategory": "string",
    "counterName": "string",
    "counterInstance": "string",
    "displayName": "string",
    "unit": "string"
  }
}
```

In this example:

- **"type"**: The type of the metric, e.g., "PerformanceCounter."
- **"name"**: The name of the metric as defined in its general settings.
- **"description"**: A short description of the metric, describing what it measures.
- **"tag"** (Optional): A label used for sorting, filtering, and searching in future Login Enterprise versions.
- **"measurement"**: An object containing properties specific to the performance counter:
  - **"counterCategory"**: The category of the performance counter.
  - **"counterName":** The specific counter within the category, e.g., "% Idle Time."
  - **"counterInstance"** (Optional): The instance of the counter, e.g., "_Total."
  - **"displayName":** A user-friendly name shown in charts.

- **"unit":** The unit of measurement, e.g., "%". This value is shown on the y-axis in charts. Only metrics with the same unit can be displayed on the same chart.

TIP: You can customize the values of these properties based on your specific use case or the actual performance metrics you are monitoring.

Now that you've structured your JSON object, you need to use it in an API request to create your Session Metric Definition. This way, you will be able to add a Session Metrics definition to a Session Metric Group and edit or delete it as necessary.

WMI

The following JSON schema consists of general properties, like `type`, `name`, and `description`, along with a nested `measurement` object that specifies WMI counter details.

```
{
  "type": "WmiQuery",
  "name": "string",
  "description": "string",
  "tag": "string",
  "wmiQuery": "string",
  "namespace": "string",
  "instanceField": "string",
  "measurements": [
    {
      "propertyName": "string",
      "summarizeOperation": "none",
      "displayName": "string",
      "unit": "string"
    }
  ]
}
```

In this example:

- **"type"**: The type of the metric, e.g., "WMIQuery."
- **"name"**: The name of the metric as defined in its general settings.
- **"description"**: A short description of the metric, describing what it measures.
- **"tag"** (Optional): A label used for sorting, filtering, and searching in future Login Enterprise versions.
- **"wmiQuery"**: WMI query string to retrieve the metrics. The string has to be in the SELECT, FROM, WHERE format.
- **"namespace"**: WMI namespace to specify the database for the query. If left blank, the default namespace "root\cimv2" will be used.
- **"instanceField"**: The field name describing the instance, not the actual instance name. For example, if you use '_Total' as an instance, you would reference the 'Name' column in the dataset listing the instances.
- **"measurement"**: An object containing properties specific to the performance counter:
  - **"propertyName"**: The property name corresponding to the WMI property.

- **"summarizeOperation"**: The summarization operation for the measurement. This way, you can aggregate multiple instances to generate results using operations such as min, max, avg, and sum. If the summarizeOperation is set to 'none,' up to 16 instances will be returned in a random order.
- **"displayName"**: A user-friendly name shown in charts.
- **"unit"**: The unit of measurement, e.g., "%". This value is shown on the y-axis in charts. Only metrics with the same unit can be displayed on the same chart.

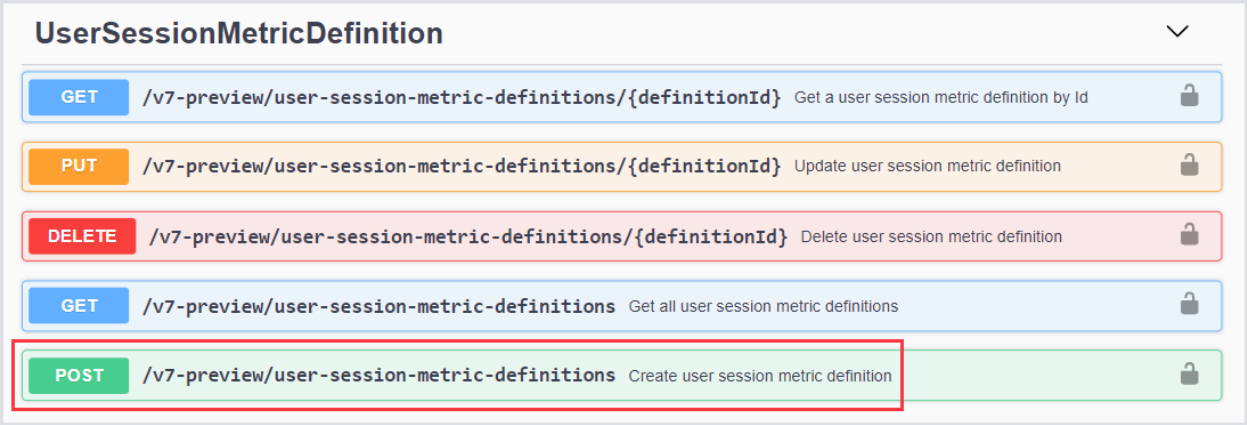
TIP: You can customize the values of these properties based on your specific use case or the actual performance metrics you are monitoring.

Now that you've structured your JSON object, you need to use it in an API request to create your Session Metric Definition. This way, you will be able to add a Session Metric Definition to a Session Metric Group and edit or delete it as necessary.

NOTE: The procedure for creating a WMI Session Metric definition in the Swagger API interface is the same as for creating a PerfMon Session Metric definition.

Creating a Session Metric Definition

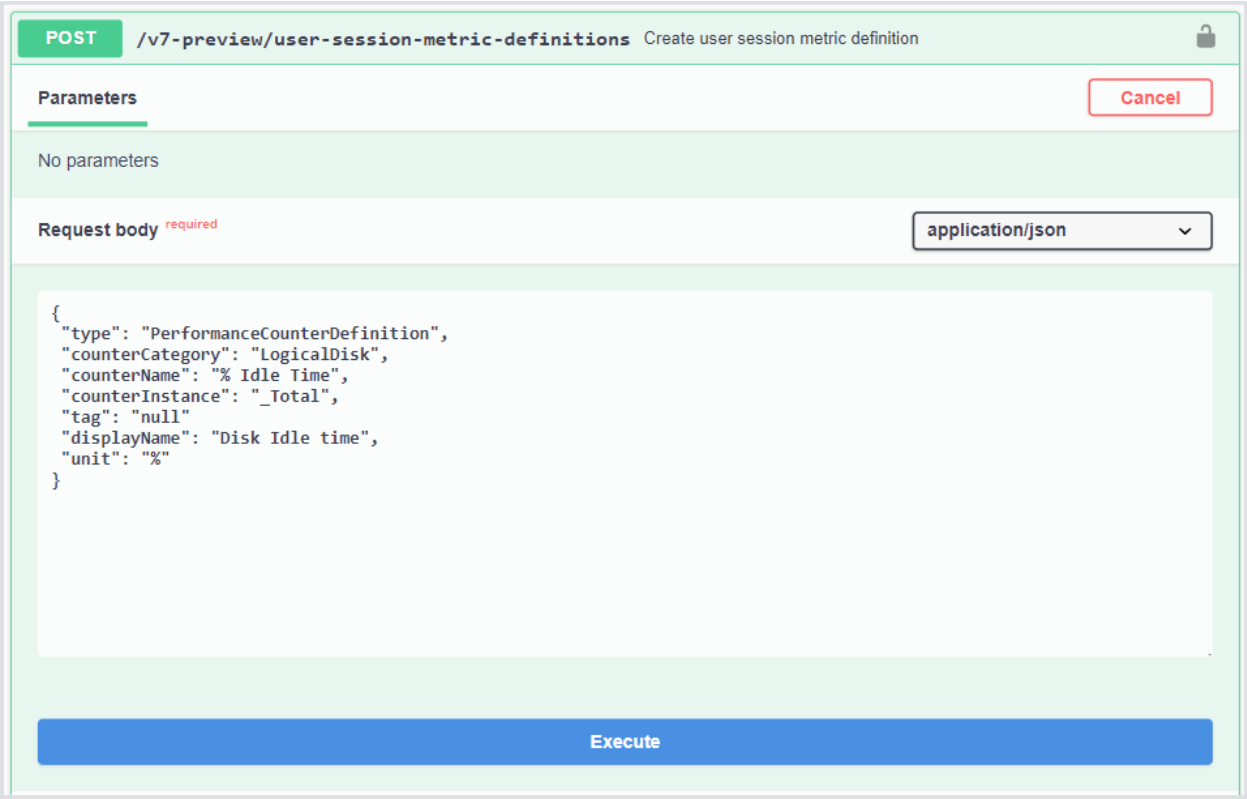
1. In the **Swagger UI console**, find **UserSessionMetricDefinition**
2. In **UserSessionMetricDefinition**, click **POST /v7/user-session-metric-definition**



3. In the top-right corner, click **Try it out** (The Request body field will open).

4. Paste the JSON object you've defined in [Structuring a PerfMon Session Metric Definition for an API Request] into the Request body field.
5. Click **Execute**.

If successful, the API response will return a key (ID) of a new Session Metric definition. Copy the ID to the clipboard, as you will need this ID when adding your Session Metric to a Session Metric Group.

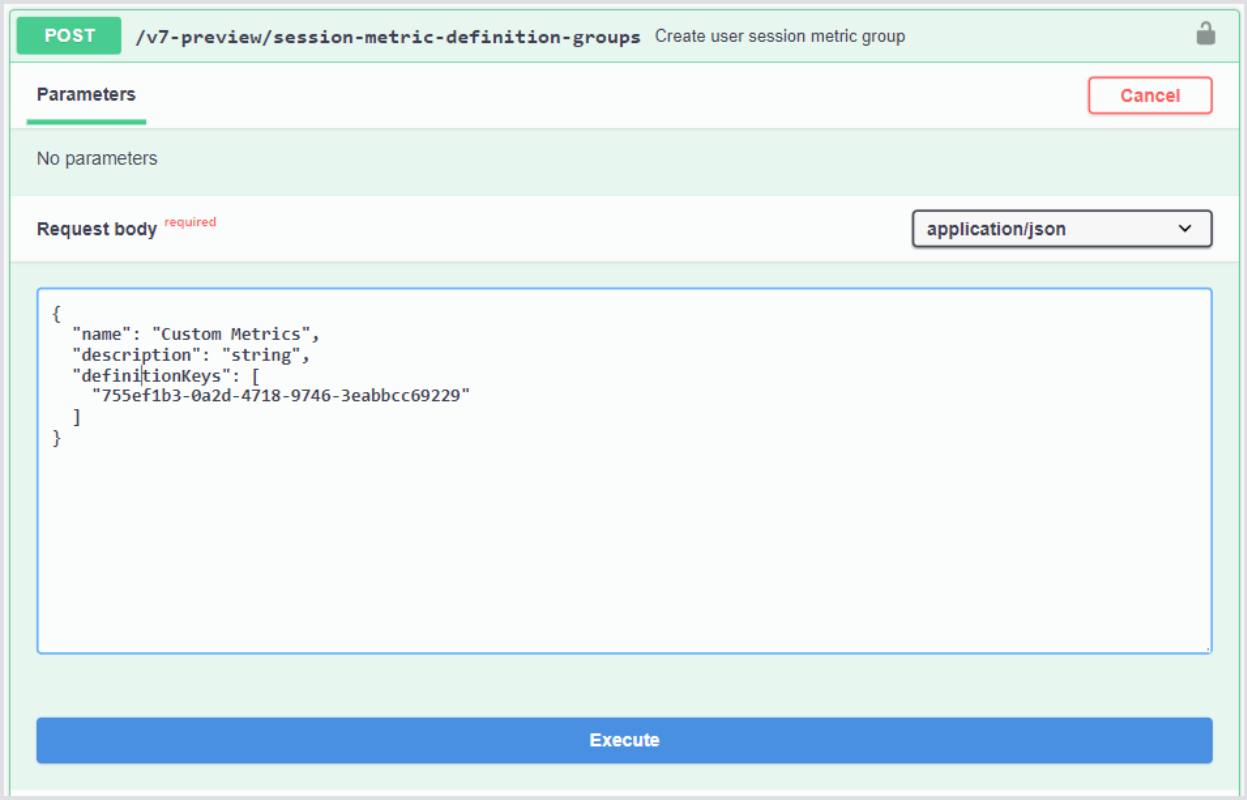


{

```
"id": "755ef1b3-0a2d-4718-9746-3eabbcc69229"
}
```

Creating a Group for Session Metric Definitions

1. In the **Swagger UI console**, find **SessionMetricDefinitionGroup**.
2. In **SessionMetricDefinitionGroup**, click **POST /v7/session-metric-definition-groups**.
3. In the top-right corner, click **Try it out** (The Request body field will open).
4. In the "**name**" parameter, specify the name of your Group, e.g., Custom Metrics.
5. The "**description**" field is optional. You can fill it out to provide additional information about the parameter, explaining that the custom metric group is a logical collection of related metrics that facilitates management and analysis.
6. In "**definitionKeys**", provide the ID of the Session Metric definition returned from the API request in Creating a Session Metric Definition.
7. Click **Execute**.



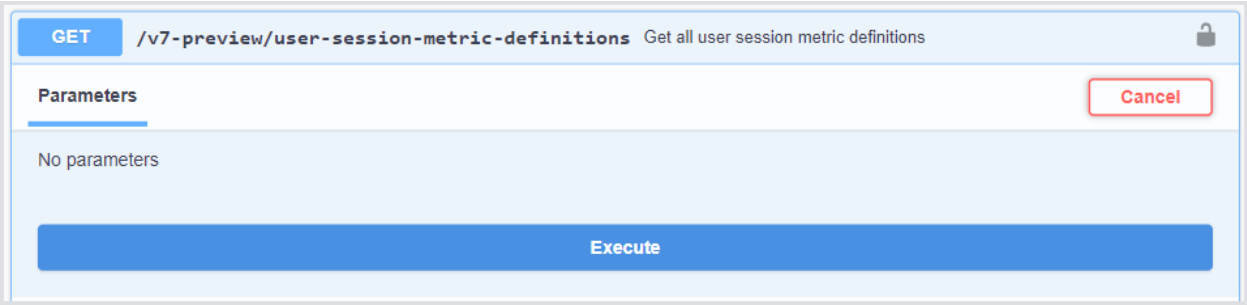
If successful, the API response will return the groupId of a new Session Metric Group. Copy the ID to the clipboard, as you will need this ID when adding members to a Session Metric Group.

```
{
"id": "a2f7d6d2-1dd8-4f42-a455-b741535527e1"
}
```

Adding a Session Metric Definition to a Group

To add Session Metric definitions, for example, CPU and Memory, to a Group:

1. In the **Swagger UI console**, find **UserSessionMetricDefinition**.
2. In **UserSessionMetricDefinition**, click **GET /v7/user-session-metric-definitions**.
3. In the top-right, click **Try it out**, and then **Execute**.



The API response will return a list of existing Session Metrics. The Default Session Metrics are shown as "BuiltIn":

```
{
"type": "BuiltIn",
"key": "12a1cbf2-cf51-ee11-92e0-000d3a1d9eab",
"metricId": 1,
"sessionMetricsType": "cpuUsage",
"displayName": "CPU",
"unit": "%"
},...
```

Adding Members to a Group

To add existing Session Metrics (Members) to a Group:

1. In the **Swagger UI console**, find **SessionMetricDefinitionGroupMembers**.
2. In **SessionMetricDefinitionGroupMembers**, click **POST /v7/session-metric-definition-groups/{group-id}/members**.
3. In the top-right, click **Try it out** (The Request body field will open).
4. In **groupId**, specify the ID of the Group returned from the API request in Creating a Group for Session Metric Definitions.
5. In the **Request body**, provide the "key" (ID) of each Session Metric you want to add to a Session Metric Group. To add, for example, only the "CPU" Session Metric, specify the "key" returned from the API request in Adding a Session Metric Definition to a Group.
6. Click **Execute**.

POST /v7-preview/session-metric-definition-groups/{groupId}/members Add members to session metric group

Parameters

Cancel

Name	Description
groupId * required string($uuid) (path)	a2f7d6d2-1dd8-4f42-a455-b741535527e1

Request body required

application/json

```
[
  "12a1cbf2-cf51-ee11-92e0-000d3a1d9eab",
  "13a1cbf2-cf51-ee11-92e0-000d3a1d9eab",
  "755ef1b3-0a2d-4718-9746-3eabbcc69229"
]
```

Execute

NOTE: Session Metric Groups can contain one or more desired performance counters.

Each Session Metric in the Group will be polled every 5 seconds. As you add more Metrics to a Group, the CPU overhead will increase correspondingly.

Using the Session Metrics Library

- Overview
- Session Metric Definitions
  - PerfMon
  - WMI
- PerfMon
  - Processor
    - % Processor Time
    - % Privileged Time
    - % User Time
    - % Interrupt Time
    - Interrupts/sec
  - Process
    - % Processor Time
    - % Privileged Time
    - % User Time
    - Handle Count
    - IO Data Bytes/sec
    - IO Read Bytes/sec
    - IO Write Bytes/sec
    - Working Set
  - System
    - Processor Queue Length
    - System Up Time
  - Memory
    - Available MBytes
    - Pages/sec
    - % Committed Bytes In Use
  - Paging File
    - % Usage
  - PhysicalDisk
    - % Disk Read Time
    - % Disk Time
    - % Disk Write Time
    - % Disk Idle Time
    - Current Disk Queue Length
    - Avg. Disk Sec/Read
    - Avg. Disk Sec/Write
    - Avg. Disk Sec/Transfer
    - Disk Reads/sec
    - Disk Writes/sec
    - Disk Bytes/sec
  - Network Interface
    - Bytes Received/sec

        - Bytes Sent/sec
        - Bytes Total/sec
        - Output Queue Length
        - Packets/sec
        - Packets Sent/sec
        - Packets Received/sec
- WMI
    - GPU Engine % Utilization
    - Citrix
        - Network Latency
        - ICA RTT
        - Output Bandwidth Used vs. Availability
        - Input Bandwidth Usage vs. Availability
    - Horizon
        - Blast Protocol Frames Per Second
        - Blast Bytes Transmitted vs. Bytes Received
        - PCoIP Round Trip Time

Overview

This section is a repository of Session Metrics Definitions that can be readily imported to Login Enterprise via the Public API. Each session metric definition below is presented in its JSON for this reason. Simply copy the desired JSON definition from below, modify the attributes as needed (Name, Display Name, Instance), and use it within the Swagger API interface.

NOTE: Using the Swagger API interface is out of scope. For details, see Managing Session Metrics Using the Public API.

Session Metric Definitions

PerfMon

- Processor
- Process
- System
- Memory
- Paging File
- Physical Disk
- Network Interface

WMI

- GPU Engine
- Citrix
- Horizon

PerfMon

Processor

% Processor Time

% Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then subtracting that value from 100%.

```
{
   "name": "Processor % Processor Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Processor",
      "counterName": "% Processor Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "% Processor Time"
   },
   "tag": "cpu"
}
```

% Privileged Time

% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data.

```
{
   "name": "Processor % Privileged Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Processor",
      "counterName": "% Privileged Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "% Privileged Time"
   },
"tag": "cpu"
}
```

% User Time

% User Time is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems.

```
{
   "name": "Processor % User Time",
```

```
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Processor",
      "counterName": "% User Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "% User Time"
   },
"tag": "cpu"
}
```

% Interrupt Time

% Interrupt Time is the time the processor spends receiving and servicing hardware interrupts during sample intervals. This value is an indirect indicator of the activity of devices that generate interrupts, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards and other peripheral devices.

```
{
   "name": "Processor % Interrupt Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Processor",
      "counterName": "% Interrupt Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "% Interrupt Time"
   },
"tag": "cpu"
}
```

Interrupts/sec

Interrupts/sec is the average rate, in incidents per second, at which the processor receives and serviced hardware interrupts. This value is an indirect indicator of the activity of devices that generate interruptions, such as the system clock, the mouse, disk drivers, data communication lines, network interface cards, and other peripheral devices.

```
{
   "name": "Processor Interrupts/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Processor",
      "counterName": "Interrupts/sec",
      "counterInstance": "_Total",
      "unit": "Interrupts/sec",
      "displayName": "Interrupts/sec"
   },
"tag": "cpu"
}
```

Process

% Processor Time

% Processor Time is the percentage of elapsed time that the processor spends to execute a non-Idle thread. It is calculated by measuring the percentage of time that the processor spends executing the idle thread and then subtracting that value from 100%.

```
{
   "name": "<PROCESS_HERE> % Processor Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "% Processor Time",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "%",
      "displayName": "<PROCESS_HERE> % Processor Time"
   },
   "tag": "process"
}
```

% Privileged Time

% Privileged Time is the percentage of elapsed time that the process threads spent executing code in privileged mode. When a Windows system service is called, the service will often run in privileged mode to gain access to system-private data.

```
{
   "name": "<PROCESS_HERE>% Privileged Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "% Privileged Time",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "%",
      "displayName": "<PROCESS_HERE> % Privileged Time"
   },
"tag": "process"
}
```

% User Time

% User Time is the percentage of elapsed time the processor spends in the user mode. User mode is a restricted processing mode designed for applications, environment subsystems, and integral subsystems.

```
{
   "name": "<PROCESS_HERE> % User Time",
   "type": "PerformanceCounter",
   "measurement": {
```

```
      "counterCategory": "Process",
      "counterName": "% User Time",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "%",
      "displayName": "<PROCESS_HERE> % User Time"
   },
"tag": "process"
}
```

Handle Count

The total number of handles currently open by this process. This number is equal to the sum of the handles currently open by each thread in this process.

```
{
   "name": "<PROCESS_HERE> Handle Count",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "Handle Count",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "Count",
      "displayName": "<PROCESS_HERE> Handle Count"
   },
"tag": "process"
}
```

IO Data Bytes/sec

The rate at which the process is reading and writing bytes in I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.

```
{
   "name": "<PROCESS_HERE> IO Data Bytes/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "IO Data Bytes/sec",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "Bytes",
      "displayName": "<PROCESS_HERE> IO Data Bytes/sec"
   },
"tag": "process"
}
```

IO Read Bytes/sec

The rate at which the process is reading bytes from I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.

```
{
```

```
   "name": "<PROCESS_HERE> IO Read Bytes/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "IO Read Bytes/sec",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "Bytes",
      "displayName": "<PROCESS_HERE> IO Read Bytes/sec"
   },
"tag": "process"
}
```

IO Write Bytes/sec

The rate at which the process is writing bytes to I/O operations. This counter counts all I/O activity generated by the process to include file, network, and device I/Os.

```
{
   "name": "<PROCESS_HERE> IO Write Bytes/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "IO Write Bytes/sec",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "Bytes",
      "displayName": "<PROCESS_HERE> IO Write Bytes/sec"
   },
"tag": "process"
}
```

Working Set

Working Set is the current size, in bytes, of the Working Set of this process. The Working Set is the set of memory pages touched recently by the threads in the process.

```
{
   "name": "<PROCESS_HERE> Working Set",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Process",
      "counterName": "Working Set",
      "counterInstance": "<PROCESS_HERE>",
      "unit": "Bytes",
      "displayName": "<PROCESS_HERE> Working Set"
   },
"tag": "process"
}
```

System

Processor Queue Length

The Processor performance object consists of counters that measure aspects of processor activity.

```
{
   "name": "Processor Queue Length",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "System",
      "counterName": "Processor Queue Length",
      "counterInstance": "",
      "unit": "Count",
      "displayName": "Processor Queue Length"
   },
"tag": "cpu"
}
```

System Up Time

System Up Time is the elapsed time (in seconds) that the computer has been running since it was last started. This counter displays the difference between the start time and the current time.

```
{
   "name": "System Up Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "System",
      "counterName": "System Up Time",
      "counterInstance": "",
      "unit": "Seconds",
      "displayName": "System Up Time"
   },
"tag": "system"
}
```

Memory

Available MBytes

Available MBytes is the amount of physical memory, in Megabytes, immediately available for allocation to a process or for system use. It is equal to the sum of memory assigned to the standby (cached), free, and zero-page lists.

```
{
   "name": "Memory Available MBytes",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Memory",
      "counterName": "AvailableMBytes",
```

```
      "counterInstance": "",
      "unit": "MB",
      "displayName": "Available MBytes"
   },
"tag": "memory"
}
```

Pages/sec

Pages/sec is the rate at which pages are read from or written to disk to resolve hard page faults. This counter is a primary indicator of the kinds of faults that cause system-wide delays.

```
{
   "name": "Memory Pages/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Memory",
      "counterName": "Pages/sec",
      "counterInstance": "",
      "unit": "Pages/sec",
      "displayName": "Pages/sec"
   },
"tag": "memory"
}
```

% Committed Bytes In Use

% Committed Bytes In Use is the ratio of Memory\\Committed Bytes to the Memory\\Commit Limit.

```
{
   "name": "Memory % Committed Bytes In Use",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Memory",
      "counterName": "% Committed Bytes In Use",
      "counterInstance": "",
      "unit": "%",
      "displayName": "% Committed Bytes In Use"
   },
"tag": "memory"
}
```

Paging File

% Usage

The amount of the Page File instance in use in percent.

```
{
   "name": "Paging File % Usage",
   "type": "PerformanceCounter",
```

```
   "measurement": {
      "counterCategory": "Paging File",
      "counterName": "% Usage",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "Paging File % Usage"
   },
"tag": "memory"
}
```

PhysicalDisk

% Disk Read Time

% Disk Read Time is the percentage of elapsed time that the selected disk drive was busy servicing read requests.

```
{
   "name": "PhysicalDisk % Disk Read Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "% Disk Read Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "PhysicalDisk % Disk Read Time"
   },
"tag": "disk"
}
```

% Disk Time

% Disk Time is the percentage of elapsed time that the selected disk drive was busy servicing read or write requests.

```
{
   "name": "PhysicalDisk % Disk Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "% Disk Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "PhysicalDisk % Disk Time"
   },
"tag": "disk"
}
```

% Disk Write Time

% Disk Write Time is the percentage of elapsed time that the selected disk drive was busy servicing write requests.

```
{
   "name": "PhysicalDisk % Disk Write Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "% Disk Write Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "PhysicalDisk % Disk Write Time"
   },
"tag": "disk"
}
```

% Disk Idle Time

% Idle Time reports the percentage of time during the sample interval that the disk was idle.

```
{
   "name": "PhysicalDisk % Disk Idle Time",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "% Disk Idle Time",
      "counterInstance": "_Total",
      "unit": "%",
      "displayName": "PhysicalDisk % Disk Idle Time"
   },
"tag": "disk"
}
```

Current Disk Queue Length

Current Disk Queue Length is the number of requests outstanding on the disk at the time the performance data is collected. It also includes requests in service at the time of the collection. This is an instantaneous snapshot, not an average over the time interval. For good performance, this difference should average less than two.

```
{
   "name": "PhysicalDisk Current Disk Queue Length",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Current Disk Queue Length",
      "counterInstance": "_Total",
      "unit": "Count",
      "displayName": "PhysicalDisk Current Disk Queue Length"
   },
"tag": "disk"
}
```

Avg. Disk Sec/Read

Avg. Disk sec/Read is the average time, in seconds, of a read of data from the disk.

```
{
   "name": "PhysicalDisk Avg. Disk sec/Read",
   "type": "PerformanceCounter",
      "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Avg. Disk sec/Read",
      "counterInstance": "_Total",
      "unit": "Seconds",
      "displayName": "PhysicalDisk Avg. Disk sec/Read"
   },
"tag": "disk"
}
```

Avg. Disk Sec/Write

Avg. Disk sec/Write is the average time, in seconds, of a write of data to the disk.

```
{
   "name": "PhysicalDisk Avg. Disk sec/Write",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Avg. Disk sec/Write",
      "counterInstance": "_Total",
      "unit": "Seconds",
      "displayName": "PhysicalDisk Avg. Disk sec/Write"
   },
"tag": "disk"
}
```

Avg. Disk Sec/Transfer

Avg. Disk sec/Transfer is the time, in seconds, of the average disk transfer.

```
{
   "name": "PhysicalDisk Avg. Disk sec/Transfer",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Avg. Disk sec/Transfer",
      "counterInstance": "_Total",
      "unit": "Seconds",
      "displayName": "PhysicalDisk Avg. Disk sec/Transfer"
   },
"tag": "disk"
}
```

Disk Reads/sec

Disk Reads/sec is the rate of read operations on the disk.

```
{
   "name": "PhysicalDisk Disk Reads/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Disk Reads/sec",
      "counterInstance": "_Total",
      "unit": "OPS",
      "displayName": "PhysicalDisk Disk Reads/sec"
   },
"tag": "disk"
}
```

Disk Writes/sec

Disk Writes/sec is the rate of write operations on the disk.

```
{
   "name": "PhysicalDisk Disk Writes/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Disk Writes/sec",
      "counterInstance": "_Total",
      "unit": "OPS",
      "displayName": "PhysicalDisk Disk Writes/sec"
   },
"tag": "disk"
}
```

Disk Bytes/sec

Disk Bytes/sec is the rate bytes are transferred to or from the disk during write or read operations.

```
{
   "name": "PhysicalDisk Disk Bytes/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "PhysicalDisk",
      "counterName": "Disk Bytes/sec",
      "counterInstance": "_Total",
      "unit": "Bytes",
      "displayName": "PhysicalDisk Disk Bytes/sec"
   },
"tag": "disk"
}
```

Network Interface

Bytes Received/sec

Bytes Received/sec is the rate at which bytes are received over each network adapter, including framing characters.

```
{
   "name": "Network Interface Bytes Received/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Bytes Received/sec",
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Bytes",
      "displayName": "Network Interface Bytes Received/sec"
   },
"tag": "network"
}
```

Bytes Sent/sec

Bytes Sent/sec is the rate at which bytes are sent over each network adapter, including framing characters.

```
{
   "name": "Network Interface Bytes Sent/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Bytes Sent/sec",
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Bytes",
      "displayName": "Network Interface Bytes Sent/sec"
   },
"tag": "network"
}
```

Bytes Total/sec

Bytes Total/sec is the rate at which bytes are sent and received over each network adapter, including framing characters. Network Interface\Bytes Total/sec is a sum of Network Interface\Bytes Received/sec and Network Interface\Bytes Sent/sec.

```
{
   "name": "Network Interface Bytes Total/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Bytes Total/sec",
```

```
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Bytes",
      "displayName": "Network Interface Total Bytes/sec"
   },
"tag": "network"
}
```

Output Queue Length

Output Queue Length is the length of the output packet queue (in packets). If this is longer than two, there are delays, and the bottleneck should be found and eliminated, if possible. Since the requests are queued by the Network Driver Interface Specification (NDIS) in this implementation, this will always be 0.

```
{
   "name": "Network Interface Output Queue Length",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Output Queue Length",
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Count",
      "displayName": "Network Interface Output Queue Length"
   },
"tag": "network"
}
```

Packets/sec

Packets/sec is the rate at which packets are sent and received on the network interface.

```
{
   "name": "Network Interface Packets/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Packets/sec",
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Packets",
      "displayName": "Network Interface Packets/sec"
   },
"tag": "network"
}
```

Packets Sent/sec

Packets Sent/sec is the rate at which packets are sent on the network interface.

```
{
   "name": "Network Interface Packets Sent/sec",
   "type": "PerformanceCounter",
```

```
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Packets Sent/sec",
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Packets",
      "displayName": "Network Interface Packets Sent/sec"
   },
"tag": "network"
}
```

Packets Received/sec

Packets Received/sec is the rate at which packets are received on the network interface.

```
{
   "name": "Network Interface Packets Received/sec",
   "type": "PerformanceCounter",
   "measurement": {
      "counterCategory": "Network Interface",
      "counterName": "Packets Received/sec",
      "counterInstance": "<INTERFACE_HERE>",
      "unit": "Packets",
      "displayName": "Network Interface Packets Received/sec"
   },
"tag": "network"
}
```

WMI

GPU Engine % Utilization

GPU Engine % Utilization is the proportion of time the GPU (Graphics Processing Unit) engine is actively processing data compared to its total available processing time. This metric indicates how much of the GPU's capabilities are being used over a given period, with higher percentages suggesting more intensive use of the GPU's processing power.

NOTE: The measurements below will capture the min, max, average, and total GPU % Utilization across all instances. Simply delete the measurements in the query below that you do not wish to collect.

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT UtilizationPercentage FROM
Win32_PerfFormattedData_GPUPerformanceCounters_GPUEngine",
   "namespace": "root\\cimv2",
   "instanceField": "Name",
   "measurements": [
   {
      "propertyName": "UtilizationPercentage",
      "summarizeOperation": "min",
      "displayName": "GPU Min Utilization %",
      "unit": "%"
```

```
    },
    {
       "propertyName": "UtilizationPercentage",
       "summarizeOperation": "max",
       "displayName": "GPU Max Utilization %",
       "unit": "%"
    },
    {
       "propertyName": "UtilizationPercentage",
       "summarizeOperation": "avg",
       "displayName": "GPU Avg Utilization %",
       "unit": "%"
    },
    {
       "propertyName": "UtilizationPercentage",
       "summarizeOperation": "sum",
       "displayName": "GPU Sum Utilization %",
       "unit": "%
    },
    {
       "propertyName": "UtilizationPercentage",
       "summarizeOperation": "none",
       "displayName": "GPU None Utilization %",
       "unit": "%"
    }
   ],
  "tag": "gpu"
}
```

Citrix

Network Latency

Network latency is the delay that occurs while transmitting data over the network.

```
{
    "type": "WmiQuery",
    "wmiQuery": "SELECT NetworkLatency FROM Citrix_Euem_RoundTrip",
    "namespace": "root\\Citrix\\EUEM",
    "instanceField": "InstanceId",
    "measurements": [
        {
            "propertyName": "NetworkLatency",
            "summarizeOperation": "none",
            "displayName": "Network Latency",
            "unit": "ms"
        }
    ],
    "tag": "citrix"
}
```

ICA RTT

ICA RTT is the time delay experienced between the user making an action, and the screen's graphical response. This includes the ICA Latency, and any processing required to deliver the application or desktop. This is an indicator of sluggishness.

```
{
    "type": "WmiQuery",
    "wmiQuery": "SELECT RoundtripTime FROM Citrix_Euem_RoundTrip",
    "namespace": "root\\Citrix\\EUEM",
    "instanceField": "InstanceId",
    "measurements": [
        {
            "propertyName": "RoundtripTime",
            "summarizeOperation": "none",
            "displayName": "ICA RTT",
            "unit": "ms"
        }
    ],
    "tag": "citrix"
}
```

Output Bandwidth Used vs. Availability

Output Bandwidth Used is the amount of data being transmitted from the Citrix VDI to the user's device to display sessions to users and support any other interaction channels like audio. Output Bandwidth Available is a measure of the total bandwidth available to transmit data from the Citrix VDI to the user's device. Comparing these two counters can help the admin determine whether output bandwidth is a bottleneck.

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT OutputBandwidthAvailable, OutputBandwidthUsed FROM
Citrix_Euem_RoundTrip",
   "namespace": "root\\Citrix\\EUEM",
   "instanceField": "InstanceId",
   "measurements": [
      {
         "propertyName": "OutputBandwidthAvailable",
         "summarizeOperation": "none",
         "displayName": "OutputBandwidthAvailable",
         "unit": "Mbps"
      },
      {
         "propertyName": "OutputBandwidthUsed",
         "summarizeOperation": "none",
         "displayName": "OutputBandwidthUsed",
         "unit": "Mbps"
      }
   ],
    "tag": "citrix"
```

```
}
```

Input Bandwidth Usage vs. Availability

Input Bandwidth Usage is the amount of data being transmitted from the user's device to Citrix VDI to transmit the user's inputs to sessions. Input Bandwidth Available is a measure of the total bandwidth available to transmit data from the user's device to Citrix VDI.

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT InputBandwidthAvailable, InputBandwidthUsed FROM
Citrix_Euem_RoundTrip",
   "namespace": "root\\Citrix\\EUEM",
   "instanceField": "InstanceId",
   "measurements": [
      {
         "propertyName": "InputBandwidthAvailable",
         "summarizeOperation": "none",
         "displayName": "InputBandwidthAvailable",
         "unit": "Mbps"
      },
      {
         "propertyName": "InputBandwidthUsed",
         "summarizeOperation": "none",
         "displayName": "InputBandwidthUsed",
         "unit": "Mbps"
      }
   ],
    "tag": "citrix"
}
```

TIP: If collecting all Citrix_Euem_RoundTrip counters, combine the WMI queries for optimized performance, as shown below:

```
{
    "type": "WmiQuery",
    "wmiQuery": "SELECT NetworkLatency, RoundtripTime,
InputBandwidthAvailable, InputBandwidthUsed, OutputBandwidthAvailable,
OutputBandwidthUsed FROM Citrix_Euem_RoundTrip",
    "namespace": "root\\Citrix\\EUEM",
    "instanceField": "InstanceId",
    "measurements": [
       {
           "propertyName": "NetworkLatency",
           "summarizeOperation": "none",
           "displayName": "Network Latency",
           "unit": "ms"
       },
       {
           "propertyName": "RoundtripTime",
           "summarizeOperation": "none",
           "displayName": "ICA RTT",
           "unit": "ms"
       },
```

```
        {
           "propertyName": "InputBandwidthAvailable",
           "summarizeOperation": "none",
           "displayName": "InputBandwidthAvailable",
           "unit": "Mbps"
        },
        {
           "propertyName": "InputBandwidthUsed",
           "summarizeOperation": "none",
           "displayName": "InputBandwidthUsed",
           "unit": "Mbps"
        },
        {
           "propertyName": "OutputBandwidthAvailable",
           "summarizeOperation": "none",
           "displayName": "OutputBandwidthAvailable",
           "unit": "Mbps"
        },
        {
           "propertyName": "OutputBandwidthUsed",
           "summarizeOperation": "none",
           "displayName": "OutputBandwidthUsed",
           "unit": "Mbps"
        }
     ],
     "tag": "citrix"
}
```

Horizon

Blast Protocol Frames Per Second

Frames Per Second measures the number of frames that are transmitted every second from a virtual desktop to the Horizon client device, using the VMware Blast display protocol. Within Horizon environments, this allows for assessing the visual performance and smoothness of the session-- a crucial component in understanding the end-user experience.

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT Framespersecond FROM
Win32_PerfFormattedData_Counters_VMwareBlastImagingCounters",
   "namespace": "root\\cimv2",
   "instanceField": "Name",
   "measurements": [
   {
      "propertyName": "Framespersecond",
      "summarizeOperation": "none",
      "displayName": "Frames Per Second None",
      "unit": "Frames Per Second"
   }
   ],
 "tag": "horizon"
}
```

Blast Bytes Transmitted vs. Bytes Received

Bytes Transmitted reflects the total number of bytes for imaging data that have been transmitted for a Blast session (server -> client), while Received Bytes reflect the number of bytes for imaging data that have been received for a Blast session (client -> server).

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT ReceivedBytes, TransmittedBytes FROM
Win32_PerfFormattedData_Counters_VMwareBlastImagingCounters",
   "namespace": "root\\cimv2",
   "instanceField": "Name",
   "measurements": [
   {
      "propertyName": "ReceivedBytes",
      "summarizeOperation": "none",
      "displayName": "Received Bytes",
      "unit": "Bytes"
   },
   {
      "propertyName": "TransmittedBytes",
      "summarizeOperation": "none",
      "displayName": "Transmitted Bytes",
      "unit": "Bytes"
   }
   ],
 "tag": "horizon"
}
```

TIP: If collecting all Win32_PerfFormattedData_Counters_VMwareBlastImagingCounters counters, combine the WMI queries for optimized performance, as shown below:

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT Framespersecond, ReceivedBytes, TransmittedBytes FROM
Win32_PerfFormattedData_Counters_VMwareBlastImagingCounters",
   "namespace": "root\\cimv2",
   "instanceField": "Name",
   "measurements": [
   {
      "propertyName": "Framespersecond",
      "summarizeOperation": "none",
      "displayName": "Frames Per Second None",
      "unit": "Frames Per Second"
   },
   {
      "propertyName": "ReceivedBytes",
      "summarizeOperation": "none",
      "displayName": "Received Bytes",
      "unit": "Bytes"
   },
   {
      "propertyName": "TransmittedBytes",
```

```
      "summarizeOperation": "none",
      "displayName": "Transmitted Bytes",
      "unit": "Bytes"
   }
   ],
 "tag": "horizon"

}
```

PCoIP Round Trip Time

This is the round trip latency in milliseconds between the Horizon Agent and the Horizon Client, i.e. the Launcher and the Horizon session.

```
{
   "type": "WmiQuery",
   "wmiQuery": "SELECT RoundTripLatencyms FROM
Win32_PerfRawData_TeradiciPerf_PCoIPSessionNetworkStatistics",
   "namespace": "root\\cimv2",
   "instanceField": "Name",
   "measurements": [
   {
      "propertyName": "RoundTripLatencyms",
      "summarizeOperation": "none",
      "displayName": "PCoIP RTT",
      "unit": "ms"
   }
   ],
   "tag": "horizon"
}
```

Using the Session Metrics Import Script (Optional)

- Overview
- Example usage
- Code explanation
- Example output
- Next steps

Overview

Session Metrics can be captured from both Windows Performance Monitor and Windows Management Instrumentation. You can add these metrics both via the API and the Login Enterprise user interface. For more information, see Session Metrics.

To automate the process of adding Session Metric definitions to your Login Enterprise Virtual Appliance via the API, you can use the Session Metrics import script. The script outputs the name and ID of each created Session Metric, aimed at improving ease of use for those relying on manual efforts within the Swagger API interface.

Example usage

Many IT administrators rely on Performance Monitor or WMI performance counters to analyze the performance of their virtual desktop and application environments effectively. Different scenarios and analyses may require different performance counters.

For example, when conducting Load Testing, metrics such as Disk Queue Length, Error Rates, and Throughput are crucial for assessing performance before pushing a build to production. In contrast, during Continuous Testing, metrics like CPU, Memory, and Disk IOPS can provide valuable insights into user experience.

As an IT administrator using Login Enterprise to analyze performance and user experience, you might consider creating a Session Metrics Group for each analysis. If you prefer to do this using the Swagger API interface, you will need a Session Metric import script.

You can download the `sessionMetricDefinitions.txt` script from here, and the `ImportSessionMetrics.ps1` script from here.

This script allows administrators to define and import all their metrics at once or in subsets. The output provides IDs that should be added when creating Session Metric Groups. For maximum flexibility, the script does not create or modify existing Session Metric Groups.

```
.\ImportSessionMetrics.ps1 -filePath "\pathToMyJSON.txt" -fqdn
"myappliance.contoso.com" -token "nsc24-
lfcCjkYwRRq1VmPiWrevWiXVe29TTabPTlCgE"
```

IMPORTANT: Because this functionality is only available in the v7-preview API, this script is subject to breaking changes. The script may or may not be updated to handle those breaking changes at the moment.

Code explanation

The script accepts three parameters:

Parameter	Description
FilePath	The path to a file containing valid JSON objects, defining Session Metrics to add.
FQDN	The URL of your Login Enterprise Virtual Appliance, excluding the protocol and API path.
Token	A Configuration-level API access token created on your Virtual Appliance.

First, the script starts by getting the raw content of the provided text file. The text file must be in proper JSON format. Please see the attached sessionMetricDefinitions.txt for an example of the schema.

For each JSON object in the provided text file, the raw JSON is extracted and used as the request body when performing a POST request with the API to create the Session Metric definition.

After each metric definition is created, there is additional logic to extract the display name and the ID of the newly created Session Metric for output to simplify the group creation process.

Example output

The script will output the display name and ID of the newly created Session Metrics. This output helps simplify the process of creating a Session Metric Group.

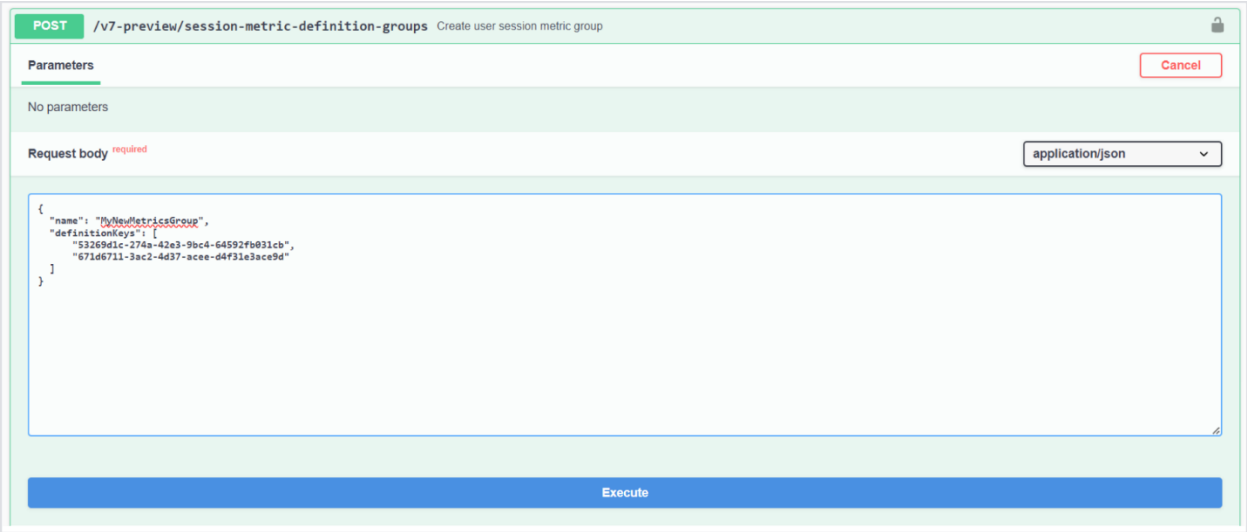
NOTE: This example shows the output of the SSLHandler definition, which allows the script to bypass certificate validation. This may not meet your organization's security requirements.

```
SSL Handler already exists.
id                                   name
--                                   ----
53269d1c-274a-42e3-9bc4-64592fb031cb Disk Transfers/sec
671d6711-3ac2-4d37-acee-d4f31e3ace9d {Avg Writes Per Sec, Avg Reads Per Sec}
```

Next steps

This script is designed to simplify the process of creating Session Metric definitions in bulk, compared to manually creating each metric through the Login Enterprise user interface or the Swagger API interface. However, it does not create a Session Metrics Group or add metrics to an existing group. For creating and updating a Session Metrics Group, the Login Enterprise user interface remains the most straightforward option. For details, see Creating a Group for Session Metric definitions. However, you can also use the API to manually populate Session Metrics Groups.

The script's output aims to streamline the creation process by providing the display name and ID of newly created session metrics. Using this output, you can easily copy and paste the ID of the newly added Session Metric definition into the "definitionKeys" list in the Swagger API interface. Note that the script assumes not all users will want to import and add all metrics to the same group.



Configuring Providers (Optional)

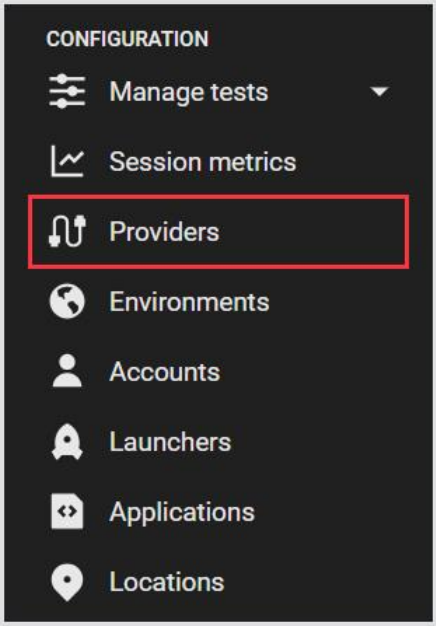
- Overview
- Creating a Provider
- Connecting the Provider with an Environment

Overview

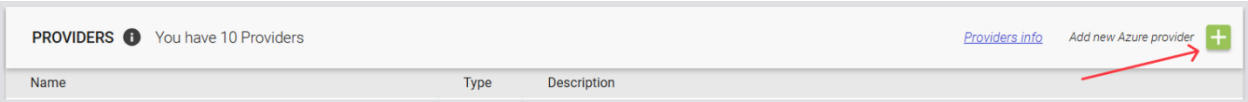
Data Providers is a way to collect information via an API for active user counts within existing Azure Virtual Desktop Environments.

Creating a Provider

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Providers**.



2. In **Providers**, click the green "+" to add a new Azure provider.



3. Define your Provider settings:

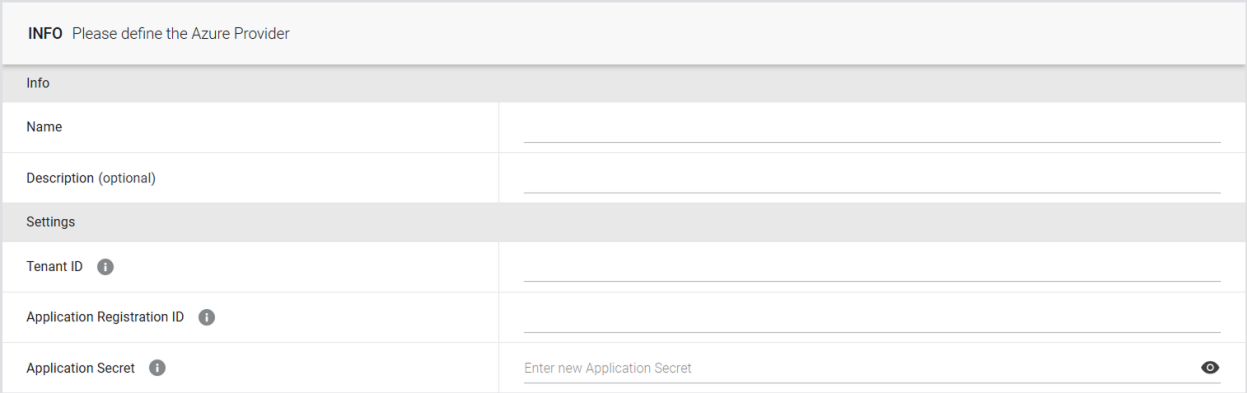
Info	
Name	A designated name for the Provider.

Description (optional)	A summary outlining the purpose of the Provider.

4. Create a managed application identity within your organization's Azure Active Directory. For details, see the Microsoft Documentation.
5. In **Settings**, provide the following:

Settings	
Tenant ID	The unique identifier for your organization's Azure Active Directory instance.
Application Registration ID	The distinct identifier for your managed application identity within Azure, accessible through the App Registrations service in the Azure Portal.
Application Secret	During the creation of your application identity (step #4 above), you'll assign a secret. This secret is crucial for authenticating the Login Enterprise application identity with your Azure Active Directory.

6. Click **Save** to create your Azure provider object.



Connecting the Provider with an Environment

Once you've created a Provider, it can be attached to an Environment to generate accurate active user data to populate the Operations Dashboard.

To proceed, gather the following details from your organization's Azure Portal:

- **Resource Group**: This refers to the name of the Azure resource group housing your AVD Host Pool.
- **Host Pool**: This denotes the name of the AVD Host Pool from which data should be retrieved.

For information on creating Environments, see Environments.

Configuring Environments (Optional)

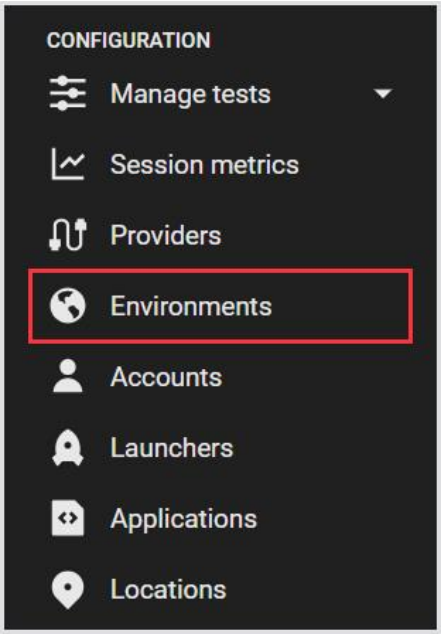
- Overview
- Creating an Environment
- Configuring Provider Settings (Optional)
- Configuring Cost Settings

Overview

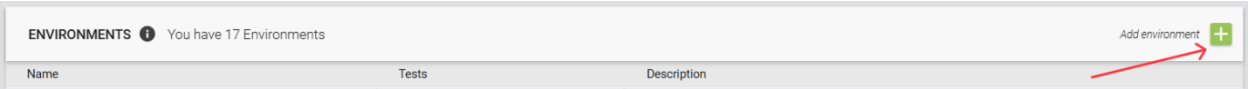
Environment entities define costs and capacity for a group of Continuous and Load Tests in the Operations Dashboard. For more information, see Operations Dashboard.

Creating an Environment

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Environments**.



2. In **Environments**, click the green "+" on the top right to create a new Environment entity.



3. Define your Environment settings:

Info

Name	The designated title or identifier for the environment.
Description (optional)	A summary that outlines the purpose and key characteristics of the environment.

Data providers can be integrated into the environment to collect active user counts from an existing Azure Virtual Desktop environment via an API. For more information on data providers, see Configuring Providers.

Configuring Provider Settings (Optional)

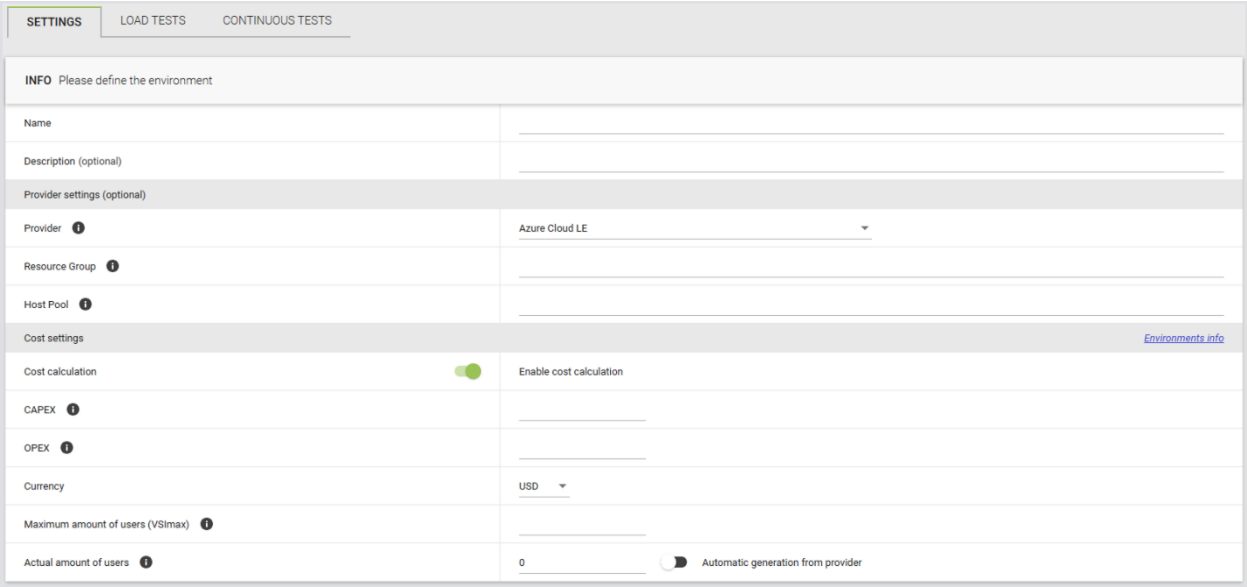
4. To eliminate manual data input and enhance data accuracy in the Operations dashboard, you can specify the following:

Provider (optional)	
Provider	The virtual desktop provider where the environment's desktops and/or applications are hosted.
Resource Group	The name of the Azure resource group that contains your AVD Host Pool.
Host Pool	The name of the AVD Host Pool from which data should be retrieved.

Configuring Cost Settings

5. In **Cost Settings**, select the toggle to trigger additional configuration:

Cost settings	
CAPEX	The total capital expenditures required for the initial setup and infrastructure of the environment.
OPEX	The ongoing operational expenditures necessary for the maintenance and daily operation of the environment.
Currency	The type of currency used for all cost-related entries, such as USD, EUR, or GBP.
Maximum number of users (VSI Max)	The highest number of users that the environment is designed to support simultaneously.
Actual number of users	The projected or current number of users actively using the environment.



This data will be displayed in the Operations Dashboard under the corresponding environment. For more information, see the Operations Dashboard.

Configuring Locations (Optional)

- Overview
- Adding a Location
- Assigning a Location to a Launcher
- Locations and Tests
- Launchers and Launcher sessions in map view

Overview

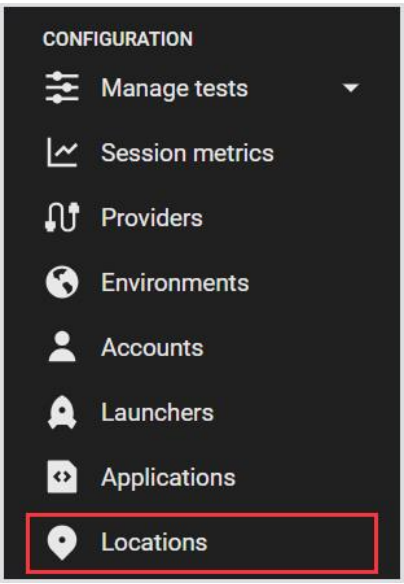
You can add Launchers to defined Locations. This is to more closely represent successful connections and latency that real-life users would experience in, for example, an office branch. Having a Launcher set up in each location needing to be monitored is a best practice. However, if you narrow down your selection to one location that's not connecting to targets successfully, targeted troubleshooting can occur.

TIP: In a Load Test scenario, the best practice ratio for simultaneous Test connections per Launcher is 30 to 1. So, if running a 100-user Load Test, you need 4 Launcher hosts.

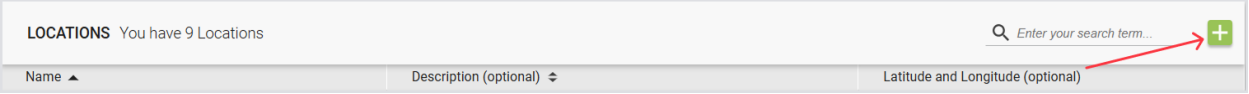
If Launchers are attached to configured locations when a Continuous Test type is in progress, these Launchers will be visible in the map view.

Adding a Location

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Locations**.



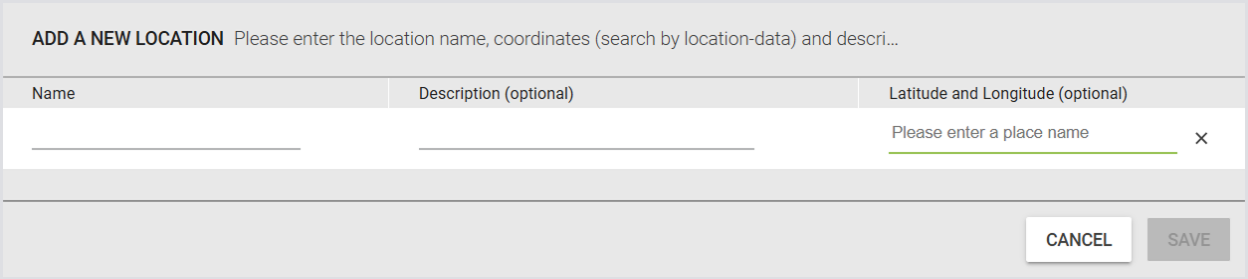
2. In **Locations**, click the green "+" on the top right to add a new Location.



3. In the **Add a new location** window, provide the following:

Name	Description (optional)	Latitude and Longitude (optional)
The name of the location.	A brief description of the location.	The location's coordinates. You can search by location data to fill these automatically.

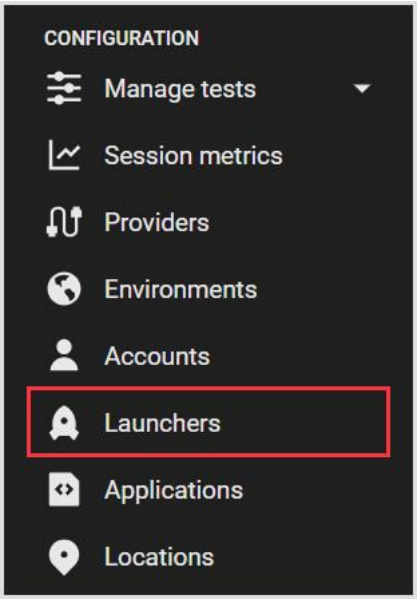
4. Click **Save** to apply the configuration.



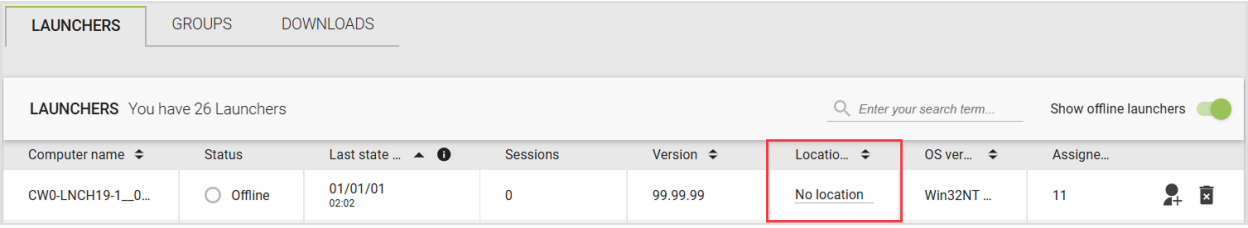
Assigning a Location to a Launcher

Now that you've added a location, you need to assign the location to a Launcher. To do this:

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration > Launchers**.



2. In **Launchers > Location (optional)**, select the location for your Launcher.

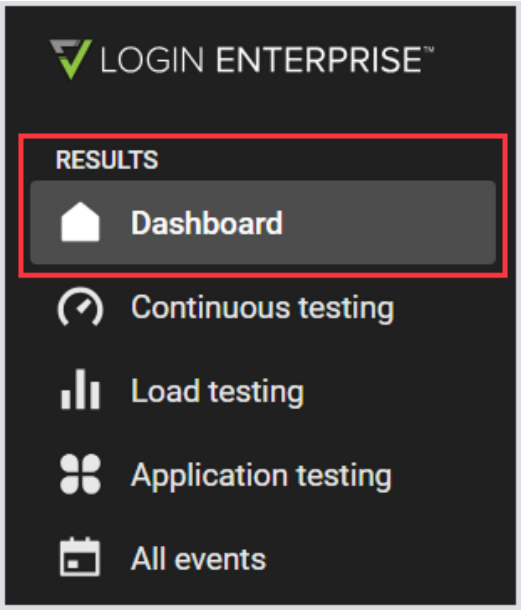


Locations and Tests

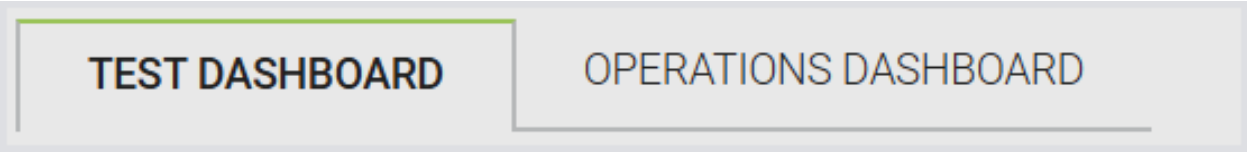
Although you cannot add locations to a Test directly, in a way the locations are indirectly connected to the Test via the Launchers. This, however, only works for the Continuous Test type.

To view locations for your Continuous Test:

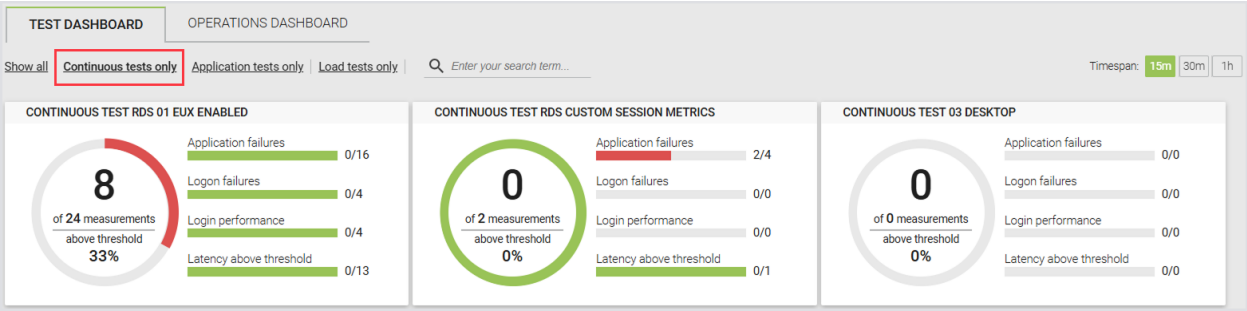
1. In the **Login Enterprise sidebar menu**, navigate to **Results > Dashboard**.



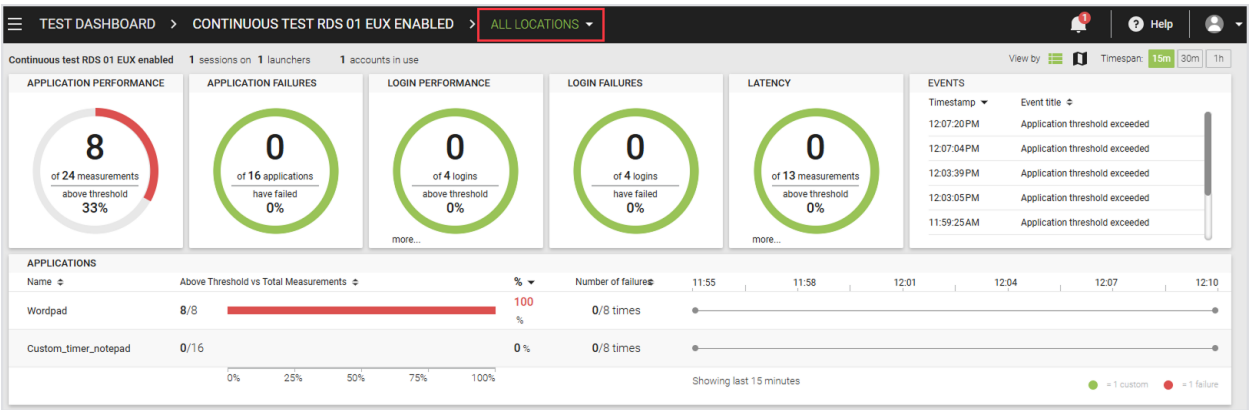
2. Make sure you're viewing the **Test dashboard**.



3. Under the **Test dashboard**, click **Continuous Tests only**, and then click on one of the Continuous Test instances. This way, you'll see more details about this particular Test.



4. In the **top bar menu** > **All locations**, select the location. These locations are retrieved from the Launchers that are linked to those particular Tests. When you select one location, the page will only show the results of that location/Launcher for that particular Test.

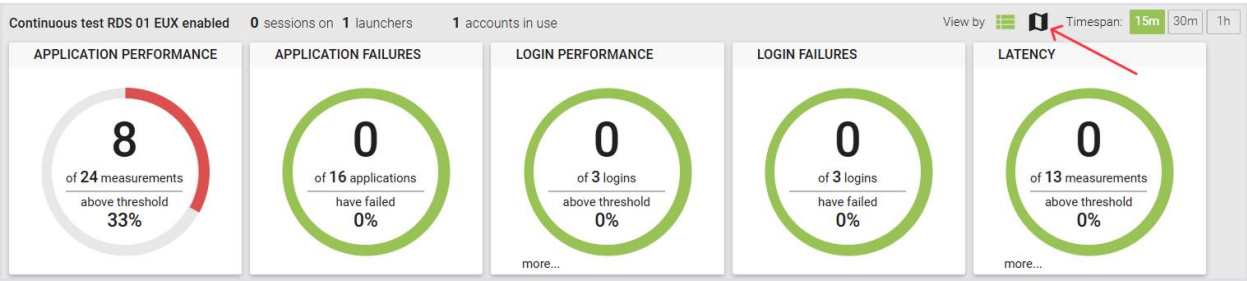


Launchers and Launcher sessions in map view

To view Launchers and Launcher sessions on the world map:

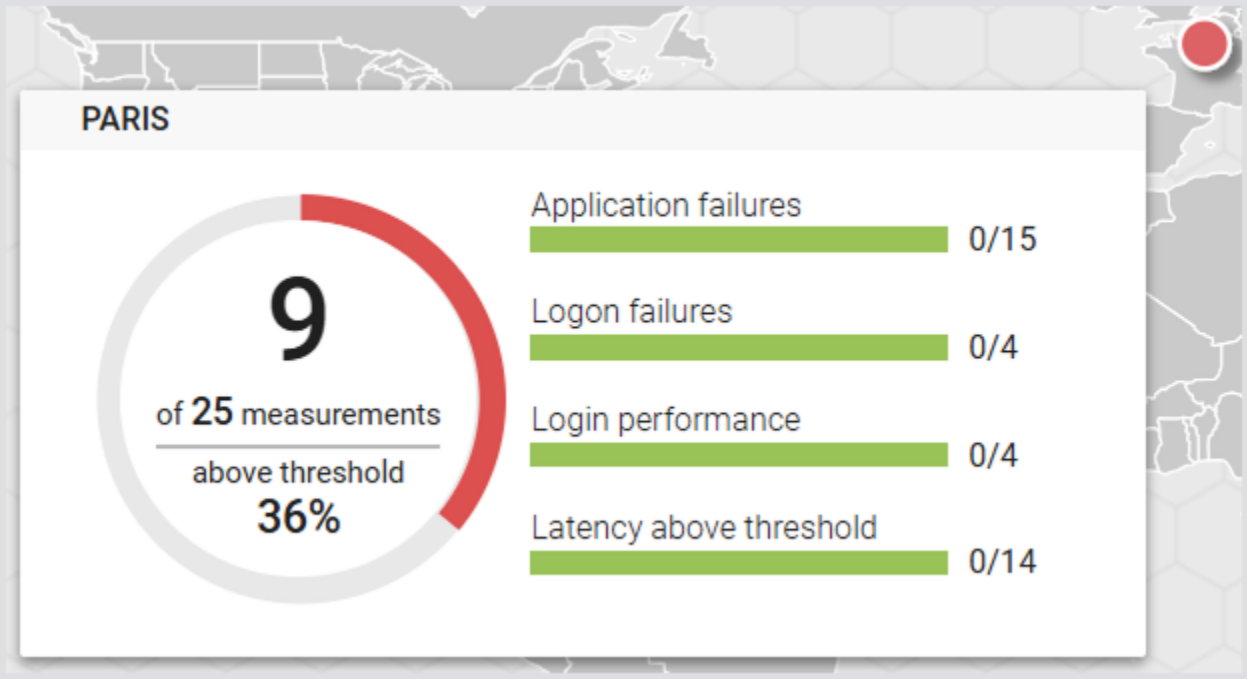
In the **Top bar menu > View by**, select the world map view. This will give you a world map with all the locations of the Launchers linked to that Test and the results (summary) of each Launcher/location. If you have 2 (or more) Launchers in the same location, it will just add up the results of those 2 Launchers.

Now, you can view the Launcher location plotted on the world map. The hoverable tooltip shows connection and Test successes and failures.



Now, you can view the Launcher location plotted on the world map.

The hoverable tooltip shows connection, and Test successes and failures.



Configuring Applications for Testing

- Overview
- Creating an Application
  - Adding an Application
  - Importing an Application
    - Application Scripts
    - Browser Scripts
- Creating an Application Group
  - Adding Wait Actions
  - Run Once and Leave Application Running
    - Run Applications Once
    - Leave Applications Running
    - Resource Usage
- Managing Access to Applications and Application Groups
- Downloading and Editing Application Scripts
- Scripting Secure Application Credentials
  - Configuring Credentials
  - Using Credentials in an Application Script
- Login Enterprise Script Editor
- Editing Application Scripts
- Troubleshooting Application Errors

Overview

Login Enterprise enables testing of Applications installed on Target Windows hosts, focusing on critical metrics such as Application launch times and the performance of specific actions. These Applications serve as Workload files that the Virtual Appliance Target Engine executes, simulating Virtual User interactions.

Multiple Application Workloads can be organized into collections called Application Groups, which are assigned to Test scenarios. This structure allows for comprehensive and structured testing, ensuring that various Application sequences are effectively evaluated during each Test run.

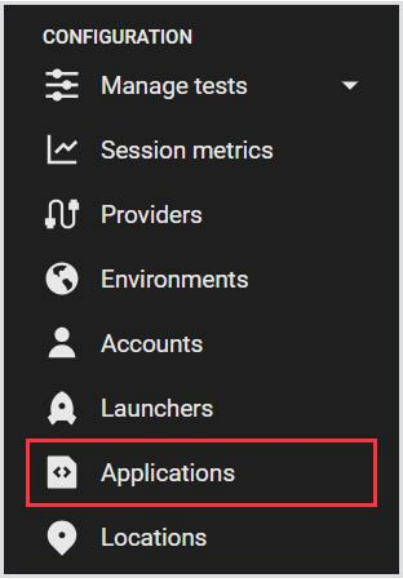
Creating an Application

You can create an Application in one of the following ways:

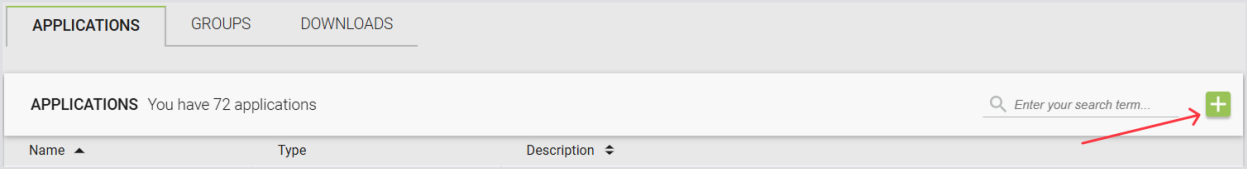
- Add an Application: Configure a new Application manually.
- Import Application: Import an existing browser automation script.

Adding an Application

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Applications**.



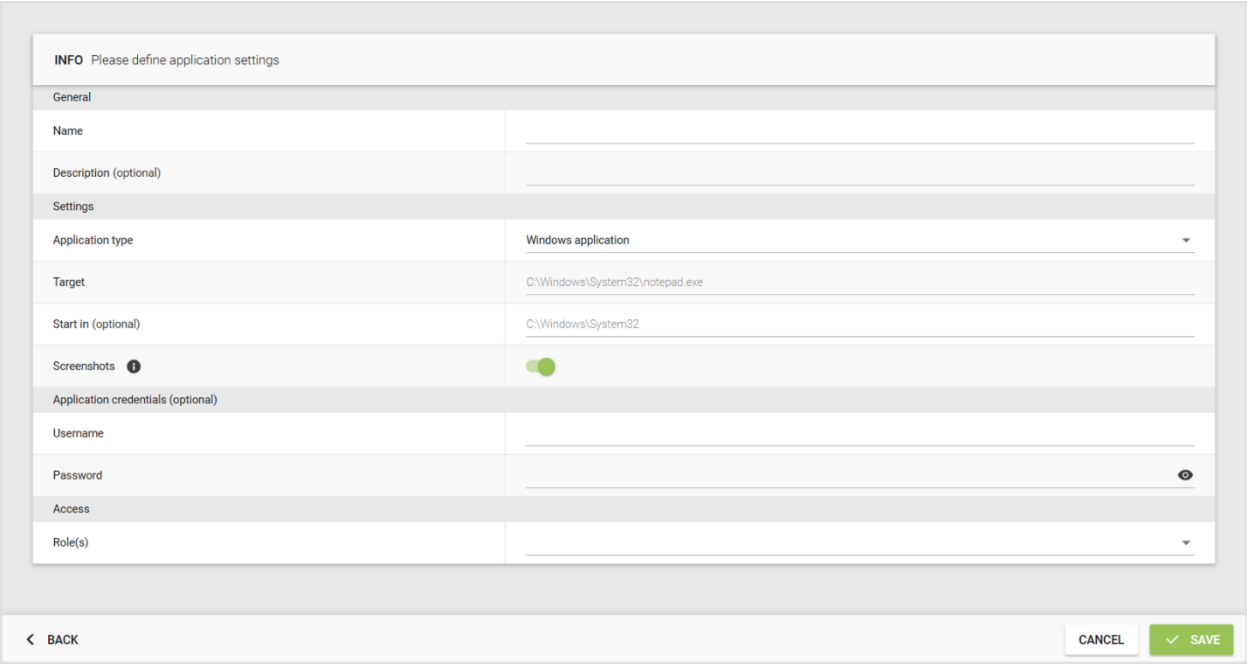
2. In **Applications**, click the green "+" on the top right, and then select **Add Application**.



3. Define the Application settings by configuring the following:

General	
Name	The name of the Application.
Description	A brief description of the Application and its purpose.
Settings	
Application Type	The type of Application workload to run.
Target	The Target Windows host where the Application will be executed.
Start In (optional)	The working directory from which the Application will launch.
Screenshots	Enable screenshots to help identify the Application or assist with action scripting. For details, see Taking Screenshots.
Application Credentials (optional)	
Username	The username for Application access. For more information, see Scripting Secure Application Credentials.

Password	The password for the specified user. For more information, see Scripting Secure Application Credentials.
Access	
Roles	Assign roles that are allowed to access or manage this Application. For details, see Managing Access to Applications and Application Groups.



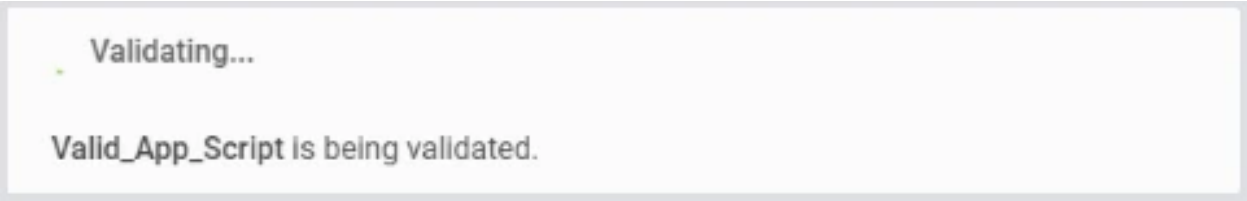
This will create an empty Application script, which you will normally want to update. For more information, see Downloading and Editing Application Scripts.

NOTE: Adding an Application here will automatically configure an Application Start Timer.

Importing an Application

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Applications**.
2. In **Applications**, click the green "+" on the right, and then select **Import Application**.

After clicking **Import Application** and completing the process, you'll see a 'Validating...' pop-up message. This indicates that your script is undergoing validation.



Application Scripts

When uploading an application script, the system automatically prefills the following fields:

1. Name
2. Application Type
3. Target

You can configure the remaining fields, as necessary. For details, see Adding an Application.

Browser Scripts

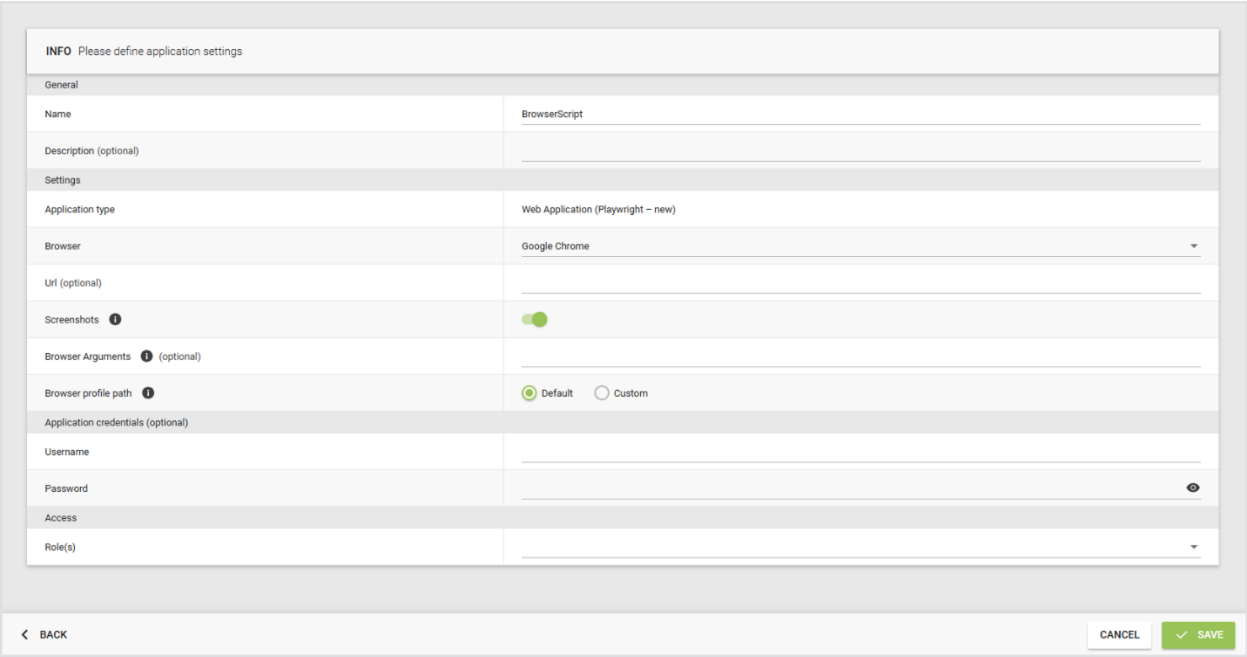
When uploading a browser script, the system automatically prefills the following fields:

1. Name
2. Application Type
3. Browser
4. URL (if available)

You can configure the remaining fields, as necessary. See the full configuration table below for details:

General	
Name	The name of the Browser Script.
Description	A brief description of the script and its purpose.
Settings	
Application Type	The type of Application workload to run.
Browser	Select the browser in which the script will be executed (for example, Chrome, Edge, or Firefox).
URL (optional)	The URL that the browser opens when the script starts.

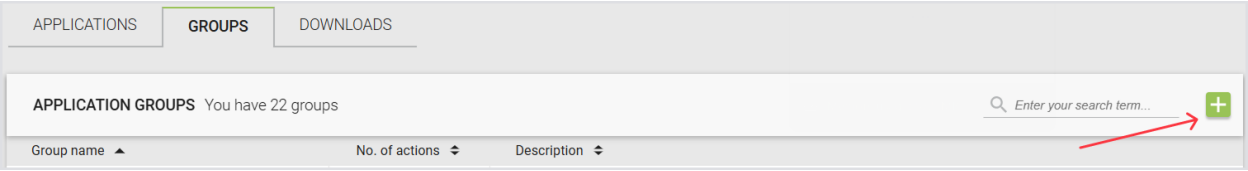
Screenshots	Enable screenshots to help identify the Application or assist with action scripting. For details, see Taking Screenshots.
Browser Arguments (optional)	Add arguments using the syntax specific to each browser. For example: use `--incognito` for Chrome, `--inprivate` for Edge, and `-private` for Firefox to launch the browser in incognito or private mode.
Browser Profile Path	Specify a custom user profile path if it isn't set to the default. • **Default**, such as those used by Chromium-based browsers (Microsoft Edge and Google Chrome) and Firefox, are automatically used when browsers are installed with standard settings. These paths store user-specific data such as cookies and configurations. • **Custom** can be specified when tailored configurations are required, e.g., for centralized management or compliance purposes. Using a custom path ensures the correct user settings are applied during testing.
Application Credentials (optional)	
Username	The username to access the application or website. For more information, see Scripting Secure Application Credentials.
Password	The password for the specified user. For more information, see Scripting Secure Application Credentials.
Access	
Roles	Assign roles that are allowed to access or manage this script. For details, see Managing Access to Applications and Application Groups.



Creating an Application Group

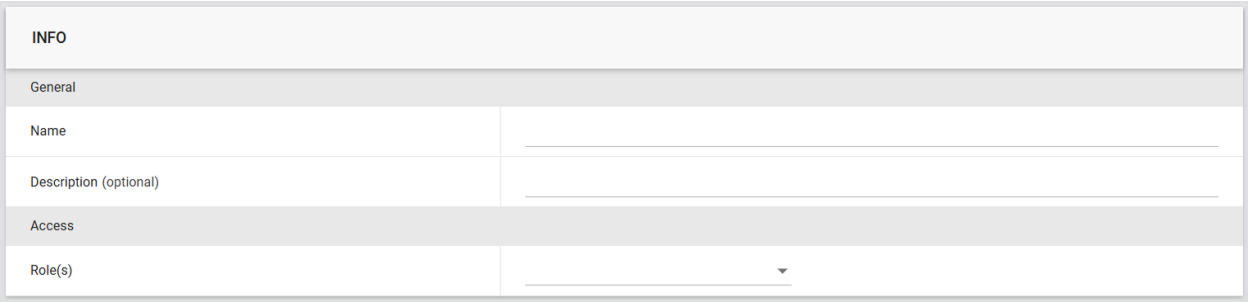
Application Groups allow you to run multiple specified Applications together. You can attach these Groups to Continuous Tests, Load Tests, or Application Tests.

1. In the **Login Enterprise sidebar menu**, navigate to **Applications** and select **Groups** in the top menu.
2. In **Groups**, click the green "+" on the top right to create a new Application Group.

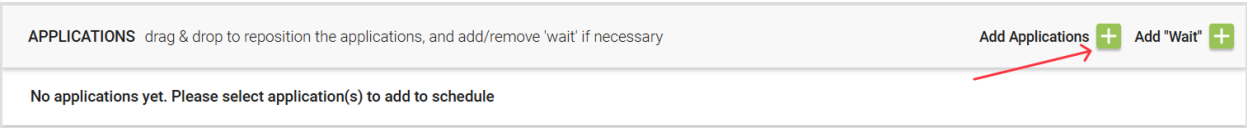


3. Define the Application Group settings:

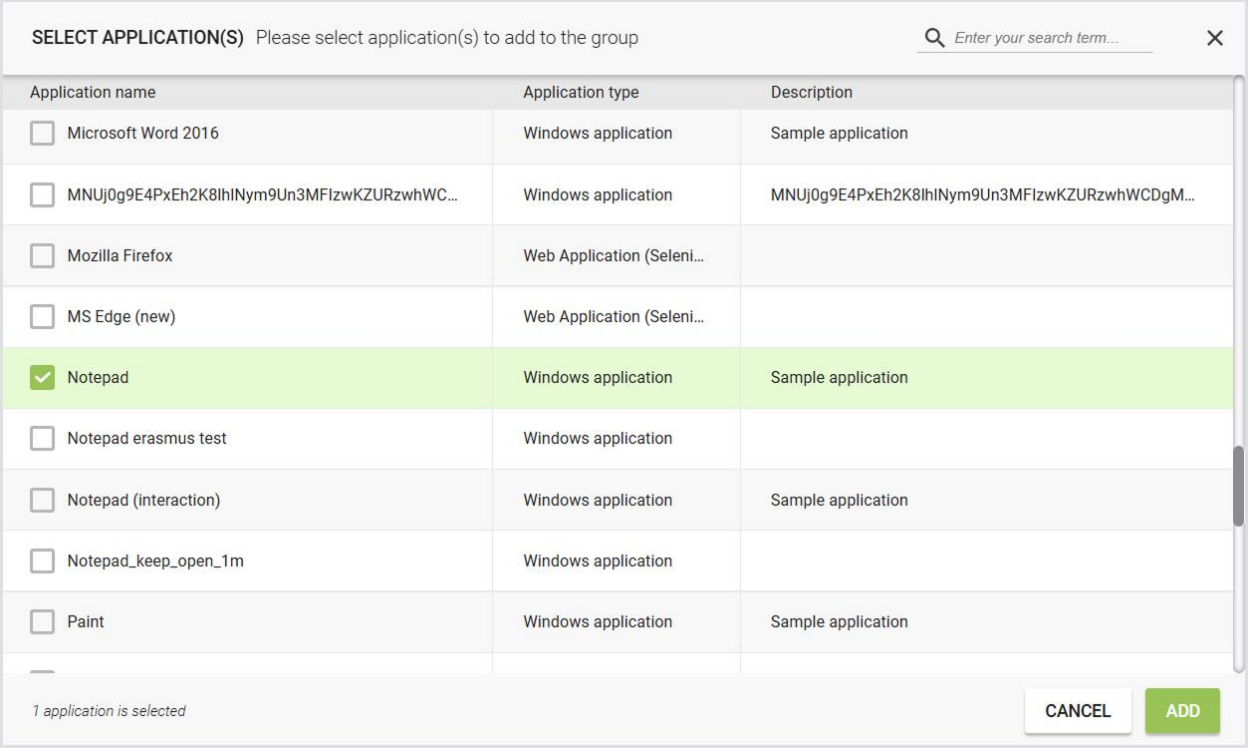
General	
Name	The name of the Application Group.
Description	A brief description of the Application Group and its purpose.
Access	
Roles	Assign roles that are allowed to access or manage this Application Group.



4. In **Applications**, click **Add Applications**.



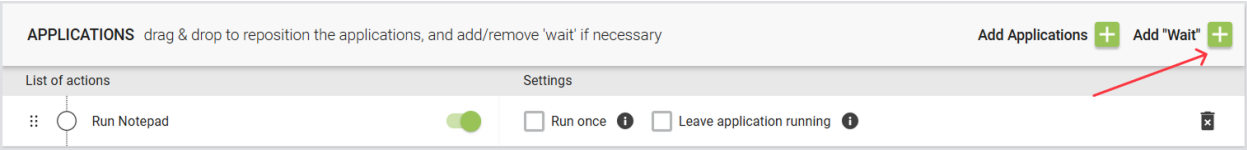
5. In the pop-up window, select the Applications you want to include in the Group, and click **Add** to confirm.



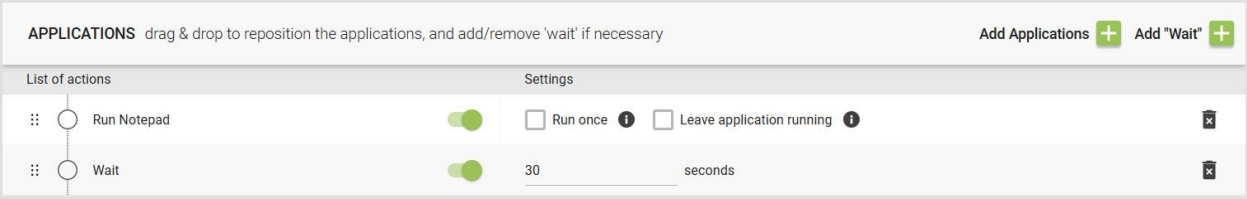
Adding Wait Actions

You can insert Wait actions between Applications to control the timing of execution.

1. In **Applications**, click **Add Wait**.



2. Specify the duration of the wait (in seconds).



3. Add multiple Wait actions as needed.

TIP: Applications and Wait actions can be reordered within the list:

- Drag and drop items to change their position.
- Wait actions can be placed between Applications to create pauses during the execution sequence.

This allows you to define the exact order and timing of Application workloads executed during a Test.

Run Once and Leave Application Running

You can configure how Applications behave during the Test run using the following options:

- **Run Applications Once**: Each Application in the sequence is launched only once during the Test execution. For more information and use cases, see Run Applications Once.
- **Leave Applications Running**: Applications remain open after they are launched, instead of closing before the next Application starts. For details and use cases, see Leave Applications Running.

These options allow you to control whether Applications are executed as independent launches or remain active throughout the Test scenario, depending on the testing requirements.

NOTE: There are 2 possible configuration options for Application behavior:

1. This can be configured at the Test level for single Tests.
2. This can also be set within Application Groups.

However, the configuration is not available for a single Application in the Application settings.

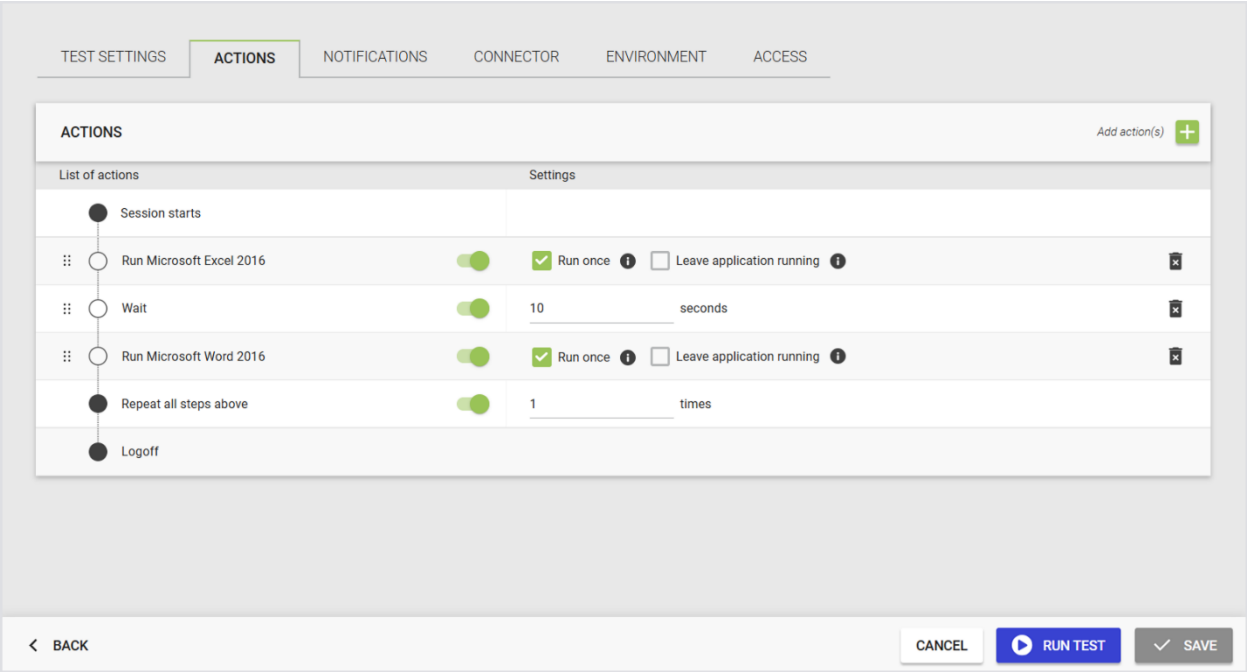
Run Applications Once

This option is intended for executing tasks that are only required once, such as configuring registry keys or performing specific automation setups at the beginning of a Test.

Load Test

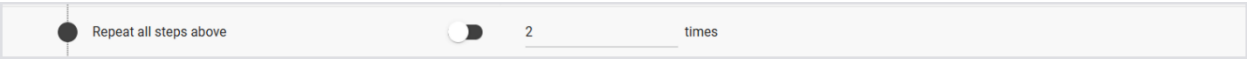
In a Load Test scenario, you can attach multiple Application scripts, such as Excel and Word. For Excel, you can enable "Run once", while Word follows its default behavior.

During user sessions, actions will initiate with Excel first, followed by a 10-second pause, then Word. After that initial pass, the remainder of the Test will consist of Word running followed by a 10-second pause. This sequence repeats a # of times until the user logs off.



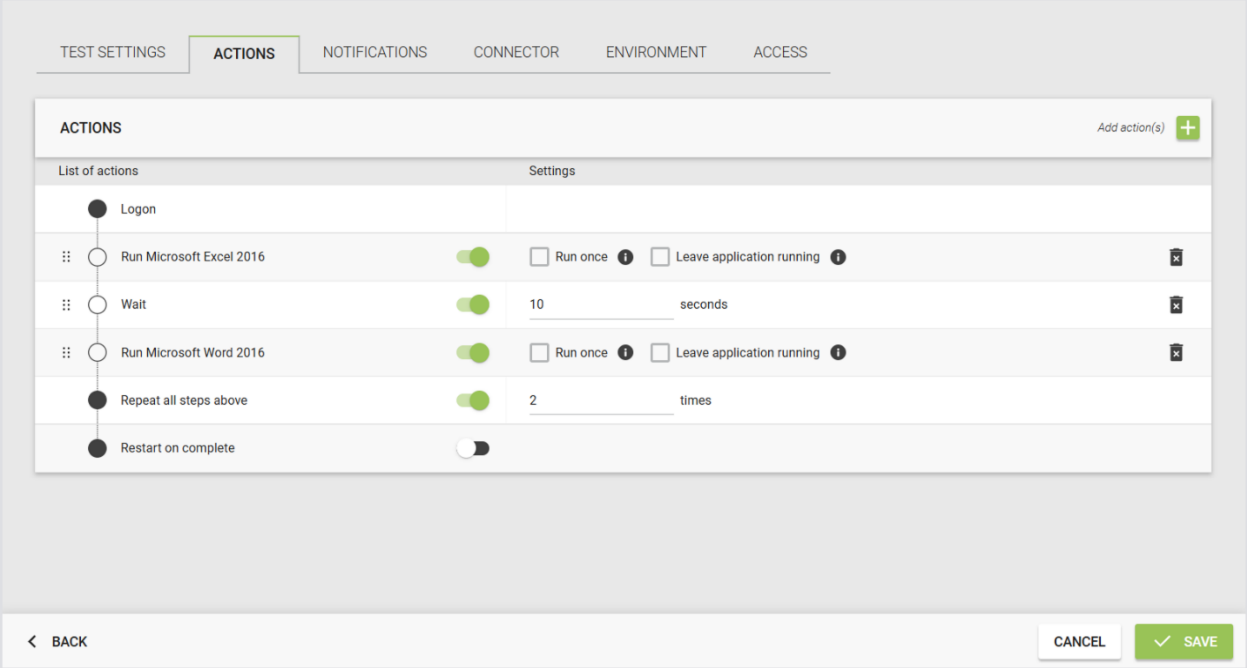
Continuous Test

In a Continuous Test, enabling "Run once" is effective **only when** "Repeat all steps above" is configured to **2** or higher.



If disabled or set to **1**, "Run once" has no effect, as the default behavior runs everything once. When enabled with a setting of **2** or higher, in a Continuous Test setup with Excel set to "Run once" and Word set to default, the behavior is as follows:

The user logs on, runs Excel once, waits 10 seconds, runs Word, waits 10 seconds, runs Word again, and then logs off.



Application Test

In Application Testing, there is no "Run once" option for Applications. Application Tests execute each Application once and do not repeat.

Leave Applications Running

This option ensures that the Application specified in the configured Application script continues to run in the background as Login Enterprise moves to the next Application script.

IMPORTANT: Leaving Application Running does not apply to web scripts; it is only applicable to Application scripts.

This behavior mimics real user interactions, where Applications are often left open rather than closed after use. “Leave Application Running” can be used with “Run Once” to start a program and have it in the background for the entire Test Run without touching it again. However, if you have a repeating Test and you have not set “Run Once”, we will execute the script each time.

NOTE: It’s your responsibility to make sure any “Leave Application Running” script can handle it when it is executed a second time, and that any other Application Scripts that want to interact with the process know how to find its main window.

This feature operates consistently across all use cases of Login Enterprise. Here are some important considerations for successful use:

Application Title

Each command sent to an Application within Login Enterprise scripts requires identification of the Application. Typically, this identification is based on the Application's title. When using "Leave Application Running," it is crucial that the title of the focused action within a script is as unique as possible. This prevents commands from being sent to the wrong Application inadvertently open in the background.

Repeat Cycle

In Continuous Testing, there is an option to repeat steps, while in Load Testing, all actions are repeated by default. When "Leave Application Running" is enabled for an Application script, it may result in the same script (e.g., Excel) running multiple times without closing. Therefore, the script must be designed to handle subsequent runs seamlessly. For instance, if the first action in an Excel script is to open a file, ensure this action can be repeated at the script's end.

Process Started vs Process Running

Sometimes, the process you start, e.g., "start.exe", differs from the process that is running, e.g., "App.exe". If an Application is left running and the script is rerun, the script may attempt to start the initial process again, potentially causing unintended behavior.

Multiple Application Scripts of the Same App

When multiple Application scripts use the same Application and "Leave Application Running" is enabled, Login Enterprise may struggle to differentiate between Application instances. For example, if a script starts Word and another script attempts to start Word again, the Appliance may recognize that "word.exe" is already running and refrain from starting a new instance. Leaving multiple instances running concurrently can lead to unpredictable behavior. Consider launching the Application using a specific file, e.g., starting Word using a Word document to manage instances effectively.

Resource Usage

Leaving Applications open consumes more resources during testing compared to closing them. Be mindful of resource utilization when opting to keep Applications running after a script completes. Make sure you do not end up in a situation where you have 5 apps open that each take 25% of your memory.

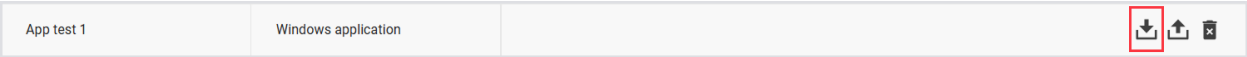
Managing Access to Applications and Application Groups

You can control which users can view or manage specific Applications and Application Groups by assigning access roles. This helps ensure that only authorized users, the “owners” of the Application, can manage its settings or include it in Tests. For details, see Managing Access to Applications and Application Groups.

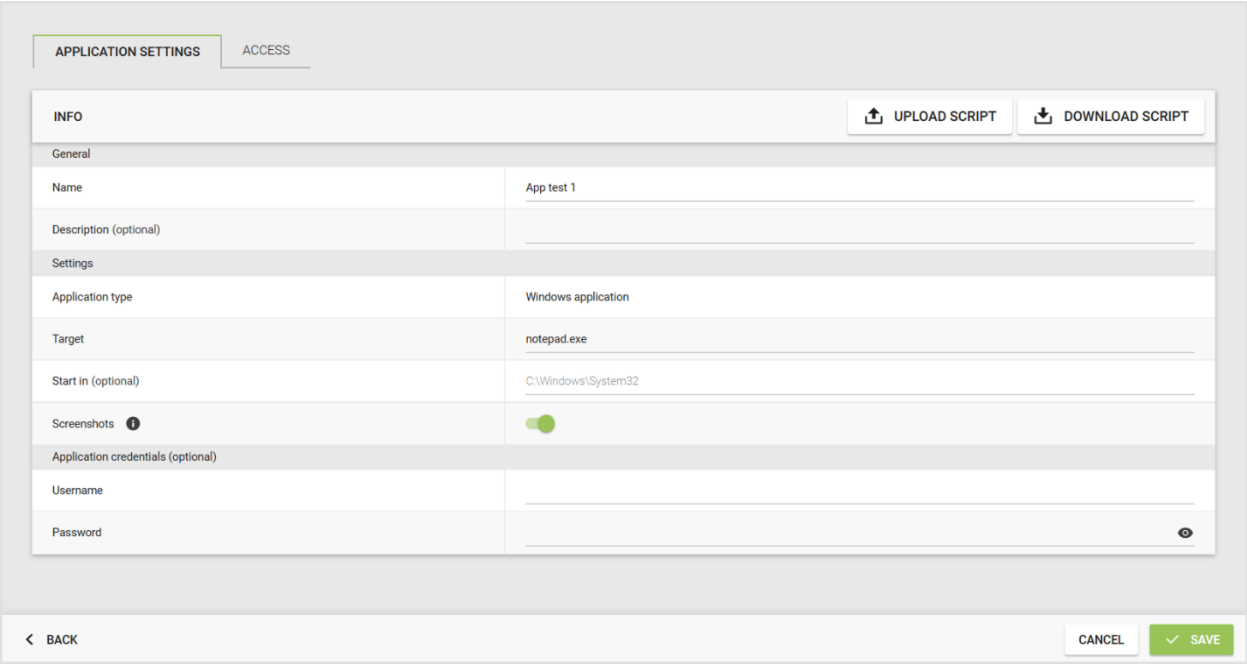
Downloading and Editing Application Scripts

Once you've added the Application, click the **Download** icon on the right to download the script. This serves as a backup and allows you to edit the Application script to include new timers and functions.

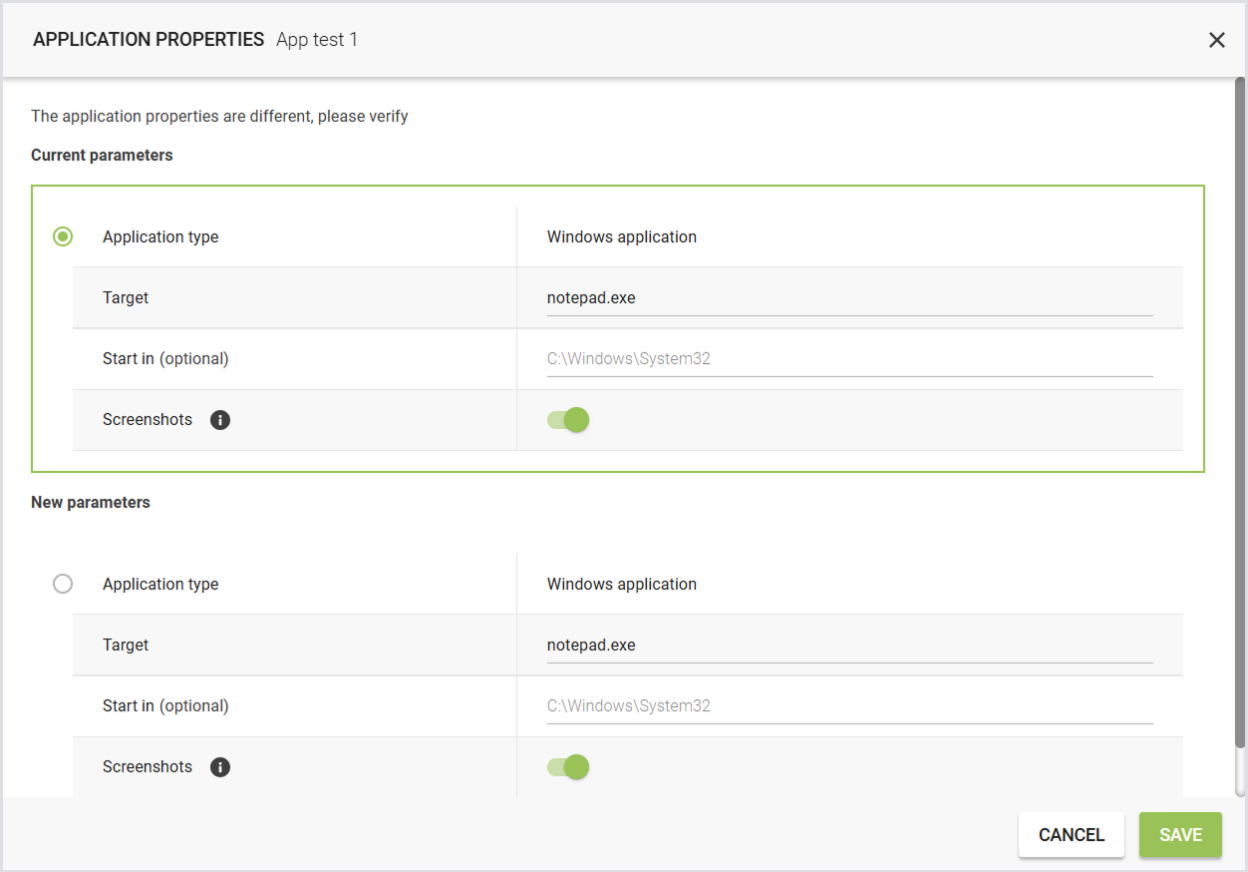
NOTE: After making updates to the script, upload the modified file back to Login Enterprise.



Click on the Application to make changes in its configuration if necessary. The **Application Settings** window opens.



You can also upload and download scripts from this window.

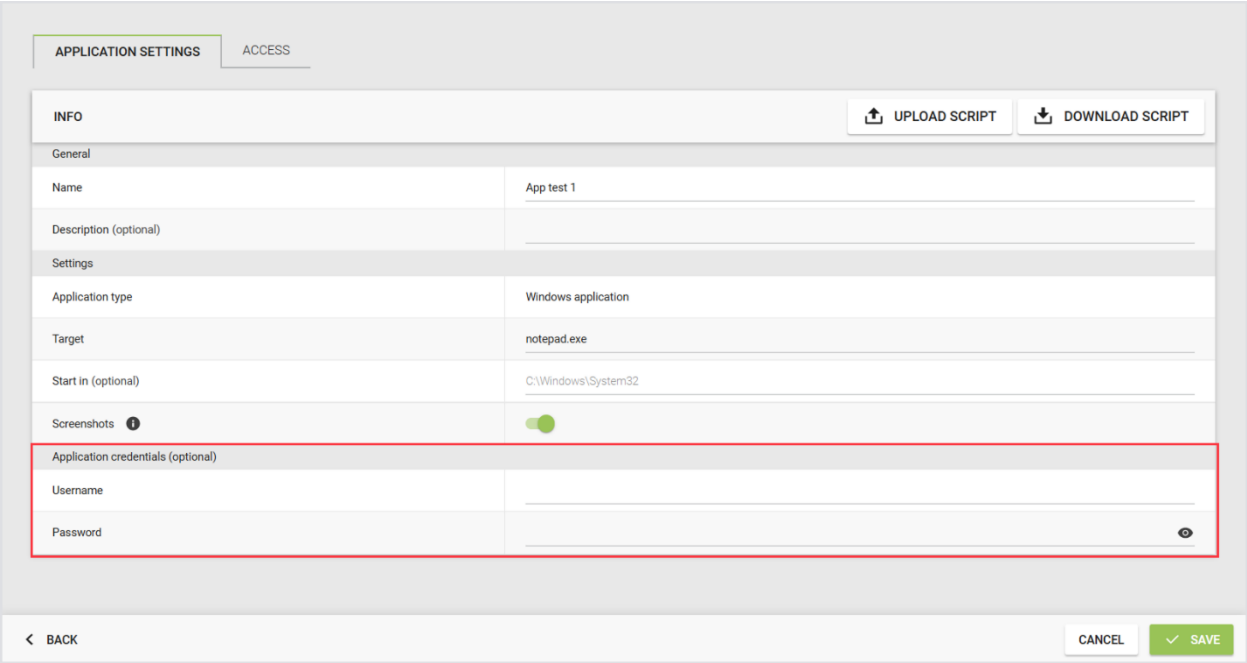


Scripting Secure Application Credentials

Login Enterprise lets you configure encrypted passwords for Applications. This password is encrypted and saved in a variable to be used in the Application scripts.

Configuring Credentials

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Applications**.
2. Click on the Application you want to configure the credentials for.
3. In **Application Credentials (Optional)**, fill in the username and password you want to use in your Application script.
4. Click **Save** to apply the settings.



Using Credentials in an Application Script

The credentials are transformed into a variable that can be used during scripting. There are two variables that you can use in the script:

- ApplicationUser
- ApplicationPassword

To use this, you can use the following command when you want to type it into a window. We assume that the user is called Login VSI and the password is Password!

```
Type(ApplicationUser,hideInLogging:true);
Type("{Enter}")
Type(ApplicationPassword,hideInLogging:true);
```

The result would be:

User

Password!

The option hideInLogging is best used when using passwords, as this function hides them from logging, so there is no potential security breach.

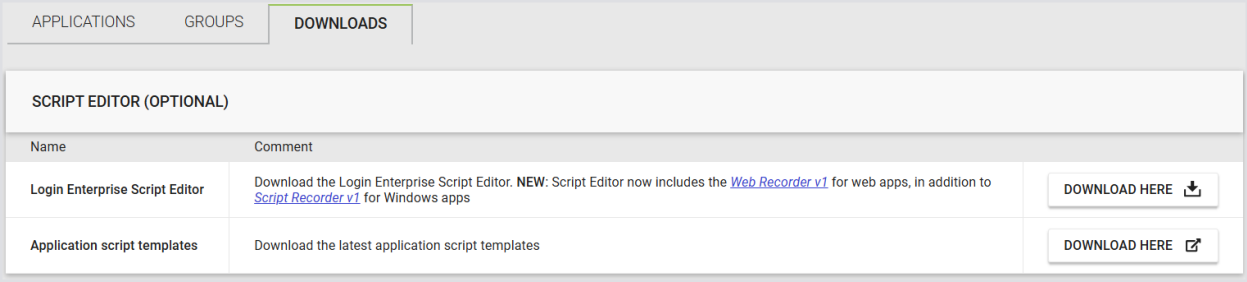
Script Editor

The Login Enterprise Script Editor is a tool designed to assist administrators in creating custom Application workflows. These workflows can be seamlessly incorporated into various testing scenarios, including Application, Load, and Continuous Tests. By leveraging the Script Editor, administrators can streamline Test creation and execution, ensuring comprehensive and efficient testing processes tailored to specific Application requirements.

For more details on the Script Editor, its benefits, use cases, and more, see the Script Editor.

Script Editor includes the Web Recorder for web apps, in addition to Script Recorder v1 for Windows apps. To learn more about the Script Recorder, its use cases, benefits, and more, see Script Recorder.

NOTE: While you may prefer using Visual Studio Code (VSCode), the Login Enterprise Script Editor is designed to provide a more integrated and supported experience. Although you can still use VSCode for scripting, it is not officially supported, and continued functionality is not guaranteed. We recommend using the Script Editor for best results.

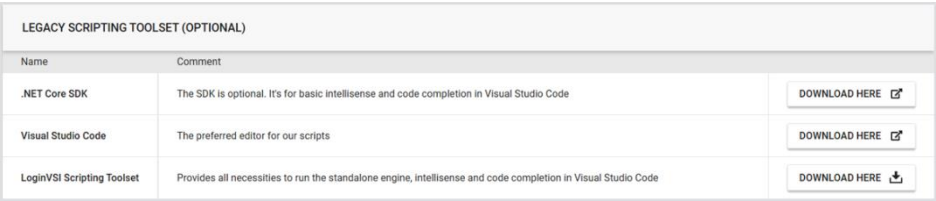


Editing Application Scripts

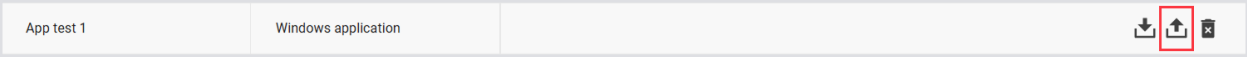
You can edit Application scripts using VSC. For this, ensure you have the necessary software:

- Visual Studio Code with .NET Core support.
- Scripting Toolset for code snippets, example workflows, and syntax highlighting.

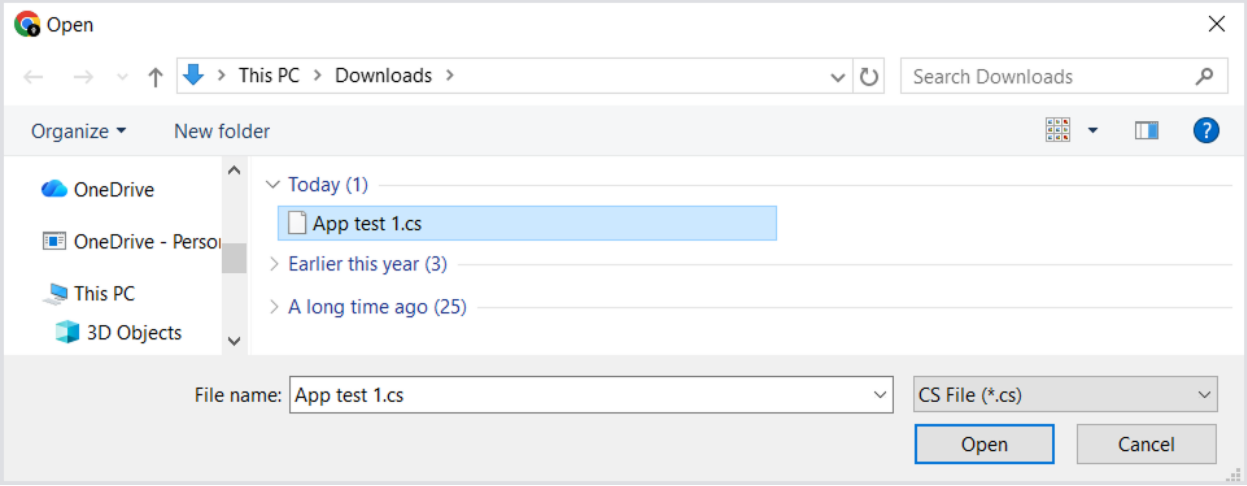
Open Visual Studio and navigate to the extracted folder of the Script Toolset for customization.



Once you've edited the Application script, click the **Upload** icon on the right to upload the script back to Login Enterprise. Login Enterprise checks scripts for syntax errors before uploading. Error messages provide clues for debugging.



NOTE: Script uploading will only support built-in scripting functions. For more information, see Scripting Functions Overview.



Troubleshooting Application Errors

To learn how to resolve Application errors, see Troubleshooting Application Errors.

Troubleshooting Application Errors

- Overview
- Diagnosing Application failure
- In-session troubleshooting
- Troubleshooting Post-Test Application Errors
  - Using the All Events Page
  - Viewing Events by Test Type
  - Analyzing Screenshots and Logs

Overview

When running Login Enterprise, you may encounter situations where an Application fails to execute correctly. If such an error occurs, Login Enterprise will notify you through an event and/or email (if configured), and it may halt the execution of the Application script or proceed to your designated error-handling routine.

If you encounter connection issues, you can:

- Attempt to resolve the issue using the steps below.
- Contact us for further assistance. You can also send the log files along with your inquiry.

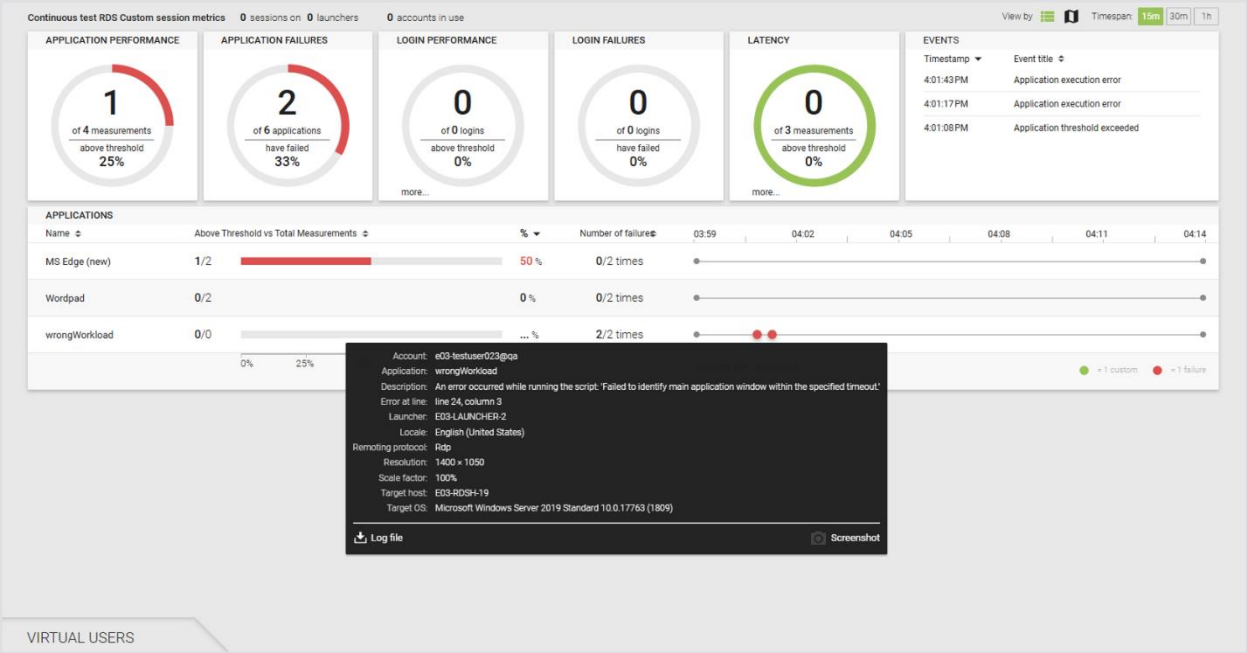
NOTE: This is a demonstration of a troubleshooting scenario for the Notepad application failure.

Diagnosing Application failure

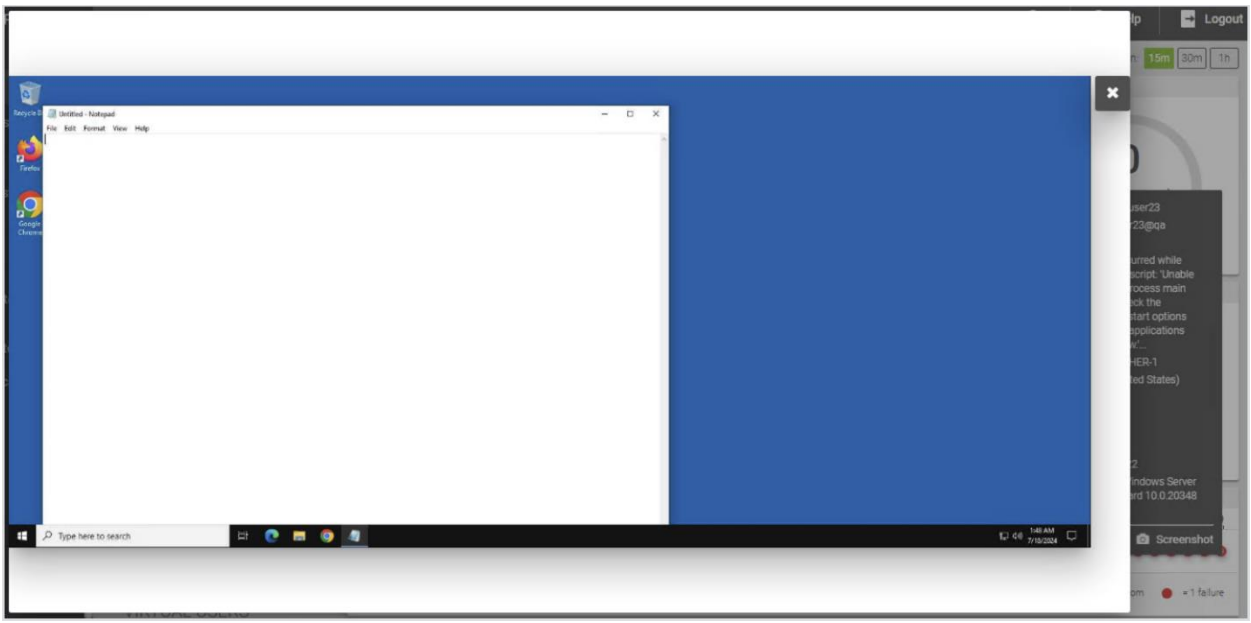
During an Application Test run, for example, in Continuous Testing, if an error occurs (for example, the engine fails due to the wrong window title being defined in the workload for the Notepad Application), follow these steps to troubleshoot:

1. **Identify the error:** In the Continuous Test scenario, a red dot will appear in the timeline on your Test dashboard whenever an error occurs for any Application or workload within the scenario. For example, if your scenario includes multiple Applications like Notepad and others, each Application's errors will be marked by a red dot. Click on any red dot to view detailed error information.

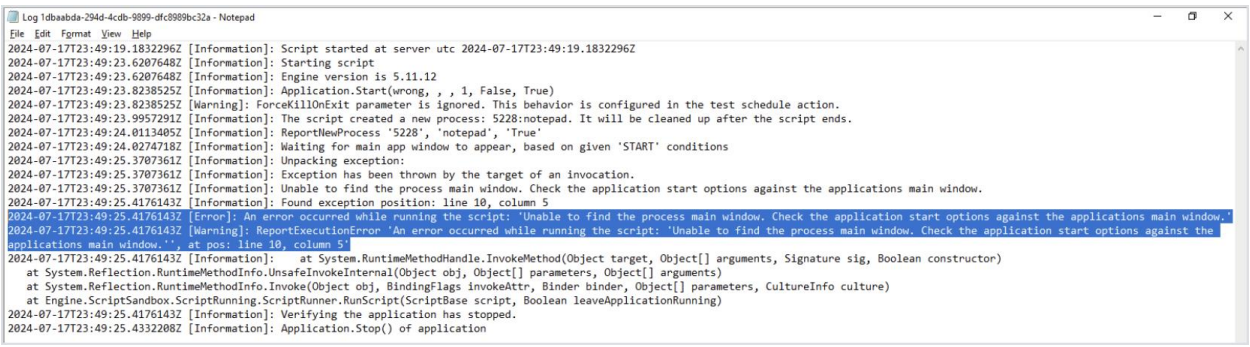
This popup provides information about the Application name, the Application failure description, and other relevant details.



2. **Review error details:** Click the Screenshot button to see the desktop's state at the time of the error. This visual representation helps identify what was occurring during the Application Test run when the error occurred.



3. **Analyze the log**: Click the Log file button to download the log files generated by the session. These logs detail the actions performed during the session and indicate where and how the execution encountered issues. For instance, the following log shows an error "Unable to find [specific element or resource]" during the execution of the Notepad Application.



```
Log 1dbaabda-294d-4cdb-9899-dfc8989bc32a - Notepad
File  Edit  Format  View  Help
2024-07-17T23:49:19.1832296Z [Information]: Script started at server utc 2024-07-17T23:49:19.1832296Z
2024-07-17T23:49:23.6207648Z [Information]: Starting script
2024-07-17T23:49:23.6207648Z [Information]: Engine version is 5.11.12
2024-07-17T23:49:23.8238525Z [Information]: Application.Start(wrong, , , 1, False, True)
2024-07-17T23:49:23.8238525Z [Warning]: ForceKillOnExit parameter is ignored. This behavior is configured in the test schedule action.
2024-07-17T23:49:23.9957291Z [Information]: The script created a new process: 5228:notepad. It will be cleaned up after the script ends.
2024-07-17T23:49:24.0113405Z [Information]: ReportNewProcess '5228', 'notepad', 'True'
2024-07-17T23:49:24.0274718Z [Information]: Waiting for main app window to appear, based on given 'START' conditions
2024-07-17T23:49:25.3707361Z [Information]: Unpacking exception:
2024-07-17T23:49:25.3707361Z [Information]: Exception has been thrown by the target of an invocation.
2024-07-17T23:49:25.3707361Z [Information]: Unable to find the process main window. Check the application start options against the applications main window.
2024-07-17T23:49:25.4176143Z [Information]: Found exception position: line 10, column 5
2024-07-17T23:49:25.4176143Z [Error]: An error occurred while running the script: 'Unable to find the process main window. Check the application start options against the applications main window.'
2024-07-17T23:49:25.4176143Z [Warning]: ReportExecutionError 'An error occurred while running the script: 'Unable to find the process main window. Check the application start options against the
applications main window.'', at pos: line 10, column 5'
2024-07-17T23:49:25.4176143Z [Information]:    at System.RuntimeMethodHandle.InvokeMethod(Object target, Object[] arguments, Signature sig, Boolean constructor)
   at System.Reflection.RuntimeMethodInfo.UnsafeInvokeInternal(Object obj, Object[] parameters, Object[] arguments)
   at System.Reflection.RuntimeMethodInfo.Invoke(Object obj, BindingFlags invokeAttr, Binder binder, Object[] parameters, CultureInfo culture)
   at Engine.ScriptSandbox.ScriptRunning.ScriptRunner.RunScript(ScriptBase script, Boolean leaveApplicationRunning)
2024-07-17T23:49:25.4176143Z [Information]: Verifying the application has stopped.
2024-07-17T23:49:25.4332208Z [Information]: Application.Stop() of application
```

TIP: You can also find the event in the Events overview of the Test dashboard; clicking on it generates the same pop-up, providing another way to access detailed error information.

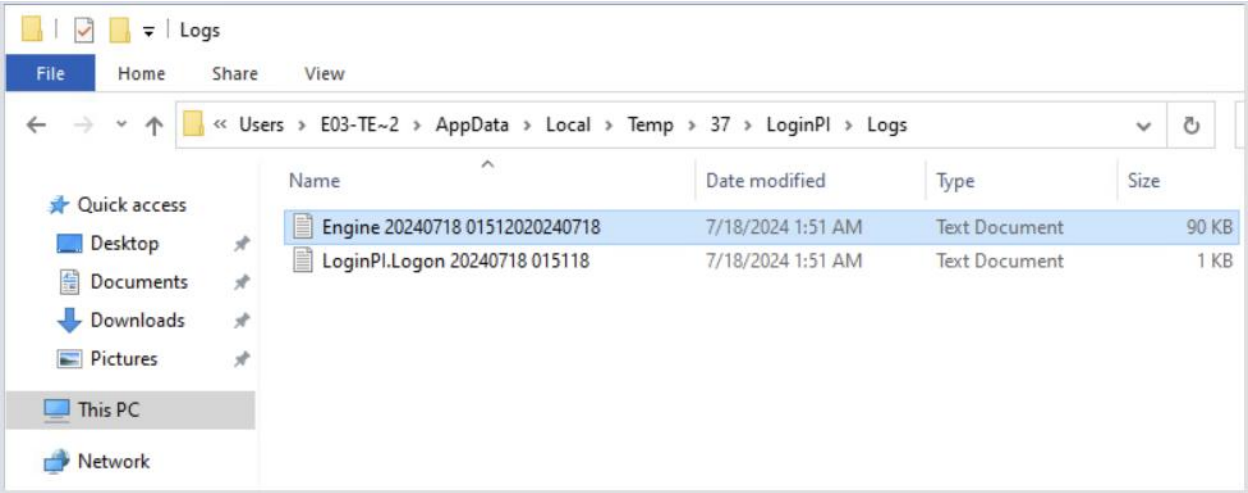
If the error occurred some time ago, you can find it in the **Login Enterprise sidebar menu** > **Continuous Testing results** > **Events** or **All Events**, which provides historical event data.

In-session troubleshooting

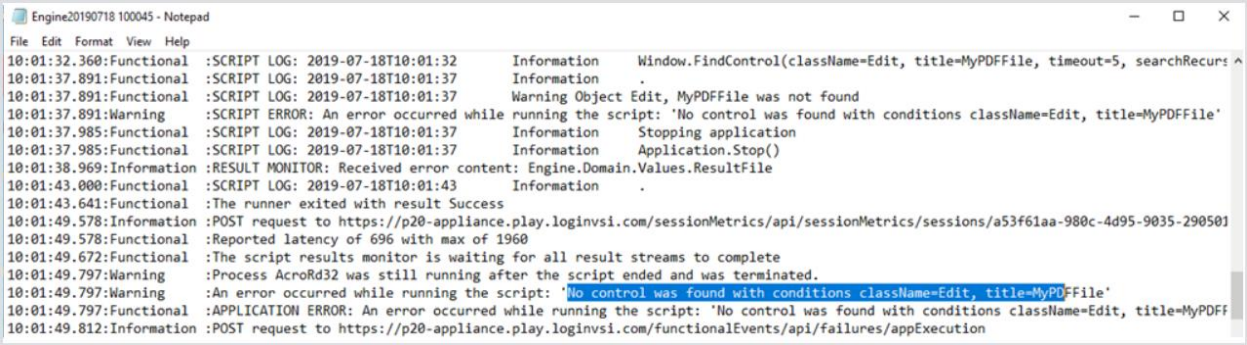
Sometimes, relying solely on screenshots isn't enough, and you may need to troubleshoot issues directly within an active session. Here's how to do it:

1. **Accessing the session**: Open the launcher responsible for starting the connection you wish to troubleshoot. Alternatively, if your environment allows for manual desktop connections, initiate the session directly from your desktop.
2. **Monitoring session activity**: Once the session is active, observe the Login Enterprise activity. You can monitor this through indicators such as a green bar.
3. **Using the pause button**: Within the session interface, look for a pause button located to the right of the green bar. Click this button to pause the session.

4. **Accessing log files**: With the session paused, navigate to the **%tmp%\LoginPI\Logs** folder on the machine where the session is running.



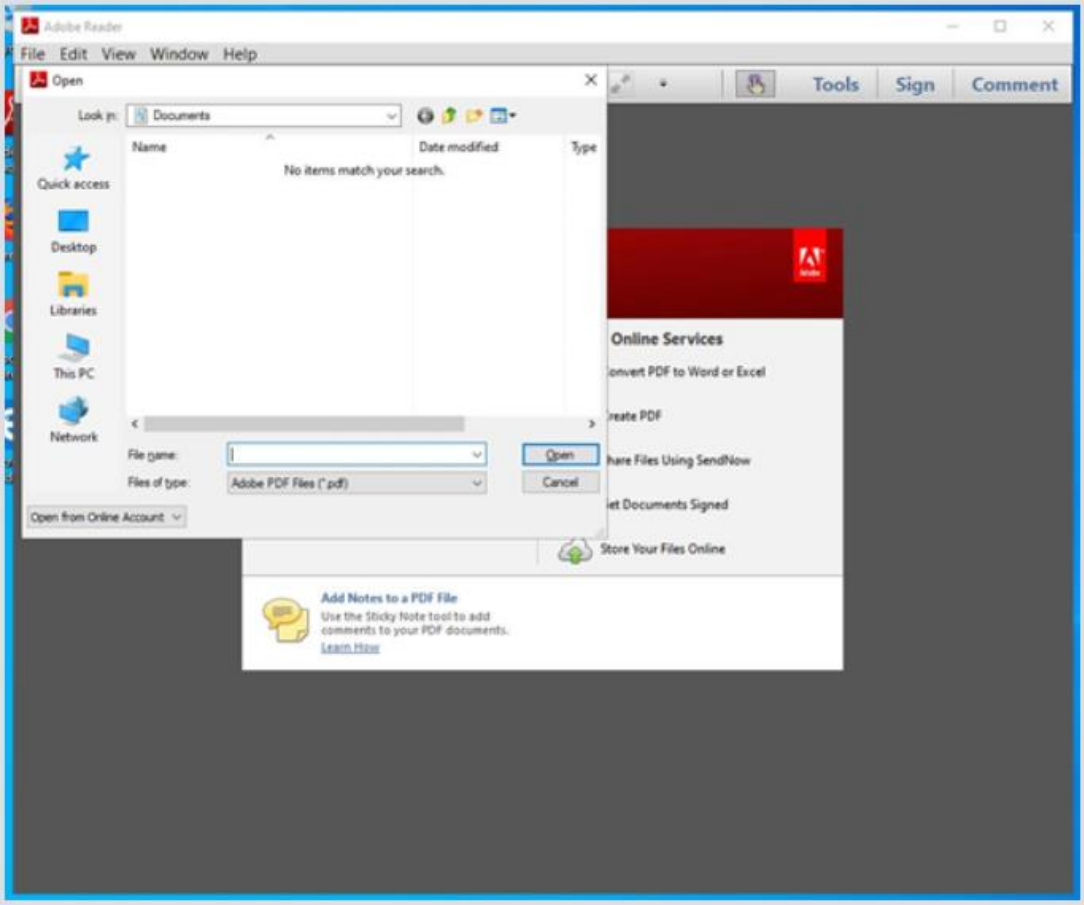
- **Sorting log files**: Sort the log files by date, arranging them from most recent to least recent to locate the one closest to the time of the Application Failure.
- **Reviewing logs and screenshots**: Each log file captures session activities, while screenshots are automatically generated upon encountering errors. They provide visual context to complement textual logs.
- **Identifying errors**: Open the log file closest to the Application Failure timestamp. Use the search function to locate the term "error," which identifies specific issues logged by Login Enterprise.
- **Analyzing errors**: Once you've identified the error message in the log file, review it to understand what went wrong. For example, it may indicate that Login Enterprise could not find an expected control or object.



NOTE: If requested by our support team, you can copy and zip the log files from the %tmp%\LoginPI\Logs folder. Send the zipped file to support@loginvsi.com for further analysis and assistance.

Once you've pinpointed the error line, you can see the specific message sent to the Login Enterprise server, along with more details. For instance, you might encounter an error indicating that a required control couldn't be found. By checking the log and moving up 12 lines from the error, you'll see it was looking for an object like "MyPDFFile". When you open the screenshot taken at that moment, you might spot where the script failed.

In the following example, the screenshot shows an open file window as expected. However, Login Enterprise couldn't find the PDF file it was supposed to load. The issue was resolved by placing the file into the user's documents folder, which fixed the Application failure.



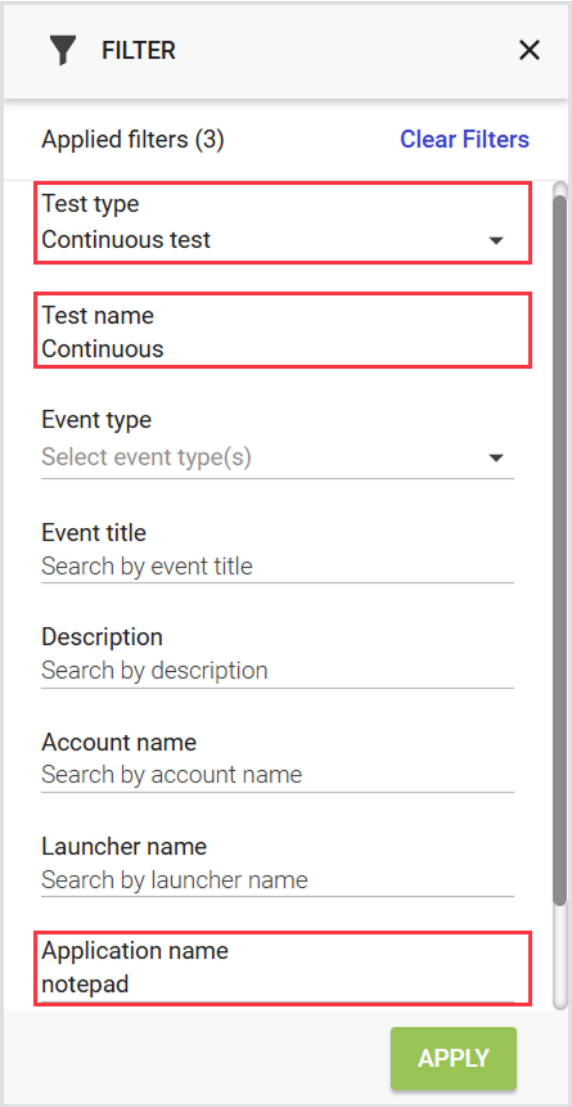
Troubleshooting Post-Test Application Errors

After a test completes, you can review application errors to diagnose issues and identify failures. You can examine errors across all events or focus on a specific test.

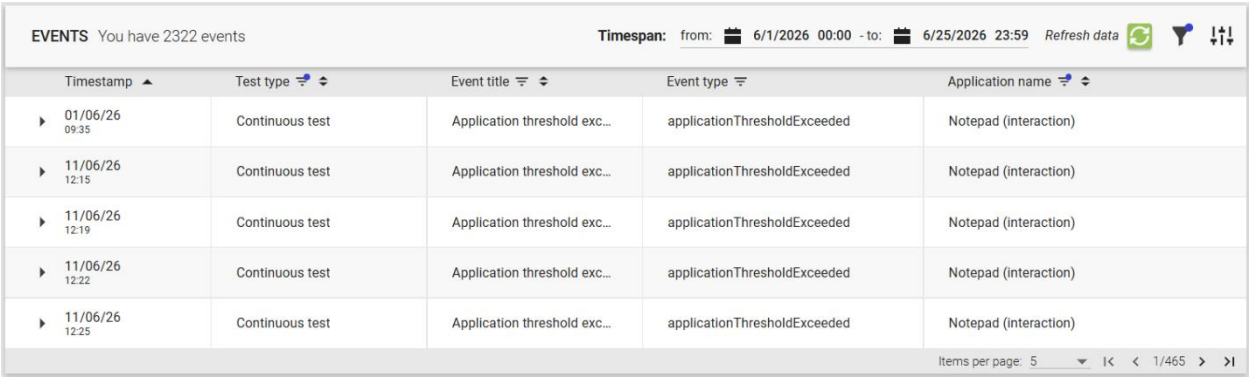
Using the All Events Page

The **All Events** page provides a centralized view of all events generated across your tests. You can use filters to isolate application-specific errors.

1. Go to the **All Events** page.
2. To restrict the view to application events, you can adjust the following filters:
    - **Test Type**: Select the specific test type you want to analyze.
    - **Test Name**: Enter the name of the test to limit the results to a specific test.
    - **Application Name**: Enter the name of your application to filter out unrelated events.



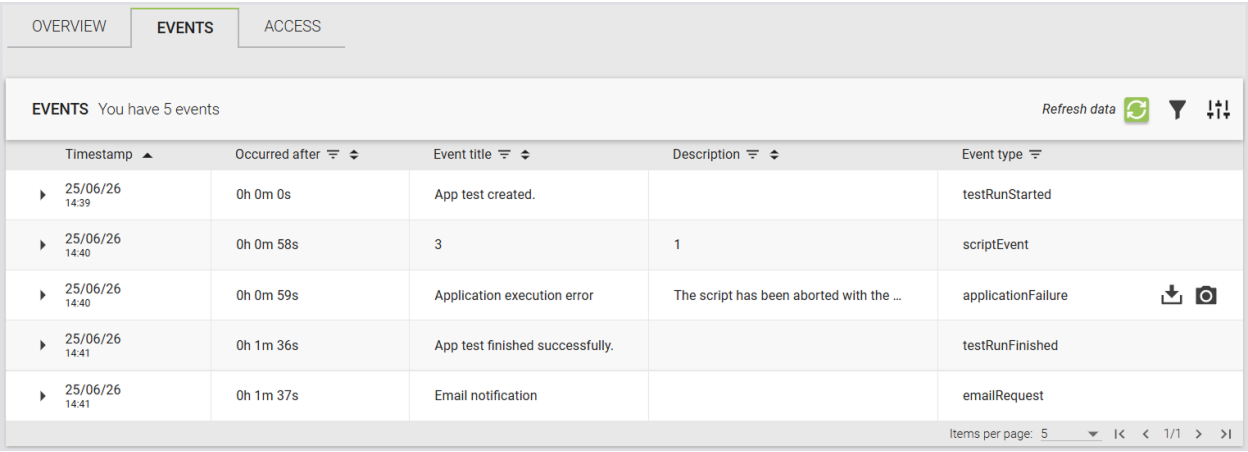
3. Review the filtered list to locate relevant application errors.



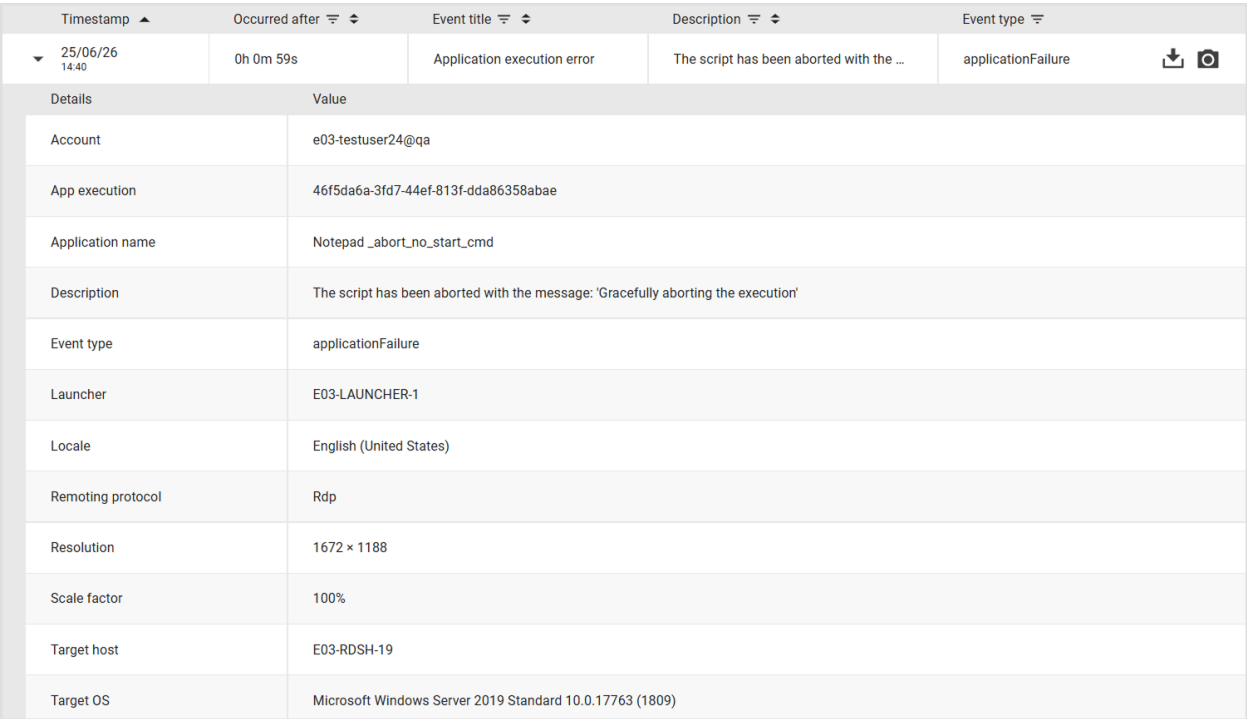
Viewing Events by Test Type

To troubleshoot errors within a specific Test run, navigate directly to that Test's details.

1. Select the test you want to review, e.g., Application Test.
2. Click **View** to open the test results details.
3. Select the **Events** tab.



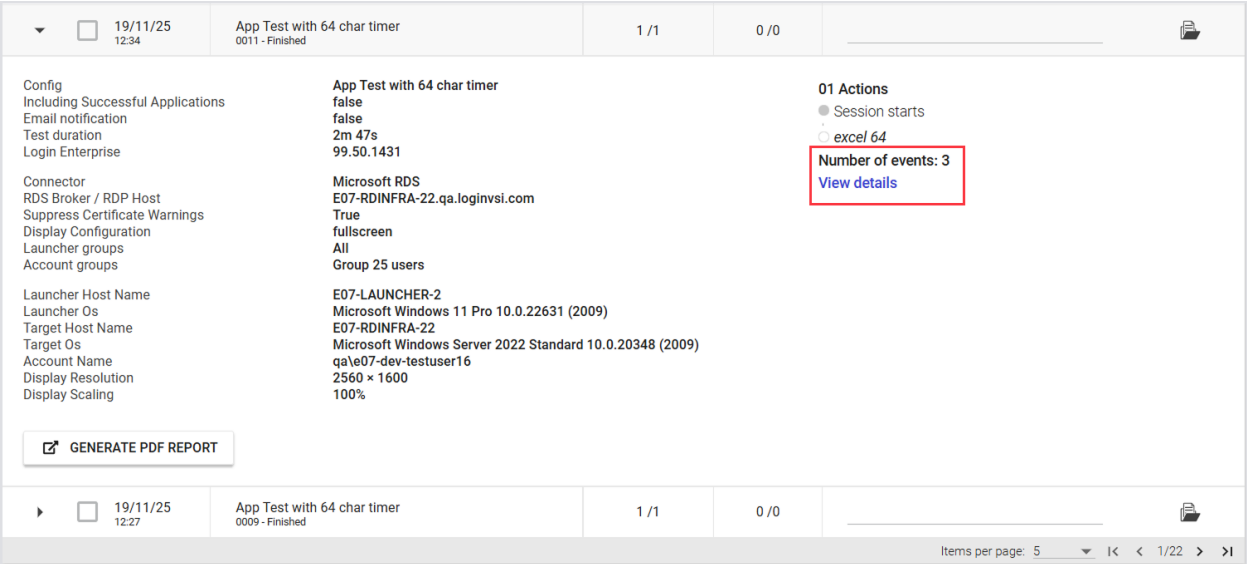
4. Locate the relevant event in the list to view its details.



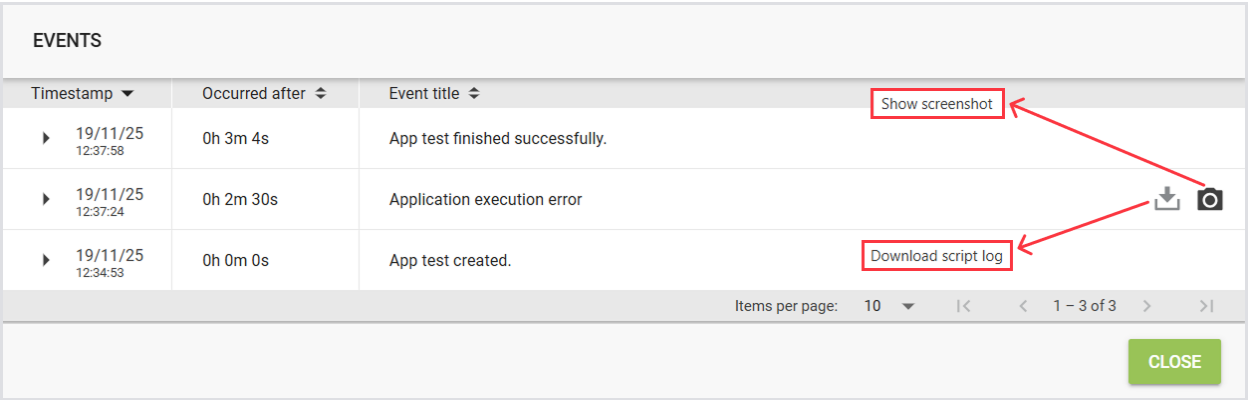
Analyzing Screenshots and Logs

When you identify an application error, use the diagnostic tools to investigate the root cause. For example, you can do so by:

1. Going to the Test results page, e.g., Application test.
2. Expanding the Test details.
3. Clicking **View Events** under Number of events. The Events window opens.



4. Download script logs and/or view application error screenshots, if available:
    1. Logs provide detailed system outputs and error codes that explain what caused the application to fail.
    2. Screenshots provide a snapshot of the application at the exact moment the error occurred. Use screenshots to visually verify UI states, error messages, or unexpected behaviors.



Although the reasons for application failures can differ, these steps will help you effectively diagnose issues in your environment.

Setting the Engine Start Timeout on a Test

- Overview
- Using the PowerShell Script
- Using the API

Overview

In the Test configuration JSON you get from the API, there is a block of timeouts:

```
"timeouts": {
  "connectionRequested": null,
  "connectionAttempted": null,
  "sessionStarted": null,
  "sessionEnded": null,
  "sessionAborted": null,
  "abortConfirmed": null
}
```

The connectionAttempted parameter governs how long the Appliance will wait for the Engine to start on a new session before declaring a login failure. This is useful when you have especially slow logins that take close to, or more than, 5 minutes.

You can edit this timeout by accessing the `/publicApi/v8-preview/tests/{testId}` endpoint and specifying the "`timeouts.connectionAttempted`" parameter.

TIP: To automate the process, use the PowerShell script. This way, you don't need to perform actions in the APIs.

Using the PowerShell Script

To use this script, download it, unblock it, and then get an API key from the Appliance web UI. To create a Configuration access token, see Adding a System Access Token.

Run the command "`get-help .\engine-start-timeout.ps1 -detailed`" for specific guidance on running this script. Examples:

List all available Tests:

```
PS> .\engine-start-timeout.ps1 -server FQDN -key KEY -action list
```

Get the currently configured timeout for a specific Test.

```
PS> .\engine-start-timeout.ps1  -server FQDN -key KEY -action get -testguid
XXX
```

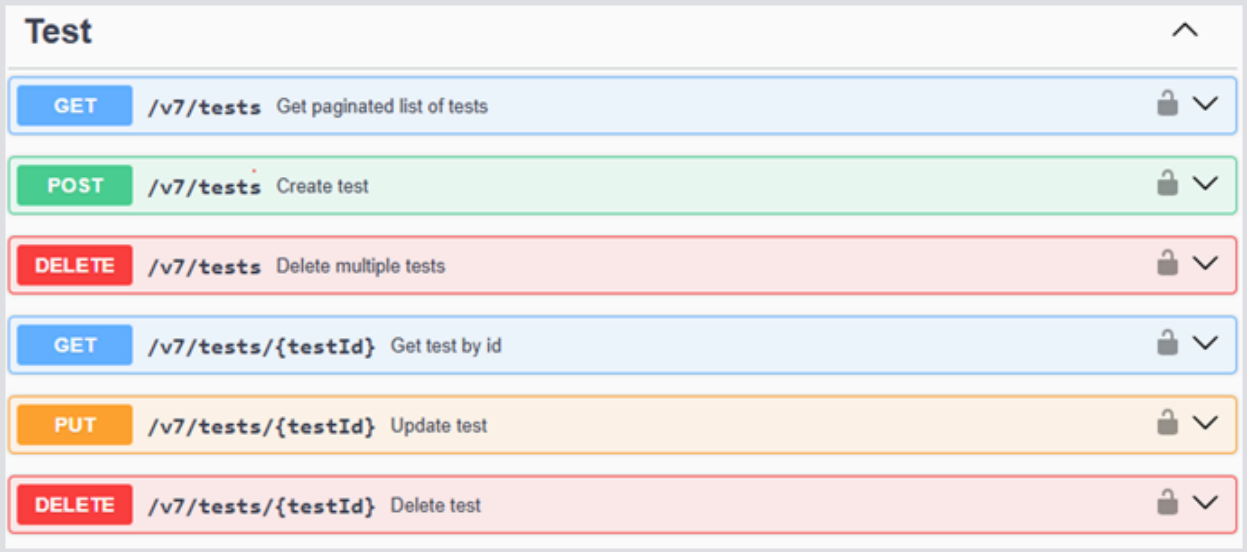
Set the timeout on the specific Test to 300 seconds.

```
PS> .\engine-start-timeout.ps1  -server FQDN -key KEY -action set -testguid
XXX -value 300
```

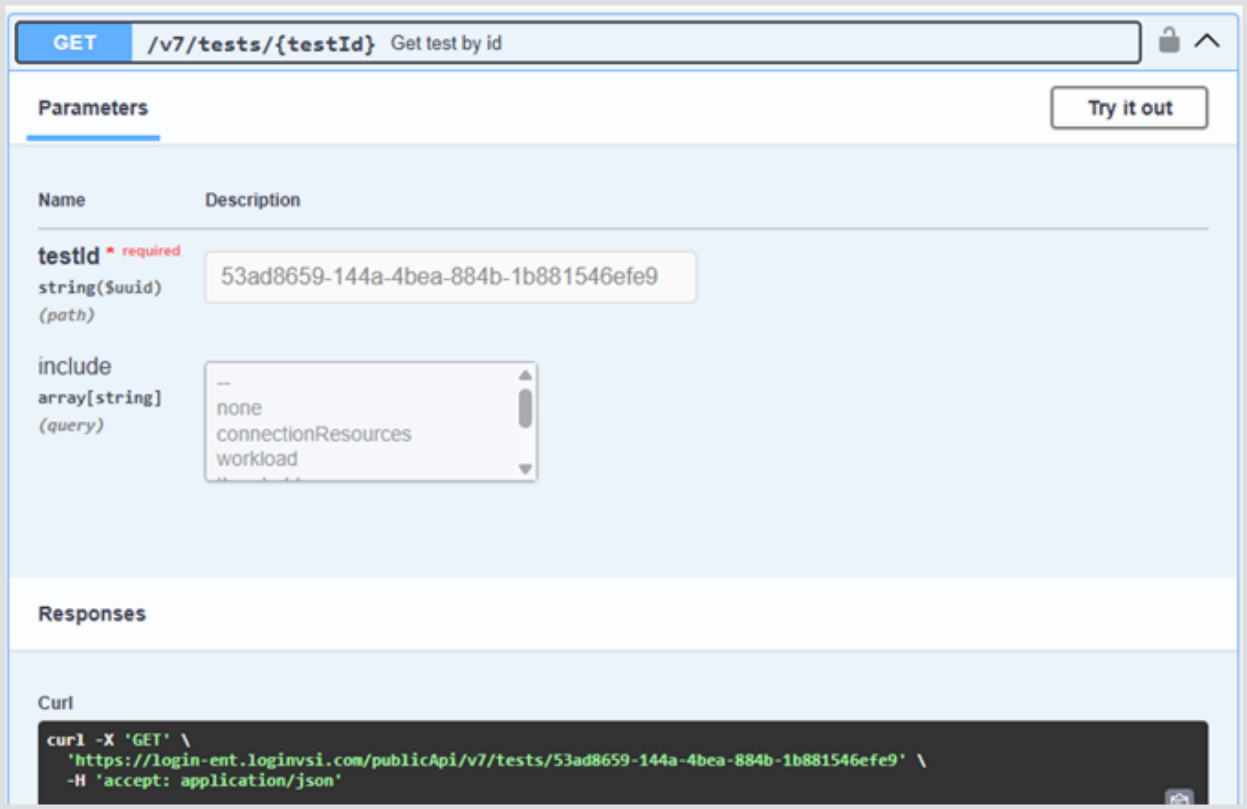
Using the Public API

If you cannot use the PowerShell script, you can use our Swagger interface to access the API. For details on accessing the Public API in Login Enterprise, see Accessing the Public API.

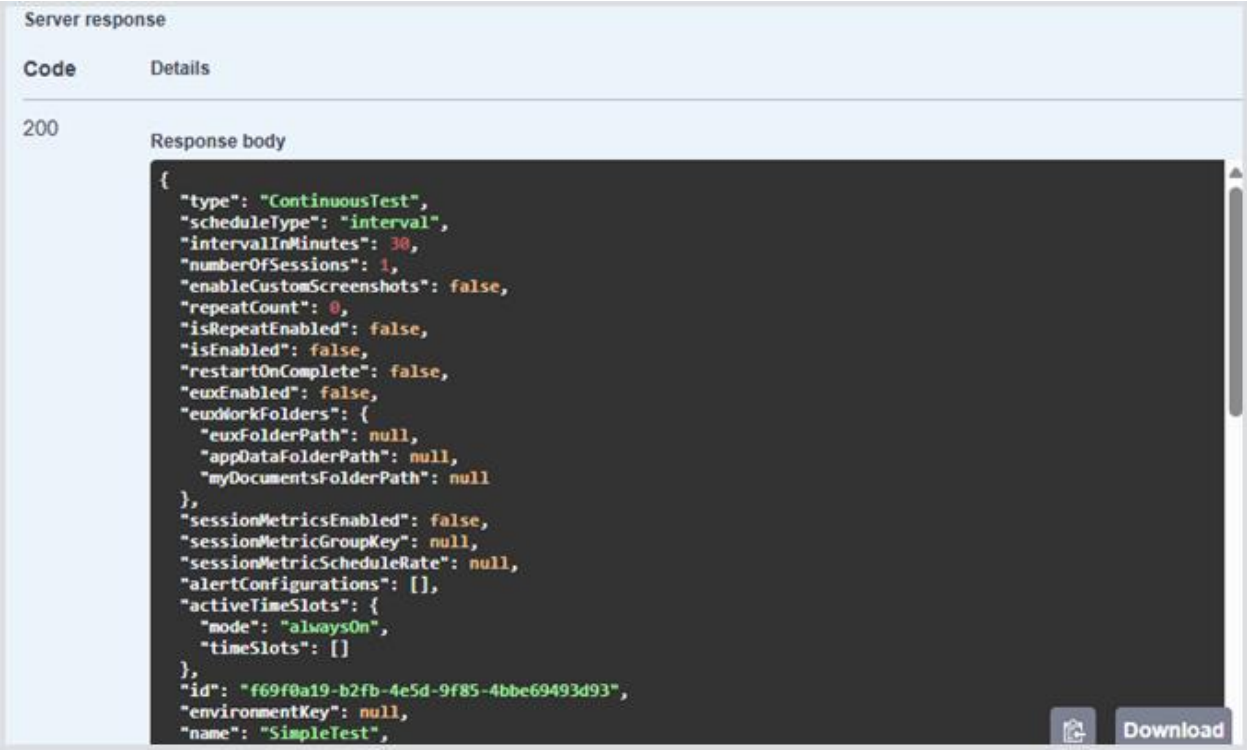
Once you have accessed the 8.0-preview API Console (Swagger) link, scroll down to the **Test** API section:



1. To modify a Test, you must first use the GET method with {testId} to get the current configuration, and then use the PUT method with {testId} to update it. To get your Test ID, modify your Test in the Login Enterprise UI. For this:
    1. In the **Login Enterprise sidebar menu > Configuration > Manage tests**, select the type of Test, and click on your specific Test to modify it.
    2. Then, look at your location bar at the top of the browser. Your URL will look like this: https://login-ent.loginvsi.com/configuration/tests/**f69f0a19-b2fb-4e5d-9f85-4bbe69493d93**/settings
    3. The GUID in that URL is the ID of your Test. Copy it into your clipboard.
2. Click on the second GET method, the one that includes testId, and open it as follows:



3. Click **Try it out** to enable the fields.
4. In the **testId** field, paste your test ID.
5. Leave the **include** options untouched.
6. Scroll down and click **Execute**.
7. In the **Server response** block, you should have a return code of 200 and the JSON body of your Test configuration:

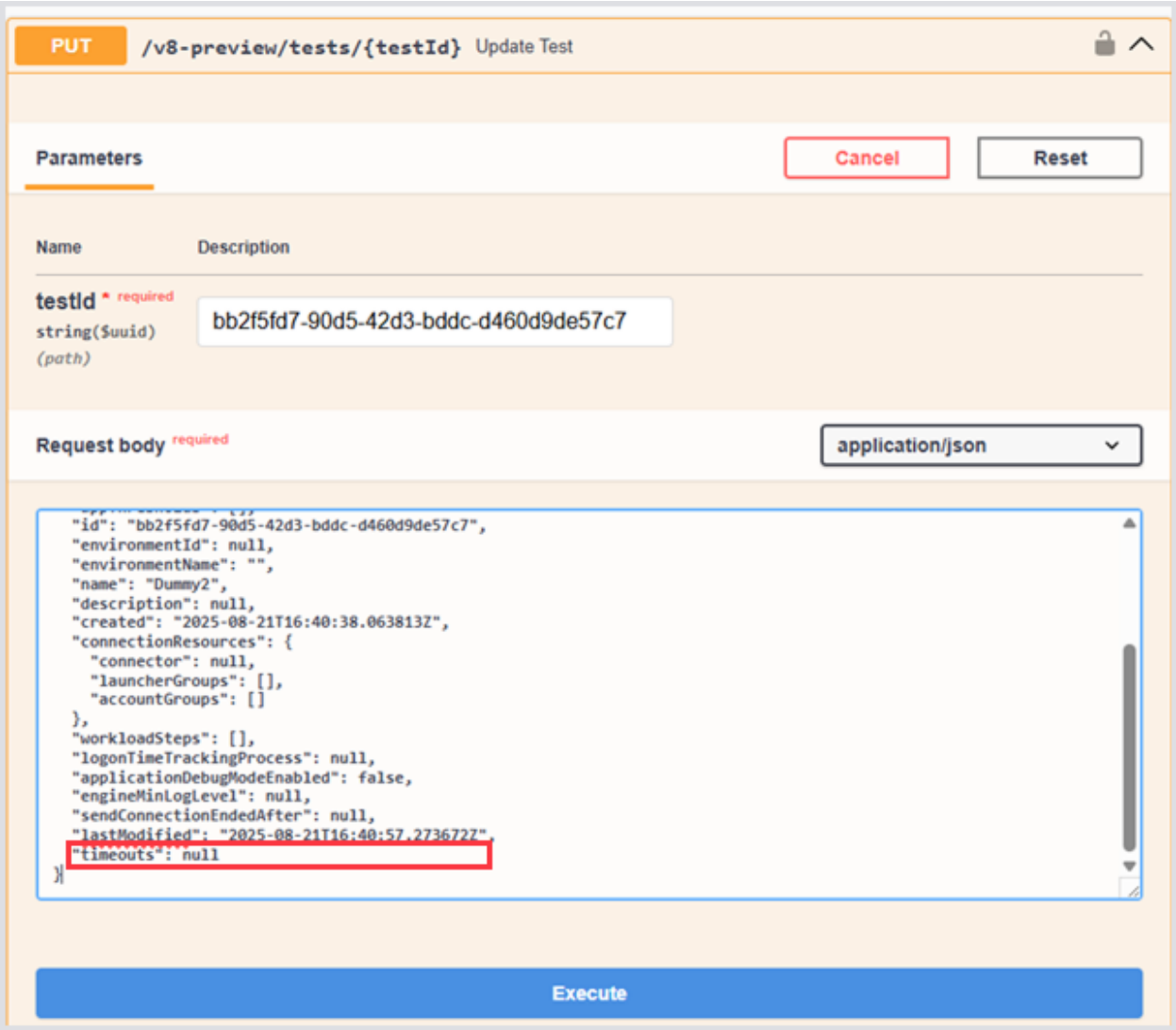



```
{
  "type": "ContinuousTest",
  "scheduleType": "interval",
  "intervalInMinutes": 30,
  "numberOfSessions": 1,
  "enableCustomScreenshots": false,
  "repeatCount": 0,
  "isRepeatEnabled": false,
  "isEnabled": false,
  "restartOnComplete": false,
  "euxEnabled": false,
  "euxWorkFolders": {
    "euxFolderPath": null,
    "appDataFolderPath": null,
    "myDocumentsFolderPath": null
  },
  "sessionMetricsEnabled": false,
  "sessionMetricGroupKey": null,
  "sessionMetricScheduleRate": null,
  "alertConfigurations": [],
  "activeTimeSlots": {
    "mode": "alwaysOn",
    "timeSlots": []
  },
  "id": "f69f0a19-b2fb-4e5d-9f85-4bbe69493d93",
  "environmentKey": null,
  "name": "SimpleTest",
```


NOTE: The Copy button copies the entire JSON output into your clipboard. Don't do that yet. Be careful not to scroll too far down to the Responses section. That section lists possible response examples, and not your actual Test configuration.

8. Click on the **PUT** method with {testId} to open it up, and then click **Try it out**.
9. In the **testId** field, paste your testId.

10. Go back to the results of your GET and use the **Copy** button to copy the JSON into your clipboard.
11. Return to the PUT method and paste it into the **Request Body**, replacing the data that is there.
12. Scroll down to the bottom of the JSON you just pasted and look for the "timeouts" entry. If this Test has never had connectionAttempted set, "timeouts" will be null:



Otherwise, it will list a set of timeouts as follows:

13. If your “timeout” block is null, change it to include just your new timeout as follows:

```
"timeouts": {
  "connectionAttempted": 0
```

```
}
```

14. If the block already exists, update the value you want. Ignore the other settings.

IMPORTANT: If you are modifying a Load Test or a Continuous Test, you must make one other modification.

15. Look for the following setting:

```
"sessionMetricDefinitionGroupId": "...GUID...",
```

16. If the setting exists and is not “null”, you must change the field name to sessionMetricGroupId:

```
"sessionMetricGroupId" ": "...GUID...",
```

17. Click **Execute**. You should see a “200” response.
18. Scroll back up to the GET with {testId} section and click **Execute** again. You should see your updated value there. The next Test execution will use this new timeout value.

Back to Managing Tests

Dashboards and Reporting

This chapter covers how to review the outcomes of your Tests, including graphical measurements and event data generated during execution. The Virtual Appliance dashboard offers a comprehensive view of ongoing Test statuses, relevant metrics, and important messages. Additionally, the results pages provide detailed insights into each Test run, allowing you to examine specific measurements and events for thorough analysis.

Key features:

- **Dashboard overview:** Monitor the real-time status of ongoing Tests, view key metrics, and access important notifications.
- **Results pages:** Explore detailed results for each Test run, including comprehensive measurements and event logs.
- **Data visualization:** Use graphs and charts to visualize Test performance and identify trends or issues.
- **Event tracking:** Analyze events generated during Tests for deeper insights and enhanced troubleshooting.

The chapter includes the following topics:

- Dashboards
  - Test Dashboard
  - Operations Dashboard
- Test Results
  - Viewing Continuous Testing Results
  - Viewing Load Testing Results
  - Viewing Application Testing Results
- Viewing All Events
- Performance Measurements
  - Calculating the EUX score and VSImax
  - Monitoring Login Performance
  - Viewing Session Metrics

Dashboards

The Dashboard lists the progress for all currently running Tests, including in-progress data about timers, failures, logins, and Applications. Any Tests currently enabled and running will show up in this list. You can select to view all Tests or only Continuous, Application, or Load Tests.

This sub-chapter includes the following topics:

- Test Dashboard
- Operations Dashboard

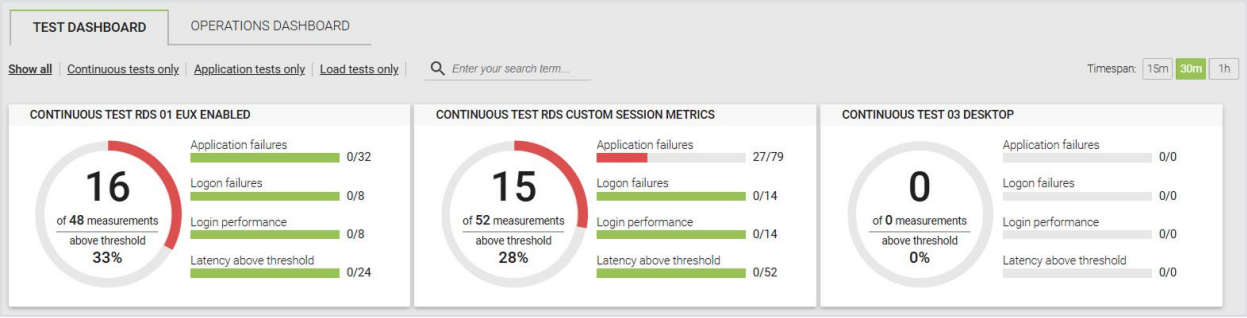
Test Dashboard

- Overview
- Test Run view
  - Login performance
  - Latency
  - Continuous Testing results
- Location view

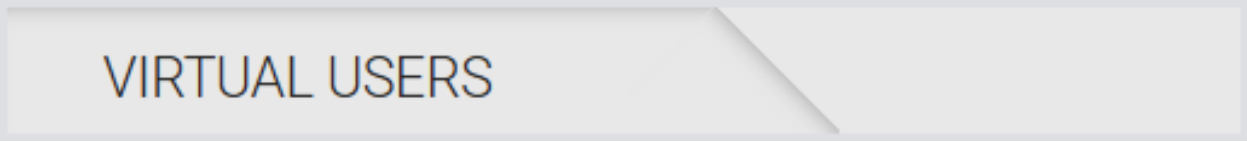
Overview

The Test Dashboard provides an overview of login failures and login performance metrics. This allows you to quickly identify issues and monitor the efficiency of the login process. The Dashboard only shows the status for currently enabled and running Tests. Once a Load Test or Application Test is completed, it will no longer show on the Dashboard.

NOTE: A Continuous Test is only visible when it's enabled.

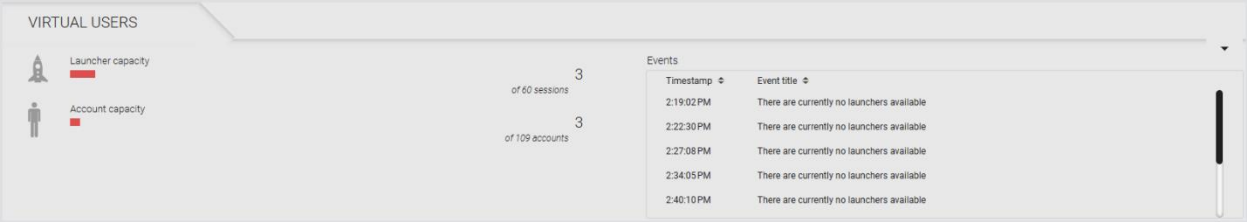


At the bottom of the page, you can also click to expand the Virtual Users section.



This displays information about Launcher capacity and Account capacity. Events are filtered, and only the following types are shown:

- LauncherOffline
- LauncherCapacityExceeded
- AccountCapacityExceeded
- AccountDisabled

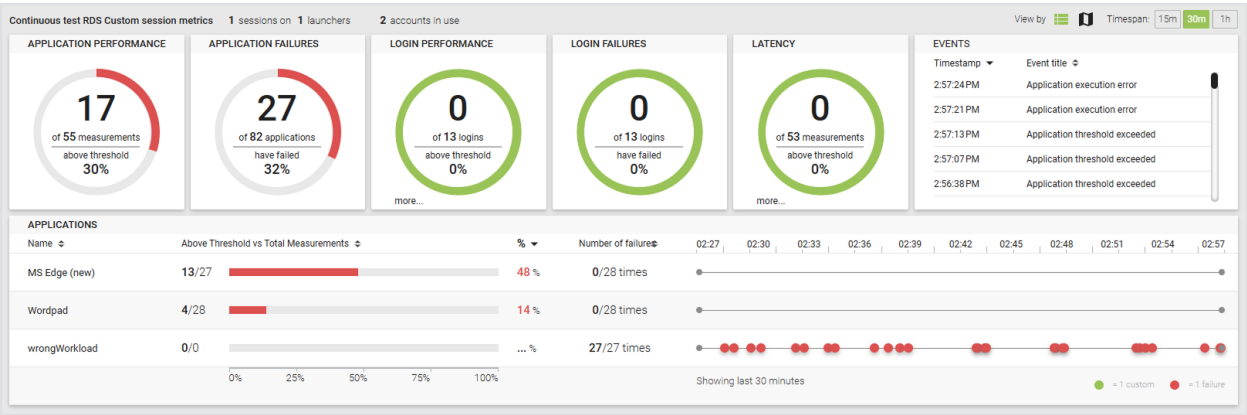


For more information on Virtual Users, see Virtual Users.

Test Run View

Within the Test Run, you can further analyze login performance data and login failures separately. This breakdown allows for a more detailed investigation of each aspect, helping you pinpoint specific issues and optimize the login process.

To get more details on a particular Test Run, click on its tile in the Overview. You can limit your view to specific locations, if you have set locations for your Launchers, by selecting them from the pulldown at the top of the page.

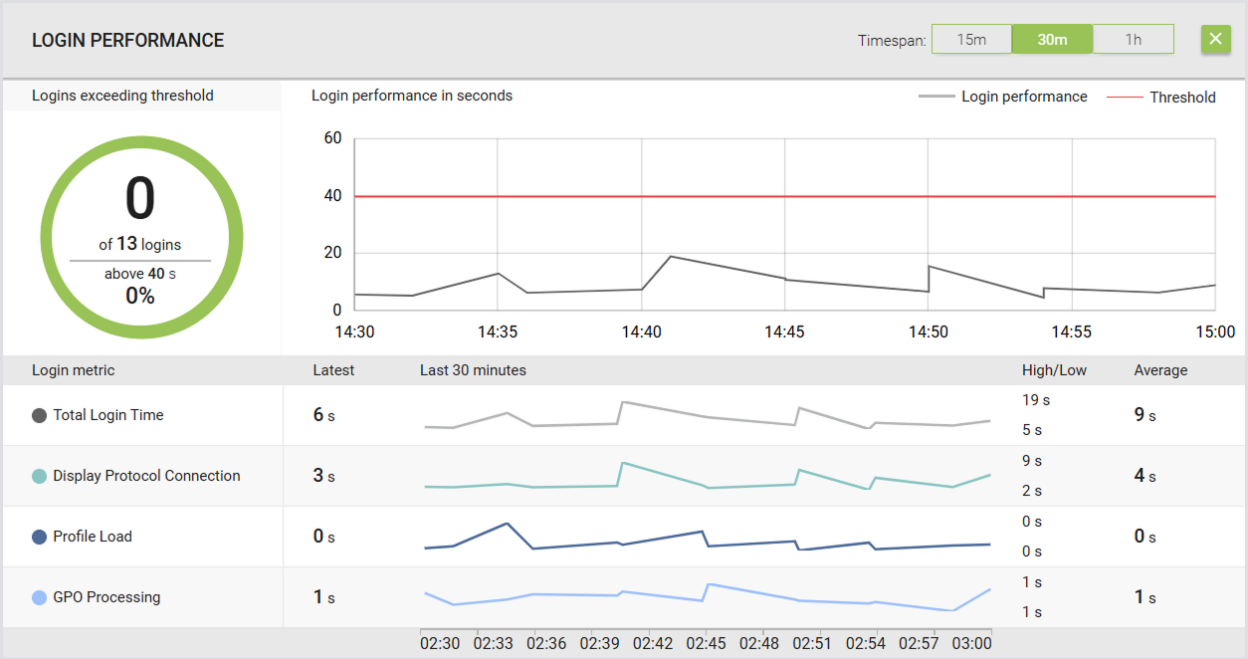


Login Performance

This is where you can view a breakdown of various components in the logon process, along with our threshold marker measured in seconds. Additionally, you can refine your view by selecting different time spans, such as:

- 15m
- 30m
- 1h.

NOTE: The Logon failure data will only appear if the system has encountered any logon failures.



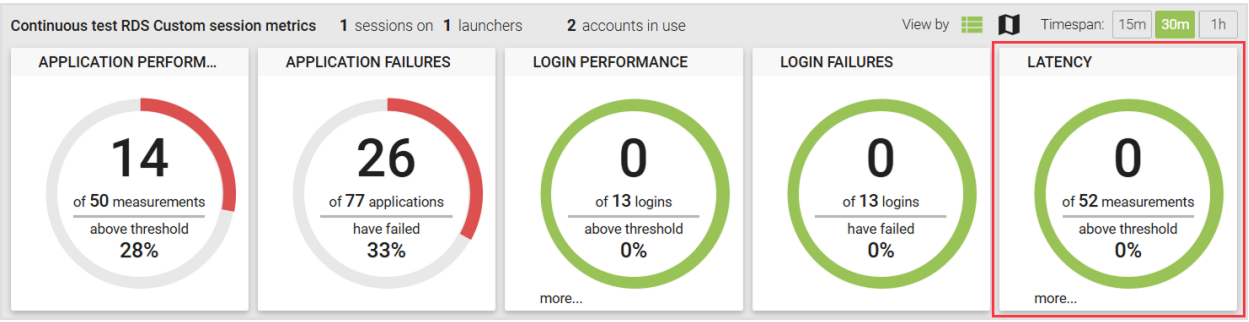
To learn more about the Login Performance, see [Monitoring Login Performance](#).

Latency

Latency measurements are crucial for understanding the performance of your applications. By configuring and monitoring these measurements, you can identify performance issues and ensure your Test Run operates efficiently.

Once you have configured Latency measurements and set a threshold, you can view the data on the dashboard. The main dashboard will display Latency above threshold alongside other key metrics.

TIP: Click on the Latency tile to interact with the Latency data.

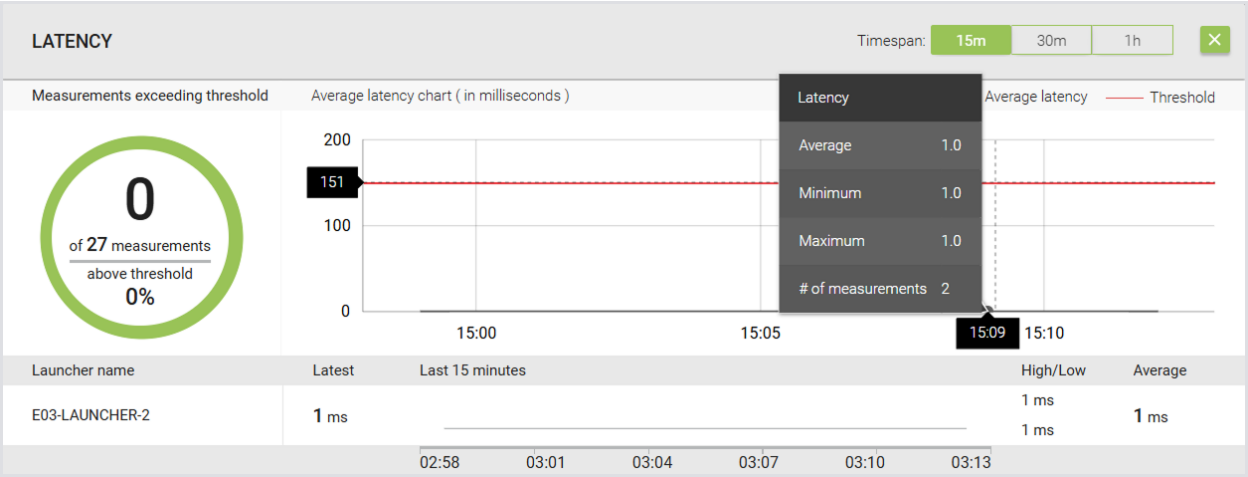


This is where you can view detailed activity and measurements over time. This includes:

- Launcher name
- Threshold

- Number of measurements.

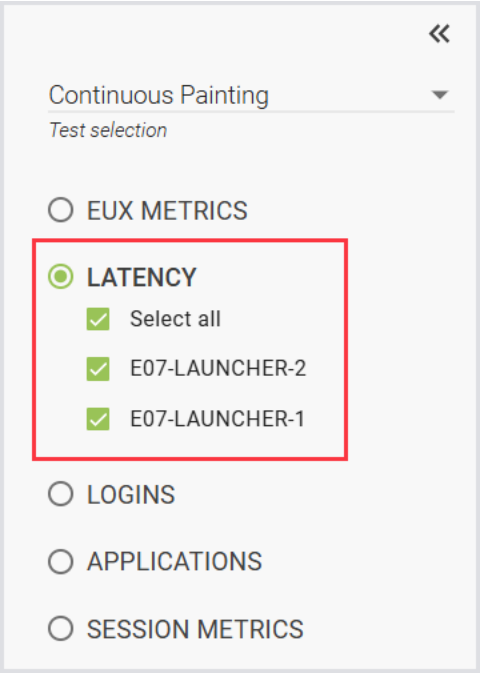
TIP: Hover over a Latency measurement in the Latency popup window to see a tooltip with precise measurement details.



Continuous Testing Results

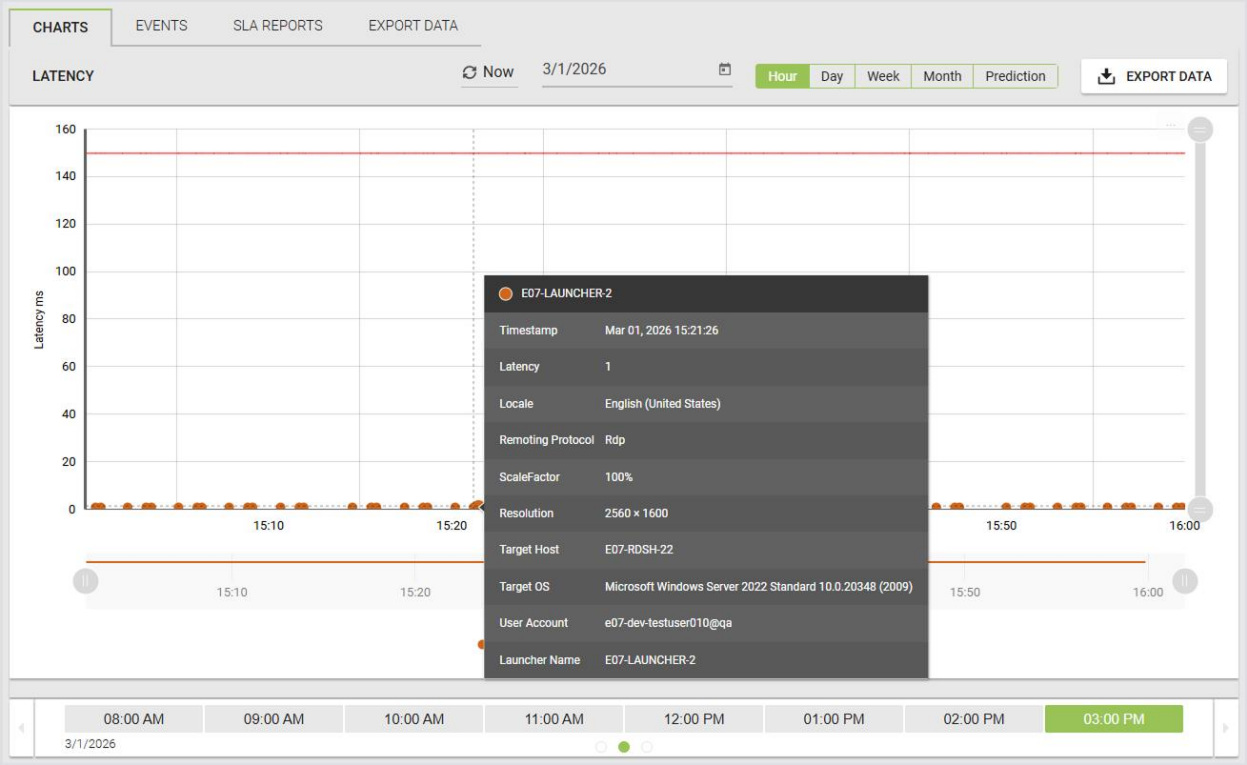
This is where you can view collected Latency measurements.

TIP: Select Latency measurements and Launchers set in the Test Run to see their specific Latency data.



The Latency chart operates similarly to other application charts. You can view Latency in milliseconds and adjust the time range to an Hour, Day, Week, or Month.

TIP: Use the Prediction to forecast future Latency trends.

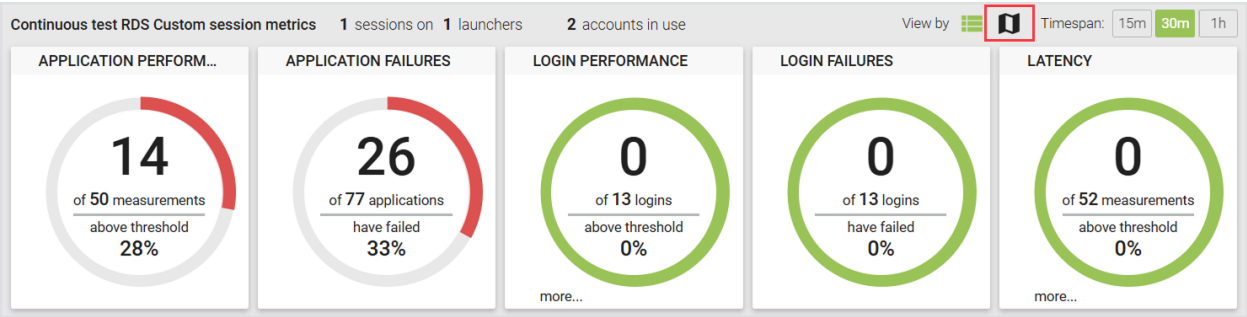


By following these steps, you can efficiently monitor Latency measurements through both the dashboard and reporting tools. Regular analysis of this data helps maintain optimal performance and address potential issues proactively.

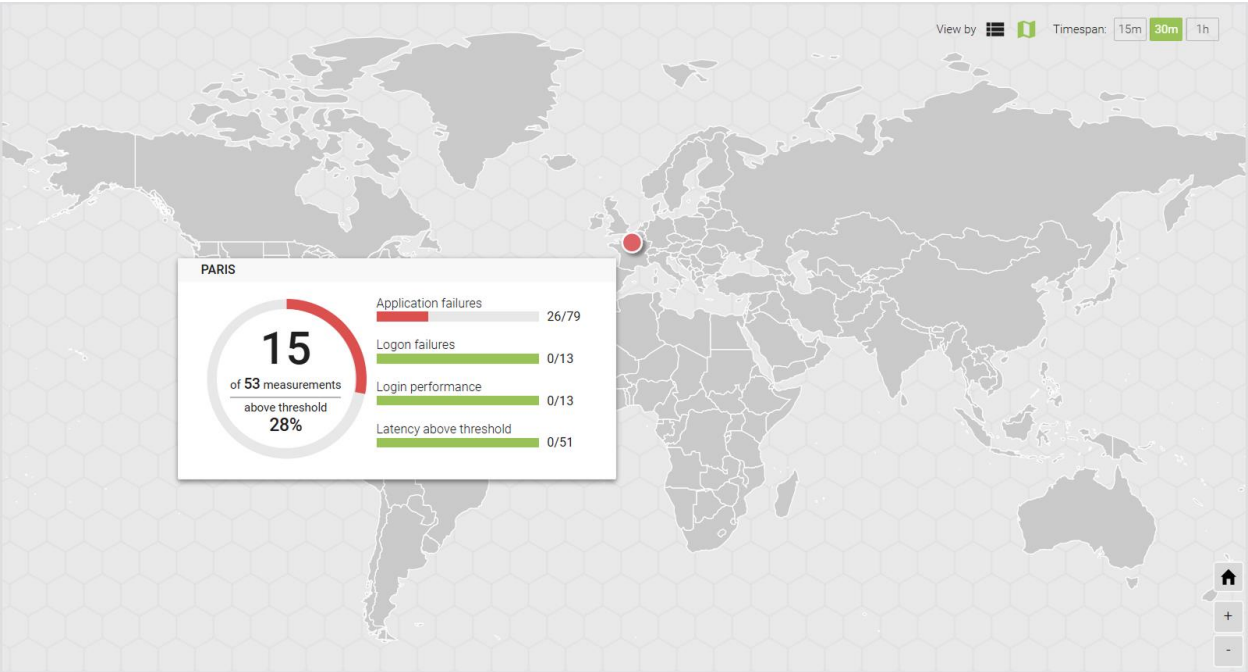
For more information on Latency, see Monitoring Latency.

Location View

If Launchers are attached to configured locations when a Continuous Test type is in progress, these Launchers will be visible in the map view. To access the map view, click the Map icon at the top of the Login performance page.



Now, you can view the Launcher location plotted on the world map. The hoverable tooltip shows connection and Test successes and failures.



For more details on Locations, see [Configuring Locations](#).

Operations Dashboard

- Overview
- Accessing the Operations dashboard
- Cost dashboard
- Capacity dashboard
- Performance dashboard

Overview

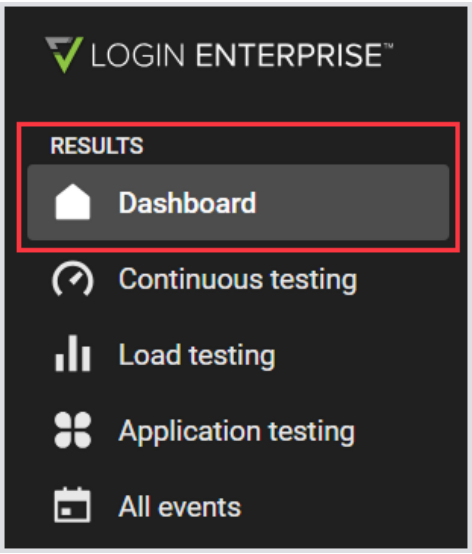
The Operations Dashboard allows you to monitor the cost and capacity of your production VDI/DaaS environments.

You can view a plethora of graphs and charts that provide real-time visualizations of the collected data. As a result, you can gain enhanced visibility into system performance, quickly identify and resolve issues, and make informed, data-driven decisions based on the valuable insights these dashboards provide.

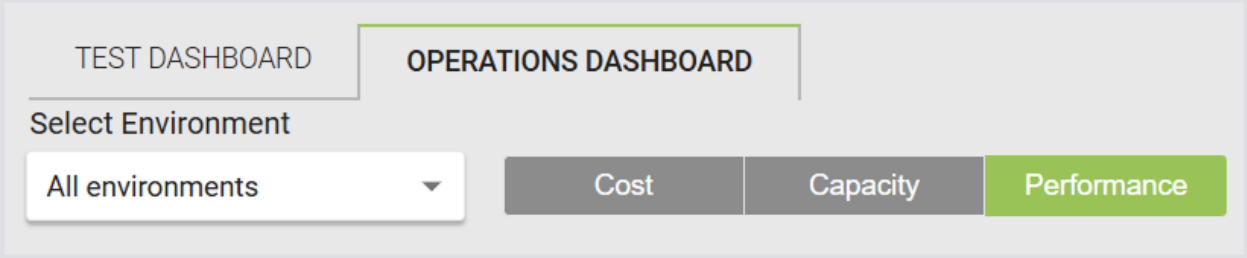
Switch between the two dashboards, select a single environment, or view the aggregated data from all of them in read-only mode. Each Dashboard also provides a way to select the desired time range, refresh the dashboard, and zoom in or out of the view. You can find these options in the top-right corner of the Dashboard, as shown below.

Accessing the Operations dashboard

1. In the **Login Enterprise sidebar menu** > **Results**, navigate to **Dashboard**.



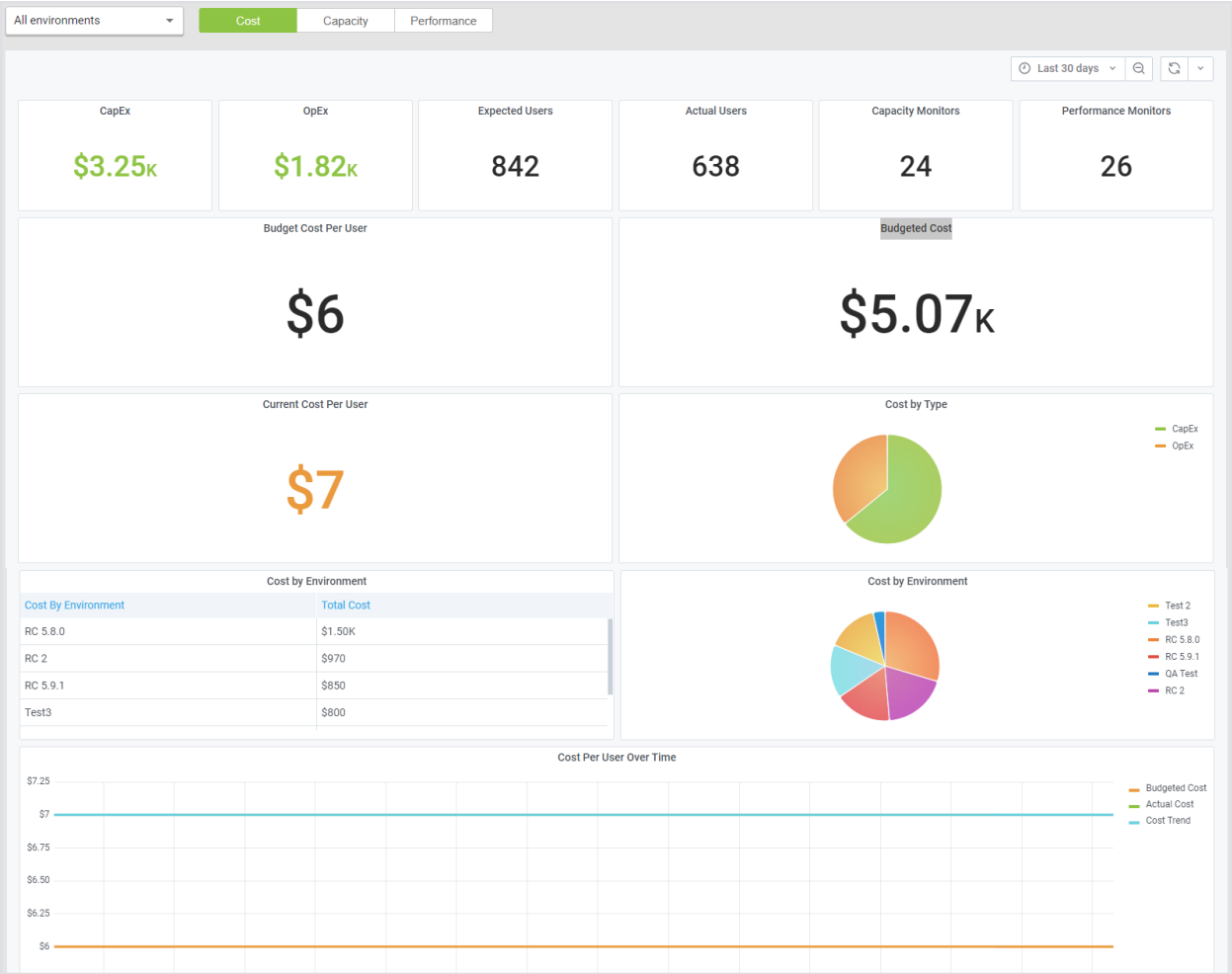
2. In **Dashboard**, click the **Operations dashboard** to view your environments' Cost, Capacity, and Performance data.



Cost dashboard

The Cost Dashboard provides insights into the cost considerations of the selected environment, specifically:

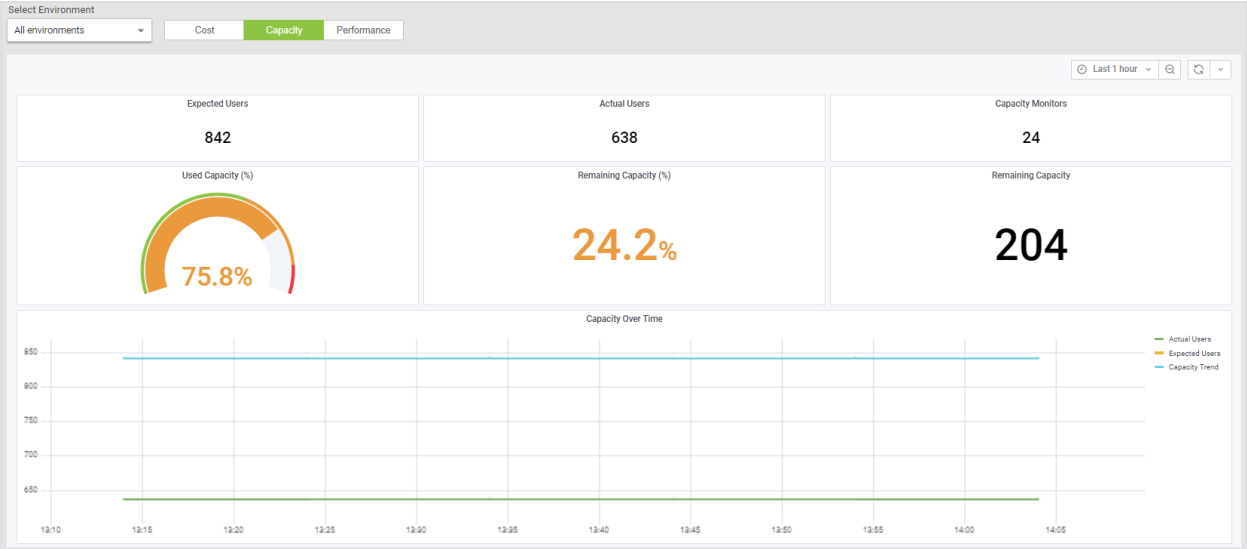
CapEx	OpEx	Expected Users	Actual Users	Capacity Monitors	Performance Monitors
The total capital expenses associated with the environment.	The total operational expenses associated with the environment.	The anticipated maximum number of users (VSIMax)	The current actual number of users utilizing the environment.	The number of Load Tests assigned to the environment to monitor its capacity.	The number of Continuous Tests assigned to the environment to monitor its performance.
Budgeted Cost Per User			**Budgeted Cost**		
The budgeted cost divided by the number of expected users, indicating the projected cost per user.			The combined cost comprising Capital Expenditures (CapEx) and Operational Expenditures (OpEx).		
Current Cost Per User			**Cost By Type**		
The budgeted cost divided by the actual number of users, representing the current cost per user.			The percentage breakdown of costs between Capital Expenditures and Operational Expenditures.		
Costs By Environment			**Costs By Environment**		
The total cost of each environment as a percentage of the total costs.			The total cost of each environment as a percentage of the total costs.		
Cost Per User Over Time					
Trends showing the cost per user over time, including budgeted and actual users.					



Capacity dashboard

The Capacity Dashboard presents the current status of the selected environment's capacity with the following details:

Expected Users	**Actual Users**	**Capacity Monitors**
Number of Expected Users	Number of Actual Users	The number of Load Tests assigned to the environment.
Used Capacity	**Remaining Capacity (%)**	**Remaining Capacity**
Ratio of Actual Users to Expected Users	Percentage of remaining capacity (100 - Used Capacity %)	Expected Users minus Actual Users
Capacity Over Time		
Trends depicting Expected Users and Actual Users over time.		

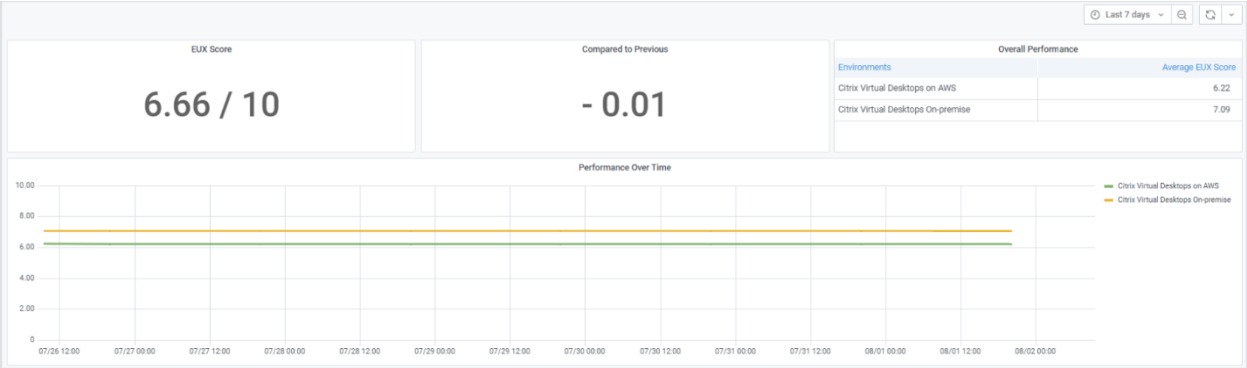


Performance dashboard

The Performance Dashboard offers insights into the performance of various environments during the selected time period. By leveraging the EUX scores from Continuous Tests, the graphs demonstrate whether performance has improved or declined over time.

EUX Score	Compared to Previous	Overall Performance
Displays the average EUX Score of Continuous Tests assigned to the selected environment over a given time frame, whether it's a few minutes or several years. However, if no new tests have been running during that time frame, the graph may show "No data." When all environments are selected, the general average EUX Score is calculated by adding up the average EUX Scores of each environment.	Shows how the EUX Score has improved or declined. If the user selects a time frame of 5 minutes, the current EUX Score of the past 5 minutes will be compared to the preceding 5 minutes. For example, suppose the current time is 10:05, and the user selects a time frame of 5 minutes. In that case, the current EUX average will be calculated using data from 10:00 to 10:05, and the previous EUX average will be calculated using data from 09:55 to 10:00.	Provides a tabular view of average EUX Scores. When "All Environments" is selected, a list of all environments with their average EUX Score is shown. When a specific environment is selected, the table will contain a list of all Continuous Tests that are assigned to the specified environment, with their average EUX Score.
Performance Over Time		

Shows a graphical view of average EUX Scores. When "All Environments" is selected, each environment's EUX Scores over a selected period of time will be shown. When a specific environment is selected, each Continuous Test's EUX Score over a selected period of time will be shown.



Test Results

This subchapter helps you explore detailed results for each Test run, including comprehensive measurements and event logs. Utilize graphs and charts to visualize Test performance, helping you identify trends and issues effectively.

The subchapter covers the following topics:

- Viewing Continuous Testing Results
- Viewing Load Testing Results
- Viewing Application Testing Results

Aside from the Login Enterprise user interface, you can also retrieve Test results via the Public API. For more information, see Using the Public API.

Viewing Continuous Testing Results

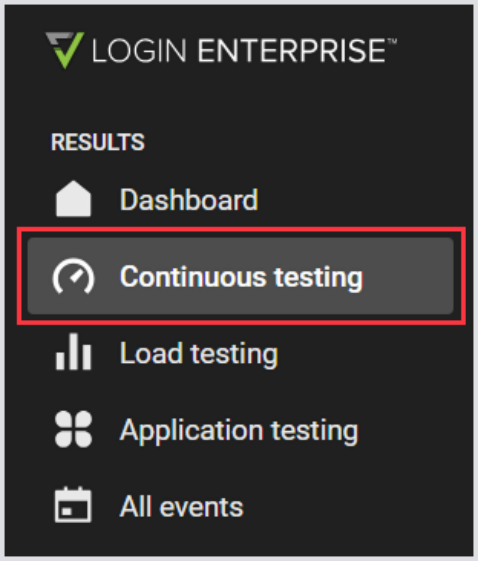
- Overview
- Single Test
  - Single Test Summary
  - Single Test Results
    - Application Performance
  - Generating a Test PDF Report
- Compared Tests
  - Compared Test Summary
  - Compared Test Results
  - Aggregation Options
  - EUX Metrics
  - Latency
  - Logins
  - Applications
  - Session Metrics
  - Application Performance
- Events
  - Possible Events
  - Generating a Test PDF Report
- Configuring a Test PDF Report
- Test PDF Report Breakdown
- SLA Reports
  - Benefits of SLA reporting
  - Configuring SLA Reports
  - Generating SLA Reports
  - Scheduled Reports
- Export Data
  - RawData.zip File
  - EUX Raw Data
    - Retrieving the EUX Raw Data
    - Cleaning the EUX Data (Data Retention)
- Tips and Tricks
  - Chart Legend
  - Chart Control
  - Downloading Charts as an Image

Overview

You can view the results of individual Continuous Tests and compare up to five Continuous Test instances. The Continuous Test Results page provides a comprehensive overview of measurements, including charts, graphs, and detailed events for each Test. You can also configure and download SLA reports and view Events. See the instructions below for more details.

Single Test

You can start looking at the Continuous Test results at any time. For this, in the **Login Enterprise sidebar menu** > **Results**, navigate to **Continuous testing**.

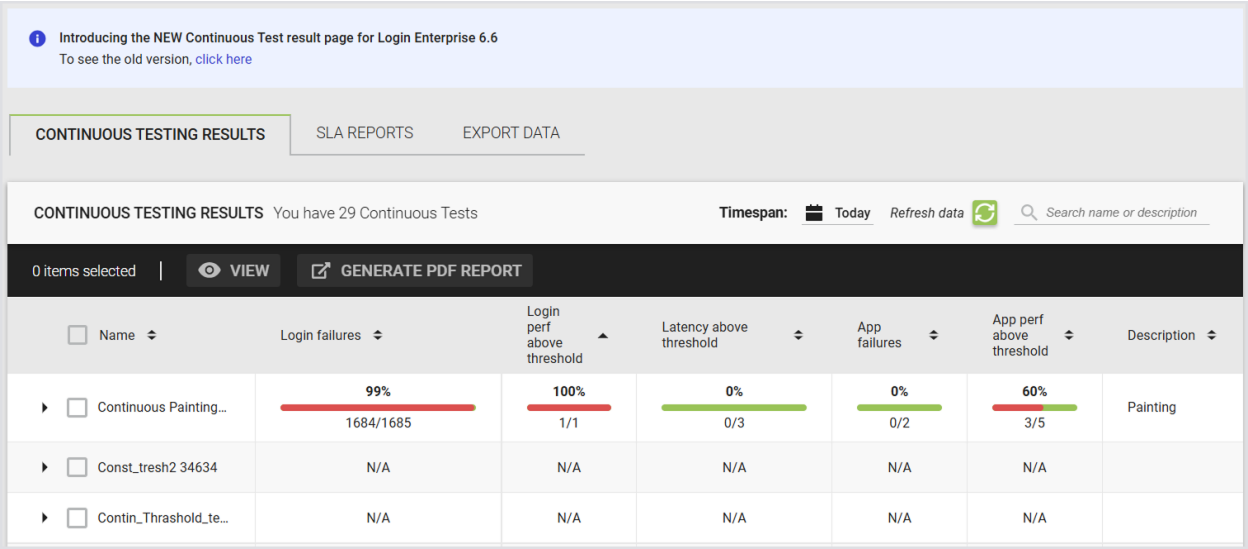


Single Test Summary

This page displays a list of previously executed Continuous Tests. The table is sorted by default on "Login failures," and includes the following:

NOTE: An info box at the top of the page includes a link to the previous (legacy) results page for reference.

Name	Login failures	Login performance above threshold	Latency above threshold	Application failures	App performance above threshold	Description
Name and sequence number of the Test.	Number of login failures (displayed as x out of y) and corresponding percentage.	Indicates how many login performance measurements exceeded the defined threshold.	Shows how many latency measurements exceeded the acceptable threshold.	Number of application-level failures detected during test execution.	Number of application performance metrics exceeding defined thresholds.	Description defined in the test configuration.

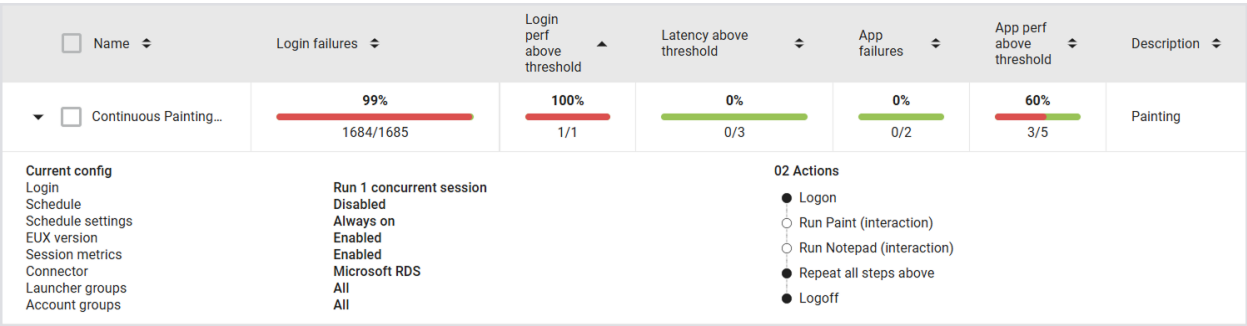


All results displayed on the page are filtered based on the selected time span. Use the date picker in the top right corner to select any time range. **Default setting: Last 1 hour**.

NOTE: If the selected time span includes a future time range, a **Refresh** button appears on the Overview page and Result pages. Click **Refresh** to reload the page and update the results with the latest available data.

Search results dynamically filter the table content. Use the search field to find tests by **Test Name** and/or **Description**.

TIP: Click on a test to open its details and review the current test configuration. Historical configurations are not shown.



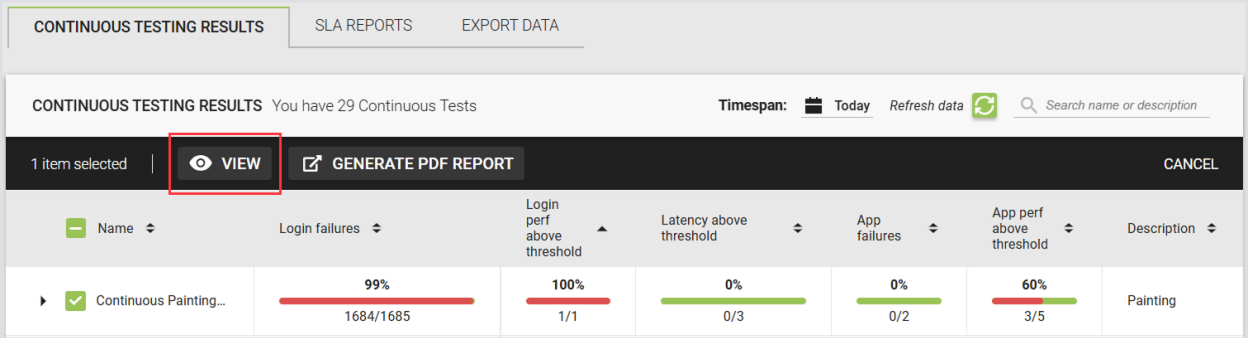
There are two additional tabs available for extended functionality:

- **SLA Reports**
- **Export Data**

See the SLA Reports and Export Data sections for details.

Single Test Results

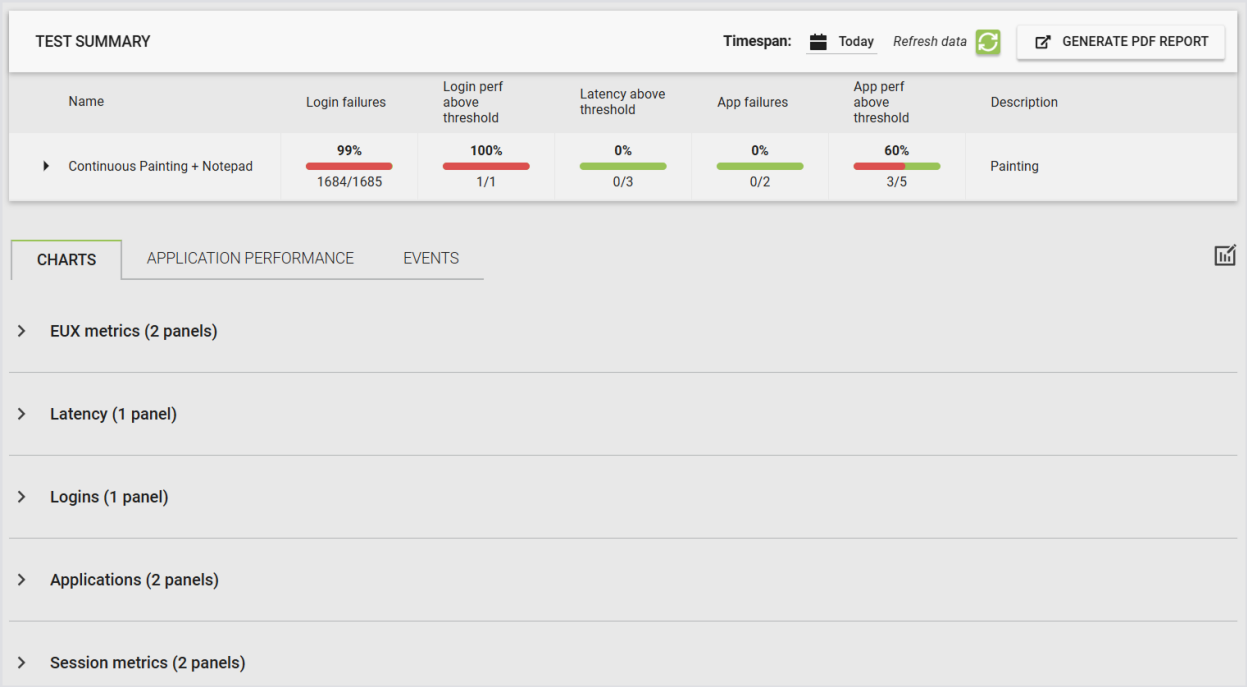
To view the test results for a single test, select a test and click **View**.



The following page displays the Test summary, Charts (EUX metrics, Latency, Logins, Applications, and Session metrics), Application Performance, and Events.

For the Charts section, you can select any time span using the date picker. Based on the selected time span, different aggregation options become available. For details, see Aggregation options.

NOTE: If the selected time span includes a future time range, a **Refresh** button appears on the Overview page and Result pages. Click **Refresh** to reload the page and update the results with the latest available data.



Application Performance

NOTE: This feature is available only in the Single Test view. It isn't yet available in Compare.

Use the Application performance tab to review performance details for applications in a Continuous Test result. The tab shows a list of applications included in the test. For each application, you can review:

- Above-threshold measurements compared with total measurements
- Failure rate
- Status bar showing failed and successful measurements
- Success rate

Select an application to open the application details page.

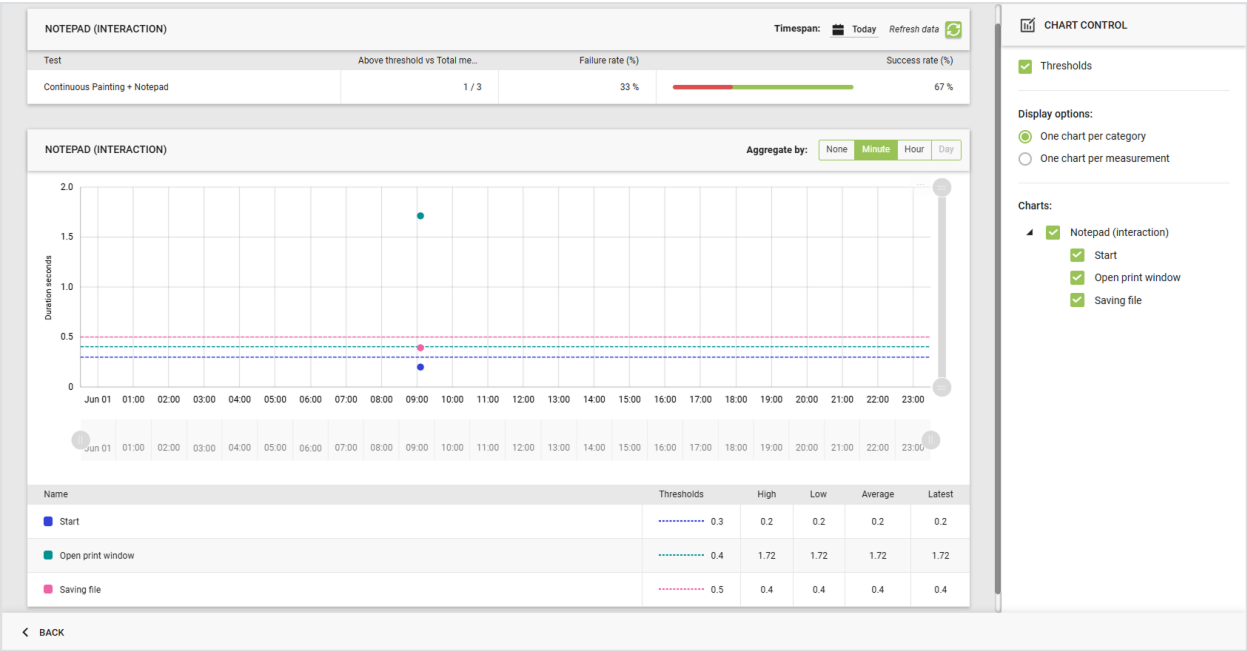
The application details page shows performance data for the selected application. The statistics table includes:

- Above-threshold measurements compared with total measurements
- Failure rate
- Status bar showing failed and successful measurements
- Success rate

Select the row in the statistics table to open the related events list.

NOTE: The event list shows only events of type **Application Threshold exceeded**.

The application details page also includes charts that show measurements for the selected application.

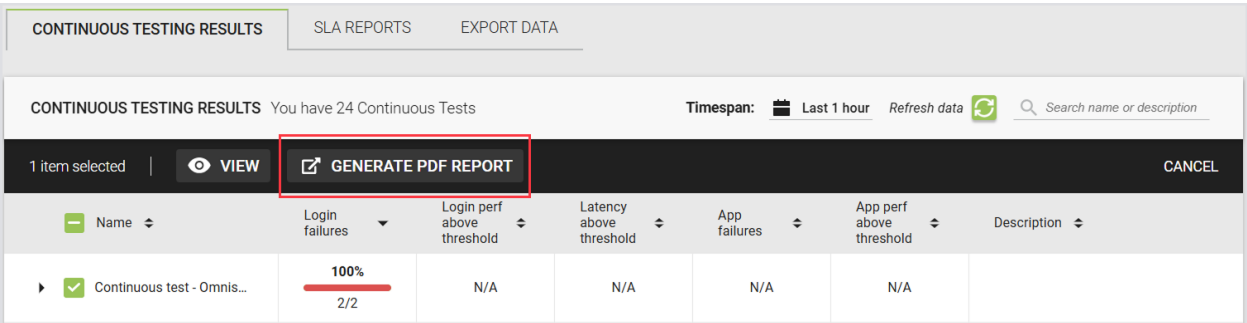


TIP: To change the time span, go back to the previous (Test Results Overview) page.

Generating a Test PDF Report

You can generate a single Continuous Test PDF report at any time. To generate the report:

1. In the **Continuous Testing results**, select the checkbox next to the test you're interested in.
2. Click **Generate report** in the top menu toolbar.
3. In the **Generate PDF report pop-up**, configure what information the report will include. For details, see Configuring a Test PDF report.
4. Click **Generate**, and then **Save**.



To learn about what the Test PDF report consists of, see the Test PDF Report Breakdown.

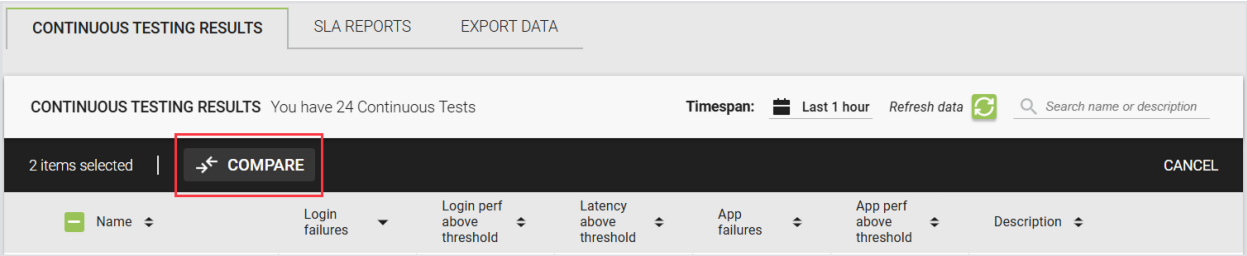
Compared Tests

Compared Test Summary

This page displays a list of compared Continuous Tests. The information you're viewing is the same as in the single Continuous Test summary. For details, see the Single Test Summary.

Compared Test Results

To view the test results for the compared tests, select up to 5 tests and click **Compare**.



The following page displays the Test summary, Charts (EUX metrics, Latency, Logins, Applications, and Session metrics), Application performance, and Events.

For the Charts section, you can select any time span using the date picker. Based on the selected time span, different aggregation options become available. For details, see Aggregation options.

Aggregation Options

Time Span	Available Aggregations	Default Aggregation	Disabled (Greyed Out)
< 2 hours	None (raw data), Per minute	None (raw data)	Per hour, Per day
2–24 hours	None (raw data), Per minute, Per hour	Per minute	Per day
24 hours - 1 month	Per hour, Per day	Per hour	None, Per minute
> 1 month	Per hour, Per day	Per day	None, Per minute

NOTE: Available aggregation options dynamically adjust based on the selected time span. Unavailable options are visible but disabled (greyed out).

EUX Metrics

This is where you can see the EUX score and EUX timers. Available EUX timers:

- Generic Applications score

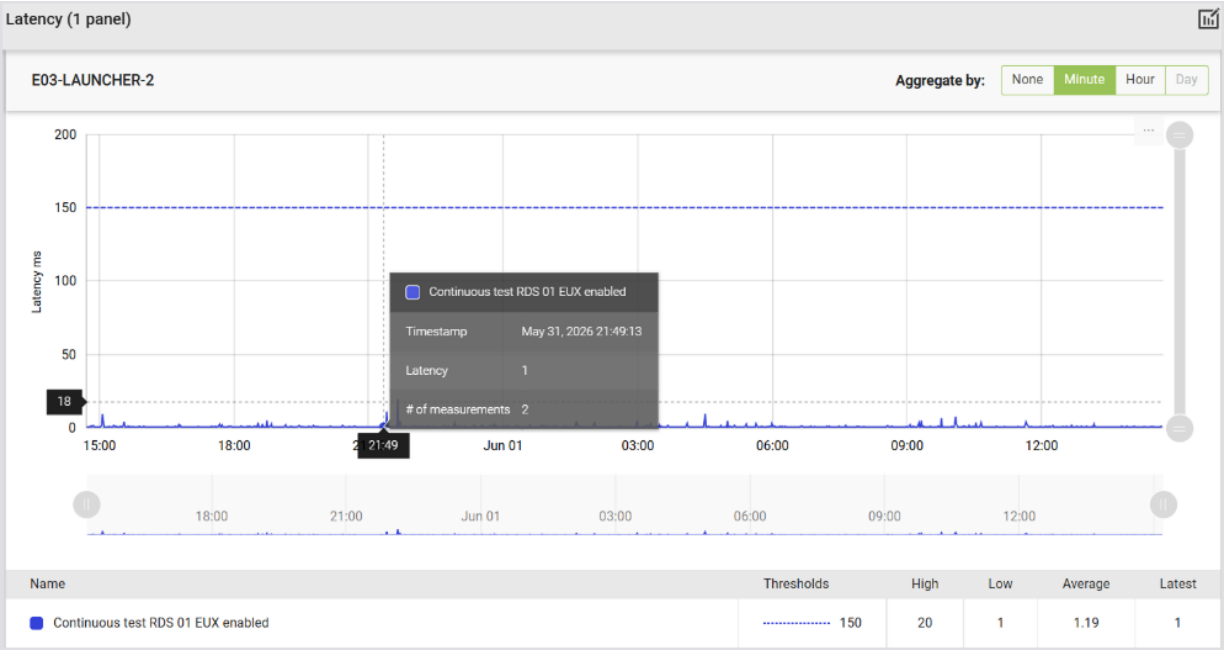
- CPU score
- My Documents I/O score
- Local AppData I/O score
- Mixed CPU I/O score
- User input score

Here's an example of an EUX score chart:

For more information on the EUX score, see [Calculating the EUX score and VSImax](#).

Latency

This is where you can see the latency metrics.



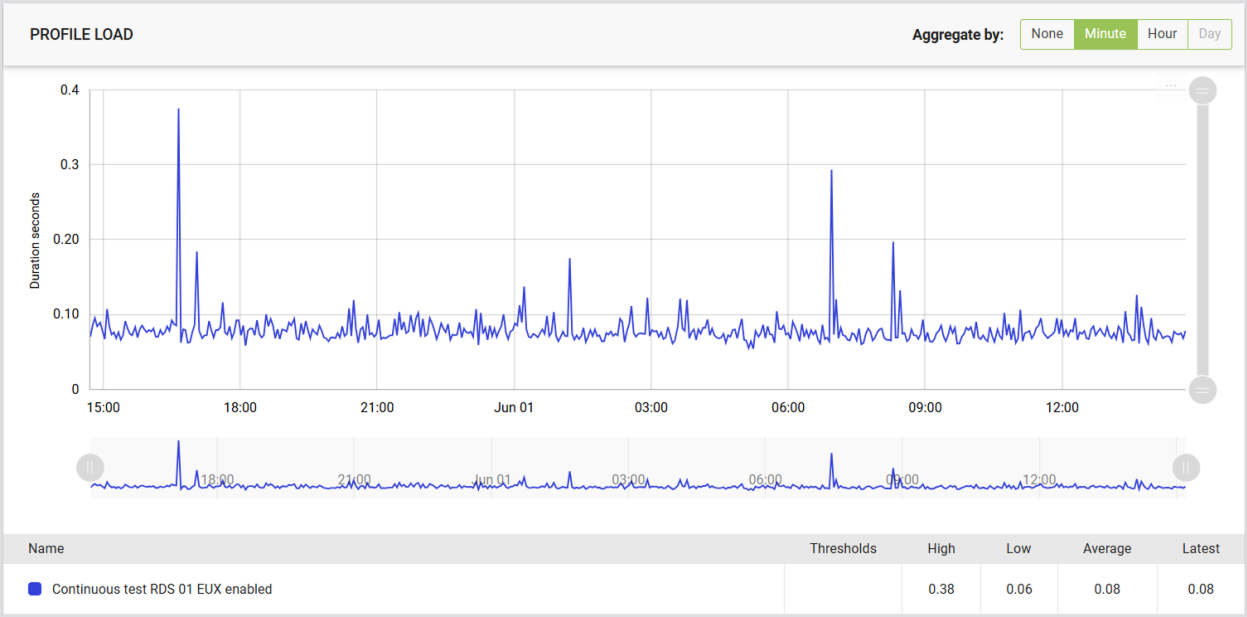
To learn more about latency, see [Monitoring Latency](#).

Logins

This is where you can see login performance metrics, such as:

- Display Protocol Connection
- Profile load
- GPO Processing
- Total login time

Here's an example of a Profile Load chart:

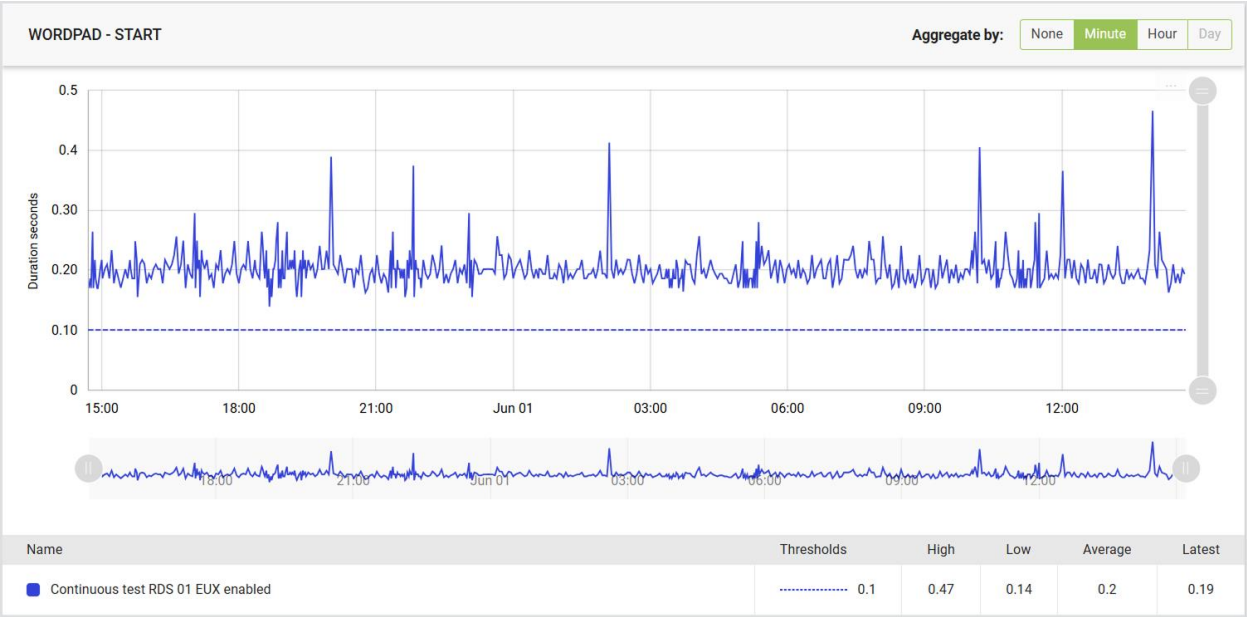


For more information on login performance, see [Monitoring Login Performance](#).

Applications

This is where you can analyze application performance metrics, such as start times and Virtual User workload functions.

Here's an example of a start time chart:

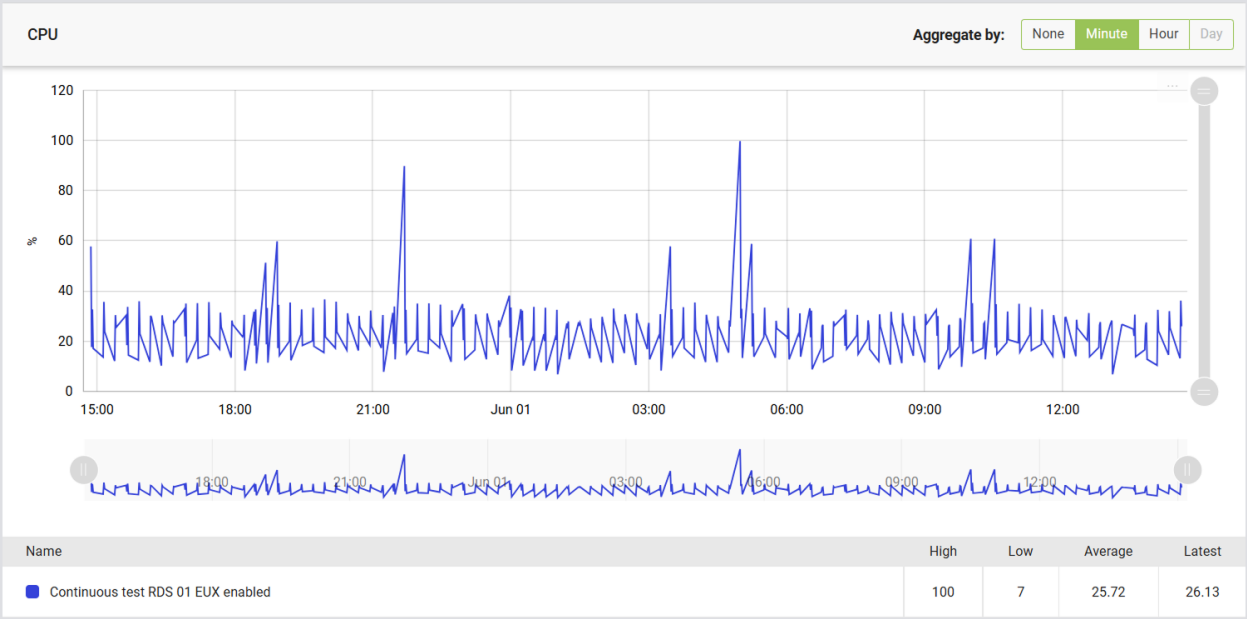


For details on Applications, see [Configuring Applications for Testing](#).

Session Metrics

This is where you can view session metrics, such as CPU and Memory.

Here's an example of a CPU chart:

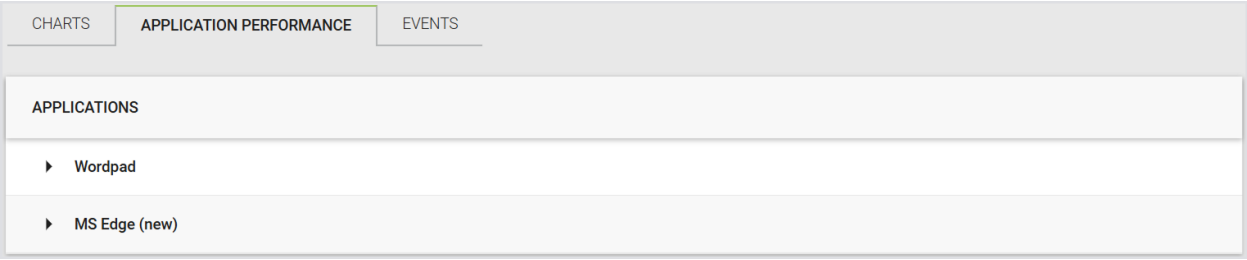


To learn more about Session Metrics, see Managing Session Metrics. For general guidelines on how to analyze Continuous Testing Session Metrics, refer to Viewing Session Metrics.

Application Performance

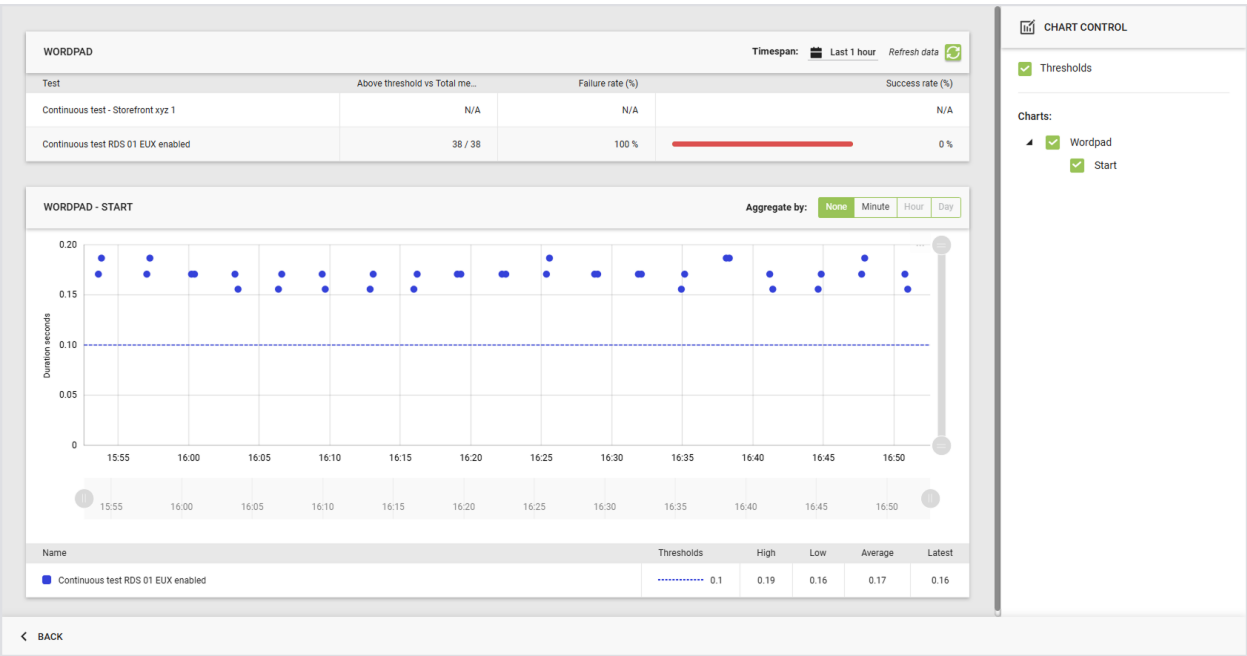
For general information, see the section of the same name in the Single Test Results section.

The tab displays application-level success and failure rates across the selected tests.



Select an application name in the table to open a page containing performance details and charts for that application across all selected tests.

NOTE: Application names in the table are links. Hover over an application name and select it to view its details.

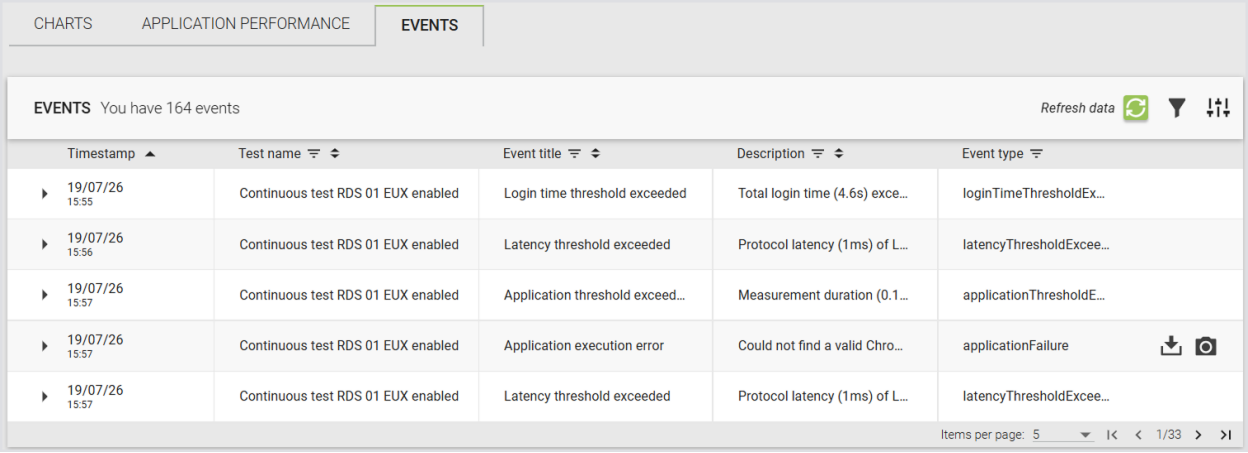


Events

The Events tab includes events related to Continuous Test and system events. To access Events:

1. In the **Login Enterprise sidebar menu**, navigate to **Results** > **Continuous testing**.
2. Select the test you're interested in, click **View**, and then select **Events** from the tab menu.

The Events table lists all recorded events:



3. Click an event to view more detailed information:

19/07/26 15:57	Continuous test RDS 01 EUX enabled	Application execution error	Could not find a valid Chro...	applicationFailure

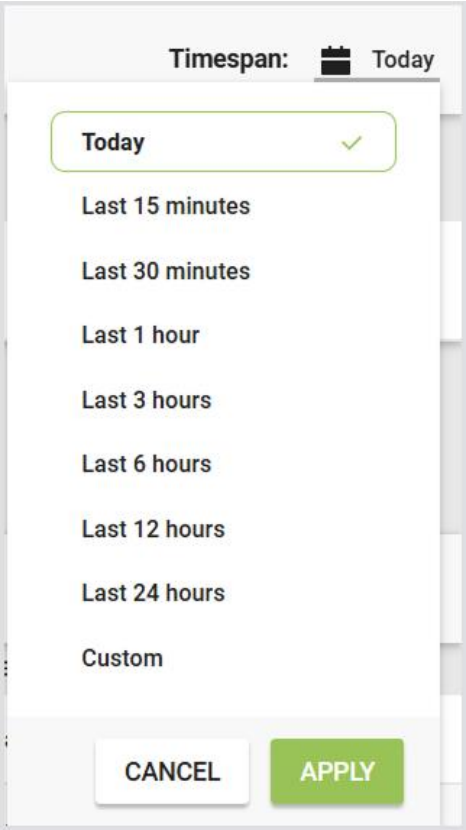
Details	Value
Account	e03-testuser018@qa
App execution	d0d6c124-30fb-40df-a515-531995ad2849
Application name	MS Edge (new)
Description	Could not find a valid Chrome installation
Event type	applicationFailure
Launcher	E03-LAUNCHER-2
Locale	English (United States)
Remoting protocol	Rdp
Resolution	1400 × 1050
Scale factor	100%
Target host	E03-RDSH-16
Target OS	Microsoft Windows Server 2016 Standard 10.0.14393 (1607)

TIP: If available, click on the Screenshot icon to the right of the event for more details about an application that failed. Click on the Download icon to download the script log.

Time Span Selection

You can select a custom time span to view events. The date picker includes quick-select options:

- Today
- Last 15 minutes
- Last 30 minutes
- Last 1 hour
- Last 3 hours
- Last 6 hours
- Last 12 hours
- Last 24 hours

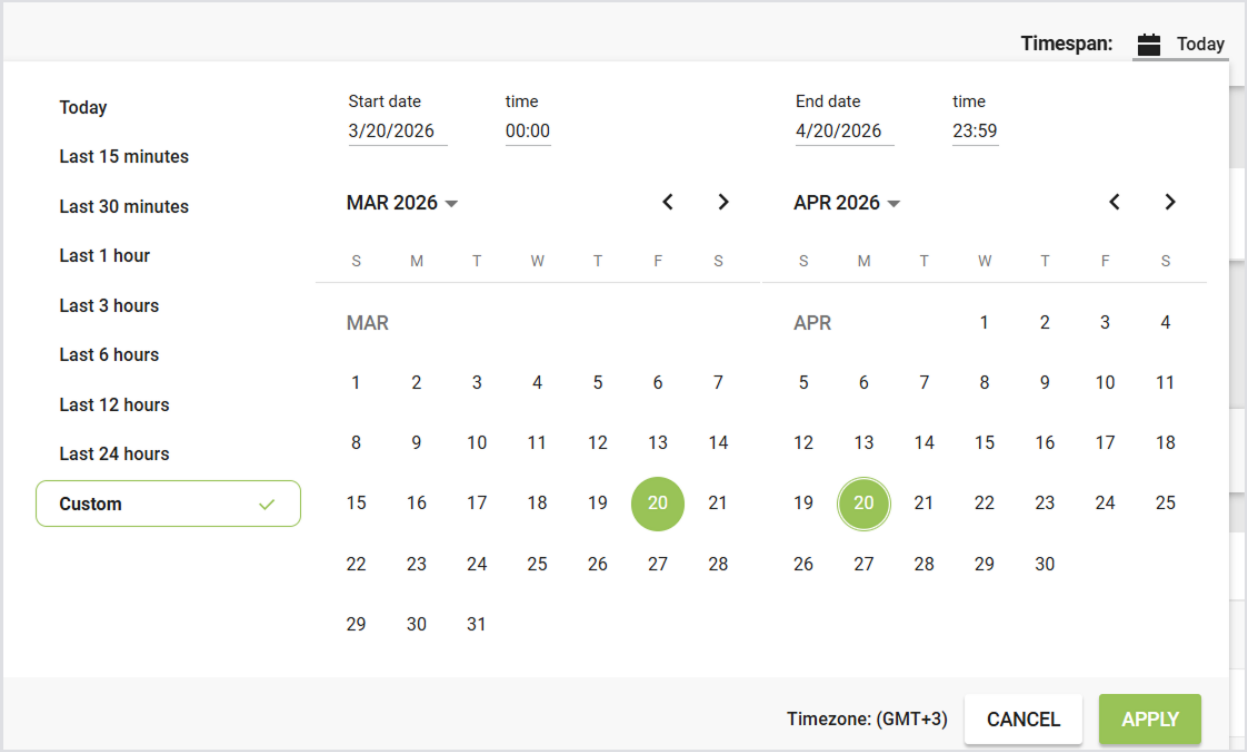


Custom Time Span

To select a custom time span:

1. Choose **Custom** in the date picker.
2. Use the **From** and **To** fields or the calendar view to select a start and end date.
3. Click **Apply**.

By default, the page shows events from the **last 1 hour**. When you change the time span, your selection is saved in the browser. Returning to the page will display your previously selected time span.



Live Updates

The page automatically updates with new events:

- If you select a quick option like **Today**, new events appear immediately in the list.
- If you select a time span such as **Last 15 minutes** or **Last 1 hour**, new events are added automatically, and the time span updates to include the new events.

TIP: A search field above the table allows you to filter events by Test name. This makes it easier to locate specific events quickly.

Possible Events

- `accountCapacityExceeded`
- `accountDisabled`
- `appExecutionAbandoned`
- `applicationFailure`
- `applicationThresholdExceeded`
- `connectionInitializationTimeout`
- `customUserSessionEvent`
- `dataRetentionFailed`
- `dataRetentionFinished`
- `dataRetentionStarted`
- `databaseFailure`
- `emailFailure`

- `emailRequest`
- `emailSent`
- `engineLogs`
- `enginePaused`
- `engineResumed`
- `euxExecutionFailure`
- `euxInitializationFailure`
- `latencyMeasurementFailed`
- `latencyThresholdExceeded`
- `launcherCapacityExceeded`
- `launcherLogs`
- `launcherOffline`
- `licenseSessionLimit`
- `loginFailure`
- `loginTimeMeasurementFailed`
- `loginTimeThresholdExceeded`
- `remoteSessionDisconnected`
- `screenshotFailure`
- `scriptEvent`
- `scriptScreenshot`
- `sessionDiscoveryError`
- `sessionFailure`
- `sessionRequestEndedBeforeEngineBecameOnline`
- `testRunCancelled`
- `testRunFailed`
- `testRunFinished`
- `testRunStarted`

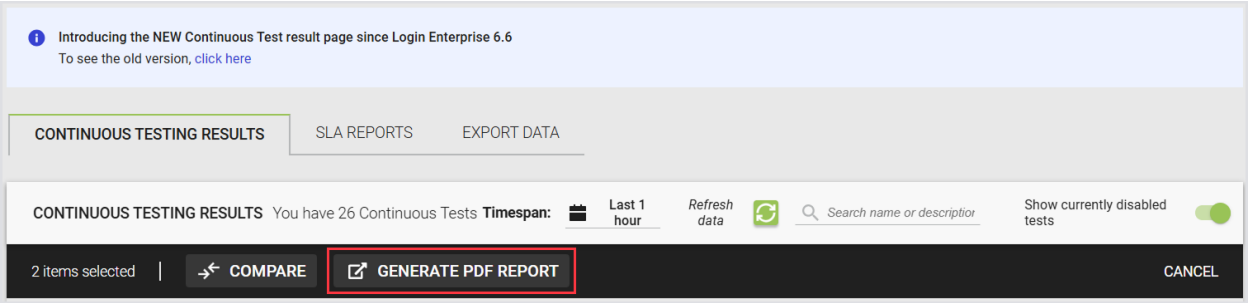
Generating a Test PDF Report

A report for multiple Continuous tests contains the same types of results as a report for a single test, but presents them differently. Tables show results for each test, and each measurement chart includes a separate line for every test.

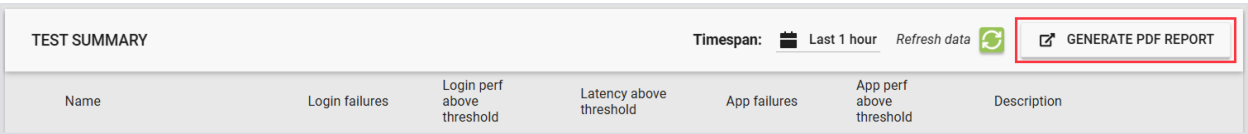
NOTE: The report presents results for multiple tests side by side but doesn't highlight differences between them.

You can generate and download a PDF report with the results of up to 5 Continuous Tests in one of the following ways:

1. Using the toolbar on the **Continuous Test Results** page:



2. Using the **Generate PDF Report** button on the **Compare** page:



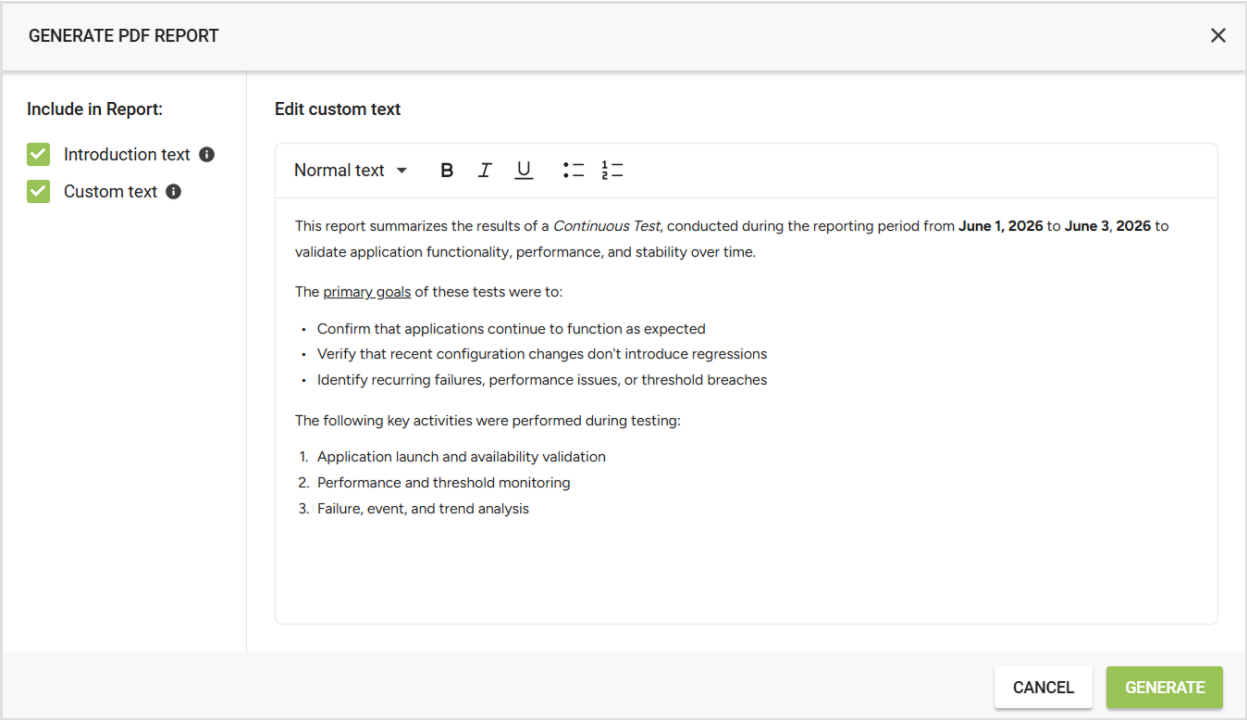
3. In the **Generate PDF report** pop-up, configure what information the report will include. For details, see Configuring a Test PDF Report. To learn about what the report consists of, see the Test PDF Report Breakdown.

Configuring a Test PDF Report

You can choose what you would like in your Continuous Test PDF report. Once you click **Generate PDF report**, you can configure the following:

Setting	Description	Options / Details	Notes
Introduction text	Include general information about Login Enterprise at the beginning of the report.	• *Include introduction text*: Adds the standard intro section. • *Exclude introduction text*: Omits it from the report.	-

Custom text	Add your own text to the PDF report. Useful for describing what was tested, why, or summarizing conclusions.	• *Include custom text*: Opens a pop-up to enter your content.
		• *Exclude custom text*: Skips this section.
			• The text editor has the following options:
			  ○ Font size (heading or normal text)
			  ○ **Bold**
			  ○ *Italic*
			  ○ Underlined
			  ○ Bullet points
			  ○ Numeric lists
			• Any info entered in the custom text section will be added to the beginning of the report, directly under the introduction text.
			• Use the Enter key to create paragraph breaks. If an uninterrupted block of text exceeds one page, the Continuous Test PDF will generate a blank page. The custom text section supports up to five pages.



Test PDF Report Breakdown

The Test PDF report summarizes the findings obtained from the testing process. It provides an overview of the Test specifics, results, and measurements. The report is divided into the following sections:

- **Test details section**: Shows the current configuration of the test.
- **Environment summary**
  - A table that shows the failure and success rates for logins, login performance, and latency performance
  - A chart that shows the success rates of all 3 measurements over time
    - Logins
    - Login performance
    - Latency performance
- **Application summary**
  - A table that shows the failure and success rates for application execution and performance
  - A chart that shows the success rates of the 2 measurements over time
    - Application execution
    - Application performance
- **EUX score**
  - 2 charts with the results of the EUX measurements of the selected time span.
    - EUX score
    - EUX timers
- **Logon time**
  - A table that shows the failure and success rates for the logon time. It contains one row for the total logon time.
  - A chart that contains the login-related measurements over time.
- **Latency performance**
  - A table that shows the failure and success rates for the latency (of all launchers).
  - A graph that shows the latency results over time for each launcher.
- **Application performance**
  - A table with all applications in the test. For each application, the table shows failure and success rates at the application level.
- **Application details**: Detailed results for each application
  - A table that shows failure and success rates for each measurement within the application.
  - Chart(s) that show the results of each measurement over time.
- **Session metrics**
  - Each session metric has its own chart.

SLA Reports

SLA reporting provides a concise and straightforward document that delivers key performance information. These reports are generated from historical data gathered by Login Enterprise during its operational processes, making them ideal for monitoring service-level agreements,

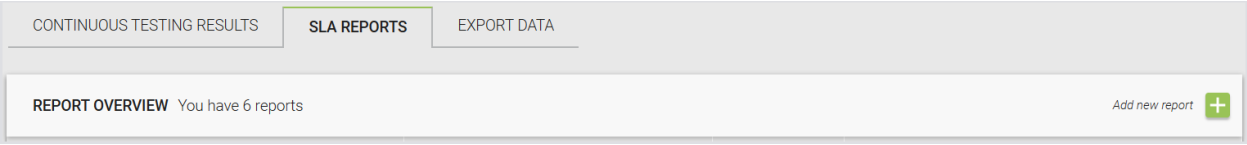
understanding infrastructure performance over time, and facilitating operations reporting to management and service customers.

Benefits of SLA reporting

Benefit	Explanation
Automated history of Test Results	SLA reports provide a clear and automatically generated history of test scenario results, making it easy to review and analyze past performance.
Multiple email recipients	Allows the distribution of results to multiple stakeholders. For example, an IT manager can quickly review the overall health and ROI indicators through a PDF attachment, while Login Enterprise administrators can focus on specific details relevant to their responsibilities within the same email.
Simplified review process	SLA reports are designed to be easy to read, streamlining the process of reviewing results within a defined time frame. This simplifies the task compared to examining Continuous test results and charts directly in the UI.

Configuring SLA Reports

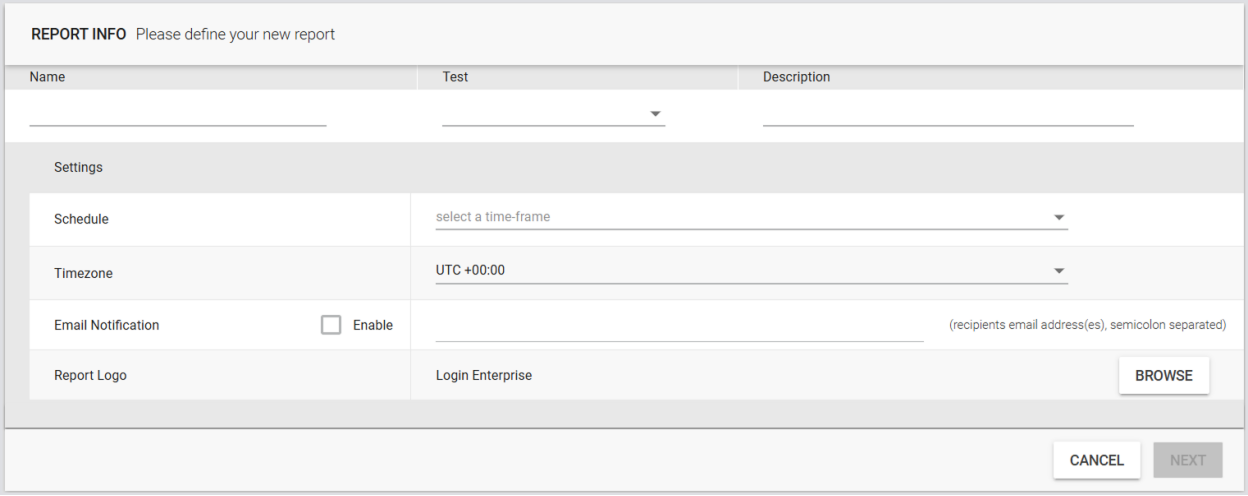
1. To create a new report, in the **Login Enterprise sidebar menu**, navigate to **Results** > **Continuous Testing**.
2. In the **Continuous Testing results**, select **SLA Reports** from the tab menu.
3. In the **SLA reports**, click the green "+" in the upper-right corner to add a new SLA report:



4. Configure the SLA report with the following settings:

Report info		
Name	**Test**	**Description**
The name of the SLA report.	The test for which the SLA report will be generated.	A description of the SLA report.
Settings		
Schedule	The time frame for generating the report, e.g., Daily, Weekly, or Monthly.	
Time zone	The time zone in UTC format.	

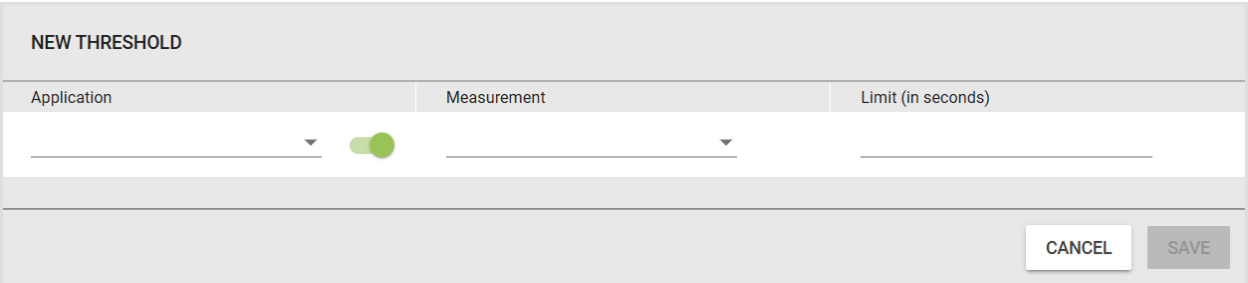
Email notification	Enable email notifications and enter the email addresses that should receive automatically generated reports. Separate multiple email addresses with a semicolon (;). **NOTE**: The Email notification function requires that the email settings be configured. For details, see Managing Notifications. If email notifications are enabled and no email addresses are specified, the global email settings will be used.
Report logo	Add your company logo to customize the report. Supported formats: JPEG, JPG, PNG, SVG (max. 1 MB).



5. Click **Next**.
6. Click **Add new threshold**. This allows you to configure the following:

Application	Measurement	Limit (in seconds)
Select the application to monitor.	Choose the specific application measurement.	Set the time limit for the measurement.

NOTE: Any measurements that exceed this limit will be included in the SLA report.

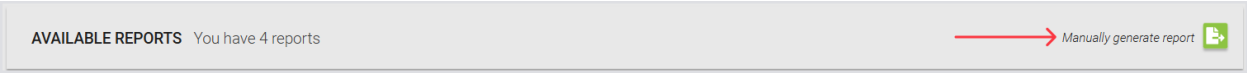


7. Click **Save** to apply SLA report configurations.

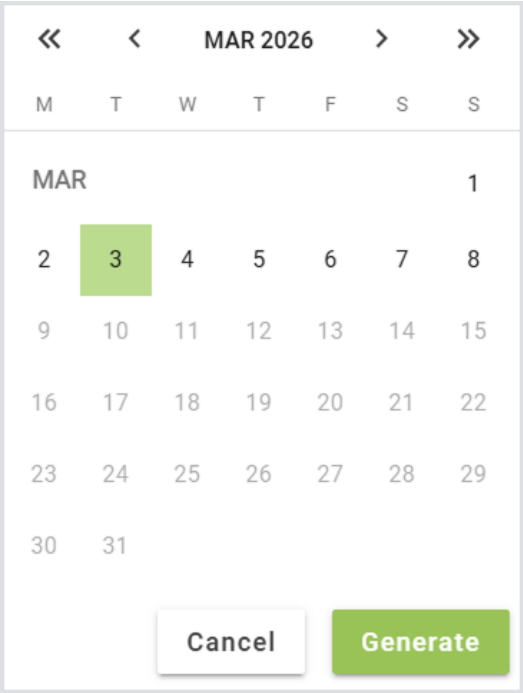
Generating SLA Reports

Reports are not generated immediately unless you click **Manually generate report**. If the schedule is set to Daily, the first report will be available after one day. To generate a report manually, go to **Available reports**:

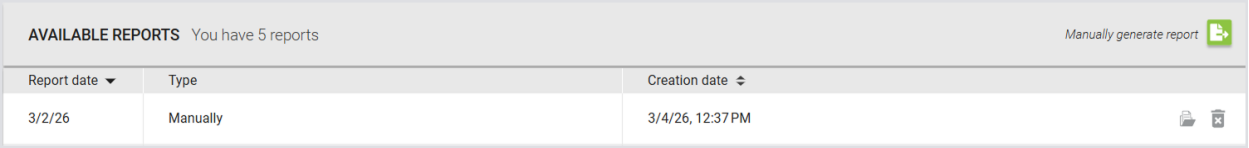
1. Click **Manually generate report**.



2. Select the desired date.
3. Click **Generate**.



To view the report, go to **Available reports** and click the report you just generated. The PDF file will download automatically:



You can also download a sample SLA report from here.

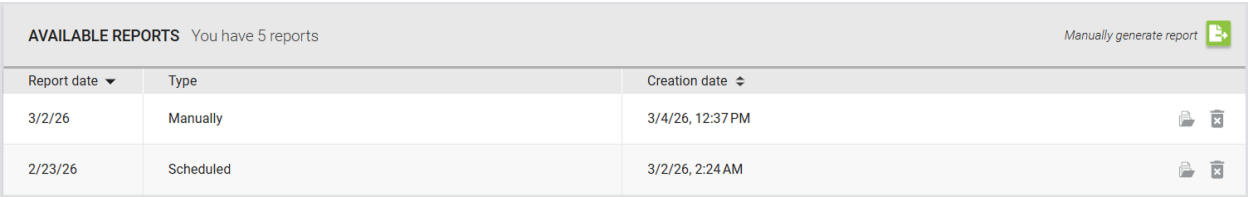
Scheduled Reports

- **Daily reports**: Generated every day.
- **Weekly reports**: Generated every week.
- **Monthly reports**: Generated every month.

Scheduled reports can be deleted as needed.

The SLA report displays the report date, which indicates the time frame it covers.

NOTE: The Application Success value in the SLA Report represents the status of the entire workload script to complete successfully.



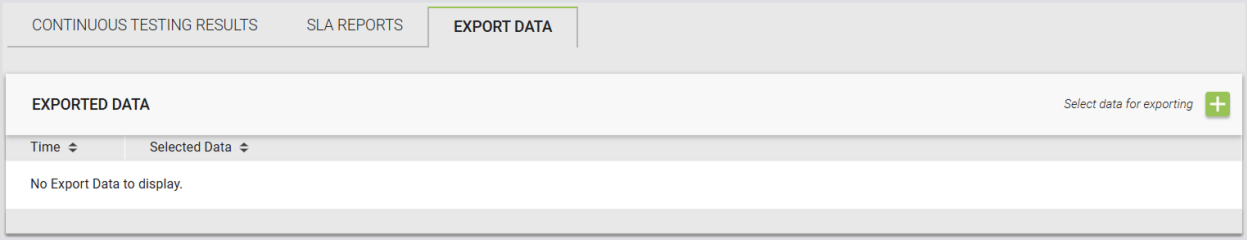
Export Data

You can export data from Continuous Tests in CSV format. This is useful for integrating data from different sources.

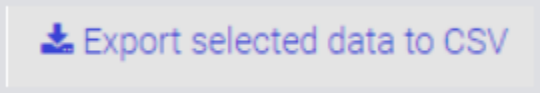
Benefit	Explanation
Comprehensive data analysis	Exporting raw data allows you to perform detailed analysis and compare metrics from different tests or sources.
Data integration	Easily combine and correlate data from various tests or external systems for more robust reporting.
Error tracking	Access detailed information about application execution and performance to help troubleshoot issues.

You can export the data using:

1. **The Export Data Tab**:
    1. In the **Continuous Testing Results** page, select **Export Data** from the tab menu.
    2. Click the green + icon to create a new export.
    3. In **From date** and **To date**, specify the timeframe for exporting data, and click **Next**.
    4. In **Test selection**, select environments for exporting data, and click **Export**.
    5. In **Export data**, click the folder icon to download the .zip file.



2. **The Charts Tab**:
    1. In the **Continuous Test Results** page, select **Charts** from the tab menu.
    2. Use the drop-down menu on the left to select the configuration.
    3. Click **Export selected data to CSV** under the chart window.
    4. The RawData.zip file will be prepared and downloaded.



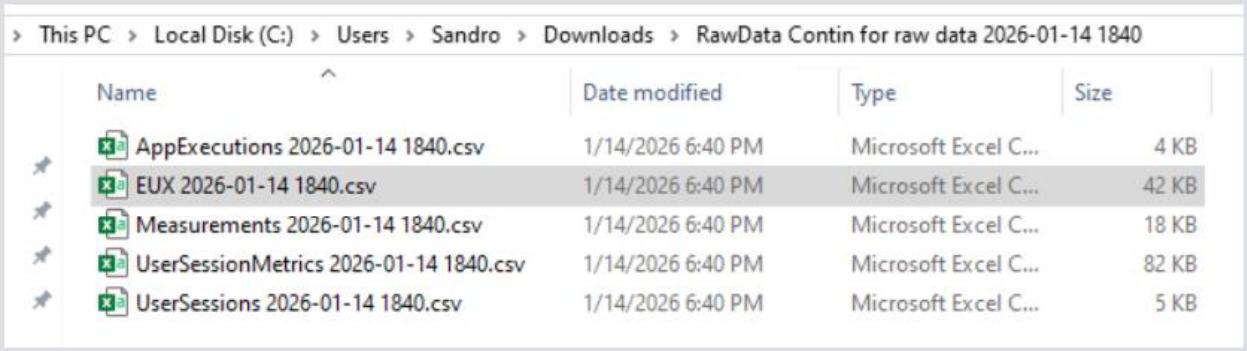
NOTE: The data you downloaded will also be displayed in the **Export Data** tab for your reference.

RawData.zip File

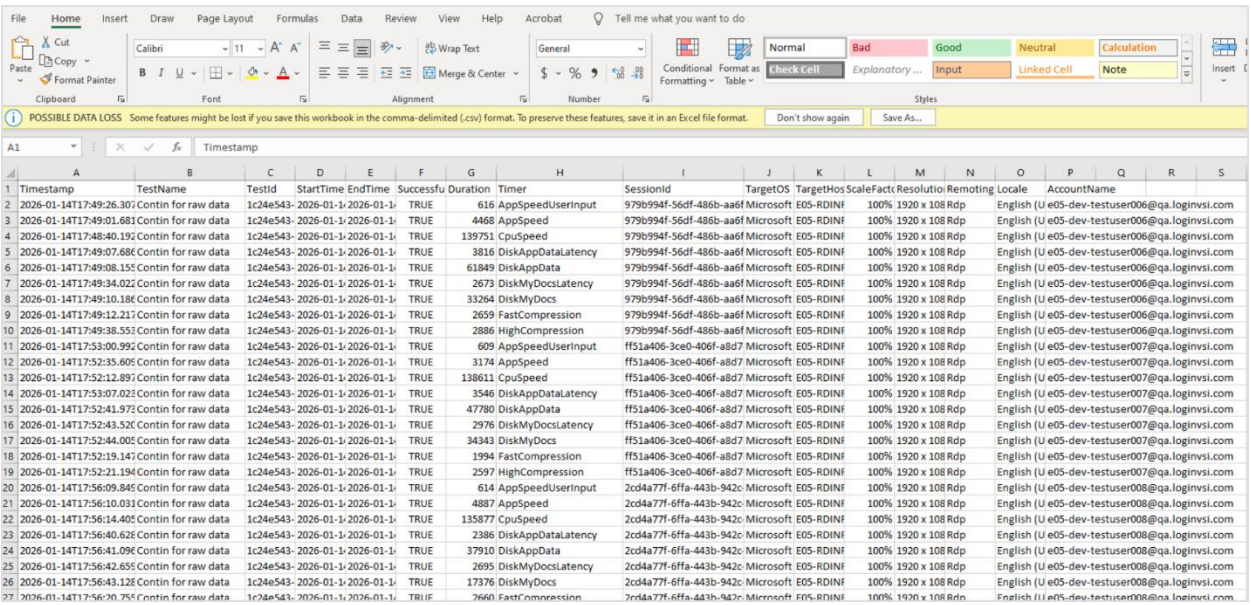
The RawData.zip file is named according to the following schema: RawData <TEST_NAME> <DATE> <TIME>.zip

IMPORTANT: If the raw data export was generated before v6.5, and you download it, it will not contain the EUX data.

File Name	Description
AppExecutions	Data on application script executions. Each entry represents a Virtual User starting an app and includes details on whether any errors occurred during the execution.
EUX	EUX data that you can import or export into your own data tools for reporting and analysis.
Measurements	All measurements captured during the Test Run. It covers application response times and other timers configured in the test settings.
UserSessionMetrics	CPU and Memory utilization metrics for each user session in the Test Run, expressed as percentages.
UserSessions	Information about the Virtual User accounts used during the Test Run. It details the Launcher of origin, the target host, and the logon results.



An example of the RawData.zip file looks as follows:



Timestamp	TestName	TestId	StartTime	EndTime	Successfu	Duration	Timer	SessionId	TargetOS	TargetHos	ScaleFact	Resolutio	Remoting	Locale	AccountName
2026-01-14T17:49:26.307	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	616	AppSpeedUserInput	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:01.681	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	4468	AppSpeed	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:48:40.192	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	139751	CpuSpeed	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:07.686	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	3816	DiskAppDataLatency	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:08.155	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	61849	DiskAppData	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:34.022	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2673	DiskMyDocsLatency	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:10.186	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	33264	DiskMyDocs	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:12.217	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2659	FastCompression	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:49:38.553	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2886	HighCompression	979b994f-56df-486b-aa6f	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser006@qa.loginvsi.com
2026-01-14T17:53:00.992	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	609	AppSpeedUserInput	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:35.605	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	3174	AppSpeed	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:12.897	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	138611	CpuSpeed	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:53:07.023	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	3546	DiskAppDataLatency	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:41.973	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	47780	DiskAppData	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:43.520	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2976	DiskMyDocsLatency	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:44.005	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	34343	DiskMyDocs	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:19.147	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	1994	FastCompression	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:52:21.194	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2597	HighCompression	ff51a406-3ce0-406f-a8d7	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser007@qa.loginvsi.com
2026-01-14T17:56:09.845	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	614	AppSpeedUserInput	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:10.031	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	4887	AppSpeed	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:14.405	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	135877	CpuSpeed	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:40.628	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2386	DiskAppDataLatency	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:41.096	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	37910	DiskAppData	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:42.655	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2695	DiskMyDocsLatency	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:43.128	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	17376	DiskMyDocs	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com
2026-01-14T17:56:20.755	Contin for raw data	1c24e543-	2026-01-1	2026-01-1	TRUE	2660	FastCompression	2cd4a77f-6ffa-443b-942c	Microsoft	E05-RDINF	100%	1920 x 108	Rdp	English (U	e05-dev-testuser008@qa.loginvsi.com

EUX Raw Data

To enhance debuggability and facilitate root-cause analysis, EUX raw timer data can be essential. Login Enterprise stores the raw data by default, and the data retention can be adjusted based on your specific needs.

Retrieving the EUX Raw Data

You can retrieve the EUX raw data via the Public API with the following request:

```
GET https://appliance.loginvsi.com/publicApi/v7/test-runs/{testRunId}/eux-script-executions
```

TIP: The `testRunId` is a Continuous Test Id.

The API will return all the EUX measurements for each user session. Below is an example of how the response is structured:

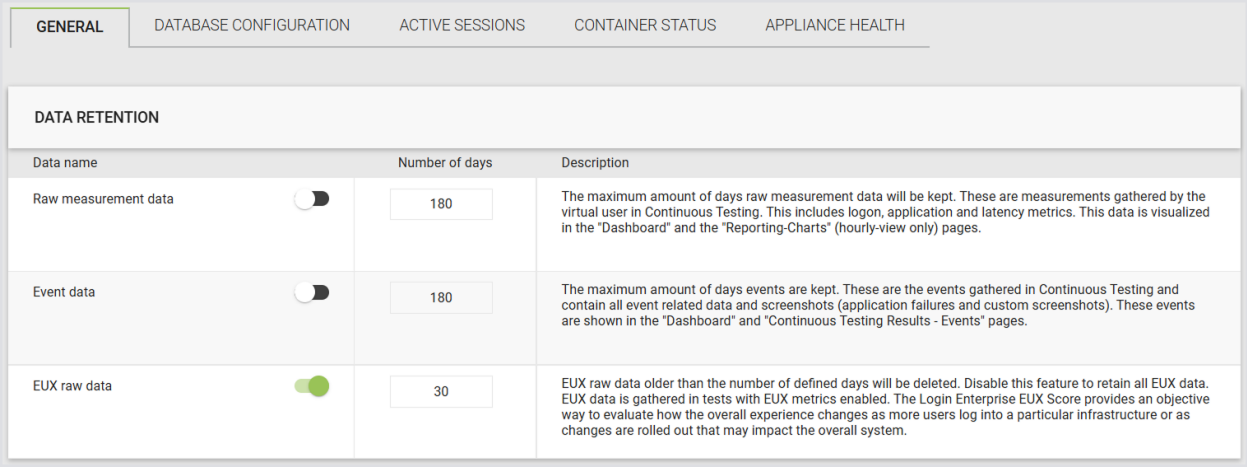

```
{
"items": [
{
"userSessionId": "6f838f90-b9ed-4e63-b04e-2291a50e34cb",
"startTime": "2025-01-06T15:20:55.288615Z",
"endTime": "2025-01-06T15:22:08.481351Z",
"successful": true,
"euxMeasurements": [
{
"timestamp": "2025-01-06T15:21:55.998094Z",
"timer": "appSpeedUserInput",
"duration": 635
},
{
"timestamp": "2025-01-06T15:21:32.025Z",
"timer": "appSpeed",
"duration": 5656
},
{
"timestamp": "2025-01-06T15:22:00.373194Z",
"timer": "cpuSpeed",
"duration": 210623
},
{
"timestamp": "2025-01-06T15:21:37.61784Z",
"timer": "diskAppDataLatency",
"duration": 4329
},
{
"timestamp": "2025-01-06T15:22:02.388249Z",
"timer": "diskAppData",
"duration": 16952
},
{
"timestamp": "2025-01-06T15:22:03.918992Z",
"timer": "diskMyDocsLatency",
"duration": 4424
},
{
"timestamp": "2025-01-06T15:22:04.419015Z",
"timer": "diskMyDocs",
"duration": 14851
},
{
"timestamp": "2025-01-06T15:21:42.172781Z",
"timer": "fastCompression",
"duration": 920
},
{
"timestamp": "2025-01-06T15:21:19.825964Z",
"timer": "highCompression",
"duration": 925
}
```

```
],
```

Cleaning the EUX Data (Data Retention)

You can configure the data retention period for EUX raw data, ranging from 1 to 9999 days. Customizing this setting is crucial to managing storage resources effectively. New installations and upgrades default to a 30-day retention period, but any customizations persist through subsequent upgrades. To adjust the EUX raw data:

1. In the **Login Enterprise sidebar menu**, navigate to **Other** > **System**.
2. In the **System**, click **General**.
3. In **Data retention**, configure **EUX raw data**.



Tips and Tricks

Chart Legend

The chart legend provides additional statistical insights for the selected time span:

- **High**: Highest value within the selected time span
- **Low**: Lowest value within the selected time span
- **Average**: Average value within the selected time span
- **Latest**: Most recent value within the selected time span

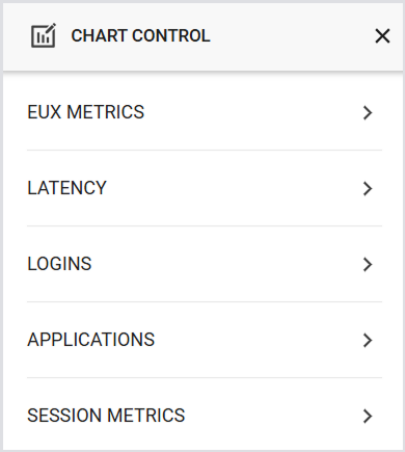
NOTE: Displayed values depend on the selected aggregation. For example, the *Low* value may differ when using **Per minute** vs **Per hour** aggregation.

Name	High	Low	Average	Latest
Cont Test RC (Notepad + Paint)	8.2	4.7	7.83	7.8

Chart Control

Use the Chart Control to navigate between the chart sections and select or deselect measurements.

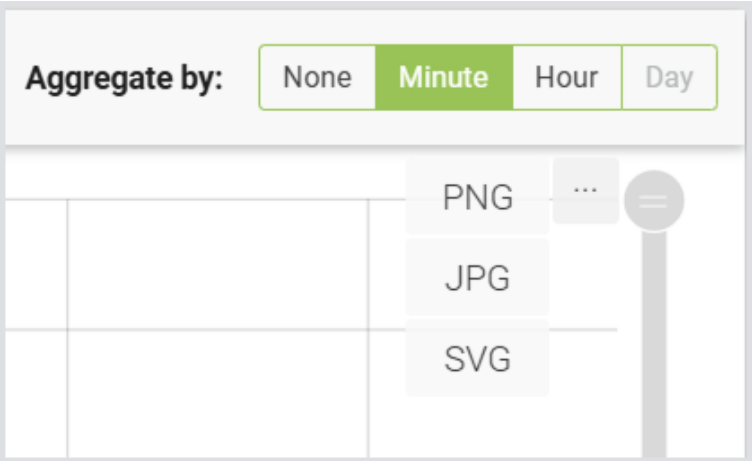
NOTE: Chart Control is the same as in Load Testing Results, except the Continuous Testing Results page doesn't support side-by-side chart viewing. For details, see Chart Control.



Downloading Charts as an Image

You can download charts as images for sharing or use in custom reports and presentations. Downloading a chart as an image applies to all charts: Load Test, Continuous Test, and Appliance Health. To download a chart:

1. Locate the Download icon in the top right corner of the chart.
2. Select image format: PNG, SVG, or JPG.
3. Download the image. It will be saved to your Downloads folder, or the directory you've configured as your default Downloads location.



Viewing Load Testing Results

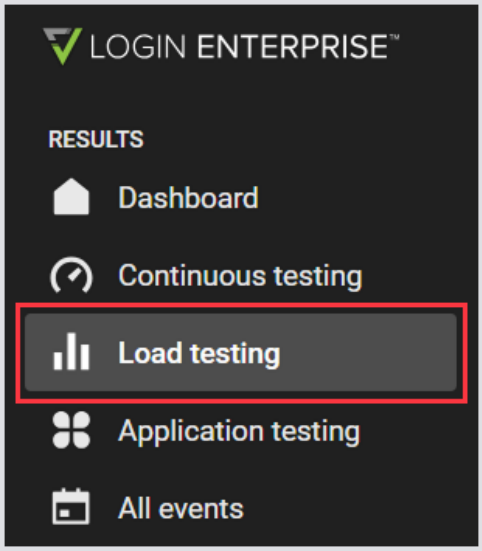
- Introduction
- Single Test
  - Single Test Summary
  - Single Test Results
  - Generating a Test PDF Report
- Two Load Tests
  - Quick Results View
- Compared Tests
  - Compared Test Summary
  - Compared Test Results
    - EUX Score and VSImax
    - Logon Time
    - Application Response Time
  - Charts
  - Events
  - Generating a Test PDF Report
- Configuring a Test PDF Report
- Test PDF Report via the Public API
- Test PDF Report Breakdown
- Export Data
- EUX Raw Data
- Tips and Tricks
  - Customizing Table Columns
  - Chart Legends with a Table
  - Chart Control
  - Downloading Charts as an Image

Introduction

You can view the results of individual Load Tests and compare up to five Load Test instances. The Load Test Results page provides a comprehensive overview of measurements, including charts, graphs, and detailed events for each Test. You can also download both single and multiple Test results as PDF reports for easy sharing and analysis. See the instructions below for more details.

Single Test

When a Load Test is completed, you can start looking at the results. To do this, in the **Login Enterprise sidebar menu** > **Results**, navigate to **Load testing**.

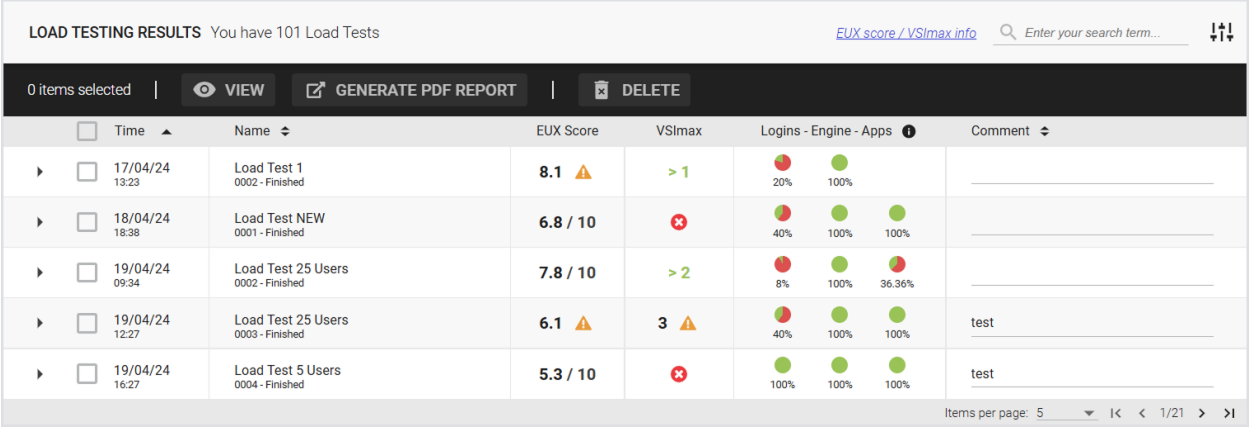


Single Test Summary

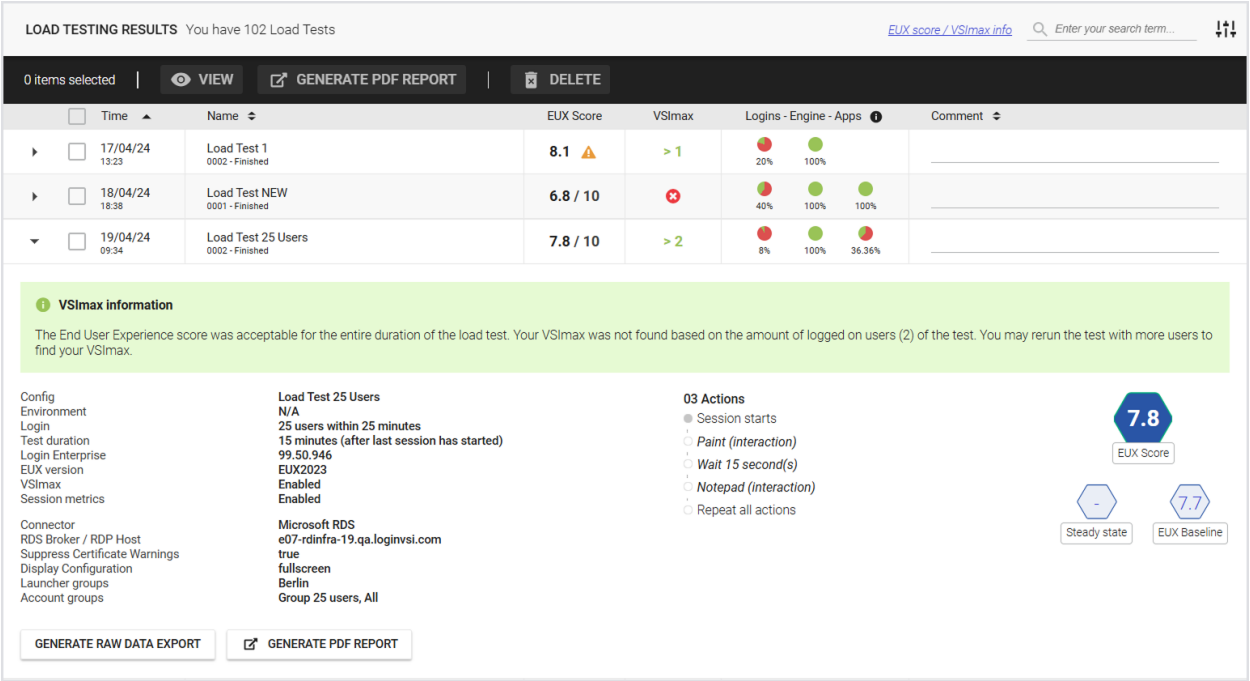
This page displays a list of previously executed Load Tests and includes the following:

Time	Name	EUX score	VSImax	Logins	Engine	Apps	Comment
Date and time of the Test.	Name and sequence number of the Test.	End User Experience score, i.e., a specific set of tools that produce measurements meant to closely approximate how a human would feel about using this machine. A score from 0 to 10. For details, see the EUX score.	The maximum number of concurrent active VDI sessions your hardware will support without compromising responsiveness. For details, see the VSImax.	Percentage of sessions that succeeded in logging in on the target environment.	Percentage of sessions that succeeded in logging in on the target environment.	Percentage of successfully executed Application scripts.	The configured comment for the Test.

NOTE: VSImax results are available only if the EUX score was enabled during the Load Test creation. If the EUX score is turned off, VSImax is automatically disabled and excluded from Test results (this includes single test results, compared test results, and PDF test reports).



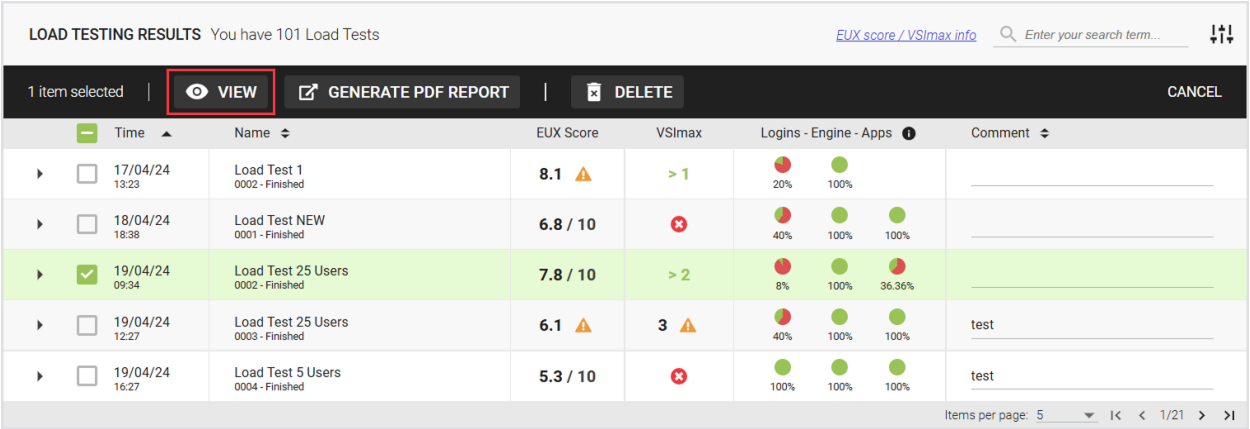
To view the Test details, click on the Test you're interested in:



For information on the EUX score and VSImax, see [Calculating the EUX score and VSImax](#).

Single Test Results

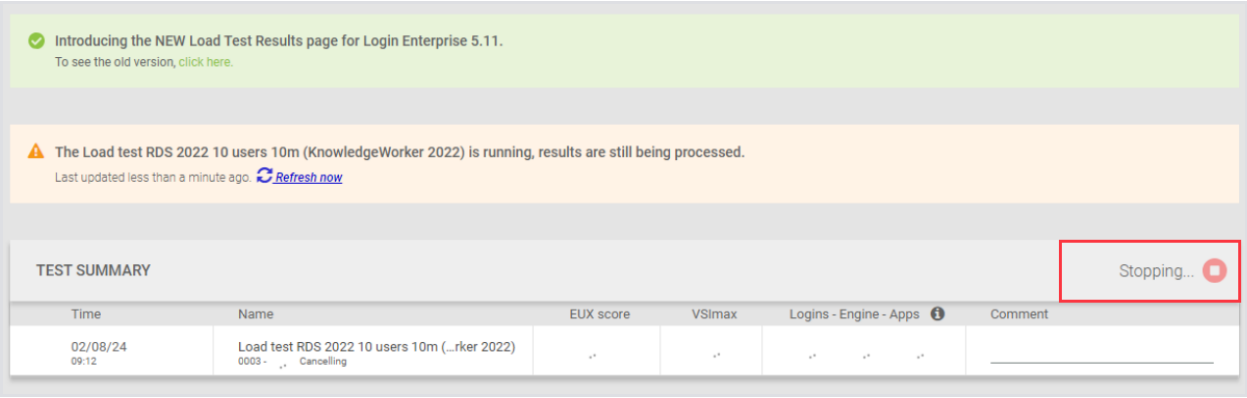
To view the test results for a single test, select a test and click **View**.



The following page displays the Test summary, Overview (EUX Score/VSImax, Logon Time, and Application Response Time), Charts, Events, and Access. For details, see the Test Results.

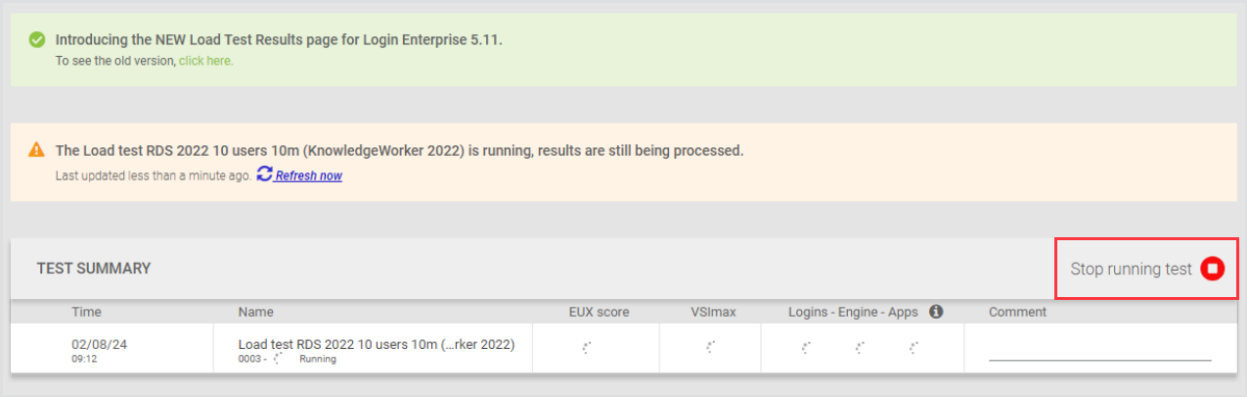
Auto Refresh

The page will automatically refresh every minute while a Load Test is running. This feature is available only in the single Load Test view and not in the Compared Tests views.



Stopping the Test

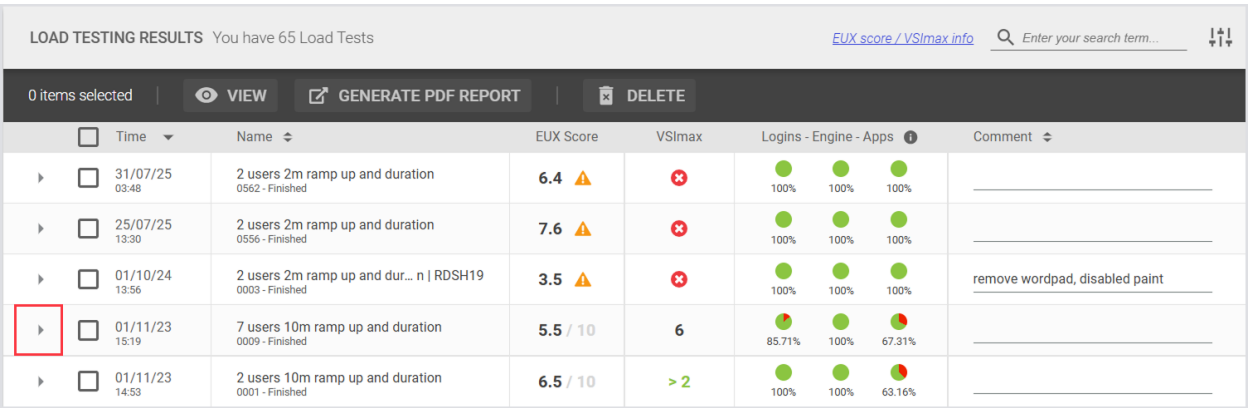
You can stop a Load Test directly from the Results page, provided the Test is still running. This option is available only for single Load Tests. To stop the Load Test, click **Stop Running Test**. The screenshot below highlights the button to help you locate it easily.



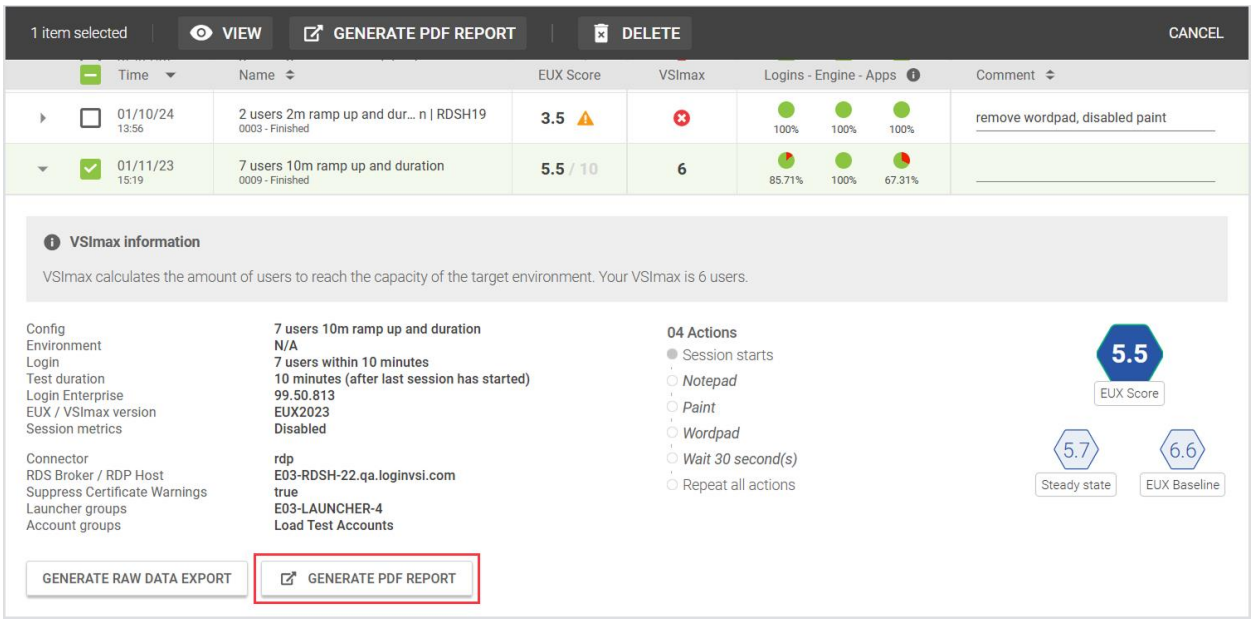
Generating a Test PDF Report

Generate a single Test PDF report manually or using the Public API. To generate the report:

1. In the **Load Testing results**, click on the Test you're interested in.



2. Click **Generate PDF report** (A browser pop-up will open).



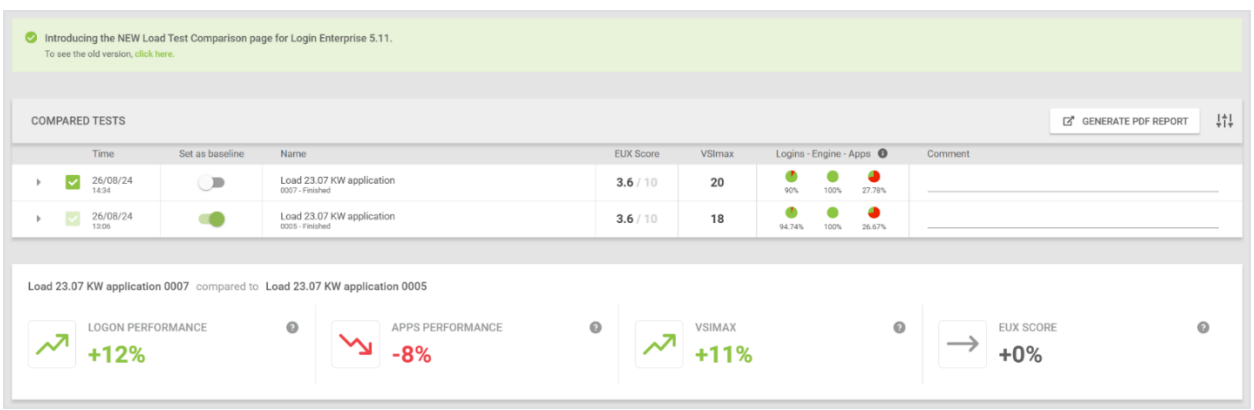
3. In the **Generate PDF report** pop-up, configure what information the report will include. For details, see Configuring a Test PDF Report.
4. Click **Generate**, and then **Save**.

TIP: Alternatively, select the checkbox next to the specific Test, and in the top menu toolbar, click **Generate PDF report**.

Two Load Tests

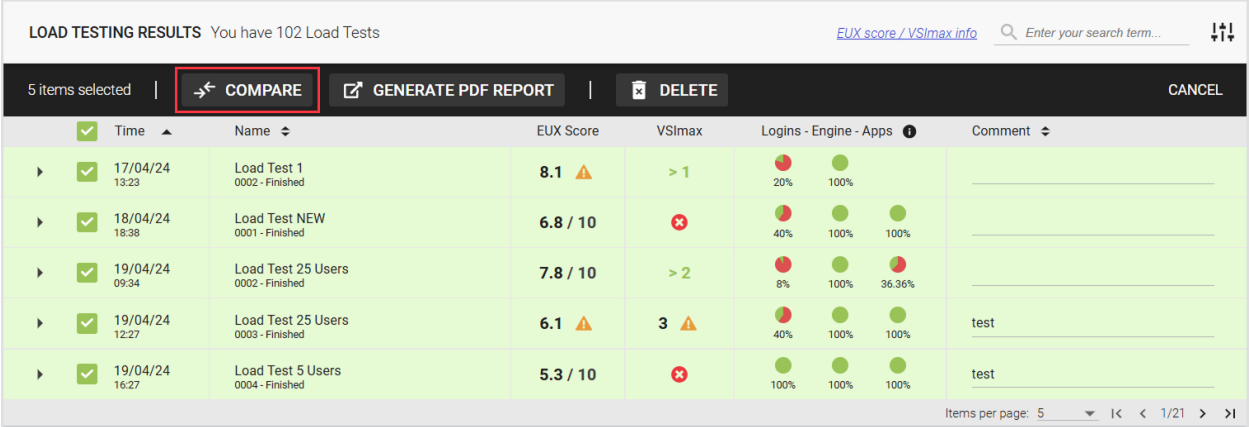
Quick Results View

When you compare 2 Load Tests, you'll see a banner with quick results. The banner includes the same measurements as the Compared Test Results.



Compared Tests

To view the test results for the compared tests, select up to 5 tests and click **Compare**.



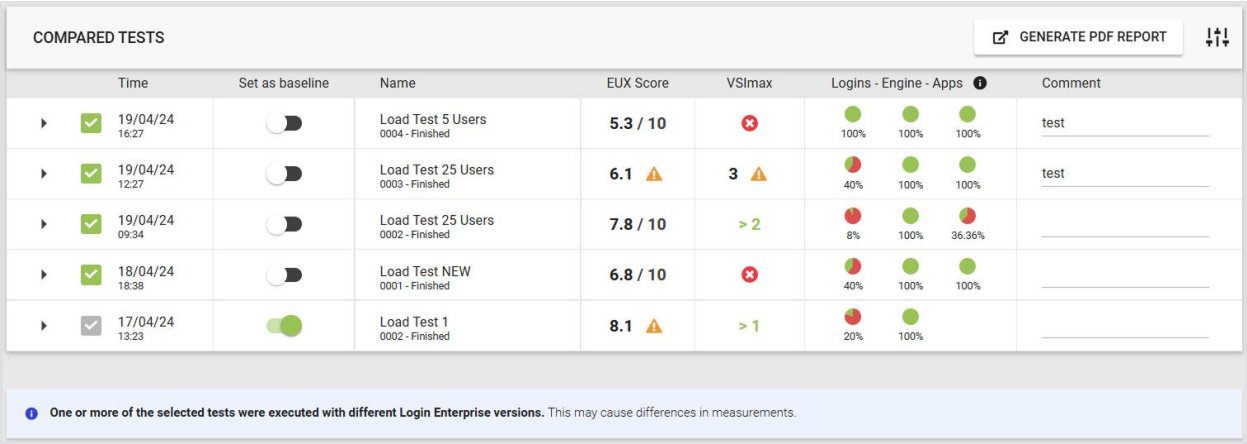
The following page displays the Test summary, Overview (EUX Score/VSImax, Logon Time, and Application Response Time), Charts, and Events.

Compared Test Summary

- Displays all selected Tests with an option to designate one Test as the baseline.
- The selected Tests are compared against the Baseline Test.
- By default, the oldest Test is set as the baseline.
- Tests are sorted by date, with the oldest at the bottom and the newest at the top.
- In the comparison tables, columns are sorted by date from left (oldest) to right (newest).
- Baseline Test results are highlighted in green in the header.
- If you change the baseline Test, the highlighting and the percentage differences will update accordingly, but the sorting will remain unchanged.

NOTE: You can include or exclude a Test from the comparison without needing to go back a step. For this, unselect the preferred Test on the Results page. The Test will be removed from the comparison and will not appear in the PDF report if generated.

You cannot deselect the Baseline Test. You cannot set a deselected Test as the Baseline.



Compared Test Results

EUX Score VSImax

This is where you can see the results of all Tests, even if the comparison isn't possible, e.g., missing data in the Baseline Test.

NOTE: When comparison isn't possible, values are shown without percentage differences.

OVERVIEW | CHARTS | EVENTS

EUX SCORE / VSIMAX — Knowledge base

Measurement	7 users 10m ram... d duration 0001 - Finished	7 users 10m ram... d duration 0004 - Finished	7 users 10m ram... d duration 0005 - Finished	7 users 10m ram... d duration 0009 - Finished
EUX Score	6.6	(-2%) 6.5	(-12%) 5.8	(-17%) 5.5
EUX Baseline Score	-	-	-	6.6
EUX Steady State Score	-	-	-	5.7
VSImax	6	(0%) 6	7	6

Logon Time

This is where you can see various stages and metrics associated with a user's login process on a Windows-based system. Here's a breakdown of what each metric measures:

- **Display protocol connection**: Measures the time from when the launcher starts the session to when profile loading begins. This metric is found in Windows Event Logs.
- **Profile load**: Measured using User Profile events from the Windows Event Viewer.
- **GPO processing**: Only measures synchronous group policies. Asynchronous ones have minimal impact on user experience and are therefore not measured.
- **Total login time**: Measures how quickly a user account can log in, load the user profile, and become fully interactive.

LOGON TIME Average duration (in seconds)

Measurement	7 users 10m ram... d duration 0001 - Finished	7 users 10m ram... d duration 0004 - Finished	7 users 10m ram... d duration 0005 - Finished	7 users 10m ram... d duration 0009 - Finished
Display Protocol Connection	2.3	(-30%) 3	(-57%) 3.6	(-87%) 4.3
Profile load	0.1	(0%) 0.1	(-100%) 0.2	(0%) 0.1
GPO Processing	0.1	(0%) 0.1	(-100%) 0.2	(-100%) 0.2
Total login time	4.9	(-37%) 6.7	(-61%) 7.9	(-57%) 7.7

Application Response Time

- The first row always shows the "Apps Performance" value based on the average response times of all Baseline Applications.
- The "Apps Performance" value is shown only if the Test includes all Applications and measurements present in the Baseline.
- The average response time is calculated solely for the Baseline Applications and measurements, excluding any additional ones.
- Tests that do not include all the Baseline Applications and measurements will not have a value in the "Apps Performance" field.
- Results per Application are displayed below the total.
- Only Applications present in the Baseline Test are shown.
- If you switch the Baseline Test, the Applications shown will be updated accordingly.

APPLICATION RESPONSE TIME Average duration (in seconds)

Measurement	7 users 10m ram... d duration 0001 - Finished	7 users 10m ram... d duration 0004 - Finished	7 users 10m ram... d duration 0005 - Finished	7 users 10m ram... d duration 0009 - Finished
Apps performance	0.487	(-3%) 0.515	(-52%) 0.772	N/A
Notepad				
Measurement	7 users 10m ram... d duration 0001 - Finished	7 users 10m ram... d duration 0004 - Finished	7 users 10m ram... d duration 0005 - Finished	7 users 10m ram... d duration 0009 - Finished
Start	0.289	(+11%) 0.258	(-18%) 0.342	(-269%) 1.067
Wordpad				
Measurement	7 users 10m ram... d duration 0001 - Finished	7 users 10m ram... d duration 0004 - Finished	7 users 10m ram... d duration 0005 - Finished	7 users 10m ram... d duration 0009 - Finished
Start	0.632	(-7%) 0.676	(-67%) 1.055	(-211%) 1.968
Paint				
Measurement	7 users 10m ram... d duration 0001 - Finished	7 users 10m ram... d duration 0004 - Finished	7 users 10m ram... d duration 0005 - Finished	7 users 10m ram... d duration 0009 - Finished
Start	0.539	(-14%) 0.612	(-71%) 0.92	-

Charts

This is where you can see and interact with charts and graphs displaying the EUX Metrics, Logon Time, Application Response Time, and Session Metrics.

Charts include a tree structure that lets you expand or collapse metric groups and select or deselect multiple items at once. This applies to both single Load Tests and comparisons, and helps streamline the selection of complex measurement categories such as application response times, session metrics, and platform metrics.

NOTE: The Display options section is only available in single Load Test views. It does not appear in the Compared Test views.

TIP: You can interact with the Charts in different ways using the Chart Control. For more information, see the Chart Control.

Events

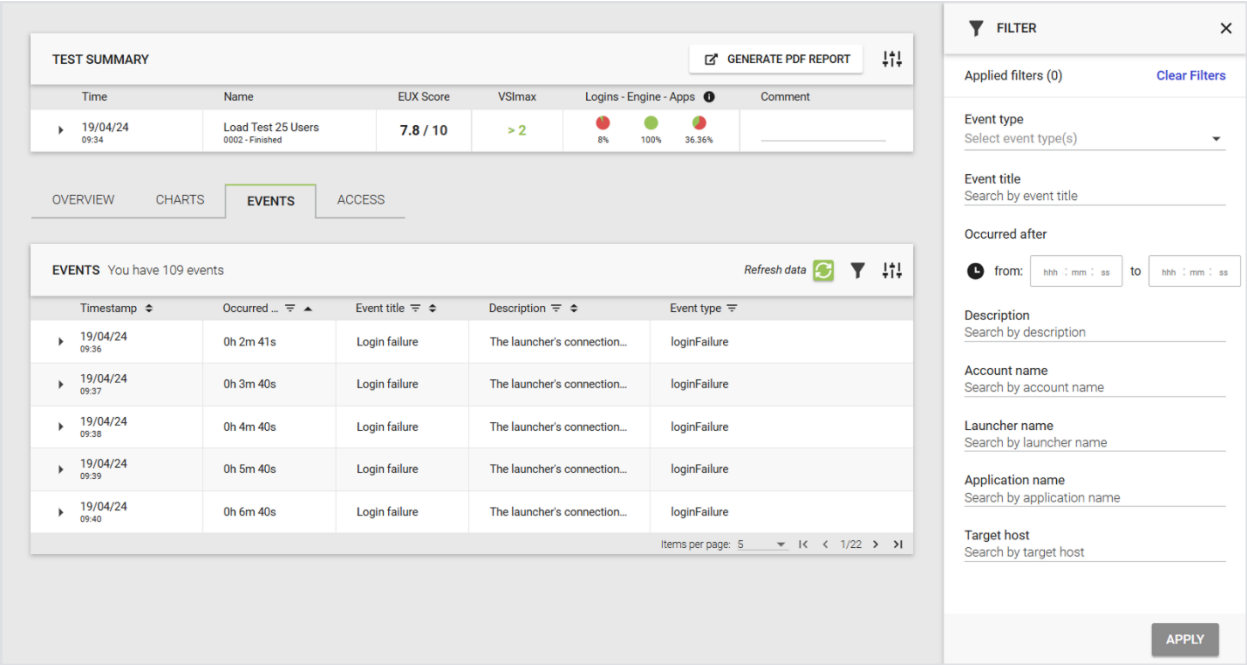
The Events tab displays a table of key events across selected Load Tests. Use this tab to identify patterns and troubleshoot issues, including when comparing Load Tests. The table includes the event date, time since test start, Test run name, and event title. You can filter results to show specific tests or event details.

Use the event filter on the Test run results page to refine results for single or compared Test runs. You can apply filters from:

- The filter sidebar
- Column headers

You can filter by:

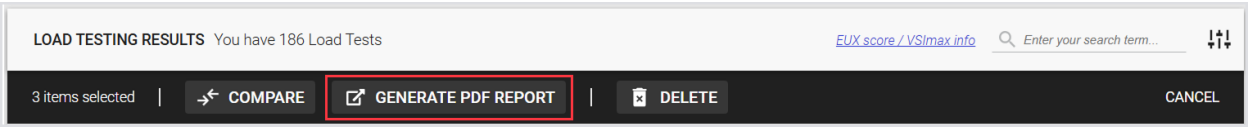
- Event type
- Event title
- Occurred time
- Description
- Account name
- Launcher name
- Application name
- Target host



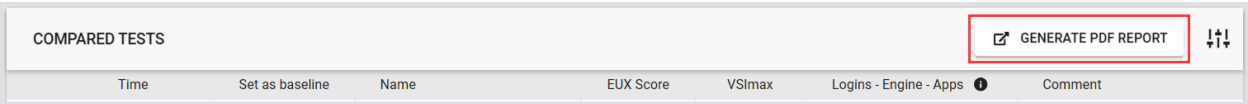
Generating a Test PDF Report

You can generate and download a PDF report with the results comparing up to 5 Load Tests in one of the following ways:

1. Using the toolbar on the **Load Test Results** page:



2. Using the **Generate PDF Report** button on the **Compare** page:



3. In the **Generate PDF report** pop-up, configure what information the report will include. For details, see Configuring a Test PDF Report. To learn about what the report consists of, see the Test PDF Report Breakdown.

Configuring a Test PDF Report

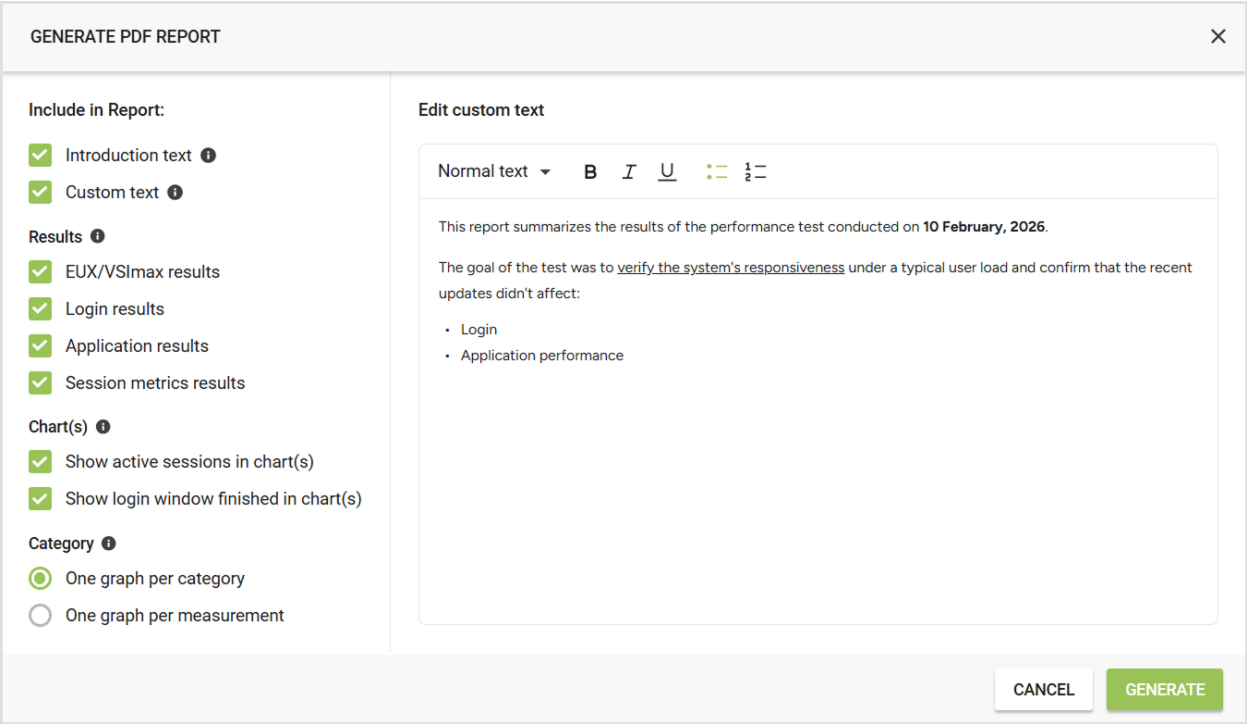
You can choose what you would like in your Load Test PDF report. Once you click **Generate PDF report**, you can configure the following:

Setting	Description	Options / Details	Notes
Baseline test	Select the test run to use as the baseline.	—	• On the Test Results page, this option isn't included. The baseline test is determined by the test selected on the page. • Available only for multiple test reports.
Introduction text	Include general information about Login Enterprise at the beginning of the report.	• *Include introduction text*: Adds the standard intro section. • *Exclude introduction text*: Omits it from the report.	—

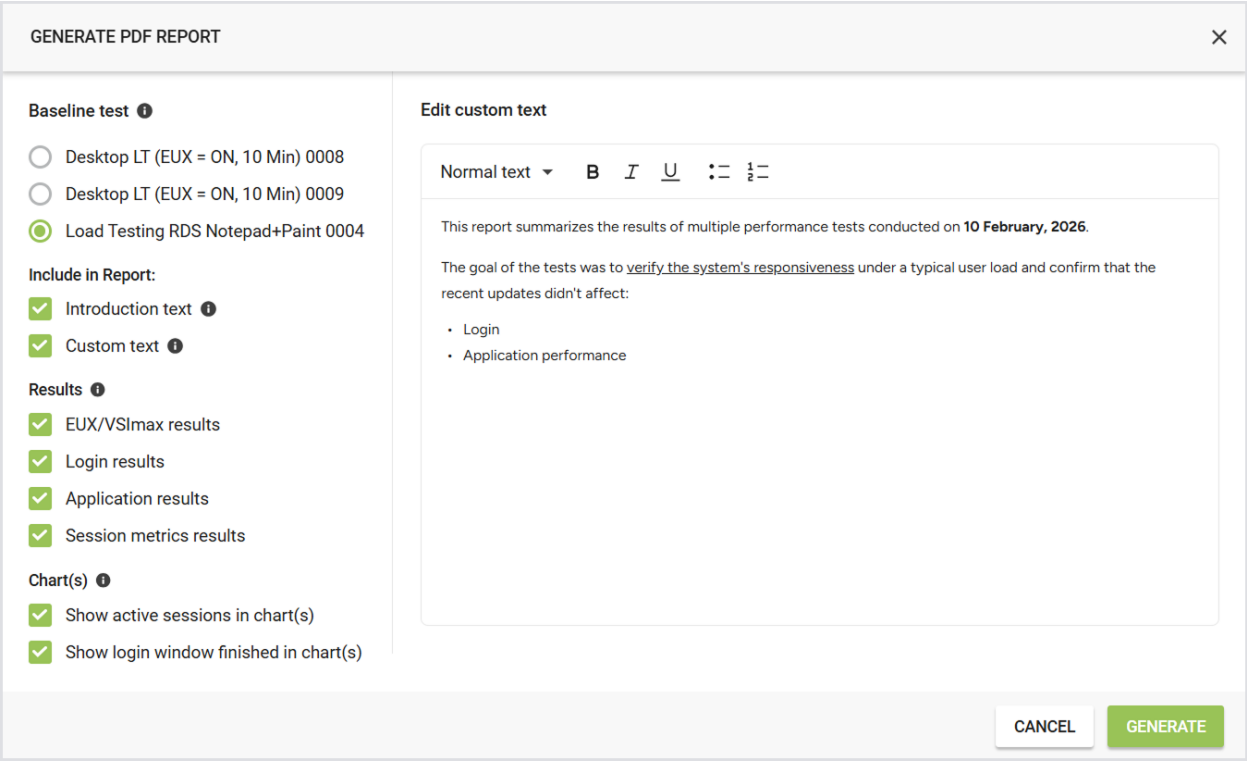
Custom text	Add your own text to the PDF report. Useful for describing what was tested, why, or summarizing conclusions.	• *Include custom text*: Opens a pop-up to enter your content.
		• *Exclude custom text*: Skips this section.
			• The text editor has the following options:
			  ○ Font size (heading or normal text)
			  ○ **Bold**
			  ○ *Italic*
			  ○ <u>Underlined</u>
			  ○ Bullet points
			  ○ Numeric lists
			• Whatever is entered in the custom text section will be added to the beginning of the report, directly under the introduction text.
			• Use Enter to create paragraph breaks. If an uninterrupted block of text exceeds one page, the Load Test PDF will generate a blank page. The custom text section supports up to five pages.

Results	Choose which test results to include or exclude.	Available result types: • EUX score and VSImax results • Login results • Applications results • Session metrics results	—
Charts	Configure what graphical data to include in the report.	• *Show the Login window finished in charts*: Marks when the login window closes. • *Show Active sessions in charts*: Adds a line showing active sessions.	—
Category	Define how measurement data is grouped in charts. This does not apply to session metrics.	• **One graph per category**: Combines all measurements of the same category (e.g., EUX score, Logins) into one chart. • **One graph per measurement**: Displays each timer separately (e.g., if EUX score has 7 measurements and Logins 4, each has its own chart).	• When selected, categorization is done per application. • Available only for reports of a single test. • When comparing multiple tests, only *one graph* is used *per measurement*.

A PDF report configuration with the results of a **single** Load Test looks as follows:



A PDF report configuration with the results of the **compared** Load Tests looks as follows:



Test PDF Report via the Public API

You can download the Load Test PDF report via the Public API using the following endpoint: `/publicApi/v7/reports/load-test-runs/pdf`. For more information, see Accessing the Public API.

Test PDF Report Breakdown

The Test PDF report summarizes the findings obtained from the testing process. It provides an overview of the Test specifics, results, and measurements. The report is divided into the following sections:

- **General information**: Including Introduction, EUX score, and Metrics.
- **Test specifics**: Such as the product version, EUX version, VSImax version, Connector, Launcher group, number of users, Login window, Test duration, Date, and Workload.
- **Test results**: all the results, such as Success rates (table), EUX score (table), and VSImax (table).
- **Charts**: Including the EUX score, EUX timers, and Login-specific charts with Display protocol connection, Profile load, GPO processing, and Total login time.
- **Application-specific charts**: Depending on the scripts/workloads attached to the Test, the report will contain a chart for each timer.
- **Application response time table**: For single Load Tests, a table lists the average response time for each timer. This provides a summary of application performance not directly shown in the charts.
- **Session Metrics charts**: Depending on the Session Metrics linked to the Test, each session metric will have its own chart.

NOTE: When comparing Tests, if an application is present in the Baseline Test but missing in a compared test, the corresponding measurement lines (Active Sessions, Login Window Finished) are not displayed in the PDF chart. The test is also excluded from the chart legend.

Active sessions and login window finished:

- In a single Load Test PDF report, all charts will include lines for active sessions and the login window finished line.
- In a PDF report comparing two Load Tests, all charts will include lines for active sessions, but none will include the login window finished line.

Comparing logic:

- The report will determine which Test is compared to which, mirroring the UI. To switch the Tests in the report, switch them in the UI first, then generate the PDF report.
- The Test considered as the Baseline will be labeled "Baseline" in tables.

EUX & VSImax-related messages

The EUX and VSImax information, warning, and error messages are shown in the comment column within their respective tables.

Export Data

Exporting data for Load Tests follows the same procedure as Continuous Tests. For detailed steps, see the Continuous Testing Results section.

EUX Raw Data

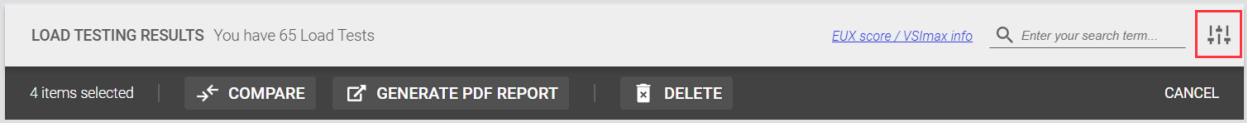
Similarly, refer to the Continuous Testing Results section.

Tips and Tricks

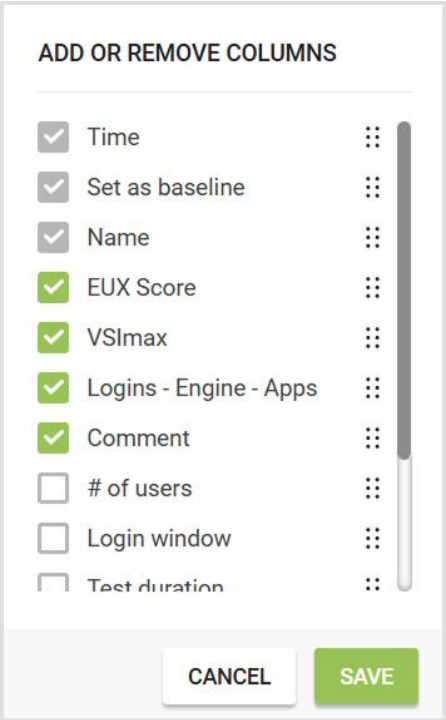
Customizing Table Columns

You can customize the Test Results table by adding, hiding, or sorting columns.

- **Configuration icon**: In the top right corner of the **Load Test Results** table, click the configuration icon to open the column management pop-up.



- **Adding columns**: Add additional columns to the table. The following columns are available:
  - Connector (Type)
  - Login Window
  - # of Users
  - Test Duration
  - EUX Steady State
  - EUX Baseline
- **Hiding columns**: Hide columns you do not want to display using the configuration options.
- **Sorting columns**: To sort columns, drag and drop them into your preferred order.



NOTE: The Time, Set as Baseline, and Test Name columns are fixed and cannot be hidden or moved.

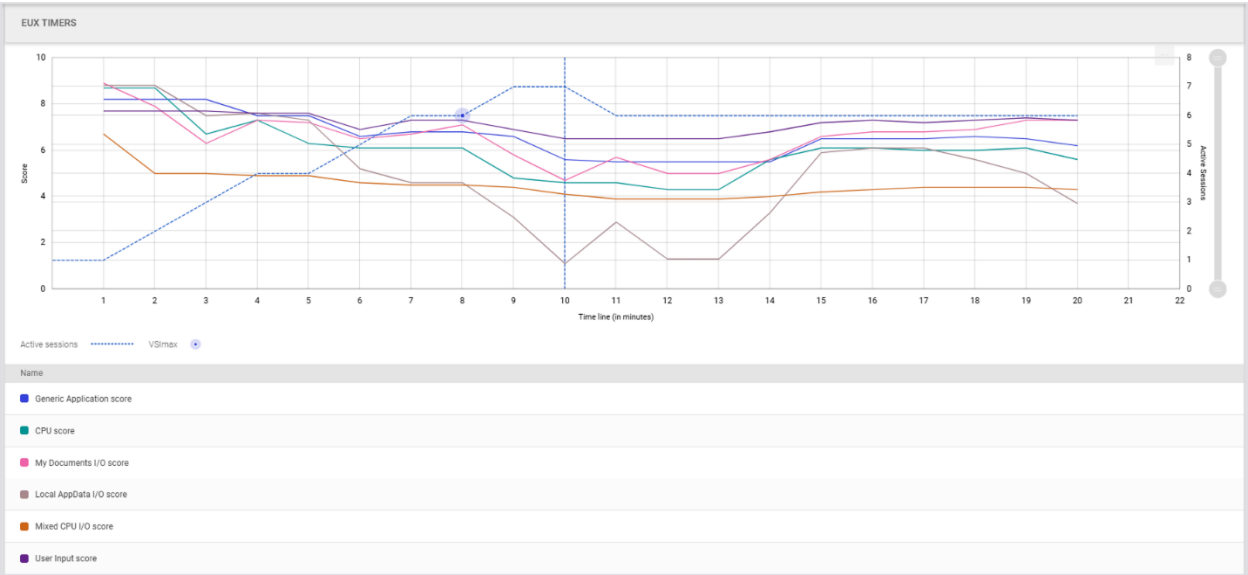
Your configuration settings are saved in your local storage, i.e., your column preferences will persist when you navigate away from the page or log out and back in.

The configuration may reset if you use a different browser or clear your browser history.

Chart Legends with a Table

For a single Load Test, the chart legend is displayed in a table format:

- **Chart Legend Table**: Column 1 shows the Test name.
- **Active Sessions and VSImax Indicators**: Appear between the chart and the table. When the active sessions line is disabled, the associated icons are hidden.
- **Highlighting Table Rows**: Highlights the corresponding line on the chart and fades the others. The selected row is greyscaled.
- **Active Sessions Line**: Shown as a blue dotted line, replacing the previous green line.



NOTE: In the Single Test view, the chart table does not appear when the chart contains only one result. This could be the case with the EUX chart or when selecting one chart per measurement. In the Test Compare view, the table is always displayed, even if the graph contains only one result for one of the Tests.

For the compared Load Tests, the chart legend is also displayed in a table:

- **Chart Legend Table**:
  - Column 1: Test name
  - Column 2: VSImax value (if applicable)
  - Column 3: Active sessions (dotted line matching the measurement line color)
- **Disabling Active Sessions**: Hides the Active Sessions and VSImax columns.
- **Highlighting Table Rows**: Highlights the corresponding line on the chart, fading others. The selected row is greyscaled. The following lines are highlighted:
  - Measurement line
  - Active sessions line (including Login Window Finished)
  - VSImax (if available)
- **Active Sessions Line**: Shown as a dotted line matching the measurement line color.

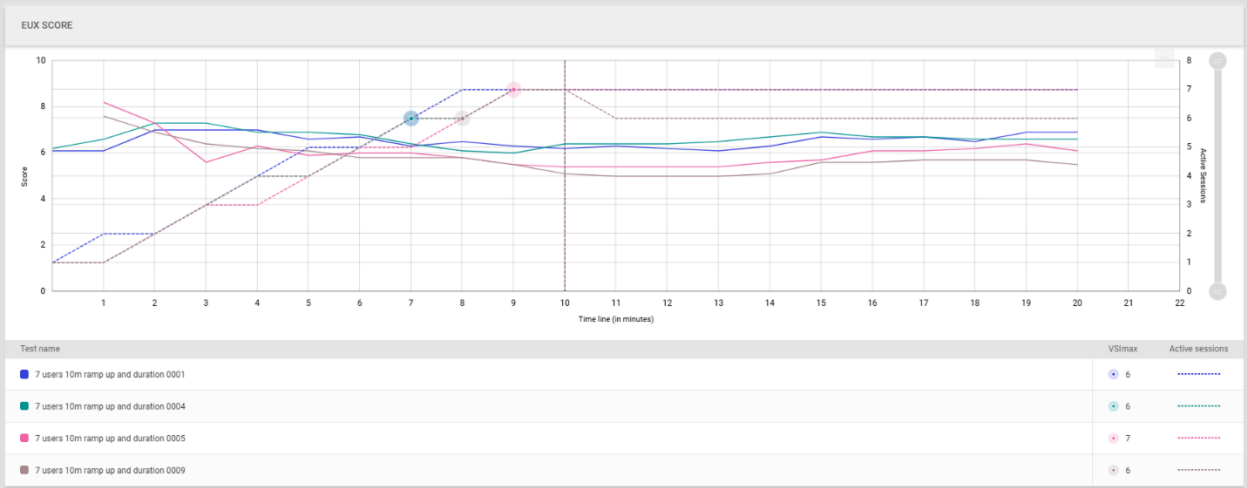
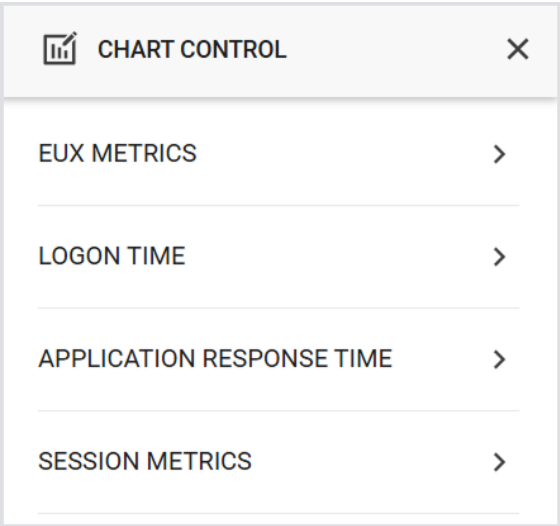


Chart Control

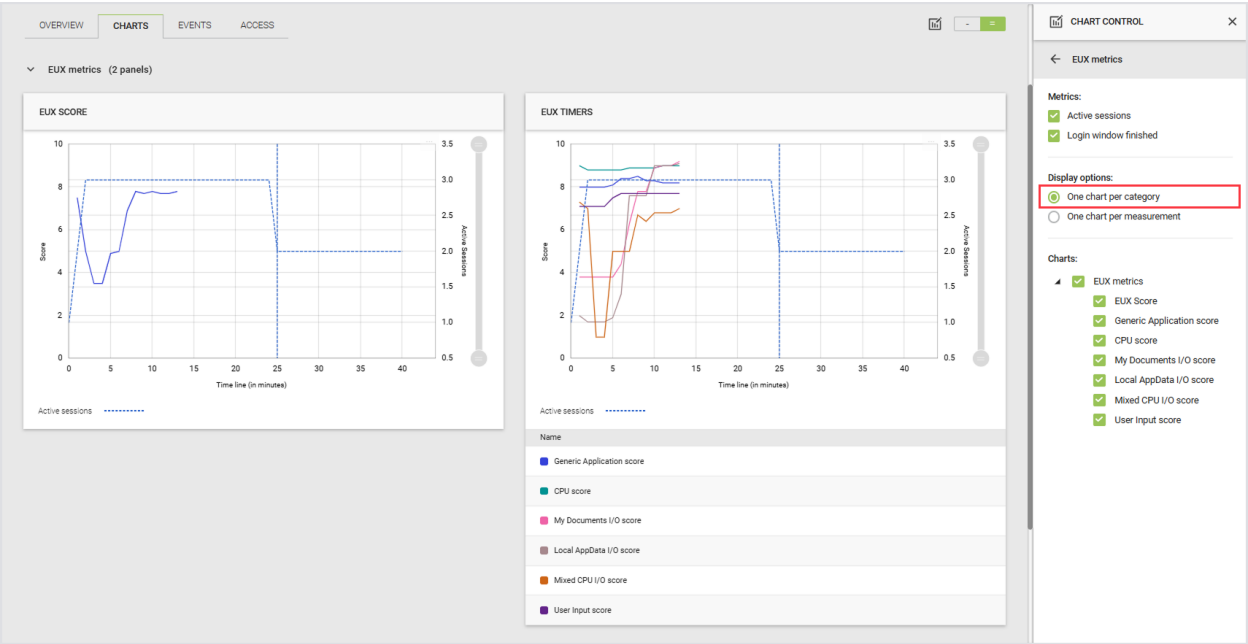
You can choose from 4 data sources for your charts:

- **EUX Metrics**: Built-in EUX measurements.
- **Logon Time**: Login phase times.
- **Application Response Time**: Timers from your Application scripts.
- **Session Metrics**: Custom-configured session metric data.

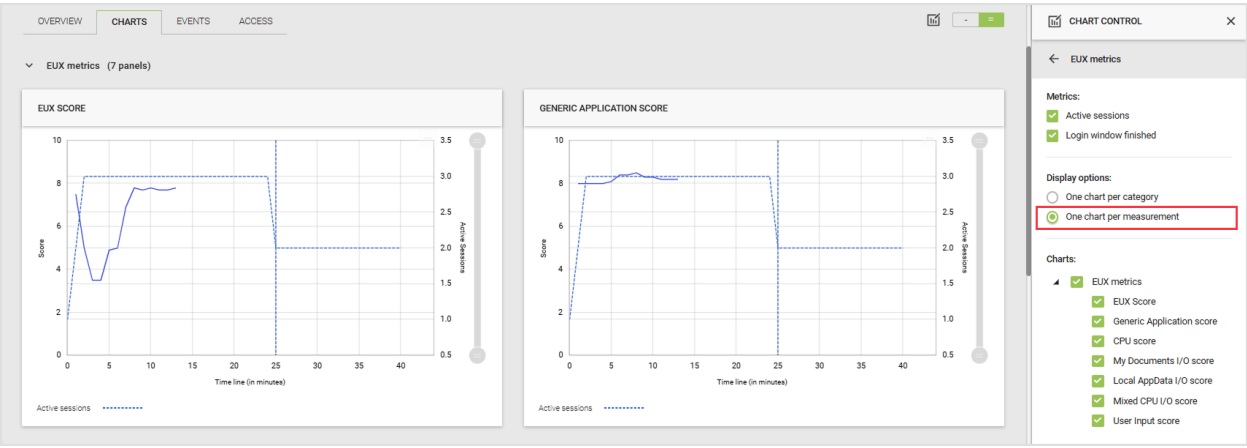


For the **EUX Metrics**, **Logon Time**, and **Application Response Time**, you can switch between 2 ways of presenting data:

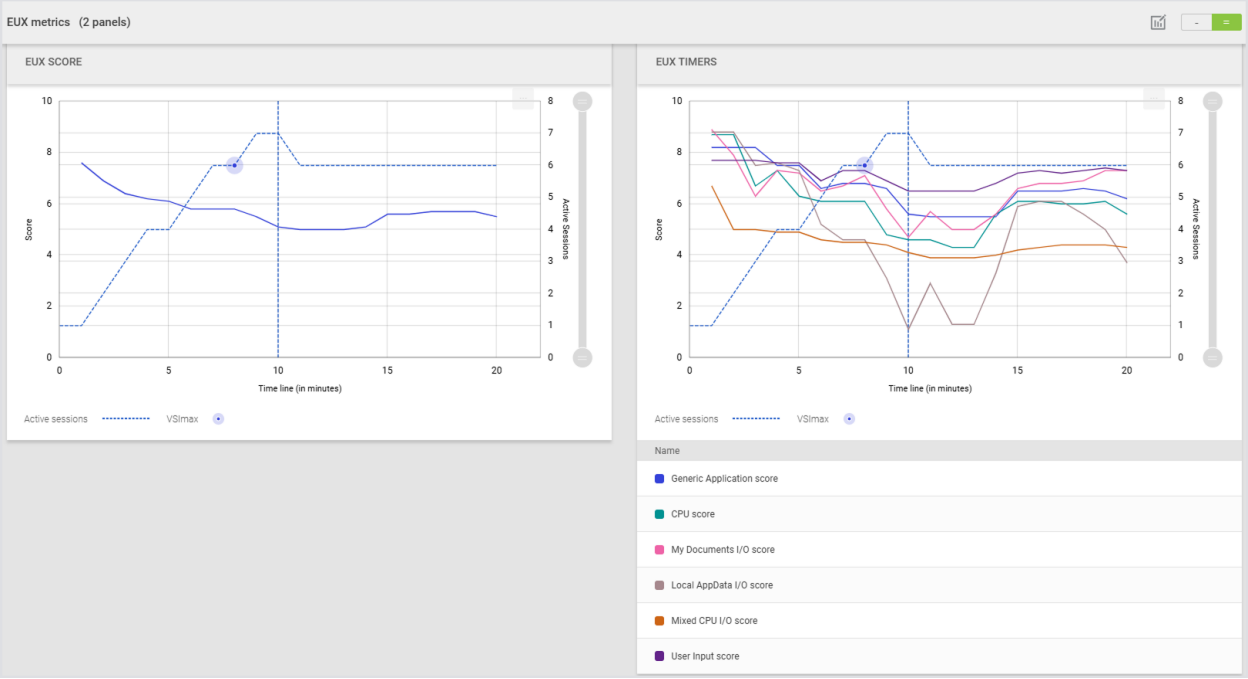
1. **One graph per category**: Displays each measurement in a separate graph:



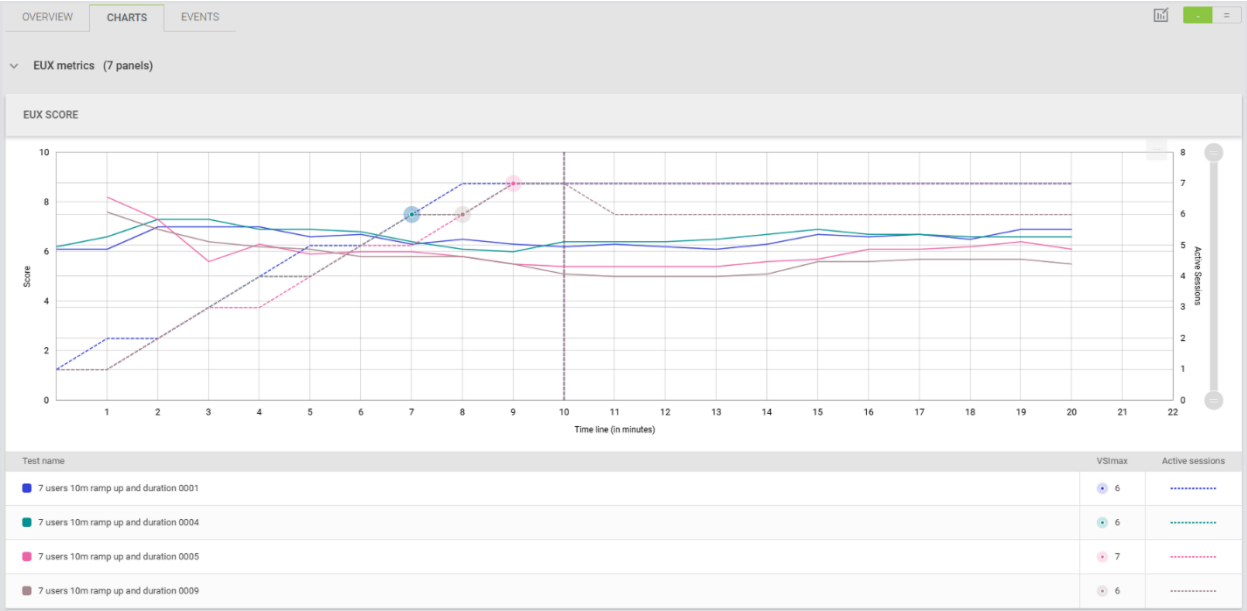
2. **One chart per measurement**: Combines multiple measurements into a single graph:



For **Session Metrics**, only one measurement per chart is shown:



You can choose to display charts either one per row or two side by side:



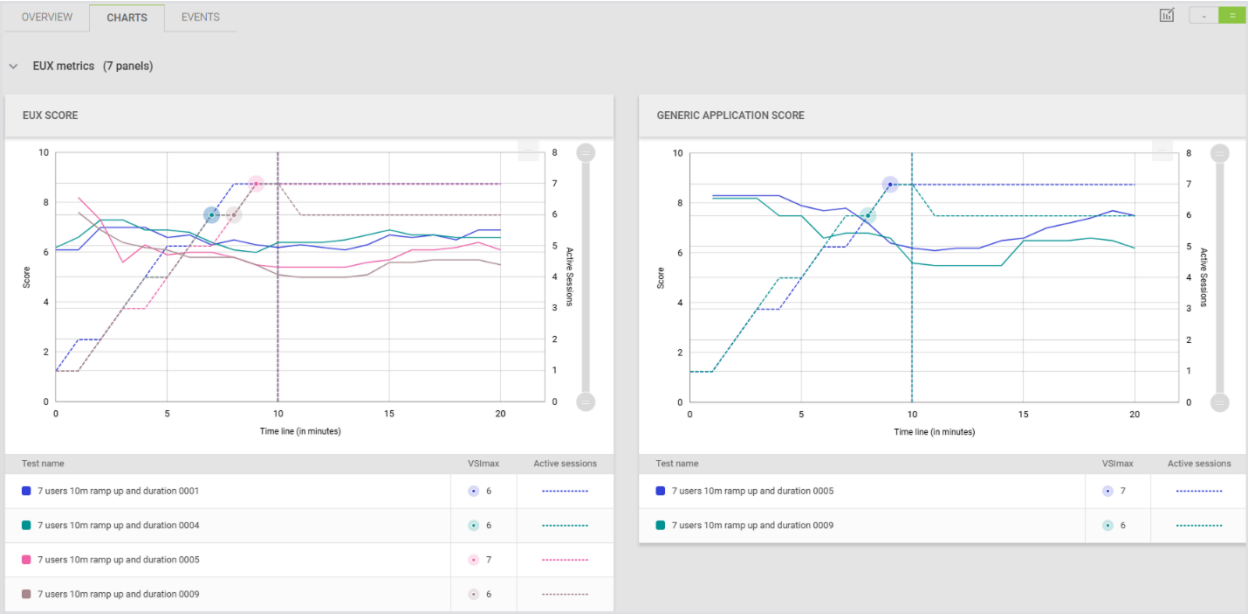
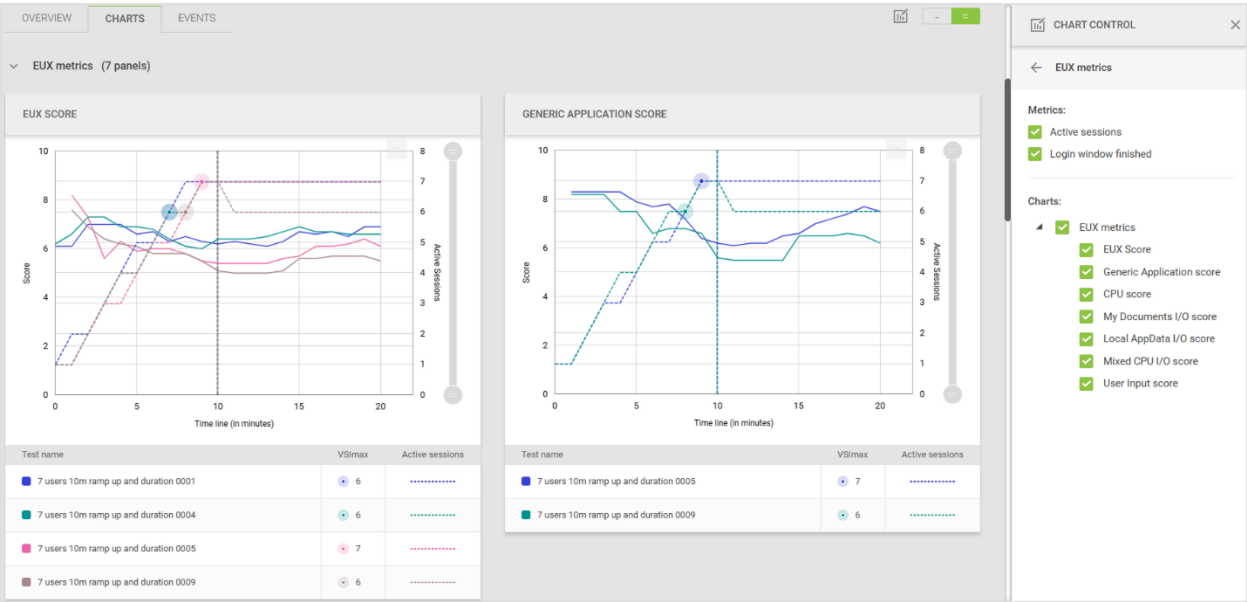


Chart Control lets you navigate and manage chart visibility and metrics. Categories can be expanded to show related charts, and specific metrics, such as active sessions or login duration, can be shown or hidden as needed. Categories without data are displayed but greyed out and cannot be selected.



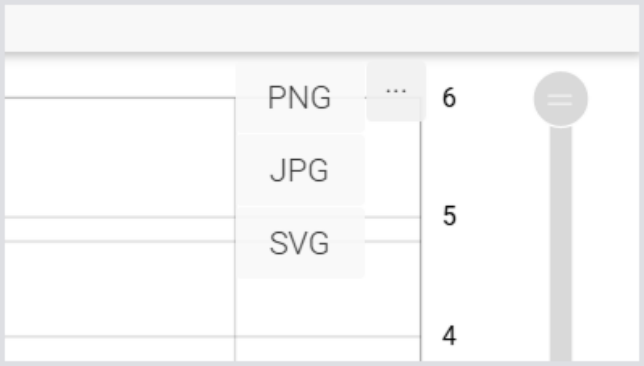
- **Sticky Header**: Ensures the chart control and display options are accessible regardless of your position on the page.
- **Interactive Legend**: Click a legend item, and it will highlight that item in the chart, with other items being greyed out. Click the item again, and it will remove the highlight.
- **VSImax**
    - **Representation**: In the charts, the VSImax is displayed as an orange dot.

    - **Tooltip**: When hovering over the chart, the orange VSImax dot will be displayed before the session count in the tooltip.
    - **Legend**: There is only one item for VSImax in the chart legend, and it applies to all Tests.
    - **Condition**: Be aware that not all Tests will have a VSImax. If the VSImax in the table is denoted as ">5" (greater than 5), it means that during the Test, the performance of the system was acceptable throughout. This indicates that the VSImax is greater than the number of users in the Test, and there will not be an orange icon in the charts for such cases.
- **EUX and Logons**: Behave the same way in the charts as they do in the tables. The charts are always displayed regardless of whether the Baseline Test has data.
- **Applications and Session Metrics**: Depend on the Baseline Test. Charts for these measurements are only shown if the Baseline Test includes the relevant data. For these categories, we will only display the applications and measurements that are present in the Baseline Test. If the other Tests have applications and/or measurements that are not in the Baseline Test, they will not be displayed.
- **Test Name Display:** In the chart legend and tooltip, Test names are truncated in the middle if they are too long.
- **Information Messages**: Displayed in the following scenarios:
    - When one or more Tests were run with different Login Enterprise versions.
    - When one or more Tests were run with different EUX versions.

Downloading Charts as an Image

You can download charts as images for sharing or use in custom reports and presentations. Downloading a chart as an image applies to all charts: Load Test, Continuous Test, and Appliance Health. To download a chart:

1. Locate the Download icon in the top right corner of the chart.
2. Select image format: PNG, SVG, or JPG.
3. Download the image. It will be saved to your Downloads folder, or the directory you've configured as your default Downloads location.



Viewing Application Testing Results

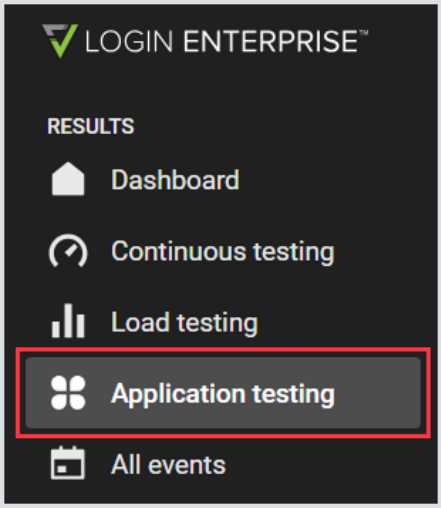
- Introduction
- Single Test
  - Single Test Summary
  - Single Test Results
    - Platform Summary
    - Application Summary
  - Generating a Test PDF Report
    - Automatically
    - Manually
- Compared Tests
  - Compared Test Summary
  - Compared Test Results
    - Platform Summary
    - Application Summary
  - Events
  - Generating a Test PDF Report
- Configuring a Test PDF report
- Test PDF report via the Public API
- Test PDF report breakdown
- Tips and Tricks
  - Customizing Table Columns

Introduction

You can view the results of individual Application Tests and compare up to five Application Test instances. The Application Test Results page provides a comprehensive overview of measurements and detailed events for each Test. You can also download both single and multiple Test results as PDF reports for easy sharing and analysis. See the instructions below for more details.

Single Test

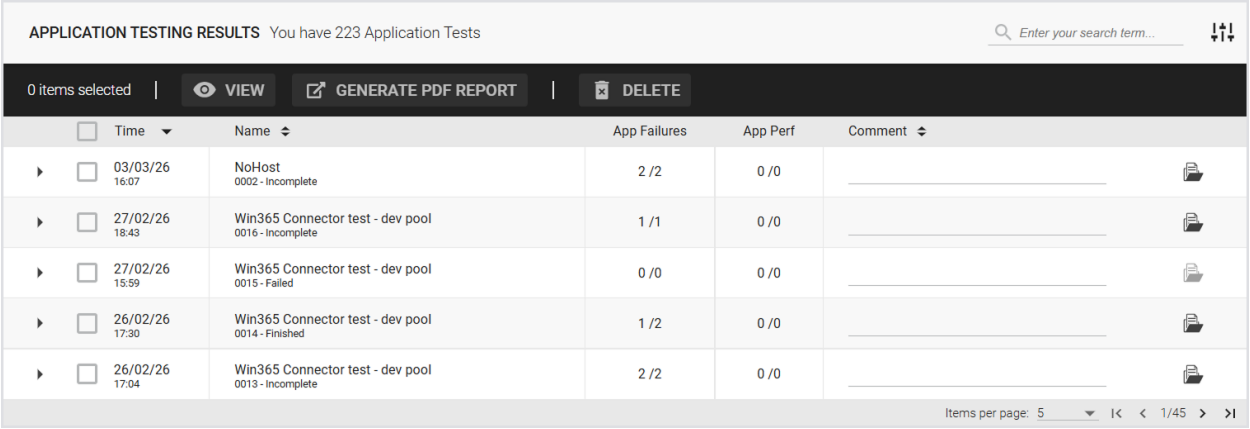
When an Application Test is completed, you can start looking at the results. To do this, in the **Login Enterprise sidebar menu > Results**, navigate to **Application testing**.



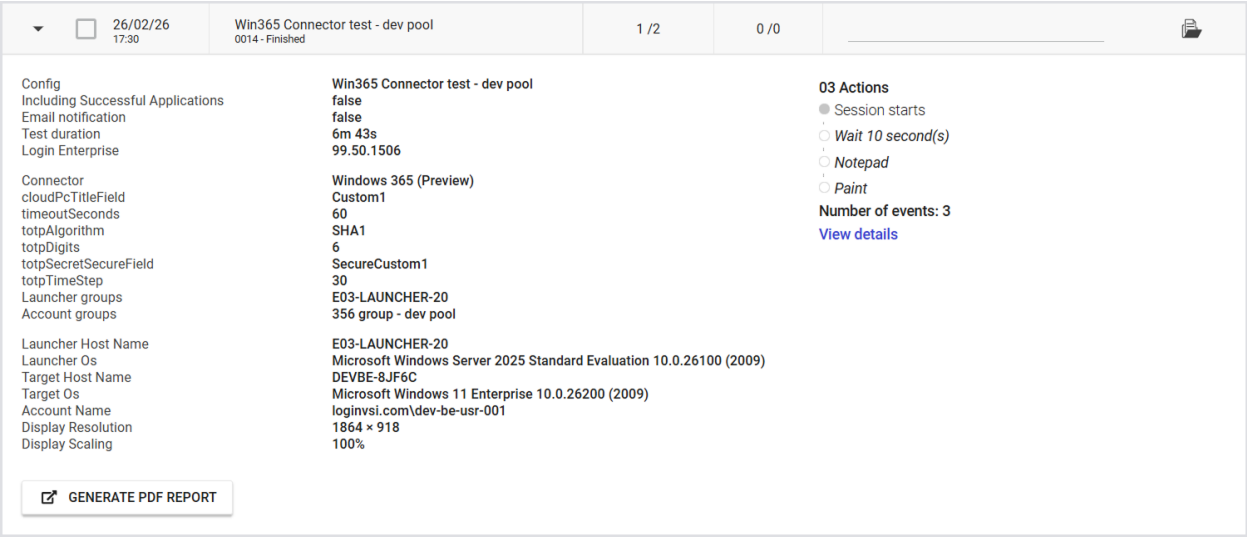
Single Test Summary

The summary page displays a list of previously executed Application Tests and includes the following:

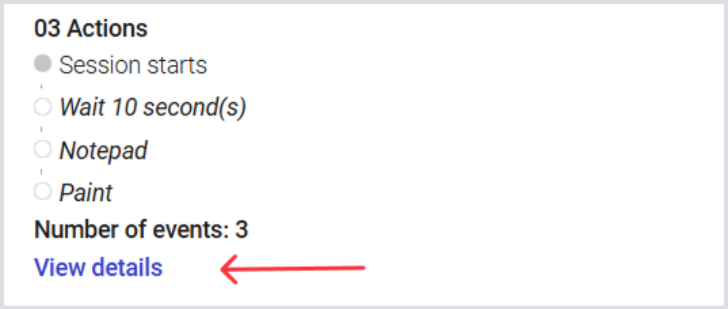
Time	Name	App. Failures	App. Perf	Comment
Date and time of the Test.	Name and sequence number of the Test.	The number of failed applications relative to the total number of applications tested.	The number of out-of-bound measurements, compared to the configured thresholds. The number on the right shows the total number of configured measurements.	The configured comment for the Test.

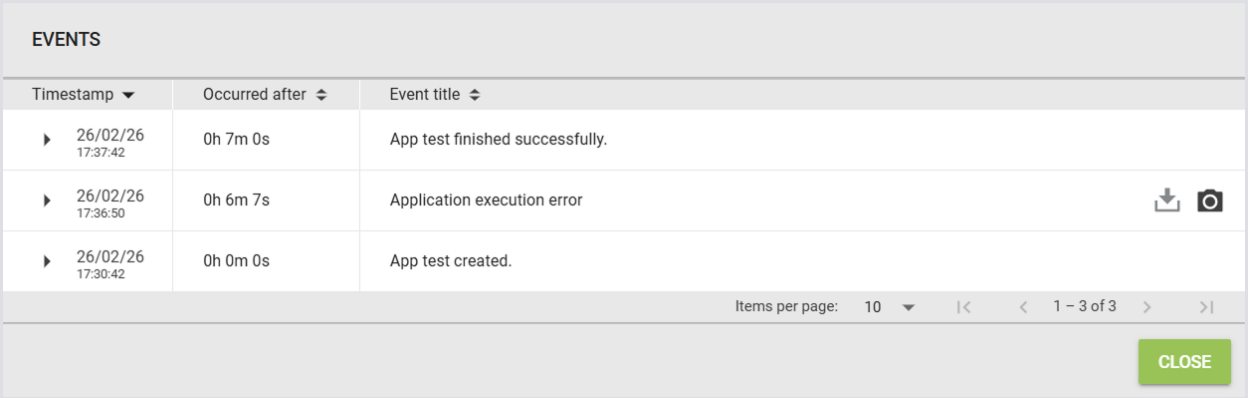


To view the Test details, click on the Test you're interested in:



TIP: View details of events that occurred during the test. Click **View details** to open a pop-up with additional information.





If the Test includes application errors, you can expand the details, download the log file, or open the screenshot using the **camera** icon.

26/02/26 17:36:50	0h 6m 7s	Application execution error

Details	Value
Account	dev-be-usr-001@loginvsi.com
Application	Notepad
Description	An error occurred while running the script: 'The process did not start successfully. Verify its installation or change the start options to fix the detection.'
Launcher	E03-LAUNCHER-20
Locale	English (United States)
Remoting protocol	Rdp
Resolution	1864 × 918
Scale factor	100%
Target host	DEVBE-8JF6C
Target OS	Microsoft Windows 11 Enterprise 10.0.26200 (2009)

Single Test Results

To view the test results for a single test, select a test and click **View**.

APPLICATION TESTING RESULTS You have 223 Application Tests

Enter your search term...

1 item selected | VIEW | GENERATE PDF REPORT | DELETE CANCEL

Time	Name	App Failures	App Perf	Comment
03/03/26 16:07	NoHost 0002 - Incomplete	2 /2	0 /0	
27/02/26 18:43	Win365 Connector test - dev pool 0016 - Incomplete	1 /1	0 /0	
27/02/26 15:59	Win365 Connector test - dev pool 0015 - Failed	0 /0	0 /0	
26/02/26 17:30	Win365 Connector test - dev pool 0014 - Finished	1 /2	0 /0	
26/02/26 17:04	Win365 Connector test - dev pool 0013 - Incomplete	2 /2	0 /0	

Items per page: 5 1/45

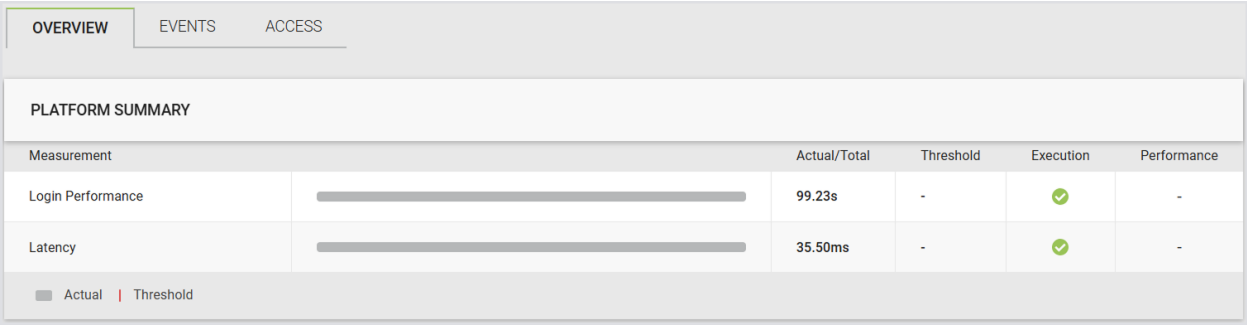
The following page displays the Test summary, Overview (Platform summary and Application summary), Events, and Access.

Platform Summary

Logon performance

Actual	Threshold	Execution	Performance
The actual time it took to log in.	The login time threshold set in the Test.	Green if the login was successful, regardless of the time taken.	• Green if the actual login time was below the threshold or if no threshold was set. • Red if the actual time exceeded the threshold.

Latency				
	Actual	**Threshold**	**Execution**	**Performance**
	The actual latency in milliseconds (ms).	The latency threshold set in the Test, in milliseconds (ms).	• Green if latency was measured. • Red if no latency could be measured.	• Green if the actual latency was below the threshold or if no threshold was set. • Red if the actual latency exceeded the threshold.

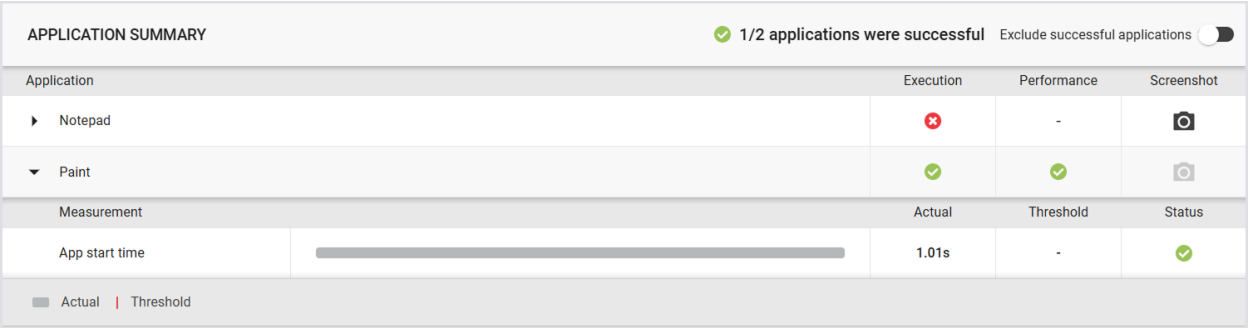


Application Summary

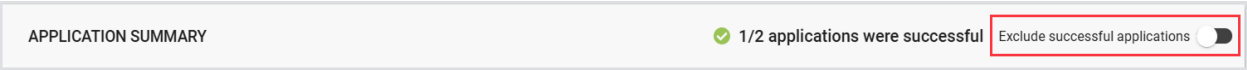
Application(s)		
Execution	**Performance**	**Screenshot**

• Green if the script was successfully executed. • Red if the script had a failure during execution.	• Green if all measurements/timers in the script were below the threshold or if no threshold was set. • Red if one or more measurements exceeded the threshold.	• If a screenshot was taken, click the camera icon to open a pop-up displaying the screenshot. • If multiple screenshots are available, arrows will appear in the pop-up to navigate through them. • A download button is also available to save the screenshot on your desktop.
Measurements (Timers)		
Actual	**Threshold**	**Status**
The actual time recorded for the measurement.	The threshold set in the Test for the specific measurement.	• Green if the actual time is below the threshold or if no threshold is set. • Red if the actual time exceeds the threshold.

By default, the measurements (timers) include the app start time. The app start time is automatically recorded when the START function in the workload scripts is executed. This function is used by default to launch target apps. Additionally, any custom timers defined in the workload scripts will also be displayed in the UI. To learn how to configure custom timers, see the [Scripting Functions Overview](#).



You can choose to show only applications with failed events. This helps you quickly identify failures and troubleshoot issues.

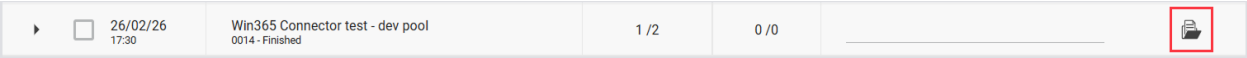


TIP: To scroll horizontally, drag the scrollbar, or place your cursor over the table, hold **Shift**, and use the mouse wheel.

Generating a Test PDF Report

Automatically

A single Application Test PDF report is automatically generated once the Test is finished. Download the report by clicking the **Download** button next to the Test:

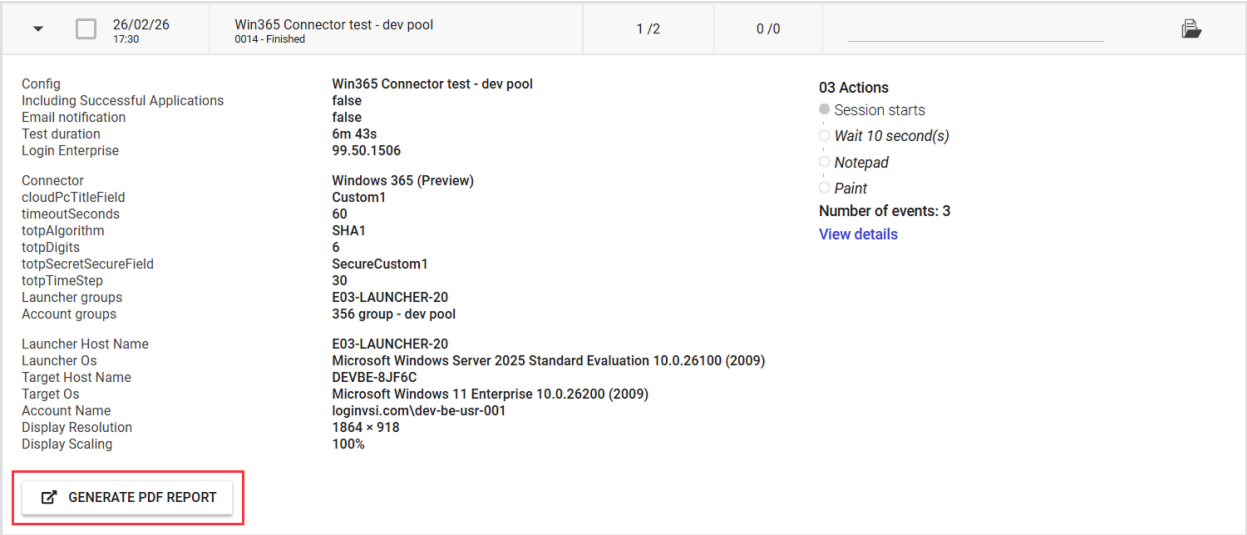


NOTE: This is the report for which you can set email notifications. For more information, see the Report Settings.

Manually

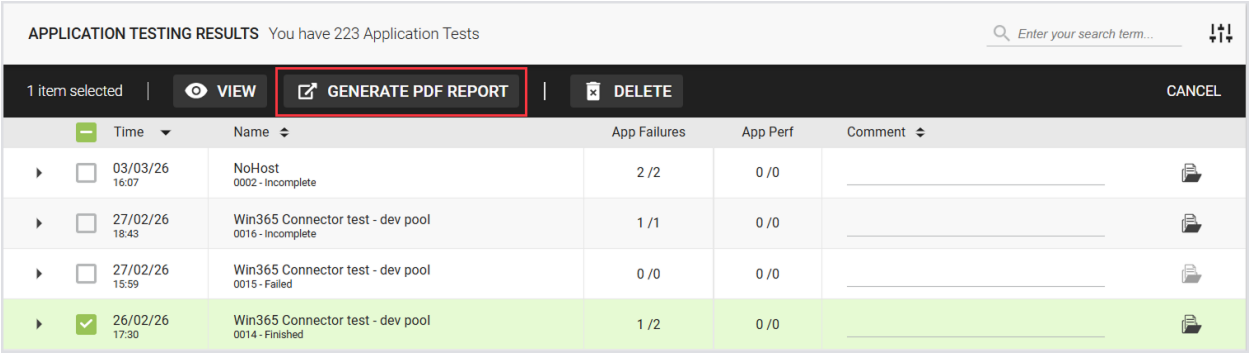
Generate a new single Test PDF report manually or using the Public API. To generate the report:

1. In the **Application Testing results**, click on the Test you're interested in.
2. Click **Generate PDF report** (A browser pop-up will open).



3. In the **Generate PDF report** pop-up, configure what information the report will include. For details, see Configuring a Test PDF Report.
4. Click **Generate**, and then **Save**.

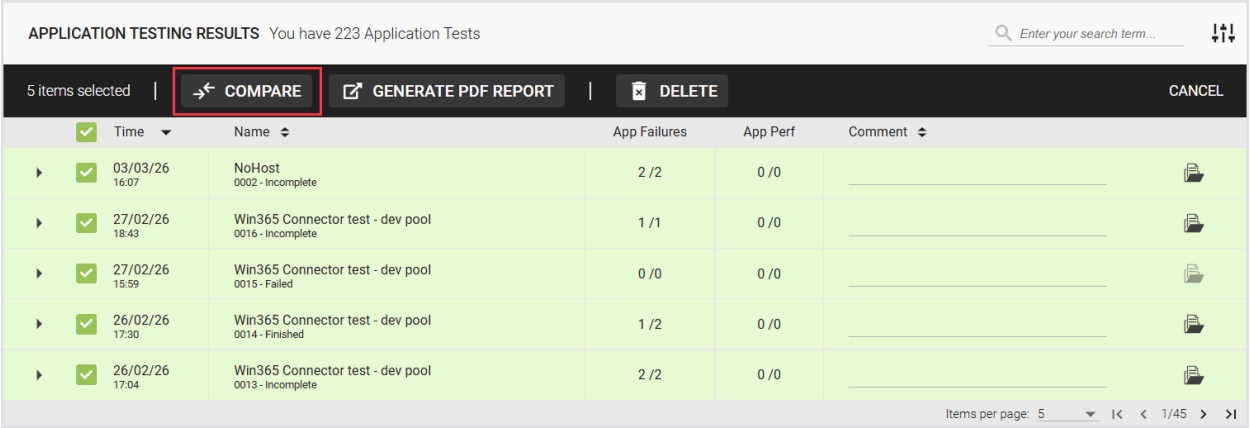
TIP: Alternatively, select the checkbox next to the specific Test, and in the top menu toolbar, click **Generate PDF report**.



NOTE: Email notifications for a single Application Test report generated manually aren't supported.

Compared Tests

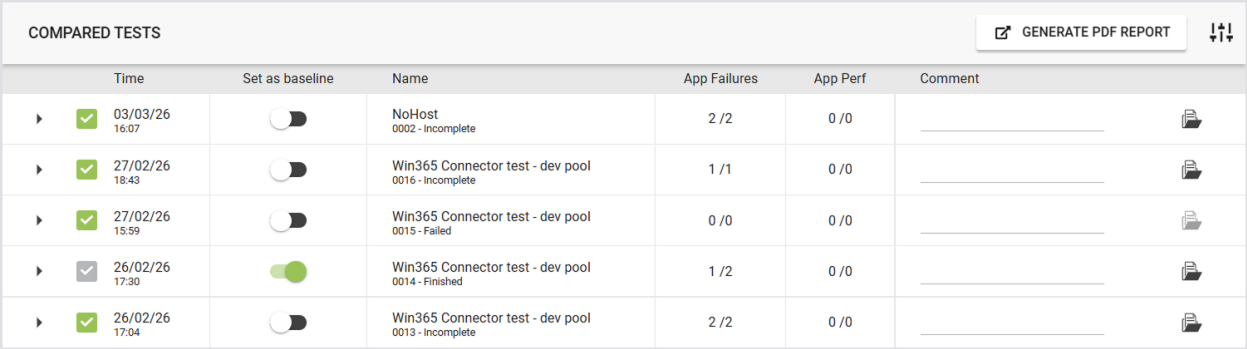
To view the test results for the compared tests, select up to 5 tests and click **Compare**.



The following page displays the summary of the compared Tests, Overview (Platform summary and Application summary), and Events.

Compared Test Summary

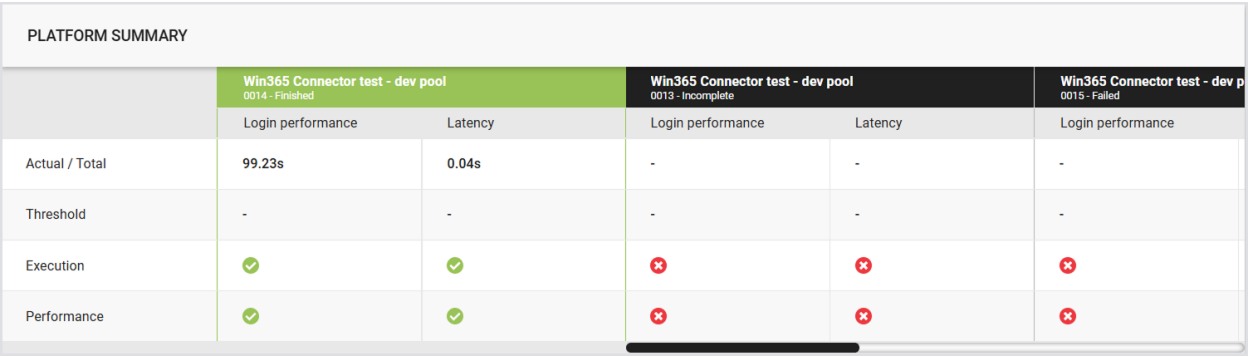
This is where you can select a Baseline Test to compare other Application Tests against. Use the checkboxes to add or remove tests from the comparison.



Compared Test Results

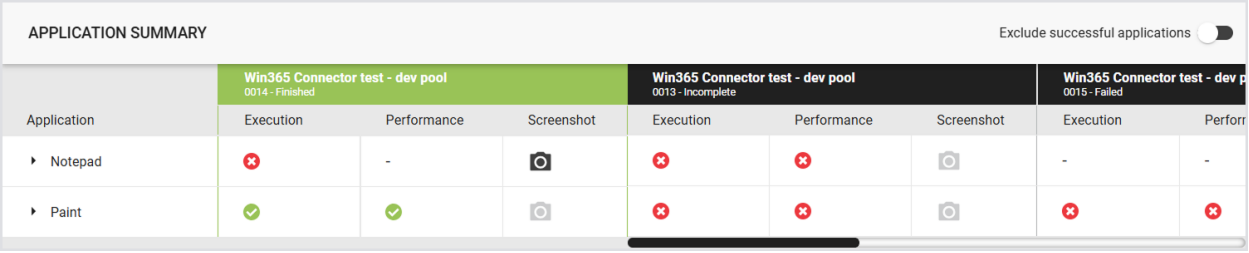
Platform Summary

Displays the login time and latency, along with the percentage difference between the compared Tests. For details, see the [Platform Summary](#).



Application Summary

Compares the results of the Application scripts. For details, see the [Application Summary](#).



NOTE: The results of the Baseline Test are pinned in the first column. The other Test columns are sorted from oldest to newest.

If the results do not fit the screen, scrollbars will appear in the table. The baseline Test column remains pinned and does not move with the scrollbar.

The results of the Baseline Test are the primary reference. All Applications and timers from the baseline are displayed. If a selected Test has additional Applications or timers not present in the Baseline Test, those will only be shown when that Test is designated as the baseline.

The PDF report for the multiple Application Test comparison isn't available yet.

Events

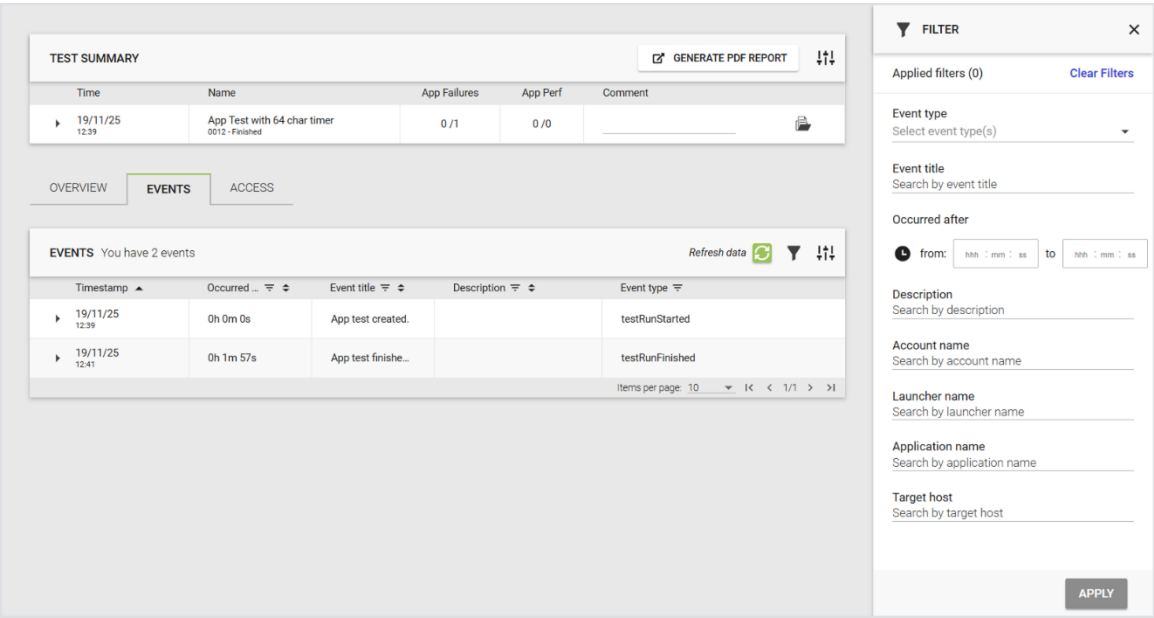
The Events tab displays a table of key events across selected tests. The table includes the event date, time since test start, Test run name, and event title. Use this tab to identify patterns and troubleshoot issues, including when comparing tests.

Use the event filter on the Test run results page to refine results for single or compared Test runs. You can apply filters from:

- The filter sidebar
- Column headers

You can filter by:

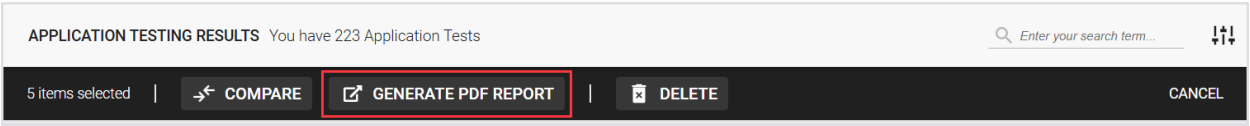
- Event type
- Event title
- Occurred time
- Description
- Account name
- Launcher name
- Application name
- Target host



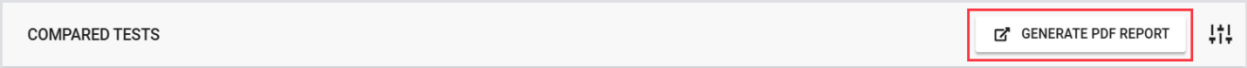
Generating a Test PDF Report

You can generate and download a Test PDF report with the results comparing up to 5 Tests in one of the following ways:

1. Using the toolbar on the **Application Test Results** page:



2. Using the **Generate PDF Report** button on the **Compare** page:



3. In the **Generate PDF report** pop-up, configure what information the report will include. For details, see Configuring a Test PDF Report. To learn about what the report consists of, see the Test PDF Report Breakdown.

NOTE: An Application Test report comparing multiple Tests is not generated automatically. You can generate the report manually or using the Public API.

Email notifications for this type of report aren't supported.

Configuring a Test PDF Report

You can choose what you would like in your Application Test PDF report. Once you click **Generate PDF report**, you can configure the following:

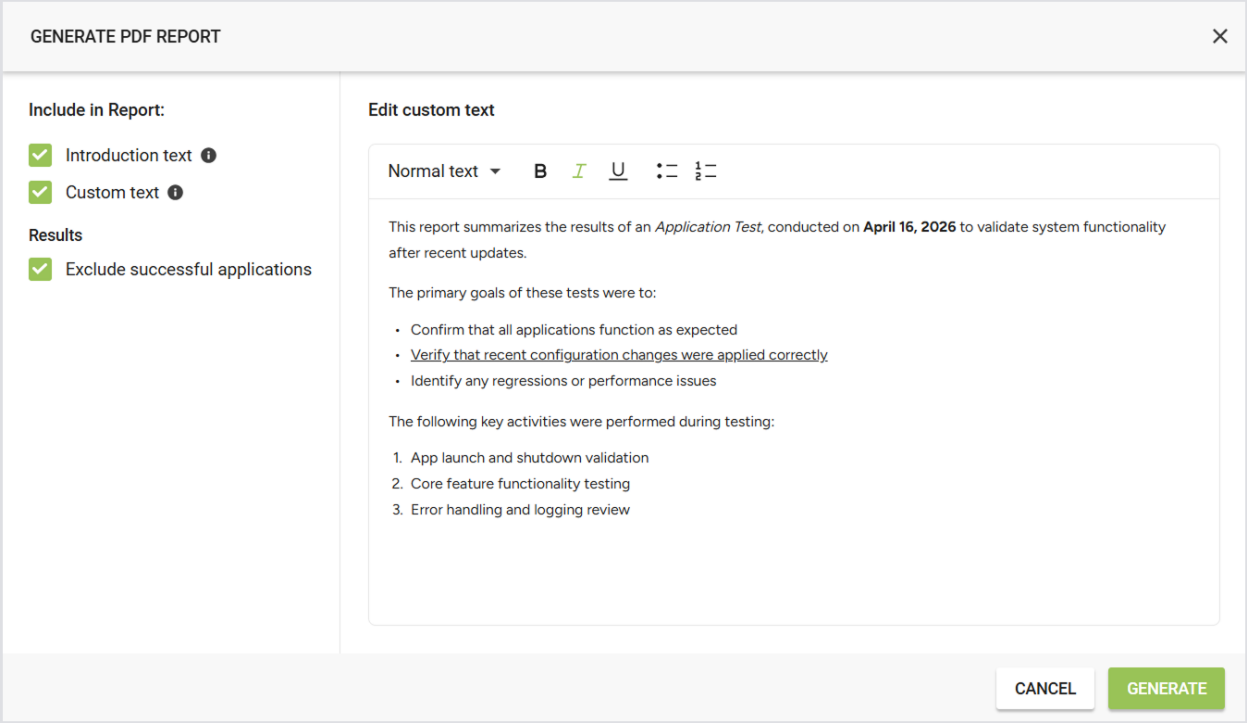
Setting	Description	Options / Details	Notes
Baseline test	Select the test run to use as the baseline.	-	• On the Test Results page, this option isn't included. The baseline test is determined by the test selected on the page. • Available only for multiple test reports.

Introduction text	Include general information about Login Enterprise at the beginning of the report.	• *Include introduction text*: Adds the standard intro section. • *Exclude introduction text*: Omits it from the report.	-

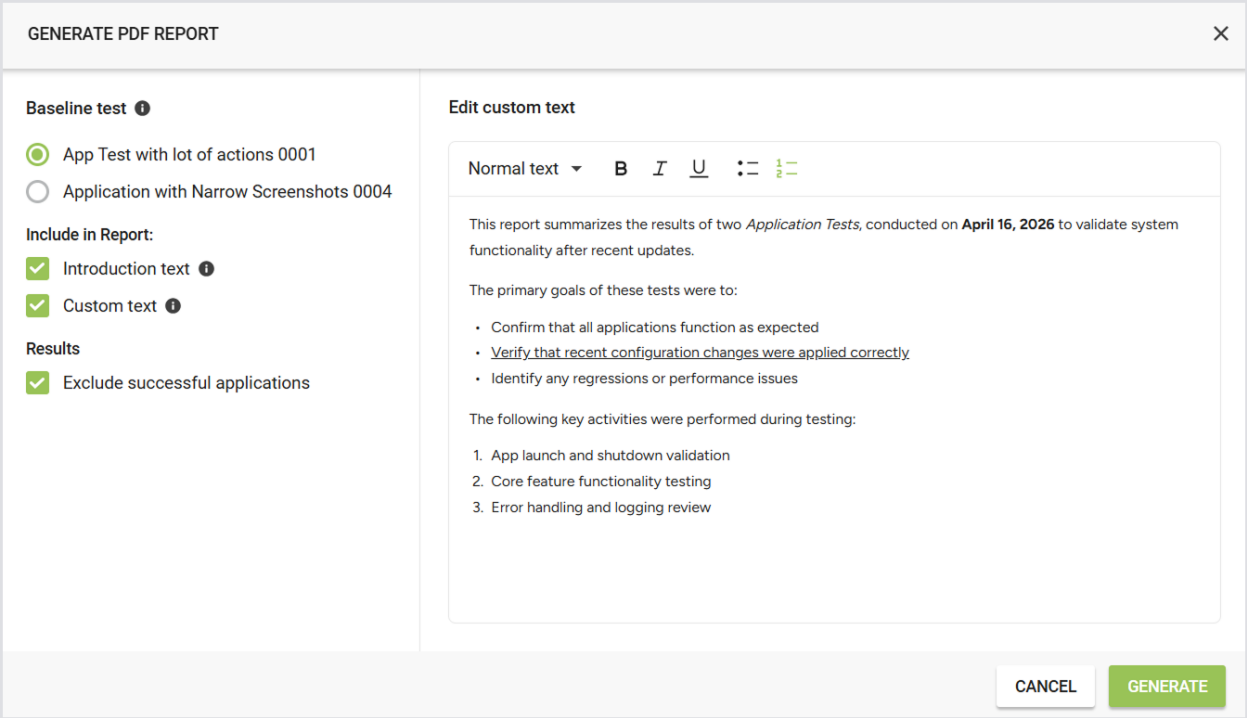
Custom text	Add your own text to the PDF report. Useful for describing what was tested, why, or summarizing conclusions.	• *Include custom text*: Opens a pop-up to enter your content.
		• *Exclude custom text*: Skips this section.
			• The text editor has the following options:
			  ○ Font size (heading or normal text)
			  ○ **Bold**
			  ○ *Italic*
			  ○ <u>Underlined</u>
			  ○ Bullet points
			  ○ Numeric lists
			• Whatever is entered in the custom text section will be added to the beginning of the report, directly under the introduction text.
			• Use the Enter key to create paragraph breaks. If an uninterrupted block of text exceeds one page, the Application Test PDF will generate a blank page. The custom text section supports up to five pages.

Results	Exclude successful applications from the PDF report to keep it concise and focused on relevant information.	• *Include successful applications*: Adds all successful applications to the report. • *Exclude successful applications*: Includes only failed applications in the report.	• Successful applications are included by default (checkbox unselected). • Available for both single multiple test reports.

A PDF report configuration with the results of a **single** Application Test looks as follows:



A PDF report configuration with the results of the **compared** Application Tests looks as follows:



Test PDF Report via the Public API

There are two types of Test PDF reports: old and new.

Old Test PDF report

- Endpoint: `/publicApi/v8-preview/reports/{reportId}/pdf`
- Automatically generated after each Test Run.
- Available for single Application Test Runs only. Comparison reports are not supported.

New Test PDF report

- Endpoint: `/publicApi/v8-preview/reports/application-test-runs/pdf`
- Not saved automatically. You must generate it via the Public API.
- Supports single and multiple Application Test Run IDs, allowing you to create comparison reports.

For more information, see Accessing the Public API.

Test PDF Report Breakdown

The Test PDF report summarizes the findings obtained from the testing process. It provides an overview of the Test specifics, results, and measurements. The report is divided into the following sections:

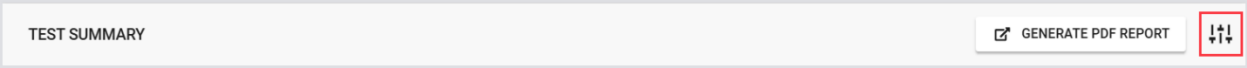
1. **Introduction**: Provides an overview of Application testing and the metrics used during this type of testing.
2. **Test Specifics**: Provides an overview of the key details of the test setup:
    1. **Product version**: The Virtual Appliance version.
    2. **Connector**: The specific Connector used for the Test.
    3. **Launcher group(s)**: The Group(s) of Launchers involved in the Test.
    4. **Test duration**: The total duration of the Test.
    5. **Date**: The date on which the Test was conducted.
    6. **Workload**: The type of Workload or Test scenario used.
3. **Test Summary**: Highlights key metrics for the Test:
    1. **Application failure**: Displays the number of failed Applications. For example: *script failed to run*.
    2. **Application performance**: Shows the number of Applications where the actual time exceeded the performance threshold.
4. **Platform Summary**: Provides insights into the platform's performance based on latency and login performance:
    1. **Login performance**: Indicates the performance during login operations.
    2. **Latency**: This shows any latency issues that occurred during the Test.
        1. **Login Performance** and **Latency** include the following:
            1. **Test name**: The name of the Test conducted.
            2. **Bar chart**: A visual representation of the performance data.
            3. **Actual time**: The time taken for the specific Test.
            4. **Threshold**: The predefined threshold, if applicable.
            5. **Execution**: The result of the Test execution.
            6. **Performance**: The performance analysis based on the Test outcome.
5. **Application Results:** Dives deeper into the results for each Application with the following subsections:
    1. **Summary**: A high-level result of the Application's performance during the Test.
    2. **Measurements**: Includes a table for each measurement or timer taken during the Test. Each table will include:
        1. **Test name**: The name of the Test.
        2. **Bar chart**: A visual representation of the performance data.
        3. **Actual time**: The actual time recorded for the measurement.
        4. **Threshold**: The threshold value (if applicable).
        5. **Status**: The status of the Test.
        6. **Screenshots**: This section includes screenshots taken on a particular application under test.

Tips and Tricks

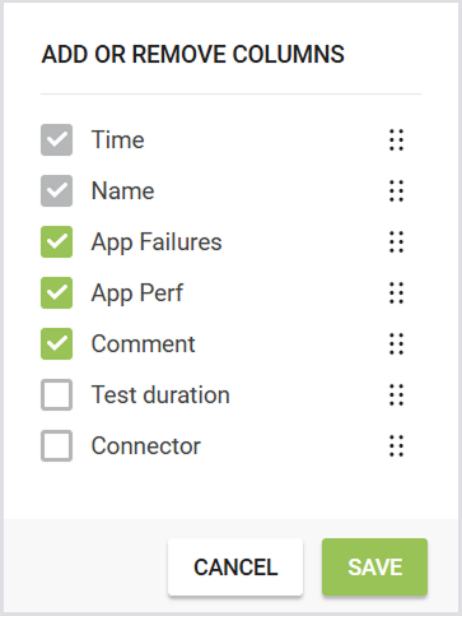
Customizing Table Columns

You can customize the Test Results table by adding, hiding, or sorting columns.

- **Configuration icon**: In the top-right corner of the **Application Test results** table, click the configuration icon to open the column management pop-up.



- **Adding columns**: Add additional columns to the table. The following columns are available:
    - App Failures
    - App Performance
    - Comment
    - Test duration
    - Connector
- **Hiding columns**: Hide columns you do not want to display using the configuration options.
- **Sorting columns**: To sort columns, drag and drop them into your preferred order.



NOTE: The Time and Test Name columns are fixed and cannot be hidden or moved.

Your configuration settings are saved in your local storage, i.e., your column preferences will persist when you navigate away from the page or log out and back in.

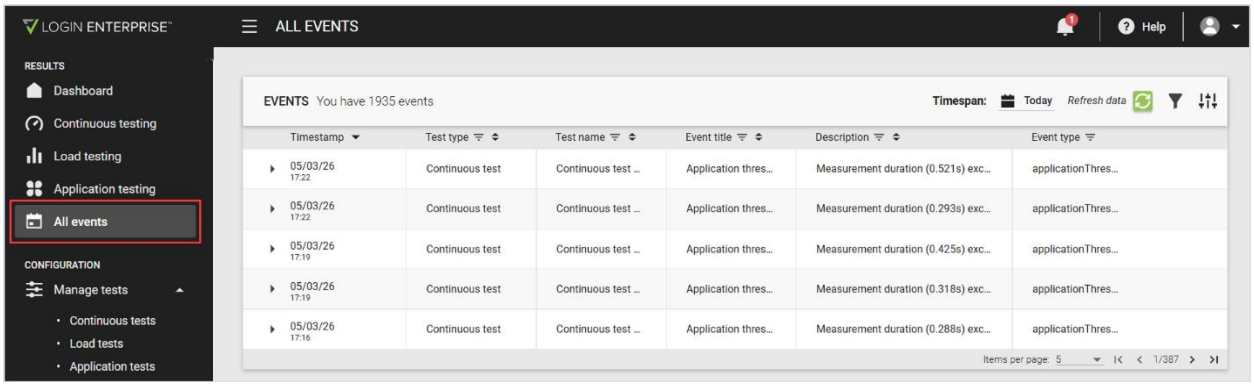
The configuration may reset if you use a different browser or clear your browser history.

Viewing All Events

- Overview
- Filtering events
    - Filter sidebar
    - Using column header filters
- Configuring columns

Overview

The **All Events** page allows you to view and filter all events in the system, including all Test types and System events.



Filtering Events

You can use the **filter sidebar** or **column headers** to refine which events appear on the page.

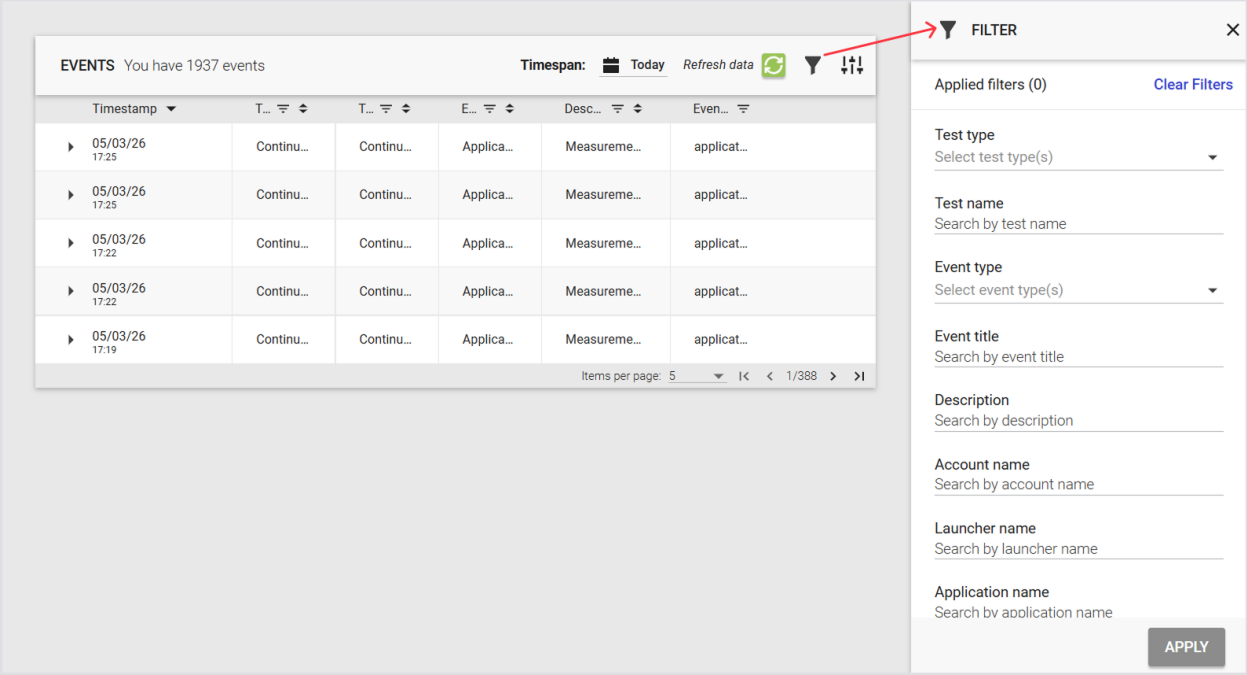
Filter Sidebar

The sidebar lets you combine multiple filters. The following options are available:

Filter	Description
Test type	Select one or more test types from a list of options.
Test name	Search for a test by name.
Event type	Select one or more event types from a list of options.
Event title	Search for events by title.
Description	Search event descriptions.
Account name	Search by account name.
Launcher name	Search by launcher name.
Application name	Search by application name.
Target host	Search by target host name.

Search fields filter results by any text that contains the search term. For example, typing `test` displays all items with “test” in their name or description.

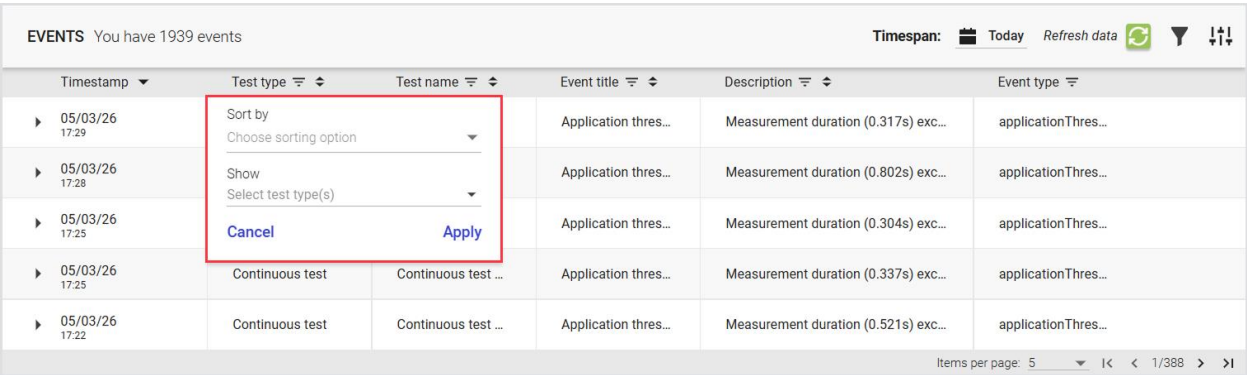
List-of-options filters let you select one or more checkboxes to include only those items. For example, selecting specific test types will show only events that match those types.



Column Header Filters

Click a column header to open a small menu that lets you:

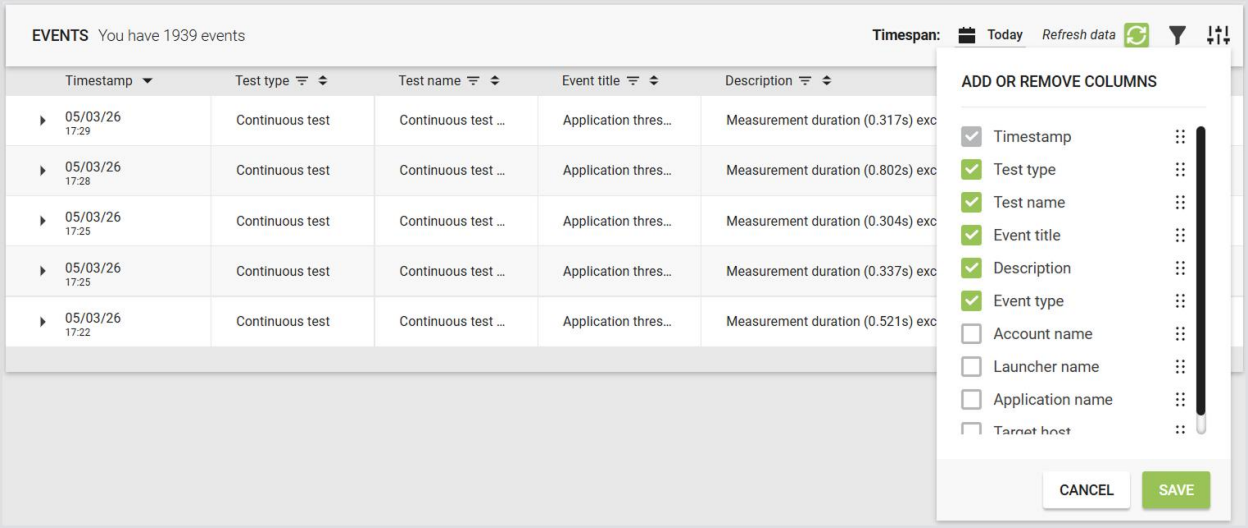
- **Set a sorting option**, such as ascending or descending order.
- **Apply quick filters** specific to that column, such as selecting a test type.



Configuring Columns

You can customize which columns are displayed and how they're arranged.

- You can show or hide any column except **Date**. The Date column is fixed and always visible.
- You can reorder columns, but **Date** always appears first.



Performance Measurements

Login Enterprise collects a comprehensive range of measurements to support in-depth analysis and facilitate informed decision-making. These metrics include:

- **EUX scoring:** Evaluate end-user experience effectively.
- **Application performance:** Monitor the time it takes for applications to load and perform Tasks.
- **Launcher Latency:** Measure the round-trip latency time between the Launcher and the target.
- **Logon metrics:** Track the duration of various steps within the Windows user logon and loading processes.
- **Session Metrics:** Gather data through PerfMon and WMI to assess system performance during user sessions.
- **Platform Metrics:** Collect data at the hypervisor layer to ensure overall stability and efficiency of the virtual environment.

This subchapter covers the following topics:

- Calculating the EUX score and VSImax
- Monitoring Login Performance
- Viewing Session Metrics

By leveraging these diverse measurements, Login Enterprise provides valuable insights that help optimize your IT infrastructure and enhance user satisfaction.

Calculating the EUX score and VSImax

- Overview
- Why calculate the EUX score?
- Terms and related concepts
- How Login Enterprise gets data for EUX score
  - Determining EUX score
- EUX score
  - Calculating EUX score
  - Baseline and Steady state EUX scores
  - Representative score check
  - Changing EUX work folders
    - Why change the EUX work folder
    - How to change the EUX folder
- VSImax
  - Calculating VSImax

Overview

The EUX score (End User Experience Score) is a rating that represents the performance of a Windows machine (virtual, physical, cloud, or on-premises) on a scale of 0 to 10 as experienced by the Virtual User. The EUX score can be calculated with any number of users (1 or more).

As you add more users to your VDI platform, expect your EUX Score to drop. As more users demand a greater share of a VDI system's shared resources, performance and user experience decrease.

The EUX score was introduced in Login Enterprise version 4.11 and is updated to reflect the evolving hardware and software platforms that VDI infrastructure runs on.

Why calculate the EUX score?

Calculating an EUX (End User Experience) score can provide several benefits in various contexts, helping ensure that the products, systems, and services you provide deliver a positive and satisfying user experience. Here are some reasons why calculating an EUX score might be valuable:

Benefit	Description
Quantitative assessment	An EUX score allows you to quantify the overall user experience of a product, system, or service. This can provide a clear and measurable indication of how well it meets the needs and expectations of its users.

Benchmarking	By establishing a baseline EUX score, you can track changes over time or compare different versions or iterations of a product. This enables you to assess the impact of design changes or improvements on the user experience.
Identifying pain points	An EUX score can highlight specific areas of the user experience that need attention or improvement. By identifying pain points or areas of dissatisfaction, you can prioritize efforts to address them and enhance the overall user experience.
Informing design decisions	Quantitative data from EUX scores can provide valuable insights for making informed design decisions. It can help designers and developers understand which aspects of the user experience are most important to users and where to focus their efforts for maximum impact.
Demonstrating value	For businesses and organizations, an EUX score can serve as a metric for demonstrating the value of investing in user experience improvements. A positive EUX score can indicate that the product or service is meeting user needs effectively, which can contribute to customer satisfaction, loyalty, and ultimately, business success.

Terms and related concepts

- **EUX score**: End User Experience score, i.e., a specific set of tools that produce measurements meant to closely approximate how a human would feel about using this machine. In other words, a score from 0 to 10. For more details, see the EUX score.
- **EUX App**: An application or tool designed to measure or optimize the respective aspects of the system's performance. For more details, see How Login Enterprise gets data for EUX score.
- **EUX timer**: EUX applications collect a bunch of data, which are referred to as "timer results". Each piece of data has a corresponding "timer" associated with it. The "timer" or the "action" is an EUX application with specific parameters. For more details, see Determining the EUX score.
- **VSImax**: The maximum number of concurrent active VDI sessions your hardware will support without compromising responsiveness. For more details, see the VSImax.

How Login Enterprise gets data for EUX score

The EUX score is based on the data derived from the performance of the EUX apps on the machine. EUX apps are small applications or tools designed to measure or optimize the respective aspects of the system's performance. To calculate the EUX score, Login Enterprise uses 4 EUX apps:

1. **Appspeed**: This application is like a specialized tool on your computer that tracks its performance. When you launch it, it sets itself up, prepares its window, and gets ready to interact with you. It keeps track of how quickly it's able to start up and be ready for your

input. Because the application itself is very small, we simulate the load by doing a bunch of CPU operations.

2. **Compressionspeed**: The purpose of this program is to test the performance of a multi-threaded file compression and decompression system. It measures how many blocks of data can be compressed and decompressed within a specified duration.
3. **CPUspeed**: The purpose of this program is to measure the performance of a CPU by running a set of computations in multiple threads for a specified duration. It creates a specified number of threads and then initiates a CPU workload test on each of them. Then it counts how many iterations of the workload were performed within the given duration.
4. **Diskspeed**: The purpose of this application is to measure the performance of disk I/O operations by running a set of read/write operations in multiple threads for a specified duration. It creates a specified number of threads and then initiates a disk I/O workload test on each of them. The program measures the number of I/O operations per second (IOPS) and various latency metrics.

Determining EUX score

The EUX score comprises various timers representing common user operations that closely correlate with human experience on the system. These actions enable Login Enterprise to measure application responsiveness, keyboard input processing, CPU-intensive tasks, and storage I/O latency. In addition, the EUX score for Load Testing also incorporates application and session failures, resulting in a score reduction when these events occur. For more information on the 2023 EUX timers, see the following table:

EUX Timers	Description
My Documents I/O score	Perform disk read and write operations (mostly sequential) in the 'My Documents' folder with caching disabled. We measure IOPS and latency.
Local AppData I/O score	Perform disk read and write operations (mostly random) in the 'Local AppData' folder with caching disabled. We measure IOPS and latency.
CPU score	Perform a series of mixed CPU operations and see how many can be done in a fixed period.
Mixed CPU I/O score	Perform a mix of cached and not cached, compression, and decompression operations.
Generic Application score and User Input score	Start a proprietary purpose-built text editor application. This application performs a series of actions on startup that are similar to Microsoft Office, but shorter. Measure the time from start to ready for user input. Then type a sequence of characters and measure the number of characters per second.

EUX score

Calculating EUX score

The EUX score calculation is the same whether it's the minute-by-minute score or the total EUX score. It is done in three steps. For the minute-by-minute calculation, we only take the results in the 1-minute "bucket"; for the total EUX, we take into account the results from the whole Test.

- Multiply every timer score by a hard-coded weight parameter and sum all of those results together
- Sum all of the actual weight parameters together
- Divide the weighted score sum from the sum of all weights.

In the end, the EUX score is the weighted average of all the EUX timer scores. A weighted average is an average where each value has a specific weight or frequency assigned to it. This means that some values contribute more to the average than others.

We also do not calculate the EUX score if any required timers (with a weight higher than zero) are missing. This usually only affects the beginning of the test, and even one value per timer is enough. We do not have any checks to ensure the amount of values that should be available.

Baseline and Steady state EUX scores

The Baseline and Steady state EUX scores provide additional dimensions to the experience your Virtual Users are having. The standard EUX score provides a single score for the duration of the entire Test, including the login period and the Application interaction period during the test run. As more users are steadily added to the system being tested, naturally the system will work harder and start to impact the user experience. The Steady state and Baseline EUX scores show us what the user experience is like during specific periods of the Test run.

- **Baseline EUX Score**: represents the best possible performance of the system and is the average EUX score of the best 5 minutes of the Test. This score indicates how the system performs when it is not under stress. Typically, the Baseline score is captured at the beginning of the Test before the system is fully loaded.
- **Steady State EUX Score**: Represents the time after all users have logged in (login storm) and the system starts to normalize. The Steady State EUX score is the average of the EUX scores captured 5 minutes after all sessions are logged in, until the end of the Test.
- **Representative Score Check**: The total score is checked to see if it is representative. This is determined by checking if all timers have at least 10 samples. If the total score is representative, it is used as the final score. Otherwise, the final score is set to 0.
- **Application Failures**: Currently, the application failures, such as the workload scripts or EUX application (timer) failures, do not have an impact on the EUX score. Workload application failures had an impact on the EUX2021 implementation, and it was a percentage-based penalty; an 80% success rate was translated to a 20% reduction in the EUX score.

Changing EUX work folders

Why change the EUX work folder

To ensure optimal performance and accuracy of EUX calculations, it may be necessary to configure the location of the EUX work folder. This helps avoid potential issues associated with using folders that are mapped to network drives or other locations unsuitable for EUX I/O operations. By selecting an appropriate location, sensitive directories are avoided, resulting in more realistic EUX and Max scores. You can customize the properties of the EUX work folder when updating a test configuration for both the Load and Continuous tests. You can update 3 EUX folders through the Public API.

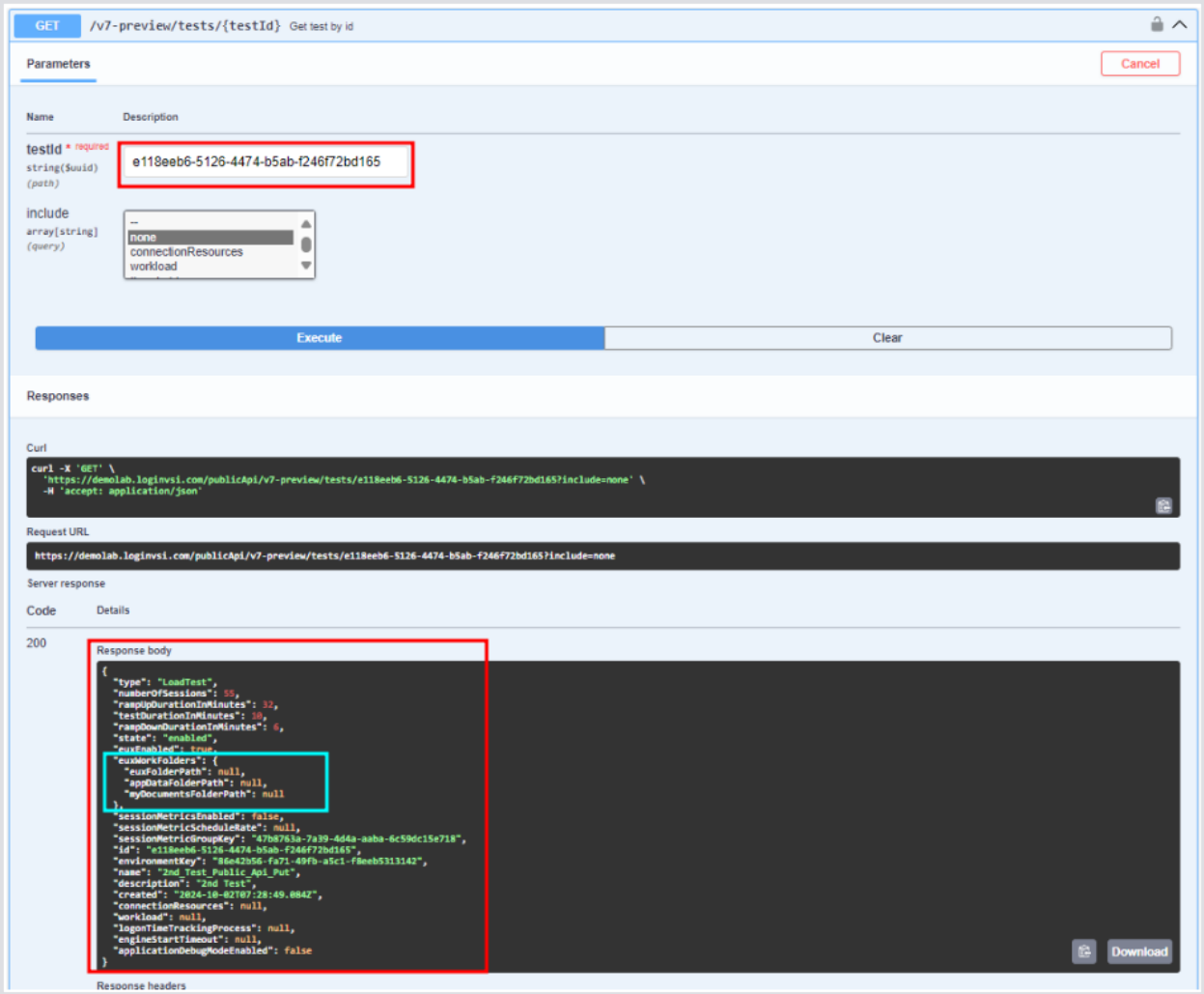
NOTE: The following folders are default:

- **EUX Folder location**: "euxFolderPath": "%USERPROFILE%\AppData\Local\Temp"
- **EUX Appdata location**: "appDataFolderPath": "%USERPROFILE%\AppData\Local\eux2023"
- **My Documents location**: "myDocumentsFolderPath": "%USERPROFILE%\Documents\eux2023"

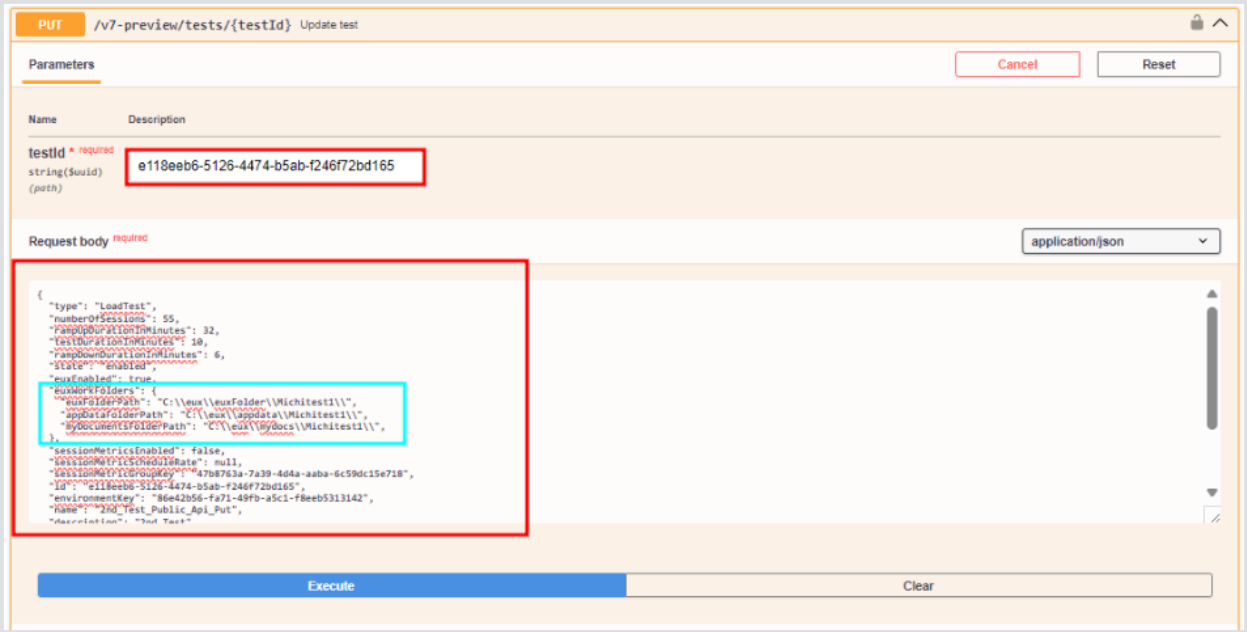
How to change the EUX folder

To change the EUX folder to a custom EUX folder in the Public API, you need to use a corresponding endpoint. To do this:

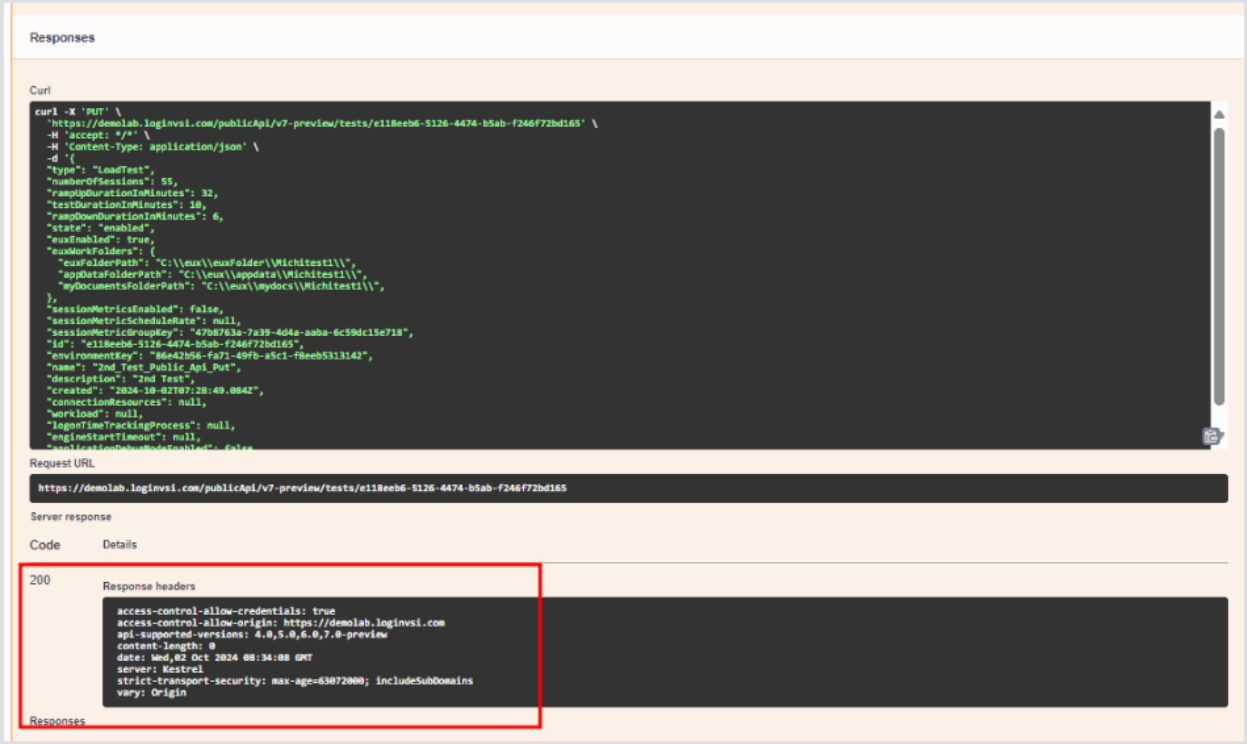
1. Open the **Swagger UI console** with the **v7 API**. To learn how to access the API in Login Enterprise, see Accessing the Public API.
2. Find **Test**, and click **GET /v7/tests{testId} Get test by id**.
3. Perform a GET operation on the Test to get the current configuration of the Test. Make sure to leave “Include” set to "none”.



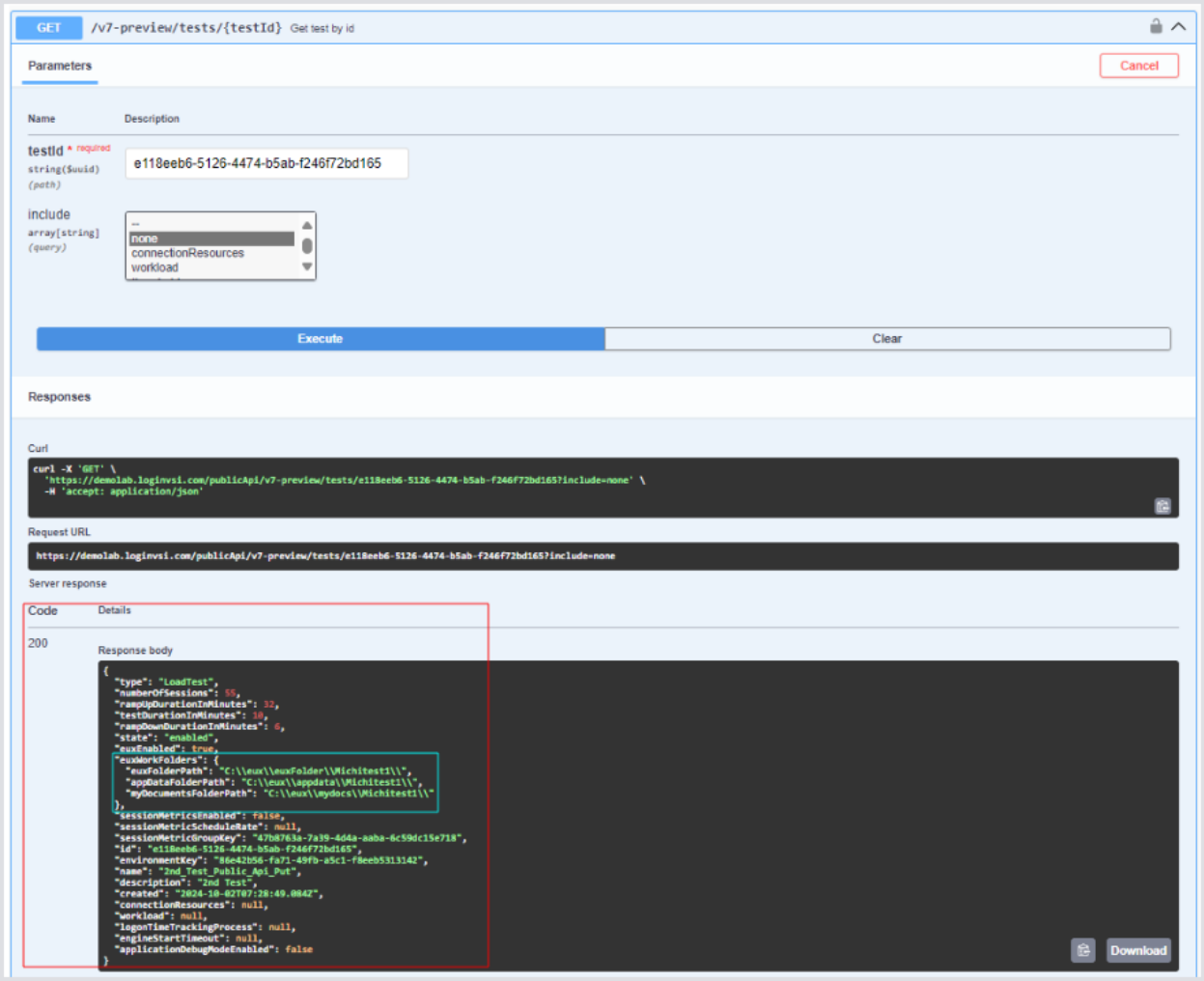
4. Copy the entire body from the GET output into your PUT request and modify your desired values.



5. Click **Execute**. The response status code should show **200**:



6. Finally, go back to GET and verify the changes:



IMPORTANT: When using custom EUX folders, it's essential to consider several cautions to ensure the security, integrity, and efficiency of the process:

- During creation, EUX work folders cannot be set; any properties configured will default to null.
- For successful operation, custom EUX work folders must grant full access to Virtual User Accounts. By default, the first session creates the EUX folder and files, restricting access to subsequent sessions, and causing EUX errors. Access permissions should be carefully managed to avoid these issues.

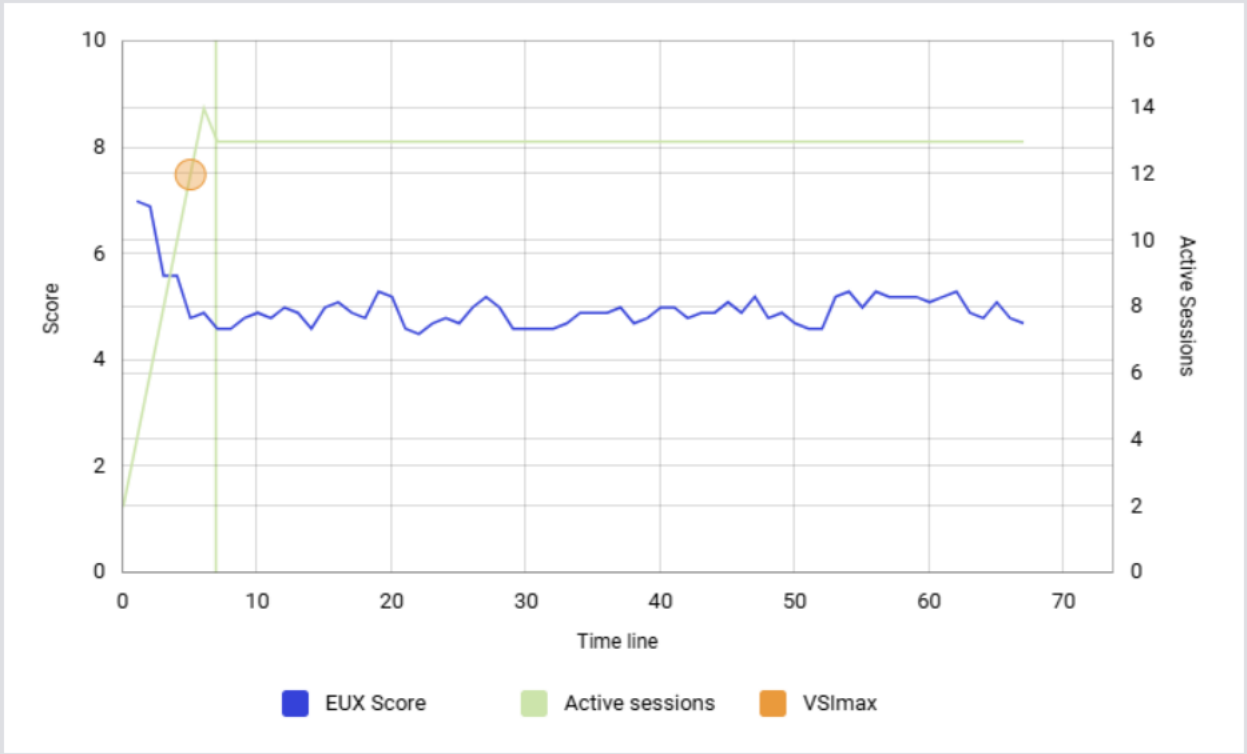
VSImax

VSImax defines the maximum number of concurrent users that can operate with an acceptable EUX score. If a system exceeds the number of sessions reported in the VSImax score, the End User Experience is expected to progressively degrade.

Calculating VSImax

VSImax is triggered when the EUX score dips below a certain value. By default, this value is defined to be an EUX score of 5.5 VSImax will be the last number of sessions currently logged in before the EUX average dropped below 5.5.

For instance, in this chart, EUX dropped below 5.5 after 12 sessions were logged in, so VSImax was set to 12:



There are two possible criteria for triggering VSImax:

- An absolute value that indicates at which point the experience starts to degrade (the default).
- A relative value based on the performance of the system when it is running optimally.

By default, we use the “absolute value” option, set to 5.5. However, it’s possible to configure Login Enterprise to choose VSImax based on a relative drop instead. For instance, we can trigger VSImax when the average EUX score drops to 85% of the baseline. For more information, see Baseline and Steady State EUX scores.

It is not normally necessary to change how VSImax is calculated. However, if your VSImax scores do not reflect your users' experiences, or if your EUX scores are consistently above or below 5.5, it’s possible to change the trigger threshold.

Monitoring Login Performance

- Overview
- Key metrics for Login performance
- Login times
  - Why measure login time?
  - Common definitions
  - Login steps calculations
- Measuring total login time
  - How it works with Login Enterprise
  - Considerations for Published Apps
  - Custom process of measuring total login time
    - In Login Enterprise UI
    - In the API
- Continuous Testing scenario
- Load Testing scenario

Overview

Login performance is a critical metric in evaluating any system's efficiency and user experience. Whether conducting Continuous or Load Tests, monitoring how quickly users can access their sessions and applications is essential for optimizing system performance and user satisfaction.

Continuous Testing provides detailed session-level insights, allowing for precise analysis of metrics like total login time, display protocol connection, profile load, and synchronous GPO processing impact. This approach enables proactive identification and resolution of performance issues within individual user sessions.

In contrast, Load Testing offers a broader perspective by aggregating performance data across multiple sessions, focusing on overall average metrics to assess system scalability and robustness under peak loads. This method provides crucial insights into system performance trends and identifies potential bottlenecks that may affect user experience during periods of high demand.

This section outlines the key metrics used to assess login efficiency and explains how these metrics vary between Continuous and Load Test scenarios. By understanding these differences, teams can better interpret and improve login performance across various operational conditions.

Key metrics for Login performance

Both Continuous and Load Tests utilize the following key metrics to measure login performance:

- **Display Protocol Connection**: Measures the time from when the launcher starts the session to when profile loading begins. This metric is found in Windows Event Logs.
- **Profile Load**: Measured using User Profile events from the Windows Event Viewer.

- **GPO Processing**: Only measures synchronous group policies. Asynchronous ones have minimal impact on user experience and are therefore not measured.
- **Total Login Time**: Measures how quickly a user account can log in, load the user profile, and become fully interactive. The time is calculated from the session start at the launcher to the start of the user's shell, e.g., explorer.exe. For more detailed information, see Measuring total login time.

For GPO processing, start and end times are pulled from the registry using the following key:

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Group
Policy\State\S-1-5-21-1642230215-40016302-1263235327-14012\Extension-
List\{00000000-0000-0000-0000-000000000000}
```

The SID value (S-1-5-21-1642230215-40016302-1263235327-14012) varies per session and is used to retrieve the following four values from the registry:

- StartTimeLo
- StartTimeHi
- EndTimeLo
- EndTimeHi

Combining these values provides the start and end times for group policies. Data is discarded if the GPO measurement start timestamp is before the session start time.

Login times

Why measure login time?

Benefit	Description
Improved User Experience	Swift login times play a vital role in creating a positive user experience. In the competitive world of applications and virtual environments, a rapid and responsive login process forms the foundation for engaging user interactions.
Performance Optimization	Measuring login time provides valuable insights into system efficiency. Identifying bottlenecks or delays enables you to make informed decisions for optimizing performance, ultimately speeding up users' access to their resources.
Scalability Planning	Understanding login times is critical for scalability planning. As your user base expands, it becomes essential to maintain efficient login processes. Accurate measurement helps anticipate and address potential challenges that may arise due to rising demand.
Availability Assurance	The login process typically marks users' first interaction with your system. Monitoring and optimizing login times contribute to overall system availability, mitigating the risk of user frustration due to delays or interruptions.

Common definitions

- **UserSession.LoginStartTime**:
  - This is calculated in the loginenterprise container when the Launcher sends the **UserSessionEvents.ConnectionStarted** event right before starting the connector process.
- **UserSession.LoginEndTime**:
  - This is calculated in the loginenterprise container when the Engine registers itself as started.
  - This happens right at the beginning while bootstrapping. After discovering the session, and before reading the login times.
  - ***This has nothing to do with the calculation of the login steps***.
- **ProfileLoadStart and ProfileLoadEnd**:
  - This is calculated by the Engine querying the WindowsEvents:

    "Event[EventData[Data[@Name='Session'] and (Data='{sessionId}')]]/System[Provider[@Name='Microsoft-Windows-User Profiles Service'] and (EventID={ProfileLoadStartEventId} or EventID={ProfileLoadEndEventId}) and Security[@UserID='{WindowsIdentityService.UserSid.Value}']]"

  - ProfileLoadStartEventId = 1
  - ProfileLoadEndEventId = 2
  - The Engine looks at the **TimeCreated** property of the events
- **GroupPolicyStart and GroupPolicyEnd**:
  - The Engine reads the following registry key:

    "SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\State\{WindowsIdentityService.WindowsIdentityService.UserSid.Value}\Extension-List\{{00000000-0000-0000-0000-000000000000}}"

  - *GroupPolicyStart* is calculated from the *StartTimeLo* and *StartTimeHi* properties of the registry key.
  - *GroupPolicyEnd* is calculated from the *EndTimeLo* and *EndTimeHi* properties of the registry key.
- **LogonTimeTrackingProcessStart**:
  - The Engine fetches the running processes on the machine looking for the *LogonTimeTrackingProcess* (different based on the platform and configured in the test). For more information, see How measuring total login time works.

    LogonTimeTrackingProcessStart = process.StartTime

Login steps calculations

- **connection** (session initialization)
  - **ProfileLoadStart - UserSession.LoginStartTime**

- **user_profile**
  - **ProfileLoadEnd - ProfileLoadStart**
- **group_policies**
  - **GroupPolicyEnd - GroupPolicyStart**
- **total_login_time**:
  - **LogonTimeTrackingProcessStart - UserSession.LoginStartTime**

Measuring total login time

To achieve peak performance for your application and virtual desktop environments, it's crucial to measure total login time. Login Enterprise assesses not only the application functionality but also the efficiency of user interactions, including the login process. The login time, measuring the duration it takes for a user to access an application or virtual desktop, is a significant performance metric influencing user satisfaction and overall system efficiency.

How it works

When a session initiates, the Engine looks for a process named 'explorer.exe'. The Engine captures the start time of this process and communicates this data to the appliance. This observation forms the foundation for the appliance to determine the total login time—the duration spanning from the moment a session is started by the launcher until the initiation of the 'explorer.exe' process. The total login time metric is then available from one of the charts in the test results.

Currently, the behavior for deciding what process to use to measure total login time is as follows:

- **Citrix Storefront and NetScaler Connector (published desktop/app)** = wfshell or picashell
- **All other connector types** = explorer

Considerations for Published Apps

In scenarios involving published apps, the presence of 'explorer.exe' in the Engine's session may not apply, as these apps operate differently. The appliance may face challenges locating 'explorer.exe' during the login process for published apps, potentially impacting the accuracy of total login time measurements.

Custom process of measuring total login time

To address the challenges posed by published apps, Login Enterprise allows you to overwrite the default process with a process of your choice. This flexibility ensures that, regardless of the intricacies of your environment, you can tailor the login time measurement to align precisely with your specific configurations and requirements.

You can customize the process of measuring total login time in one of the following ways:

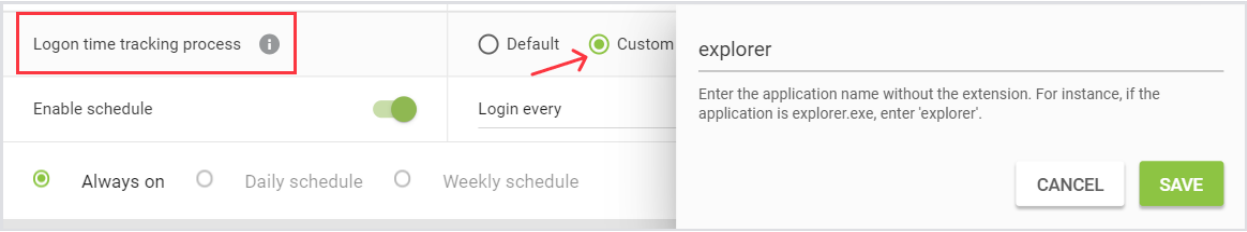
- Login Enterprise UI
- Login Enterprise API

NOTE: The process of measuring total login time is backward compatible. So, if you have already changed the process through the internal API, you don't have to change the settings again. Regardless of whether the property was named 'shell-executable' or 'shellExecutable' during the new process setup, both names will be displayed in the new field after the upgrade to 5.7.

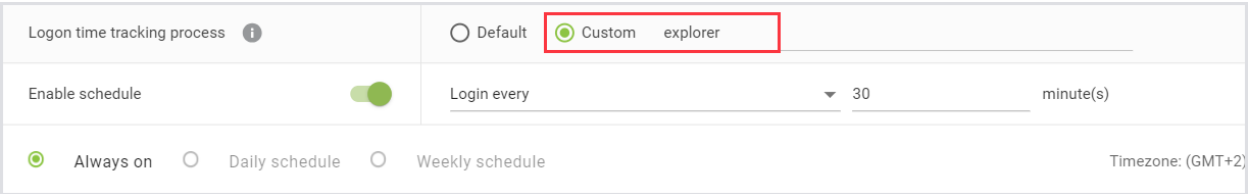
In Login Enterprise UI

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Manage tests**, and select the desired test (The process of customizing the measurement of total login time is the same for all test types).
2. Click on the test you're interested in, and in Test settings, find **Logon time tracking process**.
3. In the **Logon time tracking process**, select **Custom**. (Note that the 'Default' radio button is selected by default).

In the pop-up window, enter the value for the new process, and save the changes. The value must be without extension. For example, for explorer.exe, the value is 'explorer'.The default values for Citrix StoreFront and NetScaler Connector are 'wfshell' or 'picashell.' For other connector types, the default value is 'explorer'. You can customize it based on your platform, such as 'wfshell' for Citrix, 'vmware-view-usbd' for VMware, or 'rdpshell' for Microsoft RDP.



Cancelling the popup selects the 'Default' radio button again. Saving the popup value chooses the custom value, updating the field. If you'd like to make further changes, initiate the popup once more.



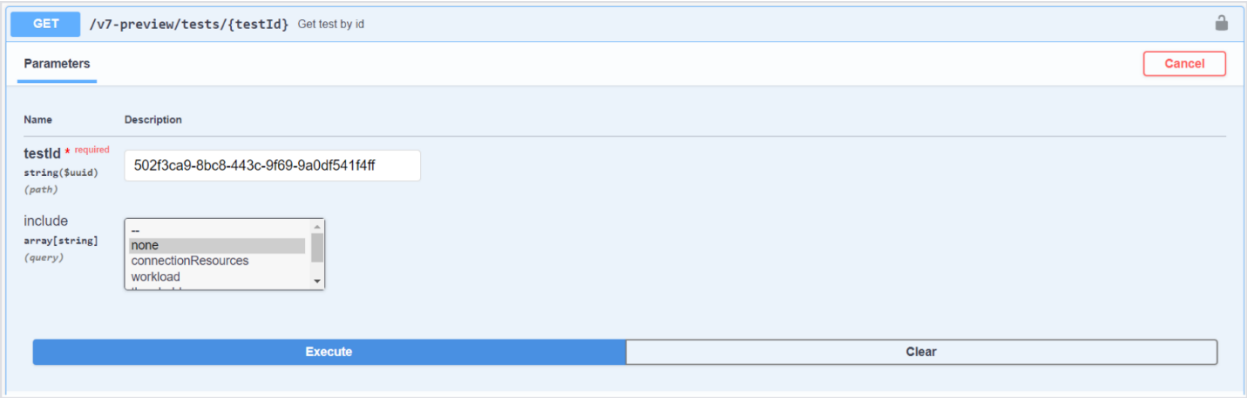
In the API

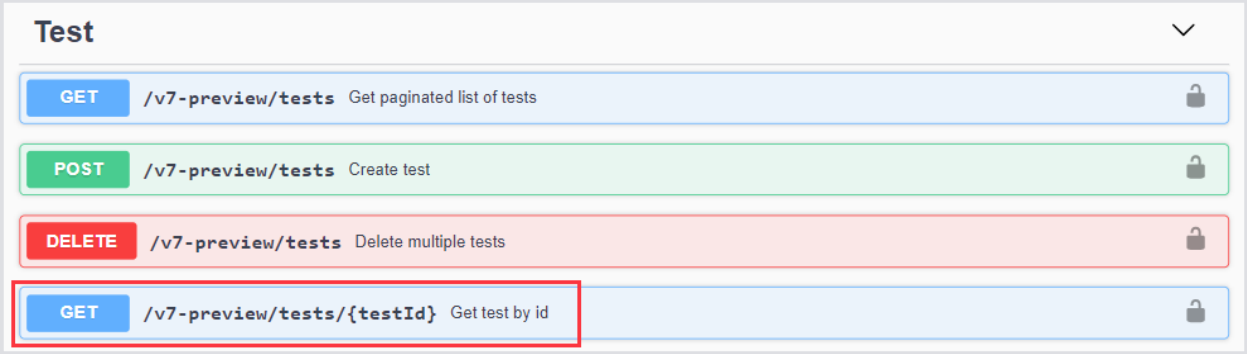
To learn how to access the Public API in the Login Enterprise, see Accessing the Public API.

Creating a GET request in the Public API

To ensure your test is configured correctly:

1. In the **Swagger UI console**, find **Test**.
2. In **Test**, click **GET /v7/tests/{testId}**.
3. In the top-right corner, click **Try it out**.
4. In **testId**, paste the ID of your test, and then **Execute**.





TIP: In the **Login Enterprise sidebar menu** > **Configuration** > **Manage tests >** (e.g., Continuous test) address bar, you will see the URL. Locate a part of the URL that contains the test Id. It often appears as a series of letters, numbers, or a combination of both. Highlight the entire Id in the URL, right-click, and select 'Copy'.

If the server responds with a 200 status code, indicating that the GET request was successful, it will return the requested information in JSON format in the response body. Copy the JSON text from the response body as you will need it when creating a PUT request.

Server response

Code	Details
200	Response body

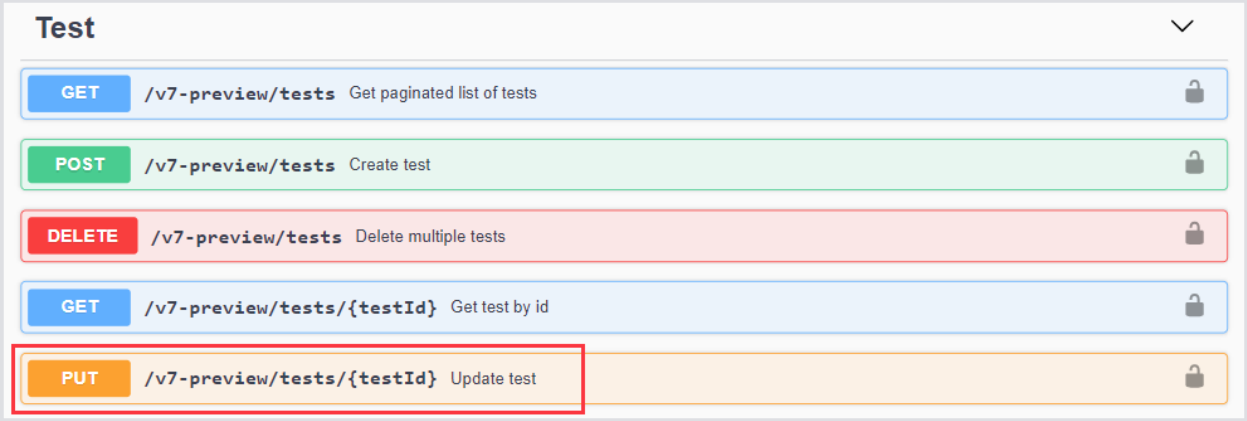

```
{
  "type": "ContinuousTest",
  "scheduleType": "concurrentSessions",
  "scheduleIntervalInMinutes": 10,
  "numberOfSessions": 1,
  "enableCustomScreenshots": false,
  "repeatCount": 2,
  "isRepeatEnabled": false,
  "isEnabled": false,
  "restartOnComplete": false,
  "euxEnabled": false,
  "sessionMetricsEnabled": false,
  "sessionMetricGroupKey": "9dec89c0-02fd-49f1-8736-04ccc7194638",
  "sessionMetricScheduleRate": null,
  "alertConfigurations": null,
  "id": "502f3ca9-8bc8-443c-9f69-9a0df541f4ff",
  "environmentKey": null,
  "name": "euxTestingSsd",
  "description": null,
  "created": "2023-06-30T23:34:28.388Z",
  "connectionResources": null,
  "workload": null,
  "logonTimeTrackingProcess": null,
  "applicationDebugModeEnabled": false
}
```


Download

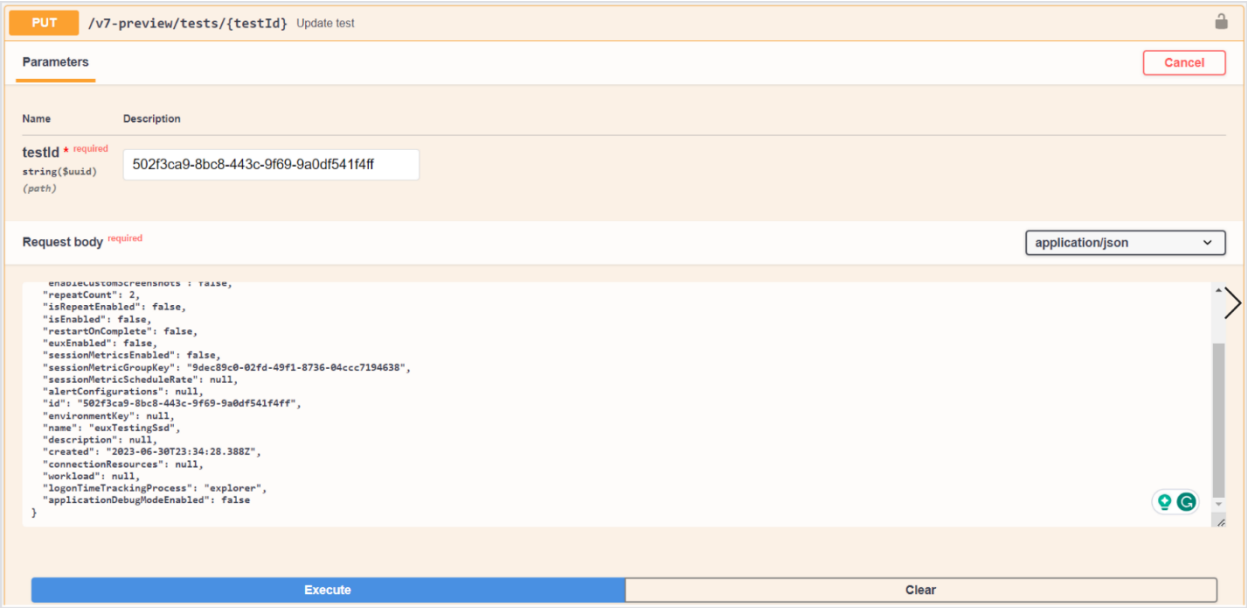
Creating a PUT request in the Public API

To customize the process of measuring total login time:

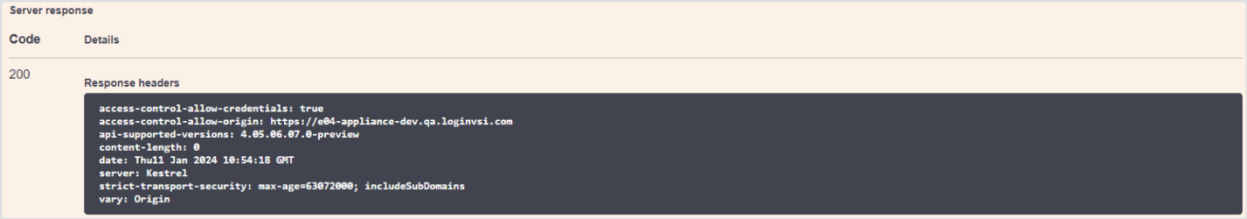
1. In the **Swagger UI console > Test**, click **PUT /v7/tests/{testId}**.
2. In the top-right corner, click **Try it out** (The Request body field will open).
3. In **testId**, paste the ID of your test.
4. In the **Request body**, paste the JSON text you copied from the response body in the GET request.
   In the "logonTimeTrackingProcess" property, specify the value for the new process. For example, if it's explorer.exe, enter "explorer".
5. Click **Execute**.



If the server responds with a 200 status code, indicating that the PUT request was successful, it will return the requested information in the response headers.



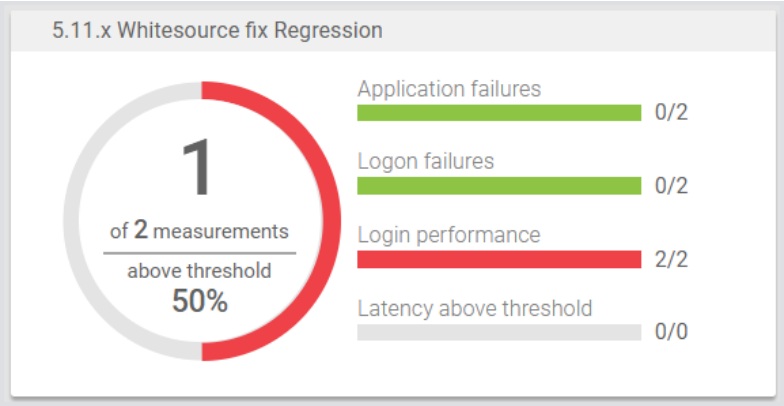
When you select the ' Custom ' radio button, the new process field will appear in the Test settings.



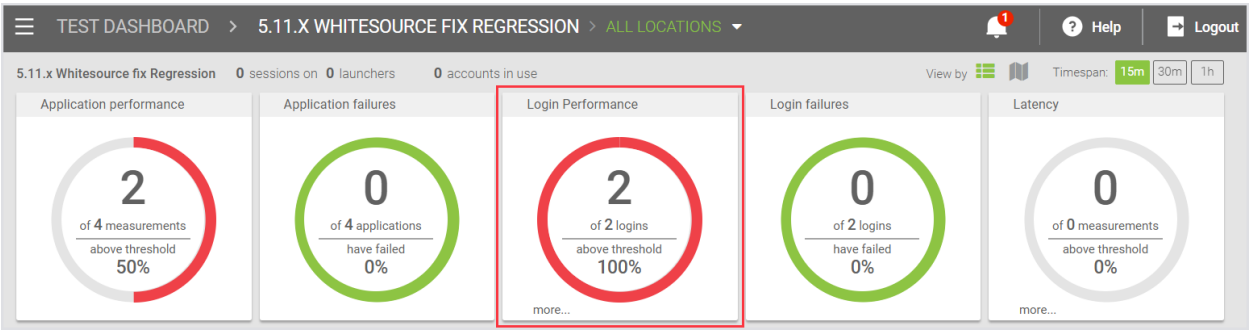
Overall, the Login Enterprise login time measurement feature provides a nuanced understanding of user experience, allowing you to make informed decisions for optimizing performance and enhancing the efficiency of your applications and virtual desktop environments.

Continuous Testing scenario

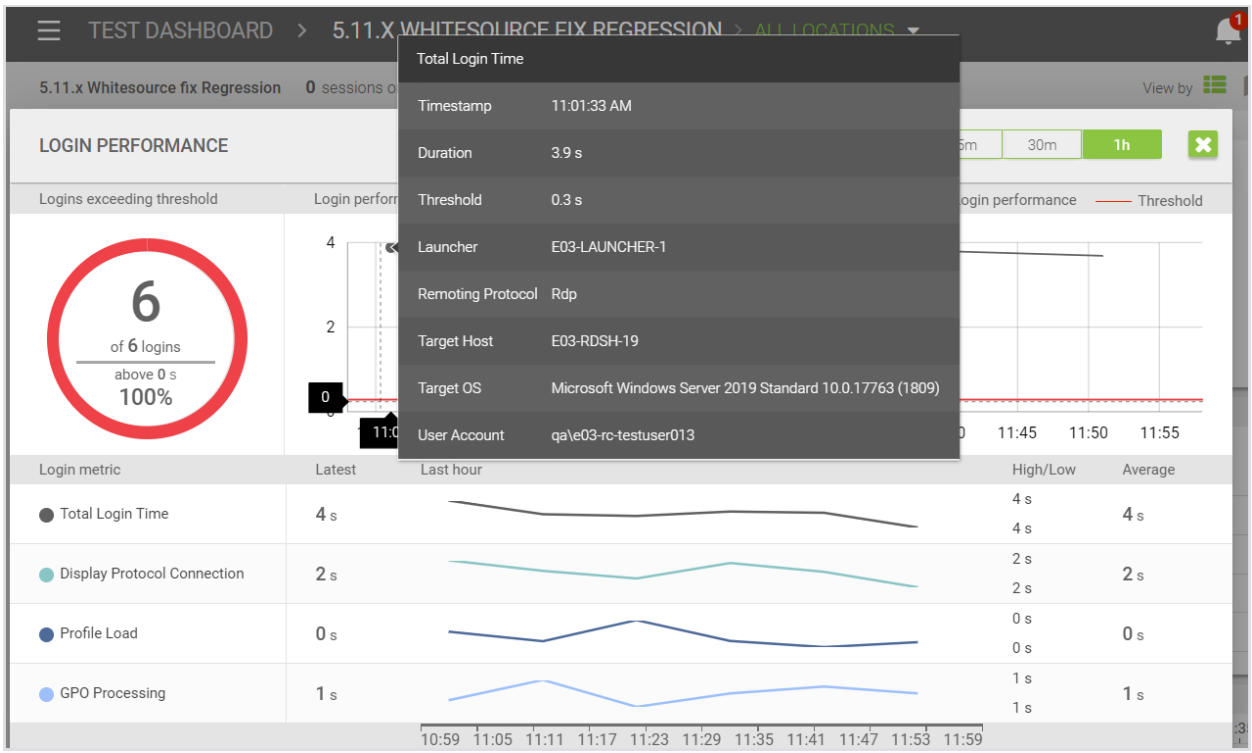
For Continuous Testing, Login performance is measured within each session. In the Test dashboard, each environment configured displays relevant metrics. For example, this this bar indicates when 1 out of 2 measurements have crossed the configured logon threshold.



Navigating deeper into the environment allows you to see detailed information about each metric. For this, click on the Login Performance title.

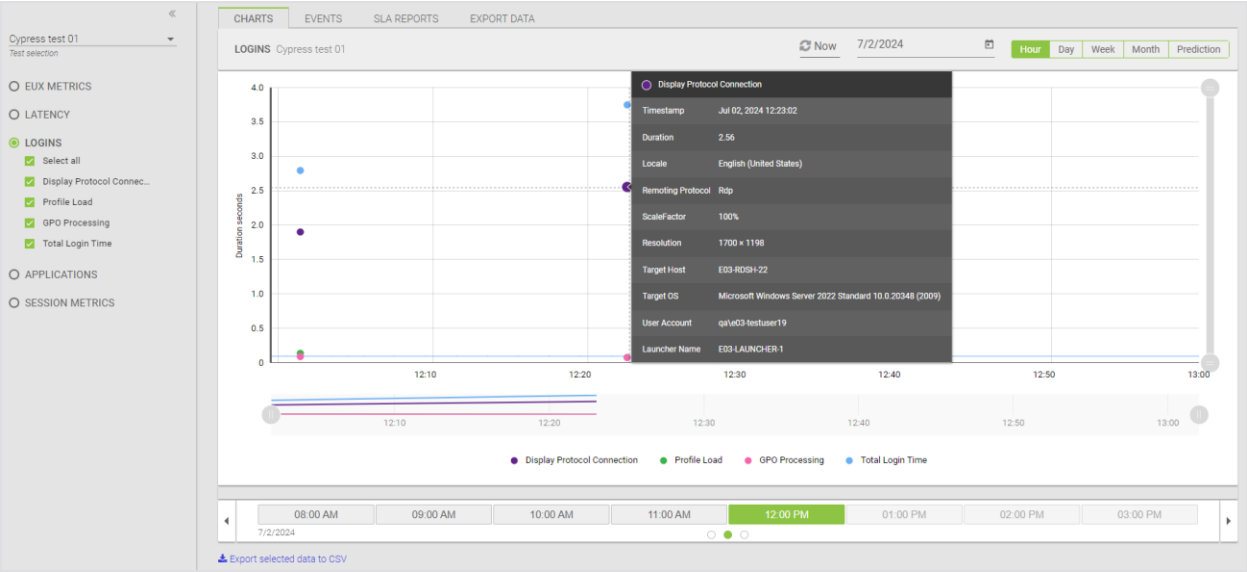


The following window reveals additional insights into measurements, including detailed information, such as timestamp, duration, launcher used, remoting protocol, target host and OS, and user account.

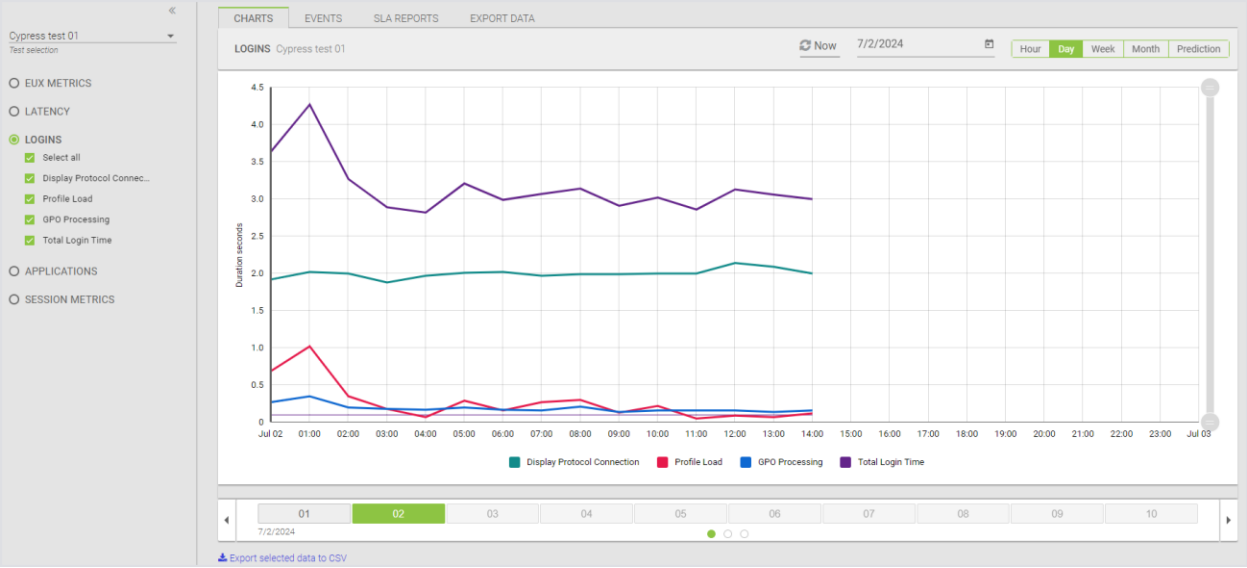


- **Timestamp**: Logging moment.
- **Duration**: Logon time at logging moment.
- **Threshold**: Defined threshold.
- **Launcher**: Launcher used during measurement.
- **Remoting Protocol**: Remoting protocol used.
- **Target Host**: VM hostname generating measurement.
- **Target OS**: OS version of target VM.
- **User Account**: Account used for measurement.

The Continuous testing charts allow you to view measurements, providing raw data points with detailed hover information at the hour level. To view the chart, in **Continuous testing > Charts > Logins**, select all the metrics for Logins (Display Protocol Connection, Profile Load, GPO Processing, Total Login Time).



When you change the view to Day, Week, or Month, the charts will show aggregated data. Specifically, the data is aggregated per hour, providing a summarized view for these longer time periods.



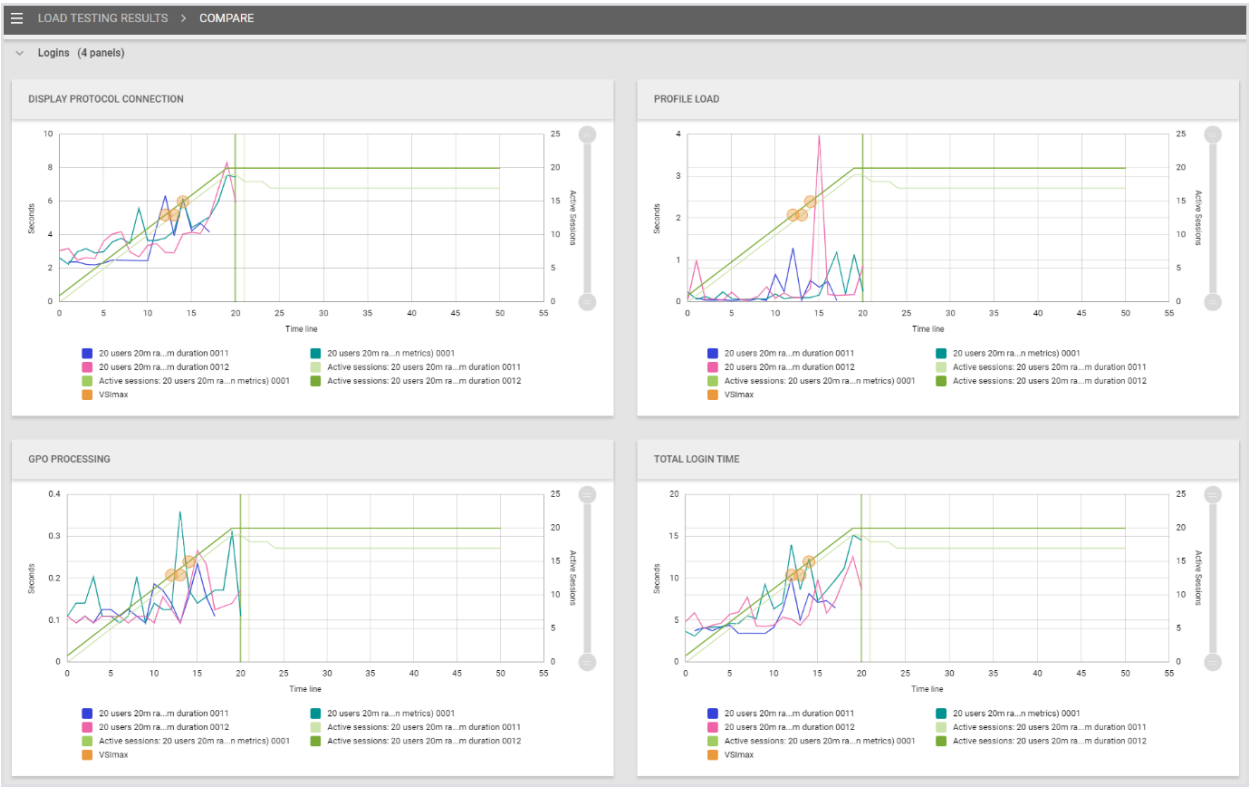
Load Testing scenario

For the Load testing, when comparing up to 5 tests, Logon performance data is represented as a table in **Load Testing results > Overview > Logon time**, showing the overall average for the entire test duration. This average is calculated across all sessions in the test.

LOGON TIME Average duration (in seconds)

Measurement	20 users 2... n metrics) 0001 - Finished	10 users 3... m duration 0002 - Finished	20 users 2... n metrics) 0002 - Finished	20 users ... on (WIN11) 0001 - Finished	20 users 2... m duration 0019 - Finished
Display Protocol Connection	4.3	(+40%) 2.6	(+2%) 4.2	(-293%) 16.9	(+5%) 4.1
Profile load	0.2	(-500%) 1.2	(0%) 0.2	(-3600%) 7.4	(-150%) 0.5
GPO Processing	0.2	(-550%) 1.3	(0%) 0.2	(-11050%) 22.3	(-50%) 0.3
Total login time	7.8	(+19%) 6.3	(-5%) 8.2	(-655%) 58.9	(+9%) 7.1

The Charts display aggregated data points per minute, showcasing average measurements across all sessions during the test period.



By leveraging insights from both Continuous and Load Test scenarios, organizations can adopt a comprehensive optimization strategy. This strategy involves continuous monitoring and iterative improvements based on session-level details from Continuous Testing and validation of overall performance stability from Load Testing. This approach ensures that login processes remain efficient, reliable, and responsive to the dynamic needs of users and business operations.

Viewing Session Metrics

- Overview
- Continuous Testing
  - Tips for Analysis
- Load Testing
  - Tips for Analysis
- Additional resources

Overview

Once you've defined your Session Metrics, created a Group, and added your Definitions, you can collect Session Metrics through your Continuous or Load Tests. Session Metrics appear as radio buttons in Continuous Testing Results and Load Testing Results charts.

IMPORTANT: Charts only display metrics with the same unit (%, Ops/Sec, Bytes, etc.) simultaneously. This unit is set when you structure a Session Metric definition for an API request.

Continuous Testing

Below is a chart showing performance counters from a Group linked to the Test. For any Test, once the Session Metrics radio button is selected, a list of all associated Metrics will appear.

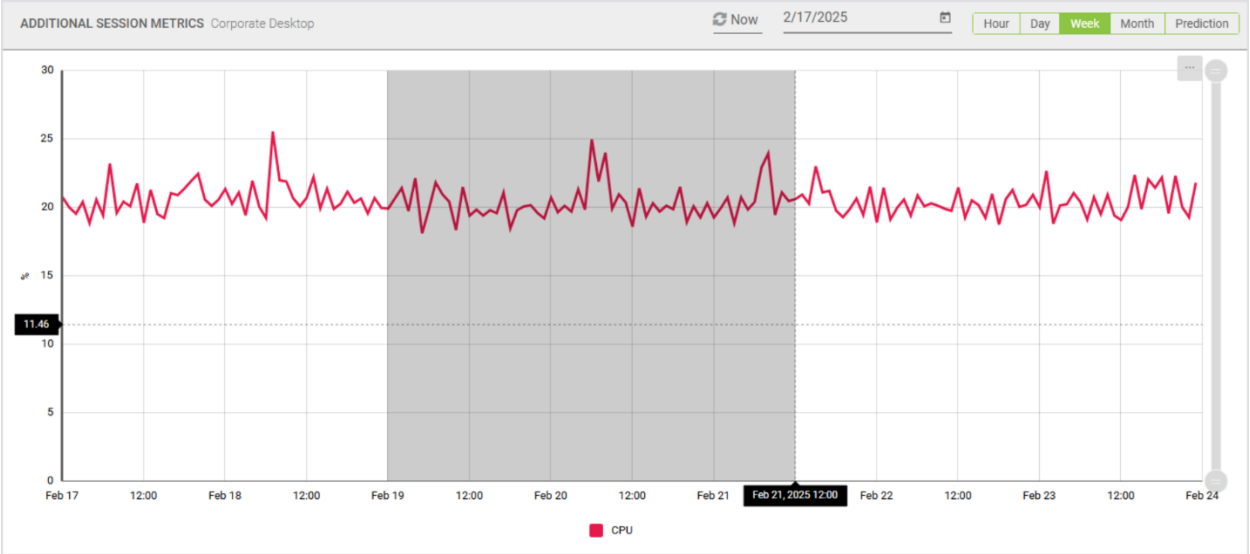
NOTE: Metrics that were unable to collect data (e.g., incorrect query definition, incorrect authorization) will still appear on this list. To verify that Session Metrics work as expected, it is recommended to create a Test without any Actions. Assign your Session Metrics Group to the Test and run it once. Use the dashboard to confirm which definitions are accurate and which Metrics may need to have their definition double-checked.

In a Continuous Test, Session Metrics are aggregated on charts as follows:

- **Hourly view:** by minute
- **Daily view:** by hour
- **Weekly view:** by hour
- **Monthly view:** by hour

Using wider time periods enables trend analysis of performance counter data while minimizing the impact of outliers from isolated incidents or anomalies in the environment. For analysis of environmental issues or anomalous behavior, use a shorter time period to visually highlight any deviation.

TIP: Drag your mouse across the chart to focus on the selected range.



Tips for Analysis

Every environment is different, and each organization has its own set of KPIs or thresholds used for observability, monitoring, and alerting. For analysis of particular scenarios or anomalies, consider a more targeted approach. Here are some general guidelines for analyzing your Continuous Testing Session Metrics:

- **Environments are different:** Consider the context of your environment when analyzing Session Metrics. Factors like the business industry, the nature of its end-users, and the infrastructure that they rely on can lead to different conclusions based on the Session Metrics you collect.
    - Is the test device a shared resource (e.g., Multi-Session or terminal server), or is it a dedicated device like VDI or a physical endpoint?
    - Are there major shift changes or user logon patterns that could impact performance?

    - Are there scheduled tasks, scans, or background processes that could influence performance at certain times?
- **Environments change:** Updates to applications or the OS, and other environmental changes, can impact performance. Use the context of your organization's environment to correlate changes in the Metrics you collect. CPU and Memory metrics (collected by default) are useful for understanding the impacts of major changes.
    - Have there been recent updates to the OS or applications?
    - Were there changes to the network, infrastructure, device, or its resource allocation?
    - Did security policies, antivirus configurations, or other system settings change?
- **Application behavior:** Certain workflows may involve resource-intensive operations, leading to increased system utilization. Identifying correlations between specific actions and performance metrics can provide valuable insights. You can track process-specific resource utilization with Session Metrics for your key Line of Business apps, AV solutions, or other security applications.
    - Are there application executions that consistently lead to higher resource consumption?
    - Are there deviations at specific steps, or is resource utilization sustained throughout the workflow?

TIP: If there are potential optimizations to configure, use a Performance Test to A/B test them.

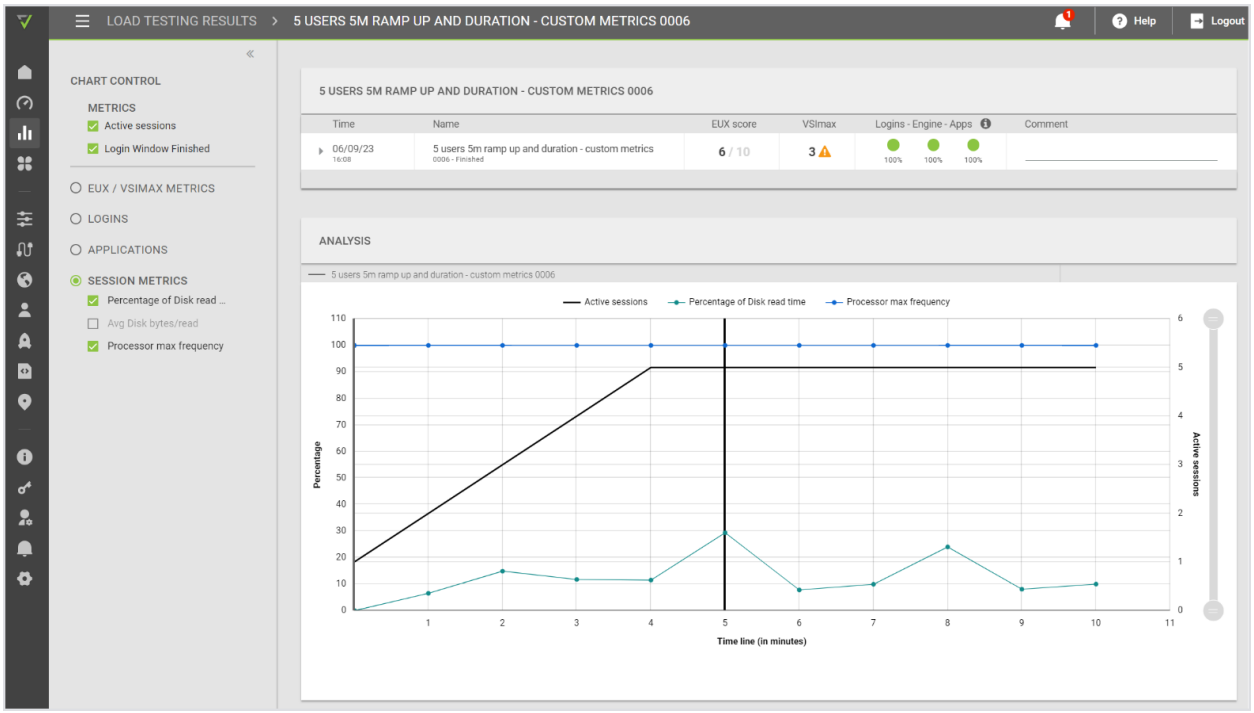
Load Testing

Unlike Continuous Testing, Session Metrics collected during Load Tests are meant to be analyzed in relation to the Test's active session count. Depending on the environment and the number of users configured for the Test, there may be different implications.

Login Enterprise supports both persistent and non-persistent virtual desktops across Client and Server OS, including Windows 11 Multi-Session.

For multi-user Load Tests, Login Enterprise can collect Session Metrics from a configurable number of sessions.

- In Multi-Session or Server OS environments, collecting metrics from a single session helps control data volume, as each user will return identical performance counters from the same OS.
- In Client OS environments, such as VDI or physical endpoints, each Session's Metrics reflect the unique performance of its respective VM or device.

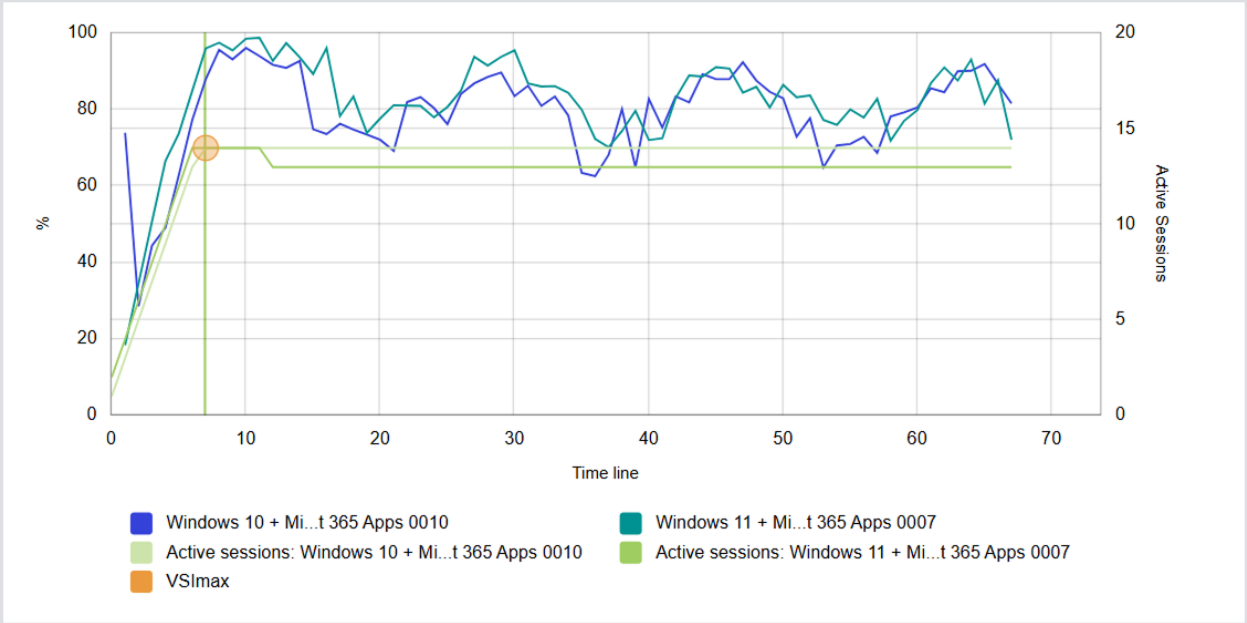


Tips for Analysis

Load Testing analysis is different from Continuous Testing. Many customers use Load Tests to performance-test their latest OS images. Unlike Continuous Tests, where you're monitoring and alerting, Load Tests are more focused on identifying issues and preventing them.

Here are some general guidelines for analyzing your Load Testing Session Metrics:

- **System behavior under load:** Systems behave differently under high user concurrency. Identifying how performance changes under different load conditions can reveal limitations and highlight bottlenecks.
    - Does the number of users correlate strongly with any Session Metrics being collected?
    - Is the system compute-bound, memory-constrained, or storage-limited? Is it network-dependent?
- **A/B Testing:** When large-scale changes are made (e.g., Gold Image updates, endpoint AV deployments), performance tests can highlight benefits and weaknesses through A/B Testing.
    - In a controlled comparison, are there major changes to resource consumption between changes?
    - Where are the changes likely to have the biggest impact: Compute? Memory? Storage? Network?



Additional resources

- For information on creating and configuring Session Metrics, see Managing Session Metrics.
- For details on viewing and analyzing Continuous Test results, see Viewing Continuous Testing Results.
- To learn how to view Load Testing results, see Viewing Load Testing Results.

Back to Dashboards and Reporting

Application Script Recording

This chapter provides an overview of Application script recording in the Login Enterprise Virtual Appliance.

It covers the creation of application scripts, offering insights into best practices and considerations for effective scripting. You will learn about the Script Editor and Script Recorder tools, which facilitate the development and refinement of your scripts. Additionally, the chapter introduces Application X-Ray, a valuable resource for analyzing application behavior, and discusses various scripting functions that enhance your automation capabilities.

The chapter covers the following topics:

- Creating an Application Script
- Difficult Applications
- Using the Script Editor
- Using the Script Recorder
  - WinApp Recorder
    - UI Inspector (Preview)
    - Understanding WinApp Recorder Implementation (FlaUI)
  - Web Recorder
- Using the Application X-Ray
- Scripting Functions Overview
  - Web Recorder Functions (Playwright)

Creating an Application Script

- Overview
- Step 1. Creating an Application priority list
- Step 2. Defining the workflow
- Step 3. Defining measurements
- Step 4. Defining requirements
- Step 5. Final review
- Additional resources

Overview

This article outlines how to prepare your Application scripts. Application scripting is straightforward, but getting the details right is essential for a smooth experience.

Begin by determining the order of Applications and the specific actions to be performed within each one. Additionally, consider the following factors before you start creating your Application scripts.

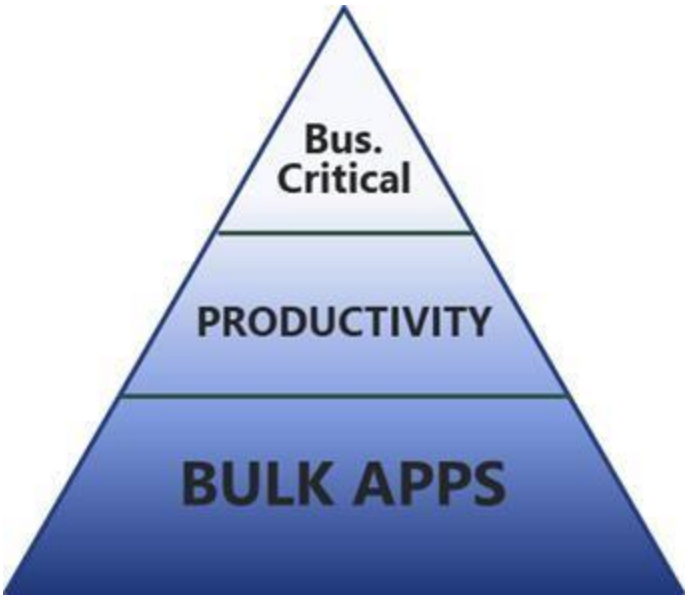
Scripting involves automating tasks within Applications, allowing for increased efficiency and consistency. By following best practices, you can ensure your scripts are effective and easy to maintain.

The section discusses the following subjects:

1. **Creating an Application Priority List**: Learn how to identify and rank Applications from most to least important.
2. **Defining the Workflow**: Understand how to outline a workflow for each Application and what it should entail.
3. **Defining Measurements**: Discover what metrics you need to track to gather accurate data.
4. **Defining Requirements**: Determine the needs of Virtual Users to successfully complete the defined steps for each Application.
5. **Final review**: Document everything thoroughly and conduct a manual verification to ensure accuracy.

Step 1. Creating an Application priority list

The first step is to create an Application priority list. You can classify Applications into 3 categories: **Business Critical**, **Productivity**, and **Bulk**. Prioritize the Applications in the same order as indicated in the image below.



Business Critical

Business-critical Applications are those that a significant portion of your user base relies on. If these Applications fail to function correctly, it could result in substantial financial losses for the business. As such, they hold the highest priority for development and should undergo thorough measurement to collect critical performance data.

In a typical Virtual Appliance environment, the number of Business Critical Applications usually ranges from 1 to 10. These Applications are often integral to use cases such as Continuous Testing, Load Testing, and Application Testing within the Virtual Appliance.

Productivity

Productivity Applications are widely used but pose minimal risk if they do not perform as intended or become unavailable. For instance, Adobe Reader can be categorized here; while a slowdown may be inconvenient, it is not catastrophic. This category is the second-highest priority for development. Focus may be less intensive, concentrating instead on ensuring that specific, frequently used actions function properly.

These Applications are primarily utilized in Continuous Testing and Load Testing use cases, depending on their impact. However, it is recommended that they always be included in the Application Testing use case.

Bulk (Rest)

Bulk Applications are those that do not fall under the Core or Business Critical categories. Their impact is minimal to nonexistent, making them non-critical for business operations. Examples of such Applications include Notepad, Paint, and Calculator. If these Applications are unavailable

for a day or two, it remains a manageable situation. However, it is still important to ensure they function at a basic level.

These Applications are primarily utilized in the Application Testing use cases, where the focus is on verifying that they can launch successfully after any changes have been made to the image.

How to gather this information

There are several ways to compile the lists mentioned above. The most common methods include:

1. **Management input**: Consult management to identify what they consider mission-critical. Understanding their needs and goals will help you generate the appropriate reports.
2. **Monitoring tools**: Utilize monitoring tools that can provide insights into the most frequently used Applications in your environment. This data can assist in classifying Applications as Core or Business Critical.
3. **Functional Testing teams**: Functional Testing teams are responsible for testing Applications before their release. They often maintain a prioritized list of Applications, which can be valuable in your assessment.
4. **Application owners/teams**: Application owners or teams are typically responsible for specific Applications. If an Application has a dedicated team, it likely indicates that it is a Core Application.
5. **Consulting Key-Users**: Key users from various departments can offer valuable insights into which Applications are critical for their roles.
6. **Support/Service desks**: Support or service desks can provide information on the most troublesome Applications based on the volume of support tickets received regarding those applications.

Step 2. Defining the workflow

Now that you have identified the Applications to measure, it is crucial to determine the actions you will perform, and potentially measure, within each Application. A combination of actions for an Application is referred to as a workflow or click path.

The objective of this step is to create a detailed workflow or click path that accurately mimics real user activity and load. It is essential to be as detailed as possible, as the individual creating this workflow may not be the one developing the Application script.

IMPORTANT: Steps should be repeatable. Modifying data within any Application workflow could disrupt the actions being performed.

To illustrate, refer to the example below, which demonstrates a less effective approach, followed by an example that illustrates a more effective method. Notice the difference in detail between the two.

Wrong way

1. Start Application x
2. Login
3. Search for DATAXXXX
4. Close Application

Right way

1. Start Application X
2. Wait for the login window to appear
    1. Type user name
    2. Type user password
    3. Select environment
    4. Click on Login
3. Wait for the main window to appear
4. Click on the search button at top of window
    1. Wait for the search window to appear
    2. Type DATAXXXX in the search field
    3. Enable Checkbox Filter
    4. Click on search
5. Wait for the search result to appear
6. Click on the top result called DATAXXXX
    1. Wait for the data set to load
    2. Browse through the dataset
    3. Close the dataset
7. Close the Application

The example above primarily applies to Core Applications. As you move lower on the priority list, the level of detail required for the actions decreases. For instance, workflows for Bulk Applications may be as simple as starting and stopping the Application.

How to gather this information

There are several methods to gather the workflows mentioned above. The most common approaches include:

1. **Consulting key users**: Key users from various departments can provide insights into which actions are critical for their roles or identify actions that are most problematic from a performance standpoint.
2. **Support/service desks**: Support or service desks can help identify the most troublesome actions within Applications by reviewing the tickets they receive related to those Applications.
3. **Involving Functional Testing teams**: Functional Testing teams are responsible for testing Applications before release. They typically maintain detailed workflows for the most important Applications, from which you can filter the actions you want to measure.

4. **Engaging Application owners**: Application owners or teams are responsible for specific Applications and should be knowledgeable about the most frequently used actions or those that are most problematic in terms of performance.

Step 3. Defining measurements

Now that you have identified the Applications and their respective workflows, the next step is to determine which actions need to be measured for performance. While automating the Applications is essential—and Login Enterprise will notify you if an Application fails to complete its actions—measuring performance is where Login Enterprise truly excels.

Challenge your measurements

It is crucial to understand the purpose behind each measurement you add. Ask yourself: Why do you want this action measured? What insights do you hope to gain from this measurement? If the action is not performing as intended, what implications does this have?

Common measurements

The most common actions measured are those that require waiting for results, as these are typically the most affected by performance issues. A useful rule of thumb is that the longer an action takes to complete, the more it is likely to be impacted by performance degradation. Below are some examples:

- Logging in to an Application
- Searching for data
- Clicking on links
- Opening a report/data
- Loading Calendar overview
- Saving a file on a share

We also recommend selecting steps in your process that have dependencies and can impact overall performance. For example, performing a search involves querying a database. Measuring this action can provide valuable insights into the health of your database. Any slowness recorded in the Application script may indicate that the database is not optimized or is reaching its maximum capacity.

Step 4. Defining requirements

Each Application and its associated actions within the workflow may have specific requirements that must be met for successful execution of the Application script. These requirements need to be clearly defined to ensure the script can run smoothly.

Requirements can vary widely and may include providing licenses for test users or specifying datasets to be used within the Application. Below are a few examples:

1. **Login information**: Username and password for accessing the Application.
2. **Application-specific Logins/Licenses**: Any necessary logins or access credentials specific to the Application.
3. **Licenses**: Office 365 licenses or other Application-specific licenses required for operation.
4. **Files to be opened**: Provide necessary files, such as .docx and .pdf documents, that will be used in the workflow.
5. **Define Save File Locations**: Specify where files should be saved during the workflow process.
6. **Create Test Data in Data-Sensitive Applications/Databases**: For example, create a test patient record in a healthcare system to prevent data mutation in live datasets.

Step 5. Final review

Once all the information mentioned above has been collected, it is advisable to document it in a Word or Excel file. This allows you to describe each step clearly, ensuring that it leads to a functioning Application script.

Additional resources

- To learn about difficult Applications, see Difficult Applications.
- For information on the Application X-Ray, see Application X-Ray.
- For creating workloads using the Application X-Ray, see Creating workloads using the Application X-Ray.
- For more details on the Script Editor, its benefits, use cases, and more, see the Script Editor.
- To learn more about the Script Recorder, see the Script Recorder.

Difficult Applications

- Overview
- 1. Java Applications
- 2. In-VDI / Desktop Remote Applications
- 3. Layered Applications
- 4. In Application Browsers

Overview

This section outlines key considerations for scripting Applications, focusing on various Application types and their compatibility with automation tools. Understanding these nuances will help ensure effective automation and interaction with different Application environments.

1. Java Applications

Java Applications typically run within a virtual machine, isolating them from direct interactions with external tools. This encapsulation means that Java Applications do not expose information accessible by Application X-Ray. However, basic operations, such as starting and stopping the Application and measuring start time, remain functional.

2. In-VDI / Desktop Remote Applications

In-VDI or Desktop Remote Applications operate similarly to RDP sessions within a remote environment. The RDP window acts as a black box for Application X-Ray, preventing access to internal information. Nonetheless, basic operations, including starting and stopping the RDP session and measuring start time, can still be performed.

3. Layered Applications

Layered Applications may obscure their windows with other elements, making it difficult for Application X-Ray to locate expected windows or fields. Despite this limitation, basic operations like starting and stopping the layered Application and measuring start time are still possible.

4. In Application Browsers

Some Applications, such as Microsoft Teams, appear as standard Windows Applications but operate with an embedded web browser for content and interactions. If the Application uses Chromium-based technology, it allows interaction through accessibility switches, enabling automation tools to function effectively. However, if the Application uses browsers like Firefox or Internet Explorer, the Application X-Ray treats it as a black box, limiting accessibility.

Using the Script Editor

- Overview
- Benefits of Using Script Editor
- Requirements
- Downloading and Installing Script Editor
- Script Editor Functionality Breakdown
- Creating a New Application
- Designing Application Scripts
- Debugging with the Script Editor
  - Verbose Logging
- Best Practices
- Tutorials
  - Using // TARGET: Comments for CLI Automation
  - Why Use // TARGET: Comments?
  - Understanding the "Magic Comment" // TARGET:
  - Format for // TARGET:
  - Using Environment Variables
  - Adding Command-line Options and Arguments
  - Practical Examples
  - Specifying the "Start in" Path

Overview

Login Enterprise Script Editor is a tool designed to assist administrators in creating custom application workflows. These workflows can be seamlessly incorporated into various testing scenarios, including Application, Load, and Continuous Tests. By leveraging Script Editor, administrators can streamline test creation and execution, ensuring comprehensive and efficient testing processes tailored to specific application requirements.

Script Editor includes Web Recorder for web apps, in addition to Script Recorder v1 for Windows apps. To learn more about Script Recorder, its use cases, benefits, and more, see Using the Script Recorder.

Benefits of Using Script Editor

Benefit	Explanation
Efficient workflow creation	Simplifies the creation of custom application workflows, whether creating new scripts or modifying existing ones.
Debugging tools	Provides in-line syntax error detection and a comprehensive Error List for efficient debugging.

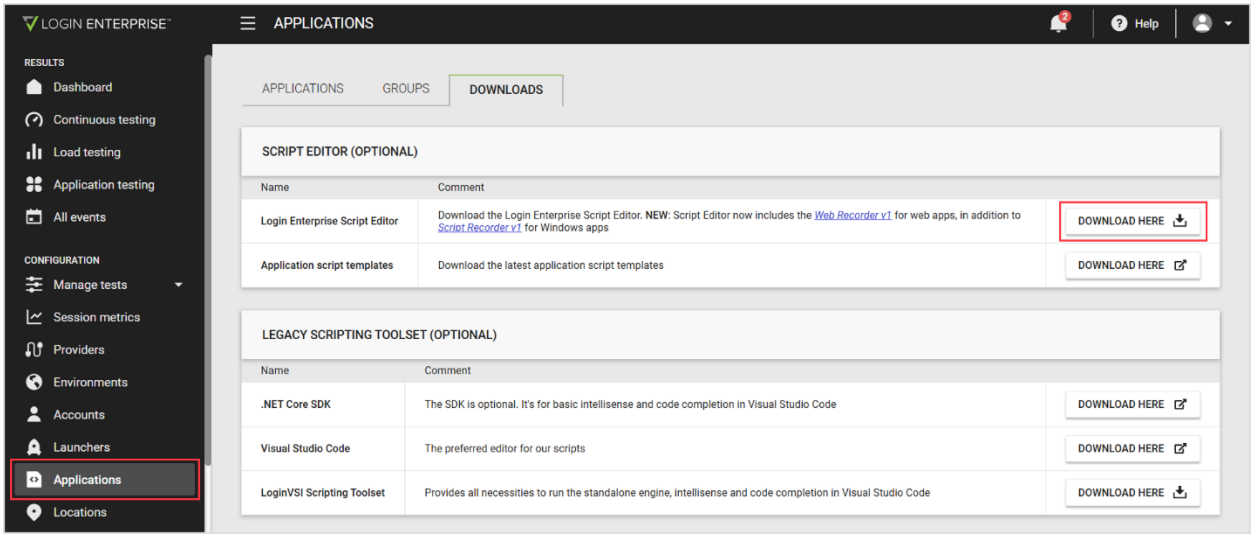
IntelliSense support	Enhances scripting accuracy with context-aware suggestions and auto-completion of the Login Enterprise native commands.
Integrated Script Recorder	Includes a built-in Script Recorder for capturing and automating user interactions, enhancing script development efficiency.
CLI Automation with // TARGET: Comments	Facilitates Command Line Interface (CLI) automation by using `//TARGET:` comments in conjunction with the Script Recorder.
Application type flexibility	Supports both executable and browser-based applications, with options to customize scripts according to specific testing needs.
Standalone and portable	Script Editor is standalone and portable, allowing it to run off a network share without needing installation.

Requirements

- Windows 8 and later
- Windows Server 2012 R2 and later
- .NET Framework 4.8

Downloading and Installing Script Editor

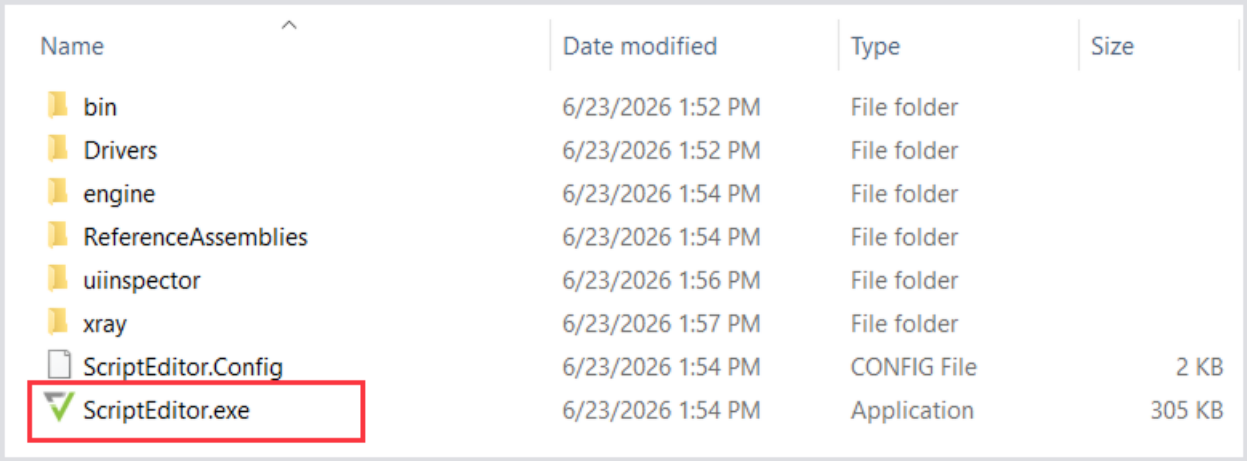
1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Applications**.
2. In **Applications**, select **Downloads** from the tab menu and download the **Script Editor** file (the file with the integrated Script Recorder will start downloading immediately).



TIP: Alternatively, Script Editor can be downloaded from the sidebar menu under **Other** > **Downloads** > **Applications**.

3. Extract the contents of the zip file onto your machine.

4. Open the folder and run the ScriptEditor.exe file.



Name	Date modified	Type	Size
bin	6/23/2026 1:52 PM	File folder	
Drivers	6/23/2026 1:52 PM	File folder	
engine	6/23/2026 1:54 PM	File folder	
ReferenceAssemblies	6/23/2026 1:54 PM	File folder	
uiinspector	6/23/2026 1:56 PM	File folder	
xray	6/23/2026 1:57 PM	File folder	
ScriptEditor.Config	6/23/2026 1:54 PM	CONFIG File	2 KB
ScriptEditor.exe	6/23/2026 1:54 PM	Application	305 KB

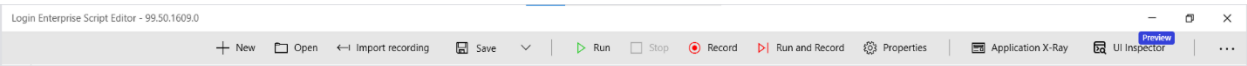
NOTE: When you click on the .exe file, a blue window may appear advising you not to run the file. This warning may be due to the file being unrecognized or from an unknown source. To proceed and run the file, click **More info**, and then click **Run anyway**.

Script Editor Functionality Breakdown

At the top of the Editor, you will find several options:

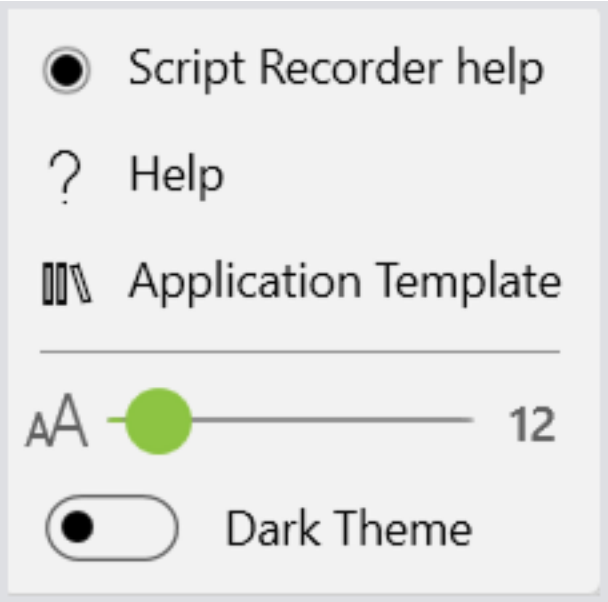
Functionality	Explanation
New	Creates a new application script.
Open	Opens an existing application script.
Import recording	Imports an existing script recording (JSON file).
Save	Saves the script. A "Save As" option is also available from the drop-down menu.
Run	Saves and executes the script against your target application.
Stop	Stops the script execution.
Record	Records interactions with the target application to generate a script automatically. For more information, see Starting the Recording: Recording the Script.
Run and record	Allows you to execute an already created script and then automatically opens the recorder at the end of the script, enabling you to continue recording from where it previously left off. For details, see Run and Continue Recording.
Properties	Allows you to change the properties defined when creating the application.

Application X-Ray	Opens Application X-Ray to analyze the target application. For more information, see Using the Application X-Ray.
UI Inspector (Preview)	Helps you identify and inspect UI elements in Windows applications. To learn more, see the UI Inspector (Preview).



From the options menu (…), there are additional choices:

Functionality	Explanation
Script Recorder help	Navigates to the instructions on using Script Recorder.
Help	Navigates to the Application Script Recording documentation.
Application Template	Navigates to the Workload Templates library.
Font Size	Changes the font size of the code in the Editor.
Dark Theme	Enables the dark theme.



Creating a New Application

To get started, create a new application or open an existing one.

Select **Create a New Login Enterprise Application** and choose the application type that matches your recording scenario:

- **Desktop Application** for WinApp Recorder.
- **Browser Application** for Web Recorder.

For step-by-step instructions, refer to the related sections in WinApp Recorder and Web Recorder guides.

Designing Application Scripts

After creating your first application, you will be presented with a default template that you can use as a foundation for building your script.

We recommend using the Login Enterprise Script Recorder or Application X-Ray to script your applications. To learn about Script Recorder, see Using the Script Recorder. For information on the Application X-Ray, see Using the Application X-Ray.

However, for scenarios where Script Recorder or Application X-Ray might not suffice, Script Editor provides IntelliSense for the Login Enterprise native commands. This feature simplifies the creation process by offering context-aware suggestions and guidance.

```
9          type
.0         S
.1     }
.2 }
```

- GetType
- Type
- TypeCommand
- typeof

void ScriptBase.Type(object component, [int cpm = 300], [bool hideInLogging = true]) (+ 1 overload)
Simulates keyboard input of a human readable form of the given component

Debugging with the Script Editor

Syntax errors are identified both in-line within the script and listed in the Error List at the bottom of the Script Editor interface.

```
8          Wait(2);
9          Type("test",cpm=500);
10         STOP();
11     }
12 }
```

Error List | Debug Output

Line	Severity	Error
9	Error	The name 'cpm' does not exist in the current context

Once you execute your script, the debug output will comprehensively display all actions performed by our engine.

Error List | Debug Output

```
Starting Standalone Engine. Please wait...
Script Log [Information] 'Application.Start(, , , 180, False, True)'
Script Log [Information] 'The script created a new process: 12160:wordpad. It will be cleaned up after the script ends.'
The script reported a created process: 'wordpad'  This process will be terminated at the end of the script
Script Log [Information] 'Waiting for the UI of the process to become responsive'
Script Log [Information] 'Waiting for the process UI to become visible'
Script Log [Information] 'Found main window with title WordPad and class WordPadClass'
The script reported a measurement: 'app_start_time' = 382
The script requested the app start screenshot
Script Log [Information] 'Wait 2 seconds'
Script Log [Information] 'Finished waiting'
Script Log [Information] 'Application.Stop()'
Script Log [Information] 'Verifying the application has stopped.'
Verifying process wordpad
The script ended
```

All in all, Script Editor is a powerful tool for administrators, offering essential features for creating, debugging, and optimizing custom application workflows. Built-in capabilities, such as syntax error detection, IntelliSense, and detailed debug output, along with the integrated Script

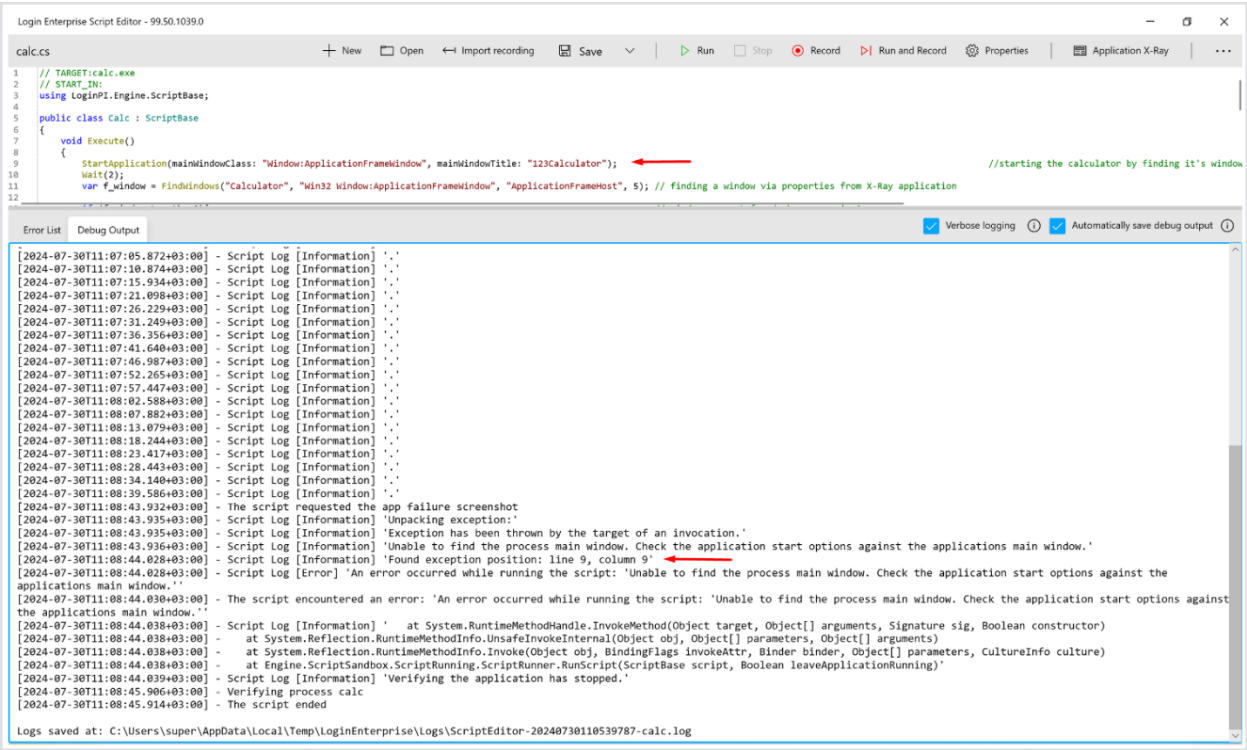
Recorder, support efficient script development and troubleshooting within the Login Enterprise environment.

Verbose Logging

To enhance debugging, you can enable the verbose logging option to include line numbers in the debug output. For this:

In Script Editor, select the **Verbose Logging** checkbox.

Whenever the script fails, the debug output displays the line of script that encountered the error:



NOTE: Enabling verbose logging may affect script runtime.

Best Practices

1. **Consistent naming**: While the specific names for command-line arguments and options are determined by the developers of the target applications, e.g., `/t` for Microsoft Office, ensure that any custom parameters or arguments you define within your scripts are clear and consistent.

2. **Quotation usage**: Encapsulate paths and arguments in double quotes to handle spaces correctly.
3. Help messages: Include comments or help messages within your script to guide users on using `// TARGET:` effectively.
4. **Testing**: Validate `// TARGET:` paths and arguments thoroughly before distributing scripts to ensure functionality across different environments.

Tutorials

Using // TARGET: Comments for CLI Automation

You can leverage command-line parameters and arguments to automate application tasks. This section explains how to use the `// TARGET:` magic comment to specify the target application, its executable path, command-line arguments, and environment variables. It covers both legacy behavior and recent enhancements.

Why Use `// TARGET:` Comments?

Benefit	Explanation
Specify executable paths	Define the exact path to the executable you want to run.
Pass command-line arguments	Include additional parameters or options needed by the executable.
Utilize environment variables	Leverage system environment variables to create more portable and adaptable scripts that work across different systems and configurations.
Maintain compatibility	Ensure scripts work seamlessly across different systems while incorporating new Script Recorder features.

Understanding the "Magic Comment" `// TARGET:`

The `// TARGET:` comment is pivotal for specifying the application to invoke, allowing you to define the executable path and any associated arguments within your script.

Format for `// TARGET:`

General syntax:

```
// TARGET: path_to_executable [options] [arguments]
```

Note there's an option to have the executable path encapsulated or not in double quote characters. For example:

```
// TARGET:"%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE" /t
"%ProgramFiles%\TestExcel.xlsx"
```

The following example is acceptable and will also work. The .exe path doesn't have double quotes around it, even though the path includes a space character:

```
// TARGET:%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE /t
%ProgramFiles%\TestExcel.xlsx
```

Clarifications:

- There has been no change in how the target path is handled in the target engine, Script Editor, or Application X-ray compared to previous versions.
- Script Recorder now supports invoking the defined target app with command-line options.
- The initial target app path can optionally have double quotes around it, even if it contains spaces. For consistency and clarity, it is recommended to always use double quotes around paths with spaces in example code blocks.
- Any argument or path after the initial target app path must have double quotes around it if it contains spaces. This ensures the command line is parsed correctly.

Example with optional quotes around the executable path:

```
// TARGET:%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE /t
"%ProgramFiles%\TestExcel.xlsx"
```

Example with quotes around the executable path:

```
// TARGET:"%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE" /t
"%ProgramFiles%\TestExcel.xlsx"
```

Both examples above are valid, but for clarity, we recommend using double quotes around paths with spaces.

Using Environment Variables

Environment variables are placeholders that automatically resolve to specific paths or values within the operating system environment. For instance, `%ProgramFiles%` resolves to the Program Files directory in Windows.

Adding Command-line Options and Arguments

Command-line options modify the behavior of the executable. These can range from simple flag options, e.g., `-v` for verbose mode, to key-value pairs, e.g., `--output=filename`.

Practical Examples

Here are examples demonstrating the use of command-line options and environment variables:

1. **Launching an application with environment variables**: Launches an application using predefined environment variables for paths.

```
// TARGET:"%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE"
```

2. **Opening a file in Excel with command-line options**: Opens a specific Excel file using Excel with command-line options.

```
// TARGET:"%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE" /t
"%ProgramFiles%\TestExcel.xlsx"
```

3. **Running a script with environment variables and options**: Launches an executable with a configuration file and enables verbose mode using environment variables.

```
// TARGET:"%ProgramFiles%\MyApp\myapp.exe" --
config="%AppData%\MyApp\config.cfg" --verbose
```

Specifying the "Start in" Path

In addition to defining the executable path and command-line arguments, you can specify the working directory of the executable using the "Start in" path. This path indicates where the executable should run from and is crucial for applications that rely on relative paths.

Example of specifying the "Start in" path:

```
// START_IN: "%ProgramFiles%\MyApp"
```

This sets the working directory for the executable to the specified path, ensuring that any relative paths used by the application are resolved correctly.

Using the Script Recorder

- Overview
- Benefits of Using Script Recorder
- Downloading and Installing Script Recorder
- Additional Resources

Overview

Login Enterprise Script Recorder is a tool that captures and records the sequence of user interactions with a computer system or software application. It records user inputs, such as mouse clicks, keyboard actions, and other commands, creating a script that can be replayed to automate repetitive tasks or test software functionality. Script recorders are commonly used in testing, automation, and workflow optimization.

In the context of Login Enterprise, Script Recorder makes it easier to create Application scripts and get maximum value from the Tests. Our vision with the Script Recorder is to make it accessible and easy to use, allowing you to generate application scripts even if you are not familiar with scripting languages or may not have a background in scripting.

Script Recorder v1 is integrated into Script Editor.

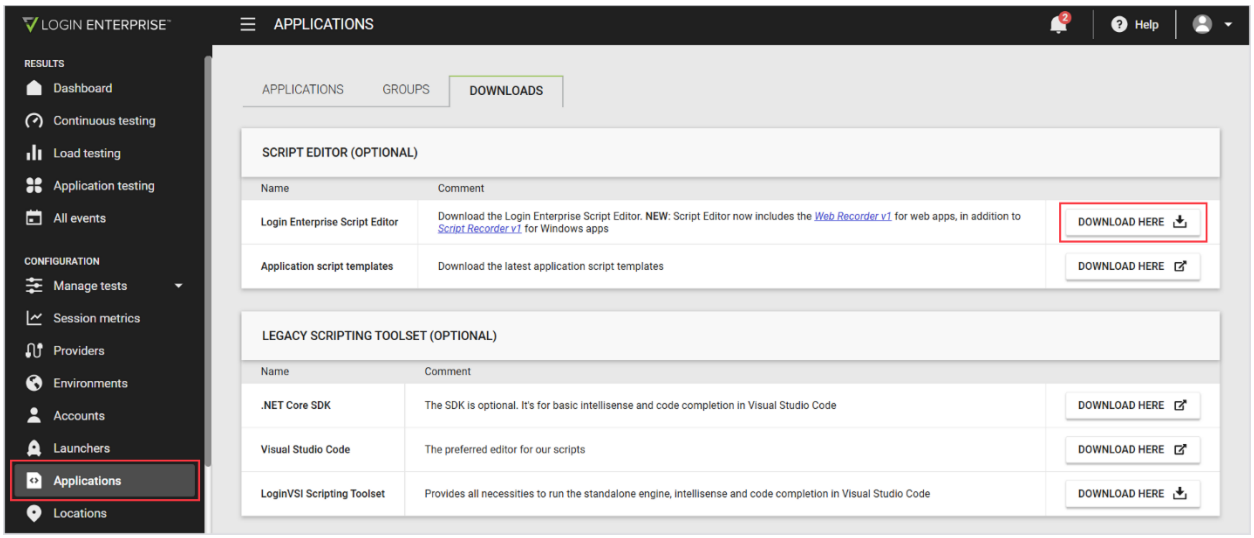
Benefits of Using Script Recorder

Benefit	Explanation
Accelerated workload creation	Develop workloads faster, enabling you to quickly perform valuable testing on target applications. For instance, if you previously spent one hour per workload, you can now complete the task in just 10 minutes. This efficiency boost allows you to script six workloads per hour instead of one. This improvement is particularly beneficial if you have extensive backlogs. You may have hundreds of applications that need scripting with Login Enterprise, but lack the necessary time and resources. Script Recorder enables you to address their backlog more rapidly, enhancing productivity and streamlining their workflow.
Enhanced multi-application support	Script Recorder enables you to quickly create multiple workloads for a single target application. This focused capability helps you manage and test different scenarios within the same application efficiently.
Backlog reduction	If you have a backlog of applications that need scripting against the target application, the Script Recorder can help you address this backlog more efficiently by speeding up the scripting process.

Natural interaction workflow	You can perform actions continuously while Script Recorder captures your interactions, making the process more intuitive. Unlike the previous method of recording line by line, this continuous recording aligns better with your natural workflow and reduces interruptions in thought processes.
User-friendly experience	Script Recorder simplifies the process by reducing the amount of code you need to manage. You can create workloads in a single recording session, moving closer to the low-code model. This accessibility allows you to perform application scripting effectively, even without a scripting or programming background.
Lower barrier to entry	Script Recorder lowers the entry barrier, enabling you to quickly gain value from Login Enterprise. This increased ease of use ensures that you can script and test applications without extensive training or experience.

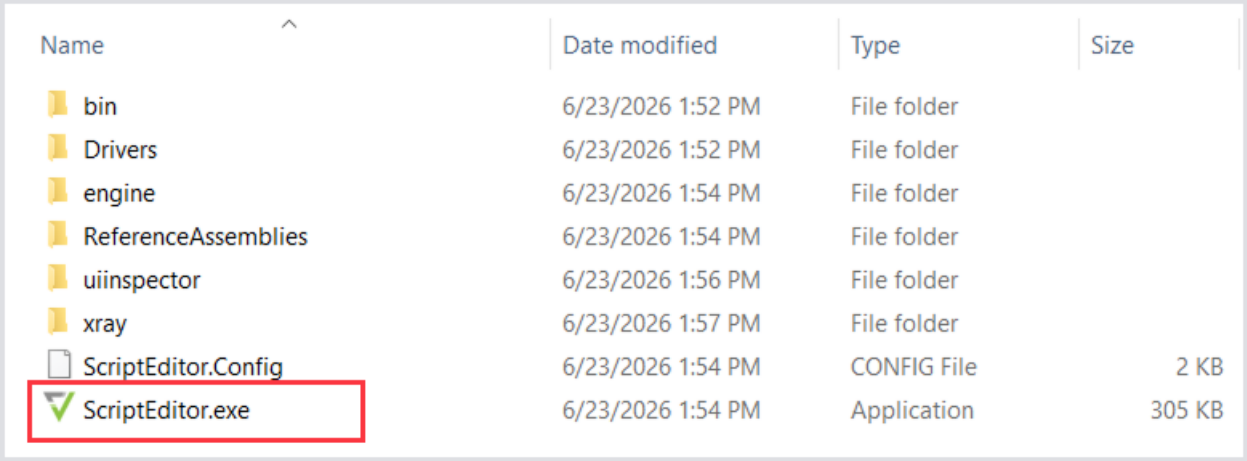
Downloading and Installing Script Recorder

1. In the **Login Enterprise sidebar menu**, navigate to **Configuration** > **Applications**.
2. In **Applications**, select **Downloads** from the tab menu and download the **Script Editor** file (the file with the integrated Script Recorder will start downloading immediately).



TIP: Alternatively, Script Editor can be downloaded from the sidebar menu under **Other** > **Downloads** > **Applications**.

3. Extract the contents of the zip file onto your machine.
4. Open the folder and run the ScriptEditor.exe file.



Name	Date modified	Type	Size
bin	6/23/2026 1:52 PM	File folder	
Drivers	6/23/2026 1:52 PM	File folder	
engine	6/23/2026 1:54 PM	File folder	
ReferenceAssemblies	6/23/2026 1:54 PM	File folder	
uiinspector	6/23/2026 1:56 PM	File folder	
xray	6/23/2026 1:57 PM	File folder	
ScriptEditor.Config	6/23/2026 1:54 PM	CONFIG File	2 KB
ScriptEditor.exe	6/23/2026 1:54 PM	Application	305 KB

NOTE: When you click on the .exe file, a blue window may appear advising you not to run the file. This warning may be due to the file being unrecognized or from an unknown source. To proceed and run the file, click **More info**, and then **Run anyway**.

Additional Resources

- For instructions on using the WinApp Recorder, see WinApp Recorder.
- To learn how to use the Web Recorder, see Web Recorder.

WinApp Recorder

- Overview
- Getting Started
- Creating a New Application
- Starting the Recording: Recording the Script
  - Recording without Target Application
- Stopping the Recording: ScriptResultWindow
- Settings
  - General
  - Action
    - Editing an Action Name
    - Advanced Parameters
    - Additional Parameters of Use
- Running the Script
  - Run and Continue Recording
  - Keeping the Application Open after Recording
- Exporting the Script as JSON
- Importing the Script as JSON
- Restrictions and Limitations
- Recommendations
- Tutorials
  - Launching Apps with a START_IN Magic Comment
  - CLI Parameters and Arguments
  - Wildcards for Class Names, Names, Titles, and Automation IDs

Overview

The instructions below describe how to use the WinApp Recorder. For general information on Script Recorder and related use cases, see Using Script Recorder.

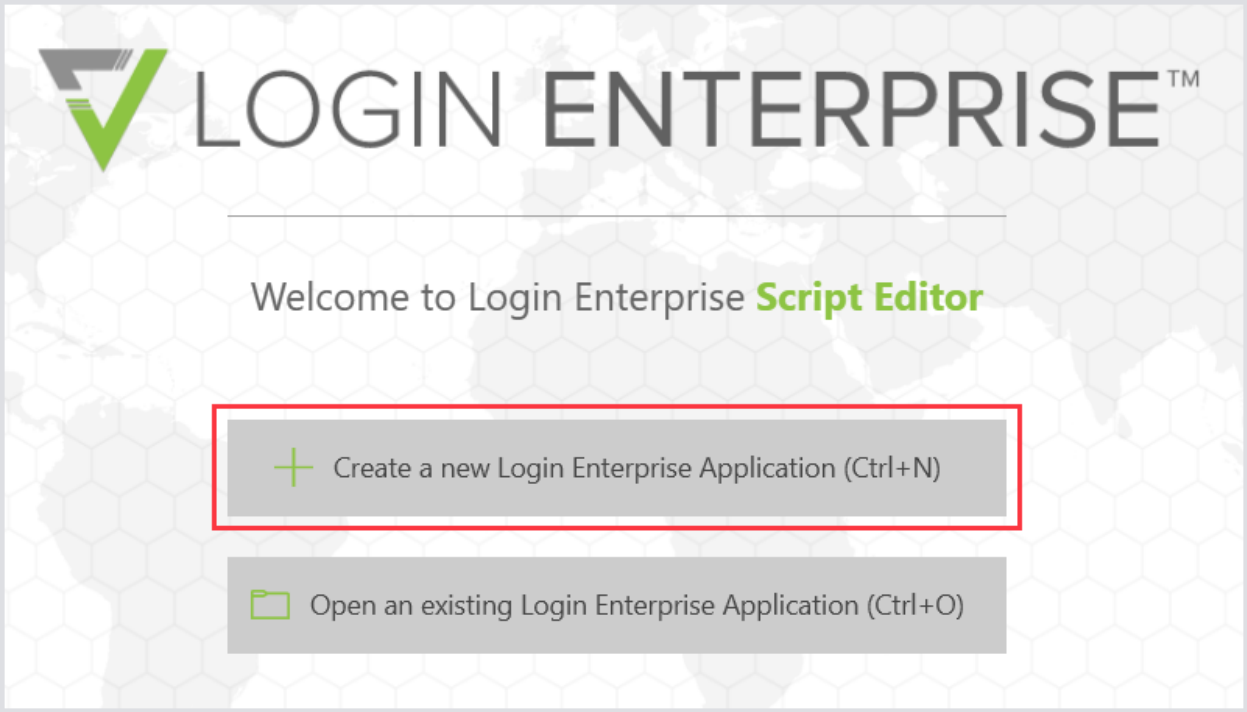
NOTE: In v6.5, Script Recorder moved to a new UI automation backend (FLAUI) and uses a standardized XPath-style locator format. Existing scripts remain supported, but locator generation and robustness have improved. For details on Script Recorder (WinApp) implementation, see Understanding Script Recorder (WinApps) Implementation.

Getting Started

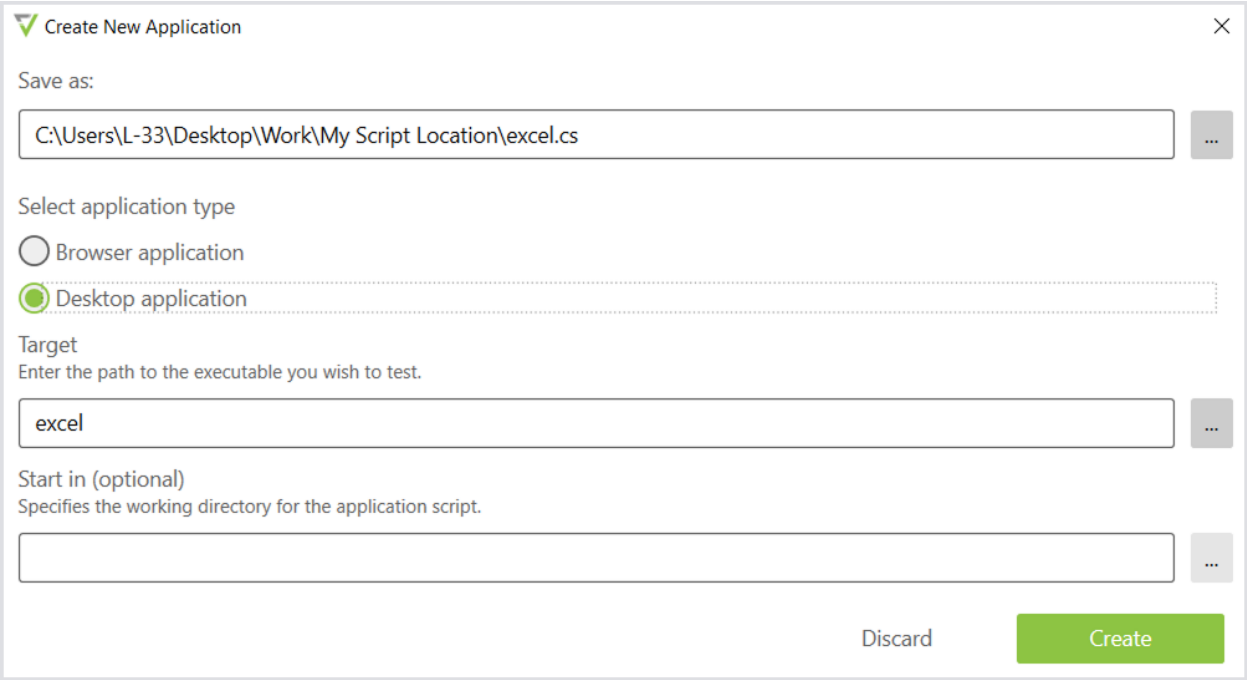
Download the latest Script Editor package containing the WinApp Recorder v1 from your admin or setup email. For details, see Downloading and installing Script Recorder.

Creating a New Application

1. In the **Login Enterprise Script Editor**, click **Create a new Login Enterprise Application**.

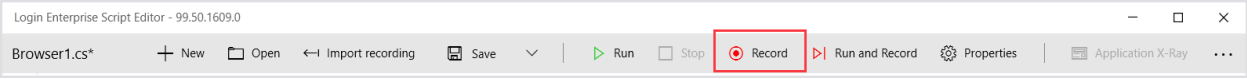


2. Click the three-dot icon to the right of **Save as**, choose a folder for the script, and enter a file name.
3. In **Application type**, select **Desktop application**.
4. In **Target**, enter the path to the executable you want to test.
5. In **Start in**, you can specify the working directory for the application script.
6. Click **Create**.



Starting the Recording: Recording the Script

In the **Script Editor top menu**, click **Record** to start the recording. When the recording starts, a player opens where you can perform actions.

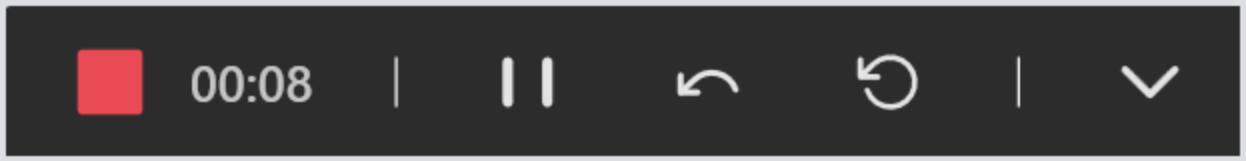


The Recorder uses visual indicators to show the status of detected elements:

- **Orange** indicates that the Recorder is in the process of locating an element.
- **Green** indicates that the element is ready and can be interacted with. Actions performed on green-highlighted elements are recorded.

The Recorder supports the following actions:

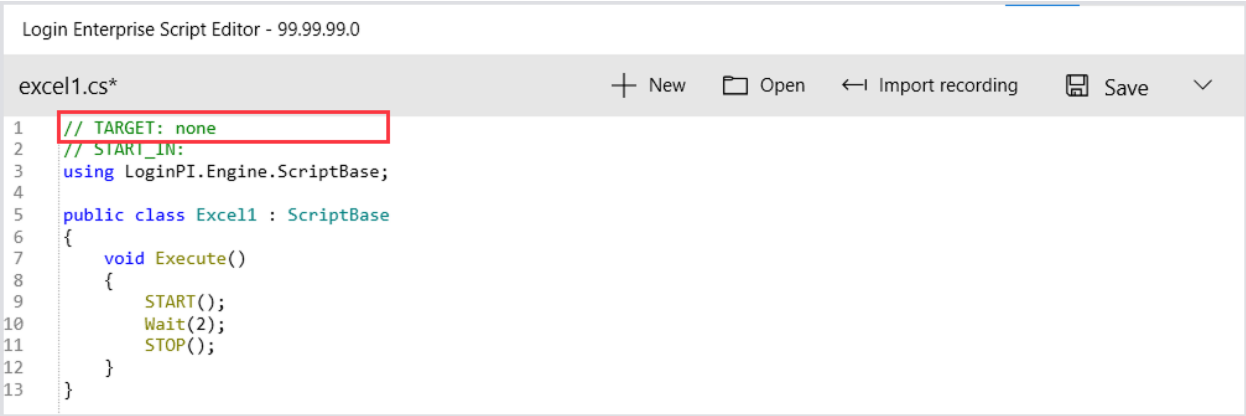
- Stop the recording of actions
- Start the recording of actions
- Pause or continue the recording
- Remove the last recorded action
- Remove all recorded actions
- Show or hide the list of recorded actions (downward/upward arrow icon)



Recording without Target Application

You can record a script without specifying a target application (the executable to test) when creating a new application. This way, the recording starts from the Windows desktop instead of launching a specific application. To do this, set the `//Target:` value in the script to `none`.

NOTE: The Recorder does not fully recognize the Windows Start menu. However, you can still interact with it by using keyboard actions such as **Tab** and **Enter** to navigate to the item you need.



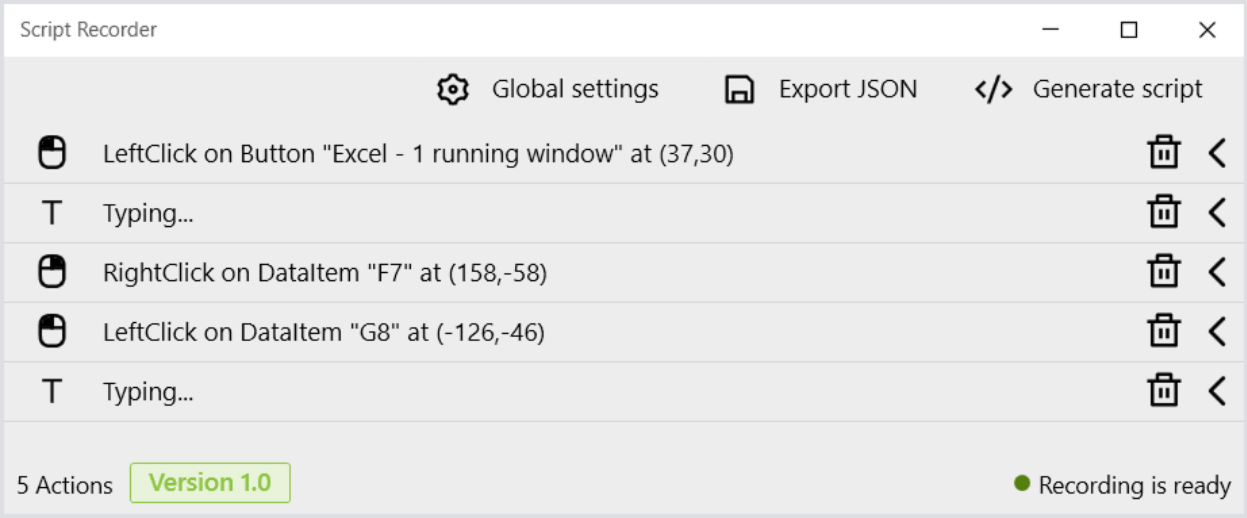
After the recording is complete, follow the standard workflow:

1. Generate the script, and then copy it to the clipboard.
2. Paste the script into the Script field. Make sure to remove the `start` and `stop` commands.
3. Click **Run**.

For details, see Running the script.

Stopping the Recording: ScriptResultWindow

When you stop the recording, the ScriptResultWindow shows an overview of all recorded steps (before generating the script, right after the recording was stopped).



NOTE: To prevent exposing sensitive information such as usernames and passwords, the Recorder does not display the actual typed values in the script output.

Instead, typing actions are shown as **"Typing…"** in the script and in the results.

In the Overview screen, you can adjust settings, export the script as JSON, and generate the script to run it.

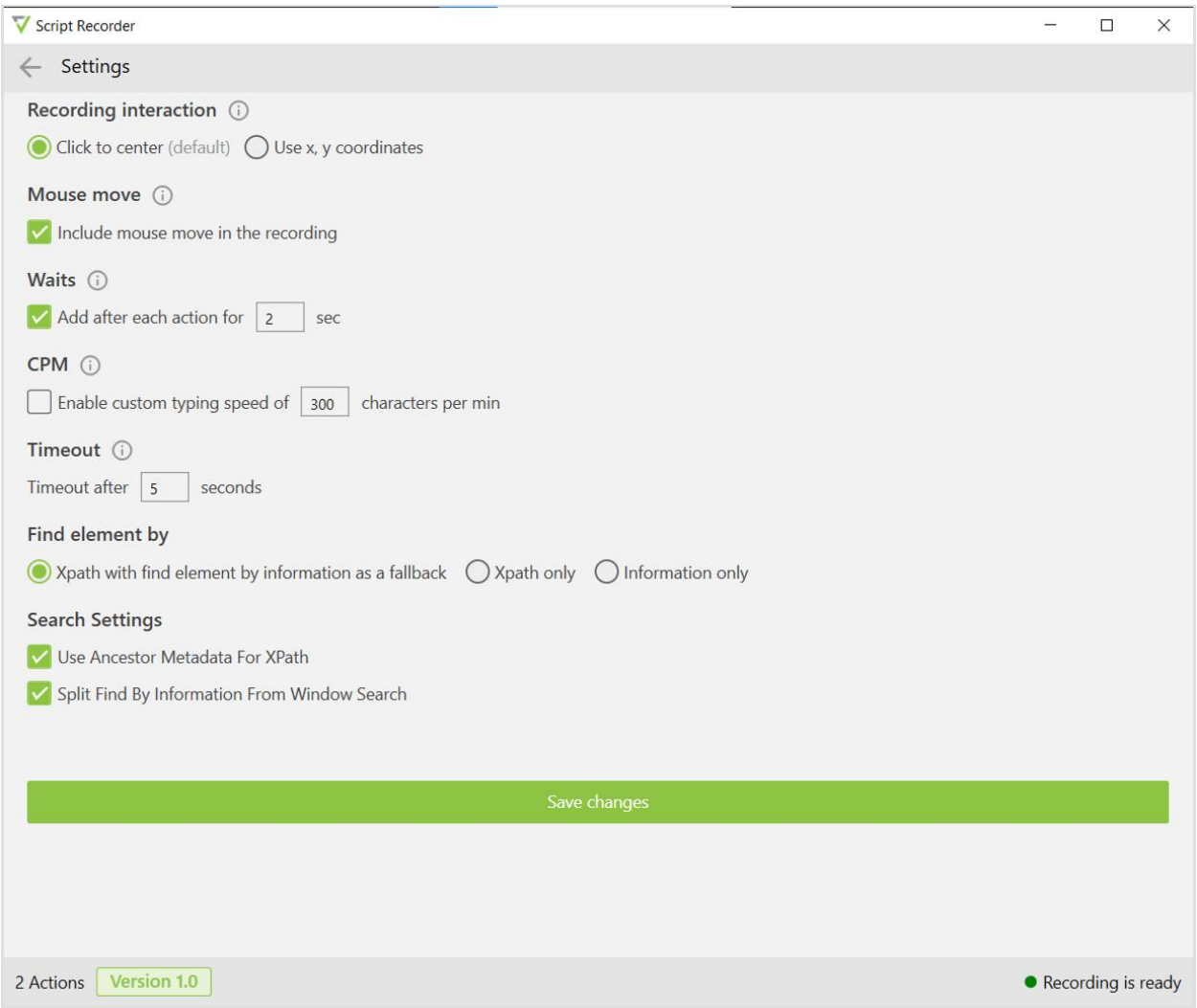
Settings

General

NOTE: Changes made in General settings apply to all action settings.

Option	Description	
Recording interaction	Determines mouse positioning when interacting with elements.	
	Click to center (default)	The script will always click in the center of the element.
	Use X and Y coordinates	The script will click within the element as it was recorded.
Mouse move	Determines the inclusion or exclusion of mouse movements in the script. Once you select the checkbox, the actual mouse movements will be included in the script. Note that this might lead to an increased runtime of the script.	
Waits	Adds a pause after each recorded step in the script.	
CPM (Characters Per Minute)	Adjusts typing speed. If you leave the checkbox unselected, the default value will be used. If you select the checkbox, the CPM value will appear in the script. The default is 300 characters per minute.	

Timeout	Specifies how long the Recorder repeatedly attempts to find the target element before the action fails. If the element is not found within the timeout, the action fails. The default is 5 seconds.	
Find element by	Defines a strategy for locating UI elements during playback.	
	XPath with information fallback (default)	Attempts to locate the element using XPath first. If XPath fails, the Recorder falls back to locating the element based on its identifying information (such as name, automation ID, class name, and other properties). Provides a balance between performance and reliability and is recommended for most scenarios.
	XPath only	Locates the element using only XPath. Typically faster but more fragile, as changes to the UI structure (such as added or removed containers) can cause the XPath to become invalid.
	Information only	Locates the element using identifying properties (such as name, automation ID, and class name) without using XPath. More resilient to UI structure changes, but may take longer to locate elements.
Search settings	Ancestor metadata	When enabled, XPath includes metadata from parent (ancestor) elements to more precisely identify the target element, which can improve precision but may reduce robustness in applications with dynamic containers or changing window titles.
	Split Find Information from Window Search	When enabled, the search first locates the window by XPath (e.g., Calculator), then finds the element within it by Information (e.g., button #8). If disabled, the search starts from the Desktop without narrowing to a specific window first.



Action

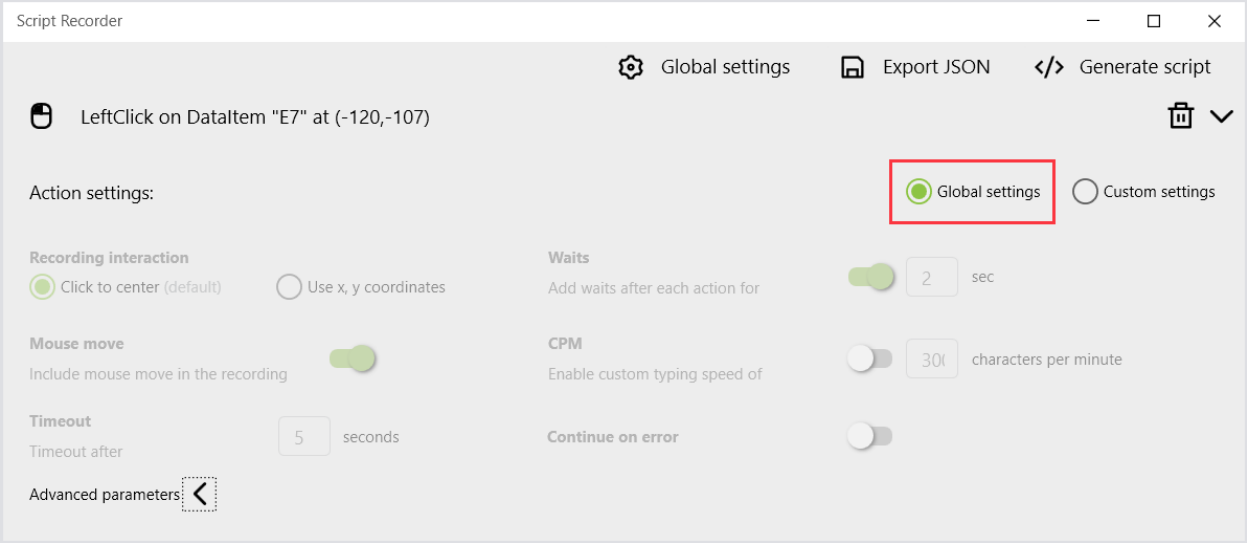
NOTE: Changes made in Action settings apply only to the selected action.

Option	Description	
Recording interaction	Determines mouse positioning when interacting with elements.	
	Click to center (default)	The script will always click in the center of the element.
	Use X and Y coordinates	The script will click within the element as it was recorded.
Waits	Adds a pause after each recorded step in the script.	

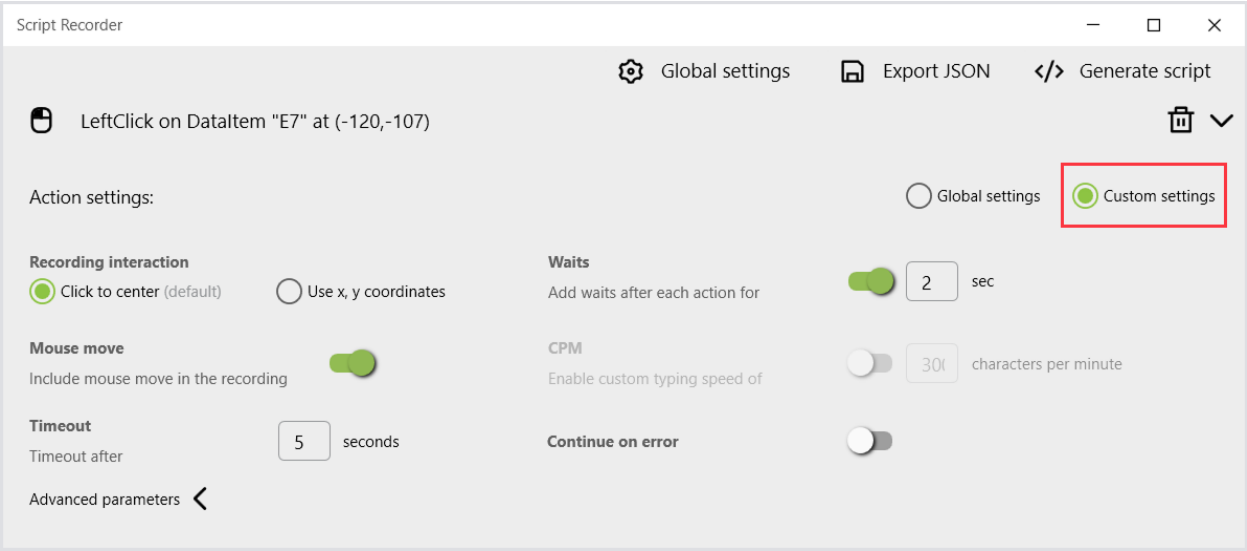
Mouse move	Determines the inclusion or exclusion of mouse movements in the script. Once you select the toggle, the actual mouse movements will be included in the script. Note that this might lead to an increased runtime of the script.
CPM (Characters Per Minute)	Adjusts typing speed. If you leave the toggle unselected, the default value will be used. If you select the toggle, the CPM value will appear in the script. The default is 300 characters per minute.
Timeout	Specifies how long the Recorder repeatedly attempts to find the target element before the action fails. If the element is not found within the timeout, the action fails. The default is 5 seconds. When using Information only, consider increasing the Timeout, as element searches may take longer and could otherwise fail prematurely.
Continue on error	Available only in Action settings. Disabled by default. When enabled for an individual action, the script doesn't abort if that action fails and instead continues with the next action.

When you open an action step, the settings are shown automatically. Action steps can use global settings or custom settings:

- **Global settings**: The action step uses the global settings. The settings are read-only and can't be changed in the action step.



- **Custom settings**: The action step uses custom settings. Select this option to make the fields available for editing in the action step.



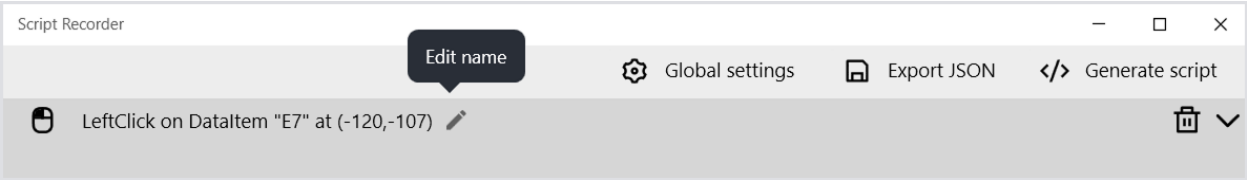
NOTE: Advanced and Additional parameters are always available for editing as they can't be configured as global settings. Some additional parameters only apply to information-based find methods. To learn more, see Advanced Parameters and Additional Parameters of Use.

Editing an Action Name

You can change the name of each action to make it more descriptive. The action name is shown in the script as a comment before the corresponding step.

To edit an action name:

1. Hover over the action.
2. Select the pencil icon.



3. Enter a new name.
4. Click **Submit**.

Advanced Parameters

You can use Advanced parameters to fine-tune how UI element locators are generated and tested for a selected action. There are 3 options available:

Option	Description

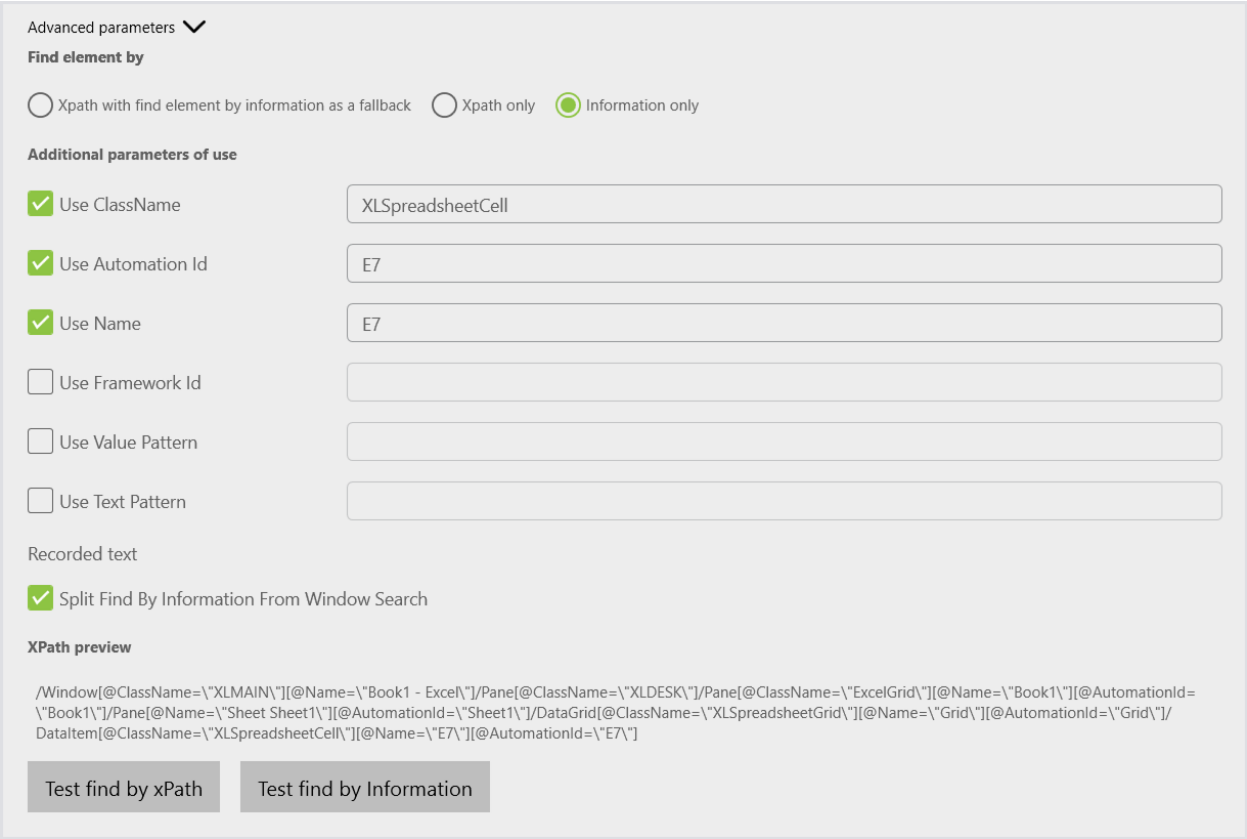
Find element by	Defines a strategy for locating UI elements during playback.	
	XPath with information fallback (default)	Attempts to locate the element using XPath first. If XPath fails, the Recorder falls back to locating the element based on its identifying information (such as name, automation ID, class name, and other properties). Provides a balance between performance and reliability and is recommended for most scenarios.
	XPath only	Locates the element using only XPath. Typically faster but more fragile, as changes to the UI structure (such as added or removed containers) can cause the XPath to become invalid.
	Information only	Locates the element using identifying properties (such as name, automation ID, and class name) without using XPath. More resilient to UI structure changes, but may take longer to locate elements.

Additional Parameters of Use

Use Additional Parameters of Use to refine which element properties are included when locating the target UI element. Framework ID, Value Pattern, and Text Pattern apply only to information-based find methods and are not used by XPath-only detection.

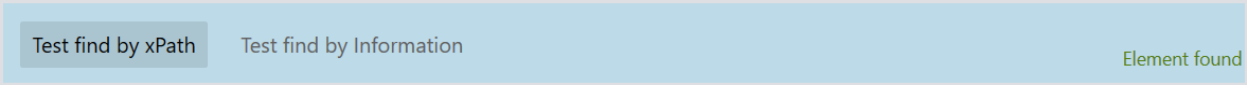
Option	Description
Use ClassName	Uses the element's class name as part of the criteria to locate the target element. The value can be manually adjusted, and wildcards (*) are supported. Use wildcards when the element's class name has a consistent prefix or pattern but may include dynamic suffixes. Example: Button*
Use Automation Id	Uses the element's automation Id to locate the target element. The value can be manually adjusted, and wildcards (*) are supported. Use wildcards when the automation Id is partially dynamic and only part of the value is stable. Example: loginButton*

Use Name	Uses the element's name property to locate the target element. The value can be manually adjusted, and wildcards (*) are supported. Use wildcards when the visible name contains variable text or additional context. Example: *Sign in*
Use Framework Id	The underlying technology used to build the UI element (for example, WPF or Win32). **Note**: Only used by find by information methods.
Use Value Pattern	The current value of elements that support the Value pattern (for example, text in a text box or a combo box selection). **Note**: Only used by find by information methods.
Use Text Pattern	The full text content of elements that support the Text pattern (for example, rich text or document controls). **Note**: Only used by find by information methods.
Use Ancestor Metadata for XPath	XPath includes metadata from parent (ancestor) elements to more precisely identify the target element.
Split Find by Information from Window Search	When enabled, the search first locates the window by XPath (e.g., Calculator), then finds the element within it by Information (e.g., button #8). If disabled, the search starts from the Desktop without narrowing to a specific window first.
XPath preview	Shows the XPath expression generated for the selected action using the current advanced parameters.

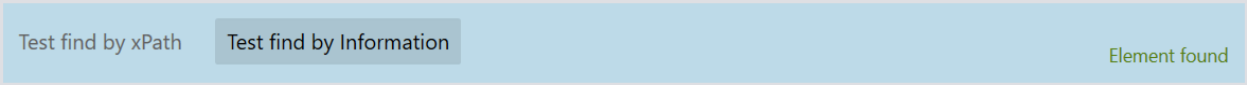


The following options allow you to verify that the target element can be successfully located:

Option	Description
Test find by XPath	Tests whether the target element can be successfully located using XPath. The application must be open at the element's location.

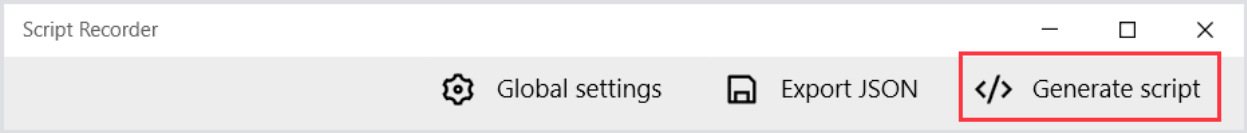


Option	Description
Test find by Information	Tests whether the target element can be successfully located using information-based identification. The application must be open at the element's location.

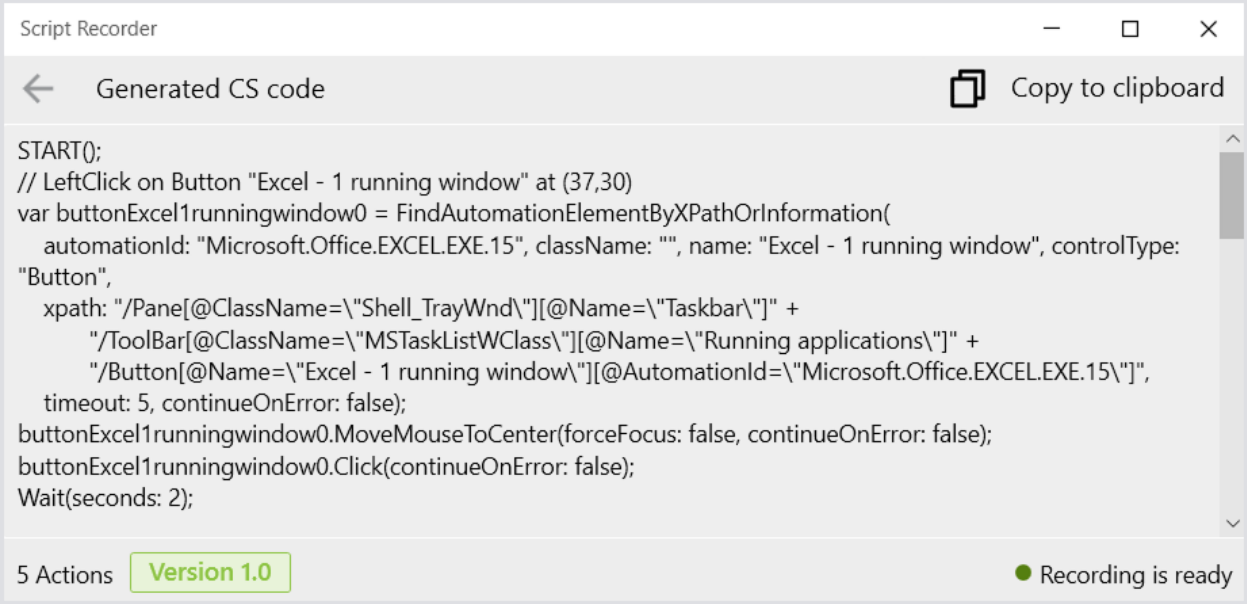


Running the Script

1. In the **ScriptResultWindow**, click **Generate script**. After generating the script, you can go back, change the settings, and generate the script (with new settings) again.



2. Copy the generated script to the clipboard:



NOTE: Script Recorder doesn't support notifications, such as "Copied to the clipboard." So, once you copy the script to the clipboard, it's saved; the Recorder is just not letting you know. You can now proceed and paste the script into the Editor.

3. Paste the copied script into the Script Editor:

IMPORTANT: Recorded scripts automatically include `start` and `stop` commands.

- When you record a script, the Recorder generates a complete sequence that includes starting the application, performing the recorded actions, and stopping the script. This differs from manually created scripts, which typically use separate `start`, `wait`, and `stop` commands.
- If you are copying recorded actions into an existing script, replace the existing `start`, `wait`, and `stop` commands with the generated `start` and `stop` commands to ensure the script runs correctly.

Login Enterprise Script Editor - 99.50.1609.0

Excell.cs* | New | Open | Import recording | Save | Run | Stop | Record | ...

```
1   // TARGET:"C:\Users\L-33\Desktop\Work\My Script Location\excel.cs"
2   // START_IN:
3   using LoginPI.Engine.ScriptBase;
4   using LoginPI.Engine.ScriptBase.Components;
5
6   public class Excell : ScriptBase
7   {
8       void Execute()
9       {
10          START();
11
12          START();
13  // LeftClick on Button "Excel - 1 running window" at (37,30)
14  var buttonExcel1runningwindow0 = FindAutomationElementByXPathOrInformation(
15      automationId: "Microsoft.Office.EXCEL.EXE.15", className: "", name: "Excel - 1 running window", controlType: "Button",
16      xpath: "/Pane[@ClassName=\"Shell_TrayWnd\"][@Name=\"Taskbar\"]" +
17             "/ToolBar[@ClassName=\"MSTaskListWClass\"][@Name=\"Running applications\"]" +
18             "/Button[@Name=\"Excel - 1 running window\"][@AutomationId=\"Microsoft.Office.EXCEL.EXE.15\"]",
19      timeout: 5, continueOnError: false);
20  buttonExcel1runningwindow0.MoveMouseToCenter(forceFocus: false, continueOnError: false);
21  buttonExcel1runningwindow0.Click(continueOnError: false);
22  Wait(seconds: 2);
23  // Typing...
24  Type(text: "{BACKSPACE}");
25  Wait(seconds: 2);
26
27  STOP();
28
29          Wait(2);
30          STOP();
31      }
32  }
```

4. In the **Script Editor top menu**, click **Run** to run the script.

Login Enterprise Script Editor - 99.50.1609.0

Browser1.cs* | New | Open | Import recording | Save | Run | Stop | Record | Run and Record | Properties | Application X-Ray | ...

The script should be available under the Debug Output tab:

Error List | Debug Output | Verbose logging | Automatically save debug output

```
[2026-06-23T20:02:34.995+03:00] - Starting Standalone Engine. Please wait...
[2026-06-23T20:02:43.697+03:00] - Logging to file by path: C:\Users\L-33\AppData\Local\Temp\LoginPI\Logs\Engine 20260623 200243.txt with minimum log level: Information
[2026-06-23T20:02:43.764+03:00] - Logging to Windows EventLog: Application. Minimum log level: Warning
[2026-06-23T20:02:47.783+03:00] - Script Log [Information] 'Starting script'
[2026-06-23T20:02:47.783+03:00] - Script Log [Information] 'Engine version is 99.50.1609+abbdc96519961c47f880dcd11160e7154b860647'
[2026-06-23T20:02:48.293+03:00] - Script Log [Information] 'Application.StartApplication(, , 180, False)'
[2026-06-23T20:02:48.293+03:00] - Script Log [Warning] 'ForceKillOnExit parameter is ignored. This behavior is configured in the test schedule action.'
[2026-06-23T20:02:48.425+03:00] - Script Log [Information] 'Found matching window already open (Book1 - Excel in EXCEL/1896). Attaching without launching.'
[2026-06-23T20:02:48.426+03:00] - Script Log [Information] 'The application is already running, the Engine will not measure the start time.'
[2026-06-23T20:02:49.142+03:00] - The script requested the app start screenshot
[2026-06-23T20:02:49.143+03:00] - Script Log [Information] 'Application.StartApplication(, , 180, False)'
[2026-06-23T20:02:49.143+03:00] - Script Log [Warning] 'ForceKillOnExit parameter is ignored. This behavior is configured in the test schedule action.'
[2026-06-23T20:02:49.155+03:00] - Script Log [Warning] 'Already tracking application.'
[2026-06-23T20:02:49.155+03:00] - Script Log [Information] 'The application is already running, the Engine will not measure the start time.'
[2026-06-23T20:02:49.875+03:00] - The script requested the app start screenshot
[2026-06-23T20:02:49.886+03:00] - Script Log [Information] 'Looking for all elements by xpath: /Pane[@ClassName="Shell_TrayWnd"][@Name="Taskbar"]/ToolBar[@ClassName="MSTaskListWClass"]
[@Name="Running applications"]/Button[@Name="Excel - 1 running window"][@AutomationId="Microsoft.Office.EXCEL.EXE.15"]'
[2026-06-23T20:02:50.041+03:00] - Script Log [Information] 'AutomationElement.MoveMouseToCenter(continueOnError:False)'
[2026-06-23T20:02:50.047+03:00] - Script Log [Information] 'this automation element has no window.'
[2026-06-23T20:02:51.829+03:00] - Script Log [Information] 'AutomationElement.Click(continueOnError:False, forceFocus:True)'
[2026-06-23T20:02:51.855+03:00] - Script Log [Error] 'AutomationElement.Focus failed'
[2026-06-23T20:02:51.880+03:00] - Script Log [Information] 'Wait 2 seconds'
[2026-06-23T20:02:53.887+03:00] - Script Log [Information] 'Finished waiting'
[2026-06-23T20:02:53.889+03:00] - Script Log [Information] 'Keyboard.Type(text="********", cpm=300, hideInLogging=True)'
[2026-06-23T20:02:54.232+03:00] - Script Log [Information] 'Wait 2 seconds'
[2026-06-23T20:02:56.235+03:00] - Script Log [Information] 'Finished waiting'
[2026-06-23T20:02:56.237+03:00] - Script Log [Information] 'Application.Stop() of application C:\Users\L-33\Desktop\Work\My Script Location\excel.cs'
[2026-06-23T20:02:56.237+03:00] - Script Log [Information] 'Trying closing main window'
[2026-06-23T20:03:01.252+03:00] - Script Log [Information] '.'
[2026-06-23T20:03:01.254+03:00] - Script Log [Information] 'Killing process'
[2026-06-23T20:03:01.257+03:00] - Script Log [Information] 'Wait 2 seconds'
[2026-06-23T20:03:03.263+03:00] - Script Log [Information] 'Finished waiting'
[2026-06-23T20:03:03.267+03:00] - Script Log [Information] 'Verifying the application has stopped.'
[2026-06-23T20:03:03.905+03:00] - The script ended
```

For more information on using this part of the Recorder, see [Debugging with Script Editor](#).

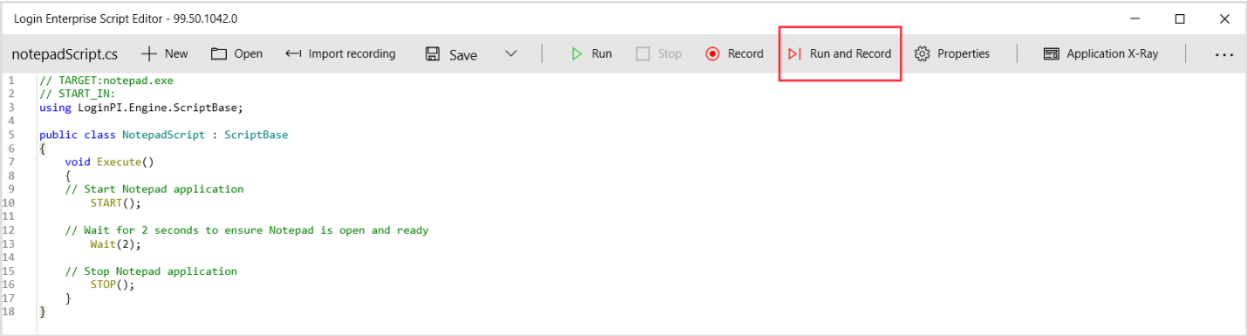
Run and Continue Recording

The **Run and Record** button in the Script Editor allows you to execute an already created script and then automatically opens the Recorder at the end of the script. This is useful if you need to add new steps to an existing script or make adjustments without re-recording everything from the beginning.

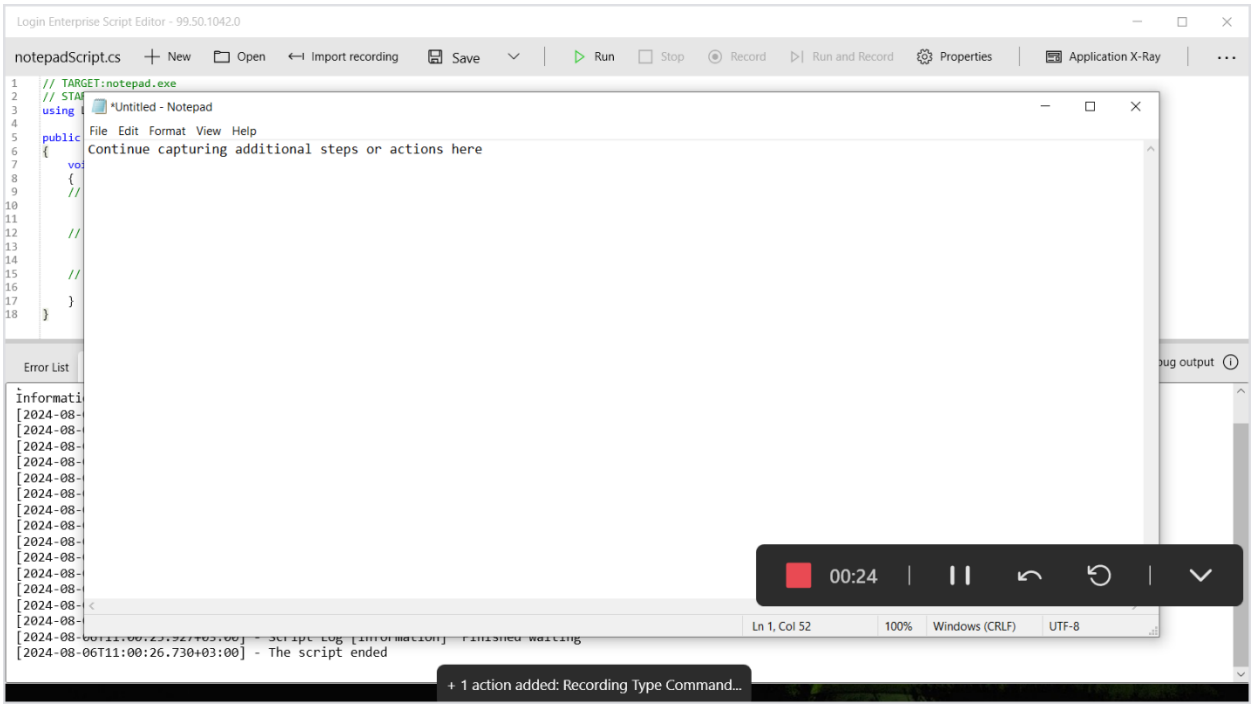
NOTE: If the script fails during execution, the Recorder automatically opens at the point where the failure occurred. You can use this to continue recording and update the script from that step.

How it works

1. **Execute the existing script**: Click **Run and Record**. This will first execute the script that is currently open in the Script Editor.

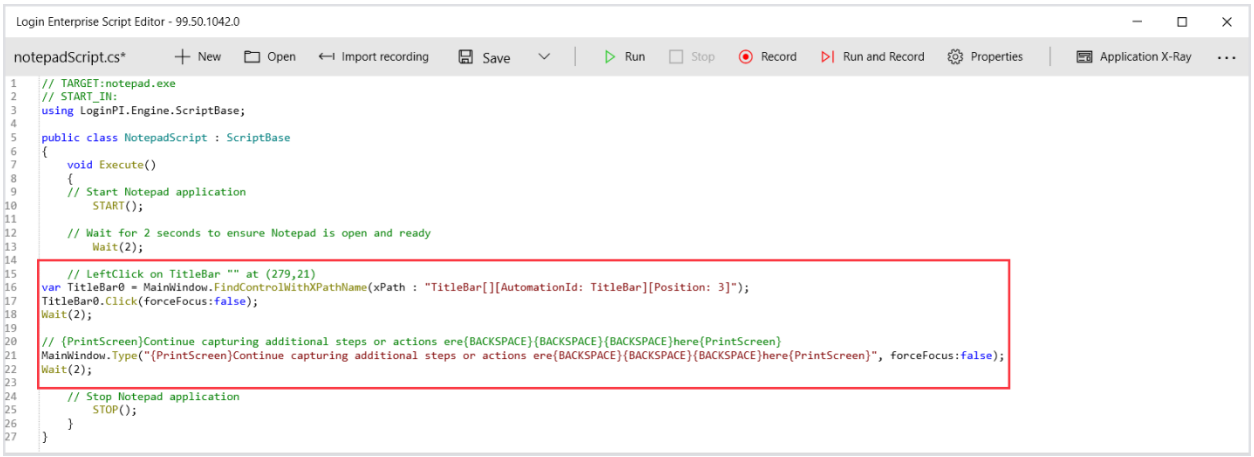


2. **Automatic recording**: Once the script has finished running, the Recorder openi automatically, enabling you to continue capturing additional actions or steps without starting from scratch.



3. Stop the Recorder, click **Generate script**, and copy the script to the clipboard. Then, paste the script into the Script Editor and click **Run**.

NOTE: In this case, the generated script does not include `start` and `stop` commands. You can paste the script into an existing script without modifying the surrounding commands.



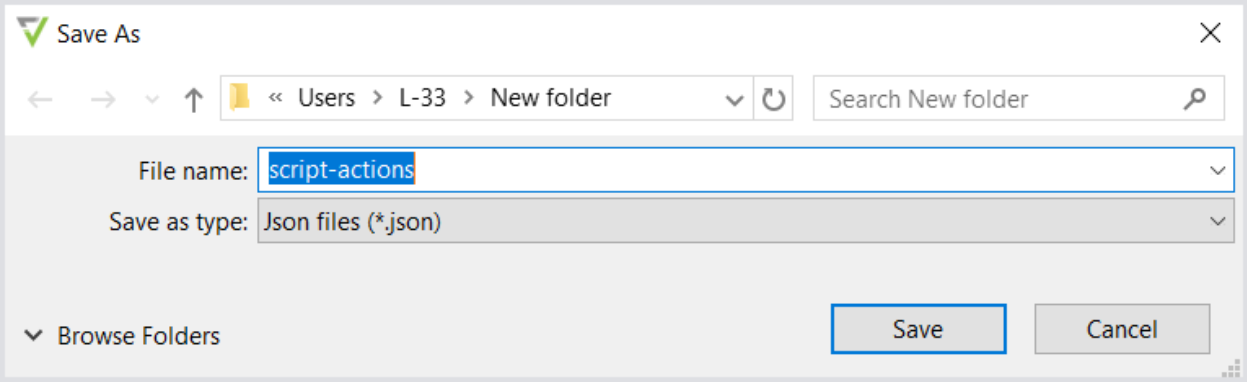
Keeping the Application Open after Recording

Applications are not automatically closed when recording stops.

This behavior is controlled by a configuration setting that enables or disables automatic closing of the application. The setting is currently available only in the ScriptEditor.Config file.

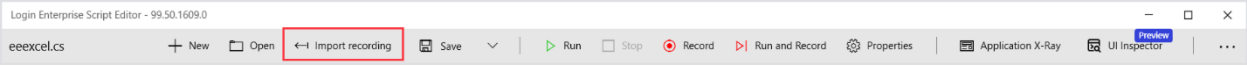
Exporting the Script as JSON

This option provides additional information on script failures, facilitating feedback submission and analysis.



Importing the Script as JSON

Alternatively, if you already have an existing script recording (JSON file), click **Import recording** in the **Script Editor top menu** and proceed with the steps described above.



Restrictions and Limitations

Aspect	Details
Script compatibility	The script generated by Script Recorder can only be run in Login Enterprise v5.10 or higher.
Supported applications	V1 supports only WinApps. Browser and browser-based apps (e.g., Teams, Spotify) are currently out of scope.
Unsupported actions	Actions that were not previously supported in the Engine remain unsupported, e.g., mouse drag and mouse scroll.
Application types with limited functionality	Applications built with Java, Web components, or those that launch separate windows with new executables do not function properly yet.

Script handling	Script Recorder now only generates the script and does not automatically import it into the editor. You must copy the script to the clipboard and paste it manually into the editor. **Important**: The generated script includes `start` and `stop` commands. Make sure to replace the default start and stop commands in the Editor with the commands generated by the Recorder.
Recording scope	Only one application per recording can be captured at a time.
Recorder no longer locks to a single application	Previously, the Recorder was locked to the application you selected at the start of recording. Any actions performed outside that application, including actions in other apps or in pop-up windows, were not recorded. The Recorder no longer locks to a single application. When you start recording, the selected application opens as before, but the Recorder captures actions across all applications. For example, if you start recording in Excel and then switch to Word, your actions in Word are also recorded. **Benefits** • Reduces recording errors caused by pop-ups or applications with multiple windows. • Improves reliability when workflows span more than one application. **NOTE**: While recording, be mindful to perform only the actions you want to capture. Pause or stop the recording if you need to perform unrelated work.

Recommendations

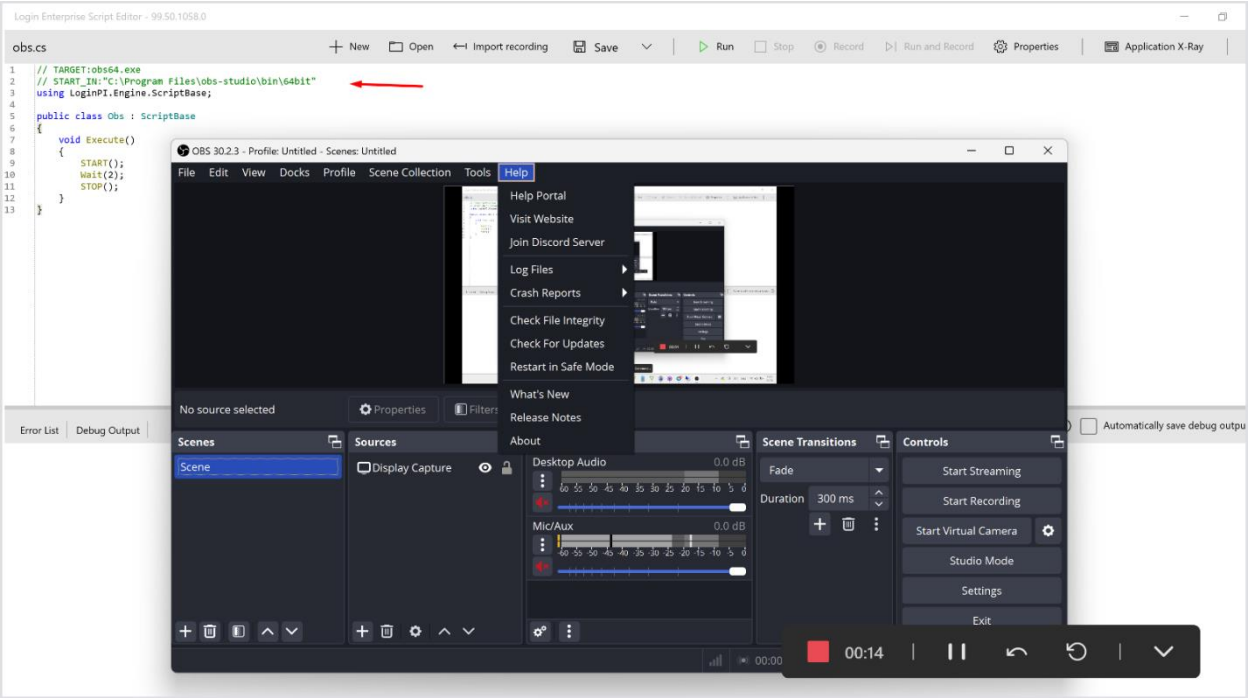
Recommendation	Explanation
Use one monitor during recording	Using multiple monitors with different scaling settings during recording may cause issues. Specifically, when working across two screens with varying scaling configurations, Script Recorder may not function correctly.
Proceed carefully	Wait for the green box to blink over a button or object before proceeding. This visual cue indicates that you can safely move to the next step.

Tutorials

Launching Apps with a START_IN Magic Comment

Script Recorder uses the START_IN magic comment to launch applications, such as OBS Studio, if this comment is present in the script.

The screenshot below shows how the START_IN path appears in the recorded script:



For information on how to use the magic comment, including its formatting, practical examples, and best practices, see [Using // TARGET: Comments for CLI Automation](#).

CLI Parameters and Arguments

Script Recorder (and the Engine) allows you to write arguments and parameters to scripts. You will write your parameters in the // Target: aka the Target application.

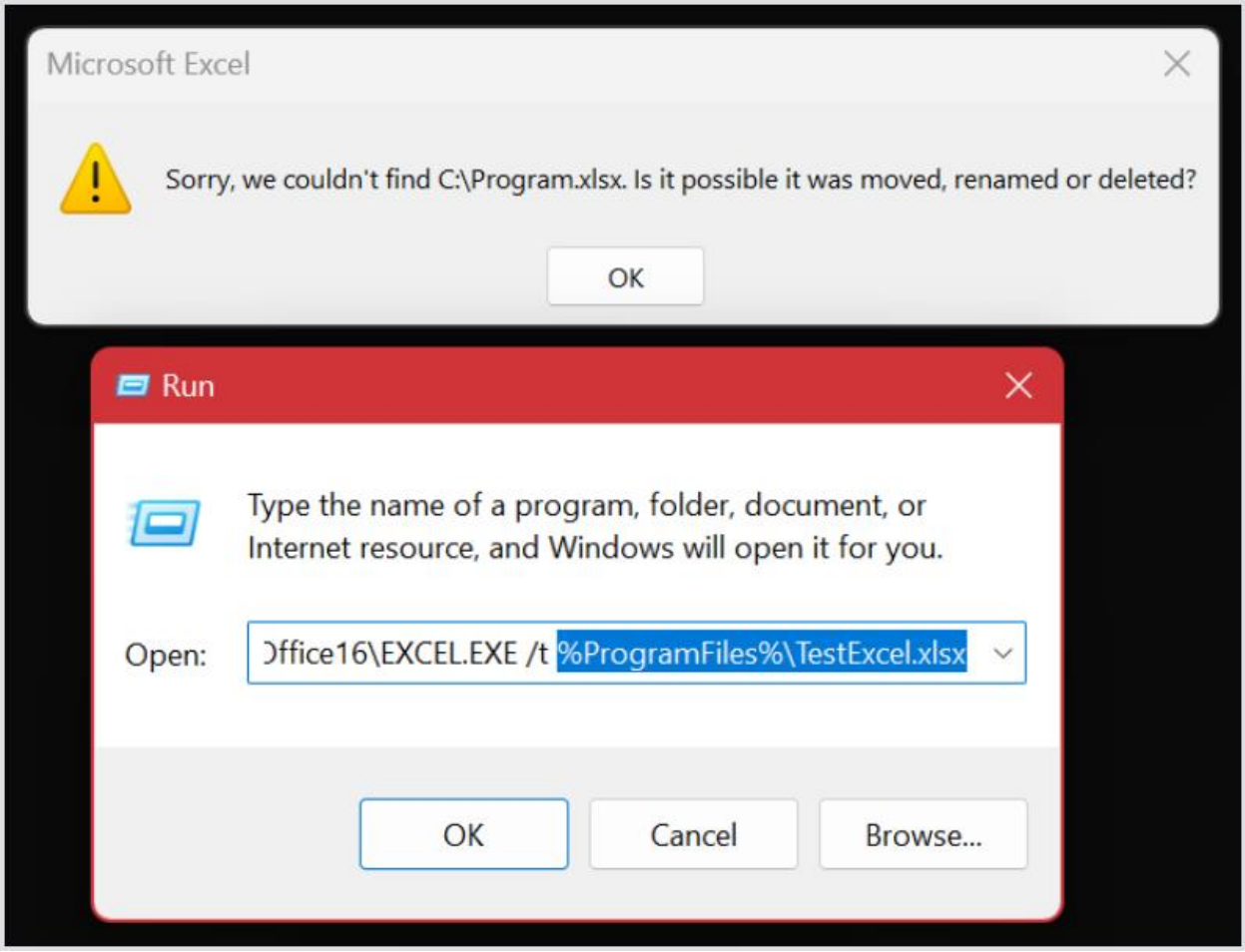
```
// TARGET:%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE /t "%ProgramFiles%\TestExcel.xlsx"
// START_IN:
using LoginPI.Engine.ScriptBase;

public class Excel : ScriptBase
{
    void Execute()
```

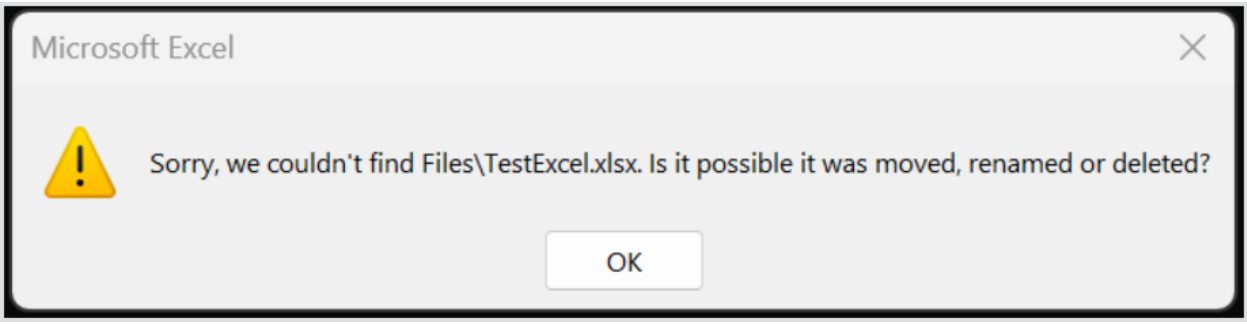
There are a few things to note:

- Script Recorder will allow the use of environment variables, and these will be expanded to the respective values
- It is possible to write in the target application **without quotations** as per legacy behavior in the Engine.
    - While Script Recorder allows the end user to write in the target application without quotations, this goes against standard Windows functionality when starting a process. Windows expects the user to wrap **any** file paths containing spaces **in quotations.** This functionality exists due to legacy behavior. See the following example. The second attempt is the standard behavior.

```
operable program or batch file.

C:\Users\c.valladares>C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE
'C:\Program' is not recognized as an internal or external command,
operable program or batch file.

C:\Users\c.valladares>"C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE"
```

- This functionality will only allow the end user to write down a target application without quotations. If the target application has the following file extensions: **".exe", ".bat", ".cmd", ".vbs", ".wsh".** This is also part of the legacy behavior in the Engine.
    - If you do not provide any of these extensions, the behavior will revert to the original Windows behavior. Thus, the end user should wrap the target application in quotations, or at least, make sure that the provided file path **does not** contain spaces.
    - Arguments provided to Script Recorder expect that any arguments with a file path be wrapped in quotations. **Otherwise**, you just have to ensure that your file path does not contain any spaces. This is consistent with standard Windows behavior.





The previous example attempts to run the following command:

%ProgramFiles%\Microsoft Office\root\Office16\EXCEL.EXE /t %ProgramFiles%\TestExcel.xlsx

This example expands environment variables to:
C:\Program Files\Microsoft Office\root\Office16\EXCEL.EXE /t C:\Program Files\TestExcel.xlsx

- This means that Excel then identifies **C:\Program** as one parameter and **Files\TestExcel.xlsx** as another parameter. Therefore, it fails to find the file
- The solution is to wrap the argument in quotations as "%ProgramFiles%\TestExcel.xlsx"

Wildcards for Class Names, Names, Titles, and Automation IDs

Wildcards are useful when dealing with elements that have dynamic properties in an application.

Why use wildcards?

Wildcards allow for flexible matching of elements when their properties are not fixed. For instance, if elements in an application have names or IDs that change dynamically, using wildcards can help you identify and interact with these elements more effectively.

How wildcards work

- A `*` (asterisk) serves as a wildcard character.
- Placing the `*` at the end of a string, e.g., `classn*` will match all elements that start with the given prefix.
- The `*` can also be placed at the beginning or middle of the string to match elements with varying prefixes or suffixes.

Syntax example

```
{TagGoesHere} : {ClassNameGoesHere} [{NameGoesHere}][AutomationId :
{AutomationIdGoesHere}][Position: {PositionGoesHere}]
```

In this syntax, wildcards can be used in the following elements:

- `{ClassNameGoesHere}`
- `{NameGoesHere}`
- `{AutomationIdGoesHere}`

Practical example

Imagine you're writing a script to interact with Microsoft Word. When you open a new document, the Script Recorder might capture the window title as:

```
var nWindow0 = FindWindowByClassAndName(className: "Window:OpusApp", name:
"Document1 - Word");
```

Here, the title "Document1 - Word" is specific and might not always be the same. If you want your script to work regardless of the specific window title, you can use a wildcard to match any title that ends with " - Word":

```
var nWindow0 = FindWindowByClassAndName(className: "Window:OpusApp", name: "*
- Word");
```

In this example, the asterisk (*) acts as a wildcard character, allowing the script to successfully find any Microsoft Word window where the title ends with " - Word", even if the part before " - Word" changes.

UI Inspector (Preview)

- Overview
- When to Use UI Inspector
- Starting UI Inspector
- Inspecting a UI Element
- UI Inspector Results
  - Element Chain
  - Pane (Chain Details)
  - Interaction
  - Settings
  - Code Preview and XPath Preview
  - Test Button
  - Find Control Button

Overview

UI Inspector is a tool that helps you identify and inspect UI elements in Windows applications. It can be viewed as a newer generation of Application X-Ray, while Application X-Ray remains available for existing use cases.

UI Inspector generates code snippets for selected UI elements/actions, not a full recorded workflow. You can use these code snippets to build or update WinApp scripts. The generated code is supported only for the WinApp script base and is not compatible with Playwright and Selenium scripts.

NOTE: UI Inspector is a preview feature.

When to Use UI Inspector

Use UI Inspector when you need to inspect a specific UI element in a Windows application and generate a WinApp script code snippet for that element. This is useful when a recorded workload does not play back as expected, when a UI element is not detected reliably, or when an application update changes a button, field, pane, or other control used by an existing script.

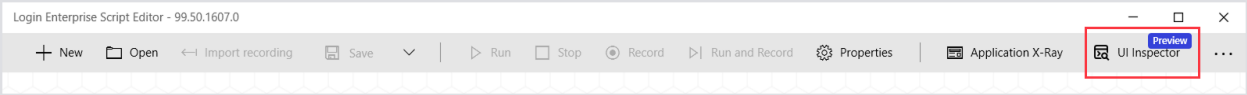
For example, you can open the application manually at the point where the script fails, use UI Inspector to inspect the target element, adjust the element chain or properties, test the selected findBy method, and then copy the generated code snippet into the relevant part of the WinApp script.

UI Inspector can also help you choose between XPath-based detection and information-based detection. XPath detection is often faster and more specific when it works reliably. Information-based detection can be easier to read and adjust, but may take longer. The default option, **XPath**

with find element by information as a fallback, provides a balance between speed and resilience.

Starting UI Inspector

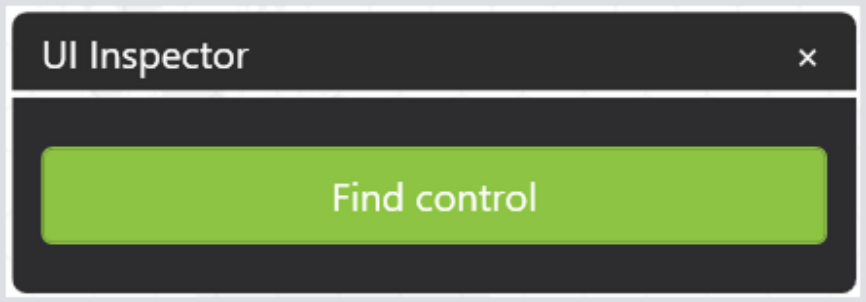
1. Click **UI Inspector** from the top of the Script Editor menu. The option is located at the far right of the menu. UI Inspector is only available for WinApp scripts and is hidden for Playwright and Selenium scripts.



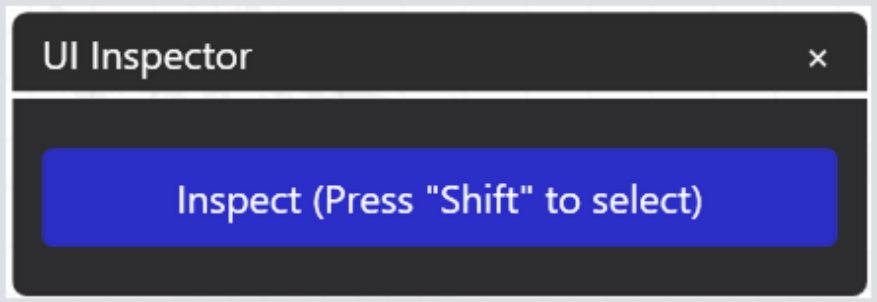
2. The **UI Inspector** window opens. Note that at this step, inspection mode is disabled.

Inspecting a UI Element

1. In the **UI Inspector** window, click **Find Control**:



2. The button changes to **Inspect (Press "Shift" to select)**, indicating that inspection mode is enabled.

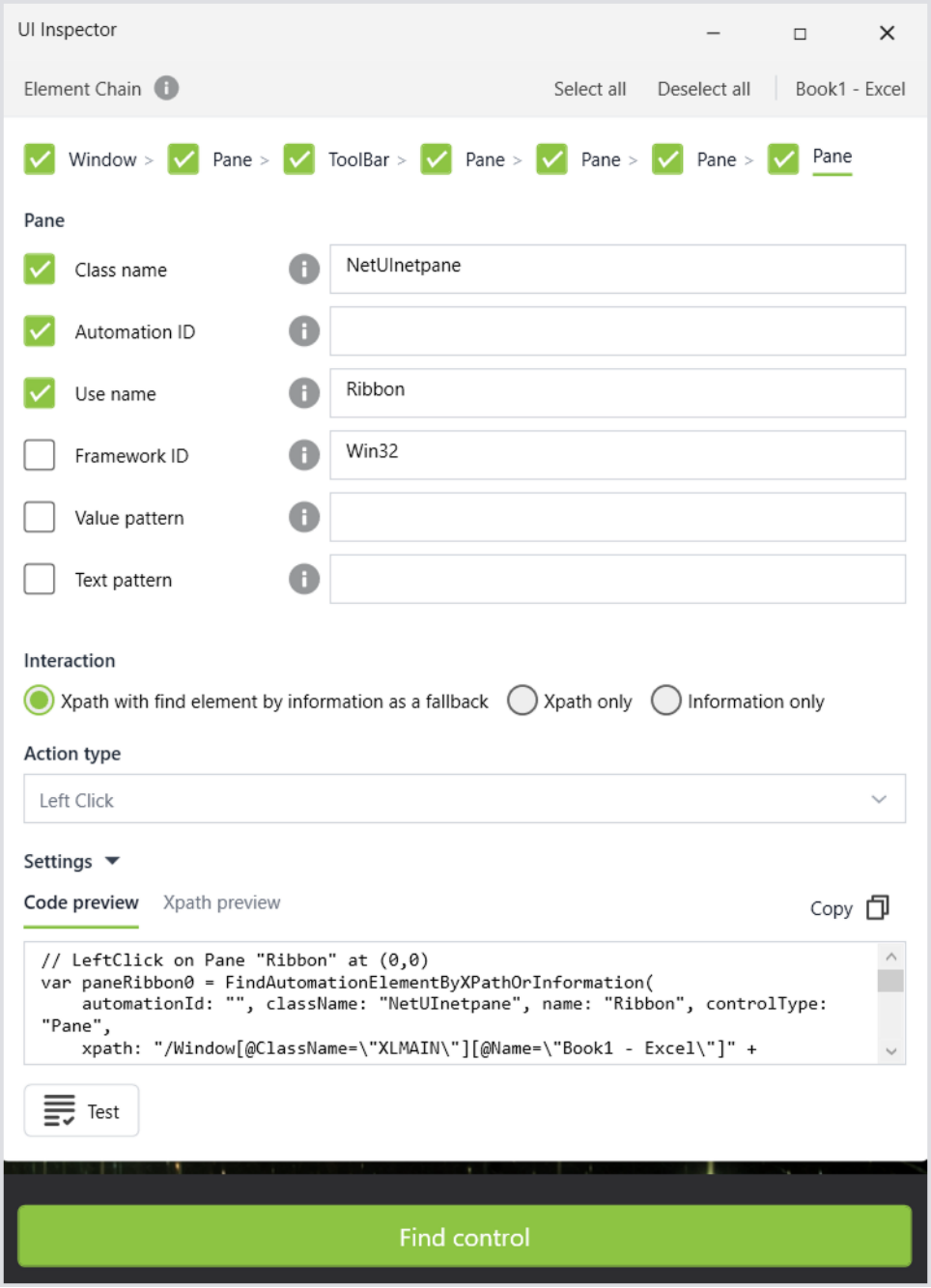


3. Move the pointer over the UI element that you want to inspect. As you move the pointer, UI Inspector highlights the UI element currently under the pointer.
    1. An **orange** border indicates that UI Inspector is locating the element.
    2. A **green** border indicates that the element is ready to be selected.
4. Press **Shift** to select the highlighted UI element. UI Inspector will display the results of the selected UI element.

UI Inspector Results

After inspecting a UI element, the Inspector Result window displays the detected element structure, interaction options, and generated automation output, such as code and XPath.

The window is divided into several sections.

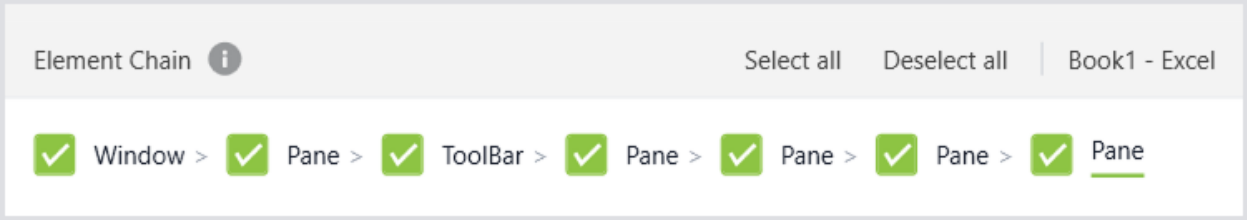


Element Chain

This section defines how the XPath is constructed. By default, all elements in the chain are selected.

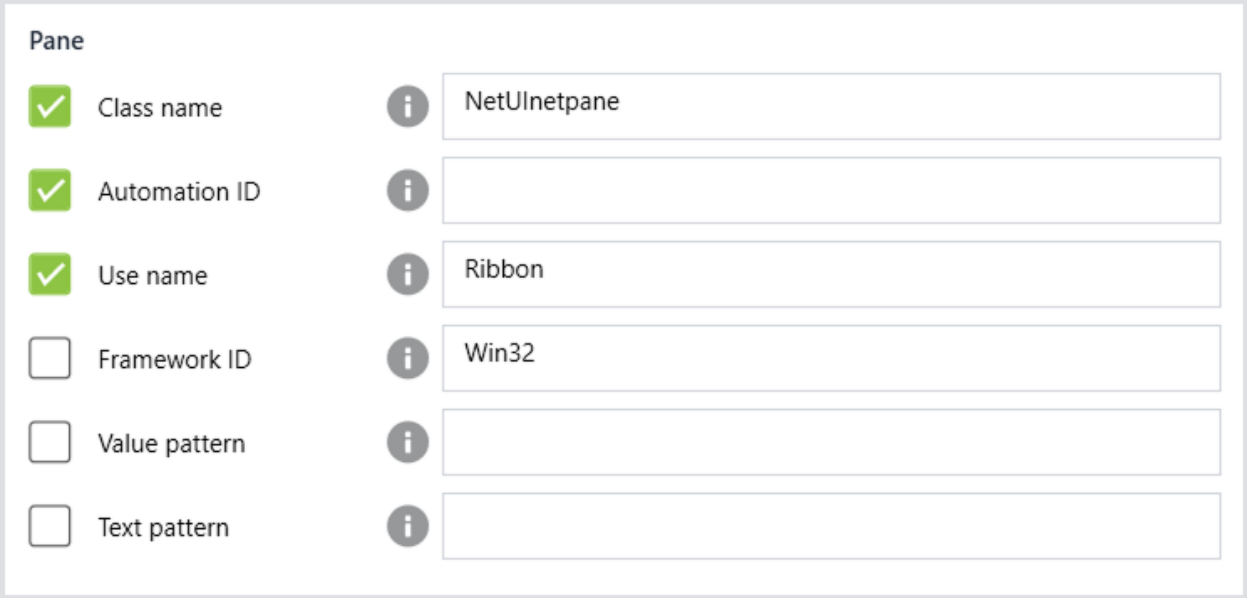
NOTE: You can uncheck individual items to exclude them from the XPath generation. This can help handle dynamic UI structures.

However, excluding elements from XPath generation may affect detection performance or reliability.



Pane (Chain Details)

This section displays detailed information for the selected elements in the chain. By default, the last element in the chain is selected. You can modify parameters of an item in the chain.

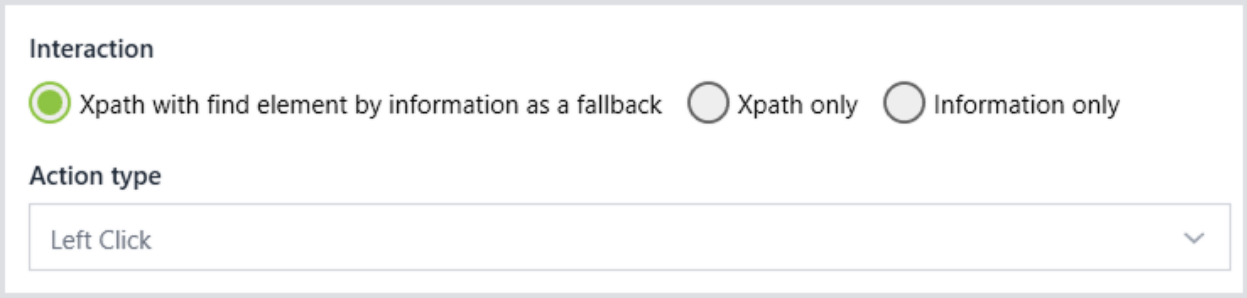


Interaction

This section defines which method and action type you want to use. The following action types are available:

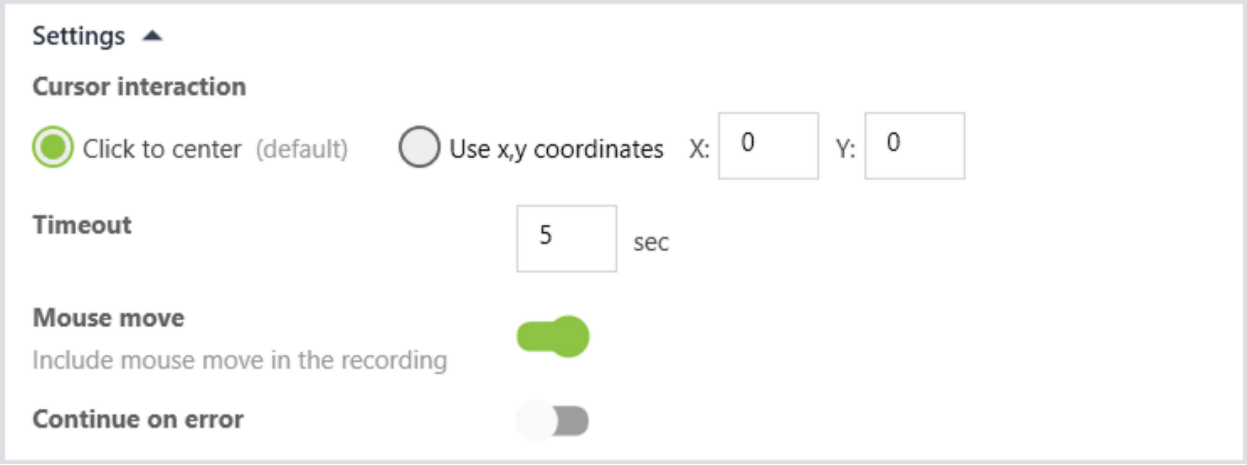
Action	Description
Left Click	Performs a standard click action.
Right Click	Performs a right-click action.
Double Click	Performs a double-click action.

The selected action is reflected in the generated code preview.



Settings

The section is hidden by default and can be expanded. Use it to configure additional options that influence the generated code preview.

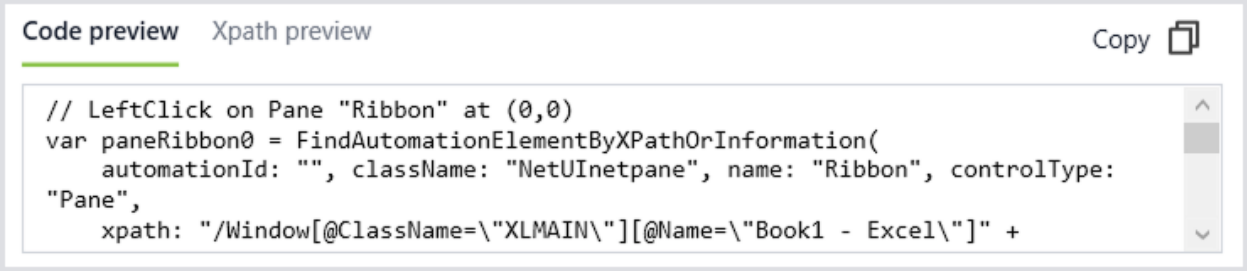


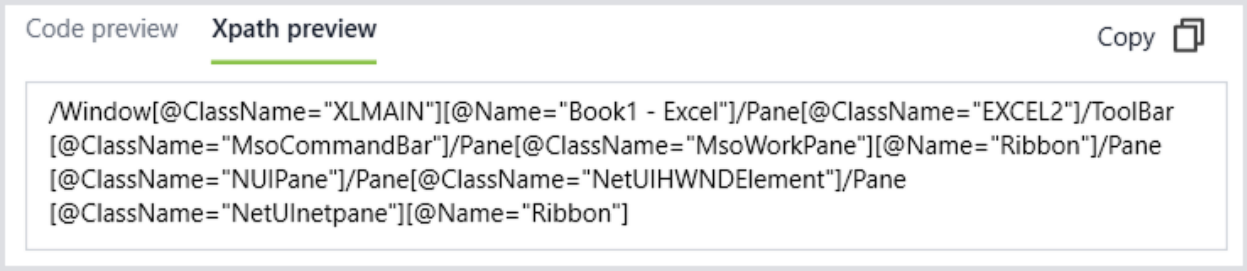
Code Preview and XPath Preview

This section shows the generated output based on the selected chain and interaction settings.

Preview Type	Description

Code Preview	Full script/code snippet of the UI element. Ready to copy and paste into a script.
XPath Preview	Raw XPath expression of the UI element (Not a standalone code). Use it within the full generated code snippet.

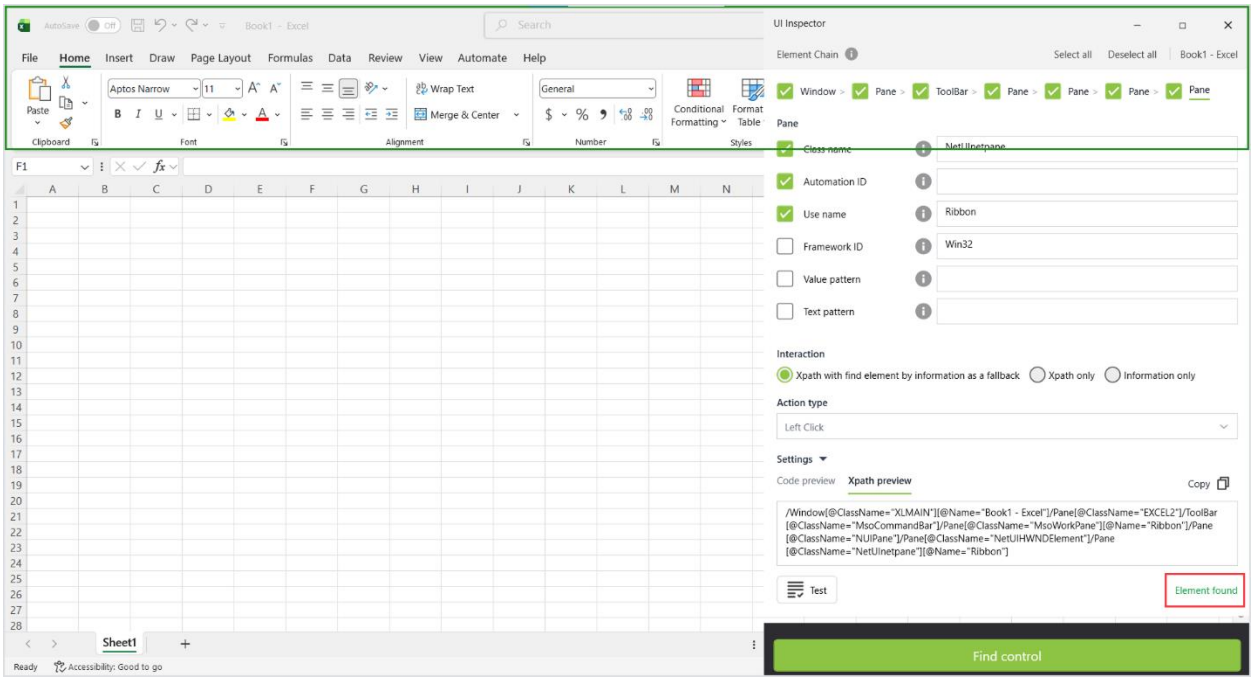




Test Button

The **Test** button validates the selected findBy method. It only tests element detection (findBy logic). It does not execute the interaction (e.g., click)

NOTE: The target application must be open. The inspected element must be visible in the UI



Find Control Button

This button clears all current inspection data and initiates a new UI element inspection.

Understanding WinApp Recorder Implementation (FlaUI)

- Overview
- Inventory of Methods (Generated by Script Recorder)
  - FindAutomationElement Methods
- XPath Query Changes
- Appendix
  - FindAll Methods

Overview

To improve the robustness, maintainability, and flexibility of Script Recorder, the following changes took place in v6.5:

- The existing automation library has been replaced with FlaUI.
- XPath matching has been made more flexible. Previously, the full sequence of element descriptors had to be provided, and every part had to match exactly. Now you can omit steps or use wildcards, making element lookup less rigid and easier to maintain.
- A new `IAutomationElement` abstraction replaces the former `IAutomatableComponent`, reducing bloat and decoupling it from the Engine.
- A more stable foundation for creating reliable and reproducible automation scripts has been established.
- Script Recorder can now record multiple applications at once, regardless of the specified target application. For example, if you set Calculator as a target when creating a new application script, you can launch and record Calculator. However, you can also record any other window present on the desktop, such as Microsoft Word, etc.

This page describes the inventory of methods used by Script Recorder and XPath changes. For information on how to use WinApp Recorder, see Using the WinApp Recorder.

Inventory of Methods (Generated by Script Recorder)

FindAutomationElement Methods

New `FindAutomationElement` methods have been added to the Recorder. These methods allow recording across multiple applications, rather than being limited to a single target application.

The `MainWindow` is still populated with the target application window for compatibility with existing methods.

NOTE: When using `FindByInformation` or `FindByXpathOrInformation` without providing window information, the search traverses all windows on the desktop, which may result in slower performance in some scenarios.

Method: FindAutomationElementByXPath(…)	
Description	Finds an`AutomationElement` that matches the specified XPath expression. The entire element sequence must match for the element to be considered found. You can omit ancestor or window information to make the search more flexible, or include it to improve performance.
Returns	An `AutomationElement` that implements the `IAutomationElement` interface.
XPath Language Changes	There is a change in the XPath syntax between the old and new methods. For more details, see the XPath Query Changes.
Parameters:	
XPath	The XPath query used to locate the automation element. Must match exactly one element for a successful result.
timeout	The maximum time, in milliseconds, to wait for the element to be found. Must be greater than or equal to zero.

A `FindAllVariant` variant of the method can also be generated manually. For details, see Appendix.

Method: FindAutomationElementByInformation(…)

Description	Finds an `AutomationElement` by its properties, without using information about ancestor elements.
	
	This method is useful in scenarios where the element hierarchy may change frequently. It provides the most flexible search approach; however, this flexibility may result in slower performance. A new optimization improves performance by splitting the element search into two steps:
	
	1. First, it locates the window via XPath.
	2. Then, it finds the desired element using `FindByInformation`, starting from that window, substantially narrowing the search scope.
	
	Example (Before optimization):
	
	var buttonNine0 = **Find**AutomationElement**ByInformation**("num9Button", "Button", "Nine", "Button", 5, continueOnError:false);
	
	Example (After optimization):
	
	var buttonNine0 = **FindAutomationElementByXPath**("/**Window**[@ClassName=\"ApplicationFrameWindow\"][@Name=\"Calculator\"]", 5, continueOnError:false).**Find**AutomationElement**ByInformation**("num9Button", "Button", "Nine", "Button", 5, continueOnError:false);
	
	This optimization is configurable in the Recorder UI. For details, see Using the WinApp Recorder.

Returns	A list of `AutomationElement` objects that implement the `IAutomationElement` interface.
Remarks	
Search Root	The old Engine required window information to locate elements. The new method performs searches from the Desktop as the root. Because ancestor and window information are omitted, the search covers all elements on the Desktop. As an optimization, you can configure the element search to first attempt to locate the window.
Parameters	
AutomationId	The automation ID of the element to find. Can be `null` or empty to ignore this property.
className	The element's class name to find. Can be `null` or empty to ignore this property.

name	The name of the element to find. Can be `null` or empty to ignore this property.
Framework ID	The underlying technology used to build the UI element (for example, WPF or Win32). **NOTE**: Only used by find by information methods.
Value pattern	The current value of elements that support the Value pattern (for example, text in a text box or a combo box selection). **NOTE**: Only used by find by information methods.
Text pattern	The full text content of elements that support the Text pattern (for example, rich text or document controls). **NOTE**: Only used by find by information methods.
controlType	The control type of the element to find. Can be `null` or empty to ignore this property.
timeout	The maximum time, in milliseconds, to wait for the element to be found. Must be greater than or equal to zero.

A `FindAll` variant of the method can also be generated manually. For details, see Appendix.

Method: FindAutomationElementByXPathOrInformation(…)	
Description	Finds an automation element by using the specified XPath expression. If no match is found, the method attempts to locate the element by matching automation properties such as automation ID, class name, name, and control type. This approach first uses a strict, performant search strategy (findByXpath) and then falls back to a more flexible property-based search (findByInformation) if metadata in the XPath chain has changed or is no longer reliable.
Returns	`AutomationElement` that implements the `IAutomationElement` contract
Remarks:	

Search Root	`FindByInformation` previously required window information to locate elements. The new method performs searches from the Desktop as the root, so you do not need to provide window information. This increases flexibility, but because it traverses all windows on the Desktop, performance may be slower in some scenarios. `FindByXpath`, by contrast, follows a specific set of breadcrumbs from the Desktop, making it considerably faster. In Win365 scenarios, window metadata (such as title or class name) can change between recording and execution. For example, if the document moves or the window title differs. Hiding window information ensures that scripts continue to work even when window details vary. For instance, if a script is recorded on one document window but executed when only a different window is open, it will succeed when window information is omitted.
XPath Language Changes	There is a change in the XPath syntax between the old and new methods. For details, see XPath Query Changes.
Parameters:	
XPath	The XPath expression used to locate the automation element. If `null` or empty, the search uses the provided automation properties instead.
AutomationId	The automation ID of the element to find. Used if the XPath expression is not successful. Can be `null` or empty to ignore this property.
className	The element's class name to find. Used if the XPath expression is not successful. Can be `null` or empty to ignore this property.
Framework ID	The underlying technology used to build the UI element (for example, WPF or Win32). **NOTE**: Only used by find by information methods.
Value pattern	The current value of elements that support the Value pattern (for example, text in a text box or a combo box selection). **NOTE**: Only used by find by information methods.
Text pattern	The full text content of elements that support the Text pattern (for example, rich text or document controls). **NOTE**: Only used by find by information methods.
name	The name of the element to find. Used if the XPath expression is not successful. Can be `null` or empty to ignore this property.

controlType	The control type of the element to find. Used if the XPath expression is not successful. Can be `null` or empty to ignore this property.
timeout	The maximum time, in milliseconds, to wait for the element to be found. Must be greater than or equal to zero.

This method provides a `FindAll` variant that returns a list of elements instead of a single element. Unlike the single-element version, it does not throw an exception when multiple elements are found.

NOTE: A `FindAll` command is not currently generated automatically by Script Recorder. Script Recorder uses `Find[single]ElementByXXXX`, a method only available for manual use. For details, see Appendix.

In addition to element search methods, the interface provides the following methods. These methods cover user interactions within a window environment, such as clicks and keyboard input.

Method Name	Description
Click	Performs a click at the specified element's location (as defined by its bounding rectangle). The mouse pointer may not move directly over the element, but the click action is still executed on the requested element.
Move Mouse To Center	Performs the mouse move from its initial location to the center of the specified element.
Move Mouse Relative To Coordinates	Performs the mouse move from its initial location to the specified coordinates relative to the current element's location. In other words, the movement is not based on the global pixel location, but on the pixel location within the element's bounding rectangle. These coordinates are captured during recording.
Right Click	Performs a right-click at the specified element's location (as defined by its bounding rectangle). The mouse pointer may not move directly over the element, but the click action is still executed on the requested element.
Double Click	Performs a double-click at the specified element's location (as defined by its bounding rectangle). The mouse pointer may not move directly over the element, but the click action is still executed on the requested element. The "delta" (the amount of time between clicks) used to register a double-click is based on the window's default values.
GetRootWindow	Attempts to fetch the containing top-level window for the current element. This allows access to window-specific methods, such as `Close()`.

AsWindow	Attempts to convert the current element to a window. The conversion fails if the element does not have a control type of `Window`.

XPath Query Changes

Script Recorder has transitioned to standard XPath-style syntax. In the new format, introduced in v6.5:

- The ControlType is used as the XPath node (tag)
- All other metadata is expressed as named attributes

/MenuBar/MenuItem[@Name='SpecificName'][@AutomationId : '123'][@ClassName='ClassName']

The standardized XPath query language makes each property explicit, readable, and independently matchable. It allows you to perform complex queries and leverage built-in XPath functions to create flexible and resilient selectors.

/MenuBar/MenuItem[@Name='SpecificName'][@AutomationId : '123'][Contains(@ClassName, 'Name')]

or

/MenuBar/MenuItem[@Name='SpecificName'][@AutomationId : '123'][Starts-with(@ClassName, 'Name')]

You can also omit requirements from an element search in two ways: by leaving metadata fields blank or by omitting specific metadata from the query altogether.

Blank metadata: /RandomWindow[@Class, " "]

Omitted metadata: /RandomWindow_____

Finally, the XPath language allows you to exclude entire chains of elements and to search within a specific subsection of elements using the following syntax:

Original Hierarchy:

```
/Window/Pane/Pane/Pane/Pane/Pane/Pane/Pane/Pane/Pane/Pane/Document/Group
/Button[@Name=\"Sign in\"]
```

Trimmed Hierarchy:

```
//Group/Button[@Name=\"Sign in\"]
```

NOTE: The first tags don't enforce any metadata, and Button enforces the Name value to be "Sign in".

An element search is considered successful when a sequence of elements includes both:

1. A group node
2. A button node with the Name property set to `"Sign In"`

These capabilities provide the following benefits:

- Partial matches for dynamic values
- More tolerant element selection
- Fewer failures caused by minor UI changes

Appendix

FindAll Methods

The following are methods that are not generated by Script Recorder, but can be performed manually.

Method: FindAllAutomationElementByXPath	
Description	Finds all `AutomationElement` objects that match the specified property criteria.
Returns	A list of `AutomationElement` objects that implement the `IAutomationElement` interface.
Method: FindAllAutomationElementByInformation(…)	
Description	Finds all `AutomationElement` objects that match the specified property criteria.
Returns	A list of `AutomationElement` objects that implement the `IAutomationElement` interface.
Method: FindAllAutomationElementByXPathOrInformation(…)	
Description	Finds all automation elements that match the specified XPath expression. If no elements are found, the method attempts to locate matches by automation properties such as automation ID, class name, name, and control type. Use this method when you need to interact with all elements that meet the specified criteria.
Returns	A list of `AutomationElement` objects that implement the `IAutomationElement` interface.

Web Recorder

- Overview
- Getting Started
- Creating a New Application
- Starting the Recording: Recording the Script
- Stopping the Recording: ScriptResultWindow
- Settings
  - General
  - Action
    - Editing an Action Name
    - Selector
      - Filtering During Recording
      - Element Resolution During Playback
    - Inner Text
- Running the Script
- Exporting the Script as JSON
- Importing the Script as JSON
- Restrictions and Limitations
- Recommendations
- Tutorials
  - Using Browser Arguments via a Magic Comment

Overview

The Web Recorder v1 is a tool integrated into the Script Editor to capture user interactions within web applications. Web scripts recorded with the new Web Recorder use Playwright. This version supports a subset of actions and is optimized for browser-based apps. Some features are still under development, and limitations are described in the related section.

- For general information on the Script Recorder and related use cases, see Using the Script Recorder.
- For an overview of Playwright functions, the specific commands used to automate and interact with web applications using Web Recorder v1, see Web Recorder Functions (Playwright). In most cases, you do not need to work with those functions directly if the recording plays back successfully as captured. That page is intended primarily as an advanced reference for users who need to review, adjust, or extend recorded functions.

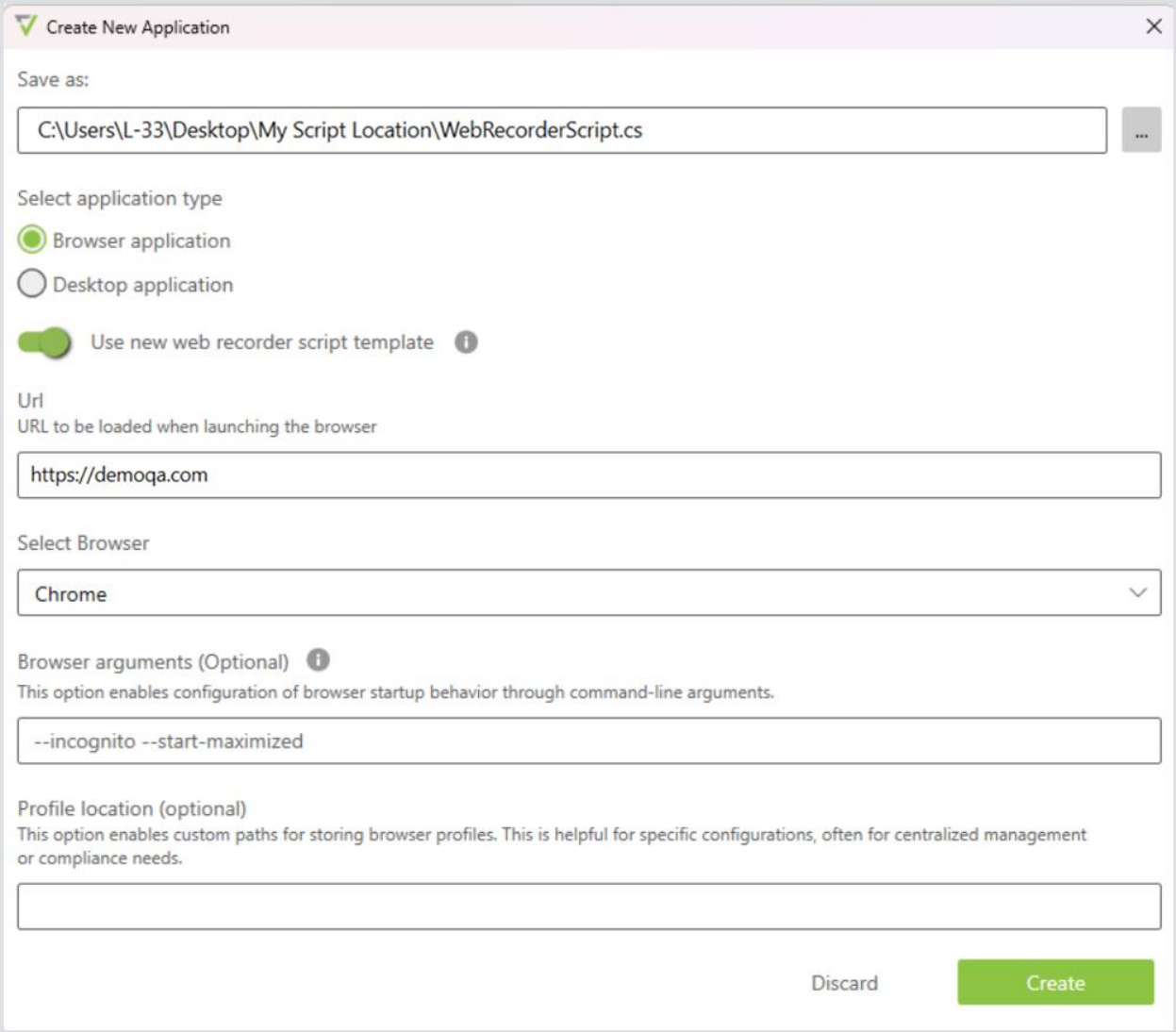
Getting Started

Download the latest Script Editor package containing the Web Recorder v1 from your admin or setup email. For details, see Downloading and installing the Script Recorder.

Creating a New Application

1. In **Login Enterprise Script Editor**, click **Create a new Login Enterprise Application**.

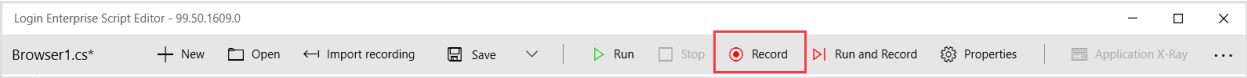
2. Click the three-dot icon to the right of **Save as**, choose a folder for the script, and enter a file name.
3. In the **Select application type**, select **Browser application**.
4. Next to **Use new web recorder script template**, enable the toggle to use Web Recorder, which creates the web script in Playwright format.
5. In the **URL**, enter the start page of your web application.
6. In the **Browser**, select your preferred browser, e.g., Chrome.
7. In **Browser arguments** (Optional), configure how the browser starts by entering command-line arguments, if needed. For example, you can add flags to enable specific features or disable extensions. For more information, see Using Browser Arguments via a Magic Comment.
8. In **Profile location** (Optional), enter a custom path for storing the browser profile if your environment requires centralized management or compliance-related configurations. Otherwise, leave this field blank.
9. Click **Create** to generate the script template. Once the template is created, the **Record** button appears; click it to start the new recording session.



NOTE: Web Recorder uses the default browser installed on your machine.

Starting the Recording: Recording the Script

In the **Script Editor top menu**, click **Record** to start the recording.



When the recording starts, a player opens where you can perform actions.

The web page must be DOM-ready before the Recorder can highlight UI elements. Once it is, the Recorder draws green borders around the elements so you can see which ones you're recording.

Web Recorder supports the following actions:

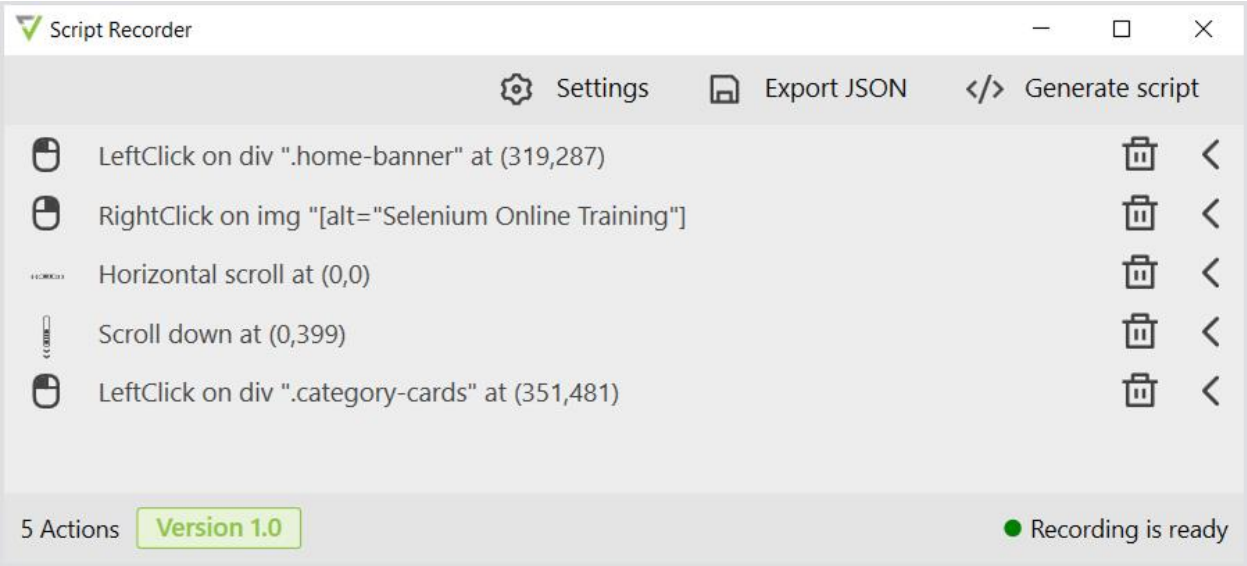
- Mouse scroll
- Left click
- Right-click (*Browser context menu clicks are not supported)*
- Double-click
- Typing and special keys (Enter, arrows, etc.)
- Hover actions using the hotkey **Alt + H**
- Browser JavaScript popups (alert, confirm, prompt)
- Interactions within iFrames
- Browser arguments

When hover capture is enabled, all hover events are recorded until you disable it. Use this feature carefully, as there are no visual indicators when it is active. Hovering over an element before enabling the hot key will **not** capture it; you must move away and back.

You can adjust waits and timeouts after the recording is complete and before generating the script. By default, the recorder applies a 2-second wait and a 5-second timeout to each step. You can change these default values in the General Settings or modify them per action.

Stopping the Recording: ScriptResultWindow

When you stop the recording, the ScriptResultWindow shows an overview of all recorded steps (before generating the script, right after the recording was stopped).



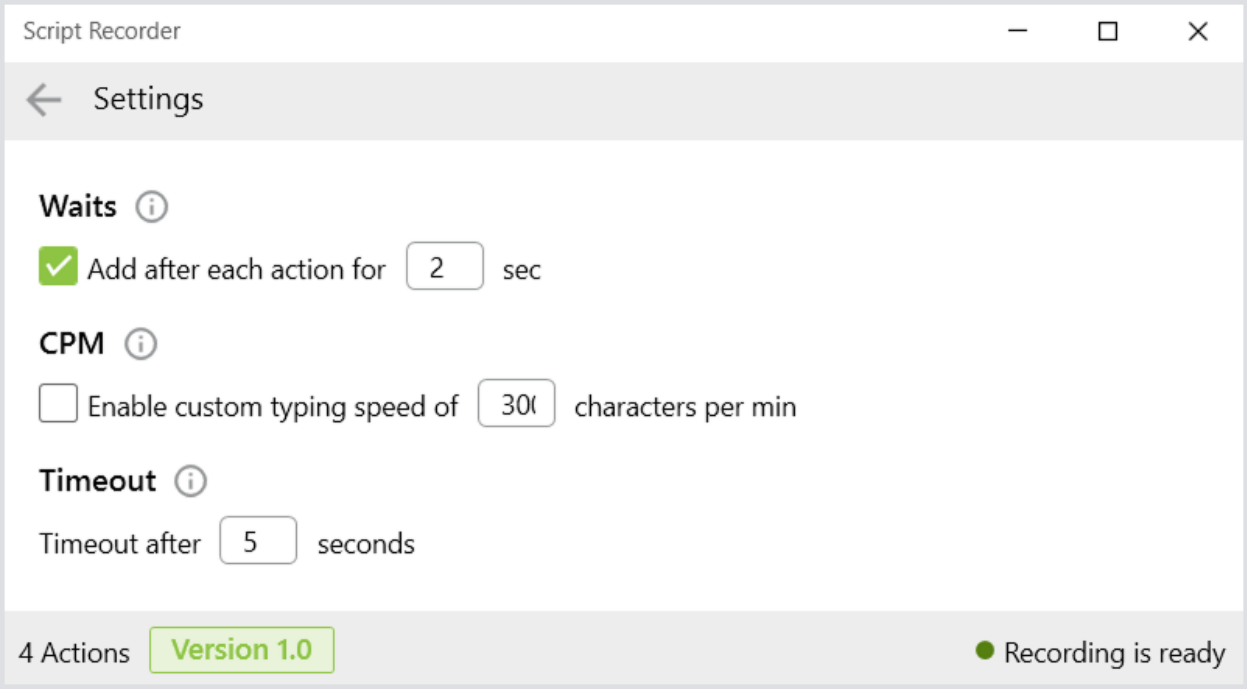
In the Overview screen, you can adjust Settings, export the script as JSON, and generate the script to run it.

Settings

General

NOTE: Changes made in General settings apply to all action settings.

Option	Description
Waits	Adds a pause after each recorded step in the script.
CPM (Characters Per Minute)	Adjusts typing speed. If you leave the checkbox unselected, the default value will be used. If you select the checkbox, the CPM value will appear in the script. The default is 300 characters per minute.
Timeout	Specifies how long the Recorder repeatedly attempts to find the target element before the action fails. If the element is not found within the timeout, the action fails. The default is 5 seconds.



Action

NOTE: Changes made in Action settings apply only to the selected action.

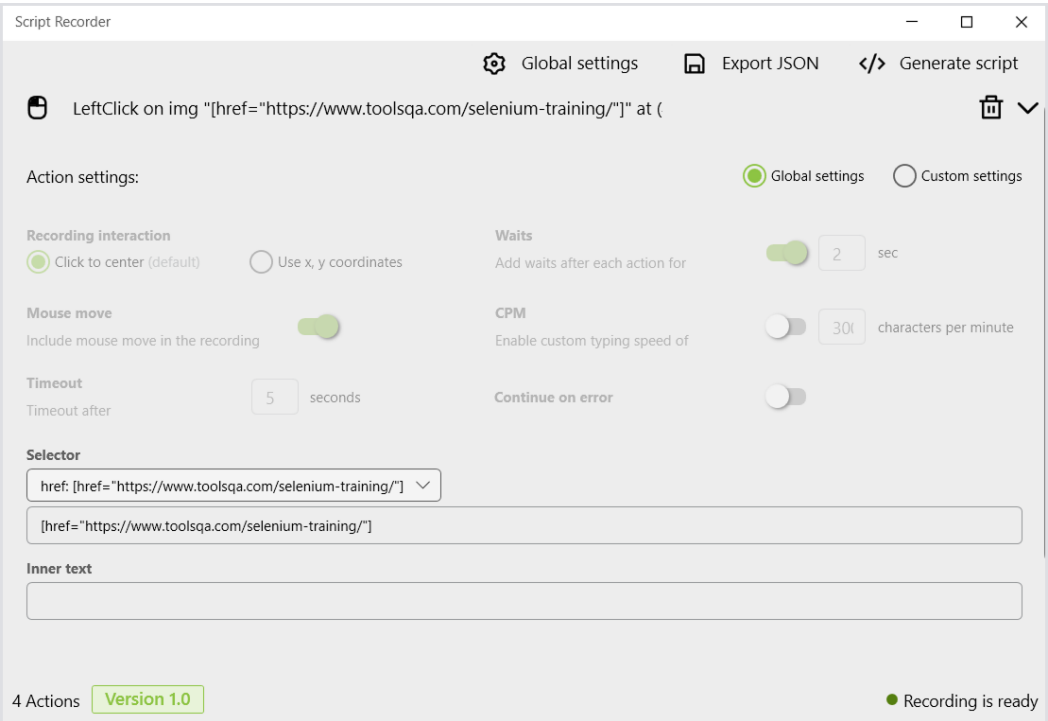
The following settings are displayed in the UI but aren't currently supported: Recording interaction, Mouse move, and Continue on error.

Option	Description

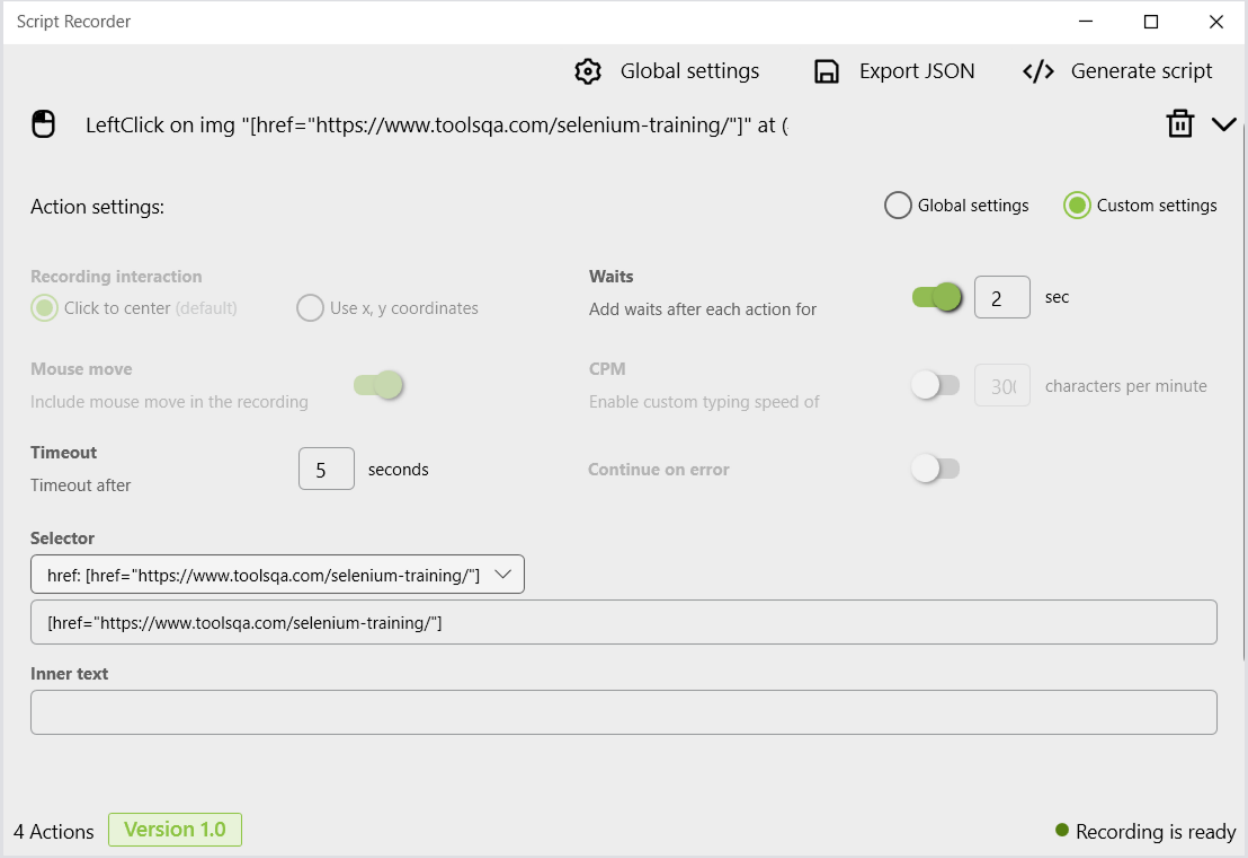
Waits	Adds a pause after each recorded step in the script. Mainly used to make script execution more realistic. Avoid adding waits between timed (Timer) actions if they would artificially increase the recorded measurement.
CPM (Characters Per Minute)	Adjusts typing speed. If you leave the toggle unselected, the default value will be used. If you select the toggle, the CPM value will appear in the script. The default is 300 characters per minute.
Timeout	Specifies how long the Recorder repeatedly attempts to find the target element before the action fails. If the element is not found within the timeout, the action fails. The default is 5 seconds.
Selector	Lets you choose and test a selector on recorded actions. For details, see Selector. **Note**: Selector testing is available while the recording is paused, because the browser page must still be attached. If the recording is stopped, the browser is closed, and selector testing will not work.
Inner text	Optional inner text filter. It is passed to Playwright as 'innerText' and used by the Engine's text-narrowing cascade. To learn more, see Inner Text.

When you open an action step, the settings are shown automatically. Action steps can use global settings or custom settings:

- **Global settings**: The action step uses the global settings. The settings are read-only and can't be changed in the action step.



- **Custom settings**: The action step uses custom settings. Select this option to make the fields available for editing in the action step.



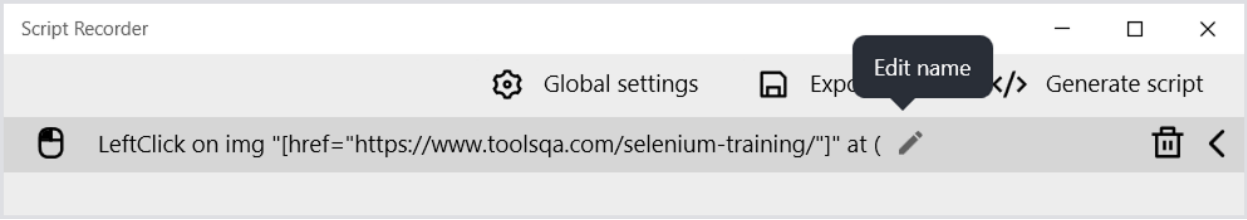
NOTE: Selector and Inner Text parameters are always available for editing and can be configured for both Global and Custom settings. To learn more, see Selector and Inner Text.

Editing an Action Name

You can change the name of each action to make it more descriptive. The action name is shown in the script as a comment before the corresponding step.

To edit an action name:

1. Hover over the action.
2. Select the pencil icon.



3. Enter a new name.
4. Click **Submit**.

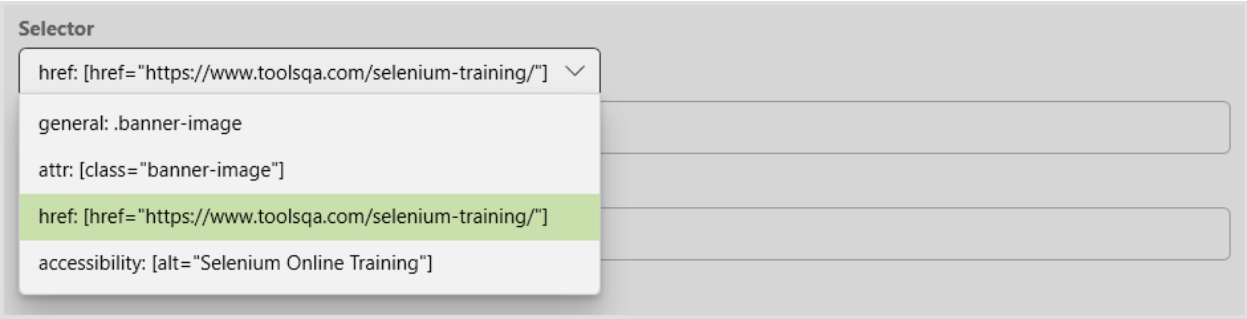
Selector

When you record actions, the Recorder detects the underlying DOM structure and generates selector candidates to identify each web element.

The Recorder checks these candidates against the page layout and automatically assigns the first option that matches exactly one element, using this priority order:

1. **test-id:** Dedicated automation attributes (like `data-testid`). *Most stable.*
2. **id:** The element's unique HTML ID.
3. **form:** Form-specific attributes (`name`, `placeholder`, or `for`).
4. **href:** The hyperlink destination URL (links only).
5. **accessibility:** Labels used by assistive tech (`aria-label`, `alt`, or `title`).
6. **general:** The shortest unique CSS path.
7. **attr:** CSS paths using generic attributes. *Most brittle; used as a last resort.*

You can choose or override the selector while the recording is paused, then test whether the selected selector resolves to the intended element.



NOTE: If a higher candidate matches multiple elements, the Recorder skips it and moves down the list. If no single selector is unique, the element's innerText is used as a tiebreaker.

Filtering During Recording

The Recorder automatically discards unstable data during recording, including session-randomized IDs (such as `:r1:`), state classes (such as `-hover` or `-active`), and direct clicks on icons inside buttons (the Recorder automatically targets the parent button instead).

Element Resolution During Playback

During test playback or when using the **Test** button, the Engine uses a fallback chain to locate the element, stopping at the first step that finds a single match:

1. **Inner text alone:** If the text is unique, the selector is ignored entirely. The Test button follows the same resolution logic as playback, so the test result should match what the generated script will do.
2. **Selector alone:** Used if the text matches multiple elements.
3. **Selector + inner text combined:** Used if steps 1 and 2 are both ambiguous.
4. **Partial selector + inner text:** The final fallback strategy.

Inner Text

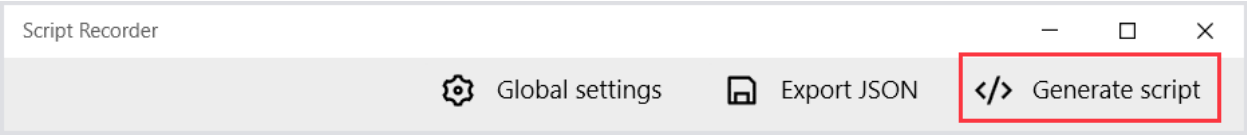
The `innerText` refers to the text content within a web element. The Engine uses it during playback in the following ways:

- **Recording tiebreaker:** If no selector candidate is unique on its own (a candidate matches two or more elements), the Engine automatically uses the element's `innerText` as a tiebreaker to find a unique match.
- **Playback step 1:** During test execution or when you click the **Test** button, the Engine evaluates the `innerText` alone first. If the text is completely unique on the page, the Engine matches the element immediately and does not evaluate the selector.

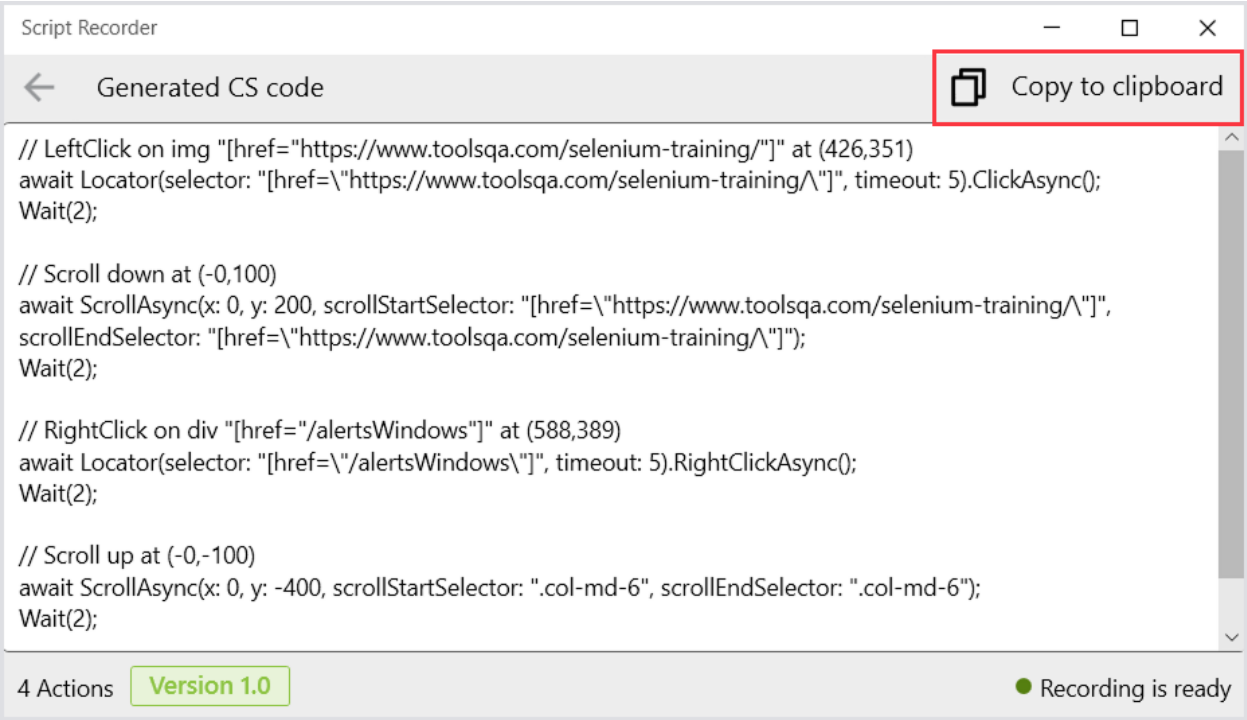
NOTE: The Engine applies the combination of a selector and `innerText` (Step 3 of playback) only when the text alone and the selector alone are both ambiguous.

Running the Script

1. In the **ScriptResultWindow**, click **Generate script**.

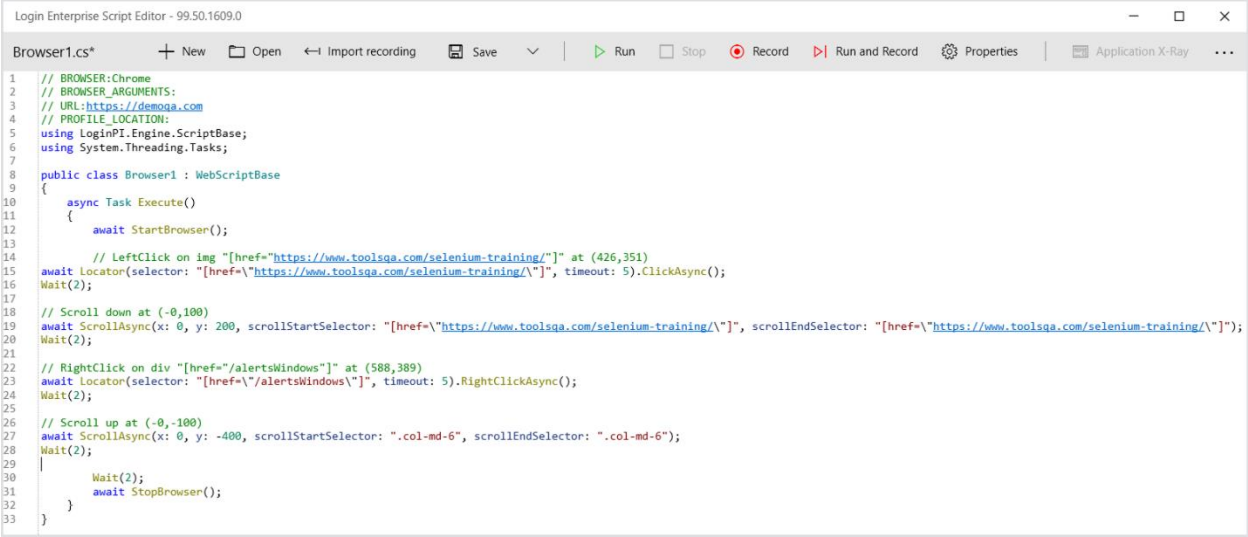


2. Copy the generated script to the clipboard:

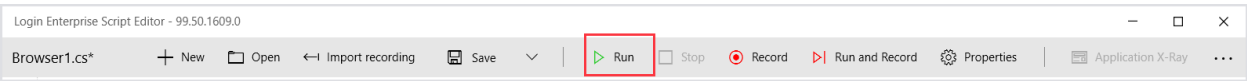


NOTE: Script Recorder doesn't support notifications, such as "Copied to the clipboard." So, once you copy the script to the clipboard, it's saved; the Recorder is just not letting you know. You can now proceed and paste the script into the Editor.

3. Paste the copied script into the Script Editor:



4. In the **Script Editor top menu**, click **Run** to run the script.

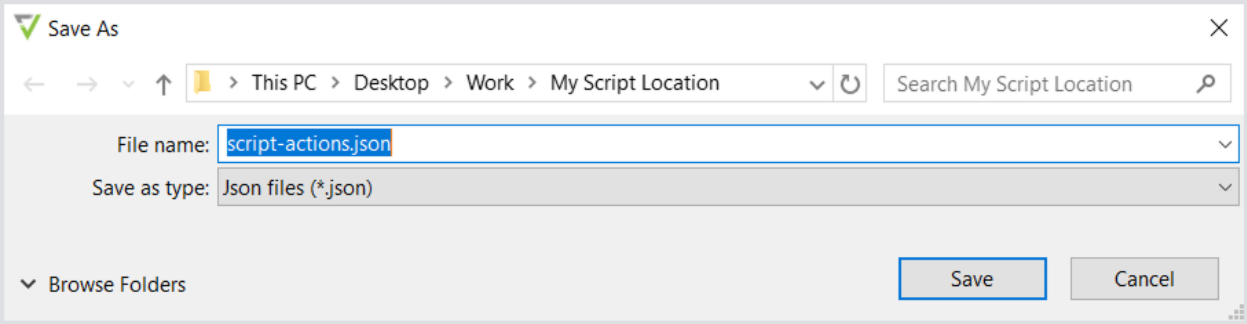


The script should be available under the Debug Output tab:

For more information on using this part of the Recorder, see [Debugging with Script Editor](#).

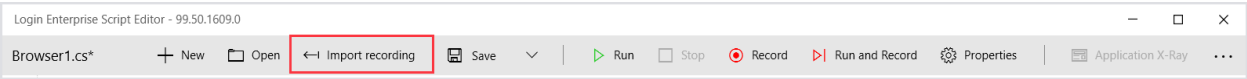
Exporting the Script as JSON

This option provides additional information on script failures, facilitating feedback submission and analysis.



Importing the Script as JSON

Alternatively, if you already have an existing script recording (JSON file), click **Import recording** in the **Script Editor top menu** and proceed with the steps described above.



Restrictions and Limitations

Aspect	Details
Supported applications	Browser-based only. Browser apps such as Teams and Spotify are not supported.
Supported functions	Not all functions are supported in the new template. For details, see the Web Recorder Functions (Playwright).
Framework used	Uses the Playwright framework. Mixing Playwright scripts with Selenium or WinApps scripts is unsupported.
Missing features	Run and record toggle, click coordinate settings, and mouse movement playback (from WinApps recorder).
Unsupported actions	Mouse drag, multi-tab recording, right-click browser menu, key combinations (e.g., Ctrl+C/V).
Wildcard selectors	Not supported.
Web Recorder script template	Code snippets captured by Application X-Ray and/or UI Inspector are not supported.

Recommendations

Recommendation	Explanation
Browser installation	The Web Recorder will use the browser installed on the machine where the script is run. To ensure reliable playback, make sure a supported browser is installed on all machines where the test will run.
Script management	Export and import scripts regularly.
Script compatibility	Avoid mixing different script engines and script types within your projects.
Cookie pop-ups	We recommend waiting a few seconds before interacting with cookie pop-ups, as the recorder may need time to detect them.
Hover recording	**Alt + H** captures the hover action for the element your cursor is on at the moment you press the hotkey. Place your cursor on the desired element and press **Alt + H** to record the hover.
New recordings	Web scripts created with the new Web Recorder use Playwright. Older Selenium-based web workloads are still supported for backward compatibility. However, for new recordings, we recommend using the new Web Recorder because the Playwright-based format is more modern and generally more resilient.

Tutorials

Using Browser Arguments via a Magic Comment

You can pass arguments to the browser by using a magic comment in your script. This allows you to control how the browser behaves during recording and playback.

The most common use cases include:

- Recording and playing back web applications in incognito or private mode.
- Recording and playing back with the browser window maximized.

This feature also supports any other valid browser arguments.

How to Use

Insert the magic comment at the top of your script with the appropriate browser arguments. See the screenshot below for an example.

NOTE: Browser arguments may vary depending on the browser type. Make sure to use the format required by your browser.

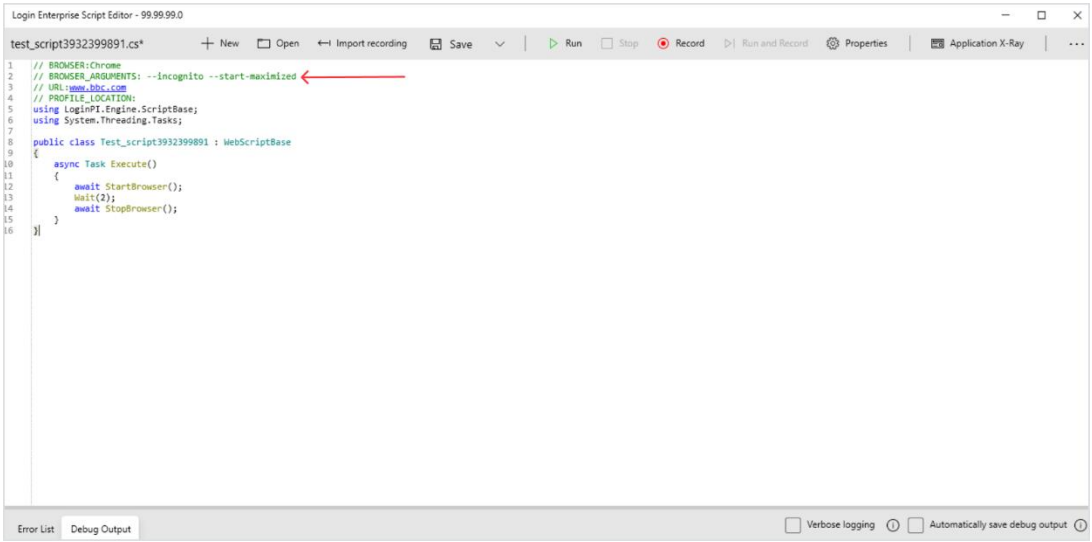
Examples

Chrome

```
--incognito --start-maximized
```

Edge

```
--inprivate --start-maximized
```



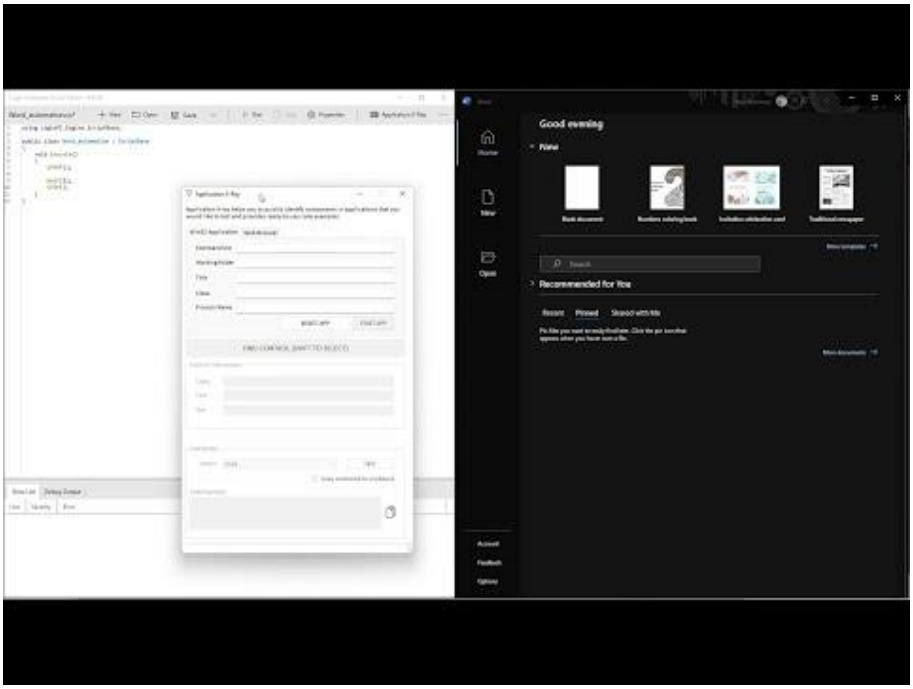
Using Application X-Ray

- Overview
- Downloading and Installing Script Editor
- Starting the Application X-Ray
- Win32 Application
  - Control Information
  - Interaction
- Web Browser
  - Control Information
  - Interaction

Overview

Application X-Ray helps you identify UI components in an application and generates ready-to-use code examples for interacting with them. It's included with every Login Enterprise installation as part of the Script Editor ZIP package. To use Application X-Ray, download and install the Script Editor.

This demonstration shows how to use the Script Editor, with a focus on Application X-Ray, to inspect application components and build automation workflows.



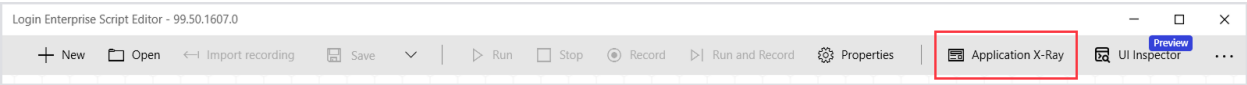
https://youtu.be/v9kFvbxxMwo?feature=shared

Downloading and Installing Script Editor

For details, see a section of the same name in the Script Editor documentation.

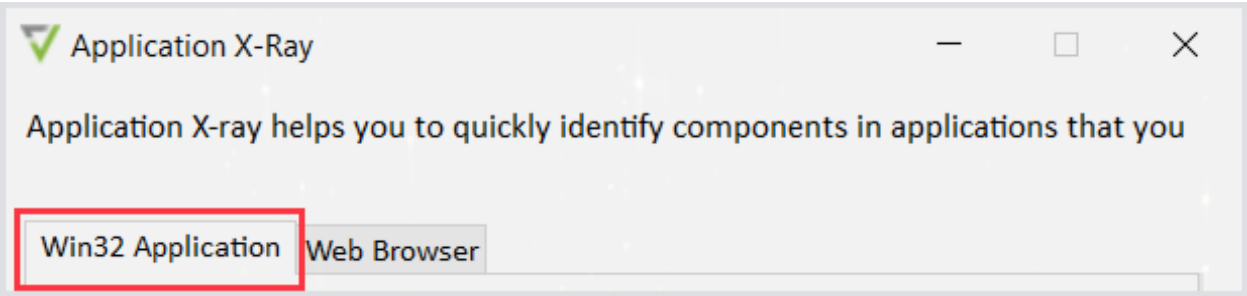
Starting the Application X-Ray

Once you've downloaded and installed the Script Editor, you can access the Application X-Ray from the Script Editor top menu:



Win32 Application

By default, the Application X-Ray opens on the **Win32 Application** tab:

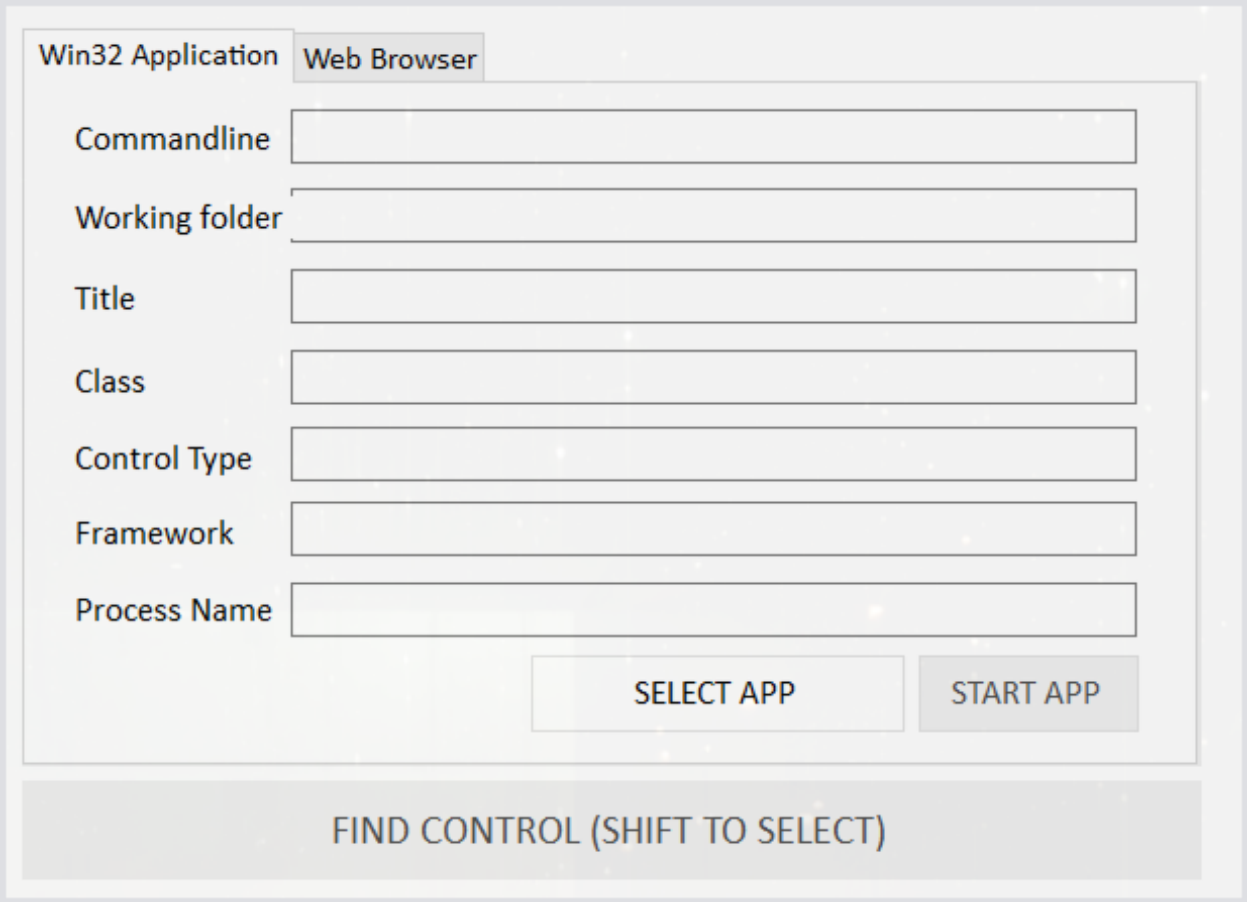


The main section includes the following:

Field	Description
Commandline	Defines the executable file location. You can pre-define the location of the executable, working folder, and title of the application you wish to start. Alternatively, you can start the application yourself, and Application X-Ray will auto-fill this information once activated.
Working folder	Specifies the folder where the file can be found.
Title	Specifies the window title associated with the element.
Class	Specifies the Win32 class name of the element.
Control Type	Specifies the control type of the UI element.
Framework	Specifies the UI automation framework used by the application.
Process Name	Displays the name and ID of the process associated with the window/application.

Other functionality includes:

Option	Description
Select App	Selects the application for use.
Deselect App	Deselects the application, allowing you to choose a new one.
Start app	Launches the application using Application X-Ray.
Find Control	Initiates the search for objects. Press **Shift** to stop searching and lock onto the identified object. After locating an object and pressing **Shift**, the Control Information section of X-Ray will lock to that control in its current state. For example, if the type of the control changes, this will affect the code example (specifically the type), but it will not impact the **Control Information**.



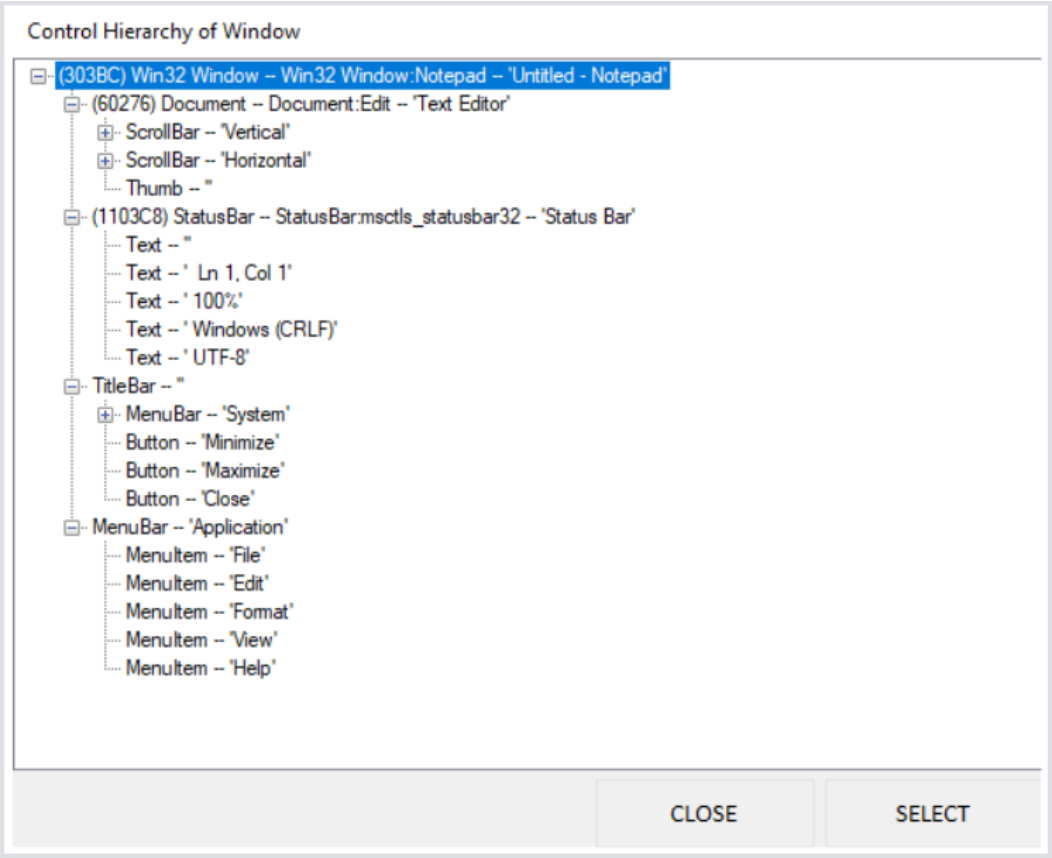
Control Information

In applications, controls typically include text fields, buttons, table fields, and other elements that comprise the user interface. The items displayed are:

Field	Description
Control Type	Specifies the control type of the UI element.
Framework	Specifies the UI automation framework used by the application.
Class	Specifies the Win32 class name of the element.
Title	Specifies the window title associated with the element.
Text	Specifies the visible text of the element.

The **From Tree** button appears when a control within an application is selected. Clicking it opens a new window displaying the complete process/object tree of the application. This feature is useful when certain buttons cannot be located by hovering over them, but can still be identified within the application.

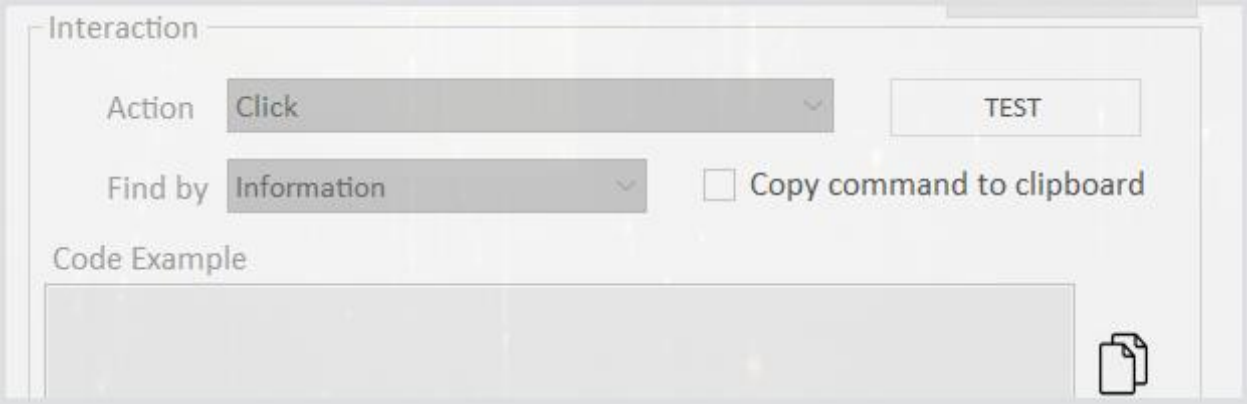
You can browse through the list to find your preferred object. The selected object will be highlighted in the application for easy identification. Once you locate the desired object, click **Select** to use it in Application X-Ray.



Interaction

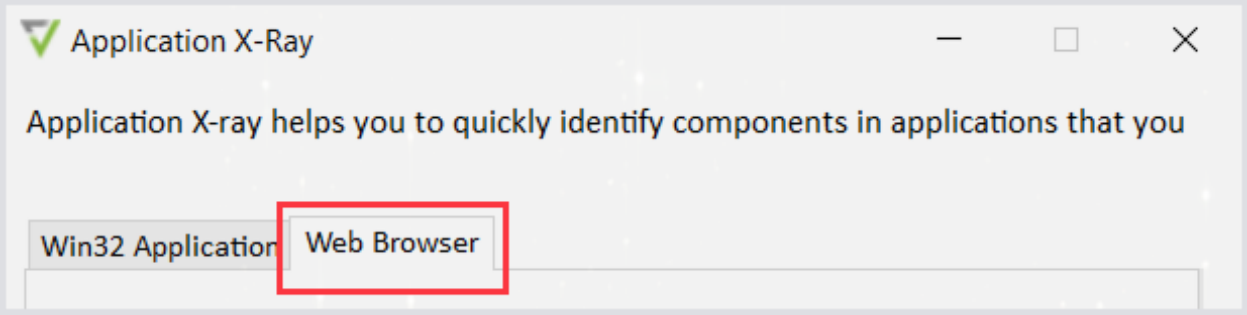
This section allows you to define, test, and copy the selected command for use in your script.

Option	Description
Action	Selects the preferred action. For more information on available functions, see the Scripting Functions Overview.
Find by	Selects the locator strategy to identify a UI element.
Test	Executes the selected action on the chosen window/control with the corresponding defined action.
Copy command to clipboard	Enables automatic copying of all executed actions to the clipboard so you can paste the entire sequence at once.
Code Example	Displays the code string, which is a combination of the preferred action and the selected object.



Web Browser

In the Application X-Ray window, click on the **Web Browser** tab:



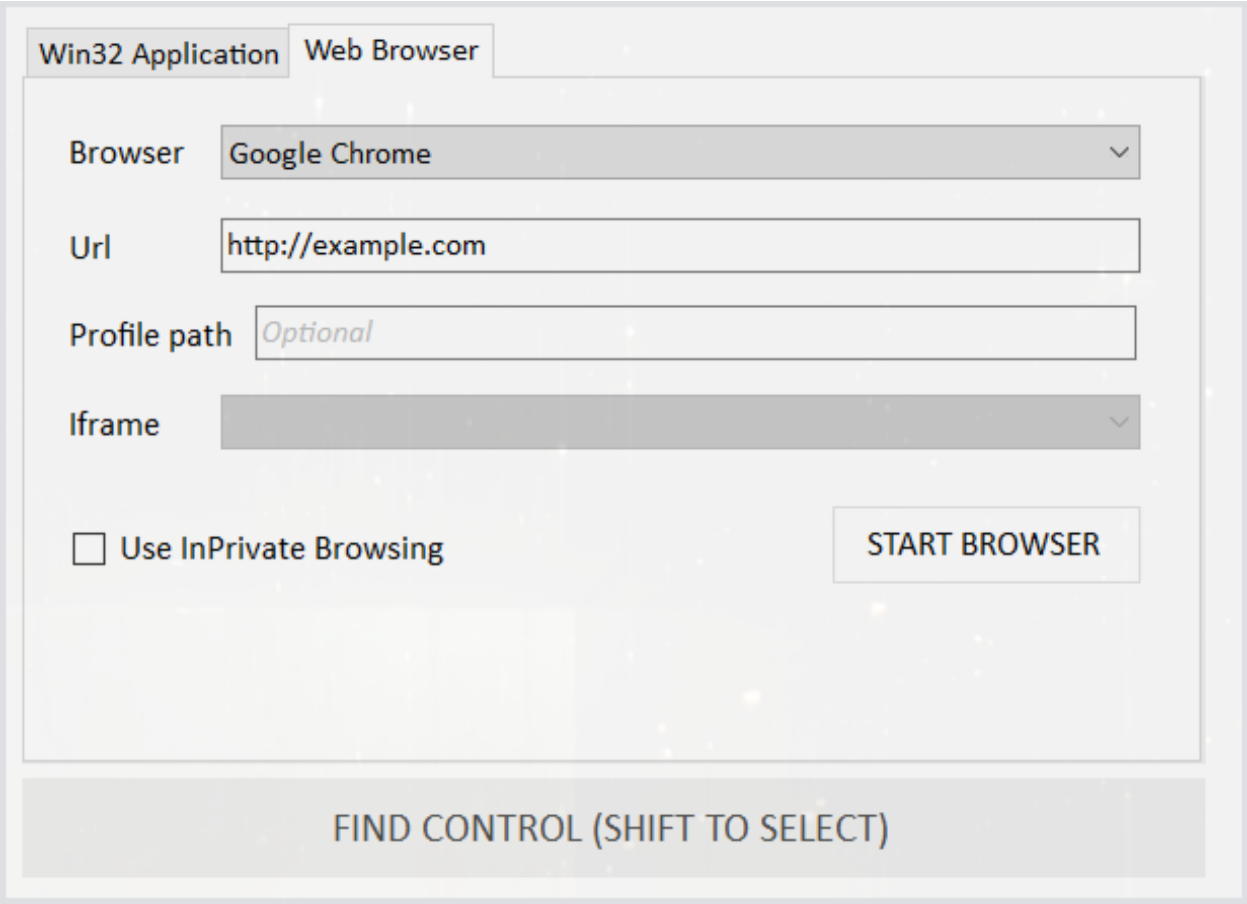
The main section of the **Browser** tab includes the following:

Field	Description
Browser	Specifies the browser and version to use.
URL	Specifies the URL to connect to.
Profile path	Specifies the path to the browser profile to use.
iFrame	Specifies the iframe that contains the target web element.

Other functionality includes:

Option	Description
Use InPrivate Browsing	Enables InPrivate browsing mode when starting the web browser.

Start Browser	Launches the browser with the specified URL.
Find Control	Initiates the search for objects. Press **Shift** to stop searching and lock onto the identified object. After locating an object and pressing **Shift**, the Control Information section of X-Ray will lock to that control in its current state. For example, if the type of the control changes, this will affect the code example (specifically the type), but it will not impact the Control Information.



Control Information

This section describes the selected control on the website, such as a button or text field.

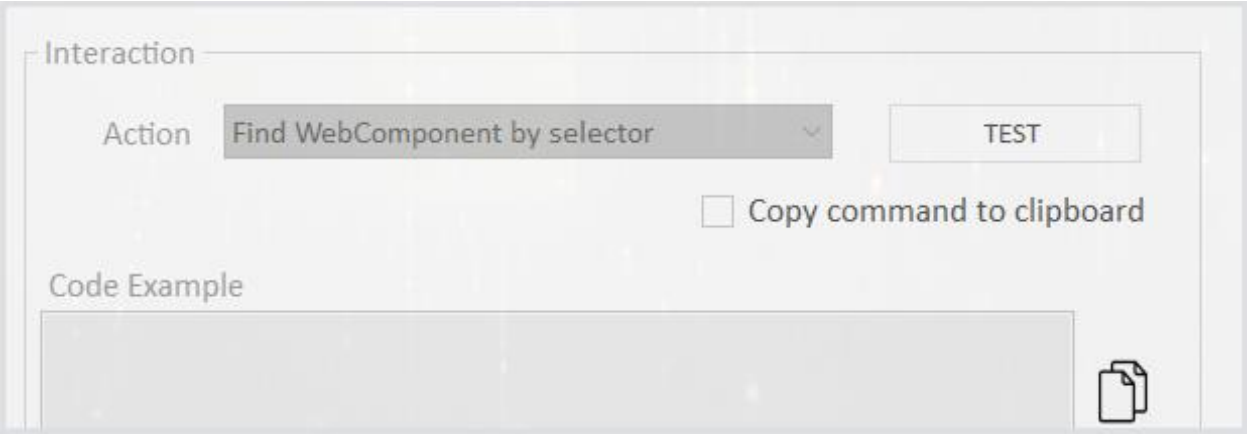
Field	Description
Tag	The HTML tag of the selected control.
Text	The display text of the selected control.

Selector	The CSS selector for the selected control.

Interaction

This section allows you to define, test, and copy the selected command for use in your script.

Option	Description
Action	Specifies the action. For more information on available functions, see the Scripting Functions Overview.
Test	Executes the selected action on the chosen window/control with the corresponding defined action.
Copy command to clipboard	Enables automatic copying of all executed actions to the clipboard so you can paste the entire sequence at once.
Code Example	Displays the code string, which is a combination of the preferred action and the selected object.


Interaction
Action
Find WebComponent by selector
TEST
Copy command to clipboard
Code Example

Scripting Functions Overview

- Overview
- Basic functions
  - Start
  - Stop
  - StartTimer
  - StopTimer
  - CancelTimer
  - SetTimer
  - Wait
  - Log
  - Abort
  - ShellExecute
  - Create Event
  - Get Environment
  - Take Screenshot
- Windows functions
  - FindWindow
  - FindWindows
  - Control Actions
  - FindControl
  - FindControlWithXPath
  - DumpHierarchy
  - FindAutomation Elements
    - FindAutomationElementByXPath
    - FindAutomationElementByInformation
    - FindAutomationElementByXPathOrInformation
- Mouse and Keyboard functions
  - Type
  - Click
  - RightClick
  - DoubleClick
  - MouseMove
  - MouseMoveBy
  - MouseDown
  - MouseUp
  - KeyDown
  - KeyUp
  - KeyCodes
- File and Registry functions
  - CopyFile
  - CopyFile - download existing example files
  - CopyFolder
  - FileExists

    - RemoveFile
    - RemoveFolder
    - DirectoryExists
    - UnzipFile
    - RegImport
- Browser functions
    - StartBrowser
    - StopBrowser
    - Navigate
    - FindWebComponentBySelector
    - FindAllWebComponentsBySelector
- Native automation scripting
    - NativeWindowHandle
    - FromNativeWindowHandle
    - NativeAutomationElement
- Control Flow functions
    - If Else versus Try Catch
    - While
    - ForEach
- Additional resources

Overview

This section provides an introduction to the key functions available for scripting within the Login Enterprise Virtual Appliance. Our scripting framework offers a comprehensive set of tools to enhance automation and control over Windows applications.

1. **Basic Functions**: Essential commands for managing application execution and performance, including launching and closing applications, measuring action durations, and logging key events.
2. **Windows Functions**: Tools for interacting with and controlling windows and their elements, allowing precise manipulation and error handling in your automation scripts.
3. **Mouse and Keyboard Functions**: Functions that simulate user input through typing and mouse actions, enabling you to script user interactions and test application workflows effectively.
4. **File and Registry Functions**: Capabilities for managing file operations and interacting with the Windows registry, ensuring smooth automation of file and registry tasks.
5. **Browser Functions**: Functions designed to control web browsers, navigate URLs, and interact with web elements, streamlining browser automation for web interactions.
6. **Native Automation Scripting**: Advanced capabilities for deeper control over Windows applications using native OS elements, allowing for precise targeting and robust UI automation.
7. **Control Flow Functions**: Conditional logic and iteration tools that enable dynamic scripting, error handling, and adaptable automation workflows.

Login Enterprise also comes with the Script Editor, a tool designed to assist administrators in creating custom application workflows. These workflows can be seamlessly incorporated into various testing scenarios, including Application, Load, and Continuous Tests. By leveraging the Script Editor, administrators can streamline test creation and execution, ensuring comprehensive and efficient testing processes tailored to specific application requirements.

- For more details on the Script Editor, its benefits, use cases, and more, see the Script Editor.
- Script Editor includes the Web Recorder for web apps, in addition to Script Recorder v1 for Windows apps. To learn more about the Script Recorder, its use cases, benefits, and more, see Script Recorder.
- For an overview of Playwright functions, the specific commands used to automate and interact with web applications using Web Recorder v1, see the Web Recorder Functions (Playwright).
- To learn about new methods/functions introduced in the WinApp Recorder, the Inventory of Methods (Generated by Script Recorder).

Basic functions

Basic functions offer essential tools for managing application execution and performance. Commands such as `START` and `STOP` are used to launch and close applications, while `StartTimer` and `StopTimer` help measure the duration of actions within your scripts. You can log key events, take screenshots for validation, and capture environment variables. These functions also include error handling features, providing greater control and visibility over script execution, which ensures smooth automation of your testing processes.

Start

START executes the associated application and creates the mainWindow variable. The START command will auto-detect the application's title, class, and process name. All parameters are optional.

Syntax

```
START(string mainWindowTitle = null, string mainWindowClass = null, string
processName = null, int timeout = 180, bool continueOnError = false);
```

Parameters	Type	Default Value	Description
mainWindowTitle	string		Window title. Example: "Untitled - Notepad"
mainWindowClass	string		Window class. Example: "Win32 Window:Notepad"
processName	string		Window process name. Example: "notepad"

timeout	int	180	Timeout in seconds for the window to appear
continueOnError	bool	false	Continue with the script on the error

Code Example

```
START(mainWindowTitle:"Untitled - Notepad", mainWindowClass:"Win32
Window:Notepad", processName:"notepad", timeout:10);
```

Stop

The STOP command terminates the associated application. The timeout parameter specifies the maximum amount of seconds an application can take to gracefully stop. After the timeout is reached, the application is closed forcefully.

Syntax

```
STOP(int timeout = 5);
```

Parameter	Type	Default Value	Description
timeout	int	5	Timeout in seconds to close an application

Code Example

```
STOP(timeout:5);
```

StartTimer

The StartTimer function allows you to create a custom timer to measure the time one or more actions take within a script. The timers are reported back to the Login Enterprise server as measurements and can be used to trigger a notification if the measurement is over the configured threshold. Measurements are displayed in the Login Enterprise dashboard and charts.

IMPORTANT: The timer name must only contain alphanumeric characters and underscores.

Syntax

```
StartTimer(string name);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 32 characters

IMPORTANT: The timer name must only contain alphanumeric characters and underscores.

Code Example

```
StartTimer(name:"Create_New_File");
StopTimer(name:"Create_New_File");
```

StopTimer

StopTimer is used to stop a timer that's started by StartTimer. The StopTimer name should be the same as the StartTimer.

Syntax

```
StopTimer(string name);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 32 characters

IMPORTANT: The timer name must only contain alphanumeric characters and underscores.

Code Example

```
StartTimer(name:"Save_File_To_Network");
StopTimer(name:"Save_File_To_Network");
```

CancelTimer

The CancelTimer command is used to cancel a timer. The CancelTimer name should be the same as the StartTimer.

Syntax

```
CancelTimer(string name);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 32 characters

IMPORTANT: The timer name must only contain alphanumeric characters and underscores.

Code Example

```
StartTimer(name:"Create_New_File");
CancelTimer(name:"Create_New_File");
```

SetTimer

Create your own custom timers with the SetTimer function. In many cases, you could call an external value of a system resource like CPU % utilization and have it visible in the Login Enterprise reporting.

SetTimer does not require a start and stop function as it collects an integer at the time it is executed.

Syntax

```
SetTimer("name", value);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 32 characters
value	integer		Value collected when the timer is called

IMPORTANT: The timer name must only contain alphanumeric characters and underscores.

Code Example

```
int myCounter = {some code that gets a Windows performance counter in
milliseconds};
SetTimer("MyTimer",myCounter);
```

Wait

The Wait function will wait for an X amount of time in seconds before continuing with the script.

Syntax

```
Wait(double seconds);
```

Parameter	Type	Default Value
seconds	double	
showOnScreen	boolean	true
onScreenText	string	

Code Example

```
Wait(seconds:10, showOnScreen:true, onScreenText:"Waiting 10 seconds");
```

Log

The log function will log to console if you are using the scriptrunner. When using it as part of the Login Enterprise appliance, it logs to %temp%\LoginPI\Logs.

Syntax

```
Log(string message);
Log(object component);
```

Parameters	Type	Default Value
message	string	
component	object	

Code Example

```
Log(message:"Log Message");
Log(component:MainWindow);
```

Abort

Abort is used to gracefully exit the script and throw an error message. The error will be shown as an 'Application failure' on the Login Enterprise dashboard, under events.

Syntax

```
ABORT(string error);
```

Parameter	Type	Default Value
error	string	

Code Example

```
ABORT(error:"An error occurred");
```

ShellExecute

Run a script or application in the Windows Shell

Syntax

```
ShellExecute(string commandLine, string workingFolder = "", bool
waitForProcessEnd = true, int timeout = 180, bool forceKillOnExit = true);
```

Parameter	Type	Default Value	Description
commandLine	string		
workingFolder	string		
waitForProcessEnd	bool	true	Waits for the process to finish before continuing.
timeout	int	180	Defines how long to wait for the process to end before it's killed.
forceKillOnExit	bool	true	Forces the application/process to close or stay open after the command has ended. The default is true.

Code Example

```
ShellExecute("mspaint", waitForProcessEnd: false, timeout: 15,
forceKillOnExit: false);
```

Create Event

The CreateEvent function will generate a custom event on the Enterprise Appliance which is shown in the results page of a Continuous, Load, and Application Test.

Syntax

```
CreateEvent(title:"", description:"");
```

Parameters	Type	Default Value
Title	string	
Description	string	

Code Example

```
CreateEvent(title:"Windows Update Available", description:"Update is
available on hostname x");
```

Get Environment

GetEnvironmentVariable is used as the name suggests to retrieve an environment variable. The GetEnvironmentVariable can be stored in a variable to use multiple times in your script. Note that if a continueOnError is not set and if you do not have permission to retrieve the variable, the script will stop.

Syntax

```
GetEnvironmentVariable(variableName:"", continueOnError:false);
```

Parameters	Type	Default Value
variableName	string	
continueOnError	bool	false

Code Example

```
var TempEnv = GetEnvironmentVariable("temp", continueOnError:true);
Log ($"The environment path = {TempEnv}");
Type ($"{TempEnv}");
```

Take Screenshot

The TakeScreenshot() command is used to create custom screenshots that are sent to the Login Enterprise appliance as Events. This functionality can be used to verify the visual output of an application. You must enable “Custom Screenshots” in the Test configuration to allow this function. This function is not available for Load Tests.

The Event created has a subject of “The script screenshot '{name}' was created”. The description is the {description} field in your function call. The screenshot bitmap is attached to the Event.

IMPORTANT: The timer name must only contain alphanumeric characters and underscores.

Syntax

```
TakeScreenshot(name:"", description:"");
```

Parameters	Type	Default Value
name	string	
description	string	null

Code Example

```
TakeScreenshot(name:"Notepad",description: "Check if notepad has loaded the
correct file");
```

Windows functions

Windows functions provide capabilities for interacting with and controlling windows and their elements in a Windows environment. Use `FindWindow` to detect windows by title, class, or process name. The `FindControl` and `FindControlWithXPath` functions enable interaction with specific controls, such as buttons or text fields. You can manipulate windows and controls using

`Click`, `Maximize`, and `GetBounds`, allowing for precise actions and automation. Additionally, `DumpHierarchy` is available for troubleshooting by exporting a control tree to a file. These functions offer granular control over window elements and include conditional error handling.

FindWindow

FindWindow can be used to detect a new window or a window that is already opened. Each window has a set of information: class, title, and process name, that can be searched on. The ApplicationXRay tool can be used to detect the title, class, and process name of a window or control to allow you to find it in code later.

Syntax

```
FindWindow(string title = null, string className = null, string processName =
null, int timeout = 30, bool continueOnError = false);
```

Parameter	Type	Default Value	Description
title	string		Find window matching title
className	string		Find window matching classname
processName	string		Find window matching process name
timeout	int	30	Timeout in seconds to find the window
continueOnError	bool	false	Continue with the script if an error is encountered

Code Example

```
FindWindow(title:"*Untitled*", timeout: 5);
```

FindWindows

Find all windows by title, class, and/or process. If you specify multiple parameters, only the windows matching all criteria will be returned. The parameters do accept wildcards (*). This method will return whenever even if zero windows are found. If no window is found within the given timeout, it will return an empty array.

Syntax

```
var windowList = FindWindows(title:"", classname:"", processname:"", timeout
= 30);
foreach (var myWindow in windowList) {
    ....
}
```

Parameter	Type	Default Value	Description

title	string		Title of window
className	string		Window class
processName	string		Name of Windows process
timeout	int	30	Timeout value

Code Example

```
FindWindow(title:"*Untitled*", timeout: 5);
```

Control Actions

When the engine finds the window or object, it can perform multiple actions on the window/object. Using the above example, the following window control actions are available:

Name	Description
openDialogue.**Click();**	
openDialogue.**Close();**	
openDialogue.**DoubleClick();**	
openDialogue.**FindControl();**	
openDialogue.**FindControlWithXPath();**	
openDialogue.**Focus();**	
openDialogue.**GetAllChildControls();**	Gets all direct child controls for the window (can also be used on a control)

*openDialogue.***GetBounds();**	Retrieves the window or control boundary coordinates in relation to the screen resolution, returning a `WindowsCoordinates` structure that includes X and Y coordinates along with additional properties. Returns a `WindowsCoordinates` structure containing properties and methods for manipulating and interacting with the retrieved coordinates. **GetBounds() contains the following properties:** • **Bottom** • **Center** • **East** • **Height** • **Left** • **LeftBottom** • **LeftTop** • **North** • **Right** • **RightBottom** • **RightTop** • **South** • **West** • **Width** • **X** • **Y** **Available methods on coordinate pair properties (Center, LeftBottom, LeftTop, RightBottom, RightTop)**: • **Click()**: Click the left mouse button once. • **DoubleClick()**: Double-click the left mouse button. • **MouseMove()**: Move the mouse pointer to the specified location. • **Move(dx: int, dy: int)**: Move the mouse to a location relative to the selected point, specified by pixels. • **MoveDown(dy: int)**: Move the mouse down by the specified number of pixels. • **MoveLeft(dx: int)**: Move the mouse left by a specified number of pixels.

	• **MoveRight(dx: int)**: Move the mouse right by a specified number of pixels. • **MoveUp(dy: int)**: Move the mouse up by a specified number of pixels. • **RightClick()**: Press and release the right mouse button. • **X**: Integer X coordinate of the location. • **Y**: Integer Y coordinate of the location. **Chaining example**: The methods can be chained together for more complex actions. For instance, the following code snippet retrieves the center of control, moves slightly down and to the left, and then performs a right-click: `javascriptCopy codevar myThing = MainWindow.FindControl(…); myThing.GetBounds().Center.Move(dx:-10, dy:10).RightClick();` **NOTE**: The `GetBounds()` method not only retrieves the boundary coordinates but also allows for dynamic interaction through the `WindowsCoordinates` structure.
openDialogue.**GetClass();**	Gets the class name for the window or control
openDialogue.**GetProcess();**	Gets the processname for the window or control
openDialogue.**GetTitle();**	Gets the title for the window or control
openDialogue.**Maximize();**	
openDialogue.**Minimize();**	
openDialogue.**MoveMouseToCenter();**	Moves the mouse to the center of the window
openDialogue.**Restore();**	
openDialogue.**RightClick();**	
openDialogue.**Type("");**	Types directly to the window and keeps the focus
openDialogue.**GetText();**	Retrieves the value from the object
openDialogue.**SwitchTopMostWindow();**	Allows a window to stay on top rather than behind a newly opened window. It provides a better defense against popups overlapping your active window while running a workload.
NativeWindowHandle	For details, see NativeWindowHandle

NativeAutomationElement	For details, see NativeAutomationElement

FindControl

Each window has a set of controls (buttons, input fields, text fields, etc.). Using FindControl allows you to find and subsequently interact with these controls.

Syntax

```
FindControl(string className = null, string title = null, int timeout = 5,
bool searchRecursively = true, bool continueOnError = false);
```

Parameter	Type	Default Value	Description
className	string		Defines the class name of the control to find
title	string		Defines the title of the control to find
timeout	int	5	Timeout period
searchRecursively	bool	true	Defines if only the direct children or all indirect children should be searched
continueOnError	bool	false	Continue with the script if control is not found

Code Example

```
MainWindow.FindControl(className : "Button:Button", title : "&Open").Click();
```

FindControlWithXPath

Each window has a set of controls (buttons, input fields, text fields, etc...). Using FindControlWithXPath allows you to find and subsequently interact with these controls.

Syntax

```
FindControlWithXPath(string xPath, int timeout = 5, bool continueOnError =
false);
```

Parameter	Type	Default Value	Description
xPath	string		(x)Path to the control
timeout	int	5	Timeout period
continueOnError	bool	false	Continue with the script if control is not found

Code Example

```
MainWindow.FindControlWithXPath(xPath : "Button:Button").Click();
```

DumpHierarchy

Every window has a control tree with all classes and names. For troubleshooting purposes, we have a DumpHierarchy method. DumpHierarchy dumps the control tree into a text file.

Syntax

```
DumpHierarchy(string filePath, bool refreshBeforeDump = false);
```

Parameter	Type	Default Value	Description
filePath	string		Defines where to store the dump
refreshBeforeDump	bool	false	Defines if we should refresh the control tree before the dump

Code Example

```
var Open = FindWindow(title:"Open");
Open.DumpHierarchy(filePath:"C:\\dump.txt");
```

FindAutomationElements

The following are the methods generated by Script Recorder and can be used to find UI elements.

FindAutomationElementByXPath

Finds an `AutomationElement` that matches the specified XPath expression. The entire element sequence must match for the element to be considered found.

FindAutomationElementByInformation

Finds an `AutomationElement` by its properties, without using information about ancestor elements.

FindAutomationElementByXPathOrInformation

Finds an automation element by using the specified XPath expression. If no match is found, the method attempts to locate the element by matching automation properties such as automation ID, class name, name, and control type.

For detailed information on these methods, see the Inventory of Methods (Generated by Script Recorder).

Mouse and Keyboard functions

Mouse and keyboard functions automate interactions with Windows applications by simulating user input through typing and mouse actions. The `Type` command lets you enter text or press keys at a specified speed, supporting key combinations and encrypted application credentials. Mouse functions like `Click`, `RightClick`, and `DoubleClick` provide precise control over application elements based on X-Y coordinates. You can also use `MouseMove` and `MouseMoveBy` to reposition the cursor, while `KeyDown` and `KeyUp` simulate holding or releasing keys. These functions enable you to script user actions, test application workflows, and ensure that inputs and interactions behave as expected in real-world conditions.

Type

The Type command is used to type text or press keys.

Syntax

```
Type(string text, int cpm = 300, hideInLogging:true)
```

Parameter	Type	Default Value	Description
text	string		Text to type or keys to press
cpm	int	300	Characters per minute
hideInLogging	Boolean	true	Hides entry from being logged

Variable options:

- **ApplicationPassword -** using this variable you can enter an encrypted password for a specific application.
- **ApplicationUser -** using this variable you can enter an encrypted user for a specific application.

For information on scripting application credentials, see Scripting secure application credentials.

To learn about the encryption types and certificates, see Encryption.

Code Example

```
// use type to type text
Type("Insert Text Here");
// use type to press the enter key
Type("{ENTER}");
// use type to press the enter key twice
```

```
Type("{ENTER}".Repeat(2));
// use type to press ctrl + O
Type("{CTRL+O}");
// use type to enter a path. backslash is an escape character.
Type("C:\\Folder\\Folder1\\something.txt");
// use @ sign to type path
Type(@"C:\Folder\Folder1\something.txt");
// type quotes
// use backslash to escape quote e.g. \"
Type("\"test\""); // this will type "test"
// type quotes
// use @ in front of string and use double quotes
Type(@"""test"""); // this will type "test"
// typing to a window variable will keep the focus on that window
MainWindow.Type("Insert text here");
KeyDown(KeyCode.SHIFT); // Holds the SHIFT key
Type("insert non-capital text here"); // This text will be typed in uppercase
KeyUp(KeyCode.SHIFT); // Releases the SHIFT key
Type("insert non-capital text here"); // This text will be typed in lowercase
```

Click

The Click command is used to mouse-click on an X-Y coordinate.

Syntax

```
Click(double x, double y, bool continueOnError = false);
```

Parameter	Type	Default Value	Description
x	double		Horizontal coordinate
y	double		Vertical coordinate
continueOnError	bool	false	Continue with the script on the error

Code Example

```
Click(x:10, y:10);
```

RightClick

The RightClick command is used to right click on an X Y coordinate.

Syntax

```
RightClick(double x, double y, bool continueOnError = false);
```

Parameter	Type	Default Value	Description

x	double		Horizontal coordinate
y	double		Vertical coordinate
continueOnError	bool	false	Continue with the script on the error

Code Example

```
RightClick(x:10, y:10);
```

DoubleClick

The DoubleClick command is used to double click on an X-Y coordinate.

Syntax

```
DoubleClick(double x, double y, bool continueOnError = false);
```

Parameter	Type	Default Value	Description
x	double		Horizontal coordinate
y	double		Vertical coordinate
continueOnError	bool	false	Continue with the script on the error

Code Example

```
DoubleClick(x:10, y:10);
```

MouseMove

The MouseMove command is used to move the mouse/cursor to a set of X-Y coordinates.

Syntax

```
MouseMove(double x, double y, bool continueOnError = false);
```

Parameter	Type	Default Value	Description
x	double		Horizontal coordinate
y	double		Vertical coordinate
continueOnError	bool	false	Continue with the script on the error

Code Example

```
MouseMove(x:10, y:10);
```

MouseMoveBy

The MouseMoveBy command is used to move the mouse/cursor by a set of pixels relative to the current position.

Syntax

```
MouseMoveBy(double dx, double dy, bool continueOnError = false);
```

Parameter	Type	Default Value	Description
dx	double		Horizontal coordinate
dy	double		Vertical coordinate
continueOnError	bool	false	Continue with the script on the error

Code Example

```
MouseMoveBy(dx:10, dy:10);
```

MouseDown

The MouseDown command presses the left mouse button without letting go.

Code Example

```
MouseDown();
```

MouseUp

The MouseUp command lets go of the left mouse button.

Code Example

```
MouseUp();
```

KeyDown

The KeyDown command holds down a key.

Parameter	Description
KeyCode	See the available keycode list

Code Example

```
KeyDown(KeyCode.SHIFT);
KeyUp(KeyCode.SHIFT);
```

KeyUp

The KeyUp command lets go of a key.

Parameter	Description
KeyCode	See the available keycode list

Code Example

```
KeyDown(KeyCode.CTRL);
KeyUp(KeyCode.CTRL);
```

KeyCodes

Available KeyCodes

```
/// Control-break processing
CANCEL = 0x03,
BREAK = CANCEL,
/// BACKSPACE key
BACK = 0x08,
BACKSPACE = BACK,
/// TAB key
TAB = 0x09,
/// CLEAR key
CLEAR = 0x0C,
/// ENTER key
RETURN = 0x0D,
ENTER = RETURN,
/// SHIFT key
SHIFT = 0x10,
/// CTRL key
CONTROL = 0x11,
CTRL = CONTROL,
/// ALT key
ALT = 0x12,
/// PAUSE key
PAUSE = 0x13,
/// CAPS LOCK key
CAPITAL = 0x14,
CAPSLOCK = CAPITAL,
/// ESC key
ESCAPE = 0x1B,
ESC = ESCAPE,
/// SPACEBAR
SPACE = 0x20,
```

```
/// PAGE UP key
PRIOR = 0x21,
PAGEUP = PRIOR,
/// PAGE DOWN key
NEXT = 0x22,
PAGEDOWN = NEXT,
/// END key
END = 0x23,
/// HOME key
HOME = 0x24,
/// LEFT ARROW key
LEFT = 0x25,
/// UP ARROW key
UP = 0x26,
/// RIGHT ARROW key
RIGHT = 0x27,
/// DOWN ARROW key
DOWN = 0x28,
/// PRINT SCREEN key
SNAPSHOT = 0x2C,
PRINTSCREEN = SNAPSHOT,
/// INS key
INSERT = 0x2D,
INS = INSERT,
/// DEL key
DELETE = 0x2E,
DEL = DELETE,
/// 0 key
KEY_0 = 0x30,
/// 1 key
KEY_1 = 0x31,
/// 2 key
KEY_2 = 0x32,
/// 3 key
KEY_3 = 0x33,
/// 4 key
KEY_4 = 0x34,
/// 5 key
KEY_5 = 0x35,
/// 6 key
KEY_6 = 0x36,
/// 7 key
KEY_7 = 0x37,
/// 8 key
KEY_8 = 0x38,
/// 9 key
KEY_9 = 0x39,
/// A key
KEY_A = 0x41,
/// B key
KEY_B = 0x42,
/// C key
KEY_C = 0x43,
/// D key
KEY_D = 0x44,
/// E key
KEY_E = 0x45,
```

```
/// F key
KEY_F = 0x46,
/// G key
KEY_G = 0x47,
/// H key
KEY_H = 0x48,
/// I key
KEY_I = 0x49,
/// J key
KEY_J = 0x4A,
/// K key
KEY_K = 0x4B,
/// L key
KEY_L = 0x4C,
/// M key
KEY_M = 0x4D,
/// N key
KEY_N = 0x4E,
/// O key
KEY_O = 0x4F,
/// P key
KEY_P = 0x50,
/// Q key
KEY_Q = 0x51,
/// R key
KEY_R = 0x52,
/// S key
KEY_S = 0x53,
/// T key
KEY_T = 0x54,
/// U key
KEY_U = 0x55,
/// V key
KEY_V = 0x56,
/// W key
KEY_W = 0x57,
/// X key
KEY_X = 0x58,
/// Y key
KEY_Y = 0x59,
/// Z key
KEY_Z = 0x5A,
/// Left Windows key (Microsoft Natural keyboard)
LWIN = 0x5B,
/// Right Windows key (Natural keyboard)
RWIN = 0x5C,
/// Applications key (Natural keyboard)
APPS = 0x5D,
MENU = APPS,
/// Numeric keypad 0 key
NUMPAD0 = 0x60,
/// Numeric keypad 1 key
NUMPAD1 = 0x61,
/// Numeric keypad 2 key
NUMPAD2 = 0x62,
/// Numeric keypad 3 key
NUMPAD3 = 0x63,
```

```
/// Numeric keypad 4 key
NUMPAD4 = 0x64,
/// Numeric keypad 5 key
NUMPAD5 = 0x65,
/// Numeric keypad 6 key
NUMPAD6 = 0x66,
/// Numeric keypad 7 key
NUMPAD7 = 0x67,
/// Numeric keypad 8 key
NUMPAD8 = 0x68,
/// Numeric keypad 9 key
NUMPAD9 = 0x69,
/// Multiply key
MULTIPLY = 0x6A,
NUMPAD_MULTIPLY = MULTIPLY,
/// Add key
ADD = 0x6B,
PLUS = ADD,
NUMPAD_ADD = ADD,
/// Subtract key
SUBTRACT = 0x6D,
MINUS = SUBTRACT,
NUMPAD_SUBTRACT = SUBTRACT,
/// Decimal key
DECIMAL = 0x6E,
NUMPAD_DECIMAL = DECIMAL,
/// Divide key
DIVIDE = 0x6F,
NUMPAD_DIVIDE = DIVIDE,
/// F1 key
F1 = 0x70,
/// F2 key
F2 = 0x71,
/// F3 key
F3 = 0x72,
/// F4 key
F4 = 0x73,
/// F5 key
F5 = 0x74,
/// F6 key
F6 = 0x75,
/// F7 key
F7 = 0x76,
/// F8 key
F8 = 0x77,
/// F9 key
F9 = 0x78,
/// F10 key
F10 = 0x79,
/// F11 key
F11 = 0x7A,
/// F12 key
F12 = 0x7B,
/// NUM LOCK key
NUMLOCK = 0x90,
/// SCROLL LOCK key
SCROLL = 0x91,
```

```
SCROLL_LOCK = SCROLL,
/// Left SHIFT key
LSHIFT = 0xA0,
/// Right SHIFT key
RSHIFT = 0xA1,
/// Left CONTROL key
LCONTROL = 0xA2,
LCTRL = LCONTROL,
/// Right CONTROL key
RCONTROL = 0xA3,
RCTRL = RCONTROL,
/// Left MENU key
LMENU = 0xA4,
LALT = LMENU,
/// Right MENU key
RMENU = 0xA5,
RALT = RMENU,
/// Windows 2000/XP: Volume Mute key
VOLUME_MUTE = 0xAD,
/// Windows 2000/XP: Volume Down key
VOLUME_DOWN = 0xAE,
/// Windows 2000/XP: Volume Up key
VOLUME_UP = 0xAF,
```

File and Registry functions

File and registry functions provide the capability to manage file operations and interact with the Windows registry in your automation scripts. Use `CopyFile` and `CopyFolder` to move files or directories, while `FileExists` and `DirectoryExists` verify whether the items were successfully copied. Commands such as `RemoveFile` and `RemoveFolder` facilitate cleanup of files and folders as needed. For handling compressed data, `UnzipFile` extracts contents from ZIP archives. Registry interactions are managed through `RegImport`, which imports registry settings from a specified file. These functions ensure efficient file management and registry operations.

CopyFile

The CopyFile command will copy a file from source to destination, overwriting the destination path.

Syntax

```
CopyFile(string sourcepath, string destinationPath, bool continueOnError,
bool overwrite);
```

Parameter	Type	Default Value	Description
sourcepath	string		Copy file source path
destinationPath	string		Copy file destination path
continueOnError	boolean	false	Continue if an error occurs

overwrite	boolean	true	Overwrite the target file, if it exists

Code Example

```
CopyFile(sourcePath:"C:\\somefile.txt",destinationPath:"D:\\somefile.txt",con
tinueOnError:false,overwrite:true)
```

CopyFile - download existing example files

You can download known files from the Appliance. This is part of the existing CopyFile command. These files can be found on the Appliance under "/loginVSI/content/scriptcontent/" when using WINSCP to connect to the Appliance.

Syntax

```
CopyFile(KnownFiles.FileType, string destinationPath, bool continueOnError,
bool overwrite);
```

Parameter	Type	Default Value	Description
KnownFiles.	filetype		KnownFiles.PdfFile 1. PdfFile. Downloads a LoginVSI pdf file 2. Excelsheet. Downloads a LoginVSI xlsx file 3. OutlookConfiguration. Downloads a LoginVSI PRF file 4. OutlookData. Downloads a LoginVSI PST file 5. PowerPointPresentation. Downloads a LoginVSI pptx file 6. RichTextFile. Downloads a LoginVSI rtf file 7. WordDocument. Downloads a LoginVSI txt file
destinationPath	string		Copy file destination path
continueOnError	boolean	false	Continue if an error occurs
overwrite	boolean	true	Overwrite the target file, if it exists

Code Example

```
CopyFile(KnownFiles.PdfFile,
"C:\\Temp\\loginvsi.pdf",continueOnError:false,overwrite:true);
```

CopyFolder

The CopyFolder command will copy a folder from source to destination, overwriting the destination path.

Syntax

```
CopyFolder(string sourcepath, string destinationPath);
```

Parameter	Type	Default Value	Description
sourcepath	string		Copy folder source path
destinathPath	string		Copy folder destination path

Code Example

```
CopyFolder(@"c:\testfolder", @"c:\temp\testfolder");
CopyFolder(string sourcepath, string destinationPath);
```

FileExists

Check if a file exists, can be used as a verification to check if the file was successfully copied.

Syntax

```
FileExists(string path);
```

Parameter	Type	Default Value	Description
path	string		Path of the file to verify

Code Example

```
FileExists("C:\\Temp\\loginvsi.pdf");
```

RemoveFile

The RemoveFile command will remove/delete the specified file.

Syntax

```
RemoveFile(string path);
```

Parameter	Type	Default Value	Description
path	string		Path of the file to delete

Code Example

```
# literal string path
RemoveFile(path: "c:\\testfolder\\some file.txt");
# verbatim string path
RemoveFile(@"c:\testfolder\some file.txt");
```

RemoveFolder

The RemoveFolder command will remove/delete the specified folder.

Syntax

```
RemoveFolder(string path);
```

Parameter	Type	Default Value	Description
path	string		Path of the folder to remove

Code Example

```
# literal string path
RemoveFolder(path: "c:\\temp\\testfolder");
# verbatim string path
RemoveFolder(@"c:\temp\testfolder");
```

DirectoryExists

The DirectoryExists function checks if a specified folder exists. Can be used after you create a folder or if you wish to verify if the folder exists before executing an action

Syntax

```
DirectoryExists(string path);
```

Parameter	Type	Default Value	Description
pathOfDirectory	string		Path of the directory to verify

Code Example

```
DirectoryExists("C:\\Temp");
```

UnzipFile

The DirectoryExists function checks if a specified folder exists. Can be used after you create a folder or if you wish to verify if the folder exists before executing an action

Syntax

```
UnzipFile(string SourcePath, destinationFolder, continueOnError, overWrite);
```

Parameter	Type	Default Value	Description
sourcePath	string		Path of the .zip file
destinationFolder	string		Path of the folder to unzip the contents into
continueOnError	boolean	false	Continue if an error occurs
overWrite	boolean	true	Overwrite if the file(s) exist

Code Example

```
UnzipFile(sourcePath:"C:\\Temp\\loginvsi.zip", destinationFolder:"C:\\temp\\l
oginvsiwebsite",continueOnError:false,overWrite:false);
```

RegImport

The RegImport command will import a registry file to the system registry.

Syntax

```
RegImport(string registryFile);
```

Parameter	Type	Default Value	Description
registryFile	string		Path of the registry file to import

Code Example

```
# literal string path
RegImport(registryFile: "C:\\temp\\regfile.reg");
# verbatim string path
RegImport(@"C:\temp\regfile.reg");
```

Browser functions

Browser functions provide the capability to control web browsers in your application scripts. Use `StopBrowser` to close a previously opened browser. The `Navigate` function directs the browser to a specified URL and tracks the time it takes to load the page, which is useful for performance monitoring. `FindWebComponentBySelector` and `FindAllWebComponentsBySelector` allow you to locate and interact with web elements using CSS selectors, enabling actions such as clicking or typing directly into the identified components. These functions streamline browser automation, allowing you to script web interactions and measure performance within your application workflows.

StartBrowser

The StartBrowser command allows you to start a browser within an application script. Right now we only support Chrome80 and Edge42/44.

Syntax

```
StartBrowser(useInPrivateBrowsing, expectedURL, timeout, continueOnError);
```

Parameter	Type	Default Value	Description
useInPrivateBrowsing	boolean	false	Define if the browser starts in private mode
expectedURL	string		Defines the URL that you want to wait for
timeout	int	60	Waits for # seconds before timing out if not found
continueOnError	boolean	false	Continues the process if an error occurs

Code Example

```
StartBrowser(useInPrivateBrowsing:false,expectedUrl:"https://www.loginvsi.com
",timeout:30,continueOnError:true);
```

StopBrowser

The StopBrowser command allows you to stop a browser within an application script.

Syntax / Code Example

```
StopBrowser();
```

Navigate

The navigate command will tell the open browser to browse to a specific URL. It automatically times the time it takes to browse the page.

Syntax

```
Navigate(string url, string timerName);
```

Parameter	Type	Default Value	Description
Url	string		URL to browse too
timerName	string		Name of the timer

Code Example

```
Navigate(url: "https://loginvsi.com", timerName: "Website_LoginVSI");
```

FindWebComponentBySelector

The findWebComponentBySelector finds a web element based on the given CSS selector. You can attach existing control actions. For more details, see Control Actions. You can also move your mouse to the component or search your component by text.

NOTE: Make the search field as small as possible to eliminate long searches.

Syntax

```
FindWebComponentBySelector(string selector, int timeout, bool
continueOnError);
```

Parameter	Type	Default Value	Description
selector	string		CSS Selector on page
timeout	int	30	Timeout in seconds
continueOnError	boolean	false	Continue if an error occurs

Code Example

```
FindWebComponentBySelector(selector:"",timeout:30,continueOnError:false);
// It's also possible to type to the web element via
FindWebComponentBySelector(selector:"",timeout:30,continueOnError:false).Type
("Example text");
// You can also send a Click to the found web element
FindWebComponentBySelector(selector:"",timeout:30,continueOnError:false).Clic
k();
Browser.FindWebComponentBySelector("button[id='logonbutton']",timeout:30,cont
inueOnError:false).Click();
```

FindAllWebComponentsBySelector

Sometimes the CSS selector appears multiple times within the webpage, which makes it hard to find the one you are looking for. To list the selectors you can use this command. The result can be used by findWebComponentBySelector.

You can also move your mouse to the component. For more information, see Control Actions.

Syntax

```
FindAllWebComponentsBySelector(string selector, int timeout);
```

Parameter	Type	Default Value	Description
selector	string		Selector object name
timeOut	string	30	Timeout value

Code Example

```
var components = FindAllWebComponentsBySelector(selector:"A", timeout:30);
```

Native automation scripting

Native automation scripting enhances the capabilities of scripting within the Virtual Appliance by providing deeper control over Windows applications through native OS elements. The `NativeWindowHandle` and `FromNativeWindowHandle` functions enable you to identify and work directly with windows using their native OS handles, allowing for precise targeting. `SwitchTopMostWindow` ensures that the desired window remains on top, preventing pop-ups from interrupting automated tasks. `NativeAutomationElement` offers direct access to UIAutomation features, providing robust control over UI elements when default methods, such as typing, are insufficient. These functions significantly enhance your ability to script and automate complex interactions with Windows applications.

NOTE: Using these functions requires you to include the following line at the top of your script:

```
using Interop.UIAutomationClient;
```

NativeWindowHandle

Returns the unique identifier of the window as it is known to the Windows OS. This solves the issue where we cannot uniquely identify any given window.

Syntax

```
IntPtr NativeWindowHandle;
```

Code Example

```
FindWindow(title:"*Untitled*", timeout: 5).NativeWindowHandle;
```

FromNativeWindowHandle

Constructs a scripting window object from a native window handle. This allows us to work with a window that was found by its handle using native technologies like win32 calls.

Syntax

```
IWindow FromNativeWindowHandle(IntPtr handle);
```

Code Example

```
var myNativeWindow =
Process.GetProcessesByName("notepad")[0].MainWindowHandle;
var myLEWindow = FromNativeWindowHandle(myNativeWindow);
```

NativeAutomationElement

This provides access to the underlying automation (UIAutomation) library that Login Enterprise Engine uses. It allows us to use low-level automation actions like manipulating checkboxes directly or setting the value of a text edit control in a more resilient way if typing doesn't work properly.

NOTE: In-depth information about the UIAutomationClient functions is beyond the scope of this user guide. You can download the code example from here.

Syntax

```
IUIAutomationElement NativeAutomationElement;
```

Code Example

```
MainWindow.FindControl(classname:"*Document*").NativeAutomationElement;
```

Control Flow functions

Control flow functions provide options for implementing conditional logic and iteration in your scripts. The `If Else` statement allows you to execute code based on whether a condition is true or false. The `Try Catch` construct enables error handling, ensuring scripts continue running smoothly even if an element is not found. Loops like `While` and `ForEach` let you repeat actions until a condition is met or iterate over a collection of values. These control flow functions are essential for building dynamic and resilient scripts, helping you automate workflows that adapt to changing conditions and handle errors gracefully.

These functions represent standard programming concepts that are common across many languages, including C#. While they may be familiar to experienced programmers, we recognize that many of our users may not have a programming background. This section aims to clarify these concepts for those who may be new to scripting, particularly as our workload scripting in C# supports these features, facilitating more advanced and conditional scenarios.

If Else versus Try Catch

Function name: If Else statement

Description: The If...Else statement allows you to create conditional logic in your scripts. If the condition is true, the code within the If block executes. If the condition is false, the code within the Else block executes. Note that the Else statement is not mandatory.

Example:

```
var Condition = MainWindow.FindControl(title:"Condition");
if (Condition != null)
{
Type("Condition is true");
}
else
{
Type("Condition is not true");
}
```

Challenges:

Finding a control may throw an exception if the control is not found. If "Continue on Error" is not enabled, this will terminate the workflow.

Solution

Reword the script using a Try...Catch statement. This approach can eliminate inconsistent pop-ups and allow the script to continue running even if a control is not found.

Example:

```
Try{
var Condition = MainWindow.FindControl(title:"Condition");
Condition.Click();
}
catch{
Log("Was unable to find conditional window");
};
```

While

Function name: While

Description: The While statement executes a block of code in a loop as long as the specified condition evaluates to true. The condition is evaluated before each iteration of the loop.

Example:

In the following example, we specify an integer variable that equals zero. While the number is less than 10, we log the number to the console and add +1 to it until it evaluates false.

```
int number = 0;
```

```
while(number < 10)
{
Log(number);
number++;
}
```

ForEach

Function name: foreach

Description: The ForEach statement iterates through each item in an array of values.

Example:

In the following example, we iterate through 3 sample values.

```
string[] example = {"example 1","example 2","example 3"};
foreach (string value in example)
{
Log(value);
}
```

Additional Uses

When working with a browser application, collecting all web selectors can be beneficial. You can then use a ForEach loop to find a specific condition or to identify a particular selector from an array of many.

Web Recorder Functions (Playwright)

- Overview
- Basic Functions
  - StartBrowser
  - StopBrowser
  - Navigate
  - Locator
  - StartTimer
  - StopTimer
  - CancelTimer
  - SetTimer
  - Wait
  - Log
  - Abort
  - ShellExecute
  - CreateEvent
  - GetEnvironmentVariable
  - TakeScreenshot
- Mouse and Keyboard Functions
  - Type
  - Key Combinations
  - Press
    - KeyCodes
  - LeftClick
  - RightClick
  - DoubleClick
  - Mouse Scroll
  - Hover
- Control Flow Functions
  - If Else vs Try Catch
  - While
  - ForEach

Overview

This section provides the commands available in Playwright scripts created with Web Recorder. These functions let you automate and interact with web applications and are grouped into categories to help you find what you need quickly.

In most cases, you do not need to work with these functions directly if the Web Recorder captures and plays back the required actions successfully. This page is intended primarily as an advanced reference for users who need to review, adjust, or extend recorded functions, such as when the recorder does not capture an interaction correctly, when conditional logic needs to be added, or when a recorded step needs to be adjusted:

1. **Basic Functions:** Core actions like opening pages, locating elements, and performing simple operations.
2. **Mouse and Keyboard Functions:** Simulate user interactions, including clicks, typing, scrolling, and dragging.
3. **Browser Functions:** Manage browser contexts, tabs, and sessions.
4. **Control Flow Functions:** Add logic to your automation scripts, including loops, conditions, and waits.

Basic Functions

StartBrowser

Starts a browser within an application script. We support Microsoft Edge and Chrome; the application uses whichever version of these browsers is installed on the target machine. Note that Firefox is currently not supported.

Syntax

```
StartBrowser()
```

Parameter	Type	Default Value	Description
(none)	-	-	This command does not require any parameters.

Code example

```
await StartBrowser();
```

StopBrowser

Stops a browser within an application script.

Syntax

```
StopBrowser()
```

Parameter	Type	Default Value	Description
(none)	-	-	This command does not require any parameters.

Code example

```
await StopBrowser();
```

Navigate

Navigates the open browser to a specified URL and automatically measures the time required to complete the navigation.

Syntax

```
await NavigateAsync(string url, string timerName);
```

Parameter	Type	Default Value	Description
Url	string		URL to navigate to
timerName	string		Name of the timer

Code example

```
await NavigateAsync(url: "https://loginvsi.com", timerName:
"Website_LoginVSI");
```

Locator

Identifies a specific element on the page by retrying to locate it until the specified timeout is reached. Use this method as a starting point for actions such as click, hover, type, or get text.

Syntax

```
Locator(string selector, int timeout);
```

The `locator` value is generated by the Web Recorder. Creating locators manually is not supported. Please use the Web Recorder to capture locators.

Code example

```
Locator("input[autocomplete='username']", timeout: 10);
```

StartTimer

Creates a custom timer to measure the time one or more actions take within a script. The timers are reported back to the Login Enterprise server as measurements and can be used to trigger a notification if the measurement is over the configured threshold. Measurements are displayed in the Login Enterprise dashboard and charts. StartTimer must be stopped with either CancelTimer or StopTimer.

Syntax

```
StartTimer(string name);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 64 characters

The timer name must only contain alphanumeric characters and underscores.

Code example

```
StartTimer(name:"Create_New_File");
StopTimer(name:"Create_New_File");
```

StopTimer

Stops a timer started by StartTimer. The StopTimer name should be the same as the StartTimer.

Syntax

```
StopTimer(string name);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 64 characters

The timer name must only contain alphanumeric characters and underscores.

Code example

```
StartTimer(name:"Save_File_To_Network");
StopTimer(name:"Save_File_To_Network");
```

CancelTimer

Cancels a timer. The CancelTimer name should be the same as the StartTimer.

Syntax

```
CancelTimer(string name);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 64 characters

The timer name must only contain alphanumeric characters and underscores.

Code example

```
StartTimer(name:"Create_New_File");
CancelTimer(name:"Create_New_File");
```

SetTimer

Helps create your own custom timers. In many cases, you can call an external value for a system resource, such as CPU utilization %, and have it appear in the Login Enterprise reporting.

SetTimer does not require a start and stop function as it collects an integer at the time it is executed.

Syntax

```
SetTimer("name", value);
```

Parameter	Type	Default Value	Description
name	string		Defines the name of the timer, max of 64 characters
value	integer		Value collected when the timer is called

The timer name must only contain alphanumeric characters and underscores.

Code example

```
int myCounter = {some code that gets a Windows performance counter in
milliseconds};
SetTimer("MyTimer",myCounter);
```

Wait

Waits for X seconds before continuing the script.

Syntax

```
Wait(double seconds);
```

Parameter	Type	Default Value
seconds	double	-

Code example

```
Wait(10);
```

Log

Logs to the console if you are using the Scriptrunner. When using it as part of the Login Enterprise appliance, it logs to %temp%\LoginPI\Logs.

Syntax

```
Log(string message);
Log(object component);
```

Parameters	Type	Default Value
message	string	
component	object	

Code example

```
Log(message:"Log Message");
Log(component:MainWindow);
```

Abort

Exits the script and throws an error message. The error will be shown as an 'Application failure' on the Login Enterprise dashboard, under events.

Syntax

```
ABORT(string error);
```

Parameter	Type	Default Value
error	string	

Code example

```
ABORT(error:"An error occurred");
```

ShellExecute

Runs a script or application in the Windows Shell. Only the first parameter is required; all remaining parameters are optional.

Syntax

```
ShellExecute(string commandLine, string workingFolder = "", bool
waitForProcessEnd = true, int timeout = 180, bool forceKillOnExit = true);
```

Parameter	Type	Default Value	Description
commandLine	string		
workingFolder	string		
waitForProcessEnd	bool	true	Waits for the process to finish before continuing.
timeout	int	180	Defines how long to wait for the process to end before it's killed.
forceKillOnExit	bool	true	Forces the application/process to close or stay open after the command has ended. The default is true.

Code example

```
ShellExecute("mspaint", waitForProcessEnd: false, timeout: 15,
forceKillOnExit: false);
```

CreateEvent

Generates a custom event on the Appliance, shown in the results page of a Continuous, Load, and Application Test.

Syntax

```
CreateEvent(title:"", description:"");
```

Parameters	Type	Default Value
Title	string	-
Description	string	-

Code example

```
CreateEvent(title:"Windows Update Available", description:"Update is
available on hostname x");
```

GetEnvironmentVariable

Retrieves an environment variable. The GetEnvironmentVariable can be stored in a variable to use multiple times in your script. Note that if a continueOnError is not set and if you do not have permission to retrieve the variable, the script will stop.

Syntax

```
GetEnvironmentVariable(variableName:"", continueOnError:false);
```

Parameters	Type	Default Value
variableName	string	
continueOnError	bool	false

Code example

```
var TempEnv = GetEnvironmentVariable("temp", continueOnError:true);
Log ($"The environment path = {TempEnv}");
Type ($"{TempEnv}");
```

TakeScreenshot

Creates custom screenshots that are sent to the Login Enterprise appliance as Events. This functionality can be used to verify an application's visual output. You must enable “Custom Screenshots” in the Test configuration to allow this function. This function is not available for Load Tests.

The Event created has a subject of “The script screenshot '{name}' was created”. The description is the {description} field in your function call. The screenshot bitmap is attached to the Event.

The name field cannot handle spaces; these need to be underscored.

Syntax

```
TakeScreenshot(name:"", description:"");
```

Parameters	Type	Default Value
name	string	
description	string	null

Code example

```
 TakeScreenshot(name:"Notepad",description: "Check if notepad has loaded the
correct file");
```

Mouse and Keyboard Functions

Some functions in this section include a `timeout` parameter. This parameter specifies how long the script will retry locating the target element before failing.

By default, the Web Recorder sets the timeout to 5 seconds, but you can adjust it either globally or on a per-action basis.

If the `timeout` is omitted, the script falls back to Playwright's default timeout of 30 seconds.

Type

Types text or presses keys.

Syntax

```
await TypeAsync(input: "text", cpm: 0);
```

Parameter	Type	Default Value	Description
text	string		Text to type or keys to press
cpm	int	300	Characters per minute

Code example

```
await TypeAsync(input: "some Text", cpm: 0);
```

Key Combinations

Key combinations can be performed using the typing function. For example, the **Ctrl+V** (paste) command appears in the script as:

```
await TypeAsync(input: "{Control+v}", cpm: 0);
```

Press

Presses keys, such as Key up/down, tab, etc.

Parameter	Type	Default Value	Description
key	string	-	The key to press on the keyboard

Syntax:

```
PressAsync(key)
```

Code example:

```
await PressAsync("Tab");
```

KeyCodes

Valid keys for PressAsync()

```
F1
F2
F3
F4
F5
F6
F7
F8
F9
F10
F11
F12

Insert
Cancel
Help
Backspace
Tab
Clear
Enter
Shift
Control
Alt
Delete
Pause
Space
PageUp
PageDown
End
Home
ArrowLeft
ArrowUp
ArrowRight
ArrowDown
Escape
Meta
```

LeftClick

Performs a left-mouse click on the specified element.

Syntax

```
Locator(selector).ClickAsync(options)
```

Parameter	Description

selector	Selector identifying the element to click.
timeout	How long (in seconds) the script retries locating the element before failing.

The `locator` value is generated by the Web Recorder. Creating locators manually is not supported. Please use the Web Recorder to capture locators.

Code example

```
await Locator(selector: "#password-input").ClickAsync(new(){Timeout = 5 });
```

RightClick

Performs a right-mouse click on the specified element.

Syntax

```
RightClick(double x, double y, bool continueOnError = false);
```

Parameter	Description
selector	Selector identifying the element to click.
timeout	How long (in seconds) the script retries locating the element before failing.

The `locator` value is generated by the Web Recorder. Creating locators manually is not supported. Please use the Web Recorder to capture locators.

Code example

```
await Locator(selector: "#password-input").RightClickAsync(new(){Timeout = 5
});
```

DoubleClick

Performs a double-mouse click on the specified element.

Syntax

```
Locator(selector).DoubleClickAsync(options)
```

Parameter	Description
selector	Selector identifying the element to click.
timeout	How long (in seconds) the script retries locating the element before failing.

The `locator` value is generated by the Web Recorder. Creating locators manually is not supported. Please use the Web Recorder to capture locators.

Code example

```
await Locator(selector: "#password-input").DoubleClickAsync(new(){Timeout = 5
});
```

Mouse Scroll

Scrolls the page relative to the current mouse position by the given number of pixels horizontally (x) and vertically (y).

Parameter	Type	Default value	Description
x	-	-	Horizontal scroll amount in pixels (negative = left, positive = right)
y	-	-	Vertical scroll amount in pixels (negative = up, positive = down)

Code example

```
// Horizontal scroll at (0,0)
await ScrollAsync(x: 0, y: 0);

// Scroll down at (0,659)
await ScrollAsync(x: 0, y: 659);

// Scroll up at (0,-32)
await ScrollAsync(x: 0, y: -32);
```

Hover

Moves the mouse pointer over the specified element. It is used to trigger hover effects like tooltips, dropdowns, or style changes.

Syntax

```
Locator(selector).HoverAsync()
```

The `locator` value is generated by the Web Recorder. Creating locators manually is not supported. Please use the Web Recorder to capture locators.

Code example

```
await Locator(selector: "#password-input").HoverAsync();
```

Control Flow Functions

If Else vs Try Catch

Creates conditional logic in your scripts. If the condition is true, the code within the If block executes. If the condition is false, the code within the Else block executes. Note that the Else statement is not mandatory.

Code example

```
var Condition = MainWindow.FindControl(title:"Condition");
if (Condition != null)
{
Type("Condition is true");
}
else
{
Type("Condition is not true");
}
```

Challenges:

Finding a control may throw an exception if the control is not found. If "Continue on Error" is not enabled, this will terminate the workflow.

Solution

Reword the script using a Try...Catch statement. This approach can eliminate inconsistent pop-ups and allow the script to continue running even if a control is not found.

Code example:

```
Try{
var Condition = MainWindow.FindControl(title:"Condition");
Condition.Click();
}
catch{
Log("Was unable to find conditional window");
};
```

While

Executes a block of code in a loop as long as the specified condition evaluates to true. The condition is evaluated before each iteration of the loop.

Code example:

In the following example, we specify an integer variable that equals zero. While the number is less than 10, we log the number to the console and add +1 to it until it evaluates to false.

```
int number = 0;
while(number < 10)
{
Log(number);
number++;
}
```

ForEach

Iterates through each item in an array of values.

Code example:

In the following example, we iterate through 3 sample values.

```
string[] example = {"example 1","example 2","example 3"};
foreach (string value in example)
{
Log(value);
}
```

Additional uses

When working with a browser application, collecting all web selectors can be beneficial. You can then use a ForEach loop to find a specific condition or to identify a particular selector from an array of many.

Back to Application Script Recording

Back to Table of Contents