

# Troubleshoot Qlik Sense using logs

Qlik Sense®

February 2019

Copyright © 1993-2019 QlikTech International AB. All rights reserved.



© 2019 QlikTech International AB. All rights reserved. Qlik®, Qlik Sense®, QlikView®, QlikTech®, Qlik Cloud®, Qlik DataMarket®, Qlik Analytics Platform®, Qlik NPrinting®, Qlik Connectors®, Qlik GeoAnalytics®, Qlik Core®, Associative Difference®, Lead with Data™, Qlik Data Catalyst™, Qlik Associative Big Data Index™ and the QlikTech logos are trademarks of QlikTech International AB that have been registered in one or more countries. Other marks and logos mentioned herein are trademarks or registered trademarks of their respective owners.

---

<b>1 Introduction</b>	<b>11</b>
1.1 Conventions	11
Style coding	11
Environment variables	12
1.2 Additional documentation	12
<b>2 Qlik Sense Repository Service</b>	<b>13</b>
2.1 Update user	13
Procedure	13
Success	13
2.2 Delete user	13
Procedure	13
Success	13
2.3 Open app	14
Procedure	14
Success	14
Errors	14
2.4 Create app	14
Procedure	14
Success	14
Errors	15
2.5 Delete app	15
Procedure	15
Success	15
Errors	15
2.6 Publish app	15
Procedure	15
Success	16
2.7 Export app	16
Procedure	16
Success	16
Errors	16
2.8 Import app	16
Procedure	16
Success	17
Errors	17
2.9 Reload app	18
Procedure	18
Success	18
2.10 Duplicate app	18
Procedure	18
Success	18
Errors	19
2.11 Add app object	19
Procedure	19

---

---

Success .....	19
2.12 Update app object .....	19
Procedure .....	19
Success .....	19
2.13 Delete app object .....	20
Procedure .....	20
Success .....	20
2.14 Publish app object .....	20
Procedure .....	20
Success .....	20
2.15 Unpublish app object .....	20
Procedure .....	20
Success .....	20
2.16 Add extension .....	21
Procedure .....	21
Success .....	21
2.17 Create extension .....	21
Procedure .....	21
Success .....	21
2.18 Upload extension .....	21
Procedure .....	21
Success .....	21
Errors .....	22
2.19 Delete extension .....	22
Procedure .....	22
Success .....	22
2.20 Add extension content .....	22
Procedure .....	22
Success .....	22
2.21 Delete extension content .....	23
Procedure .....	23
Success .....	23
2.22 Add content library .....	23
Procedure .....	23
Success .....	23
2.23 Delete content library .....	23
Procedure .....	23
Success .....	23
2.24 Upload content library content .....	24
Procedure .....	24
Success .....	24
Errors .....	24
2.25 Delete content library content .....	24
Procedure .....	24

---

---

Success .....	24
2.26 Add user access .....	24
Procedure .....	24
Success .....	25
Errors .....	25
2.27 Update user access .....	25
Procedure .....	25
Success .....	25
2.28 Delete user access .....	25
Procedure .....	25
Success .....	26
2.29 License user access request .....	26
Procedure .....	26
Success .....	26
2.30 License user access .....	26
Procedure .....	26
Success .....	26
Errors .....	26
2.31 Add user access from license .....	27
Procedure .....	27
Success .....	27
Errors .....	27
2.32 Add app privilege .....	27
Procedure .....	27
Success .....	27
2.33 Export certificates .....	28
Procedure .....	28
Success .....	28
2.34 Download license .....	28
Procedure .....	28
Success .....	28
Errors .....	28
2.35 Add license .....	28
Procedure .....	28
Success .....	28
Errors .....	29
2.36 Update license .....	29
Procedure .....	29
Success .....	29
Errors .....	29
2.37 Delete license .....	29
Procedure .....	29
Success .....	29
2.38 Add rule .....	30

---

Procedure .....	30
Success .....	30
2.39 Update rule .....	30
Procedure .....	30
Success .....	30
2.40 Delete rule .....	30
Procedure .....	30
Success .....	30
2.41 Add stream .....	31
Procedure .....	31
Success .....	31
2.42 Delete stream .....	31
Procedure .....	31
Success .....	31
2.43 Server node registration .....	31
Procedure .....	31
Success .....	31
Errors .....	32
2.44 Server node configuration .....	32
Procedure .....	32
Success .....	32
2.45 Create task .....	32
Procedure .....	32
Success .....	32
2.46 Update task .....	32
Procedure .....	32
Success .....	33
2.47 Delete task .....	33
Procedure .....	33
Success .....	33
2.48 Start task .....	33
Procedure .....	33
Success .....	33
Errors .....	34
2.49 Stop task .....	34
Procedure .....	34
Success .....	34
2.50 Synchronize user directory .....	35
Procedure .....	35
Success .....	35
2.51 Start repository .....	35
Procedure .....	35
Success .....	35
Errors .....	35

---

2.52 Stop repository .....	36
Procedure .....	36
Success .....	36
2.53 Check service status .....	36
Procedure .....	36
Success .....	36
Errors .....	36
2.54 Load plugin .....	36
Procedure .....	36
Success .....	37
2.55 Audit rules .....	37
Procedure .....	37
Success .....	37
2.56 Audit security .....	37
Procedure .....	37
Success .....	37
2.57 Audit license .....	37
Procedure .....	37
Success .....	37
2.58 Audit license rule .....	38
Procedure .....	38
Success .....	38
2.59 License maintenance .....	38
Procedure .....	38
Success .....	38
2.60 Distribute certificate .....	38
Procedure .....	38
Success .....	38
Errors .....	39
<b>3 Qlik Sense Proxy Service .....</b>	<b>40</b>
3.1 Start proxy .....	40
Procedure .....	40
Success .....	40
Errors .....	40
3.2 Stop proxy .....	41
Procedure .....	41
Success .....	41
3.3 Open connection .....	41
Procedure .....	41
Success .....	41
Errors .....	41
3.4 Close connection .....	43
Procedure .....	43
Success .....	43

---

Errors .....	43
3.5 Start session .....	44
Procedure .....	44
Success .....	44
Errors .....	44
3.6 Stop session .....	45
Procedure .....	45
Success .....	45
Errors .....	45
3.7 Log out .....	45
Procedure .....	45
Success .....	45
Errors .....	46
3.8 Log in .....	46
Procedure .....	46
Success .....	46
Errors .....	46
3.9 Install certificate .....	47
Procedure .....	47
Success .....	47
<b>4 Qlik Sense Scheduler Service .....</b>	<b>48</b>
4.1 Start task .....	48
Procedure .....	48
Success .....	48
Errors .....	48
4.2 Finish task .....	51
Procedure .....	51
Success .....	51
Errors .....	51
4.3 Execute task .....	52
Procedure .....	52
Success .....	52
Errors .....	52
4.4 Start master .....	53
Procedure .....	53
Success .....	53
4.5 Start slave .....	54
Procedure .....	54
Success .....	54
4.6 Resume master .....	54
Procedure .....	54
Success .....	54
4.7 Resume slave .....	54
Procedure .....	54



---

Success .....	54
4.8 Read initial settings .....	54
Procedure .....	54
Success .....	55
4.9 Log hardware information at the startup of the service .....	55
Procedure .....	55
Success .....	55
Errors .....	55
4.10 Stop master .....	55
Procedure .....	55
Success .....	55
4.11 Stop slave .....	56
Procedure .....	56
Success .....	56
4.12 Pause master .....	56
Procedure .....	56
Success .....	56
4.13 Pause slave .....	56
Procedure .....	56
Success .....	56
4.14 Settings change for slave .....	56
Procedure .....	56
Success .....	57
<b>5 Qlik Sense Engine Service .....</b>	<b>58</b>
5.1 Open app .....	58
Procedure .....	58
Success .....	58
Errors .....	58
5.2 Create app .....	59
Procedure .....	59
Success .....	59
Errors .....	59
5.3 Delete app .....	60
Procedure .....	60
Success .....	60
Errors .....	60
5.4 Export app .....	60
Procedure .....	60
Success .....	60
Errors .....	61
5.5 Import app .....	61
Procedure .....	61
Success .....	61
Errors .....	61

5.6 Reload app .....	62
Procedure .....	62
Success .....	62
5.7 Duplicate app .....	62
Procedure .....	62
Success .....	62
Errors .....	62
5.8 Publish app .....	63
Procedure .....	63
Success .....	63
Errors .....	63
5.9 Unpublish app .....	63
Procedure .....	63
Success .....	64
Errors .....	64
5.10 Replace app .....	64
Success .....	64
5.11 Start engine .....	64
Procedure .....	64
Success .....	65
Errors .....	65
5.12 Stop engine .....	65
Procedure .....	65
Success .....	65
Errors .....	66


# 1 Introduction

Troubleshooting may be needed when Qlik Sense does not behave as expected (for example, if the system responds with an error message that needs further investigation or does not respond at all when an error occurs).

The log messages produced by Qlik Sense provide important information that can be used to detect security incidents, operational problems, and policy violations.

The description of how to troubleshoot Qlik Sense using logs is based on "use cases", each of which corresponds to a typical user or system action, such as opening an app or stopping a task.

Each use case is described using the sections listed in the following table.

Section	Description
Procedure	This section lists the actions that are performed (and logged) by Qlik Sense when the use case is carried out.
Success	This section lists the log files to which Qlik Sense writes log entries in case of success.
Errors	<p>This section lists errors that may occur when the use case is performed.</p> <p>It is recommended to check the <code>&lt;MachineName&gt;_Service_&lt;Service&gt;.txt</code> file in case of an error as all errors are logged in this file. Each error section also includes a description of the actions to perform in case of an error.</p> <div style="border: 1px solid gray; padding: 5px; margin-top: 10px;">  <p><i>For some use cases, there is no Error section. Information on errors will be added in future releases of the Qlik Sense documentation.</i></p> </div>

This document is derived from the online help for Qlik Sense. It is intended for those who want to read parts of the help offline or print pages easily, and does not include any additional information compared with the online help.

## 1.1 Conventions

The following conventions are used in the documentation for Qlik Sense.

### Style coding

- Menu commands and dialog options are written in **bold**.
- File names and paths are written in *Italics*.
- Sample code is written in Lucida Console.

## Environment variables

The paths used in the documentation for Qlik Sense may use environment variables. The variables and the equivalent paths in the Microsoft Windows operating system are listed below.

<b>Environment variable</b>	<b>Microsoft Windows</b>
<i>%LocalAppData%</i>	<i>C:\Users\&lt;username&gt;\AppData\Local</i>
<i>%ProgramData%</i>	<i>C:\ProgramData</i>
<i>%ProgramFiles%</i>	<i>C:\Program Files</i>
<i>%UserProfile%</i>	<i>C:\Users\&lt;username&gt;</i>

## 1.2 Additional documentation

Besides this document, the following related documentation is available for Qlik Sense:


- Manage Qlik Sense sites: Describes how to manage a Qlik Sense site.
- Plan and deploy Qlik Sense: Describes Qlik Sense and provides reference information on the architecture, security, logging, and licensing.

## 2 Qlik Sense Repository Service

This section describes how to use the Qlik Sense logs to troubleshoot problems related to the Qlik Sense Repository Service (QRS).

---

### See also:

 [Introduction \(page 11\)](#)

### 2.1 Update user

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS verifies that the update does not disable any service account that the user is allowed to update.
3. The QRS updates the user.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### 2.2 Delete user

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS verifies that the user to be deleted is not a service account or the last user with root admin access.
3. The QRS removes the user.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### 2.3 Open app

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS checks that the app exists.
3. The QRS checks that the user is allowed to open the app.
4. The QRS sends a request to the Qlik Sense Engine Service (QES).
5. The QES checks the access to the data set in the app.
6. The QES loads the app and returns it.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*

#### Errors

##### Your account is inactive

<b>Description</b>	Command=Open app;Result=403;ResultText=Error: Security
<b>Message</b>	Your account is inactive. Contact your administrator to activate it. (HTTP code: 403)
<b>Action</b>	Contact the Qlik Sense system administrator to obtain the correct access rights.

### 2.4 Create app

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS checks that the user is allowed to create an app.
3. The QRS sends a request to the Qlik Sense Engine Service (QES).
4. The QES creates the app.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### Errors

#### Forbidden

<b>Description</b>	Command=Create app;Result=403;ResultText=Error: Security
<b>Message</b>	Forbidden (HTTP code: 403)
<b>Action</b>	The user is not allowed to create an app.

## 2.5 Delete app

### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS marks the app as deleted in the repository database.
3. The QRS requests the Qlik Sense Engine Service (QES) to delete the app from disk.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*

### Errors

#### The Delete operation failed

<b>Description</b>	Command=Delete app;Result=400;ResultText=Error: PersistenceFailed
<b>Message</b>	The Delete operation failed (HTTP code: 400)
<b>Action</b>	Probable cause of error: The app does not exist.

## 2.6 Publish app

### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS links the app to a stream.
3. The QRS checks if the published app is renamed.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

## 2.7 Export app

### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS provides the Qlik Sense Engine Service (QES) with the information needed to find the app.
3. The app ID is downloaded from the QES to the client.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*

### Errors

#### Resource not found

<b>Description</b>	Command=Export app;Result=9003;ResultText=Error: EngineQix
<b>Message</b>	Resource not found Parameters: QVF header error (read) (HTTP code: 400)
<b>Actions</b>	Proceed as follows: <ol style="list-style-type: none"><li>1. Check that the resource exists in the Qlik Management Console (QMC).</li><li>2. Verify that you can open the app in the Qlik Sense hub.</li><li>3. Check if the QVF file exists on the central node or on a rim node in the Qlik Sense site.</li></ol>

## 2.8 Import app

### Procedure

Qlik Sense performs the following procedure:



1. A request is sent to the Qlik Sense Repository Service (QRS).
2. If you have access to the Qlik Sense system where the app was created, the following happens:
  - a. The QRS stores metadata in the repository database.
  - b. The QRS contacts the Qlik Sense Engine Service (QES).
  - c. The app is migrated (if necessary).
3. The QES imports the app (including objects) and persists it.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*

### Errors

#### Corrupt data

<b>Description</b>	Command=Import app;Result=11;ResultText=Error: EngineQix
<b>Message</b>	Corrupt data Parameters: QVF File corrupt (HTTP code: 500)
<b>Action</b>	The format of the app is invalid and it cannot be imported.

#### Write failed

<b>Description</b>	Command=Import app;Result=9000;ResultText=Error: EngineQix
<b>Message</b>	Write Failed Parameters: REST client response error (HTTP code: 500)
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"><li>1. Check that the Qlik Sense Engine Service (QES) is up and running as the request may have timed out.</li><li>2. Verify that you can open the app where it was created in Qlik Sense.</li><li>3. Verify that you can load app data in the data load editor and that you can reload the app in the Qlik Management Console (QMC).</li><li>4. Check the reload history of the app.</li></ol>

#### Resource not found

<b>Description</b>	Command=Import app;Result=2;ResultText=Error: EngineQix
<b>Message</b>	Resource not found Parameters: QVF object error (HTTP code: 500)
<b>Action</b>	Provide a working QVF file for import.

### 2.9 Reload app

The data in an app can be reloaded in different ways:

- The user manually reloads the app data in the Qlik Management Console (QMC).
- The app data is reloaded by a scheduled task in the repository database.
- The user manually loads data in the data load editor. Information on such reloads is logged in *<MachineName>\_AuditActivity\_Engine.txt*.

*Reload app (page 62)*

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS contacts the Qlik Sense Scheduler Service (QSS).
3. The QSS starts the reload task.
4. The QSS contacts the Qlik Sense Engine Service (QES) and initiates a reload of the app.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*
- *<MachineName>\_AuditActivity\_Scheduler.txt*

### 2.10 Duplicate app

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS checks that the app exists.
3. The QRS checks that the user is allowed to duplicate the app.
4. The QRS sends a request to the Qlik Sense Engine Service (QES).
5. The QES makes a copy of the app.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*

### Errors

#### Resource not found

<b>Description</b>	Command=Duplicate app;Result=-2146233074;ResultText=Error: Resource not found [LOCERR_PERSISTENCE_NOT_FOUND]:Resource not found(QVF header error (read))
<b>Message</b>	Duplicate failed. App 'Test' duplicated to 'Test(1)'. Additional info: 'Resource not found [LOCERR_PERSISTENCE_NOT_FOUND]:Resource not found(QVF header error (read))'
<b>Action</b>	Probable cause of error: The app has been deleted from disk. Try to import the app again.

### 2.11 Add app object

#### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Engine Service (QES) contacts the Qlik Sense Repository Service (QRS).
2. The app object is added to the repository database in a bulk operation.
3. The QES persists the app object.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.12 Update app object

#### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Engine Service (QES) contacts the Qlik Sense Repository Service (QRS).
2. The app object is updated in the repository database in a bulk operation.
3. The QES updates the persisted app object.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.13 Delete app object

#### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Engine Service (QES) contacts the Qlik Sense Repository Service (QRS).
2. The app object is set to be deleted in the repository database in a bulk operation.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.14 Publish app object

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS sets the app object as published in the repository database.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.15 Unpublish app object

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS sets the app object as unpublished in the repository database.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.16 Add extension

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS adds the extension.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.17 Create extension

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS creates the extension.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.18 Upload extension

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS uploads the extension.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### Errors

The process cannot access the file

<b>Description</b>	Command=Upload extension;Result=500;ResultText=Error: IO
<b>Message</b>	The process cannot access the file '<Filename>' because it is being used by another process. (HTTP code: 500)
<b>Action</b>	Try the following measures: <ul style="list-style-type: none"><li>• Wait for any ongoing process in the Qlik Management Console (QMC) to finish and then upload the extension again.</li><li>• Restart the QRS.</li></ul>

### 2.19 Delete extension

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS deletes the extension.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.20 Add extension content

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS adds the extension content.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.21 Delete extension content

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS deletes the extension content.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.22 Add content library

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS adds the content library.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.23 Delete content library

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS deletes the content library.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.24 Upload content library content

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS uploads the content library content.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

#### Errors

Exception of type 'Qlik.Sense.Common.Exceptions.ConflictException' was thrown

<b>Description</b>	Command=Upload content library content;Result=409;ResultText=Error: Conflict
<b>Message</b>	Exception of type 'Qlik.Sense.Common.Exceptions.ConflictException' was thrown. (HTTP code: 409)
<b>Action</b>	The library content already exists. Rename or replace the library content.

### 2.25 Delete content library content

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS deletes the content library content.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.26 Add user access

#### Procedure

Qlik Sense performs the following procedure:



1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS verifies that the access can be added.
3. The QRS adds the access.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### Errors

#### Validation failed

<b>Description</b>	Command=Add User Access;Result=400;ResultText=Error: Bad Request
<b>Message</b>	<BrokenRules>
<b>Action</b>	Make sure that you are allowed to add the access.

## 2.27 Update user access

### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS verifies that the access can be updated.
3. The QRS updates the access.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

## 2.28 Delete user access

### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS verifies that the access can be deleted.
3. The QRS deletes the access.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

## 2.29 License user access request

### Procedure

Qlik Sense performs the following procedure:

- Qlik Sense checks that the user has a valid access pass. This is done every time a user requests access to a resource (such as an app).

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

## 2.30 License user access

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) checks the Qlik Sense license to determine if the user can be given an access pass.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### Errors

#### Login access denied

<b>Description</b>	Command=License user access;Result=403;ResultText=Error: Access denied
<b>Message</b>	Login access denied for SessionID: 'xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx', Hostname: 'xx.xx.xx.xx', OperationType: 'UsageDenied'
<b>Action</b>	Contact the Qlik Sense system administrator to obtain the correct access rights.

### 2.31 Add user access from license

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS verifies that the user access can be added.
3. The QRS identifies the user access.
4. The QRS adds the user access.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

#### Errors

##### Cannot add user access

<b>Description</b>	Command=Add user access from license;Result=400;ResultText=Error: Bad Request
<b>Message</b>	<BrokenRules>
<b>Action</b>	Check the following: <ul style="list-style-type: none"><li>• That the license has not expired</li><li>• That the license is not blacklisted</li><li>• That there are enough tokens available in the Qlik Sense license</li></ul>

### 2.32 Add app privilege

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) adds app privileges to the appropriate user groups.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*.

### 2.33 Export certificates

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) exports the certificates.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### 2.34 Download license

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) validates the license and downloads it.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

#### Errors

##### License: Invalid serial number or control number

<b>Description</b>	Command=Download license;Result=400;ResultText=Error: ValidationFailed
<b>Message</b>	License: Invalid serial number or control number (HTTP code: 400)
<b>Action</b>	Provide a valid Qlik Sense license.

### 2.35 Add license

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) adds the license provided by the user.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### Errors

#### License: Invalid serial number or control number

<b>Description</b>	Command=Update license;Result=400;ResultText=Error: ValidationFailed
<b>Message</b>	License: Invalid serial number or control number (HTTP code: 400)
<b>Action</b>	Provide a valid Qlik Sense license.

## 2.36 Update license

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) adds the license provided by the user.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### Errors

#### License: Invalid serial number or control number

<b>Description</b>	Command=Update license;Result=400;ResultText=Error: ValidationFailed
<b>Message</b>	License: Invalid serial number or control number (HTTP code: 400)
<b>Action</b>	Provide a valid Qlik Sense license.

## 2.37 Delete license

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) deletes the license.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.38 Add rule

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS adds the rule.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.39 Update rule

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS updates the rule.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.40 Delete rule

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS deletes the rule.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.41 Add stream

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS adds the stream.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.42 Delete stream

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS deletes the stream.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.43 Server node registration

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) establishes a connection to the specified server address.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### Errors

Cannot reach the following remote host when distributing certificates

<b>Description</b>	Command=Server node registration;Result=400;ResultText=Error: REST
<b>Message</b>	Cannot reach the following remote host when distributing certificates: http://<ServerAddress>:4444/setup/certificateDistribution (HTTP code: 400)
<b>Action</b>	Provide a valid server address.

## 2.44 Server node configuration

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) updates the configuration for a server node.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- <MachineName>\_AuditActivity\_Repository.txt

## 2.45 Create task

### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Repository Service (QRS) creates a task and stores it in the repository database.
2. The task information is synchronized by the QRS to the rim nodes.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- <MachineName>\_AuditActivity\_Repository.txt

## 2.46 Update task

### Procedure

Qlik Sense performs the following procedure:



1. The Qlik Sense Repository Service (QRS) updates the task and stores it in the repository database.
2. The updated task information is synchronized by the QRS to the rim nodes.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

## 2.47 Delete task

### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Repository Service (QRS) deletes the task and removes it from the repository database.
2. The deletion of the task is synchronized by the QRS to the rim nodes.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

## 2.48 Start task

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) requests the Qlik Sense Scheduler Service (QSS) to execute the task.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### Errors

#### Resource not found

<b>Description</b>	Command=Start task;Result=-1;ResultText=Error: -----System.Net.WebException: Unable to connect to the remote server ---> System.Net.Sockets.SocketException: No connection could be made because the target machine actively refused it 127.0.0.1:5050 at System.Net.Sockets.Socket.EndConnect(IAsyncResult asyncResult) at System.Net.ServicePoint.ConnectSocketInternal(Boolean connectFailure, Socket s4, Socket s6, Socket& socket, IPAddress& address, ConnectSocketState state, IAsyncResult asyncResult, Exception& exception) --- End of inner exception stack trace --- at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult, TransportContext& context) at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult) at System.Net.WebClient.UploadBitsRequestCallback(IAsyncResult result)-----
<b>Message</b>	Trailing task exception in SchedulerClient.StartTask('1xx1111x-111x-111x-1xx1-111xxx1x1x11'): '↓-----↓System.Net.WebException: Unable to connect to the remote server ---> System.Net.Sockets.SocketException: No connection could be made because the target machine actively refused it 127.0.0.1:5050↵ ↓ at System.Net.Sockets.Socket.EndConnect(IAsyncResult asyncResult)↵ ↓ at System.Net.ServicePoint.ConnectSocketInternal(Boolean connectFailure, Socket s4, Socket s6, Socket& socket, IPAddress& address, ConnectSocketState state, IAsyncResult asyncResult, Exception& exception)↵ ↓ --- End of inner exception stack trace ---↵ ↓ at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult, TransportContext& context)↵ ↓ at System.Net.HttpWebRequest.EndGetRequestStream(IAsyncResult asyncResult)↵ ↓ at System.Net.WebClient.UploadBitsRequestCallback(IAsyncResult result)↓-----'
<b>Action</b>	Check that the Qlik Sense Scheduler Service (QSS) is up and running.

## 2.49 Stop task

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) requests the Qlik Sense Scheduler Service (QSS) to stop the execution of the task.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.50 Synchronize user directory

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) synchronizes the user directory.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*

### 2.51 Start repository

#### Procedure

Qlik Sense performs the following procedure:

1. Validate the command line parameters for the QRS.
2. The QRS loads the logging framework.
3. The QRS validates the certificates.
4. The QRS establishes a connection to the Qlik Sense Repository Database (QRD).
5. If needed, the QRS performs the migration steps of the repository database schema.
6. Hardware information is collected.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Repository.txt*

#### Errors

##### Certificates are not correctly installed

<b>Description</b>	Command=Start Repository;Result=-1;ResultText=Certificates are not correctly installed
<b>Message</b>	Initializing certificates.
<b>Action</b>	The error is related to an invalid certificate. Contact your system administrator for support regarding the certificates.

### 2.52 Stop repository

#### Procedure

Qlik Sense performs the following procedure:

1. The internal services are stopped.
2. The background threads are stopped.
3. All Qlik Sense services, except for the Qlik Sense Repository Database (QRD), are stopped.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Repository.txt*

### 2.53 Check service status

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) checks that the communication with the specified Qlik Sense service is working.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Repository.txt*

#### Errors

The following service account does not exist

<b>Description</b>	Command=Check service status;Result=403;ResultText=Error: Security
<b>Message</b>	The following service account does not exist: <Service.Name> (HTTP code: 403)
<b>Action</b>	Restart the specified Qlik Sense service.

### 2.54 Load plugin

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) loads the specified plugins during the installation process.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Repository.txt*

## 2.55 Audit rules

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) fetches the specified rules from the repository database.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

## 2.56 Audit security

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) fetches the specified security rules from the repository database.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

## 2.57 Audit license

### Procedure

Qlik Sense performs the following procedure:

- A Qlik Sense Repository Service (QRS) thread runs in the background.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### 2.58 Audit license rule

#### Procedure

Qlik Sense performs the following procedure:

- A Qlik Sense Repository Service (QRS) thread runs in the background and monitors the license usage.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### 2.59 License maintenance

#### Procedure

Qlik Sense performs the following procedure:

- A Qlik Sense Repository Service (QRS) thread runs in the background.  
The thread continuously checks the proxy sessions linked to the license in order to:
  - Prevent users from over-consuming license tokens
  - Check if any changes linked to the license prevent the user from consuming an access pass
  - Check if the proxy session has timed out for the access pass

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Repository.txt*

### 2.60 Distribute certificate

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Repository Service (QRS) distributes a certificate to the rim node.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Repository.txt*

### Errors


#### Certificate distribution failed

<b>Description</b>	Command=Distribute certificate;Result=-2146233088;ResultText=Error: Cannot reach the following remote host when distributing certificates: http://<ServerAddress>:4444/setup/certificateDistribution
<b>Message</b>	Certificate distribution failed
<b>Action</b>	Provide a valid server address.

## 3 Qlik Sense Proxy Service

This section describes how to use the Qlik Sense logs to troubleshoot problems related to the Qlik Sense Proxy Service (QPS).

### See also:

 [Introduction \(page 11\)](#)

### 3.1 Start proxy

#### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Proxy Service (QPS) creates a proxy.
2. The QPS starts the proxy.
3. Hardware information is collected.
4. The QPS writes the hardware information in the logs.
5. The QPS installs the certificates.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- `<MachineName>_AuditActivity_Proxy.txt`
- `<MachineName>_Service_Proxy.txt`

#### Errors

##### Error occurred while retrieving hardware information

<b>Description</b>	Command=Start proxy;Result= -1;ResultText=Error occurred while retrieving hardware information
<b>Message</b>	The message may vary.
<b>Action</b>	Check the registry settings.

##### Proxy will not be listening to port '443'

<b>Description</b>	Port collisions may occur when internal services are started during the startup of the QPS: Command=Start internal service;Result=-2147467259;ResultText=Error: Only one usage of each socket address
--------------------	--



<b>Message</b>	Proxy will not be listening to port '443' (most likely bound by another process)
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"><li>1. Make sure that port 443 is available for Qlik Sense to use.</li><li>2. Restart the Qlik Sense services.</li></ol>

### 3.2 Stop proxy

#### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Proxy Service (QPS) notifies the Qlik Sense Repository Service (QRS).
2. The QPS settings and notification poller threads are stopped.
3. The QPS checks that the QRS has installed the certificate.
4. The internal services are stopped.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Proxy.txt*

### 3.3 Open connection

#### Procedure

Qlik Sense performs the following procedure:

- Open a socket to the Qlik Sense Engine Service (QES). This happens every time a user opens an app in Qlik Sense.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Proxy.txt*
- *<MachineName>\_Service\_Proxy.txt*

#### Errors

##### Web socket connection could not be opened

<b>Description</b>	Command=Open connection;Result=400;ResultText=Error: Web socket connection could not be opened
--------------------	--

<b>Message</b>	Connection '<ID of connection>' has been transferred to a streaming state to send a single error message
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"> <li>1. Check if the QES is up and running. If not, start the QES.</li> <li>2. Check that it is possible to connect to the QES.</li> </ol>

#### Web exception: Protocol error: Response stream exists

<b>Description</b>	Command=Open connection;Result=400;ResultText=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	Web exception: Protocol error: Response stream exists. Remote endpoint '{0}' + various exception message
<b>Action</b>	Check the following: <ul style="list-style-type: none"> <li>• That the QES is up and running.</li> <li>• That you can connect to the QES on port 4747 locally on the server.</li> </ul>

#### Web exception: Protocol error: No response stream exists

<b>Description</b>	Command=Open connection;Result=400;ResultText=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	Web exception: Protocol error: No response stream exists when contacting '{0}' + various exception message
<b>Action</b>	Check the following: <ul style="list-style-type: none"> <li>• That the QES is up and running.</li> <li>• That you can connect to the QES on port 4747 locally on the server.</li> </ul>

#### Web exception when contacting service uri

<b>Description</b>	Command=Open connection;Result=400;ResultText=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	Web exception when contacting service uri {0} + various exception message
<b>Action</b>	Check that service at URI is available.

#### Connection has been transferred to a streaming state to send a single error message

<b>Description</b>	Command=Open connection;Result=400;ResultText=<different from zero>;ResultText=Error: <error message>
--------------------	---

<b>Message</b>	Connection '{0}' has been transferred to a streaming state to send a single error message + various exception message
<b>Action</b>	Check the following: <ul style="list-style-type: none"> <li>• That the QES is up and running.</li> <li>• That you can connect to the QES on port 4747 locally on the server.</li> </ul>

### Web socket connection could not be opened

<b>Description</b>	Command=Open connection;Result=400;ResultText=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	Web socket connection could not be opened
<b>Action</b>	Check the following: <ul style="list-style-type: none"> <li>• That the QES is up and running.</li> <li>• That you can connect to the QES on port 4747 locally on the server.</li> </ul>

## 3.4 Close connection

### Procedure

Qlik Sense performs the following procedure:

- Close the socket to the Qlik Sense Engine Service (QES). This happens every time a user closes an app (for example, by closing a tab in a browser).

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Proxy.txt*

### Errors

#### Web socket connection could not be opened

<b>Description</b>	Command=Open connection;Result=400;ResultText=Error: Web socket connection could not be opened
<b>Message</b>	Connection '<ID of connection>' has been transferred to a streaming state to send a single error message
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"> <li>1. Check if the QES is up and running. If not, start the QES.</li> <li>2. Check that it is possible to connect to the QES.</li> </ol>

Command=Open connection;Result=<different from zero>

<b>Description</b>	Command=Open connection;Result=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	Search for any log message related to Close connection.  If you find a log entry similar to the one described above, evaluate the information in the Message field.
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"> <li>1. Analyze the memory consumption and run netstat (TCP sockets in illegal state). If you encounter any issues, restart the QPS as connections (that is, web sockets to the QES) are disposed by the Qlik Sense Proxy Service (QPS).</li> <li>2. If the problem persists, check that the QES is up and running.</li> <li>3. Contact Qlik support.</li> </ol>

### 3.5 Start session

#### Procedure

Qlik Sense performs the following procedure:

- A new proxy session starts when a user accesses Qlik Sense.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- <MachineName>\_AuditActivity\_Proxy.txt

#### Errors

##### Cannot start session

<b>Description</b>	If a user cannot start a new proxy session, it is most likely related to: <ul style="list-style-type: none"> <li>• The configuration of the Qlik Sense Proxy Service (QPS)</li> <li>• Installation-specific issues</li> </ul> Search for any log message with the following information:  Command=Start session;Result=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	If you find a log entry similar to the one described above, evaluate the information in the Message field.
<b>Action</b>	If there are no log entries for Start session, check the configuration of the QPS by examining the log entries for the Start proxy command.

### 3.6 Stop session

#### Procedure

Qlik Sense performs the following procedure:

- A proxy session stops when it times out or when the user closes the browser tab where the session is running.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Proxy.txt*

#### Errors

##### Cannot stop session

<b>Description</b>	Search for any log message with the following information:  Command=Stop session;Result=<different from zero>;ResultText=Error: <error message>
<b>Message</b>	If you find a log entry similar to the one described above, evaluate the information in the Message field.
<b>Action</b>	If there are no log entries for Stop session, check the configuration of the Qlik Sense Proxy Service (QPS) by examining the log entries for the Start proxy command.  As a last resort, restart the QPS.

### 3.7 Log out

#### Procedure

Qlik Sense performs the following procedure:

- The user is logged out when the proxy session times out or when the user actively logs out.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Proxy.txt*

## Errors

Command=Logout;Result=<return code not zero>

<b>Description</b>	Command=Logout;Result=<return code not zero>;ResultText=Error: <error message>
<b>Message</b>	If you find a log entry similar to the one described above, evaluate the information in the Message field.
<b>Action</b>	<p>Proceed as follows, if a user repeatedly experiences problem when logging out:</p> <ol style="list-style-type: none"> <li>1. Restart the Qlik Sense Proxy Service (QPS).</li> <li>2. Analyze the status of the sockets by running netstat and search for sockets in CLOSE_WAIT, SYN_SENT, or FIN_WAIT_2 state.</li> </ol>

## 3.8 Log in

### Procedure

Qlik Sense performs the following procedure:

1. Check if the user is linked to a valid user directory.
2. Check if the user exists in the repository database.
3. Check if the user has access to Qlik Sense.
4. Check if the user has access linked to the license.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- <MachineName>\_AuditSecurity\_Proxy.txt

## Errors

### Logon failed

<b>Description</b>	Command=Login;ResultCode=403;ResultText=Error: Access Denied
<b>Message</b>	Login failed for user '<username>' wrong credentials?
<b>Action</b>	<p>Proceed as follows:</p> <ul style="list-style-type: none"> <li>• Check that the user exists in the repository database.</li> <li>• Check that the user has access rights to Qlik Sense linked to a license.</li> <li>• Check that the user is not blocked.</li> </ul>

### 3.9 Install certificate

#### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Proxy Service (QPS) waits for the certificates to be installed in the repository database.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditSecurity\_Proxy.txt*

## 4 Qlik Sense Scheduler Service

This section describes how to use the Qlik Sense logs to troubleshoot problems related to the Qlik Sense Scheduler Service (QSS).

### See also:

 [Introduction \(page 11\)](#)

### 4.1 Start task

#### Procedure

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The master Qlik Sense Scheduler Service (QSS) on the central node communicates with each slave QSS to see which ones are available to perform the task.
3. The task is given to the currently available slave QSS with the least load according to the load balance rules.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- `<MachineName>_AuditActivity_Scheduler.txt`
- `<MachineName>_AuditActivity_Repository.txt`

#### Errors

##### Task not found

<b>Description</b>	Command=Start task;Result=5;ResultText=Error: Failed
<b>Message</b>	Task not found
<b>Action</b>	<p>Try the following measures:</p> <ul style="list-style-type: none"> <li>• If the repository database is accessible, search the database for the task.</li> <li>• Check if the task has been successfully synchronized to the rim nodes.</li> <li>• Create a new, identical task and see if it works.</li> </ul>

##### Scheduler is not licensed

<b>Description</b>	Command=Start task;Result=10;ResultText=Error: Failed
--------------------	---



<b>Message</b>	Scheduler is not licensed. Not allowed to run Tasks
<b>Action</b>	Provide a valid Qlik Sense license.

### Scheduler is not Master

<b>Description</b>	Command=Start task;Result=15;ResultText=Error: Failed
<b>Message</b>	Scheduler is not Master. Not allowed to run Tasks
<b>Action</b>	Make sure that the QSS on the central node is “Master” or “Master and Slave”.

### Found active session for Task

<b>Description</b>	Command=Start task;Result=20;ResultText=Error: Failed
<b>Message</b>	Found active session for Task. Task not started
<b>Action</b>	Try the following measures: <ul style="list-style-type: none"> <li>• Wait until the task has completed.</li> <li>• Stop the task.</li> <li>• Restart the QSS.</li> </ul>

### Task disabled

<b>Description</b>	Command=Start task;Result=25;ResultText=Error: Failed
<b>Message</b>	Task disabled
<b>Action</b>	Enable the task.

### TaskExecutionSession already exists

<b>Description</b>	Command=Start task;Result=30;ResultText=Error: Failed
<b>Message</b>	TaskExecutionSession already exists for App <App.Name>
<b>Action</b>	Try the following measures: <ul style="list-style-type: none"> <li>• Wait until the task that is reloading the app has completed.</li> <li>• Stop the task that is reloading the app.</li> <li>• Restart the QSS.</li> </ul>

### App is not enabled

<b>Description</b>	Command=Start task;Result=35;ResultText=Error: Failed
<b>Message</b>	App <App.Name> is not enabled
<b>Action</b>	The app was not correctly enabled by the Migration Service. Try to re-migrate the app.

### No slave nodes found to execute Task

<b>Description</b>	Command=Start task;Result=40;ResultText=Error: Failed
<b>Message</b>	No slave-nodes found to execute Task: <Task.Name>
<b>Action</b>	<p>Proceed as follows:</p> <ol style="list-style-type: none"> <li>1. Check the status of the slave nodes to determine if they can perform a reload.</li> <li>2. Check that the app to reload exists on the slave nodes.</li> </ol>

### Unable to create TaskExecutionSession

<b>Description</b>	Command=Start task;Result=45;ResultText=Error: Failed
<b>Message</b>	Unable to create TaskExecutionSession
<b>Action</b>	Restart the QSS.

### Unexpected exception when starting task

<b>Description</b>	Command=Start task;Result=50;ResultText=Error: Failed
<b>Message</b>	Unexpected exception when starting task. Exception message: <Exception.Message>
<b>Action</b>	Restart the QSS.

### Unexpected exception when trying to start task

<b>Description</b>	Command=Start task;Result=55;ResultText=Error: Failed
<b>Message</b>	Unexpected exception when trying to start task. Exception message: <Exception.Message>
<b>Action</b>	Restart the QSS.

### Max number of retries reached for task

<b>Description</b>	Command=Start task;Result=80;ResultText=Error: Failed
<b>Message</b>	Max number of retries (<Amount>) reached for task (id/name) <Task.ID>/<Task.Name>
<b>Action</b>	Restart the QSS.

### Not possible to initiate retry

<b>Description</b>	Command=Start task;Result=90;ResultText=Error: Failed
<b>Message</b>	Not possible to initiate retry. TaskExecutionSession is null
<b>Action</b>	Restart the QSS.

## 4.2 Finish task

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Scheduler Service (QSS) on the slave node that performed the task communicates the result to the master QSS.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Scheduler.txt*
- *<MachineName>\_AuditActivity\_Repository.txt*

### Errors

#### Failed in TaskCompletedFail

<b>Description</b>	Command=Finished task;Result=60;ResultText=Error: Failed
<b>Message</b>	Failed in TaskCompletedFail. Forcing task to Error
<b>Action</b>	Check if the task has been successfully synchronized to the rim nodes.

#### Task finished with state <State>

<b>Description</b>	Command=Finished task;Result=65;ResultText=Error: Failed
<b>Message</b>	Task finished with state <State>
<b>Action</b>	Check if the task has been successfully synchronized to the rim nodes.

#### Failed to remove session

<b>Description</b>	Command=Finished task;Result=70;ResultText=Error: Failed
<b>Message</b>	Failed to remove session with Id <Session.ID> for TaskId <Task.ID>
<b>Action</b>	Restart the QSS.

#### Unexpected exception in TaskCompletedSuccess

<b>Description</b>	Command=Finished task;Result=50;ResultText=Error: Failed
<b>Message</b>	Unexpected exception in TaskCompletedSuccess. Exception message: <Message>
<b>Action</b>	Restart the QSS.

### Unexpected exception in TaskCompletedFail

<b>Description</b>	Command=Finished task;Result=50;ResultText=Error: Failed
<b>Message</b>	Unexpected exception in TaskCompletedFail. Exception message: <Message>
<b>Action</b>	Restart the QSS.

### Task failed

<b>Description</b>	Command=Finished task;Result=0;ResultText=Error: Failed
<b>Message</b>	Task failed
<b>Action</b>	<p>Check the task for indications of why it failed.</p> <p>If the task is a reload task, check the engine logs and script logs for indications of why the reload failed.</p>

## 4.3 Execute task

### Procedure

Qlik Sense performs the following procedure:

- The Qlik Sense Scheduler Service (QSS) logs different states in the task execution chain during the execution of a task.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- <MachineName>\_AuditActivity\_Scheduler.txt

### Errors

#### Trying to set task id to state

<b>Description</b>	Command=Task execution;Result=55;ResultText=Error: Failed
<b>Message</b>	Trying to set task (id) <Task.ID> to state <State>. Not allowed. No statechange will take place!
<b>Action</b>	<p>Proceed as follows:</p> <ol style="list-style-type: none"> <li>1. Re-run the task.</li> <li>2. Restart the QSS.</li> </ol>

### TaskExecutionSession cannot be null

<b>Description</b>	Command=Task execution;Result=90;ResultText=Error: Failed
<b>Message</b>	TaskExecutionSession cannot be null. Unable to proceed with state change due to deleted task with proposed state: <State>
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"> <li>1. Re-run the task.</li> <li>2. Restart the QSS.</li> </ol>

### Suppressed state change

<b>Description</b>	Command=Task execution;Result=85;ResultText=Error: Failed
<b>Message</b>	Suppressed state change to <State> for <Task.Name> with Id <Task.ID>
<b>Action</b>	No action is required. The state was suppressed when the task was stopped.

### Unable to get <Task.Name>

<b>Description</b>	Command=Task execution;Result=5, 75;ResultText=Error: Failed
<b>Message</b>	Unable to get <Task.Name> with Id <Task.ID> for AuditActivityLog, only logging Id and Name for Task not for App.
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"> <li>1. Re-run the task.</li> <li>2. Restart the QSS.</li> </ol>

## 4.4 Start master

### Procedure

Qlik Sense performs the following procedure:

- The master scheduler starts when the Qlik Sense Scheduler Service (QSS) starts on the central node.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- <MachineName>\_Service\_Scheduler.txt

### 4.5 Start slave

#### Procedure

Qlik Sense performs the following procedure:

- The slave scheduler starts when the Qlik Sense Scheduler Service (QSS) starts.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.6 Resume master

#### Procedure

Qlik Sense performs the following procedure:

- Resume a paused master Qlik Sense Scheduler Service (QSS).

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.7 Resume slave

#### Procedure

Qlik Sense performs the following procedure:

- Resume a paused slave Qlik Sense Scheduler Service (QSS).

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.8 Read initial settings

#### Procedure

Qlik Sense performs the following procedure:

- Log the initial settings when the Qlik Sense Scheduler Service (QSS) starts.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

## 4.9 Log hardware information at the startup of the service

### Procedure

Qlik Sense performs the following procedure:

- Log hardware information during the startup of the Qlik Sense Scheduler Service (QSS).

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### Errors

#### Failed when logging hardware information

<b>Description</b>	Command=Start scheduler;Result=50;ResultText=Error: Failed
<b>Message</b>	Failed when logging hardware information
<b>Action</b>	Restart the QSS.

## 4.10 Stop master

### Procedure

Qlik Sense performs the following procedure:

- The master scheduler stops when the Qlik Sense Scheduler Service (QSS) shuts down on the central node.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.11 Stop slave

#### Procedure

Qlik Sense performs the following procedure:

- The slave scheduler stops when the Qlik Sense Scheduler Service (QSS) shuts down.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.12 Pause master

#### Procedure

Qlik Sense performs the following procedure:

- Pause a running master Qlik Sense Scheduler Service (QSS).

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.13 Pause slave

#### Procedure

Qlik Sense performs the following procedure:

- Pause a running slave Qlik Sense Scheduler Service (QSS).

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Scheduler.txt*

### 4.14 Settings change for slave

#### Procedure

Qlik Sense performs the following procedure:



- Log the change of settings for a slave Qlik Sense Scheduler Service (QSS).

### Success


In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Scheduler.txt*

## 5 Qlik Sense Engine Service

This section describes how to use the Qlik Sense logs to troubleshoot problems related to the Qlik Sense Engine Service (QES).

### See also:

 [Introduction \(page 11\)](#)

### 5.1 Open app

#### Procedure

Qlik Sense performs the following procedure:

1. Check if the app is already open.
2. If not, load the QVF file.
3. Read the app properties from the file.
4. Read the script from the file.
5. Open the app.

#### Success

In case of success, log entries are written in the following files throughout the procedure:

- `<MachineName>_AuditActivity_Engine.txt`

#### Errors

##### Already opened in different mode

<b>Description</b>	Command=Open app;Result=1009;ResultText=Error: App already open in different mode
<b>Message</b>	<AppId>
<b>Action</b>	Make sure that the app is not opened without data when you try to open it with data or vice versa.

##### Already opened

<b>Description</b>	Command=Open app;Result=1002;ResultText=Error: App already opened
<b>Message</b>	<AppId>
<b>Action</b>	Make sure that the app is not already open.

### Invalid path

<b>Description</b>	Command=Open app;Result=4;ResultText=Error: Invalid Path
<b>Message</b>	Malformed parameters
<b>Action</b>	Make sure that the app name, password, user name, and serial number are UTF-8 encoded strings.

### Resource not found

<b>Description</b>	Command=Open app;Result=9003;ResultText=Error: Resource not found
<b>Message</b>	QVF header error (read)
<b>Action</b>	Probable cause of error: The app has been deleted from disk. Try to import the app again.

## 5.2 Create app

### Procedure

Qlik Sense performs the following procedure:

1. Check if the app name is valid.
2. Create the app.
3. Save the QVF file.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Engine.txt*

## Errors

### Invalid name

<b>Description</b>	Command=Create app;Result=3001;ResultText=Error: App save failed
<b>Message</b>	<AppName>
<b>Action</b>	Make sure that the app name does not contain any of the following: <ul style="list-style-type: none"> <li>• Special characters: &lt;:\"/&gt; </li></ul>

## 5.3 Delete app

### Procedure

Qlik Sense performs the following procedure:

1. Unload all app instances for the app.
2. Delete the QVF file.
3. Request the Qlik Sense Repository Service (QRS) to delete the file.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Engine.txt*

### Errors

#### Delete failed

<b>Description</b>	Command=Delete app;Result=9002;ResultText=Error: Delete failed
<b>Message</b>	<REST client response error>
<b>Action</b>	As there is no response from the QRS, check if it has stopped running or if it cannot process requests.

## 5.4 Export app

### Procedure

Qlik Sense performs the following procedure:

1. Open the QVF file.
2. Copy to a new QVF file.
3. Export the app contents.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Engine.txt*

## Errors

### Disk is full

<b>Description</b>	Command=Export app;Result=9009;ResultText=Error: The disk it out of space
<b>Message</b>	CopyQvfFile
<b>Action</b>	Free up some space on the disk.

### File corrupt

<b>Description</b>	Command=Export app;Result=3002;ResultText=Error: File corrupted
<b>Message</b>	CopyQvfFile
<b>Action</b>	Make sure that the file that is exported to can be written to.

## 5.5 Import app

### Procedure

Qlik Sense performs the following procedure:

1. Check if the app can be migrated to the current version of Qlik Sense (if needed).
2. Open a copy of the app to import.
3. Remove unwanted objects (such as connections) from the copy.
4. Request the Qlik Sense Repository Service (QRS) to save the app.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Engine.txt*

## Errors

### File corrupt

<b>Description</b>	Command=Import app;Result=9008;ResultText=Error: App was created in a newer product version
<b>Message</b>	Migration
<b>Action</b>	Make sure that the app was not created in a newer version of Qlik Sense.

### Internal error

<b>Description</b>	Command=Import app;Result=10;ResultText=Error: Internal error
<b>Message</b>	Could not import file (ImportApp)
<b>Action</b>	During import, the QVF file is copied to the local <code>\App</code> folder first. If this fails, the folder is probably write-protected. Therefore check for any write-protection on the folder.

## 5.6 Reload app

### Procedure

Qlik Sense performs the following procedure:

- Execute the reload script.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- `<MachineName>_AuditActivity_Engine.txt`

## 5.7 Duplicate app

### Procedure

Qlik Sense performs the following procedure:

1. Compare the app IDs of the source and target apps.
2. Request the Qlik Sense Repository Service (QRS) to open the app.
3. Copy the QVF file.
4. Import the copy of the file.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- `<MachineName>_AuditActivity_Engine.txt`

### Errors

#### Delete failed

<b>Description</b>	Command=Duplicate app;Result=8;ResultText=Error: Invalid parameters
<b>Message</b>	Could not copy file. SrcAppId and TargetAppId are identical
<b>Action</b>	Make sure that the source and target IDs are not identical.

## 5.8 Publish app

### Procedure

Qlik Sense performs the following procedure:

1. Check if the app is a session app. If so, publishing is not supported. A session app only exists in the Qlik Sense Engine Service (QES) memory and is not linked to any file.
2. Save the object in the repository database.
3. Request the Qlik Sense Repository Service (QRS) to publish the app.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- `<MachineName>_AuditActivity_Engine.txt`

### Errors

#### Access denied

<b>Description</b>	Command=Publish app;Result=8;ResultText=Error: Access Denied
<b>Message</b>	Could not save objects
<b>Action</b>	Check if the disk is write-protected as the publish operation could not flush the objects needed before publishing the app.

#### REST connection failure

<b>Description</b>	Command=Publish app;Result=18;ResultText=Error: Rest connection failed
<b>Message</b>	REST client response error
<b>Action</b>	As there is no response from the QRS, check if it has stopped running or if it cannot process requests.

## 5.9 Unpublish app

### Procedure

Qlik Sense performs the following procedure:

1. Check if the app is a session app. If so, publishing is not supported. A session app only exists in the Qlik Sense Engine Service (QES) memory and is not linked to any file.
2. Save the object in the repository database.
3. Request the Qlik Sense Repository Service (QRS) to unpublish the app.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Engine.txt*

### Errors

#### Access denied

<b>Description</b>	Command=Unpublish app;Result=8;ResultText=Error: Access Denied
<b>Message</b>	Could not save objects
<b>Action</b>	Check if the disk is write-protected as the publish operation could not flush the objects needed before unpublishing the app.

#### REST connection failure

<b>Description</b>	Command=Unpublish app;Result=18;ResultText=Error: Rest connection failed
<b>Message</b>	REST client response error
<b>Action</b>	As there is no response from the QRS, check if it has stopped running or if it cannot process requests.

## 5.10 Replace app

Qlik Sense performs the following procedure:

1. A request is sent to the Qlik Sense Repository Service (QRS).
2. The QRS updates the repository database.
3. The QRS contacts the Qlik Sense Engine Service (QES).

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_AuditActivity\_Repository.txt*
- *<MachineName>\_AuditActivity\_Engine.txt*

## 5.11 Start engine

### Procedure

Qlik Sense performs the following procedure:



1. The Qlik Sense Engine Service (QES) sets the current execution mode (that is, server or desktop).
2. The QES parses the command line arguments.
3. The QES configures the logging.
4. The QES detects the operating system and version.
5. The QES initializes the collate and memory structures.
6. The QES sets the termination handlers, starts the internal threads, and checks the license.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Engine.txt*

### Errors

#### Registration failure

<b>Description</b>	Command=Start engine;Result=-1;ResultText=Error:Server crash
<b>Message</b>	Server crashed on registration
<b>Action</b>	Proceed as follows: <ol style="list-style-type: none"><li>1. Check the status of the Migration Service.</li><li>2. Check that the default <i>\App</i> folder exists.</li><li>3. Check the status of the network controller.</li><li>4. Check the status of the network port.</li></ol>

## 5.12 Stop engine

### Procedure

Qlik Sense performs the following procedure:

1. The Qlik Sense Engine Service (QES) stops the cache trimmer.
2. The QES clears the cache.
3. The QES terminates the process handlers.
4. The QES exits the threads.

### Success

In case of success, log entries are written in the following files throughout the procedure:

- *<MachineName>\_Service\_Engine.txt*

### Errors

#### Cache trimmer never stopped

<b>Description</b>	Command=Stop engine;Result=-1;ResultText=Warning
<b>Message</b>	Cache Trimmer never stopped
<b>Action</b>	Contact Qlik support.