Relativity®

Relativity Secret Store Guide

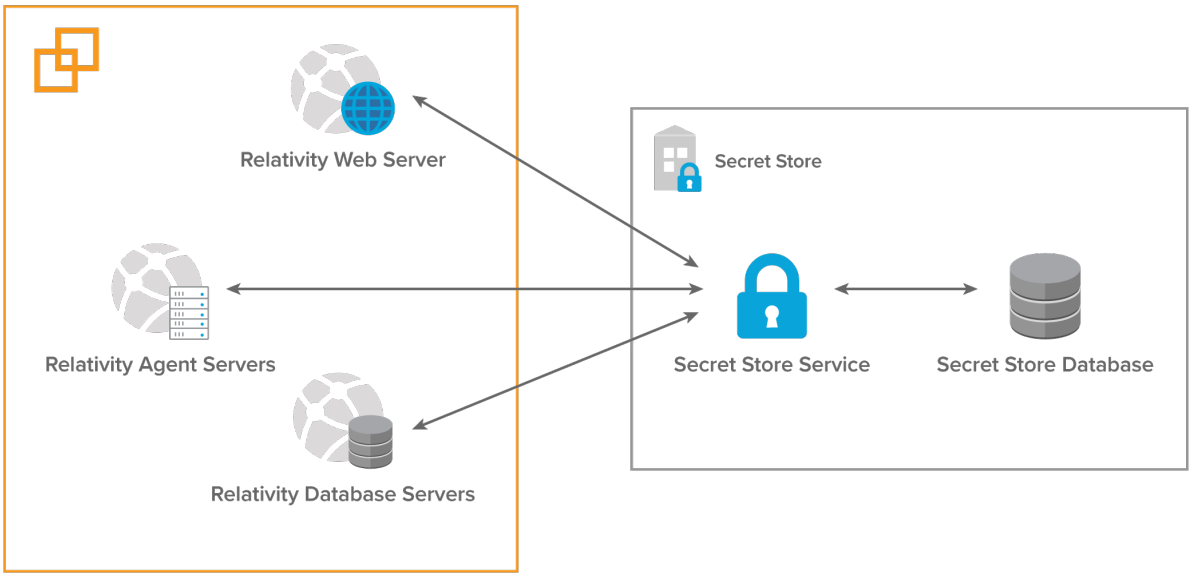
September 17, 2019 | Version 10.1.290.1

For the most recent version of this document, visit our **documentation website**.

Table of Contents

1 Relativity Secret Store

The Secret Store is a required component that provides secure, auditable storage for Relativity secrets. The secrets can be system account credentials, database connect strings, instance setting that contains confidential information (for example, your SMTP credentials), or TLS certificates. All confidential information is stored securely in the Secret Store database. The Secret Store can be accessed only by authenticated servers.



Note: In the initial Release of Relativity 9.6 (Relativity 9.6.50.31) the Secret Store is required only for installing the product. In the subsequent releases, the Secret Store is required for running Relativity and must be accessible at all times.

The Secret Store includes a SQL Server database for storing the secrets and a service for managing the secrets and client access. The Secret Store service must run at all times and be accessible to all Relativity machines (web, agent, and SQL Server). The default port is 9090. We recommend that you install the Secret Store on a dedicated server and configure it for failover or high availability.

Secret Store FAQs

I do not want to buy another server to host the Secret Store. Where else can I install it?

You can use your existing Service Bus or Agent Framework servers to host the Windows Secret Store Service, but if you start experiencing high volume requests to the Secret Store service, you may consider a dedicated server with failover or high availability. For more information, see Configuring failover and high availability on page 16.

My Secret Store server crashed and I have to replace the server. I still have my unseal key. Did I lose all my secrets?

You can still recover your secrets:

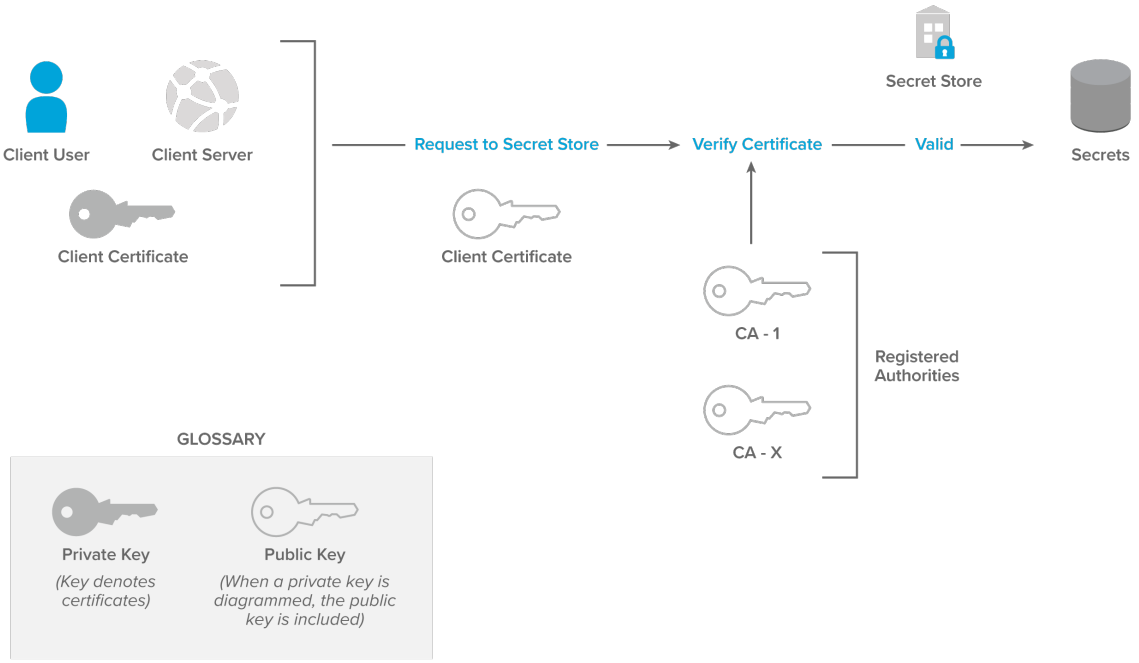
- If you had an image of your server while the Secret Store was running, restore that image and unseal the Secret Store.
- If you completely lost the Secret Store server and don't have backup:
  - Reinstall the Secret Store using the exiting database. For details, see Installing the Secret Store on page 8.
  - Use the **secretstore configure-service** command with the **repair** parameter. You will be prompted to provide your existing unseal key. For information more information, see Command reference on page 27
  - Re-register all client machines. For information on registering, see Configuring clients on page 14

I lost my unseal key. What do I do now?

- If you lose your unseal key but your Secret Store Windows service is still running in an unsealed state, reinstall Secret Store by manually repopulating the Response File.
- If you lose your unseal key and your Secret Store Windows service stopped, then all secrets are lost and you are required to reinstall the Secret Store and Relativity.

1.1 How it works

- The Relativity Secret Store generates its own Certificate Authority and Client certificates that are used to register and authenticate client servers with the Secret Store over TLS. Certificate validation verifies that both client certificate and certificate authority are well formed and not expired.



- After you initially install and configure the Secret Store, it is in a sealed state. The Secret Store API cannot read or write secrets until the Secret Store is unsealed by using the unseal key that is generated when the Secret Store service is configured.
- Each Relativity machine must be whitelisted before it can be registered.

1.2 Prerequisites

Before installing the Secret Store:

- Obtain the installer package.
- Provision the server for the Secret Store installation. The server must be accessible over the network from all machines in your Relativity environment and meet the following requirements:
  - RAM: 4 GB
  - CPU: 4 cores
  - Windows Server 2016 or Windows Server 2012 R2
  - .NET Version 4.7 or 4.6.2
- If using Windows Server 2012 R2, make sure RSA/SHA512 encryption is enabled. Fore more information, see https://support.microsoft.com/en-us/help/2973337/sha512-is-disabled-in-windows-when-you-use-tls-1-2.
- Identify the SQL Server for hosting the Secret Store database. We recommend that you use the primary Relativity SQL Server, but you can use any SQL Server. You must provide a SQL user ID and password with permissions to create a new database on your SQL Server.

- Make sure the port that will be used for accessing the Secret Store service is open for inbound traffic. The default port is 9090.
- Make sure that the ClientHostingPort used during registration is open for inbound traffic on all Secret Store client machines. The default port is 10000.

1.3 Compatibility matrix

The following is the Secret Store-Relativity compatibility matrix:

	Relativity 10.1.169.1	Relativity 10.0.318.5	Relativity 9.7.229.5	Relativity 9.6.284.6	Relativity 9.6.202.10	Relativity 9.6.134.78	Relativity 9.6.50.31
Secret Store 1.2.4.3	x	x	x	x	x	x	x
Secret Store 1.1.27.6		x	x	x	x	x	x
Secret Store 1.0.347.1			x	x	x	x	x
Secret Store 1.0.314.5					x	x	x
Secret Store 1.0.278.9						x	x
Secret Store 1.0.186.3							x

1.4 Change log

1.4.1 1.2.4.3 (June 2019)

1.4.1.1 New features

- Installing Multiple Secret Store servers using the same database will now form a Secret Store Cluster.
- The auditing level for Secret Store is now configurable through the command line utility.

1.4.1.2 Improvements

- Built-in documentation is now available for all APIs.
- Secret Store will now rate limit its clients to increase stability and recovery.
- Fixed a defect that prevented installation on servers with unusual names.
- Secret Store now logs to its install directory by default.
- Configure-Service now has a 5 minute timeout to improve reliability on slow hosts.
- Configure-Service will no longer perform factory resets on a secret store that has been unsealed at least once.
- When installing to a custom directory, a trailing backslash is now allowed but no longer required.

1.4.2 1.1.27.6 (November 2018)

1.4.2.1 New feature

- You can use a custom TLS certificate with the Secret Store Web Service.

1.4.2.2 Improvements

- When re-running **clientregistration.ps1**, the **yes** and **no** responses are not case sensitive.
- When installing Secret Store to a custom directory that does not contain the space character, Secret Store does not crash.

1.4.3 1.0.347.1 (August 2018)

1.4.3.1 Improvements

- As a performance improvement, Secret Store can now use all available CPUs.

1.4.4 1.0.314.5 (July 2018)

1.4.4.1 New features

- You can override the default installation directory with the **INSTALLDIR** parameter.

1.4.4.2 Improvements

- Running configure-service when there are no secrets in the secret store now regenerates the unseal key and **client-registration.ps1** file. Rerun this operation to get your unseal key if this operation initially times out.

1.4.5 1.0.278.9 (May 30, 2018)

1.4.5.1 New features

- Read-only support. If the database goes into read-only mode, the secrets in the Secret Store are read without writing to the audit log. All other APIs are unavailable during this time.
- New optional 'repair' flag on configure-service command that regenerates the TLS certificate used to host secret store.
- Support for IIS ARR load balancing.

1.4.5.2 Improvements

- Retries with exponential backoff for retriable SQL connection failures.

1.4.6 1.0.186.3 (February 28, 2018)

The original release. The Secret Store is required for installing Relativity.

1.5 Installing and configuring the Secret Store

Follow these steps to install and configure the Secret Store in your Relativity environment:

- Install the Secret Store.
- Configure the service and save the generated unseal key.
- Unseal the store.
- Whitelist Relativity machines.
- Register client machines.

Note: *Whitelisting* Relativity machines allows them to be registered, so that they can communicate with the Secret Store. *Registering* configures the machines for authenticating with the Secret Store by setting up the required certificates and registry values. If a machine attempts to register without being on the whitelist, registration fails. Do not whitelist and register the Secret Store server as its own client.

1.5.1 Installing the Secret Store

The Secret Store is installed by running the **install.bat** batch file provided in the installation package.

- On the Secret Store machine, launch the Windows RabbitMQ or PowerShell in Administrator mode.
- Run **install.bat** with the following parameters:

Parameter	Description
SERVICEUSERNAME	Optional. Username of the Windows account to run the Secret Store service. Defaults to LocalSystem.
SERVICEPASSWORD	Optional. Password of the Windows account to run the service. Defaults to LocalSystem.
SQLINSTANCESERVERNAME	Optional. The SQL Server instance name. Defaults to localhost.
SQLUSERNAME	Conditional. If using SQL Server authentication, the user ID with permissions to create a new database on your SQL Server. **Note:** Make sure to specify a SQL Server user account, not a Windows account.
SQLPASSWORD	Conditional. If using SQL Server authentication, the SQL Server user password.
USEWINAUTH	Conditional. Configures the installer to use Windows authentication for SQL Server access. If this parameter is set to 1, database credentials are not required.
INSTALLDIR	Optional. Used to specify a custom directory that overrides the default installation directory. The default directory is %Program Files%\Relativity Secret Store. This parameter is available in Secret Store 1.0.314.5.

Note: Depending on the installation package, the batch file name may vary. For example, it can include the Relativity version number: **Relativity 9.6.xx.xx Secret Store BAT File.bat**.

1.5.1.1 Usage examples

Install Secret Store on **SQL01** SQL Server instance using Windows authentication:

```
.\Install.bat SQLINSTANCESERVERNAME=SQL01 USEWINAUTH=1
```

Install Secret Store on **server.testing.corp\EDDSINSTANCE** with username password credentials:

```
.\Install.bat SQLINSTANCESERVERNAME=server.testing.corp\EDDSINSTANCE SQLUSERNAME=Username SQLPASSWORD-
D=Password
```

Install to a custom directory. Make sure that you include the trailing backslash.

```
.\Install.bat SQLINSTANCESERVERNAME=SQL01 USEWINAUTH=1 INSTALLDIR=E:\SecretStore\
```

Note: Windows Command interprets quoted paths that look like **F:\Dir\Secret Store**\ to end in a double quote. To avoid this error, either use Powershell to perform the installation, or double up the path separators: **F:\\Dir\\Secret Store**.

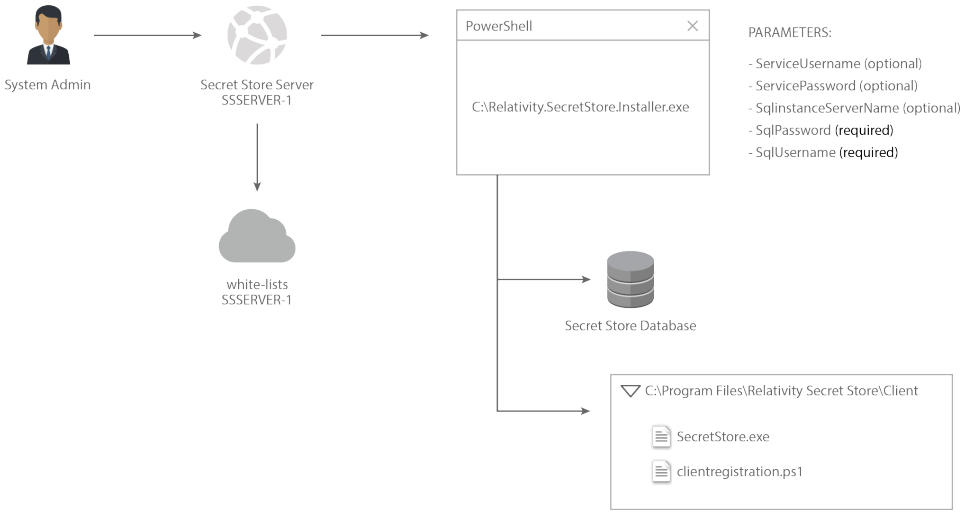
You can also use an additional command line parameter to repair an existing Secret Store installation:

```
.\Install.bat /repair SQLINSTANCESERVERNAME=server.testing.corp\EDDSINSTANCE SQLUSERNAME=Username
SQLPASSWORD=Password
```

1.5.1.2 Installation results

A successful installation creates the following artifacts:

- Folder structure on the Secret Store server:
  - **C:\Program Files\Relativity Secret Store\Client**
  - **C:\Program Files\Relativity Secret Store\Service**
- **SecretStore** database in the target SQL Server instance.
- **Relativity Secret Store** Windows Service. The service is stopped by default and requires you to run the **configure-service** CLI command.



1.5.1.3 Installer troubleshooting

If the console output displays a failure message, check the log file (InstallLog.txt) for additional information.

There are two main points of failure:

- Service failed to install
  - SQL credential validation fails - see SQL Credential exceptions for details.
  - If you run the installer again to replace the missing Windows service or corrupt files, use the repair mode.
  - Make sure you are not attempting to downgrade the application - only same and higher version installs are allowed.

- Make sure SERVICEUSERNAME and SERVICEPASSWORD are correct unless you are going to run the service as the LocalSystem account.

- Database operations failed (post-install)
  - These failures can't be rolled back.
  - Run the installer as an administrator so the installer can write and modify files on disk. Check the logs for errors related to specific script failures.

1.5.2 Configuring the service

After you install the Secret Store, you must configure the Secret Store service using the **secretstore.exe** command-line utility in **C:\Program Files\Relativity Secret Store\Client**.

Note: The CLI commands used with the Secret Store command line tool are all lowercase. Uppercase versions of these commands won't work.

For example, the following command is invalid:
.\secretstore SEAL-STATUS

This example has the correct form because it uses lowercase for the CLI command:
.\secretstore seal-status

You can specify the service port as the input parameter or use the default 9090 port. The program validates that port is available to host the service. See the following example:

```
.\secretstore configure-service 9090
```

When running the **configure-service** command, you have the option of using a custom TLS certificate.

Note: You can only use a custom TLS certificate if you are using Secret Store 1.1.27.6 and above.

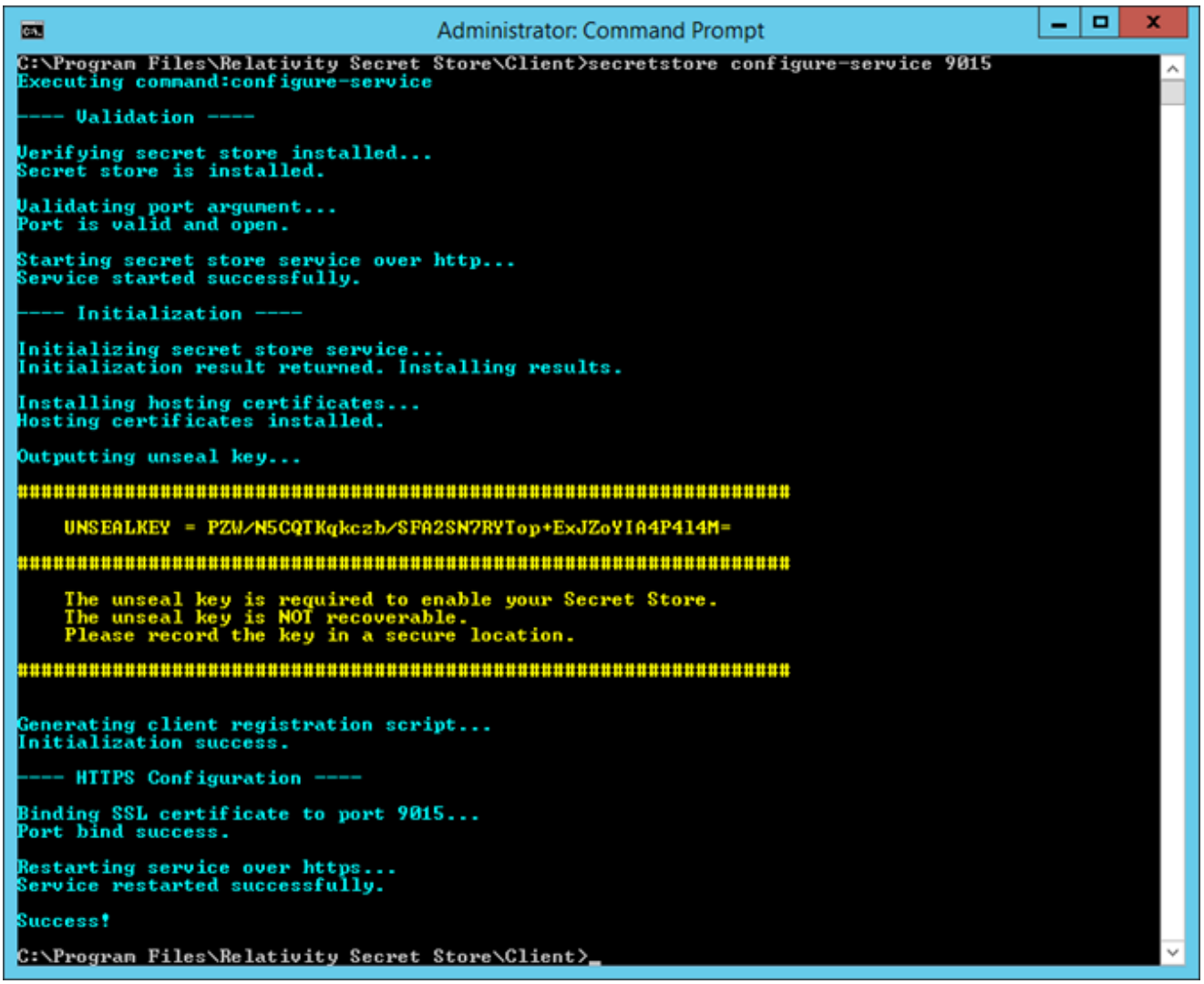
To use a custom TLS certificate:

1. Install the TLS certificate and private key in the Windows Machine Certificate store.
2. Record the thumbprint of the TLS certificate. Do not enter any spaces.
3. Add a command line option of **—serverCertificate=<TLS Certificate Thumbprint>** when running **configure-service**. For example:

   ```
   .\secretstore.exe configure-service --serverCertificate=0000000000000000000000000000000000000000
   ```

Note: If you have already installed Secret Store and would like to migrate to a custom TLS certificate, the steps are the same.

After the installation completes, it starts the Relativity Secret store service and displays the unseal key for the Secret Store in the console. An *Issued To* <Server Name> Certificate has been generated under the /Personal folder of the Windows Certificates Management application.



Copy the key and save it in a password management system. Use the key to unseal the Secret Store in the next step.

Note: If you lose the unseal key, you lose all your previously stored secrets and they will no longer be recoverable. You must re-install the Secret Store to receive a new key.

1.5.2.1 Troubleshooting service configuration

The following can cause the Secret Store Service configuration to fail:

Error	Description
This application could not be started.This application requires one of the following versions of the .NET Framework: .NETFramework,Version=v4.6.2	Make sure .NET 4.6.2 or 4.7 is installed on the server.

Error	Description
TLS certificate failed to bind to port (ServicePort). Error code: (error code).	Attempting to bind the TLS certificate to the specified port failed due to a Win32 exception. See the attached error code for more details. You can find descriptions for each error code here: https://msdn.-microsoft.com/en-us/library/cc231199.aspx
ServicePort argument is not valid. Outside valid port range.	The value of the passed in ServicePort parameter is outside the valid port range of 1-65535.
ServicePort argument is not valid. Invalid integer.	The value of the passed in ServicePort parameter is not a number.
Failed to change Secret Store start URL to start service over HTTPS.	To automatically restart the service over HTTPS, the installer changes the Windows service parameters inside the registry. If you receive this error, the installer was not able to find the key. The path to this key is setup automatically by windows when registering a new service and should be **SYSTEM\CurrentControlSet\Services\Relativity Secret Store**. Rerun the installer in repair mode to fix the registry settings.
Didn't receive timely response from Secret Store service while waiting for {status} status	While restarting the service over HTTPS, the command checks to make sure it is stopped and started correctly. If the service does not respond with a status change in less than 30 seconds, it throws this error. Check to make sure the service is set up correctly.
Access denied to Windows Cer-tificate Store. Run command with administrator privileges.	Writing the PKI certificates failed due to an access denied error. Make sure you are using an admin user. To help remedy this problem, you can also run the command prompt as an administrator.
Access denied to registry. Run command with administrator priv-ileges.	See above for access denied to Windows Certificate Store.
The service cannot start.	There may be a local computer policy that denies the user account log on as a service permissions. Make sure the service has the per-missions and then run the configure-service command again.
Timeout occurred and no unsealed key was generated.	If you run the configure-service command and a timeout occurs, you can now rerun it to get your unseal key and **client-registration.ps1** file. This functionality is available in Secret Store 1.0.314.5.

1.5.3 Unsealing the Secret Store

After you configure and start the Secret Store service, the Secret Store is still sealed and cannot be used. You can verify the Secret Store status by running this command:

```
.\secretstore seal-status
```

To unseal the Secret Store, execute this command:

```
.\secretstore unseal <unseal key>
```

Verify that the Secret Store is unsealed:

```
.\secretstore seal-status
```

Note: Every time the Windows Secret Store service is stopped or restarted, you must unseal your Secret Store to allow Relativity to read and write secrets.

1.5.4 Configuring clients

After unsealing the Secret Store, you can configure the client machines in your Relativity environment to allow them to access the Secret Store.

First, you must whitelist all machines in your Relativity environment. Then you must register them.

Note: Do not whitelist and register the Secret Store server as its own client. It is already whitelisted by default. Running registration on the server overwrites the already installed certificates, and you will have to reinstall the Secret Store.

Whitelist the machines by adding a single machine name, using wildcards, or by entering a space-delimited list of servers:

```
.\secretstore whitelist write MyMachine01.mycompany.corp
```

```
.\secretstore whitelist write *.mycompany.corp
```

```
.\secretstore whitelist write MyMachine01.mycompany.corp MyMachine02.mycompany.corp MyMachine03.-
mycompany.corp
```

To view whitelisted machines, run this command:

```
.\secretstore whitelist read
```

For more information about the available whitelist operations, see Command reference on page 27.

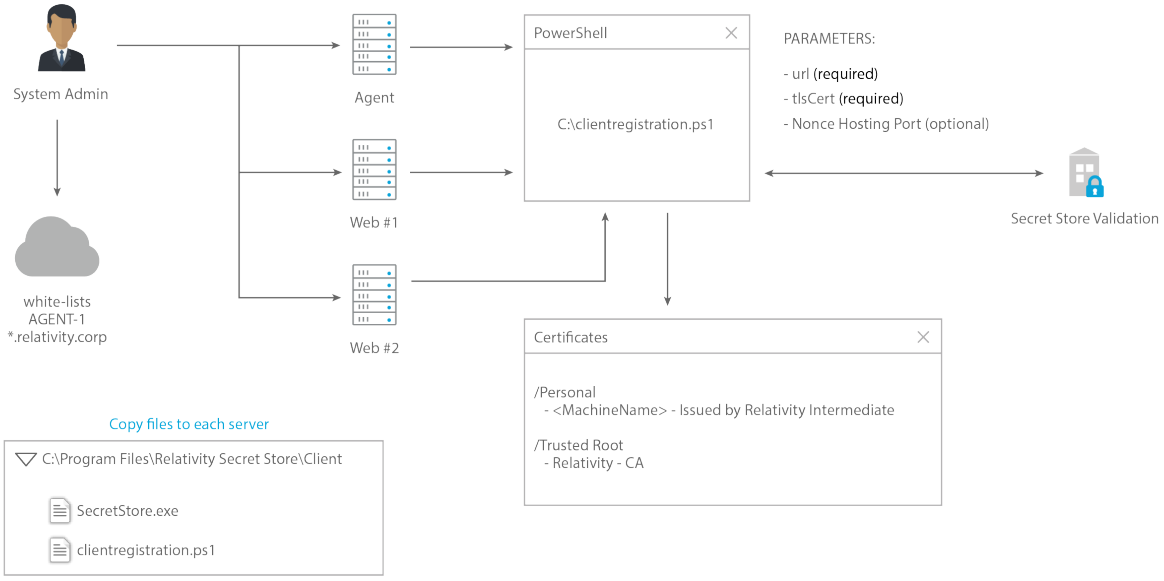
To register the client machines:

- Copy the content of **C:\Program Files\Relativity Secret Store\Client** to each of the whitelisted machines.
- Run **clientregistration.ps1** in **C:\Program Files\Relativity Secret Store\Client** from each machine.

```
.\clientregistration.ps1
```

The registration installs the following certificates:

- /Personal - Relativity Secret Store – Issued by Relativity Intermediate
- /Trusted Root – Relativity CA



1.5.5 Clean up the whitelist

After registration is complete, the whitelist is no longer necessary. We recommend leaving the whitelist empty while not actively registering clients. To clean up the whitelist, use the CLI tool.

```
.\secretstore whitelist delete MyMachine01.mycompany.corp
```

```
.\secretstore whitelist delete *.mycompany.corp
```

```
.\secretstore whitelist delete MyMachine01.mycompany.corp MyMachine02.mycompany.corp MyMachine03.-
mycompany.corp
```

Before finishing up, the whitelist should be empty. You can check using the CLI tool:

```
.\secretstore whitelist read
```

1.6 Next steps

Install Relativity. Note the difference in the response file after the installation is complete:

```
### The password for the EDDSDBO account on the SQL Primary SQL Instance you are installing to.
### The EDDSDBO login must have the same password for all Relativity Database servers (Primary or Dis-
tributed).
### If you are installing for the first time and this login doesn't exist, we will attempt to
### create it for you, so ensure this password passes any password strength restrictions that
### may be in place. Otherwise, this must match the existing account's password.
### E.g. EDDSDBOPASSWORD=MySecretPassword
### Value exported to Secret Store
### Domain (or Workgroup) and Username of the Relativity Service Account Windows login.
### This Windows login must already exist.
```

```
### E.g. SERVICEUSERNAME=domain\username
### Value exported to Secret Store
### Password for the SERVICEUSERNAME.
### E.g. SERVICEPASSWORD=MySecretPassword
### Value exported to Secret Store
```

The following secrets are migrated to the Secret Store:

- EDDSDBOPASSWORD
- SERVICEUSERNAME
- SERVICEPASSWORD
- SQLPASSWORD
- SQLUSERNAME

For detailed information about the response file, see Relativity installation documentation.

Notes:

- During a first-time installation of Relativity, make sure that your response file contains the service account user name and password. Do not copy the response file with redacted secret values from server to server.
- If you shut down the Secret Store server, it is automatically sealed. You must unseal it to resume Relativity installations.

1.7 Configuring failover and high availability

As of the May release, all Relativity secrets are being migrated to the Secret Store, and you are required to run the Secret Store to use Relativity at all times. Secret Store already supports failover and high availability, and you can set it up the your Relativity environment to ensure reliable system operation.

You can use the following configurations:

- Secondary Secret Store server (Requires manual switching in the event of primary server failure)
- High availability with a load balancer

The following sections present the general steps for setting up these configurations. The steps may vary depending on your network, load balancer, and IIS version.

1.7.1 Configuring a secondary server for failover

You can install the Secret Store on two distinct servers (a primary and a secondary), and then in the event of a failure of the primary server manually switch over Relativity servers to use the secondary server. Note that this will result in Relativity being down until the manual switch-over is performed. Also, if the secondary Secret Store server fails, there is no backup service.

1. Install the Secret Store on the primary server. For more information, see Installing the Secret Store on page 8.
2. Configure the Secret Store service on the primary server. For more information, see Configuring the service on page 11.

3. Whitelist and register the secondary server with the primary server. For more information, see Configuring clients on page 14.
4. Install the Secret Store on secondary server. You must specify the same SQL Server database instance used with the primary Secret Store service.
5. Configure the Secret Store service on the secondary server.

   Note: For Secret Store versions 1.0.314.5 and 1.0.347.1, the secondary server reinitializes as if it was a first-time installation when no secrets exist and causes the first server to break. To avoid this issue, ensure that at least one secret exists in your system before configuring a second Secret Store server. You can do this by installing Relativity, or by executing the following command through the CLI tool: **.\secretstore.exe secret write <SECRETNAME> <KEY>=<VALUE>**. For example, you would enter your secret as the following: **.\secretstore.exe secret write mysecret key=value**. This workaround is no longer necessary in Secret Store 1.2.4.3.

6. Unseal Secret Store on the secondary server.

   Notes:
   - After you install Secret Store on the secondary server and unseal it one time, a cluster is automatically formed. Cluster nodes will periodically send requests to each other, checking on seal status and availability. If a server is available but sealed, other secret store nodes will automatically unseal it.
   - The only time you will need to manually unseal a node is if there are no other cluster nodes up and unsealed. This assumes that Secret Store nodes are capable of making HTTPS requests to each other (and are not blocked by firewalls).

7. Configure the machines in your Relativity environment to use the primary server.
8. In the event of primary Secret Store server failure, whitelist and register all Relativity servers with secondary Secret Store service:
   - Copy the **clientregistration.ps1** script from the secondary server to each machine in your Relativity environment.
   - Run **clientregistration.ps1** on each machine to switch to the secondary Secret Store server.

1.7.2 Configuring high availability

You can also configure the Secret Store for high availability on a load-balanced cluster of servers using an existing IIS server in your Relativity environment with a free load balancing add-on or a third-party load balancer.

The following instructions are for an IIS-based solution:

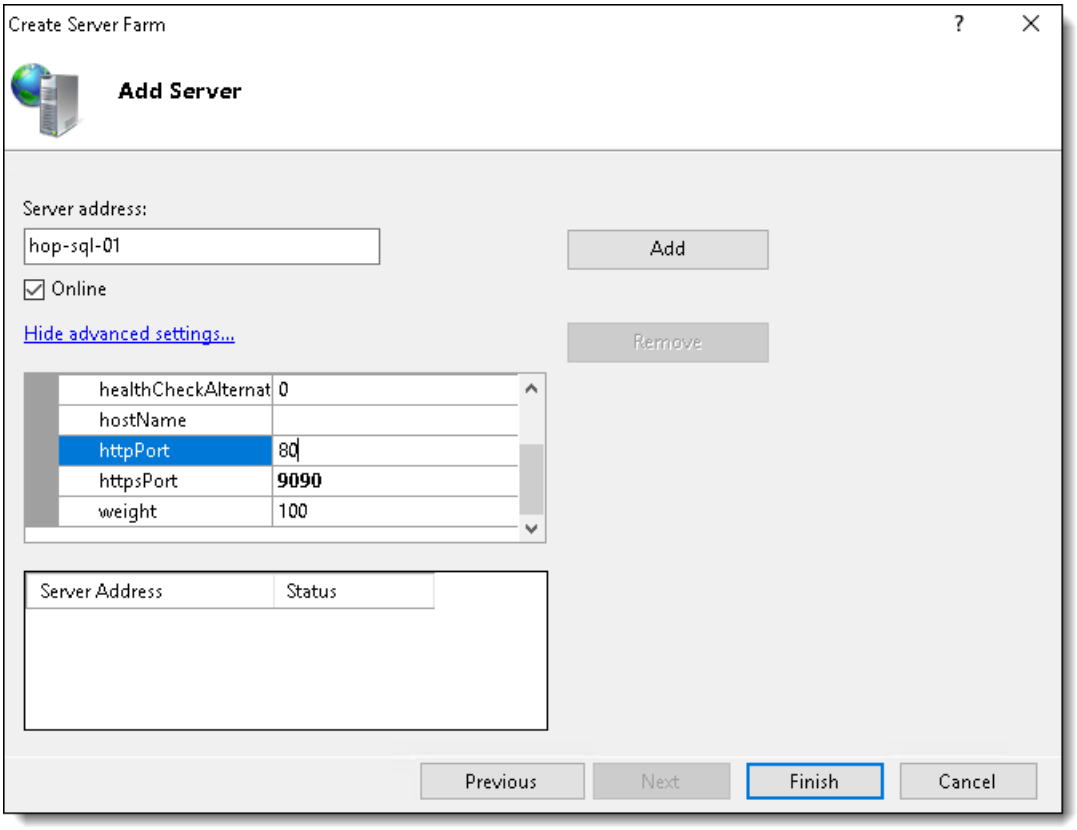
1. Identify a web server to serve as the load balancer. Download and install the IIS Application Request Routing and URL Rewrite extensions if they are not already installed.

   Note: You can't use one of the Secret Store servers as the load balancer.

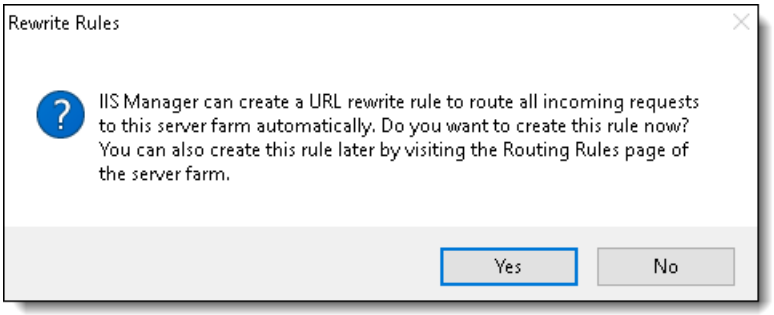
2. Install your first instance of Secret Store. If you already have it up and running, skip this step.
3. Using the CLI tool, whitelist all servers you are going to include in the Secret Store cluster.

4. Install the Secret Store on each new server:
    - Ensure that all instances of Secret Store are running on the same version.
    - Make sure the SQL Server is accessible from each machine.
    - Supply the same SQL Server instance name to each instance of Secret Store.
    - Use the same SQL credentials you did to install the first instance of Secret Store.
5. Register each server to set up necessary certificates and registry settings.
6. Run the configure-service command using the CLI tool.
    - Specify a port after the command if you want to use a port other than the default.
    - It is recommended to run all Secret Store services over the same port.

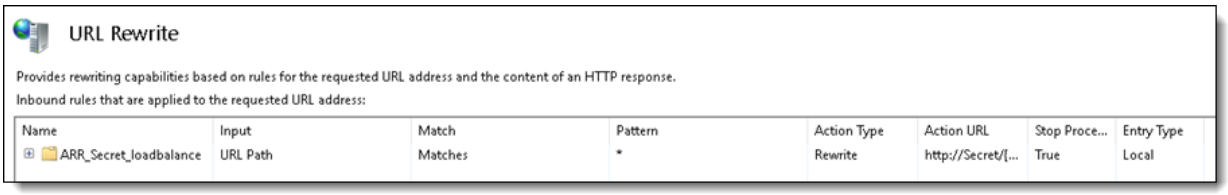
    Note: For Secret Store versions 1.0.314.5 and 1.0.347.1, the secondary server reinitializes as if it was a first-time installation when no secrets exist and causes the first server to break. To avoid this issue, ensure that at least one secret exists in your system before configuring a second Secret Store server. You can do this by installing Relativity, or by executing the following command through the CLI tool: **.\secretstore.exe secret write <SECRETNAME> <KEY>=<VALUE>**. For example, you would enter your secret as the following: **.\secretstore.exe secret write mysecret key=value**. This workaround is no longer necessary in Secret Store 1.2.4.3.
7. Unseal each instance of the Secret Store using your original unseal key.
8. Whitelist and register the selected load balancing machine with Secret Store.
9. In IIS, set up the Secret Store server farm.
10. Add all Secret Store servers to the farm.
    - Under Advanced Settings, make sure the **httpsPort** property is set to the same value (for example, 9090) for each Secret Store server.



- When you're finished, the Rewrite Rules pop-up window opens.



11. Click **Yes** on the Rewrite Rules pop-up window. This will create a default rule automatically.



If you select No, you will not be able to save the rule as the rule will not be not fully defined.

12. Modify the default rule you just created.

- Go to **Action Properties**.
- Change the scheme to **https://**.

Note: If you selected No in the previous step and no default rule was created, create the rule: **Routing Rules > Advanced Routing > URL Rewrite**.

13. Verify the HTTPS binding for the site:
    - If no binding exists for HTTPS, add the binding.
    - Select the certificate to be used for SSL.
14. Under SSL Settings, set up SSL to be required:
    - Select Require SSL.
    - Select Accept.
15. Add a health test for the servers in the farm:
    - Use this test endpoint:

      ```
      https://localhost/v1/pki/relativity/ca
      ```

    - Use the default values for the interval, timeout, and status codes.

What to do next:

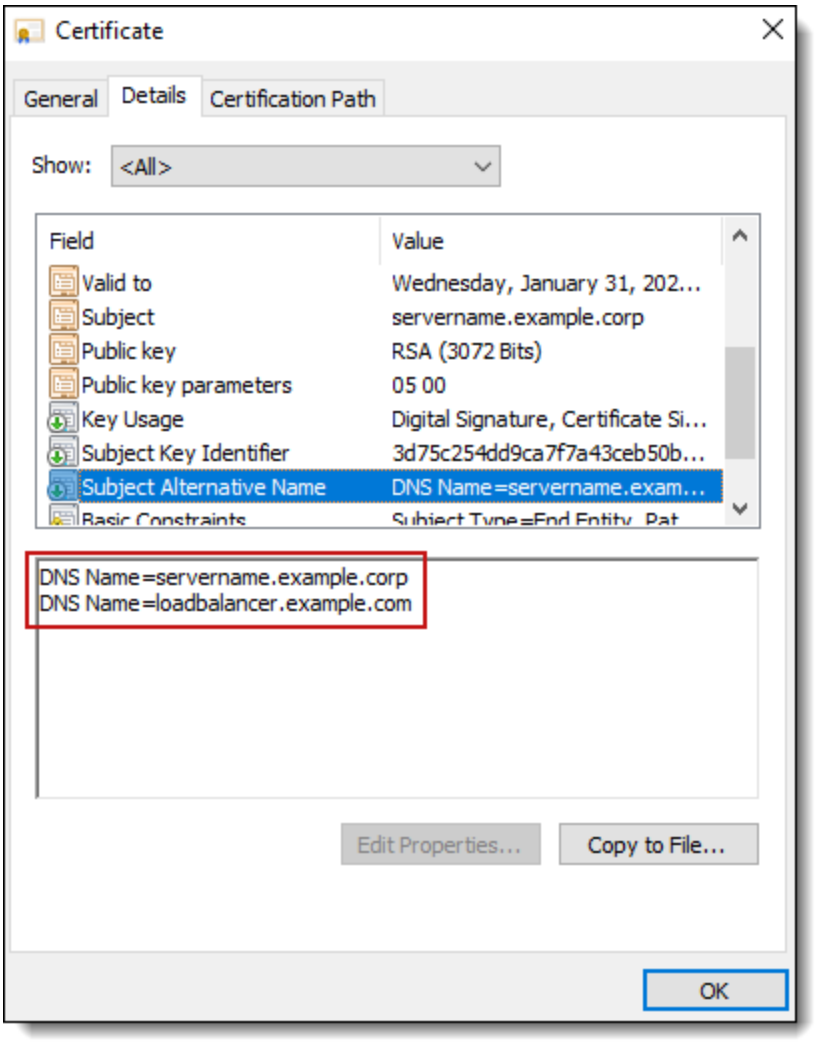
- Whitelist and register all Relativity machines using the URL of the IIS server farm.

1.7.3 Considerations for load balancing using a network device

Even though Relativity Support does not work with network devices such as F5 and NetScaler, Secret Store 1.1.27.6 and above are compatible with them.

Keep the following considerations in mind when using a network device with Secret Store:

- **Secret Store relies on valid TLS trust relationships**
  When you configure Secret Store by default, Secret Store generates a TLS certificate for itself. This certificate will be trusted for requests directed to the server's FQDN alone. As a result, the certificate is unsuitable for use with a load balancing network device.
  Generate TLS certificates for each of your secret store servers. You can buy them with a public CA or generate them with a corporate CA. Either option works as long as the generating CA is trusted. TLS certificates should have the usage flag of Server Authentication set, a subject DNS name matching the FQDN of the server, and a subject alternative name containing the DNS names of both the server and the load balancer.



Once the certificate and associated private key are installed in the windows machine certificate store, set Secret Store to use the certificate by running:

```
.\secretstore.exe configure-service --serverCertificate=0000000000000000000000000000000000000000
```

Then, unseal as normal.

- **Secret Store uses TLS for mutual authentication**
  Mutual authentication allows the Secret Store server to ensure that it only delivers secrets to servers in your Relativity environment (see registration). However, because TLS mutual authentication is baked into the TLS channel, TLS offloading and interception break authentication. Everything will appear to work at first, but none of your servers will be able to get secrets. This occurs because the Secret Store server will not be able to authenticate that the requests are coming from an authorized server. Instead, the Secret Store server sees all requests coming unauthenticated from your load balancer.
  To avoid breaking authentication, ensure that your load balancer is not intercepting, terminating, or inspecting TLS traffic to the secret store. The name of this setting varies between products, so consult your load balancer manual, or contact a support representative at the vendor. The setting will usually be called Disable TLS Offloading or Enable TLS Pass-Thru.

A layer 4 load balancer will load balance at a layer beneath TLS, and can be safely used.

- **Register clients with the load balancer URI**
  This step is the same when using IIS-ARR or a network device. When you run clientRegistration.ps1 on client servers, the script saves the URI of the Secret Store server in the machine registry.
  To update clientRegistration.ps1 to register with the load balancer:
  - Open the script in a text editor, such as Notepad.
  - Change the URI to the URI of your load balancer.

```
param(
[switch] $ForceReRegistration,
[switch] $Confirm
)

if ($ForceReRegistration) {
    ./secretstore.exe register --url=https://servername.example.corp:9090 --tlsCert=0000000000000000000000000000000000000000 --
}
else {
    ./secretstore.exe register --url=https://servername.example.corp:9090 --tlsCert=0000000000000000000000000000000000000000
}

if (!$Confirm) {
    Pause
}
```

  - Copy the script and executable to your client server and run as normal. If you are upgrading an existing install to add a load balanced network device, you will want to re-register the client servers in this way.
- **Do not break Secret Store Authentication**
  Some load balancers will allow you to configure TLS offloading by providing a client certificate to the load balancer. Avoid this type of option. The load balancer will perform authentication to the Secret Store on behalf of all requests. This effectively eliminates authentication and will expose Relativity System Secrets to anyone with network visibility to the load balancer.

1.8 Post-installation maintenance tasks

Relativity infrastructure administrator tasks for maintaining the Secret Store include:

- Maintaining the unseal key
- Changing the unseal key
- Monitoring the Secret Store status
- Updating the values of the secrets in the Secret Store
- Changing the Secret Store encryption key
- Removing unused nodes from the cluster on page 24

To perform these tasks, use the **secretstore.exe** command line tool or the Secret Store API. For more information, see Secretstore.exe CLI tool on page 26.

1.8.1 Maintaining the unseal key

The unseal key is a 256-bit key. It is the master key for Secret Store. If you lose it, the Secret Store is unrecoverable.

In most cases, we recommend that you store the unseal key separately from your Secret Store. If you store this key next to the Secret Store database, an attacker only has to steal one server to get all of your secrets.

For testing and development systems that do not contain sensitive data it is acceptable to store the unseal key on the Secret Store server.

For production systems, consider:

- A separate server that is powered off when not in use.
- A piece of paper in a safe.
- A Hardware Security Module (HSM) or equivalent.

1.8.2 Changing the unseal key

For security reasons, sometimes it may be necessary to change the unseal key. This operation decrypts and re-encrypts all encryption keys. It generates a new root key and destroys the old one. It can be very slow if there are hundreds of encryption keys and it is not transactional. We recommend that you perform it during off-peak hours.

Use the **rekey** command of the **secretstore.exe** CLI utility with the old unseal key:

```
.\secretstore rekey dGhpcyBpcyBhIHRlc3Qgc3RyaW5n=
```

When the operation completes, the console displays the new unseal key. Make sure to record the new key.

1.8.2.1 Recovering from an aborted rekey operation

If you attempt to rekey your system and something happens during the operation that causes it to halt (for example, the server crashes due to hardware failure), you won't be able to unseal the Secret Store. You can roll back a failed rekey operation.

Note: This operation (by necessity) is unauthenticated, as the system is current not unsealable.

Send a PUT request to the following endpoint using any REST client:

```
<Secret Store server URL>:<port>/v1/system/rekey/recover
```

The request body must include the following:

```
{
   "OldUnsealKey": "<the current unseal key>",
   "OtherUnsealKeys":
    [
        "<the new unseal key generated in phase 1 of rekey>",
        "<additional keys, if rekey was attempted more than once>"
    ]
}
```

When this request completes, the system is in a sealed state. Unseal the system with the original unseal key to complete the rollback. You may now attempt the rekey operation again.

Note: If you have multiple web servers, you must unseal all of them.

1.8.3 Sealed status

The Secret Store is considered sealed if it does not have access to the unseal key. Without the unseal key, the secret store is incapable of accessing any of the secrets in the database.

The Secret Store is considered unsealed if the unseal key is in memory on the Secret Store database. With the unseal key in memory, the Secret Store can decrypt the keyring (composed of one or more encryption keys), and with the decrypted keyring, it can decrypt and re-encrypt all of the secrets.

In general, you must seal the Secret Store only if you believe your network has been breached. Sealing Secret Store excises the unseal key from the server's memory and renders the Secret Store inert.

Notes:

- The seal/unseal operation fails if the key ring is corrupted (as is the case in the situation of an interrupted rekey operation).
- Each server is independently sealed or unsealed, even when used with a load balancer.

1.8.4 Updating the values of secrets in the Secret Store

To change a secret in the Secret Store (for example, Relativity database password):

1. Use the **secretstore.exe secret list** command to list all secrets:

   ```
   .\secretstore secret list /
   ```

   Note: The list operation can be very resource-intensive, and we recommend that you perform it during off-peak hours.

2. Identify the path of the secret to be updated.
3. Use **secret write** command to write the new value for the path.

   ```
   .\secretstore secret write path\to\my\secret foo=bar foo2=bar2
   ```

1.8.5 Changing the encryption key

This operation produces a new encryption key and will use it for new secret writes. It does not decrypt any data, and the old encryption keys are still in use. Because the encryption keys live only in memory unprotected, it is typically not required to rotate the encryption keys. You must do this only when the encryption key has encrypted so much data that the mathematics protecting your data starts to break down. To be conservative, do this once per terabyte of encrypted data.

Use the **secretstore.exe rotate** command:

```
.\secretstore rotate
```

1.8.6 Removing unused nodes from the cluster

When a secret store node is no longer desired (or it has crashed and is not going to be restored), you will need to remove the node from the cluster.

To do this, an API is available through REST. (NOTE: A new CLI command will be available in a future release)

Use the below Powershell snippet to remove a node from the cluster:

```
$SecretStoreRegistry = Get-Item -Path Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Relativity\SecretStore

$BaseUri = $SecretStoreRegistry.GetValue("SecretStoreUrl")
$AuthThumbprint = $SecretStoreRegistry.GetValue("ClientCertThumbprint")
$AuthCert = (Get-ChildItem -Path Cert:\\LocalMachine\My\$AuthThumbprint)[0]

$node = [Uri]::EscapeDataString("https://<server-fqdn>:<port>")

Invoke-WebRequest -Uri ($BaseUri + "/v1/cluster/node?hostname=${node}") -Certificate $AuthCert -Method
DELETE
```

Note: Capitalization matters.

To check the current state of the cluster (which includes a list of nodes), you can call the cluster health API with Powershell:

```
$SecretStoreRegistry = Get-Item -Path Registry::HKEY_LOCAL_MACHINE\SOFTWARE\Relativity\SecretStore

$BaseUri = $SecretStoreRegistry.GetValue("SecretStoreUrl")
$AuthThumbprint = $SecretStoreRegistry.GetValue("ClientCertThumbprint")
$AuthCert = (Get-ChildItem -Path Cert:\\LocalMachine\My\$AuthThumbprint)[0]

Invoke-WebRequest -Uri ($BaseUri + "/v1/cluster/health") -Certificate $AuthCert
```

1.9 Upgrading the Secret Store

To upgrade the Secret Store using a new installation package:

1. Stop the Relativity Secret Store service.
2. Run the installer. For more information, see Installing the Secret Store on page 8.
3. Run the **secretstore configure-service** command. For more information, see Configuring the service on page 11.

   Note: When you configure the Secret Store service after an upgrade, a new unseal key is not generated. You must use the original unseal key.

4. Unseal with the original key. For more information, see Unsealing the Secret Store on page 13.

   Note: You don't have to re-register client machines after a Secret Store upgrade.

1.10 Troubleshooting the Secret Store

You may experience the following problems with the Secret Store:

- **You lost the unseal key:** Backup your Secret Store database and delete it. Reinstall the Secret Store so that new database is created. Use the **secretstore secret write** CLI command to add your

response file secrets back into the Secret Store. Contact the Relativity CS department for additional assistance.

- **Secret Store service is down:** Restart the service.
- **Secret Store database down:** Restart the database.
- **Secret Store port unavailable:** Make sure no other application is using the port.
- **Relativity installation fails because a machine can't access the Secret Store:** Verify that the machine is whitelisted and registered.

1.11 Disaster recovery

Note: These instructions are for Secret Store release 1.0.278.9 and later. Secret Store is backwards-compatible, and we recommend that you always use the latest available installer.

To migrate the Secret Store to a different server and a database instance for disaster recovery:

1. Restore the Secret Store database to a new SQL Server instance.
2. Install the Secret Store on a new server and point it to the new SQL Server instance. For more information, see Installing the Secret Store on page 8.
3. Run the **configure-service --repair** command and enter the original unseal key when prompted. This regenerates the CA, intermediate, and TLS certificate. For more information, see Secret-store.exe CLI tool below.
4. Unseal the Secret Store. For more information, see Unsealing the Secret Store on page 13.
5. Whitelist and register Relativity servers with the new Secret Store instance. For more information, see Configuring clients on page 14.

1.12 Uninstalling the Secret Store

To completely uninstall the Secret Store:

- Uninstall Relativity Secret Store in **Windows > Control Panel > Programs > Programs and Features**.
- Delete the **C:\Program Files\Relativity Secret Store** folder.
- Drop the **SecretStore** database.

1.13 Secretstore.exe CLI tool

The **secretstore.exe** command line tool is used to interact with the Secret Store. It is installed alongside the Secret Store service in **C:\Program Files\Relativity Secret Store\Client**. The tool can be used to configure the Secret Store or client machines to access the Secret Store. It can also be used for most Secret Store administration and maintenance tasks.

Note: The CLI commands used with the Secret Store command line tool are all lowercase. Uppercase versions of these commands won't work.

For example, the following command is invalid:
.\secretstore SEAL-STATUS

This example has the correct form because it uses lowercase for the CLI command:
.\secretstore seal-status

1.13.1 Usage examples

Display the executable version:

```
.\secretstore --version
```

Display **secretstore.exe** help:

```
.\secretstore --help
```

Unseal the Secret Store:

```
.\secretstore unseal <unseal key>
```

Check seal status:

```
.\secretstore seal-status
```

1.13.2 Command reference

1.13.2.1 configure-service

Starts the Secret Store Windows service on the specified port. Initializes the Secret Store database if it already hasn't been set up. Sets up the required certificates and registry settings to run Secret Store and the CLI tool. Does not require the client authentication.

You can also use the repair mode to regenerate the PKI infrastructure in the database. If your secret store server crashed and you lose your client certificate, this command will set you up new root, intermediate, and client certificates. You will be prompted for the unseal key and a confirmation to continue the repair. Other than the certificate regeneration, the repair operates identically to the regular service configuration. Once the repair runs, you must re-register all client machines that connect to the Secret Store.

Parameters:

- **service port** - Required. The port for the Secret Store service.
- **repair** - Optional. If this flag is included, the service will run in repair mode. Requires the original unseal key.

Example:

```
.\secretstore configure-service 9090
```

```
.\secretstore configure-service --repair
```

1.13.2.2 register

Configures a server to access a Secret Store by installing the following:

- Certificates:
  - Client Certificate - Personal Store
  - Root Certificate - Trust Root Authorities Store
- Registry
- Secret Store URL
- Client Certificate Thumbprint
- Root Certificate Thumbprint

Does not require the client authentication. Must be run as administrator.

- **ClientCallbackPort** - Optional. Port to host the nonce value of the client. Defaults to 10000 if not specified. In the example below, the ClientCallbackPort is 5555.
- **url** - Optional. The URL of Secret Store. If not provided, the value is loaded from registry
- **tlsCert** - Optional. Certificate thumbprint of the Secret Store's HTTPS certificate. Used when the connection to the Secret Store is not yet trusted.

Example:

```
.\secretstore register 5555 --url=https://mysecretstore.testing.corp:9090 --tlsCer-
t=26f8b1a83e299874a75092ca68c4b681dc41f9f0
```

1.13.2.3 rekey

Generates a new unseal key, and rekeys the system to use it. Requires client authentication.

Parameters:

- **old unseal key** - Required. Current unseal key to be replaced.

Note: Failure to retain the new unseal key makes your system unusable. There is no way to recover without the unseal key.

Example:

```
.\secretstore rekey dGhpcyBpcyBhIHRlc3Qgc3RyaW5n=
```

1.13.2.4 rotate

Rotates the data encryption key. Requires client authentication.

Example:

```
.\secretstore rotate
```

1.13.2.5 seal

Seals the Secret Store so that secrets can't be read or written until it is unsealed. Requires client authentication.

Example:

```
.\secretstore seal
```

1.13.2.6 seal-status

Returns the current seal status of the Secret Store, sealed or unsealed. Does not require client authentication.

Example:

```
.\secretstore seal-status
```

1.13.2.7 secret

CRUD and list operations for the secrets in the Secret Store.

Parameters:

- **Action** - Required. Action to take on the secret. Possible actions are **read**, **write**, **delete**, and **list**.
- **Path** - Required. Path to the secret the action will execute against.
- **KeyValuePair** - Required for write. Optional for other actions.
- **url** - Optional.
- **clientCert** - Optional. Client certificate thumbprint to use for authentication with secret store. Read from the local machine store.

Examples:

```
.\secretstore secret read path\to\my\secret
```

```
.\secretstore secret write path\to\my\secret foo=bar foo2=bar2
```

```
.\secretstore secret list path\to\my\secret
```

```
.\secretstore secret delete path\to\my\secret
```

1.13.2.8 unregister

Removes the certificates and registry values set during registration. Does not require client authentication. Must be run as administrator. Unregistering on the Secret Store service corrupts the trust chain for HTTPS hosting.

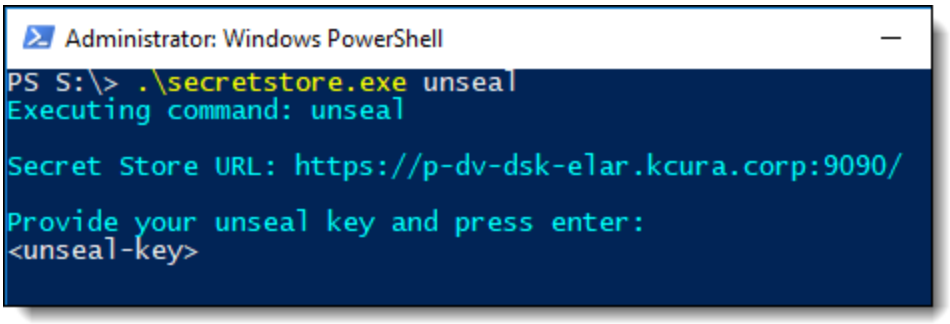
Example:

```
.\secretstore unregister
```

1.13.2.9 unseal

Unseals the Secret Store, allowing secrets to be read and written. Does not require client authentication.

Note: Starting in 1.2.12.5, the unseal key may be omitted from the command line. If omitted, you will be prompted for the unseal key. This behavior helps keep the unseal key out of logs. This change is isolated to the command line tool. If you are using an older version of Secret Store and want this enhancement, contact Relativity Support at support@relativity.com.



Parameters:

- **unseal key** - Required. Key(s) that unseal the Secret Store.

Example:

```
.\secretstore unseal dGhpcyBpcyBhIHRlc3Qgc3RyaW5n=
```

1.13.2.10 whitelist

Configures the registration white list for clients' self-registration with the specified authority. Currently registers only at with the Relativity-Intermediate authority. Requires client authentication.

Parameters:

- **action** - Required. Action to take on the whitelist. Possible actions are **read**, **write**, and **delete**.
- **domain** - Optional. The machine name or domain to be added or removed for write and delete actions.
- **url** - Optional. The URL of the Secret Store. If not specified, the value is loaded from registry.
- **clientCert** - Optional. Client certificate thumbprint to use for authentication with Secret Store. Read from the local machine store.

Examples:

```
.\secretstore whitelist write *.mycompany.com
```

```
.\secretstore whitelist delete MyMachine1.mycompany.com
```

1.13.2.11 auditing-level

Configures the auditing level of the secret store service. Defaults to BestEffort. Requires client authentication.

Parameters:

- **action** - Required. Action to take on the auditing-level. Possible actions are read, and write.
- **auditing-level** - Not required for read actions. Required for write actions. When writing, this will be the new auditing-level. Possible values include:
  - **None** - No audits are recorded.
  - **StateChange** - Only actions that change state (like Secret write or Whitelist delete) are audited.
  - **BestEffort** - All actions are audited as normal. If the auditing system becomes unavailable, secret read audits will not be captured. This is the default auditing level.
  - **Strict** - All actions are audited. If the auditing system becomes unavailable then no actions can be performed.

Some actions are never audited regardless of the above setting, like the ping action that Relativity uses to check connectivity.

Examples:

```
.\secretstore auditing-level read
```

```
.\secretstore auditing-level write StateChange
```

Proprietary Rights

This documentation ("**Documentation**") and the software to which it relates ("**Software**") belongs to Relativity ODA LLC and/or Relativity's third party software vendors. Relativity grants written license agreements which contain restrictions. All parties accessing the Documentation or Software must: respect proprietary rights of Relativity and third parties; comply with your organization's license agreement, including but not limited to license restrictions on use, copying, modifications, reverse engineering, and derivative products; and refrain from any misuse or misappropriation of this Documentation or Software in whole or in part. The Software and Documentation is protected by the **Copyright Act of 1976**, as amended, and the Software code is protected by the **Illinois Trade Secrets Act**. Violations can involve substantial civil liabilities, exemplary damages, and criminal penalties, including fines and possible imprisonment.