SigningHub Enterprise

The Essential Guide to System Recovery

ASCERTIA LTD

FEBRUARY 2021

DOCUMENT VERSION- 1.0.0.6

Contents

1 Introduction

This is an important guide for anyone managing a SigningHub Enterprise production system. The document describes how a failed system can be systematically analysed and recovered in the shortest time possible. Various potential failure scenarios are discussed and a logical sequence of checks presented. The aim of the document is to remove the time lag in gathering information to log a support call, waiting to receive the call back and then discussing what happened and walking through exactly these steps. Ascertia support staff also rely on this document!

The document assumes that a production environment with one or more SigningHub instances was operating perfectly until a system failure occurred. A comprehensive set of checks are presented that aim to identify the failure issue(s) and get SigningHub up and running. The contents are not intended to be used as general configuration guide or diagnostic guide or for checking test/development servers where configuration changes may have caused the issue, although it may be helpful in this regard.

Resolving production server failures is often difficult with feature-rich applications such as SigningHub, especially when multiple third-party components are involved.

From experience, most 'sudden' issues with a stable production system are caused by problems within the external systems, applications, or services that SigningHub relies upon. These tend to manifest themselves as a problem within SigningHub, but a quick examination of the relevant application logs will reveal whether an external issue is affecting the ability of SigningHub to give an accurate and reliable answer. For example, firewall changes can easily affect the ability to access external SMTP, ADSS Server or database services. Access to the Database, ADSS Server, optional HSM issues and important but expired certificates will immediately prevent SigningHub from working normally.

1.1 Technical Support

If technical support is required, Ascertia has a dedicated support team that provides debug, integration assistance and general customer support. Ascertia Support can be accessed in the following ways:

Support Website (for Case logging and Live Chat)	https://account.ascertia.com
Support Email	support@ascertia.com
Knowledge Base	https://www.ascertia.com/products/knowledge-base/signinghub/

In addition to the support service described above, Ascertia provides formal support agreements with all product sales. Please contact sales@ascertia.com for more details.

When sending support queries to Ascertia Support team, include all relevant SigningHub logs from the following directories:

For SigningHub Desktop Web Logs	[SigningHub Installation Directory]/web/logs
For SigningHub Admin Logs	[SigningHub Installation Directory]/admin/logs
For SigningHub API Logs	[SigningHub Installation Directory]/api/logs
For SigningHub Core Logs	[SigningHub Installation Directory]/core/logs
For SigningHub Mobile Web Logs	[SigningHub Installation Directory]/mobile/logs

If time permits, configure the SigningHub Server Logs to be created with a 'Debug' log setting. This will provide a higher level of detail and help the Ascertia Support team to quickly investigate the reported issue. To enable this setting, please follow the instructions available at the following Knowledge Article link:

https://faqs.ascertia.com/display/SK/On-Premise+Installation#On-PremiseInstallation-ChangingloggingleveIforSigningHub

1.2 Conventions

The following typographical conventions are used in this guide to help locate and identify information:

- **Bold** text identifies menu names, menu options, items you can click on the screen, file names, folder names, and keyboard keys.
- `Courier New` font identifies code and text that appears on the command line.
- **`Bold Courier New`** identifies commands that you are required to type in.

2 SigningHub – Failure Scenarios

2.1 Common Problems

SigningHub can fail to work as expected because of various issues. These can be categorised to aid problem resolution, as follows:

- Issues with the underlying operating system, host and network, including:
  - Network access to internal services such as SANs, replicated data
  - System CPU and memory resource availability to handle sudden load increases
  - Disk space availability
- Issues within third party products, including:
  - Microsoft IIS Server – the web server on which the SigningHub is deployed
  - Microsoft SQL Server, Azure SQL or Oracle database – a healthy database is essential for SigningHub
- Issues with Ascertia ADSS Server – SigningHub cryptographic services rely on ADSS Server
  - Network access to external services such as Azure Key Vault, external CAs, CRLs, OCSP and TSA Services
  - Network access to internal services such as HSM
  - Microsoft SQL Server, Oracle, MySQL, PostgreSQL database – a healthy database is essential for ADSS Server
- Other possible issues:
  - SigningHub license limits exceeded
  - Email delivery issues
  - Azure data storage issues

2.2 Problem Resolution and Document Structure

Section 3 discusses how to check the host system.

Section 4 discusses SigningHub – Server Issues.

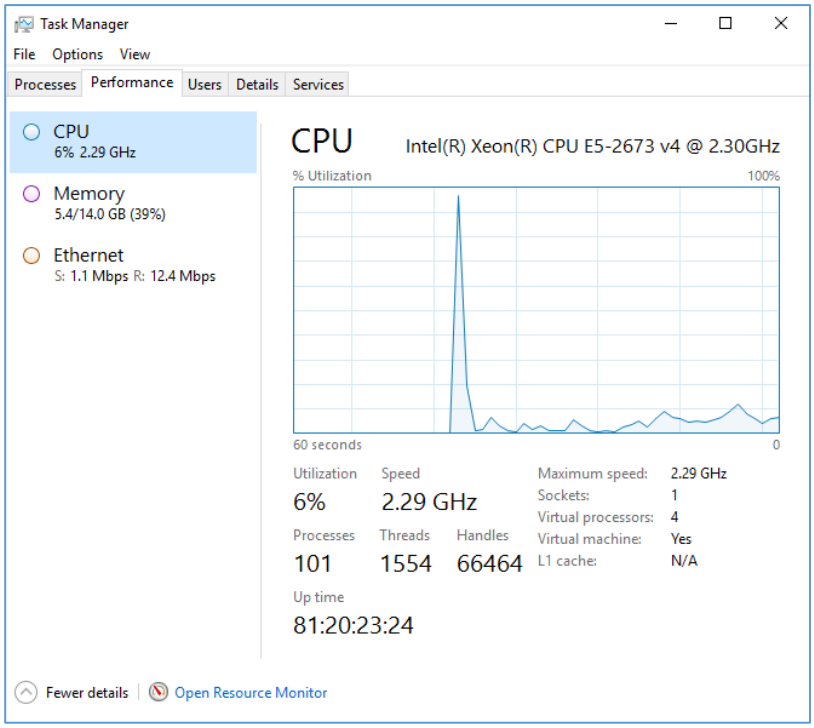
Section 5 discusses SigningHub – Document Signing Issues.

Section 6 discusses SigningHub – Miscellaneous Issues.

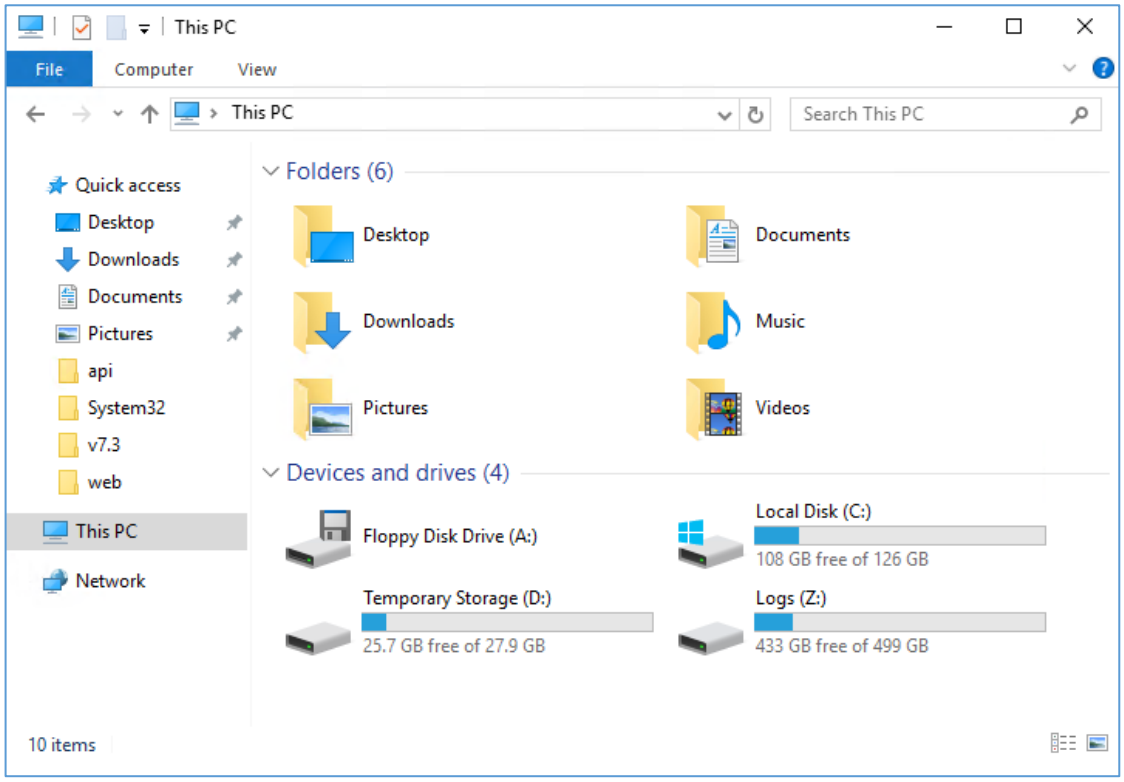
3 Checking the Host System

As with any software application SigningHub relies on an operating system and a physical or virtual platform. Hardware errors such as physical component failure are obvious. However, network timeouts or similar service connectivity failures can occur with no obvious reason as to the cause. Failures caused by hardware components or the Windows operating system are outside the scope of this document. However, summary information is presented so that the system can be checked to see it is functioning as expected.

- Check the system memory, network and CPU utilization using standard operating system tools. Launch the Windows Task Manager to quickly review the operating system resource utilization:



- Check that the host server has enough CPU and memory resources to function properly. If CPU usage, available memory or network usage is nearing the maximum limits then it could be a possible cause of the issue(s).
- Check the storage systems, physical disks or SANs are all available. For example, below screenshot shows the local disk space utilization on a SigningHub host machine:



- If configured in a High Availability (HA) environment, check that replicated data is being copied and is available.
- If the SQL Server database is installed on the same machine, then ensure that there are adequate system and storage resources available (see SigningHub Installation Guide > System Requirements). If the allocated database storage is fully used or the location for encrypted document storage is fully utilized, then this will prevent SigningHub from functioning.
- If the ADSS Server and its separate database are installed on the same physical or virtual system as SigningHub, ensure that there are adequate system and storage resources available (see SigningHub Installation Guide > System Requirements). As system usage grows it is recommended to consider separating these systems to ensure that resources can be allocated as required and that high availability configurations can be used.
- Check that the disk space is not being over consumed by having SigningHub or ADSS Server trace log levels set to DEBUG. Debug level should never be used on production systems – this should be reserved for use on pre-production test systems. In case there is no pre-production/staging environment then DEBUG level can be enabled for the short period of time to capture an event in detail, as requested by Ascertia Support.
- Check if SigningHub is continuously consuming high levels of CPU. SigningHub can consume a lot of CPU and memory when concurrent user counts are high and/or when users are uploading and working on large documents (e.g. over 1MB, certainly over 2MB and especially when documents are over 10MB).
- If SigningHub has applications connecting via its RESTful API then if these applications are creating a lot of user accounts or sending a lot of data, then this presents a very high load and can consume high levels of CPU and memory.
- It is strongly recommended to configure monitoring and alerts for SigningHub host machines to notify the system administrators about any situations involving abnormally high CPU usage, memory usage, disk usage and network usage. Microsoft Azure and other cloud providers offer such alerting services.

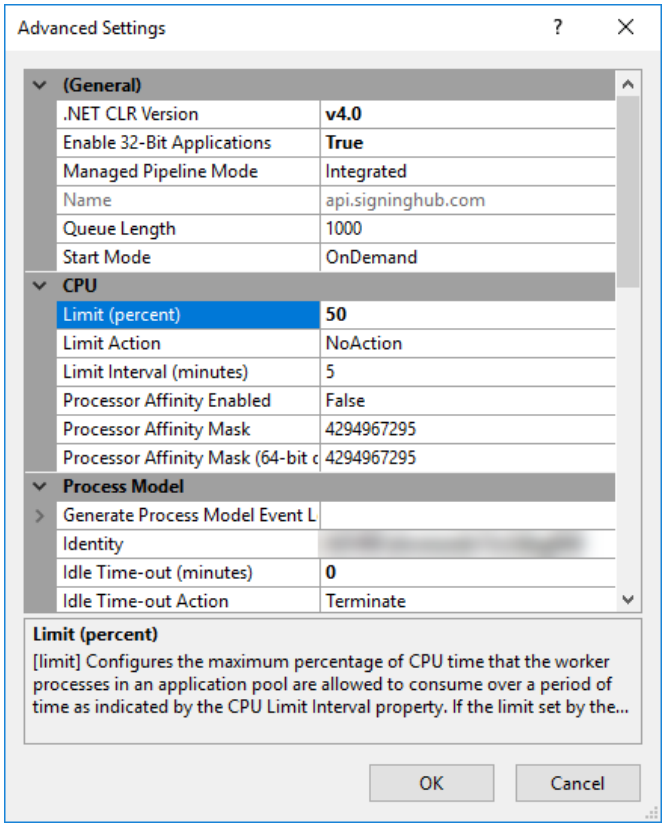
- In the situation that the SigningHub host machine is inaccessible, especially when the CPU, disk, memory uage or network usage is very high. In such a situation, restarting the host Windows machine can be an option to recover from the failed state. If the SigningHub services are partially unavailable (e.g. SigningHub Desktop Web is accessible but SigningHub API not accessible) or when some other services are also hosted on the same machine and restarting the machine is not acceptable, remotely restarting the IIS Server can also recover SigningHub. You can use PowerShell commands to restart the IIS Server remotely. See below link for more details:

  https://gallery.technet.microsoft.com/scriptcenter/Powershell-script-to-363dd543

- To make sure that the SigningHub services do not consume 100% of the system CPU, a usage limit can be applied to the SigningHub API, Web, Core, Mobile and Admin sites.

  To limit the CPU usage by a SigningHub worker process (e.g. SigningHub API), you should follow the below instructions:

  - Launch **IIS Manager**
  - Select **Application Pools** from the left tree menu
  - From the list of application pools, select application pool for the SigningHub API. Normally application pool has the same name as the website e.g. api.signinghub.com
  - Right click on the application pool and select **Advanced Settings**
  - In the **Advanced Settings** dialog, the **CPU** section allows to apply a limit on the CPU usage by the worker process. See below screenshot for details:



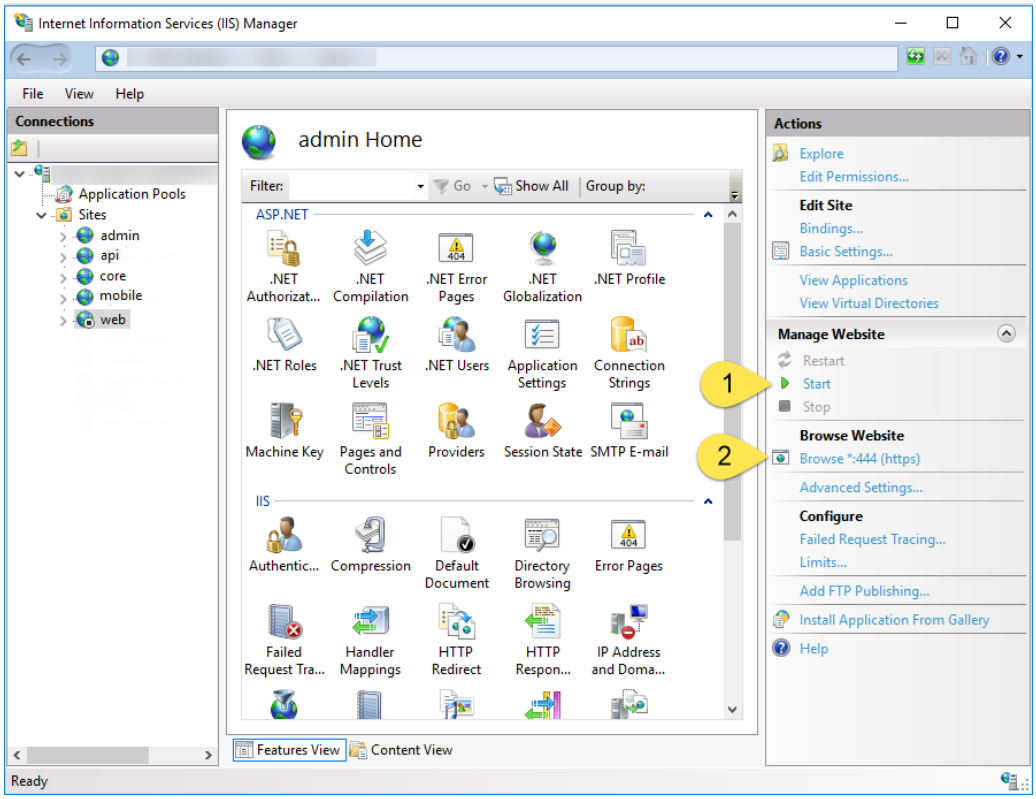
4 SigningHub – Server Configuration Issues

This section looks at IIS and database configuration issues. Both are critical third-party components for SigningHub.

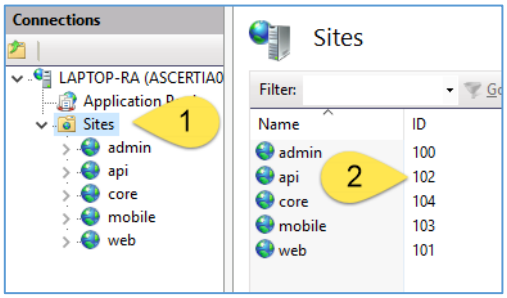
4.1 Web Application Issues

When installed SigningHub creates five separate websites. Check these are all running, their default names are: **admin**, **api**, **core**, **mobile** and **web.**

a) The website names could be different if different names were chosen at the time of installation. If all or any of these websites are not running, then start these websites and their related application pools and check if you can access the appropriate website from a browser by clicking the Browse option in IIS.



b) Assuming all of the SigningHub websites are running, but one or more are not accessible through a web browser, get the Site **ID** from **IIS > Sites** like this:



Now check the website IIS log for issues. To do this go to **C:\inetpub\logs\LogFiles** and open the log directory with the appropriate Site **ID. O**pen the **.log** file, scroll to the end and see what HTTP error code is shown there. Use a web-browser to search what that HTTP code means and the possible solutions. If this information is not understandable and/or there is no conclusive solution, then send the log file, error message or web browser screenshot to support@ascertia.com.

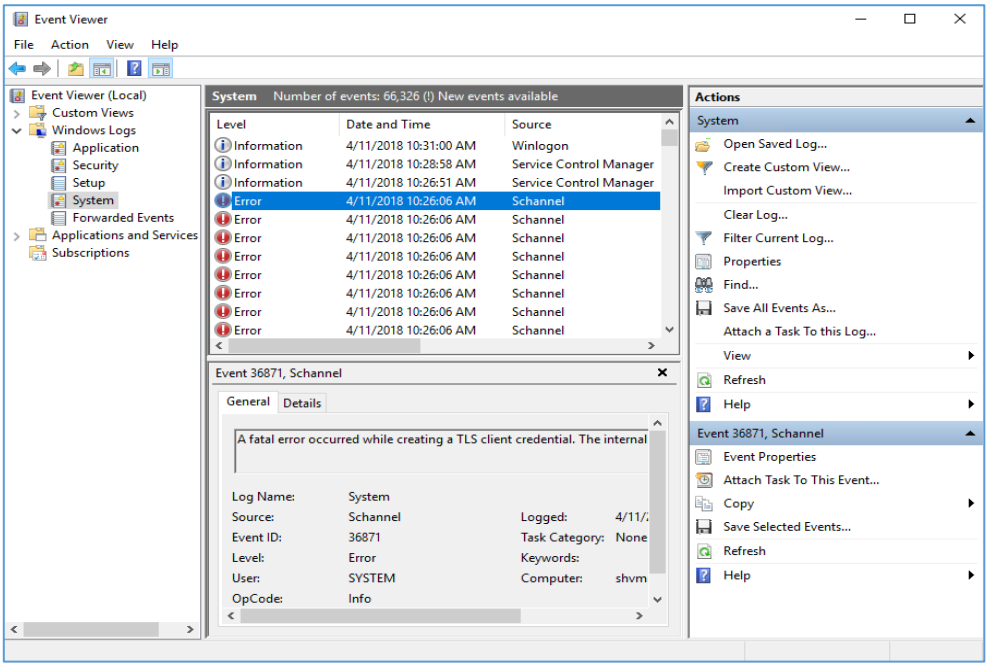
An example log snippet showing an internal server error (i.e. HTTP error code 500) while accessing a SigningHub Desktop web page can be found as:

```
2018-04-11 04:45:22 10.2.0.4 POST /Login/PreLoginCheck - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 109
2018-04-11 04:45:25 10.2.0.4 POST /Login/Login - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 323
2018-04-11 04:45:25 10.2.0.4 GET /Web - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 82
2018-04-11 04:45:25 10.2.0.4 GET /Content/themes/adocs/css - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 500 19 64 5
2018-04-11 04:45:25 10.2.0.4 GET /adss/gosign/applet/lib/adss_gosign.js - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 83
2018-04-11 04:45:25 10.2.0.4 GET /Content/themes/adocs/images/favicon/14774SH-favicon.ico - 443 - 21.14.19.18 Mozilla/5.0 - 200 0 0 118
2018-04-11 04:45:25 10.2.0.4 POST /Main/GetLoggedInUser - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 152
2018-04-11 04:45:25 10.2.0.4 POST /Utility/GetSystemSettings - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 88
2018-04-11 04:45:25 10.2.0.4 POST /Main/GetEnterpriseSettings - 443 - 21.14.19.18 Mozilla/5.0 https://web.signinghub.com/ 200 0 0 114
```

Check the date/time in the IIS log file when the error code was logged and then look at log entries in the SigningHub log files at the same time to see if the error was caused by a failure in SigningHub.

c) If the website is still not accessible or fails to start, check the Windows Event Viewer logs. If there is any issue reported for the same time period, send full information to support@ascertia.com.

An example event log entry is shown below which points to an SSL issue when setting up secure connection while accessing SigningHub Admin:



d) Check the SigningHub API service is up and running - open the web browser and type the following URL:

https://<Signinghub API URL as per your local deployment>

i.e. https://api.signinghub.com/

If the API service functioning properly then you will get the following response:

<string>Success</string>

Or

<string xmlns="http://schemas.microsoft.com/2003/10/Serialization/">Success</string>

4.2 Web Application Monitoring

There could be situations when one or more SigningHub websites are failing or not responding quickly. To detect these situations, it is strongly recommended to use an independent website monitoring and alerting service like https://www.site24x7.com/ or https://uptimerobot.com/.

Configure the public URLs of all the SigningHub websites you want to monitor e.g. **admin**, **api**, **core**, **mobile** and **web.**

4.3 Database Connectivity Issues

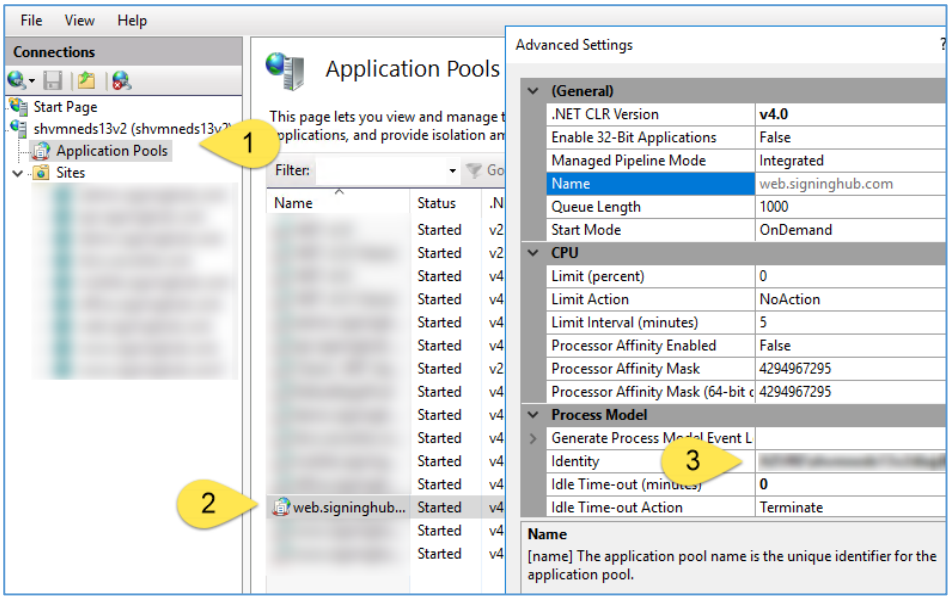
A correctly functioning database is essential for a production SigningHub system. SigningHub database issues can be investigated by:

a) Verifying that the database server is accessible from the SigningHub system. Use of standard ping and trace route utilities can achieve this. If the database host is not reachable then there is either an infrastructure problem, e.g. firewall, or internal database issue. Resolution of these is outside the scope of this document.

b) If SigningHub was installed with SQL Server authentication and you suspect that the database credentials may have been changed then follow instructions in section **5.4 - Changing Database Credentials for an Existing Installation** of SigningHub Installation Guide to update the credentials in SigningHub installation.

Similarly, if the database is configured to enforce password changes after a defined time-period then this can result in SigningHub database access being denied. In this case new database credentials would need to be configured for SigningHub using the above instructions. Frequent expiry of database credentials is not a good option for SigningHub running in production.

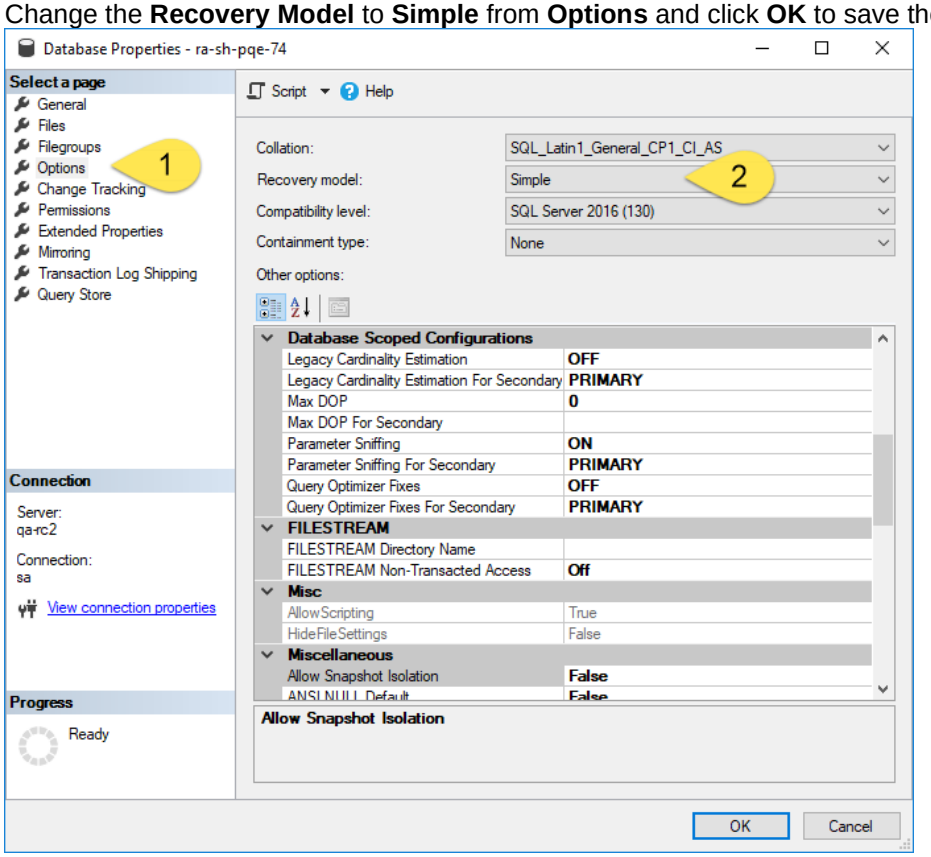
c) If SigningHub was installed with Windows authentication and you suspect that the Windows user password has changed then enter the new password for each website as shown:



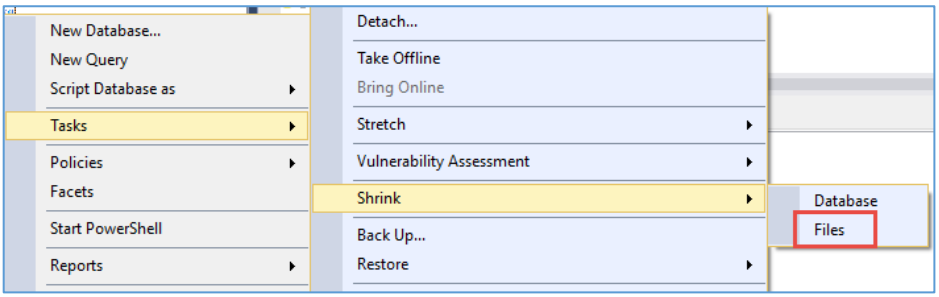
d) Check that the disk space is not fully consumed on the database server by the SigningHub database and/or its log file. If the database file or log file size is huge, you can shrink it to free up the space as following:

1) Stop the IIS Server.

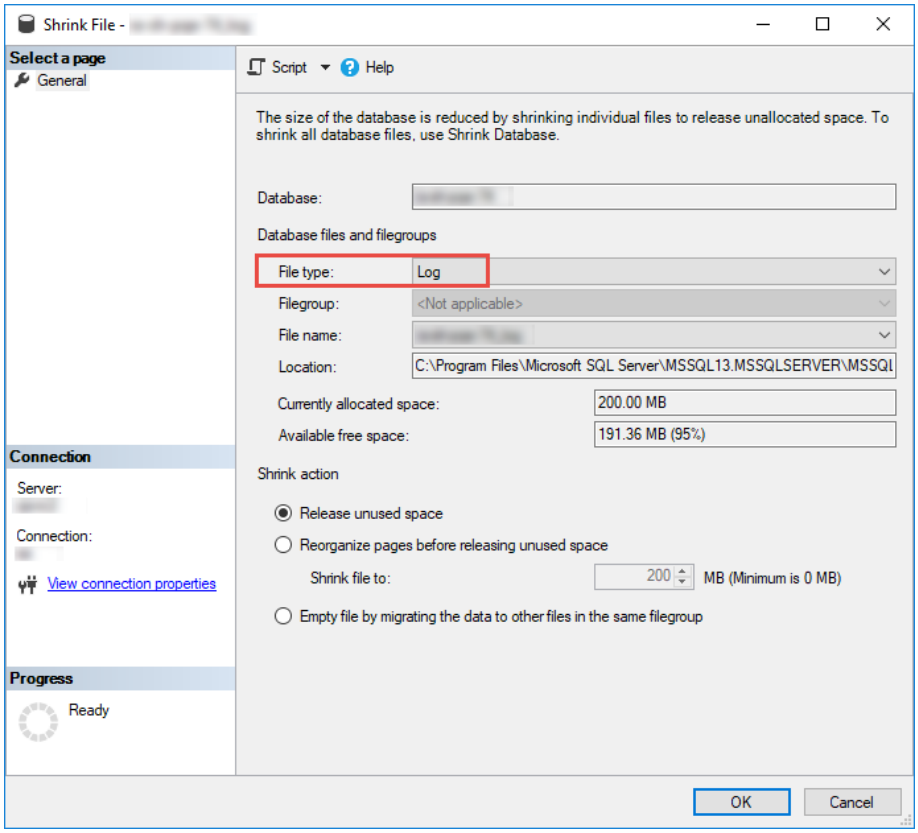
2) Login to the database server using SSMS, select the database and go to its **Properties**. Change the **Recovery Model** to **Simple** from **Options** and click **OK** to save the settings:



3) Now shrink the **Files** by right clicking the database, **Task > Shrink > Files**:



4) Set **File type** to Log in the next dialog and click **OK** to shrink the Log file.



e) Repeat the same steps to shrink the **Database**.

f) Restart IIS to make SigningHub services available.

g) If SigningHub is installed with Azure SQL and frequent SQL query timeouts are reported in the log files when communicating with Azure SQL then make sure that you are using an appropriate service tier that can handle the SigningHub load that is being applied by the concurrent users and API calls.

There is no specific recommendation for Azure SQL service tier - it should be chosen to suit the usage demands seen during peak hours. The following is a snippet of a timeout exception message:

Execution Timeout Expired. The timeout period elapsed prior to completion of the operation or the server is not responding.

This error message can be found in one of the following log files:

[SigningHub Installation Directory]/web/logs/webLog.txt

[SigningHub Installation Directory]/admin/logs/adminLog.txt

[SigningHub Installation Directory]/api/logs/apiLog.txt

[SigningHub Installation Directory]/core/logs/coreLog.txt

h) If SigningHub is installed with Azure SQL and the performance is poor, the use of Auto Tuning on the Azure portal is recommended so that necessary Indexes are created automatically. Although SigningHub creates required table indexes at installation, usage based Auto Tuning by Azure can suggest additional indexes to optimise SQL queries.

5 SigningHub – Document Signing Issues

If there are no obvious problems with the SigningHub system in the previous section and issues are only seen when signing a document, then it is likely to be an ADSS Server issue (or an external system that is affecting ADSS Server).

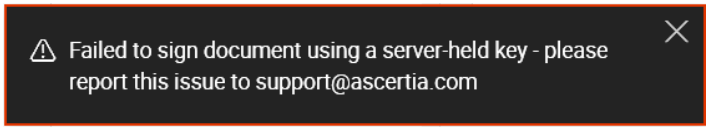
5.1 SigningHub and ADSS Server Relationship

ADSS Server is the crypto-services engine that is shipped with SigningHub and can be deployed on the same or separate system or separate site to SigningHub. It performs many of the cryptographic operations related to document signing e.g. key generation, document signing, timestamping and verification etc. However, SigningHub does carry out some cryptographic functions itself, namely:

- It hashes each document and then relies on ADSS Server Signing Service, or the optional Go>Sign and RAS/SAM Services, to sign it using the identified certificate;
- It relies on ADSS Server Verification Service to verify signed hashes, before comparing it internally with the original;
- It encrypts all documents, and in an Azure environment relies on communication with Azure BLOB storage for storing documents and its revisions. For other environments the file storage system may be used, or a separate supported database can be configured;
- If configured to do so, it securely communicates with the external authentication services e.g. Google, Microsoft, SAML v2 providers etc.

5.2 ADSS Server Issues

There can be multiple issues with ADSS Server and the external services it uses (described below). If ADSS Server encounters any of these issues, this error message (or similar) is displayed to end users at the time of document signing (the email address can be re-configured):



5.2.1 ADSS Server is unresponsive

If ADSS Server cannot be accessed, SigningHub will be unable to process user registration, signature creation and verification requests.

Check that ADSS Server is functioning using ADSS Server - The Essential Guide to System Recovery document and refer to section **3.2 ADSS Server Tomcat Instances**.

5.2.2 ADSS Server license has expired or is beyond limits

If the ADSS Server license has expired or has exceed licensed limits, the SigningHub services will stop. To check the license of ADSS Server, refer to section **4.1 ADSS Server License Management** of the ADSS Server - The Essential Guide to System Recovery document.

5.2.3 ADSS Server HSM is disconnected

Hardware Security Modules (HSM) are optional and can be configured within ADSS Server. If they are configured and the connection to the HSM fails, then SigningHub will stop signing. To check

HSM connectivity with ADSS Server, refer to the section **3.5 External HSM** of the ADSS Server - The Essential Guide to System Recovery document.

5.2.4 Certification Service Issue

ADSS Server's Certification Service is responsible for generating and managing individual user signing keys and certificates held centrally. If new users can't be registered via API or are not able to sign documents, then it could be an issue within the Certification Service or its external connections to HSMs or CAs. To check the Certification Service of ADSS Server, refer to the sections 3 and 10 of the ADSS Server - The Essential Guide to System Recovery document.

5.2.5 Signing Service Issue

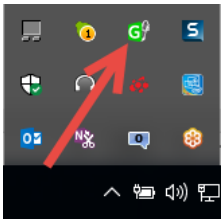
The ADSS Signing Service may have an issue internally or externally with HSMs or external OCSP or TSA services. To check the Signing Service, refer to sections 3 and 7 of the ADSS Server - The Essential Guide to System Recovery document.

If SigningHub is configured to sign using local smartcards or USB tokens, then also check the ADSS Go>Sign Service. If using Authorised Remote Signatures, then check the ADSS Signing Service and RAS/SAM Services.

5.2.6 Local Signing Issue

If users are trying to sign using smartcards or USB tokens or soft keys held on their local systems and they are failing to sign documents, there could be number of possibilities that are causing this failure as explained below:

a) Go>Sign Desktop is not installed or running. If Go>Sign Desktop is running, then an icon will be shown in the Windows task pane:



If this icon is not shown, then check if the Go>Sign Desktop is installed. You can get the latest version of Go>Sign Desktop from https://www.ascertia.com/downloads/releases/go-sign-desktop/Ascertia-Go-Sign-Desktop-Win64.msi. As soon as you launch this application from Windows program menu then this icon will be shown in the task pane.

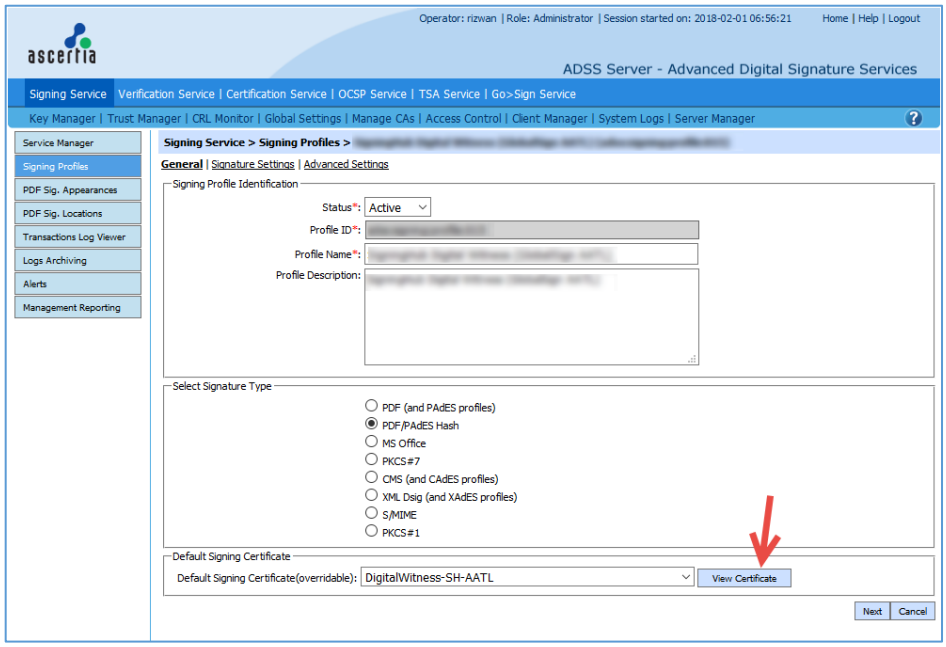
For Apple Mac, download the Go>Sign Desktop from https://www.ascertia.com/downloads/releases/go-sign-desktop/Ascertia-Go-Sign-Desktop-Mac.zip

b) Previously, if you could see the signing certificates in the certificate dropdown on the SigningHub signing dialog and now it has stopped appearing then check the validity of the

signing certificate using the relevant tools (e.g. Windows certificate viewer, Token Utility etc.). Note - signing certificates that have expired do not appear in the signing dialog!

5.2.7 Witness Certificate has expired

If SigningHub is configured to use a Digital Witness Signature, then the expiry of this certificate will cause problems! Check the Witness certificate validity by logging into the ADSS Server Admin Console. Go to **Signing Service > Signing Profile** and open the Witness Signature profile and click the **View Certificate** button to review the certificate details and check its validity.



Another window will open showing the certificate details.

5.2.8 Internal or External CA CRL has Expired

If an Internal CA CRL has expired and this is used to create long-term digital signatures, then the signature creation process will fail. To check the validity of the CA CRL, go to **CRL Monitor > CRL Details** and check the status of the CRL for the target CA. To resolve this, go to **Manage CAs > Configure Local CAs**, open the desired CA in edit mode and click **Publish CRL Now** button to generate the CRL and restart the ADSS Server from Server Manager so the CRL will be published automatically next time.

If an External CA CRL has expired, then signature creation will fail if a new CRL could not be downloaded. If a valid CRL is not available, then certificate validation will fail at the time of signature creation and an error will be returned. To check the CRL validity of an external CA in ADSS Server, refer to the section **3.6.3 External Certificate Authority OCSP / CRL Information** of ADSS Server - The Essential Guide to System Recovery document.

5.2.9 Internal or External OCSP Service issues

Online Certificate Status Protocol (OCSP) Services may be configured to provide certificate status information. This is often the case for long-term signatures. If the internal or external responder is not working properly then signature failures may result (sometimes CRLs can be used as an alternative).

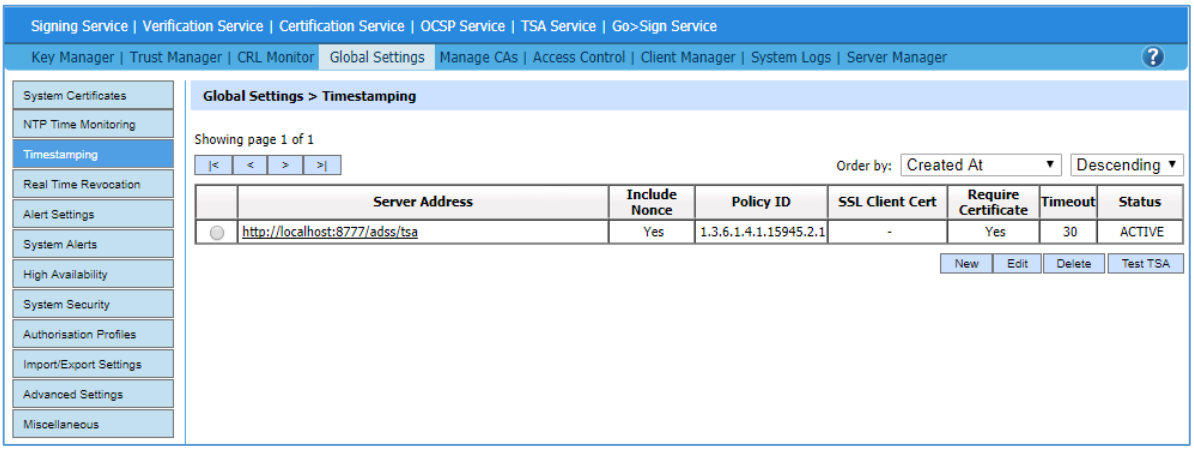
If configured check the internal OCSP Service by referring to section 3 and 6 of the ADSS Server - The Essential Guide to System Recovery document.

5.2.10 Internal or External TSA Service issues

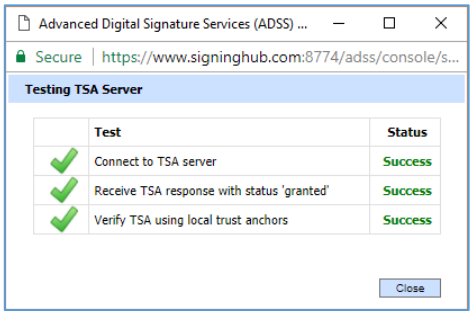
Time Stamping Authority (TSA) services are used to provide the timestamp tokens to the ADSS Signing Service at the time of signature creation.

If an internal ADSS TSA Service has been configured to provide the timestamp tokens and you suspect that it is causing the signature failure, then refer to section 3 or 5 of the ADSS Server - The Essential Guide to System Recovery document.

If an external TSA has been configured, a test tool is available within ADSS. Go to the ADSS Server Console > Global Settings > Timestamping. Select one of the configured TSAs and click Test TSA. An example Internal TSA is shown below, others can be configured.



A very clear success (or failure) message will be shown.



If errors are shown, the appropriate service provider needs to be contacted.

ADSS Server TSA Service trace logs will provide useful evidence to show where there is a request/response problem. Check that your subscription to the TSA service has not lapsed or a client TLS authentication certificate has not expired!

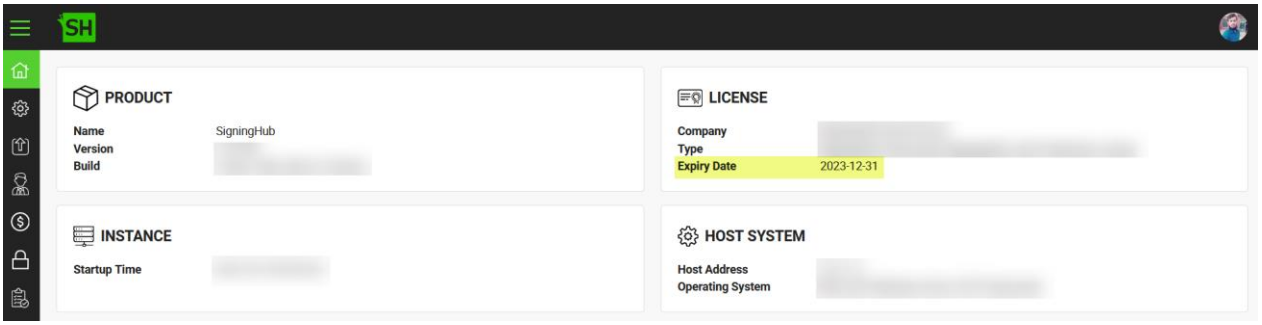
6 SigningHub - Miscellaneous Issue

Other reasons that can cause SigningHub to stop working are discussed below:

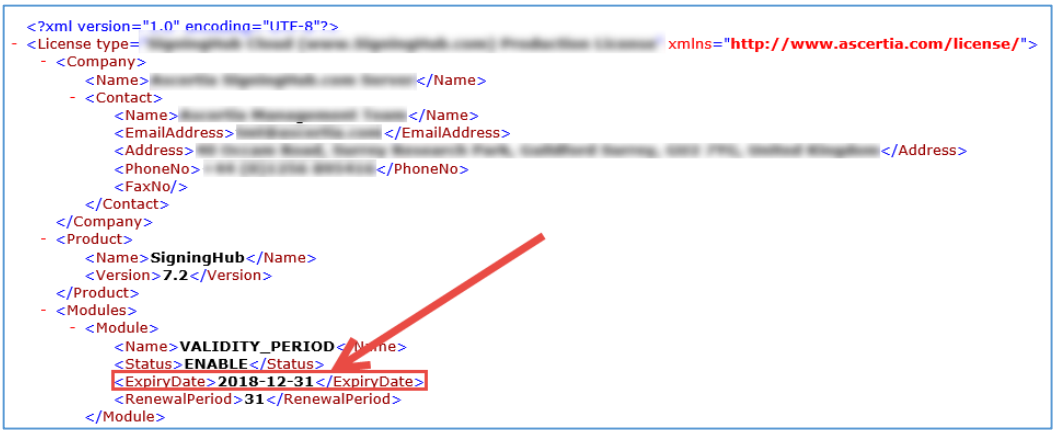
6.1 SigningHub License Expiry

Check that the SigningHub license has not expired or has run beyond issued limits.

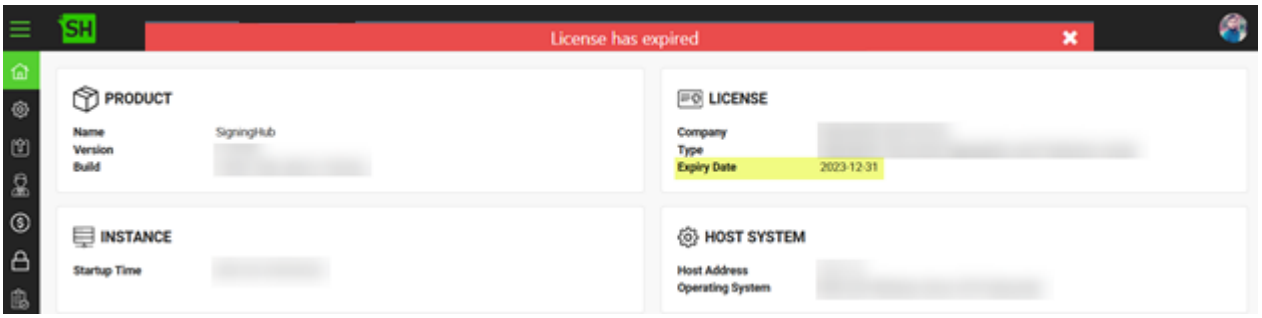
To check the SigningHub license expiry, go to **SigningHub Admin Console**,



If an Expiry Date value (as shown in the image above) is not specified, the license is 'Perpetual' and will not expire. Here is the same value in the license file:



If the license has expired, then an error message is shown like this on the SigningHub Admin Console:

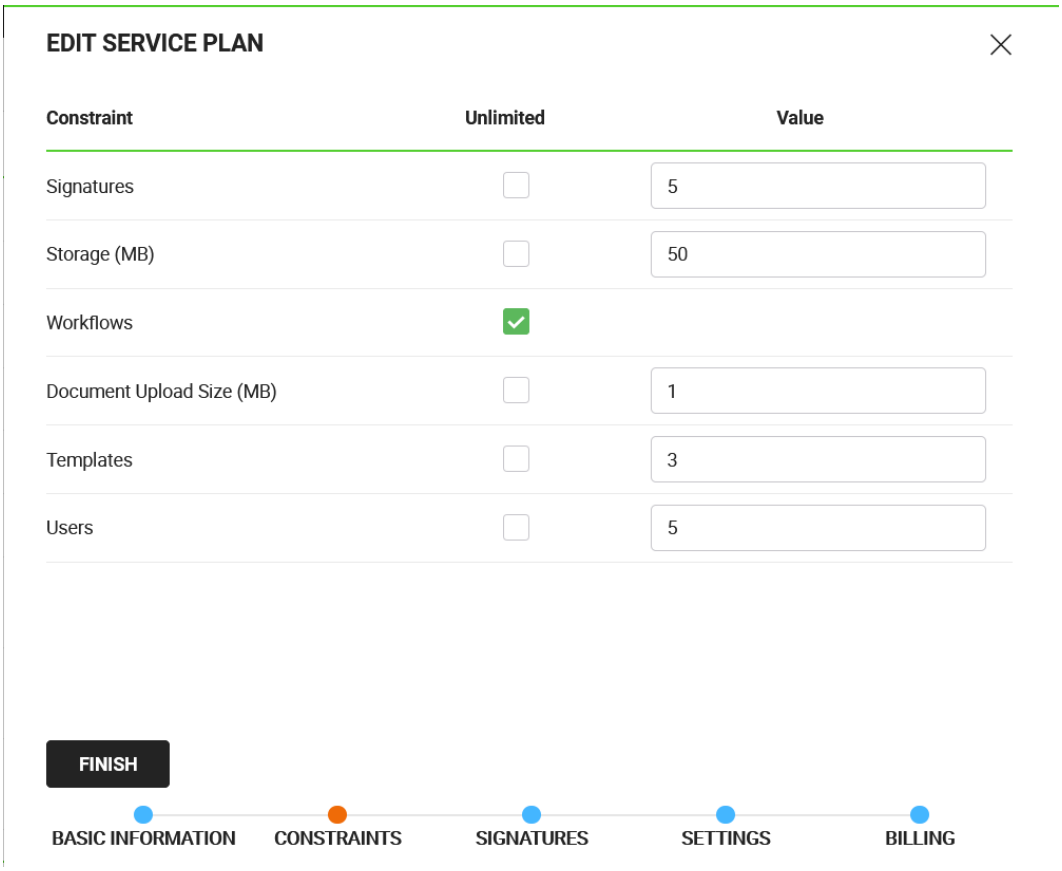


6.2 Enterprise Service Plan Limits Reached

SigningHub Service Plans are assigned to an Enterprise to control various use factors. One of the control mechanisms is how many signatures are allowed.

The SigningHub Administrator can review specific Enterprise and Individual accounts to see how many signatures have been used.

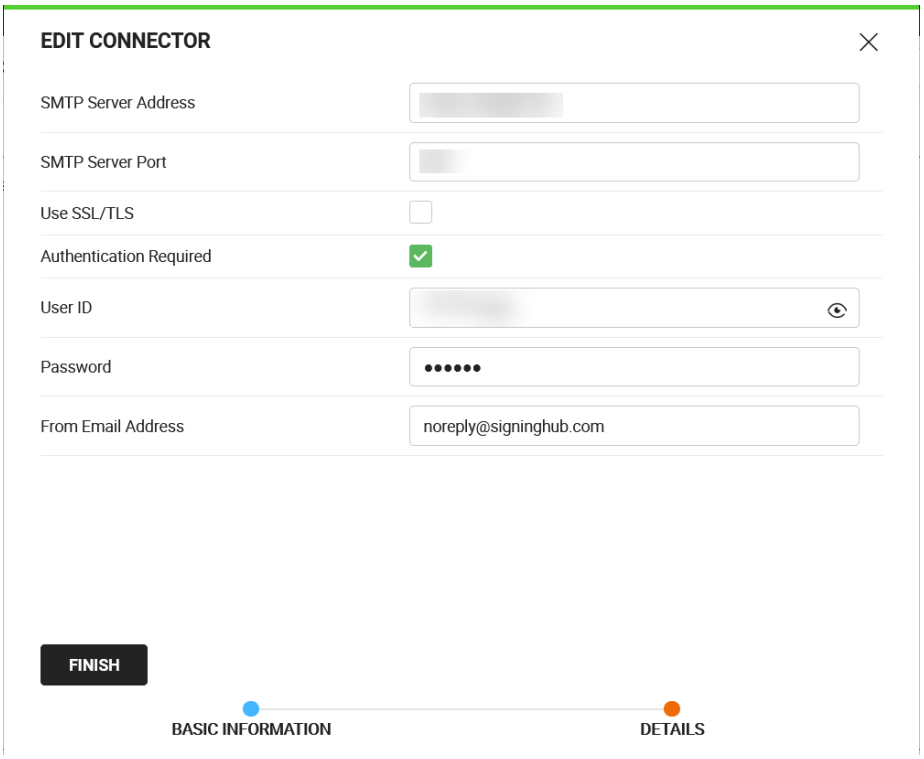
If the assigned Service Plan limits are reached, SigningHub will not allow further signatures. An account may need further signature packs to be purchased or manually changed. This can be done from **SigningHub Admin > Service Plan**, search for the appropriate enterprise or individual plan, click **Edit** icon at the end of the grid, go to **CONSTRAINTS** tab, change the values according to your needs or even you can mark them unlimited. Click the **Finish** button to save these settings



6.3 SigningHub notification emails are not delivered

If SigningHub notification emails are not being received then check the configured SMTP server's port, TLS, and authentication user's password have not expired or been changed. Multiple SMTP Servers are supported, so do check the relevant one(s).

If required update the SMTP server settings at **SigningHub Admin > Configurations > SMTP Connector**:

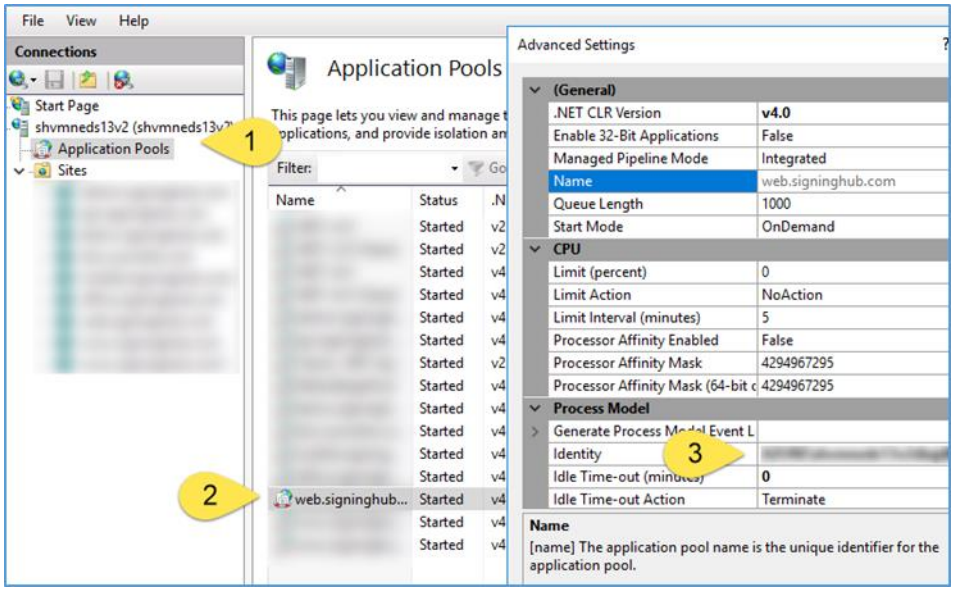


If SigningHub has been configured to use a third-party SMTP server such as SendGrid, there may be a limit on the number of emails that can be sent. If that limit is reached, emails will not be delivered to the users. Check the email statistics by logging into the SMTP portal and if the limit is reached an upgraded plan will be required.

6.4 Azure Storage Credentials Changed

If you suspect the Azure Document Storage account password has been changed then:

- Update the password on the SigningHub Server machine for the windows user from **Computer Management > Local Users and Groups > Users** screen that was created with same name as Azure Storage user.
- Update the new password against Admin, API, Core, Web in Application Pools:



*** End of Document ***