



ADSS Server A Guide to System Recovery

ASCERTIA LTD

MAY 2017

DOCUMENT VERSION- 1.0.0.6

© Ascertia Limited. All rights reserved.

This document contains commercial-in-confidence material. It must not be disclosed to any third party without the written authority of Ascertia Limited.

Commercial-in-Confidence

Contents

1	Introduction	3
1.1	Technical Support.....	3
1.2	Conventions	3
2	ADSS Server – Failure Scenarios.....	4
2.1	Document Structure.....	4
3	Host System	5
3.1	Operating System.....	5
3.2	ADSS Server Tomcat Instances.....	5
3.3	Overloaded ADSS Server	7
3.4	ADSS Server Database	7
3.5	External HSM	10
3.6	Access to External Information Sources	11
4	ADSS Server – Core Services.....	18
4.1	ADSS Server License Management.....	18
4.2	Alert Mechanisms	18
4.3	Transactions Log Archiving.....	19
4.4	ADSS Server Configuration	20
4.5	Internal CAs.....	23
5	TSA Service	26
5.1	Internal TSA Configuration.....	26
5.2	External TSA Configuration.....	26
6	OCSP Service	27
6.1	Local CA Certificate & CRL Publishing Locations	27
7	LTANS Service	29

1 Introduction

This is a vital guide for anyone managing one or more production ADSS Server instances. The document describes potential failure scenarios associated with ADSS Server production servers and provides guidance on how these can be analysed and resolved.

This document assumes that you have a production environment with one or more ADSS Servers that were previously operating perfectly and suddenly a system failure has occurred. This guide provides a comprehensive set of checks that can be made to identify the failure issue(s) and get ADSS Server running once more. It is not intended to be used as configuration guide or general diagnostic help guide or 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 ADSS Server, especially when multiple third party components are involved. This guide provides guidance and tips for issue identification and resolution in clear, logical steps.

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

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	www.ascertia.com/support
Support Email	support@ascertia.com
Skype Support	ascertia.support
Knowledge base	http://kb.ascertia.com/display/ADSS/ADSS+Server

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 ADSS Server logs. Use the Ascertia's trace log export utility to collect logs for last two days or from the date since the problems began. This will greatly assist the support team and help towards a speedy resolution. Follow this link for the instructions to run the trace log export utility:

http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=trace_logs_export_utility

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 ADSS Server – Failure Scenarios

A failure in ADSS Server can result from a number of possible issues. These can be categorised to aid problem resolution, as follows:

- Issues with the underlying operating system and host - including CPU, memory or disk space.
- Issues within third party products, including:
 - ADSS Server database – the database is a key component of ADSS Server.
 - Hardware Security Modules - relevant if an HSM or Azure Key Vault services are in use.
 - Alert transport mechanisms - including SMTP, SMS, SNMP
 - Publishing locations. For example, archive and notary components allow publishing of data records to a configured URL or physical file location.
 - External PKI related services including:
 - Time Stamp Authority
 - Trusted time sources
 - External issuing CAs
 - Trusted certificate publishing locations
 - Trusted CA CRL publishing locations and content
 - Trusted OCSP services
 - Real time certificate status database (if used)
 - Path discovery external sources such as LDAP
- Infrastructure issues that prevent access to ADSS Server from clients. These could be firewall or network related for example.
- Infrastructure issues that prevent access from ADSS Server to external CRL, OCSP, TSA or NTP services. These could be firewall or network related for example proxy server issues.
- ADSS Server configuration – an operator driven change to the configuration of any of the ADSS Server services (console, service or core components).
- ADSS Server license expiry.

There are a number of possible causes for these failure scenarios and they may be dependent upon the specific ADSS Server deployment. For example, ADSS Server deployed predominantly for signature operations could be perceived as failing because infrastructure issues are causing PKI related errors, e.g. an external TSA service or OCSP Service or CRL repository may not be responding. Ultimately, such issues manifest themselves within the ADSS Server Signing Service as a failure to complete a signing operation.

There are operational issues but not direct failings of ADSS Server. It is important to fully investigate responses received at the client along with ADSS Server transaction and trace logs before concluding that ADSS Server has failed. Often, this information will provide a useful answer.

2.1 Document Structure

Section 3 discusses how to check the host system.

Section 4 discusses the ADSS Server Core Services.

Later sections in this document explore failures from key third party components and then other sections look at specific issues that can arise for each service module including: TSA Service; OCSP Service; LTANS Service; Signing Service; Go>Sign Service; Verification Service; Certification Service.

3 Host System

As with any software application ADSS Server relies on an operating system and the physical or virtual platform. Errors to the hardware such as physical component failure are obvious. However, for requesting client applications, a network timeout or similar failure would be observed with no obvious reason as to the cause. Physical errors of components or failure of the operating system can be caused by numerous problems. Rectifying these is outside the scope of this document. However, summary information is presented to check if the host and operating system are functioning as expected.

There are certain elements to check if you suspect an underlying hardware or operating system issue.

3.1 Operating System

Check the system memory, and network and CPU utilization using operating system tools.

Now check the individual ADSS Server components usage. By default, ADSS Server components collectively require a minimum of 4GB of RAM to function although high load or high throughput deployments require more. High CPU or RAM usage by ADSS Server components indicates the system is under stress, and more resources are required. This does not always translate to more CPU or RAM. For example, it could be an exhaustion of the database connection pool or an HSM connection that is not optimised.

Check that the disk space has not been consumed by an inappropriate ADSS Server logging level. The default logging in the logs is normally set to INFO level but for high use environments this can produce too much information and for such environments the logging level ERROR should be used. DEBUG level should never be used on production systems.

None of the ADSS Server services should be continuously consuming high levels of CPU. The Core and Console instances should not use much CPU. The Service instance can consume a lot of CPU when processing a substantial load, for example of signing, verification, OCSP, TSA or other requests. The same is true for memory usage especially when large documents are passed to ADSS Server for signing or verification.

3.2 ADSS Server Tomcat Instances

Ensure that the three ADSS Server Tomcat instances: Core, Console and Service, are running. On Windows these are installed as these Windows Services:

- Ascertia-ADSS-Console
- Ascertia-ADSS-Core
- Ascertia-ADSS-Service

For Linux systems these instances are deployed as regular daemons and registered in `/etc/init.d` as:

- tomcatd-ADSS-console
- tomcatd-ADSS-core
- tomcatd-ADSS-service

If these services are not running, then refer to the local log files for each instance. Log files are located under the directory: **<ADSS_SERVER_HOME>/logs**.

Each instance has its own folder, i.e. **console**, **core**, and **service**, and therein are log files that record the start-up of these services and any possible failures. In particular, the **console.log**, **service.log**

and **core.log** files should be looked at, followed by the log files located inside the nested **tomcat** folder.

If access to the host is not possible then these same log directories are available via the ADSS Server Console. To access them, login to the administration console and select **Help > Debug Logs** as shown here:



Operator: rod | Role: Administrator | Session started on: 2017-03-09 10:28:54

Home | [Help](#) | [Logout](#)

ADSS Server - Advanced Digital Signature

Signing Service | Verification Service | Certification Service | OCSP Service | TSA Service | RA Service

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Home > System Summary

Product

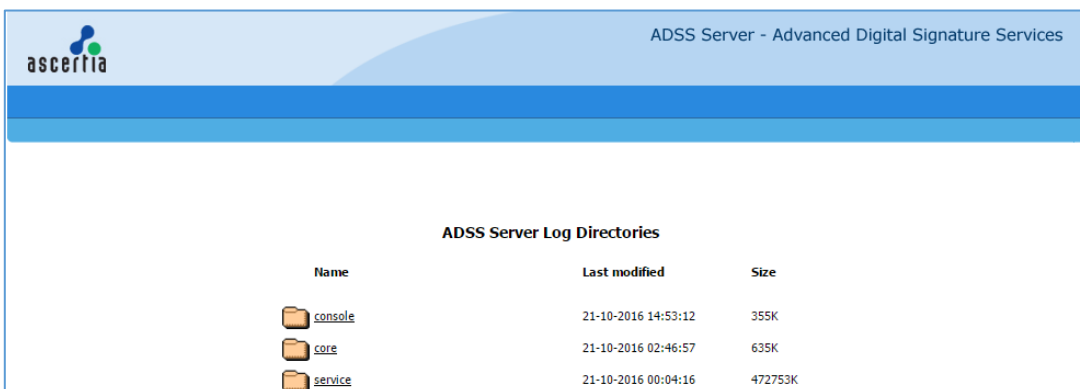
Product Name : ADSS Server - Advanced Digital Signature Services
Version : 5.0.0.7
Build : 5007.5000.250416.31407
Friendly Name : ADSS Server (10.1.0.4)

License

Company : Ascertia Signing Service US-1 Server
License Type : ADSS Signing Service Prod1 License
Contact Info : Ascertia Management Team
Contact Email : tmt@ascertia.com

[View License](#)

This provides access to log files from the three Windows Services or Unix daemons:

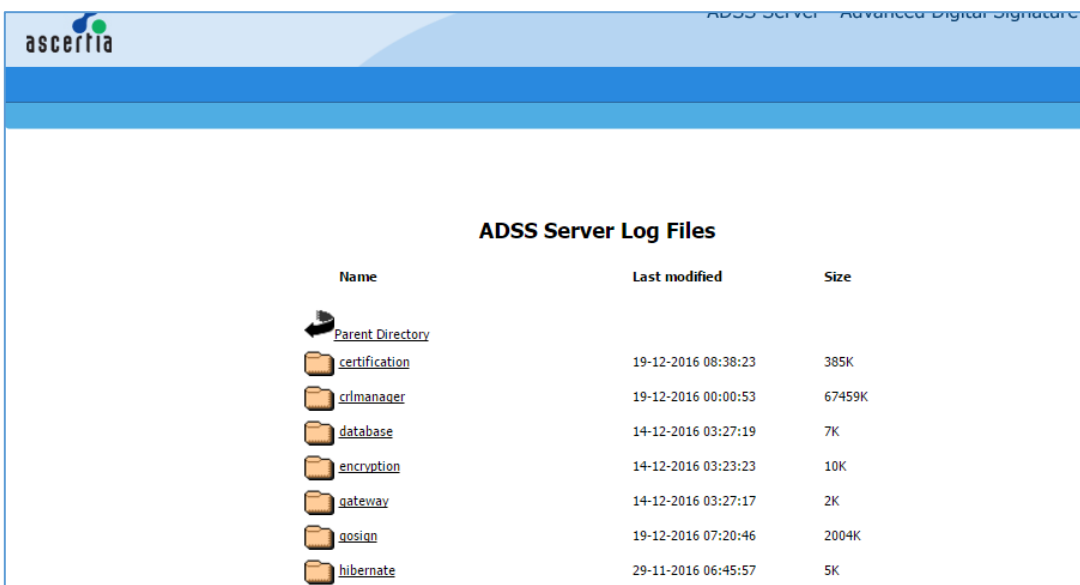


ADSS Server - Advanced Digital Signature Services

ADSS Server Log Directories

Name	Last modified	Size
console	21-10-2016 14:53:12	355K
core	21-10-2016 02:46:57	635K
service	21-10-2016 00:04:16	472753K

Underneath the 'service' folder, subfolders provide module specific logging:



ADSS Server - Advanced Digital Signature Services

ADSS Server Log Files

Name	Last modified	Size
Parent Directory		
certification	19-12-2016 08:38:23	385K
crlmanager	19-12-2016 00:00:53	67459K
database	14-12-2016 03:27:19	7K
encryption	14-12-2016 03:23:23	10K
gateway	14-12-2016 03:27:17	2K
qosign	19-12-2016 07:20:46	2004K
hibernate	29-11-2016 06:45:57	5K

There are usually multiple files in these folders, one for each day. The latest files from the current day are those which do not have a date appended to their name.

When dealing with Ascertia Support, a trace logs export utility is provided that can help provide the right logs for a defined period of time in a single zip file. Read details about this utility here:

http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=trace_logs_export_utility

3.3 Overloaded ADSS Server

Database connection configurations are located under **Global Settings > Advanced Settings > System/Core/Console/Service**. Ascertia recommends collaborating with the appropriate database administrator before making changes to the current settings. The memory and database connection usage can be monitored anytime by using the **Server Manager > System Health** function:

Signing Service Verification Service Certification Service OCSP Service TSA Service Go>Sign Service				
Key Manager Trust Manager CRL Monitor Global Settings Manage CAs Access Control Client Manager System Logs Server Manager				
Server Manager > System Health				
ADSS Server Instance	Instance Type	Status	Memory	Database Connections
signinghubr2/100.115.106.58	Core	Connected	Total Memory : 1024 MB Used Memory : 214 MB Free Memory : 810 MB Memory Status : OK	Total Connections : 100 Used Connections : 2 Free Connections : 98 Connections Status : OK
signinghubr2/100.115.106.58	Console	Connected	Total Memory : 1024 MB Used Memory : 306 MB Free Memory : 718 MB Memory Status : OK	Total Connections : 50 Used Connections : 1 Free Connections : 49 Connections Status : OK
signinghubr2/100.115.106.58	Service	Connected	Total Memory : 2048 MB Used Memory : 891 MB Free Memory : 1157 MB Memory Status : OK	Total Connections : 1000 Used Connections : 1 Free Connections : 999 Connections Status : OK

This shows the memory and database connection usage by each of the system components including load-balanced instances. If memory or database connection usage status is high then restarting the relevant component may resolve the issue.

When sending extremely large documents such as 100MB+ to ADSS Server, it may be necessary to increase the file size limit of Tomcat. Instructions on how to complete this are found here:

[http://kb.ascertia.com/display/ADSS/Tomcat+Configurations#TomcatConfigurations-ConfiguringADSSServertobeabletosignverylargedocuments\(100MB+\)](http://kb.ascertia.com/display/ADSS/Tomcat+Configurations#TomcatConfigurations-ConfiguringADSSServertobeabletosignverylargedocuments(100MB+)).

If an ADSS Server overload issue has arisen because a need to process large continuous volumes of requests then there are two options: first introduce another ADSS Server instance to distribute the load, or second, increase the system resources available to ADSS Server. This means not only memory allocation (see here for instructions on how to increase the memory allocated to each ADSS Server component: <http://kb.ascertia.com/display/ADSS/Memory+Management>) but also potentially optimising the database connection pools.

3.4 ADSS Server Database

A correctly functioning database is essential to any production ADSS Server system. Without this nothing will work. ADSS Server database issues can be investigated by:

- Verifying that the database server is accessible from the ADSS Server 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.
- If you suspect that the database credentials may have been changed then:

ADSS Server stores host, port and user identifier information in

<ADSS_Server_Home>/conf/hibernate.cfg.xml. The password is secured for security reasons. However, ADSS Server provides a script to reset the password. This is <ADSS_Server_Home>/util/bin/change_database_password.bat.

- c) Check that the database space has not been fully consumed by ADSS Server transaction log entries. Log Archiving should prevent this happening so check that archiving is enabled. OSCP and TSA service transaction log recording can be configured to record less information.
- d) Check the database is connected and the database connection pool is not exhausted by checking the “service.log”, “console.log”, “core.log” and “database.log” files in these folders/directories (or under Help in the Admin console view) and look for the phrases **“Failed to connect to database”** and/or **“Cannot get a connection, pool exhausted”**:
- [ADSS Server installation directory]/logs/service
 - [ADSS Server installation directory]/logs/service/database
 - [ADSS Server installation directory]/logs/console
 - [ADSS Server installation directory]/logs/console/database
 - [ADSS Server installation directory]/logs/core
 - [ADSS Server installation directory]/logs/core/database
- e) Another option to check the database connection usage by each of the system component is **Server Manager > System Health** function:

Signing Service Verification Service Certification Service OSCP Service TSA Service Go>Sign Service				
Key Manager Trust Manager CRL Monitor Global Settings Manage CAs Access Control Client Manager System Logs Server Manager				
Server Manager > System Health				
ADSS Server Instance	Instance Type	Status	Memory	Database Connections
signinghubr2/100.115.106.58	Core	Connected	Total Memory : 1024 MB Used Memory : 214 MB Free Memory : 810 MB Memory Status : OK	Total Connections : 100 Used Connections : 2 Free Connections : 98 Connections Status : OK
signinghubr2/100.115.106.58	Console	Connected	Total Memory : 1024 MB Used Memory : 306 MB Free Memory : 718 MB Memory Status : OK	Total Connections : 50 Used Connections : 1 Free Connections : 49 Connections Status : OK
signinghubr2/100.115.106.58	Service	Connected	Total Memory : 2048 MB Used Memory : 891 MB Free Memory : 1157 MB Memory Status : OK	Total Connections : 1000 Used Connections : 1 Free Connections : 999 Connections Status : OK

The right-hand column shows details of the database connection settings and pool. If the connection pool is exhausted, then a restart of the relevant ADSS Server component is required. In conjunction with the database administrator, decide the appropriate connection pool configurations for all three services. To change these in ADSS Server open the console and navigate to Global Settings:

Signing Service | Verification Service | Certification Service | OCSP Service | TSA Service | SCVP Service | LTANS Service | OCSP Monitor | Go>Sign Service | RA Service | OCSP Repeater

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Global Settings > Advanced Settings

Note: Click on the property value to update it

Property Type:

Property ID	Property Value
Time interval to publish the user certificates on an LDAP server when ADSS Server is configured for this purpose. Default value: 5 minutes	
CERTIFICATE_PUBLISHING_INTERVAL	5
Protocol used by the ASC_DirectoryThread to communicate with Service instance when ADSS Server is configured to generate the certificates against the Active Directory users. Default protocol: HTTP	
DEFAULT_SERVICE_PROTOCOL	http
Port used by the ASC_DirectoryThread to communicate with Service instance when ADSS Server is configured to generate the certificates against the Active Directory users. Default port: 8777	
DEFAULT_SERVICE_PORT	8777
Used to display the database connections details of logs. Default value: False To enable logging, uncomment the "File Appenders for Temporary c3p0" appender in core/log.properties file as well.	
ENABLE_DB_CONNECTION_COUNTER	TRUE
DATABASE_MONITORING_INTERVAL	
1	
Number of seconds a query will wait for the database to return the results before terminating the connection. Default value: 180	
CONNECTION_POOL_QUERY_TIMEOUT	180
Minimum number of connections a pool will maintain at any given time for ADSS Server Core instance. Default value: 30	
hibernate.c3p0.minPoolSize	30
Maximum number of Connections a pool will maintain at any given time for ADSS Server Core instance. Default value: 100	
hibernate.c3p0.maxPoolSize	100
Seconds a connection can remain pooled but unused before being discarded. Zero means idle connections never expire. Default value: 86400	
hibernate.c3p0.maxIdleTime	10
Determines how many connections at a time c3p0 will try to acquire when the pool is exhausted. Default value: 10	
hibernate.c3p0.acquireIncrement	10
The number of milliseconds a client calling getConnection() will wait for a connection to be checked-in or acquired when the pool is exhausted. Zero means wait indefinitely. Setting any positive value will cause the getConnection() call to time-out and break with an SQLException after the specified number of milliseconds. Default value: 600000	
hibernate.c3p0.checkoutTimeout	600000

New

Note Core, Console and Service settings are accessible from the **Property Type** dropdown menu. There may be a need to adjust the database connection properties to match the actual needs. The instructions to configure the database connection parameters are found here:

<http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=core>

<http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=console>

<http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=service>

The number of database connections currently opened by ADSS Server should also be checked. The instructions are found here:

<http://kb.ascertia.com/display/ADSS/Useful+ADSS+Server+Database+SQL+Commands#UsefulADSSServerDatabaseSQLCommands-Checkingtheopenconnectionsonthedatabase>.

Ask the database administrator to check the ADSS Server database size versus what is allowed. If size is an issue, a review of the transaction logs stored by ADSS Server maybe necessary, and changes may be required to the configured archiving period and frequency. For example, archive configurations settings for CRL Monitor are shown in:

http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=CRL_Logs_Archiving&SearchHighlight=archiving&condition=exactphrase

Similarly, all services have their separate archive configurations settings.

TSA and OCSP transaction log sizes can be reduced by configuring only specific information to be written to the database. Instructions for this are at:

http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=tsa_settings

http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=ocsp_service2

ADSS Server has automated reconnection features that keep trying to contact and re-establish communication with a database. These do eventually time-out. An ADSS Server service restart or a full Windows or Unix service or system restart will kick-start the reconnection process.

Keeping database backups, managing restores, managing configurations and database failover situations are outside of the scope of this document - these must be addressed by the database administrator. A database service restart or system restart may resolve an issue with the database. Restoring a recent database backup may be one way to quickly restart a system, but transaction log data may well be lost.

3.5 External HSM

The use of an HSM is optional and so this section can be ignored if an HSM is not being used.

Access to a network HSM should be checked first using standard network tools such as ping and trace route, second, vendor utilities (e.g. 'vtl verify' from the SafeNet Luna Client software suite), and finally using ADSS Server GUI and accompanying test utility.

Basic connectivity checks will reveal if there are infrastructure or HSM specific issues. These must be resolved by your staff assigned to HSM support or the HSM vendor. Once connectivity is established the vendor specific test tools will reveal if the appropriate configuration for ADSS Server are in place. For example, a dedicated partition is available on a Luna HSM, or on other HSMs a slot number should be available.

ADSS Server has an automated reconnection feature for HSMs and this reconnect process will be visible in the log files. The Key Manager module has an HSM Test Connection feature:

Key Manager > Crypto Source

Showing page 1 of 1

Order by: Created At Descending

	Profile Friendly Name	PKCS#11 Slot	PKCS#11 Module	Crypto Source	Key Encrypting Key	Status
<input checked="" type="radio"/>	Software Profile [Default]	--	--	Software	--	Available
<input type="radio"/>	MSCAPI Profile	--	--	MSCAPI	--	Available
<input type="radio"/>	Eracom	2	cryptoki.dll	PKCS#11	--	Available
<input type="radio"/>	Software Key Vault	--	--	Azure Key Vault	--	Available
<input type="radio"/>	Hardware Key Vault	--	--	Azure Key Vault	--	Available

Make Default Test Connection Import Existing Keys New Edit Delete

Select the appropriate HSM profile and click the **Test Connection** button. Alternatively, select the profile and choose **Edit**. This will allow ADSS Server operator to check the partition/slots available and verify these with the HSM administrator:

Key Manager > Crypto Source > New

Crypto Profile Settings

Status: Active

Friendly Name*: Eracom

Crypto Source Type*: PKCS#11

PKCS#11 Module*: cryptoki.dll

PKCS#11 Slot*: 0

PKCS#11 PIN*:

PKCS#11 Connection Pool Size: 30

PKCS#11 Monitoring Interval: 1 (min)

☐ Enable FIPS Mode

☐ Import Certificates to Device

ADSS Server includes a command line HSM test utility:

```
<ADSS_Server_Home>/util/bin/test_pkcs11.bat
```

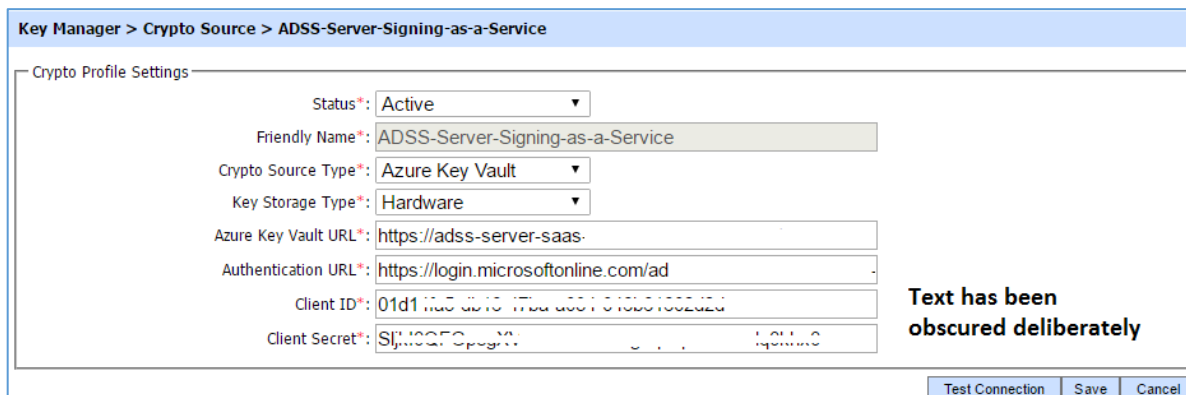
This utility performs a full set of interoperability tests on the target PKCS#11 crypto device (software or hardware). For example, creation of RSA key pair and signing operations.

This utility is far more powerful and feature rich than the simple connectivity tests and can be used to prove that the HSM supports all of the required functionality to work with ADSS Server. We have seen firmware updates change the behaviour of HSMs so do test these first before deploying into a production environment.

Full details are provided in the admin guide:

http://manuals.ascertia.com/ADSS-Admin-Guide/default.aspx?pageid=running_the_hardware_crypto_source_test_utility

ADSS Server also supports Azure Key Vault as an external HSM. As the Azure Key Vault is accessed over the Azure Network, there could be connectivity problem prohibiting access to the Key Vault itself or important infrastructure or signing keys:



Key Manager > Crypto Source > ADSS-Server-Signing-as-a-Service

Crypto Profile Settings

Status*: Active

Friendly Name*: ADSS-Server-Signing-as-a-Service

Crypto Source Type*: Azure Key Vault

Key Storage Type*: Hardware

Azure Key Vault URL*: https://adss-server-saas-

Authentication URL*: https://login.microsoftonline.com/ad

Client ID*: 01d11a5d-3b13-472a-8087-010001000000

Client Secret*: Sj4W32FSp9gM: [obscured]

Text has been obscured deliberately

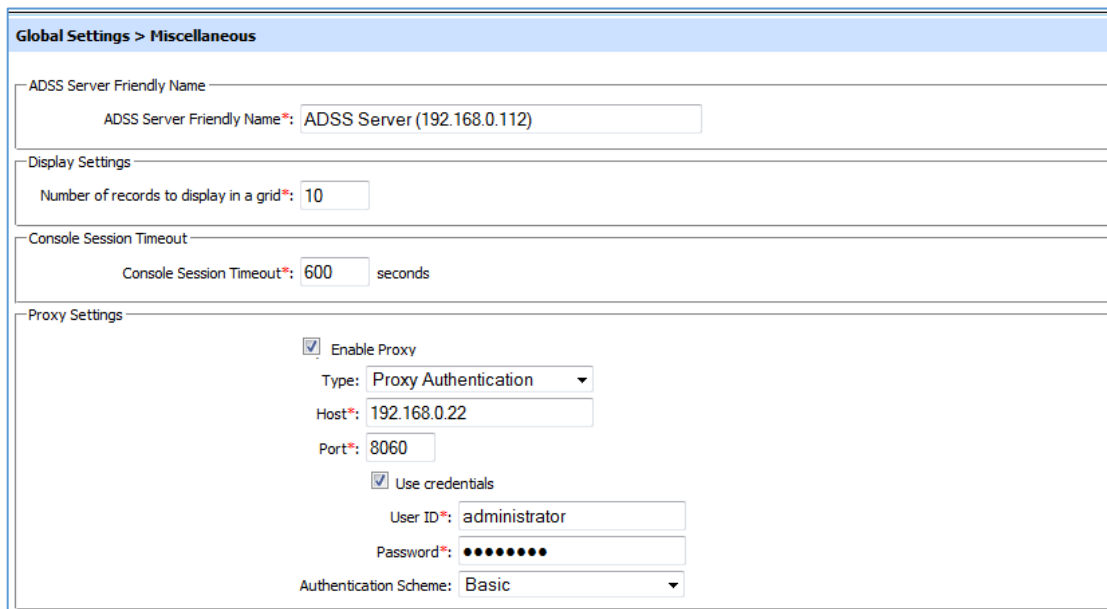
Test Connection Save Cancel

3.6 Access to External Information Sources

The following sections cover all external information sources.

3.6.1 Proxy Configuration

The proxy configuration for ADSS Server is used when there is a requirement for external communication. For example, to a third-party TSA or OCSP service. In this instance, the proxy and interconnecting infrastructure is critical to the continued success of ADSS Server functionality. The proxy is configured under **Global Settings > Miscellaneous**. Although no internal test function exists for the proxy settings, it does provide the connection details that can be verified by the proxy owner. Connectivity can be tested using third party tools. For example, telnet client will indicate if the proxy can at least be reached:



The screenshot shows the 'Global Settings > Miscellaneous' configuration page. It contains several sections: 'ADSS Server Friendly Name' with a text field containing 'ADSS Server (192.168.0.112)'; 'Display Settings' with a text field for 'Number of records to display in a grid' set to '10'; 'Console Session Timeout' with a text field for 'Console Session Timeout' set to '600' seconds; and 'Proxy Settings' which includes a checked 'Enable Proxy' checkbox, a 'Type' dropdown set to 'Proxy Authentication', 'Host' and 'Port' text fields set to '192.168.0.22' and '8060' respectively, a checked 'Use credentials' checkbox, 'User ID' and 'Password' text fields set to 'administrator' and masked with dots, and an 'Authentication Scheme' dropdown set to 'Basic'.

If configured, another option to verify the correctness of proxy settings is by going to the relevant OSCP or TSA configuration screens (already covered above) and then Test the connectivity using the provided Test function. If test shows that TSA or OSCP are accessible, it means the proxy configuration are correct and working fine.

3.6.2 Network Connectivity to ADSS Server

Given the nature of ADSS Server it is likely to be protected with firewalls or similar Unified Threat Management devices. This can lead to a loss of connectivity to critical resources. Ascertia recommends that basic connectivity to ADSS Server is tested accordingly in the event of perceived failure.

For any business application, ADSS Server services are available via three ports; namely 8777, 8778, and 8779. Telnet or similar network tools should be used to test connectivity to these ports from business applications/clients. In addition, the administration console is accessible on port 8774.

When business applications are unable to connect with the ADSS Server, then verify that ADSS Server system is network accessible from the client system using ping, telnet and trace route. If it is not accessible, check if ADSS Server is accessible from the same host and then local subnet. Establishing a functioning service that cannot be accessed externally is a connectivity issue, and hence outside the scope of this document.

If configured in a high availability set-up, all slave instances of ADSS Server Core and Console communicate with the master instance on ports 8773 and 8770 respectively. These can be tested using telnet or similar network tools to ensure the infrastructure connectivity between the two instances. To check all modules of the cluster are operating correctly and that the correct designated Master is as intended, check under **Global Settings**:

Global Settings > High Availability

Core High Availability Settings

Slave should check Master active status every (sec)*: 10

Number of times slave should re-check before becoming Master*: 3

Core Host*: laptop-jawad (Master) : ONLINE
localhost.localdomain (Slave) : ONLINE

Note: For each subsequent slave the number of times to re-check if master is still inactive is doubled from the previous slave.

Remove

Console High Availability Settings

Slave should check Master active status every (sec)*: 10

Number of times slave should re-check before becoming Master*: 3

Console Host*: laptop-jawad (Master) : ONLINE
localhost.localdomain (Slave) : ONLINE

Note: For each subsequent slave the number of times to re-check if master is still inactive is doubled from the previous slave.

Remove

3.6.3 External Certificate Authority OCSP / CRL Information

Revocation information in the form of OCSP services or CRL files is required for operations such as long term signing and verification, OCSP, SCVP and XKMS services within ADSS Server. When using an external CA, ADSS Server must have access to this information.

Access to external CRL and OCSP sources can be tested using the **Trust Manager** module of ADSS Server (**CRL Monitor** maybe used for CRLs as well but **Trust Manager** allows access to both CRLs and OCSP). The following shows how to configure and test an OCSP Responder for a given CA from with **Trust Manager**:

ADSS Server – Advanced Digital Signature Services

Signing Service | Verification Service | Certification Service | OCSP Service | TSA Service | SCVP Service | LTANS Service | OCSP Monitor | Go>Sign Service | RA Service | OCSP Repeater

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Trust Manager > Entrust CA for Adobe

General | **Validation Policy** | CRL Settings | Advanced Settings

Revocation Settings

Primary Method*: OCSP

Available Methods: AIA

Selected Methods: Configured OCSP Addresses

Secondary Method*: NONE

OCSP Responder Settings

OCSP Responder Address:

Add

List of OCSP Responders: http://ocsp.entrust.net

Remove Test

OCSP Request Settings

☐ Enable certificate status checking for responder's certificate

☒ Set Nonce

☐ Set Service Locator

☐ Sign OCSP Request

☐ Check OCSP responder is authorised by the CA

Hash Algorithm: SHA256

OCSP Response Clock Tolerance: 100 (sec)

Response Timeout: 10 (sec)

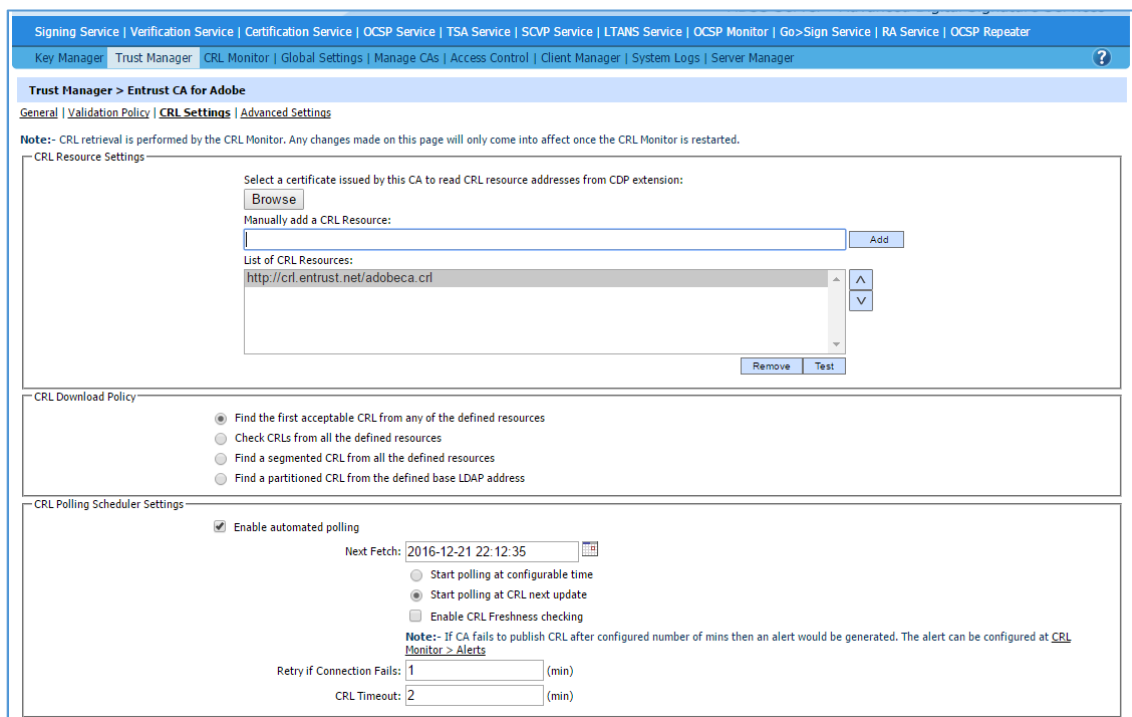
* Set to 0 if the timeout is unlimited

Next Save Cancel

If an OCSP Server has been defined then access to it can be tested by highlighting the target address and using the **Test** button. Check that the correct hash algorithm is being used. Some external OCSP Responders still insist on SHA-1 based requests and do not accept SHA-2.

If the AIA option is used then you will need to inspect an external certificate and manually extract and then check the OCSP address provided within the AIA extension.

To test the CRL location, use the **CRL Settings** page:



The screenshot shows the 'Trust Manager > Entrust CA for Adobe' page. The 'CRL Settings' tab is active. A note states: 'Note:- CRL retrieval is performed by the CRL Monitor. Any changes made on this page will only come into affect once the CRL Monitor is restarted.'

CRL Resource Settings

Select a certificate issued by this CA to read CRL resource addresses from CDP extension:

Manually add a CRL Resource:

List of CRL Resources:

http://crl.entrust.net/adobeca.crl

CRL Download Policy

- ☒ Find the first acceptable CRL from any of the defined resources
- ☐ Check CRLs from all the defined resources
- ☐ Find a segmented CRL from all the defined resources
- ☐ Find a partitioned CRL from the defined base LDAP address

CRL Polling Scheduler Settings

☒ Enable automated polling

Next Fetch: 2016-12-21 22:12:35

- ☐ Start polling at configurable time
- ☒ Start polling at CRL next update
- ☐ Enable CRL Freshness checking

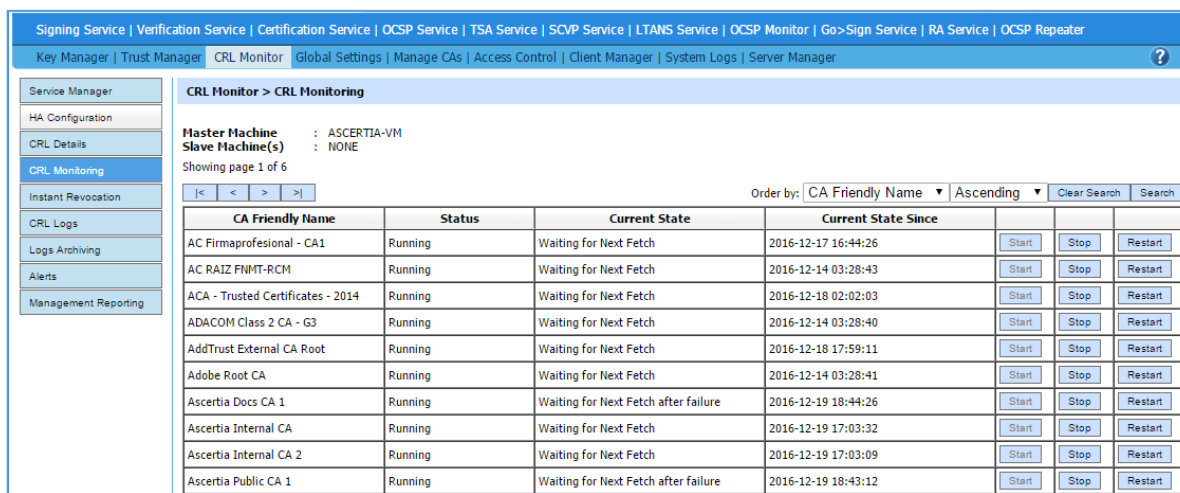
Note:- If CA fails to publish CRL after configured number of mins then an alert would be generated. The alert can be configured at [CRL Monitor > Alerts](#)

Retry if Connection Fails: 1 (min)

CRL Timeout: 2 (min)

Access to the CRL location can be checked by highlighting the target address and clicking the Test button.

Note the **CRL Monitor** will have an entry for each CA that has CRL Polling enabled (as shown above). It is important to check the status of each CA since this provides a view of all CRLs that ADSS Server is polling for:



The screenshot shows the 'CRL Monitor > CRL Monitoring' page. It displays a table of CA entries with columns: CA Friendly Name, Status, Current State, and Current State Since. Each row has Start, Stop, and Restart buttons.

CA Friendly Name	Status	Current State	Current State Since	Start	Stop	Restart
AC Firmaprofesional - CA1	Running	Waiting for Next Fetch	2016-12-17 16:44:26	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
AC RAIZ FNM-T-RCM	Running	Waiting for Next Fetch	2016-12-14 03:28:43	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
ACA - Trusted Certificates - 2014	Running	Waiting for Next Fetch	2016-12-18 02:02:03	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
ADACOM Class 2 CA - G3	Running	Waiting for Next Fetch	2016-12-14 03:28:40	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
AddTrust External CA Root	Running	Waiting for Next Fetch	2016-12-18 17:59:11	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
Adobe Root CA	Running	Waiting for Next Fetch	2016-12-14 03:28:41	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
Ascertia Docs CA 1	Running	Waiting for Next Fetch after failure	2016-12-19 18:44:26	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
Ascertia Internal CA	Running	Waiting for Next Fetch	2016-12-19 17:03:32	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
Ascertia Internal CA 2	Running	Waiting for Next Fetch	2016-12-19 17:03:09	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>
Ascertia Public CA 1	Running	Waiting for Next Fetch after failure	2016-12-19 18:43:12	<input type="button" value="Start"/>	<input type="button" value="Stop"/>	<input type="button" value="Restart"/>

Access to CRL could be over HTTP/S or LDAP/S protocols and ADSS Server makes client requests to these external resources. If they are no longer accessible to ADSS Server, it is likely to be because

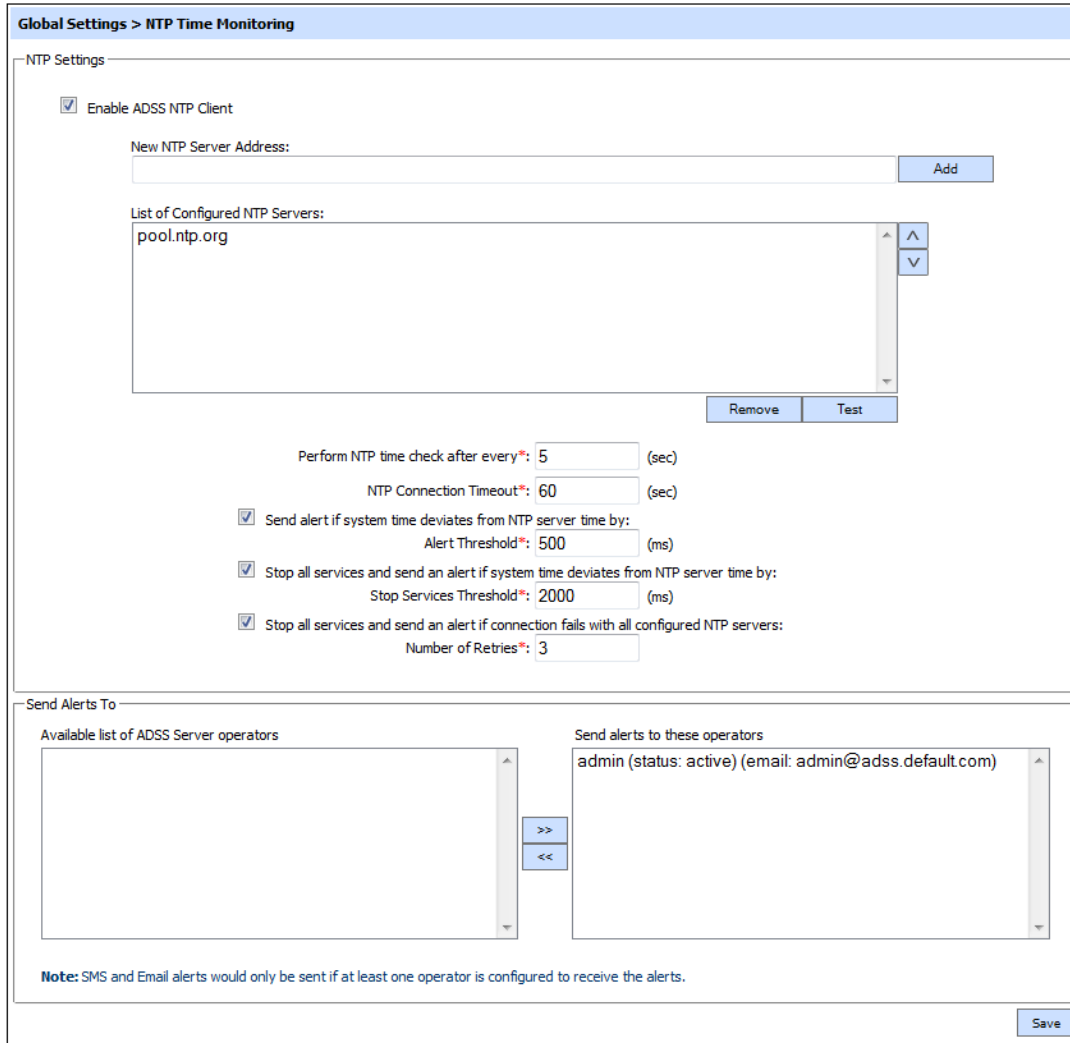
infrastructure issues are preventing access. These issues are outside of the scope of ADSS Server for investigation, but the CRL Monitor log file will report if the resource is not available or access denied, or connection is not possible:

```
<ADSS_Server_Home>/logs/service/crlmanager
```

3.6.4 External NTP Servers for Time Monitoring

ADSS Server has an optional NTP Time Monitoring service, which will need access to third-party trusted time sources, either internet based or internal GPS or Radio signal NTP servers.

If access to these fails then ADSS Server may be configured to stop (see the fourth tick box):



The screenshot shows the 'Global Settings > NTP Time Monitoring' configuration window. It is divided into two main sections: 'NTP Settings' and 'Send Alerts To'.

NTP Settings:

- ☒ Enable ADSS NTP Client
- New NTP Server Address: Add
- List of Configured NTP Servers:

pool.ntp.org

Remove Test
- Perform NTP time check after every*: 5 (sec)
- NTP Connection Timeout*: 60 (sec)
- ☒ Send alert if system time deviates from NTP server time by:
 - Alert Threshold*: 500 (ms)
- ☒ Stop all services and send an alert if system time deviates from NTP server time by:
 - Stop Services Threshold*: 2000 (ms)
- ☒ Stop all services and send an alert if connection fails with all configured NTP servers:
 - Number of Retries*: 3

Send Alerts To:

- Available list of ADSS Server operators:
- Send alerts to these operators:

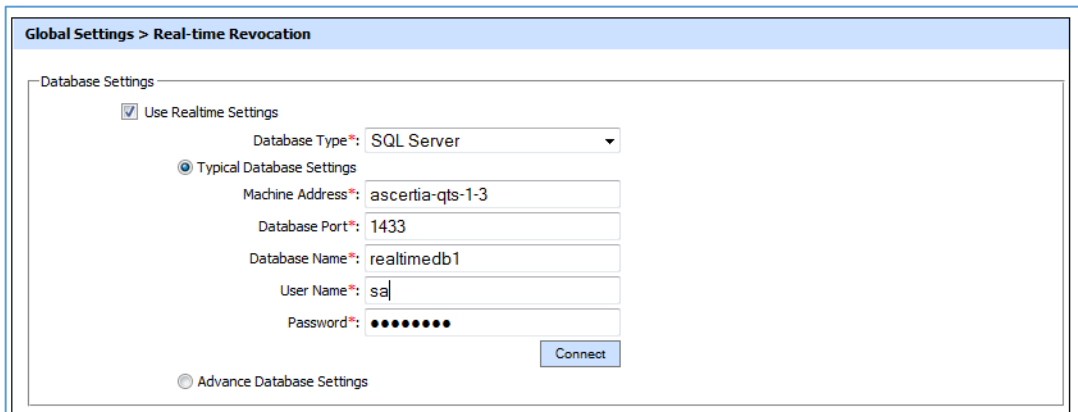
admin (status: active) (email: admin@adss.default.com)
- >> <<
- Note:** SMS and Email alerts would only be sent if at least one operator is configured to receive the alerts.
- Save

Select each of the NTP servers in turn and click the **Test** function to check the ability to connect with the external NTP Server.

Resolution of a fault when communicating with these external resources is outside the scope of Ascertia support services – speak to your Network Specialists. Before calling them, do check this is not due to a fault with the internal network / firewalls / proxy servers.

3.6.5 External Real Time Certificate Database

To provide real time full certificate status information, ADSS Server can be configured to use a second external database. This is completely separate to the main ADSS Server database:

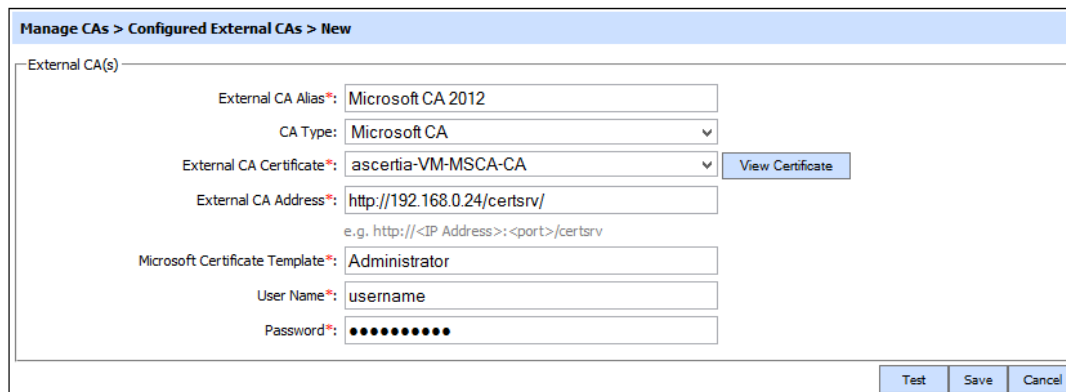


The **Connect** function can be used to check the connectivity with the configured external real time certificate database.

Resolution of a fault with the external database system is outside the scope of Ascertia support services – speak to your DBA. Before calling them, do check this is not due to a fault with the internal network / firewalls / proxy servers.

3.6.6 Manage CAs - External Issuing CAs

ADSS Server can work with multiple internal or external CAs. It supports multiple external CAs from many providers such as EJBCA and Microsoft Windows CA / Active Directory Certificate Services:



For each configured external CA, clicking the **Test** button will check the connectivity with the external CA address. Do this for all appropriate CAs.

Resolution of a fault with the external CA is outside the scope of Ascertia support services – speak to your CA service provider. Before calling them, do check this is not due to a fault with the internal network / firewalls / proxy servers.

3.6.7 Certificate Path Discovery

ADSS Server has SCVP, Verification, and XKMS modules to perform certificate validation. One particular element of this process is certificate path discovery whereby an attempt is made to build a chain of trust from the end entity certificate to a trusted root. ADSS Server supports path discovery using various methods. One of these is searching an external LDAP compliant directory. This access can be checked as shown here for SCVP:

SCVP Service > Validation Policy > Path Discovery Settings

General | Trust Anchors Settings | **Path Discovery Settings** | Path Validation Settings | Advance Settings

☐ Use basic path discovery
☒ Use advanced path discovery

☒ Build path using certificates registered in ADSS Trust Manager
☒ Build path using certificates provided in request
☒ Build path using Subject certificate's AIA extension
☒ Build path using certificates found in locally-configured LDAP directories

LDAP Repository Settings

Enter LDAP Repository Address:

List of LDAP Repositories Addresses:

Final Trust Point

☐ Build and validate certificate path up to any CA registered in ADSS Trust Manager
☒ Build and validate certificate path up to a self-signed Root CA registered in ADSS Trust Manager

Highlight the LDAP repository and click the **Test** button to check connectivity.

Resolution of a fault with the external LDAP is outside the scope of Ascertia support services – speak to your CA service provider. Before calling them, do check this is not due to a fault with the internal network / firewalls / proxy servers.

3.6.8 External Timestamp Authority

When signing or verifying, the external TSAs can be associated with particular CAs. A **Test TSA** button is available on the **Global Settings** screen to confirm if the TSA can be reached.

It may be that a TSA will reject a SHA-1 algorithm as systems move to mandating SHA-256 and above. Check that both algorithms are accepted if required.

If using a client SSL certificate to authenticate access to the TSA then check the certificate has not expired.

Global Settings > Timestamping > Update

Note:- Make sure either the TSA certificate or its issuer is added in Trust Manager with the purpose "Time Stamping Authority" or "CA" respectively.

TSA Settings

Status:

TSA Server Address*:

Policy ID:

Timeout*: (sec)

☒ Include Nonce
☒ Require TSA certificates
☐ Perform revocation status checking for TSA certificates
☐ TSA requires authentication

Test Hash Algorithm Settings

Hashing Algorithm*:

4 ADSS Server – Core Services

This section walks through the common services and allows you to review and check whether these are functioning correctly.

4.1 ADSS Server License Management

Each ADSS Server installation is licensed and licensing can be based on the expiry date as well as number of transactions allowed. There could be a situation that the administrator has not noticed and in fact the ADSS Server license has expired. Where annual licenses have been purchased an expiry date will have been set. Where limited use licenses have been agreed, a limit may be set.

The license details can be checked from **Help > License** option:

License Detail						
Installation Date/Time:		2011-08-08 06:47:36				
License Type:		ADSS Server SigningHub Production License				
Company:		Ascertia SigningHub.com Server				
Contact Details:		Name: Ascertia Management Team				
		Email Address: tmt@ascertia.com				
		Address: 40 Occam Road, Surrey Research Park, Guildford Surrey, GU2 7YG, United Kingdom				
		Phone#:				
		Fax#:				
Module Name	License Limit	Quantity Remaining	Validity Period (Days)	Expiry Date	Renewal Period (Days)	Status
SIGNING_SERVICE	-	-	-	2018-12-365	28	318 day(s) left
PROFILES	-	-	-	-	-	-
PKCS7	-	-	-	-	-	-
PDF	-	-	-	-	-	-
XML	-	-	-	-	-	-
SMIME	-	-	-	-	-	-
HASH	-	-	-	-	-	-
LONGTERM_SIGNATURES	-	-	-	-	-	-
SIG_G_SIGNATURES	-	-	-	-	-	-
AUTHORISATION_PROFILES	-	-	-	-	-	-
MS_OFFICE	-	-	-	-	-	-
VERIFICATION_SERVICE	-	-	-	2018-12-365	28	318 day(s) left
PROFILES	-	-	-	-	-	-
PKCS7	-	-	-	-	-	-
PDF	-	-	-	-	-	-
XML	-	-	-	-	-	-
HISTORICAL_VALIDATION	-	-	-	-	-	-
LONGTERM_SIGNATURES	-	-	-	-	-	-
MS_OFFICE	-	-	-	-	-	-
CERTIFICATION_SERVICE	-	-	-	2018-12-365	28	318 day(s) left
PROFILES	-	-	-	-	-	-
DIRECTORY_INTEGRATION	-	-	-	-	-	-
OCSP_SERVICE	-	-	-	2018-12-365	28	318 day(s) left
TRUSTED_AUTHORITIES	-	-	-	-	-	-
TSA_SERVICE	-	-	-	2018-12-365	28	318 day(s) left
PROFILES	-	-	-	-	-	-

By examining the table, you can easily see if there is an expiry date or zero quantity remaining for certain operations.

4.2 Alert Mechanisms

ADSS Server has built-in alert functionality to allow administrators and operators to receive alerts configured per service or core component. There are three options available: SMTP; SMS; and SNMP. Note that not all options are required and all options may even not be enabled. Each medium has a test functionality within ADSS Server Console. The screen shot below shows an example and the **Global Settings** screen location:

Operator: craig | Role: Administrator | Session started on: 2016-11-30 16:18:55 | Home | Help | Logout

ADSS Server - Advanced Digital Signature Services

Signing Service | Verification Service | Certification Service | OCSP Service | TSA Service | RA Service

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Global Settings > Alert Settings

Email Settings

☒ Enable email alerts

Email Server Address*: mail.ascertia.com

Port*: 25

Email Address*: adss-us1@ascertia.com

Email Sleep Interval*: 30 (sec)

Email Failure Retries*: 3

☐ Use SSL authentication

☒ Use username/password based authentication

User ID*: adss-us1@ascertia

Password*:

Test Email

SMS Settings

☒ Enable SMS alerts

SMS Server Address*: http://api.clickatell.com/http/sendmsg

User ID*: asc

Password*:

Vendor ID*: 3125133

Test SMS

SNMP Settings

☒ Enable SNMP alerts

SNMP Version*: 1

IP Address*: 127.0.0.1

Port*: 162

Variable OID*: 1.2.3.4.5.6.7.8

Community String*: public

Timeout*: 30 (sec)

SNMP Failure Retries*: 3

Test SNMP

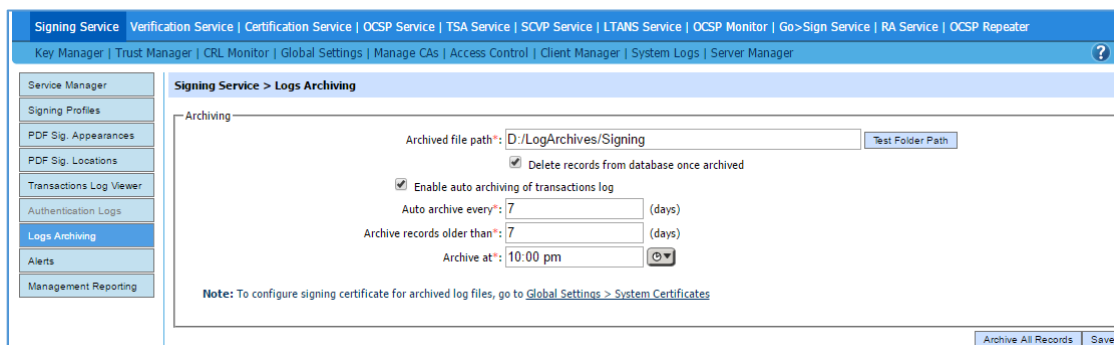
Save

Testing these will indicate if any infrastructure or authentication errors exist, or if the service does not respond as expected. Note these settings apply system wide, so wherever an alert option is configured within the system, for example within the TSA Service, these underlying settings will determine how the alert message is delivered.

4.3 Transactions Log Archiving

Each individual service produces transactions log for audit purposes. These are written to the ADSS Server database. The transactions log data must be managed carefully to avoid bloating of the ADSS Server database. ADSS Server achieves this by incorporating the transactions log archiving capability for each individual service, as well as for the overall system logs. Log archiving for each type of log can be uniquely configured and each configuration relies on a physical file host to write the data to when it is removed from the database.

The archive operations are run on a schedule, either by number of days, or age of records. Hence effective automated processing relies on the configuration being correct and appropriate for the load.



The screenshot shows the 'Signing Service > Logs Archiving' configuration page. The left sidebar contains a menu with items: Service Manager, Signing Profiles, PDF Sig. Appearances, PDF Sig. Locations, Transactions Log Viewer, Authentication Logs, Logs Archiving (selected), Alerts, and Management Reporting. The main content area is titled 'Archiving' and contains the following settings:

- Archived file path*:
- ☒ Delete records from database once archived
- ☒ Enable auto archiving of transactions log
- Auto archive every*: (days)
- Archive records older than*: (days)
- Archive at*:

A note at the bottom states: "Note: To configure signing certificate for archived log files, go to [Global Settings > System Certificates](#)". At the bottom right, there are buttons for 'Archive All Records' and 'Save'.

The **Test Folder Path** function allows an administrator to ensure that ADSS Server can write to the specified location.

Clicking **Archive All Records** is not recommended during troubleshooting. All records in the database for this service will be exported and signed. This can take a significant amount of time and resources.

4.4 ADSS Server Configuration

Changes to a system are generally the first item to look at when a failure occurs. That is, what has changed since operations began to fail. Often it is a simple case of reviewing changes made, and reverting them. ADSS Server allows this through the **System Logs** module.

Each service and module logs the debug information to flat files. These are an invaluable source of information when it comes to problem deduction and resolution. Access to these logs is already covered above in section 3.2 ADSS Server Tomcat Instances.

4.4.1 System Logs

ADSS Server employs powerful and extensive audit and traceability capabilities. Therefore, any changes to any ADSS Server core component or service are recorded and secured. These are recorded under the **System Logs** component:

Operator: admin | Role: Administrator | Session started on: 2016-11-16 16:58:41 | Home | Help | Logout

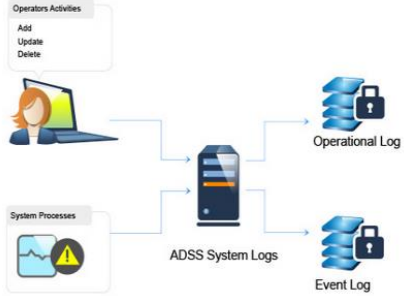
ADSS Server - Advanced Digital Signature Services

Signing Service | Verification Service | Certification Service | TSA Service | SCVP Service | LTANS Service | Go>Sign Service | RA Service

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Operational Logs

The ADSS System Log Viewer manages two types of logs, the Operation Log which contains activity of operators and the Event Log which records system initiated operations.



The System Log Viewer allows role-controlled access to:

- Search, sort and view operational logs to review the activities of all administrators or operators and see the 'before' and 'after' states
- Search, sort and view system event logs to review all system generated events such as alerts, auto-renewals, publication of CRLs, etc
- Manually copy-export records from the logs as a file
- Define an auto-archiving policy to remove records from the database
- Manually import, view or search previously archived records

© Ascertia Limited. All rights reserved.

4.4.1.1 Operational Logs

Operational logs record all changes made by ADSS Server administrators via the console to core components and services. In addition, it records access and authorisation for all ADSS Server operators. Therefore, any changes can be reviewed immediately as they are clearly highlighted:

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Operational Logs

System Logs > Operational Logs > Update Signing Profile

Operation: Update Signing Profile
Module: Signing Service
Sub Module: Signing Profiles
Performed by: CraigRoberts
Performed at: 2016-12-12 14:28:21.181

Fields	Pre State	Post State
Hashing Algorithm	SHA256	SHA256
Description	Flight Data Test profile for AFP	Flight Data Test profile for AFP
Allow copy and extraction content	false	false
Allow text access visually impaired	false	false
Allow form filling	false	false
Xml Preferences	-	-
Key Usage	false	false
Basic Constraints extension of signer certificate	false	false
Signature Policy Identifier	false	false
Signature Policy Object Id	-	-
Signature Policy URI	-	-
Signature Policy User Notice	-	-
Default Signing Area	-	-
Signing Reason	-	-
Signing Page	-	-
Signing Field	-	-
Contact Info	-	-
Visibility	-	-
Revocation Info Error	FALSE	FALSE
Default Signing Certificate	HarbourLitigationFundingTest	FlightDataTestCertificate
Embed font to be used for PDF signature	false	false
Digital Signatures-Non Repudiation	-	-

4.4.1.2 Event Logs

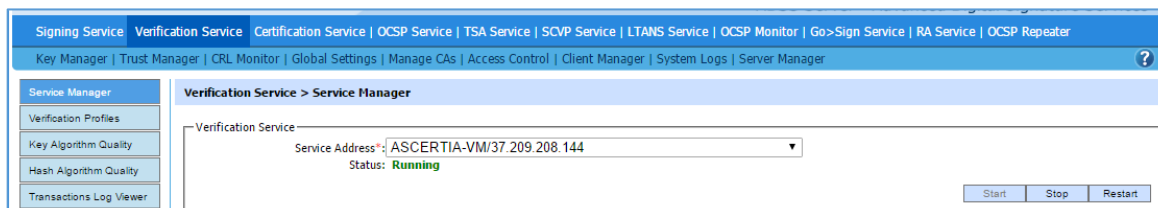
Event logs record all information related to scheduled events. For example, CRL publishing, HMAC verification, certification renewal and email notification for alerts, etc.. Note **CRL Monitor** and service related jobs are not recorded here but within their own respective transactions log viewers. The event logs can be checked as shown:



Log ID	Event	Module	Sub Module	Occurred At	Status	Detail
550000	CRL Publishing	Manage CAs	Configure Local CAs	2016-12-19 05:10:26.056	Information	View
559999	CRL Publishing	Manage CAs	Configure Local CAs	2016-12-18 05:15:23.109	Information	View
559998	Alert	OCSP Service	-	2016-12-17 23:00:13.636	Information	View
559997	Archiving Performed	OCSP Service	-	2016-12-17 23:00:13.613	Information	View
559996	CRL Publishing	Manage CAs	Configure Local CAs	2016-12-17 05:20:19.562	Information	View
559995	Alert	RA Service	-	2016-12-16 23:59:10.235	Information	View
559994	Archiving Performed	RA Service	-	2016-12-16 23:59:10.212	Information	View
559993	Alert	OCSP Service	-	2016-12-16 23:00:20.318	Information	View
559992	Archiving Performed	OCSP Service	-	2016-12-16 23:00:20.298	Information	View
559991	Alert	Certification Service	-	2016-12-16 22:45:10.73	Information	View
559990	Archiving Performed	Certification Service	-	2016-12-16 22:45:10.702	Information	View
559989	Alert	Verification Service	-	2016-12-16 22:15:13.02	Information	View
559988	Archiving Performed	Verification Service	-	2016-12-16 22:15:13.0	Information	View
559987	Alert	CRL Monitor	-	2016-12-16 22:00:11.974	Information	View
559986	Archiving Performed	CRL Monitor	-	2016-12-16 22:00:11.941	Information	View
559985	Alert	Signing Service	-	2016-12-16 22:00:10.215	Information	View
559984	Archiving Performed	Signing Service	-	2016-12-16 22:00:10.164	Information	View
559983	Alert	Go>Sign Service	-	2016-12-16 12:45:12.174	Information	View
559982	Archiving Performed	Go>Sign Service	-	2016-12-16 12:45:12.144	Information	View
559981	CRL Publishing	Manage CAs	Configure Local CAs	2016-12-16 10:58:14.579	Information	View

4.4.2 Service Manager

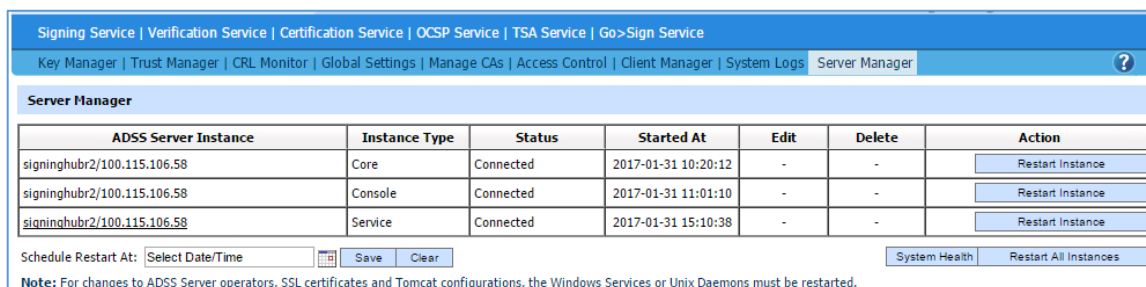
Each respective ADSS Server Service module (and **CRL Monitor**) has an associated **Service Manager sub-module**. This controls the status of the service and allows an administrator to stop, start or restart the respective service. The screen shot below shows an example of the **Verification Service** that depicts the service as running:



ADSS Server Instance	Instance Type	Status	Started At	Edit	Delete	Action
signinghubr2/100.115.106.58	Core	Connected	2017-01-31 10:20:12	-	-	Restart Instance
signinghubr2/100.115.106.58	Console	Connected	2017-01-31 11:01:10	-	-	Restart Instance
signinghubr2/100.115.106.58	Service	Connected	2017-01-31 15:10:38	-	-	Restart Instance

If the service is stopped, then it can be started by **Start** function. If there is an error reported while starting the service, then check the debug logs for that service. Access to these logs is already covered in section 3.2 ADSS Server Tomcat Instances.

The status of the Console, Core and Service components (including load-balanced instances) can be checked via **Server Manager** module:



ADSS Server Instance	Instance Type	Status	Started At	Edit	Delete	Action
signinghubr2/100.115.106.58	Core	Connected	2017-01-31 10:20:12	-	-	Restart Instance
signinghubr2/100.115.106.58	Console	Connected	2017-01-31 11:01:10	-	-	Restart Instance
signinghubr2/100.115.106.58	Service	Connected	2017-01-31 15:10:38	-	-	Restart Instance

The overall system health (including the load-balanced instances) can also be checked on the same screen by clicking on the **System Health** function. This shows the memory and database connection

usage by each of the system components. In cases where the memory or database connection usage status is high then restarting the relevant component may resolve the issue:

Signing Service Verification Service Certification Service OCSP Service TSA Service Go>Sign Service				
Key Manager Trust Manager CRL Monitor Global Settings Manage CAs Access Control Client Manager System Logs Server Manager				
Server Manager > System Health				
ADSS Server Instance	Instance Type	Status	Memory	Database Connections
signinghubr2/100.115.106.58	Core	Connected	Total Memory : 1024 MB Used Memory : 214 MB Free Memory : 810 MB Memory Status : OK	Total Connections : 100 Used Connections : 2 Free Connections : 98 Connections Status : OK
signinghubr2/100.115.106.58	Console	Connected	Total Memory : 1024 MB Used Memory : 306 MB Free Memory : 718 MB Memory Status : OK	Total Connections : 50 Used Connections : 1 Free Connections : 49 Connections Status : OK
signinghubr2/100.115.106.58	Service	Connected	Total Memory : 2048 MB Used Memory : 891 MB Free Memory : 1157 MB Memory Status : OK	Total Connections : 1000 Used Connections : 1 Free Connections : 999 Connections Status : OK

The status of all the individual services running on a Service instance can be monitored centrally via the **Server Manager** module.

If a service is not running it can be started using the **Server Manager -> Instance Type -> Service**:

Signing Service Verification Service Certification Service OCSP Service TSA Service SCVP Service LTANS Service OCSP Monitor Go>Sign Service RA Service OCSP Repeater							
Key Manager Trust Manager CRL Monitor Global Settings Manage CAs Access Control Client Manager System Logs Server Manager							
Server Manager							
Server Name : ASCERTIA-VM/37.209.208.144							
Started At : 2016-12-14 03:28:09							
Service Name	Status	Started At					
Signing	RUNNING	2016-12-14 03:28:51	Disable	Start	Stop	Restart	
Verification	RUNNING	2016-12-14 03:28:52	Disable	Start	Stop	Restart	
Certification	RUNNING	2016-12-14 03:28:50	Disable	Start	Stop	Restart	
OCSP	RUNNING	2016-12-14 03:29:01	Disable	Start	Stop	Restart	
TSA	RUNNING	2016-12-14 03:29:01	Disable	Start	Stop	Restart	
SCVP	RUNNING	2016-12-14 03:29:02	Disable	Start	Stop	Restart	
LTAN	RUNNING	2016-12-14 03:29:01	Disable	Start	Stop	Restart	
CRL Monitor	RUNNING	2016-12-14 03:28:50	Disable	Start	Stop	Restart	
Go>Sign	RUNNING	2016-12-14 03:29:03	Disable	Start	Stop	Restart	
OCSP Monitor	DISABLED	-	Enable	Start	Stop	Restart	
RA	RUNNING	2016-12-14 03:29:03	Disable	Start	Stop	Restart	
OCSP Repeater	DISABLED	-	Enable	Start	Stop	Restart	

4.5 Internal CAs

ADSS Server supports one or more internal issuing CAs. The default system CA that issues certificates that are used by the system is a local CA. This CA is created during deployment/installation.

If used, a local CA maybe configured to publish issued certificates to an LDAP directory, and CRLs to one or both of a physical file location and LDAP directory.

The publishing location can be tested via the specific CA settings using the respective 'Test' functions in the screens shown below:

Manage CAs > Configure Local CAs > ADSS Samples Test CA

CA Certificate Settings

Status:

☒ Use as default CA

CA Friendly Name*:

Description:

CA Certificate: [View Certificate](#)

Note: The CA certificate must already have been generated/imported in the ADSS Key Manager with the purpose "Cert/CRL signing".

CRL Settings

CRL Validity Period*: (min)

☒ Generate and publish CRL automatically

CRL Publishing Period*: (mins)

☐ Publish emergency CRL whenever a certificate status is changed

Hashing Algorithm:

CRL Publishing File Path: [Test](#)

e.g. /dir/sample.crl

LDAP Publishing Settings

☒ Publish CRL in LDAP

☒ Publish issued certificates in LDAP

CRL Publishing LDAP Server*:

Port*:

Bind DN/User*:

Password*: [Test](#)

[Publish CRL Now](#) [Save](#) [Cancel](#)

ADSS Server may also be configured to publish issuance and revocation information to GlobalSign when the local CA is acting under their respective root. Taken from the same location as directly above there is test function for this as well:

Manage CAs > Configure Local CAs > ADSS Samples Test CA

CA Certificate Settings

Status:

☒ Use as default CA

CA Friendly Name*:

Description:

CA Certificate: [View Certificate](#)

Note: The CA certificate must already have been generated/imported in the ADSS Key Manager with the purpose "Cert/CRL signing".

Certificate Validity Settings

If Issued Certificate Expiry is Beyond CA's Certificate Expiry:

☒ Issue the certificate

☐ Use CA's expiry date/time

☐ Return an error

Certificate Extensions

CDP Address (HTTP):

CDP Address (LDAP):

AIA Address (OCSP):

AIA Address (CA Cert):

Issuer Alternative Name OID (otherName): Value:

Certificate Issuance Reporting Settings

☒ Enable certificate issuance reporting

Account Name*:

Service URL*: [Test Connection](#)

SSL Client Certificate*: [View Certificate](#)

Local CAs can be configured to publish issued certificates to a LDAP compliant directory, and Certificate Revocation Lists to a physical path directory to allow public exposure via a hosted web site. Access to these locations should be checked.

4.5.1 Internal CA Certificate Revocation Information

If ADSS Server is running an internal CA then relying parties may require access to its revocation information i.e. CRL or OCSP, and therefore these must be available. These can be tested by using a third-party web browser and verifying access to the requested resource. The name of CRL file is as configured under the local CA settings profile:

CRL Settings

CRL Validity Period*: (min)

☒ Generate and publish CRL automatically

CRL Publishing Period*: (mins)

☐ Publish emergency CRL whenever a certificate status is changed

Hashing Algorithm:

CRL Publishing File Path: [Test](#)

e.g. /dir/sample.crl

LDAP Publishing Settings

☒ Publish CRL in LDAP

☒ Publish issued certificates in LDAP

CRL Publishing LDAP Server*:

Port*:

Bind DN/User*:

Password*: [Test](#)

[Publish CRL Now](#) [Save](#) [Cancel](#)

5 TSA Service

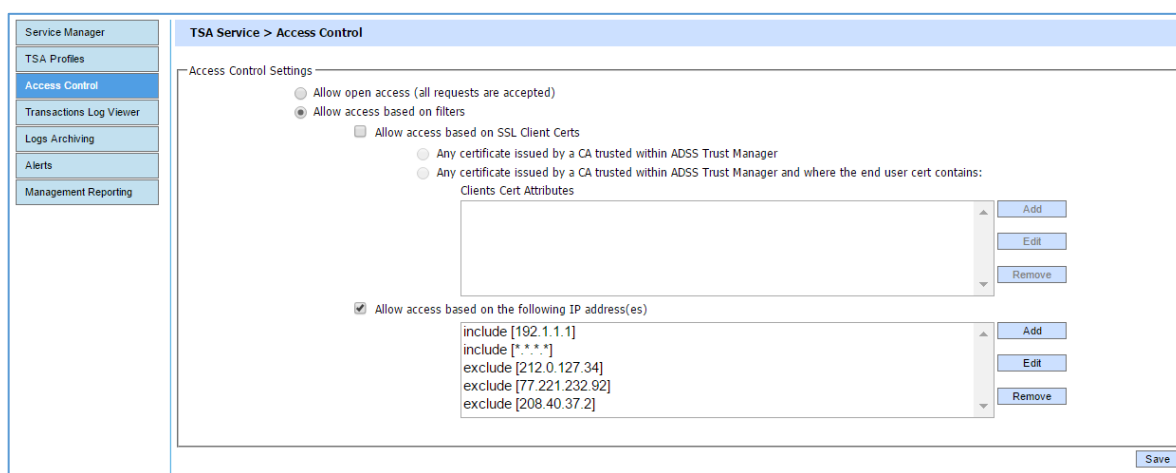
ADSS Server has an optional TSA Service that can host multiple time stamp profiles. It is very unusual for this service to fail.

5.1 Internal TSA Configuration

Check the database (see section 3.4) optional HSM (see section 3.5), the NTP Time Monitor settings (see section 3.6.4) and the Tomcat logs associated with these and the TSA service logs.

Check the TSA key is present and available in the **Key Manager** and that the associated response signing certificate has not expired.

Check the client that has issues is not barred by IP address or Authentication settings – see TSA Server Access Control:



5.2 External TSA Configuration

For completeness, a link to check external TSAs is provided in section 3.6.8.

6 OCSP Service

ADSS Server has an optional OCSP Service that can provide OCSP validation responses for one or more CAs. It is very unusual for this service to fail, but occasional issues with CRL importing have been known.

Check the database (see section 3.4) optional HSM (see section 3.5) and the logs associated with these as well as the **OCSP service** and **CRL Monitor** modules.

Check the **CRL Monitor** details to see if an important CA CRL has expired:

Operator: rod | Role: Administrator | Session started on: 2017-03-09 17:48:50 Home | Help | Logout

ascertia

ADSS Server - Advanced Digital Signature Services

Signing Service | Verification Service | Certification Service | OCSP Service | TSA Service | RA Service

Key Manager | Trust Manager | CRL Monitor | Global Settings | Manage CAs | Access Control | Client Manager | System Logs | Server Manager

Service Manager
HA Configuration
CRL Details
CRL Monitoring
Instant Revocation
CRL Logs
Logs Archiving
Alerts
Management Reporting

CRL Monitor > CRL Details

Showing page 1 of 1 Order by: CA Friendly Name Ascending

Clear Search Search View CRLs View CRL Polling Details

	CA Friendly Name	CRL Number {hex}	CRL Status	Polling Enabled	Polling Period	Next Fetch	Retain Old CRLs
<input checked="" type="radio"/>	AC Camerfirma Portugal - 2015	2	Current	True	Next Update	2017-11-02 06:47:11	False
<input type="radio"/>	Adobe Root CA	18	Current	True	Next Update	2018-07-30 19:59:59	False
<input type="radio"/>	ADSS Default Root CA	1	Current	False	-	-	False
<input type="radio"/>	ADSS Server SA11 CA3 (Configured Local CA)	d	Current	True	Next Update	2017-03-09 18:42:03	True
<input type="radio"/>	Ascertia Root CA 3	4a	Current	True	Next Update	2017-04-15 18:17:41	False
<input type="radio"/>	Baltimore CyberTrust Root	0	Expired	False	-	-	False
<input type="radio"/>	DigitalSign CA	1d0	Current	True	Next Update	2017-03-10 04:16:03	False
<input type="radio"/>	DigitalSign Primary CA	2	Current	True	Next Update	2017-11-11 04:40:42	False
<input type="radio"/>	DST ACES CA X6	53	Current	True	Next Update	2017-03-17 13:37:44	False
<input type="radio"/>	Global Chambersign Root - 2008	a	Current	True	Next Update	2017-04-28 03:25:58	False
<input type="radio"/>	GlobalSign	13	Current	True	Next Update	2017-04-14 20:00:00	False

6.1 Local CA Certificate & CRL Publishing Locations

Internal CAs will issue certificates and possibly CRLs (if configured but may use only OCSP as the only revocation status pointer). To check these publishing services, refer to **4.5.1 Internal CA Certificate Revocation Information**. This section describes how to test both options.

6.1.1 Trusted CA CRL Publishing Location

A configured trusted CA for which ADSS Server is polling the CRLs, there could be a configuration to publish the polled CRLs on a local or network file system:

Advanced CRL Handling

- ☒ CA will issue its own CRLs
 - ☐ CA has been rekeyed
- ☐ CA will use indirect CRLs
 - ☐ Use CRL in pending-update state
 - ☐ CA issues Delta CRLs
 - ☐ This CA uses PEM encoded CRLs

Note:- Encoding CRLs using PEM method is not efficient and hence not recommended. ADSS Server supports PEM encoded CRLs up to 1MB size only.
 - ☐ Check CRL issuer revocation status
 - ☐ Load CRL in memory for high speed revocation checking
 - ☐ Retain old CRLs in database
 - ☒ Publish CRL on file system

Folder Path:

CRL File Name:

☐ Keep all CRL files in the folder

The **Test Folder Path** function can be used to check the availability of the configured path.

6.1.2 Revocation Publishing Utility

If used for real time certificate status checking, the Revocation Publishing Utility (RPU) will attempt to publish the respective revocation information by inserting the CA specific revocation files in a database where ADSS Server can access it. These settings are configured for the relevant CA under **Trust Manager** module. There is no test functionality but the information can be verified:

Real-time Certificate Status Settings

- ☒ Use real-time certificate status database
 - ☐ Extended CRL status checking
 - ☒ Full certificate status checking
 - ☐ Revocation database is directly populated by the CA
 - ☒ Revocation database is populated by the Revocation Publisher Utility

Input Folder Path*:

Processed Folder Path*:

Error Folder Path*:

Temp Folder Path*:

Idle Sleep Time(Sec)*:

Batch Size*:

File Extension Filter:

File Name Filter:

Grace Period: minutes

Note: This is the time it takes for the CA to process the revocation requests. Therefore the real-time information will continue to be used for this amount of time even a fresh CRL is downloaded. If fresh CRLs are to take precedence over the real-time information then set to "0".

7 LTANS Service

The LTANS Evidence Archive service is one of the licensed options within ADSS Server. This service may not be present in your deployment.

The LTANS service allows a profile to publish archive data to a specific disk drive and location, either local host or network based, or any given URL. There are other publishing options as well but these, such as internal ADSS Server database are not external components and whose functionality is covered elsewhere within this document.

The publishing location can be tested via the specific LTANS profile using the respective 'Test' functions:

LTANS Service > LTANS Profiles > New

LTANS Profile Identification

Status*: Active

Profile ID*: adss:ltan:profile:002

Profile Name*: Test LTANS Profile

Profile Description: Test LTANS Profile

Archive Lifetime Settings

Archive Retention Period*: 1 Years 0 Months 0 Days

☐ Delete archive after validity period

Archive Evidence Settings

☐ Renew Evidence Record before TSA certificate expiry by: 0 (days)

☒ Renew Evidence Record after set period since archiving: 5 (days)

☐ Renew Evidence Record manually

Hash Algorithm: SHA256

Available TSA addresses

Assigned TSA addresses: http://localhost:8777/adss/tsa

Note: each selected TSA server will be tried in turn to obtain a timestamp. If a timestamp cannot be obtained from any of the selected TSA servers then an error will be returned.

Archive Publishing Settings

Note: You cannot change Archive Publishing Settings after the profile is created.

☐ Do not store the Archive Data

☒ Store Archive Data in internal database

☐ Store Archive Data in file system

Directory Path*:

☐ Publish to URL

Address*:

☐ Store the Client Metadata in the database

☒ Store Process-Related Metadata in database

Any errors with communication, either permissions or infrastructure, will reflect in the test results.

*** End of Document ***