ADSS Server v8.4.2 – Real-time Revocation Database – Quick Guide

ASCERTIA LTD

AUGUST 2026

Document Version- 1.0.0

© 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.

CONTENTS

1 INTRODUCTION 4
1.1 SCOPE 4
1.2 INTENDED READERSHIP 4
1.3 CONVENTIONS 4
1.4 TECHNICAL SUPPORT 4
2 CONCEPT 5
2.1 FULL CERTIFICATE STATUS CHECKING 5
2.2 EXTENDED CRL STATUS CHECKING 6
3 CONFIGURING REAL-TIME DATABASE IN ADSS SERVER 8
3.1 CONFIGURING C3P0 PROPERTIES FOR REAL-TIME REVOCATION (RTR) 10
4 FULL CERTIFICATE STATUS CHECKING 11
4.1 TRUST MANAGER CONFIGURATIONS 11
4.2 DATABASE SCHEMA 12
4.3 DATABASE SCRIPTS 14
4.4 DATA POPULATION 15
5 EXTENDED CRL STATUS CHECKING 16
5.1 TRUST MANAGER CONFIGURATIONS 16
5.2 DATABASE SCHEMA 18
5.3 DATABASE SCRIPTS 23
5.4 DATA POPULATION 26
6 UPGRADE INSTRUCTIONS 28

FIGURES

Figure 1 – Full Certificate Status Checking Concept 5
Figure 2 – Extended CRL Status Checking Concept 7
Figure 3 – Configuring Real-time Database in ADSS Server 8
Figure 4 – Trust Manager Configurations for Full Certificate Status Checking 11
Figure 5 – Trust Manager Configurations for Extended CRL Status Checking 16
Figure 6 – Database Triggers for Extended CRL Status Checking 26
Figure 7 – Revocation Publishing Utility for Extended CRL Status Checking 27

TABLES

Table 1 – Configuring Real-time Database in ADSS Server 10
Table 2 – Trust Manager Configurations for Full Certificate status Checking 11
Table 3 – Database Schema for Full Certificate Status Checking 13
Table 4 – Database Scripts for Full Certificate Status Checking 15
Table 5 – Trust Manager Configurations for Extended CRL status checking 18
Table 6 – RpInstanceQueue Table Schema for Extended CRL Status Checking 19
Table 7 – RpSettings Table Schema for Extended CRL Status Checking 20
Table 8 – RpProfile Table Schema for Extended CRL Status Checking 21
Table 9 – RevocationInfo Table Schema for Extended CRL Status Checking 22

Table 10 – RpInstanceQueue Database Scripts for Extended CRL Status Checking 23
Table 11 – RpSettings Database Scripts for Extended CRL Status Checking 24
Table 12 – RpProfile Database Scripts for Extended CRL Status Checking 25
Table 13 – Upgrade Database Scripts for ADSS Server version prior to v5.6..... 28

1 Introduction

1.1 Scope

This manual explains the use of real-time revocation information database in ADSS Server. It additionally describes the ADSS Server real-time database schema and how it can be populated with the real-time revocation information.

1.2 Intended Readership

This manual is intended for ADSS Server administrators responsible for deploying and populating the ADSS Server real-time database. It is assumed that the reader has a basic knowledge of certificates, OCSP, CRLs, revocation and IT security.

1.3 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.

1.4 Technical Support

Ascertia has a dedicated support team for your technical queries. You can reach/ contact our support team in the following ways:

Website	https://www.ascertia.com
Email	support@ascertia.com
Knowledge Base	Ascertia Community Portal

In addition to the free support services detailed 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, send the ADSS Trust Monitor logs as well. Use the Ascertia's trace log export utility to collect logs for last two days or from the date the problem arose. It will help the support team to diagnose the issue faster. Follow the instructions on how to run the trace log export utility.

2 Concept

When a certificate status is updated (revoked, suspended or un-suspended) by the CA, then typically its CRL is not published at the same time. This is because the CRL publishing is based on a fixed interval defined in the CA's CRL publishing policy. In other words, there is a time delay between a certificate being revoked and its information being made available to the relying parties (unless CRLs are issued immediately upon every revocation, which is not very common).

Similarly, in case of ADSS CRL Monitor (which is the same as any other relying party application) if a certificate is updated just after the CRL publish, its updated information will not be available in the ADSS Server database until ADSS Server successfully processes the next CRL. In the meantime, if ADSS Sever receives a Signature or OCSP validation request, it will return the status based on current valid CRL, however the actual results could be different.

To cope with this problem Ascertia has introduced the concept of a real-time certificate status database, which is synched with the CA's database upon every certificate status change, either by invoking database triggers in the CA database or by using the Ascertia Revocation Publishing Utility (RPU). The following two options will be available for real-time certificate status checking:

- Full Certificate Status Checking
- Extended CRL Status Checking

2.1 Full Certificate Status Checking

Full Certificate Status Checking means checking whether a certificate is revoked or not, plus a check to ensure that it was actually issued by the identified CA. To achieve this, ADSS OCSP Server needs to have real-time access to the full set of certificate status information maintained by the registered CAs within Trust Manager. Real-time checking of CRLs alone cannot satisfy the allowed list checking requirement because they only contain the details of revoked certificates.

To deliver real-time full certificate status checking ADSS OCSP Server needs to access a database of up-to-date certificate status information as illustrated below:

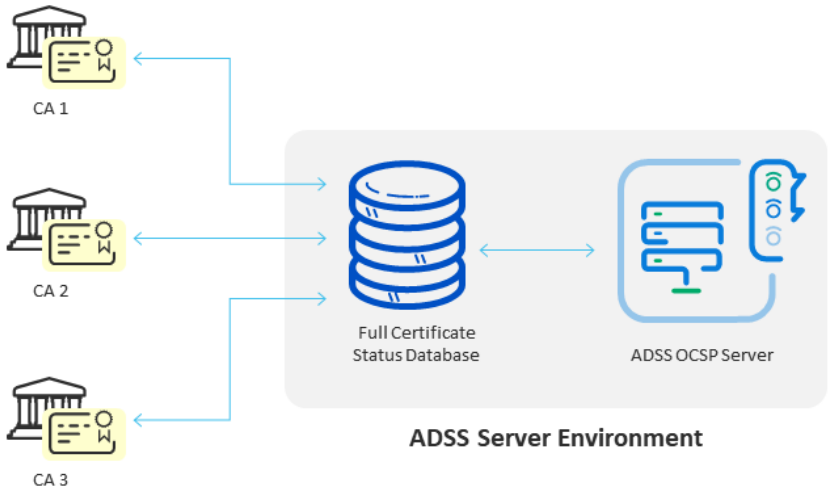

Figure 1 – Full Certificate Status Checking Concept

The Full Certificate Status Database is managed outside ADSS Server and is quite separate to the standard ADSS Server database that contains configuration data, transaction records and other information.

The Full Certificate Status Database supports multiple CAs. Each CA must be responsible for creating and managing its certificate status table, or this must be managed by a separate application written by the managed service provider. This responsibility is out-of-scope for Ascertia's products.

When ADSS Server receives an OCSP request, it determines the issuer CA identity, checks the CA validation policy and thus identifies if Full Certificate Status Checking is enabled. If this is enabled, ADSS Server checks the relevant CA's table in the Full Certificate Status Database to determine the target certificate's status. If the target certificate is found in the database then the status information is returned in the OCSP response. If the target certificate is not found in the database it is interpreted as a non-issued certificate and a **revoked** OCSP response is returned. In both cases a new "Extended Revoked Definition" extension is provided within the signed OCSP response to inform the OCSP client that the OCSP responder supports this feature. For more details, see RFC 6960 section 4.4.8.

2.2 Extended CRL Status Checking

When ADSS Server is configured to use extended CRL status checking and a validation request is sent to the ADSS Server, it checks the ADSS Server database first and then acts as follows:

1. If the requested certificate is permanently revoked (reason is not onHold) in the latest CRL in ADSS Server, result will be returned as REVOKED and ADSS real-time certificate status database will not be checked.
2. If the requested certificate is revoked with the reason onHold in the latest CRL, it will then check the ADSS real-time certificate status database to find the latest updates about the certificate.
    a. If it does not find that certificate entry in the ADSS real-time certificate status database, result will be returned as REVOKED.
    b. If it finds the certificate entry in the ADSS real-time certificate status database but with reason removeFromCRL, result will be returned as GOOD.
    c. If it finds the certificate entry in the ADSS real-time certificate status database except reason removeFromCRL and onHold, result will be returned as REVOKED.
3. If the requested certificate is not found in the latest CRL (i.e. its status is GOOD) then the following results will be returned.
    a. If it does not find that certificate entry in the ADSS real-time certificate status database, result will be returned as GOOD.
    b. If it finds the certificate entry in the ADSS real-time certificate status database with any revocation reason, result will be returned as REVOKED.

The following figure illustrates a typical ADSS Server real-time certificate status database configuration:

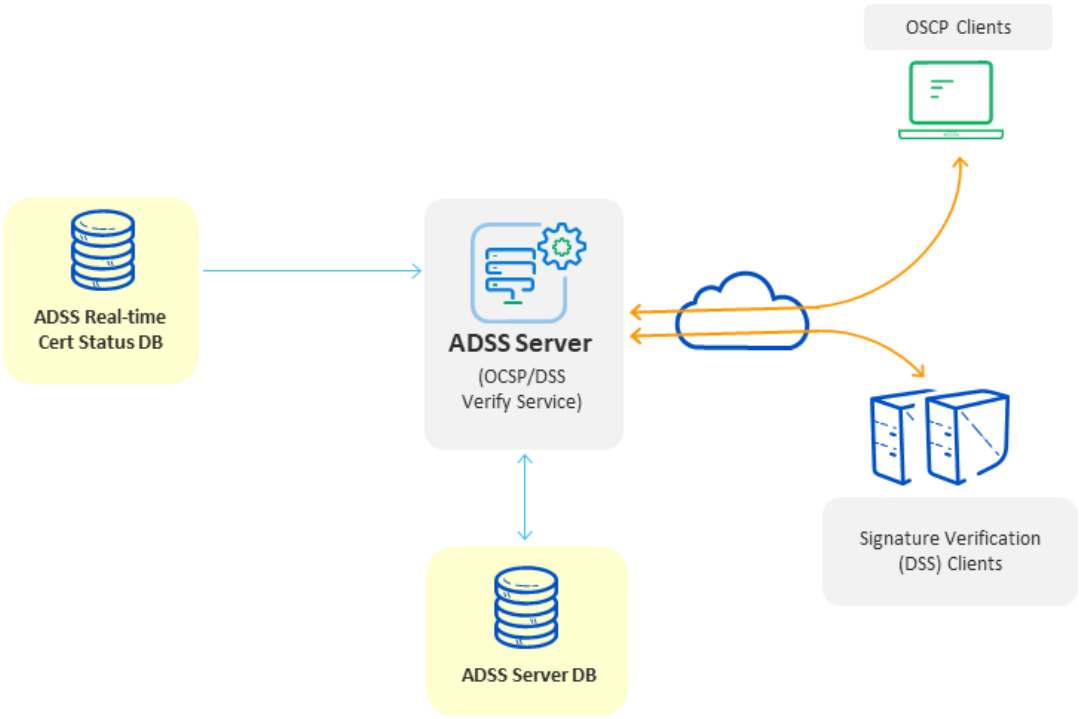

Figure 2 – Extended CRL Status Checking Concept

3 Configuring Real-time Database in ADSS Server

First of all, you must create a new real-time certificate status database, an already created real-time certificate status database can be configured for use with ADSS Server by following these instructions:

1. Launch the ADSS Server Console in a web browser
2. Navigate to Global Settings
3. Clicking on the Real-time Revocation sub module displays the following page:

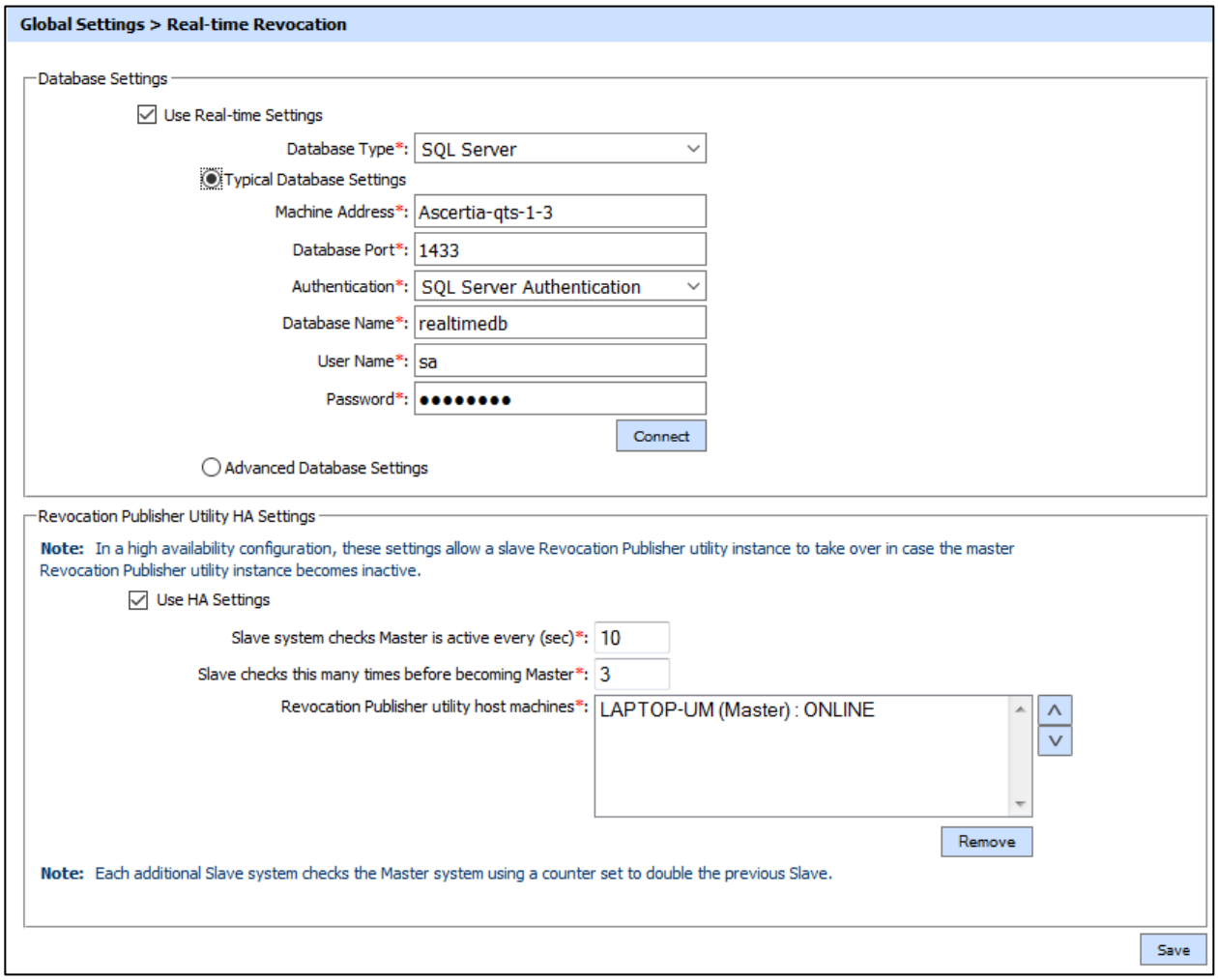

Figure 3 – Configuring Real-time Database in ADSS Server

Configuration items are as follows:

Items	Description
Use Real-time Settings	Enable this checkbox to configure the Real-time certificate status database.
Database Type	Select the type of database to which the ADSS real-time certificate status database is configured. To see the details regarding all the supported databases, refer to *ADSS Server Installation Guide – System Requirements section.*

Items	Description
	Note: The same databases used by ADSS Server apply equally to those supported for the real time certificate status databases. Click here for more details on the supported versions of each database.
Typical Database Settings	It is always suggested to use the "Typical Database Settings" and provide the credentials as described above. If it is needed to use some special parameters for the database connection string then you can opt for "Advanced Database Settings"
Machine Address	Enter the machine address (IP, Name of the machine) where the database server is installed and ADSS real-time certificate status database is created.
Database Port	Once you select the database type, this field will be populated automatically with default port number of the selected database server. If the database is not configured on the default port, then change it to the relevant port number for your database server.
Authentication	In case of ADSS Server installation with SQL Server as Database, user can be authenticated by two ways i.e. SQL Server Authentication or Windows Authentication. In case of SQL Server Authentication, user needs to enter their User Name and Password. Whereas in case of Windows Authentication, these fields will be disabled and user will be authenticated by their Windows credentials.
Database Name	Provide the name of the ADSS real-time certificate status database.
User Name	Provide the user name for the ADSS real-time certificate status database.
Password	Provide the password to connect with the ADSS real-time certificate status database.
Advanced Database Settings	The Advanced Configuration allows configuration of the low-level database drivers, URL, JARs etc.
JDBC URL	Enter the JDBC URL database connection string. This is useful for configuring a connection string manually or for database connection pooling i.e. the connection string provides details of the individual database server name, port, user ID and password running in a database pooled environment.
JDBC Driver	Shows the name of the driver used to communicate with the database.
Use HA Settings	The "HA settings" area on this page is only relevant when using the Ascertia Revocation Publishing Utility (RPU) to populate the ADSS real-time certificate status database. See the installation guide in the Ascertia RPU package for more details. **Note:** Ignore "HA settings" when populating the ADSS real-time certificate status database using database triggers.
Secondary should check Primary active status every (sec)	Defines how often a Secondary RPU will check if the Primary RPU instance is still active in seconds, the default is 10 secs.
Number of times Secondary should re-check before becoming Primary	If the Secondary finds Primary to be inactive, then this parameter defines how many times it should recheck the Primary's online status before promoting itself to become the new Primary.

Items	Description
List of Revocation Publisher utility Host Machines	This field shows the available Revocation Publisher utility Host Machines addresses.

Table 1 – Configuring Real-time Database in ADSS Server

There is no separate configuration for TLS to communicate with the real time certificate status database, i.e. it uses the same settings and configurations as the main ADSS Server database does. For detailed guidance on configuring ADSS Server with a database server running over TLS authentication, refer to the technical note titled 'How can I configure ADSS Server with database server running over SSL/TLS Authentication' in the Ascertia Community Portal knowledge base.

3.1 Configuring C3P0 Properties for Real-Time Revocation (RTR)

To customize Hibernate connection pooling settings for the Real-Time Revocation (RTR) database, you can update the c3p0.properties file located at the following path:

<ADSS_INSTALLATION_DIR>/conf/c3p0.properties

This file allows you to configure various parameters, such as the minimum and maximum number of connections that the pool should maintain, as well as the maximum idle time (in seconds) a connection can remain in the pool without being used before it is discarded.

To update the default settings:

1. Open the **c3p0.properties** file using Notepad.
2. Modify the configuration values as needed.
3. Save and close the file.
4. Restart the ADSS Server NT services to apply the changes certificates.

4 Full Certificate Status Checking

For configuring ADSS Server to use Full Certificate Status Checking, follow these steps:

4.1 Trust Manager Configurations

Once the ADSS real-time certificate status database is configured with the ADSS Server for a specific CA, the validation policy for the relevant CA should be set to use real-time revocation. To select the use of real-time revocation for a particular CA, follow these instructions:

5. Launch the ADSS Server console in a web browser
6. Navigate to **Trust Manager** module
7. Edit the relevant CA and navigate to **Validation Policy** page.
8. Select the **Primary Method** as **CRL**
9. Enable the option **Use real-time certificate status database**
10. Select the radio button **Full Certificate Status Checking**
11. Select the one of the following radio button accordingly:
    - Automatically create a table for the CA to publish certificate data
    - Use existing database table for certificate data

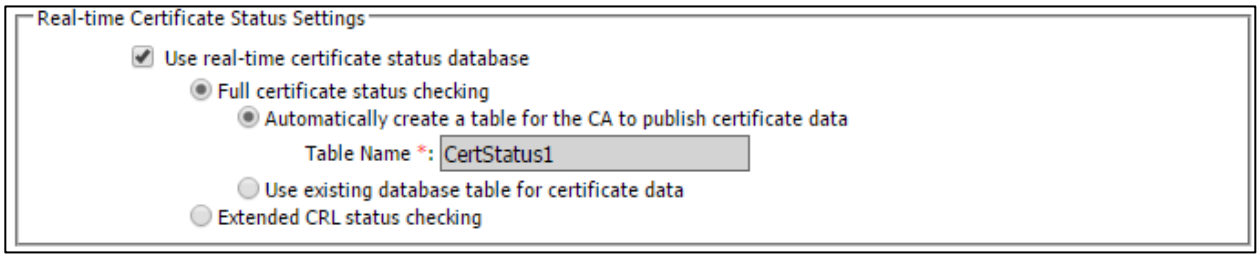

Figure 4 – Trust Manager Configurations for Full Certificate Status Checking

Configuration items for the Full Certificate Status checking are as follows:

Items	Description
Automatically create a table for the CA to publish certificate data	Select this option if the database table is to be created by ADSS Server. If the CA doesn't create this table then while configuring the Validation Policy the operator can select this settings to request ADSS Server to create the table for this CA's information.
Use existing database table for certificate data	Select this option if the database table is to be created by the external CA or your export utility application. The CA can create and populate the database table. The ADSS Server administrator can set the table name as they configure the CA's Validation Policy in Trust Manager.

Table 2 – Trust Manager Configurations for Full Certificate status Checking

4.2 Database Schema

Once the validation policy for the relevant CA has been defined to use real-time revocation then it is required to use the appropriate database schema for the table that contains real-time revocation status information. The real-time revocation database table creation depends upon the following Trust Manager settings:

- **Automatically creates a table for the CA to publish certificate data**

  If this option is configured then ADSS server automatically creates the table "CertStatus" inside the real-time database. For each CA registered in ADSS Server with real time configurations, a new instance of this table is created automatically, the table name contains a count value at the end which is assigned to the table name to keep the table name unique.

- **Use existing database table for certificate data**

  If this option is configured then the CA or a CA utility must create a table inside real-time database and populate it with certificate data. Within ADSS Server Trust Manager a suitably authorised operator can enter the name of this table when configuring the Validation Policy for this CA. The table schema must strictly comply with the table schema defined below:

Field Name	Data type (SQL)	Data type (Oracle)	Data type (Postgres)	Data type (MySQL)	Constraint	Description
SerialNo	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NOT NULL	Serial number of the certificate in hexadecimal.
Certificate	text	clob	text	longtext	NULL	The Base64 encoded certificate.
ValidFrom	datetime	timestamp	timestamp	datetime	NULL	The certificate "valid from" date/time in UTC format e.g. 2019-06-27 16:33:48.723
ValidTo	datetime	timestamp	timestamp	datetime	NULL	The certificate "valid to" date/time in UTC format e.g. 2019-06-27 16:33:48.723
Status	varchar (10)	varchar2 (10)	varchar (10)	varchar (10)	NOT NULL	The certificate status either 'REVOKED' or 'GOOD'.
RevocationDate	datetime	timestamp	timestamp	datetime	NULL	If Status is 'REVOKED' this contains the certificate revocation date/time in *UTC format e.g. 2019-06-27 16:33:48.723 If Status is 'GOOD' this is set to Null.

RevocationReason	int	int	integer	int (11)	NULL	The reason with which the certificate is revoked. This basically relates to the "Reason Code" CRL entry extension. The value must be from valid reason codes defined in RFC 5280.
HoldInstructionCode	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NULL	If the certificate is suspended (i.e. reason code is '6' or certificateHold) then this relates to "Hold Instruction Code" CRL entry extension. The value must be either from • holdInstructionNone • holdInstructionCallIssuer • holdInstructionReject
InvalidityDate	datetime	timestamp	timestamp	datetime	NULL	The date/time in UTC format at which the certificate suspension or revocation occurred e.g. 2019-06-27 16:33:48.723
CreatedAt	datetime	timestamp	timestamp	datetime	NULL	The date/time in UTC format at which this record is created or updated e.g. 2019-06-27 16:33:48.723
Hmac	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NULL	HMAC value computed on values from all columns to ensure the integrity of this record - currently not being used

Table 3 – Database Schema for Full Certificate Status Checking

4.3 Database Scripts

Follow these instructions to create a new table for Real-Time Full Certificate Status Checking according to the schema defined in previous section Database Schema:

- Launch the Database Management Client Tool e.g. SQL Server Management Studio
- Login using appropriate user credentials
- Execute the following query on your relevant database by replacing the **<TABLE_NAME>** with your desired name accordingly (Two occurrences in each script except for MySQL):

Database	Scripts

SQL/Azure Server	`Create Table <TABLE_NAME> (`
	`  SerialNo varchar (50) NOT NULL,`
	`  Certificate text NULL,`
	`  ValidFrom datetime NULL,`
	`  ValidTo datetime NULL,`
	`  Status varchar(10) NOT NULL,`
	`  RevocationDate datetime NULL,`
	`  RevocationReason int NULL,`
	`  HoldInstructionCode varchar (50) NULL,`
	`  InvalidityDate datetime NULL,`
	`  CreatedAt datetime NULL,`
	`  Hmac varchar (100) NULL,`
	`  CONSTRAINT[PK_<TABLE_NAME>] PRIMARY KEY CLUSTERED ([SerialNo])`
	`);`

Oracle	`Create Table <TABLE_NAME> (`
	`  SerialNo varchar2 (50) NOT NULL,`
	`  Certificate clob NULL,`
	`  ValidFrom timestamp NULL,`
	`  ValidTo timestamp NULL,`
	`  Status varchar(10) NOT NULL,`
	`  RevocationDate timestamp NULL,`
	`  RevocationReason int NULL,`
	`  HoldInstructionCode varchar2 (50) NULL,`
	`  InvalidityDate timestamp NULL,`
	`  CreatedAt timestamp NULL,`
	`  Hmac varchar2 (100) NULL,`
	`  CONSTRAINT PK_<TABLE_NAME> PRIMARY KEY (SerialNo)`
	`) organization index;`

Postgres SQL	`Create Table <TABLE_NAME> (`
	`  SerialNo varchar (50) NOT NULL CONSTRAINT PK_<TABLE_NAME> PRIMARY KEY,`
	`  Certificate text NULL,`
	`  ValidFrom timestamp NULL,`
	`  ValidTo timestamp NULL,`
	`  Status varchar(10) NOT NULL,`
	`  RevocationDate timestamp NULL,`
	`  RevocationReason integer NULL,`
	`  HoldInstructionCode varchar (50) NULL,`
	`  InvalidityDate timestamp NULL,`
	`  CreatedAt timestamp NULL,`
	`  Hmac varchar (100) NULL`
	`);`

MySQL	`Create Table <TABLE_NAME> (`

Database	Scripts

	`SerialNo varchar (50) NOT NULL,`
	`Certificate longtext character set utf8,`
	`ValidFrom datetime NULL,`
	`ValidTo datetime NULL,`
	`Status varchar(10) NOT NULL,`
	`RevocationDate datetime NULL,`
	`RevocationReason int(11) NULL,`
	`HoldInstructionCode varchar (50) NULL,`
	`InvalidityDate datetime NULL,`
	`CreatedAt datetime NULL,`
	`Hmac varchar (100) NULL,`
	`PRIMARY KEY (`SerialNo`)`
	`) Engine = InnoDB DEFAULT CHARSET = utf8;`

Table 4 – Database Scripts for Full Certificate Status Checking

4.4 Data Population

The external CA or CA utility application managing the Full Certificate Status Database needs to insert the data for all the issued certificates using database triggers or by some other means and make sure the data is always kept up-to-date.

5 Extended CRL Status Checking

For configuring ADSS Server to use Extended CRL Status Checking, follow these steps:

5.1 Trust Manager Configurations

Once the ADSS real-time certificate status database is configured with the ADSS Server for a specific CA, the validation policy for the relevant CA should be set to use real-time revocation. To select the use of real-time revocation for a particular CA, follow these instructions:

1. Launch the ADSS Server console in a web browser
2. Navigate to the **Trust Manager** module
3. Edit the relevant CA and navigate to the **Validation Policy** page.
4. Select **Primary Method** as **CRL**
5. Enable the **Use real-time certificate status database** option
6. Select the **Extended CRL status checking** radio button
7. Select the one of the following radio button accordingly:
   - Revocation database is directly populated by the CA
   - Revocation database is populated by the Revocation Publisher Utility

The fields starting with "" characters are only relevant when using RPU to populate the ADSS real-time certificate status database. Use any dummy values for these fields if the RPU is not being used, i.e. if you are using database triggers instead.*

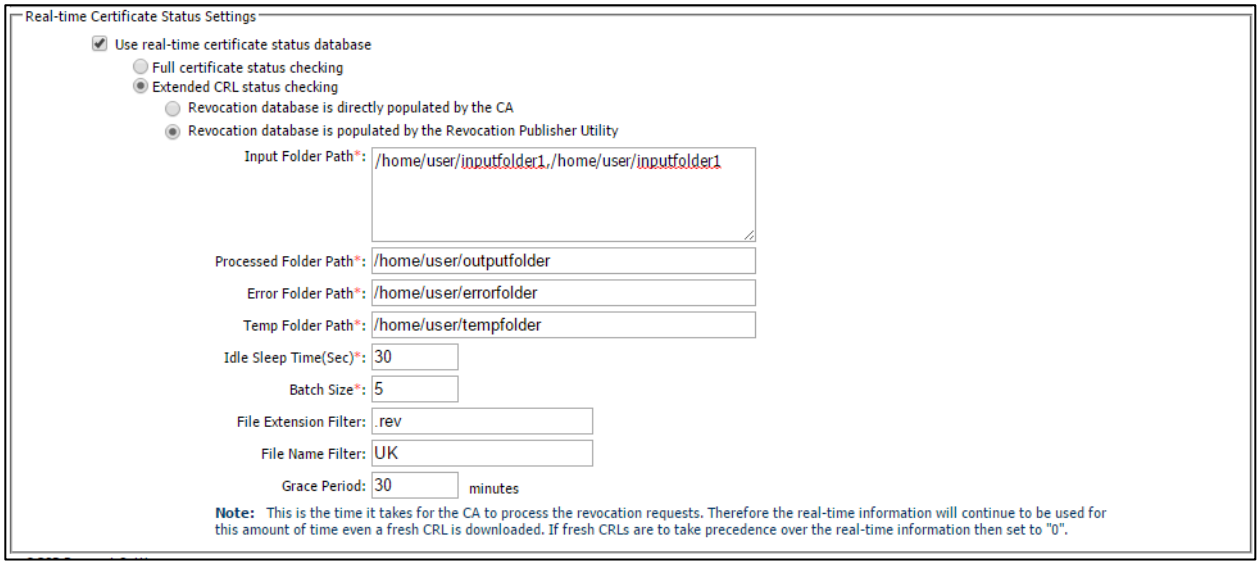

Figure 5 – Trust Manager Configurations for Extended CRL Status Checking

Configuration items for the Extended CRL Status Checking are as follows:

Items	Description
Revocation database is directly populated by the CA	Select this option if certificate status information is populated in the database table using triggers or through some other mechanism.
Revocation database is populated by the Revocation Publisher Utility	Select this option if the revocation database is populated by the Revocation Publisher Utility (RPU) in which CA publishes the revocation files against every revocation entry before publishing the next CRL. When a new CRL is published and downloaded in the ADSS Server database then real-time revocation database against that CA is cleared.
*Input Folder Path	This is the folder path from where the revocation files will be fetched from the CA machine by the Revocation Publisher Utility (RPU). Multiple paths can be configured by separating them with a comma (in case of a load-balanced CA instance with multiple CAs. The path can be either on local machine or on the network.
*Processed Folder Path	This folder is used for storing all the revocation files that are successfully processed by the Revocation Publisher Utility (RPU). As an example, the directory structure of ProcessedFolderPath for three instances of CA in load balanced mode is shown below. The processed folder contains a directory for each CA (when real-time revocation is enabled) and each directory contains sub-directory for each individual instance of this CA which may exist in high availability mode: ProcessedFolder • TestCA1 (The unique name created for the CA) 1. Instance1 2. Instance2 3. Instance3 In the above scenario there is only one CA registered in Trust Manager with real-time revocation checking enabled. If there are two CAs having real-time enabled then a new folder with TestCA2 is created which in turn contains folder for all the instances of that CA. For further details see section Real Time Revocation.
*Error Folder Path	The directory structure of ErrorFolderPath is the same as ProcessedFolderPath. This folder will contain any revocation files which the Revocation Publisher Utility failed to process.
*Temp Folder Path	To ensure a robust mechanism, the files from the input folder are copied to this temp folder for processing because the CA may also be using the original input folder and therefore this avoids any overwriting issues.
*Idle Sleep Time(Sec)	This time defines after how many seconds RPU will re-check the input folder for any new revocation profiles which may need processing.
*Batch Size	Batch size is used to indicate that how many files will be processed by RPU in one go from the input folder. Any files which are left over will be picked up by the RPU when it completes the current batch and returns back to the input folder.
File Extension Filter	This is used for setting the revocation file extension filter. If no filter is mentioned then default is used which is ".rev" filter.

Items	Description
File Name Filter	This is only applicable when revocation files with a particular name are to be filtered.
Revocation request processing grace period (minutes)	This parameter defines how long a CA should take to process the certificate revocation requests. When a new CRL is retrieved from a CA and inserted into the ADSS Server database, then this contains all the latest information, so the entries from ADSS real-time certificate status database table ca be deleted. However any revocation requests which were being processed whilst the CRL was being generated will not be included in the CRL. Hence the use of this parameter instructs the ADSS Server to only delete those records which were inserted before the newly inserted CRL thisUpdate minus N minutes. Where N is this configuration that defines how long a CA takes in minutes to actually process a revocation request so that it becomes the part of the upcoming CRL.

Table 5 – Trust Manager Configurations for Extended CRL status checking

5.2 Database Schema

Once the validation policy for the relevant CA has been defined to use real-time revocation then it is required to use the appropriate database schema for the table that contains real-time revocation status information. The real-time revocation database tables' creation depends upon the following Trust Manager settings:

- **Revocation database is directly populated by the CA**

  If this option is configured then the CA or a CA utility must create following tables inside real-time database and populate it with certificate data accordingly:

  - RpProfile
  - RevocationInfo (created automatically by the ADSS Server)

- **Revocation database is populated by the Revocation Publisher Utility**

  If this option is configured then the CA or a CA utility must create following tables inside real-time database and populate it with certificate data accordingly:

  - RpInstanceQueue
  - RpSettings
  - RpProfile
  - RevocationInfo (created automatically by the ADSS Server)

The table schema must strictly comply with the table schema defined next.

The different tables in the ADSS real-time certificate status database are explained below:

RpInstanceQueue

This table is automatically populated by RPU. This table is only relevant if RPU is being used in high availability mode. If RPU is not being used, then this table can be safely ignored.

Field Name	Data type (SQL)	Data type (Oracle)	Data type (Postgres)	Data type (MySQL)	Constraint	Description
Priority	int	int	integer	int (11)	NOT NULL	It distinguishes between the primary and secondary instances. The instance at priority 1 becomes primary automatically
MachineName	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NOT NULL	The name of the machine where RPU instance is running
Status	varchar (20)	varchar2 (20)	varchar (20)	varchar (20)	NOT NULL	Current status of RPU instance i.e. • OFFLINE • ONLINE • STOPPED
Hmac	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NULL	HMAC value computed on values from all columns to ensure the integrity of this record - currently not being used

Table 6 – RpInstanceQueue Table Schema for Extended CRL Status Checking

RpSettings

This table is automatically populated by RPU. This table is only relevant if RPU is being used in high availability mode. If RPU is not being used then this table can be safely ignored.

Field Name	Data type (SQL)	Data type (Oracle)	Data type (Postgres)	Data type (MySQL)	Constraint	Description
ParameterId	varchar (50)	varchar 2 (50)	varchar (50)	varchar (50)	NOT NULL	Id for RPU global parameter e.g. RP_PRIMARY_LIVE_TIME tells about primary instance up time
ParameterValue	varchar (50)	varchar 2 (50)	varchar (50)	varchar (50)	NULL	Context specific value of the global parameter identified by the "Id" column

Hmac	varchar (100)	varchar 2 (100)	varchar (100)	varchar (100)	NULL	HMAC value computed on values from all columns to ensure the integrity of this record - currently not being used

Table 7 – RpSettings Table Schema for Extended CRL Status Checking

RpProfile

This table is automatically populated by ADSS Server when a new CA is registered or an existing CA is updated in Trust Manager with real time configurations.

Field Name	Data type (SQL)	Data type (Oracle)	Data type (Postgres)	Data type (MySQL)	Constraint	Description
Id	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NOT NULL	CA Friendly Name defined at CA registration time in ADSS Server Trust Manager e.g. Ascertia Root CA
TableName	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NOT NULL	The name of the RevocationInfo* database table which contains real time revocation information for this CA e.g. RevocationInfo1
InputFolder Path	varchar (500)	varchar2 (500)	varchar (500)	varchar (500)	NOT NULL	Path where revocation entry files are published by the CA (only relevant if RPU is being used)
ProcessedF olderPath	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NOT NULL	Path where processed revocation entry files are placed (only relevant if RPU is being used)
ErrorFolder Path	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NOT NULL	Path where failed to process revocation entry files are placed (only relevant if RPU is being used)
TempFolder Path	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NOT NULL	Temp path utilized during revocation entry file processing (only relevant if RPU is being used)
IdleSleepTi me	int	int	integer	int (11)	NOT NULL	Time interval after which RPU looks at input folder path for new

						revocation entry files (only relevant if RPU is being used)
BatchSize	int	int	integer	int (11)	NOT NULL	The number of revocation entry files to be loaded from input folder in one iteration (only relevant if RPU is being used)
FileExtensionFilter	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NULL	The file extension filter for the revocation entry files to be searched from input folder e.g. rev (only relevant if RPU is being used)
FileNameFilter	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NULL	The file name filter for the revocation entry files to be searched from input folder e.g. RevokedCertificate (only relevant if RPU is being used)
GracePeriod	int	int	integer	int (11)	NOT NULL	The time that a CA may take to process a revocation request and to add the revocation entry in next CRL. ADSS Server takes care of this time when it deletes entries from RevocationInfo* table when a new CRL is published by the same CA.

Table 8 – RpProfile Table Schema for Extended CRL Status Checking

If RPU is not being used then the only columns which are relevant to a client application are "Id", "TableName" and "GracePeriod".

RevocationInfo

This is the most important table in ADSS real-time certificate status database schema because it contains the real time revocation information for all the certificates, which have been suspended or revoked but their revocation information is not yet published in a CRL. When ADSS Server receives an up-to-date CRL it automatically clears the matching entries from this database table.

For each CA registered in ADSS Server with real time configurations, a new instance of this table is created automatically. The table name contains a count value at the end which is assigned to the table name to keep the table name unique. The name of this table is set in column "TableName" of table "RpProfile" to associate this database table with the relevant CA.

Field Name	Data type (SQL)	Data type (Oracle)	Data type (Postgres)	Data type (MySQL)	Constraint	Description
SerialNo	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NOT NULL	Serial number of the suspended or revoked certificate
RevocationDate	datetime	timestamp	timestamp	datetime	NOT NULL	The date/time in *UTC format at which the certificate suspension or revocation occurred e.g. 2019-06-27 16:33:48.723
RevocationReason	int	int	integer	int (11)	NULL	The reason with which the certificate is revoked. This basically relates to the "Reason Code" CRL entry extension. The value must be from valid reason codes defined in RFC 5280. If no reason code provided then it is treated as "unspecified"
HoldInstructionCode	varchar (50)	varchar2 (50)	varchar (50)	varchar (50)	NULL	If the certificate is suspended (i.e. reason code is '6' or certificateHold) then this relates to "Hold Instruction Code" CRL entry extension. The value must be either from: • holdInstructionNone • holdInstructionCallIssuer • holdInstructionReject
InvalidityDate	datetime	timestamp	timestamp	datetime	NULL	The date/time in UTC format at which the certificate suspension or revocation occurred e.g. 2019-06-27 16:33:48.723
CreatedAt	datetime	timestamp	timestamp	datetime	NOT NULL	The date/time in UTC format at which this record is created or updated e.g. 2019-06-27 16:33:48.723
Hmac	varchar (100)	varchar2 (100)	varchar (100)	varchar (100)	NULL	HMAC value computed on values from all columns to ensure the integrity of this record - currently not being used

Table 9 – RevocationInfo Table Schema for Extended CRL Status Checking

5.3 Database Scripts

Follow these instructions to create a new table for Real-Time Extended CRL Status Checking according to the schema defined in previous section:

- Launch the Database Management Client Tool e.g. SQL Server Management Studio
- Login using appropriate user credentials
- Execute the following query on your relevant database:

RpInstanceQueue

Database	Scripts

SQL/Azure Server	`Create Table RpInstanceQueue (`
	`  Priority int NOT NULL,`
	`  MachineName varchar (100) NOT NULL,`
	`  Status varchar (20) NOT NULL,`
	`  Hmac varchar (100) NULL,`
	`  CONSTRAINT[PK_RpInstanceQueue] PRIMARY KEY CLUSTERED ([Priority])`
	`);`

Oracle	`Create Table RpInstanceQueue (`
	`  Priority int NOT NULL,`
	`  MachineName varchar2 (100) NOT NULL,`
	`  Status varchar2 (20) NOT NULL,`
	`  Hmac varchar2 (100) NULL,`
	`  CONSTRAINT PK_RpInstanceQueue PRIMARY KEY (Priority)`
	`) organization index;`

Postgres SQL	`Create Table RpInstanceQueue (`
	`  Priority integer NOT NULL CONSTRAINT PK_RpInstanceQueue PRIMARY KEY,`
	`  MachineName varchar (100) NOT NULL,`
	`  Status varchar (20) NOT NULL,`
	`  Hmac varchar (100) NULL`
	`);`

MySQL	`Create Table RpInstanceQueue (`
	`  Priority int(11) NOT NULL,`
	`  MachineName varchar (100) NOT NULL,`
	`  Status varchar (20) NOT NULL,`
	`  Hmac varchar (100) NULL,`
	`  PRIMARY KEY (`Priority`)`
	`) Engine = InnoDB DEFAULT CHARSET = utf8;`

Table 10 – RpInstanceQueue Database Scripts for Extended CRL Status Checking

RpSettings

Database	Scripts

SQL/Azure Server	`Create Table RpSettings (`
	`  ParameterId varchar (50) NOT NULL,`
	`  ParameterValue varchar (50) NULL,`
	`  Hmac varchar (100) NULL,`
	`  CONSTRAINT[PK_RpSettings] PRIMARY KEY CLUSTERED`
	`([ParameterId])`
	`);`

Oracle	`Create Table RpSettings (`
	`  ParameterId varchar2 (50) NOT NULL,`
	`  ParameterValue varchar2 (50) NULL,`
	`  Hmac varchar2 (100) NULL,`
	`  CONSTRAINT PK_RpSettings PRIMARY KEY (ParameterId)`
	`) organization index;`

Postgres SQL	`Create Table RpSettings (`
	`  ParameterId varchar (50) NOT NULL CONSTRAINT PK_RpSettings PRIMARY KEY,`
	`  ParameterValue varchar (50) NULL,`
	`  Hmac varchar (100) NULL`
	`);`

MySQL	`Create Table RpSettings (`
	`  ParameterId varchar (50) NOT NULL,`
	`  ParameterValue varchar (50) NULL,`
	`  Hmac varchar (100) NULL,`
	`  PRIMARY KEY (`ParameterId`)`
	`) Engine = InnoDB DEFAULT CHARSET = utf8;`

Table 11 – RpSettings Database Scripts for Extended CRL Status Checking

RpProfile

Database	Scripts

SQL/Azure Server	`Create Table RpProfile (`
	`  Id varchar (50) NOT NULL,`
	`  TableName varchar (50) NOT NULL,`
	`  InputFolderPath varchar (500) NOT NULL,`
	`  ProcessedFolderPath varchar (100) NOT NULL,`
	`  ErrorFolderPath varchar (100) NOT NULL,`
	`  TempFolderPath varchar (100) NOT NULL,`
	`  IdleSleepTime int NOT NULL,`
	`  BatchSize int NOT NULL,`
	`  FileExtensionFilter varchar (50) NULL,`
	`  FileNameFilter varchar (50) NULL,`
	`  GracePeriod int NOT NULL,`
	`  CONSTRAINT[PK_RpProfile] PRIMARY KEY CLUSTERED ([Id])`
	`);`

Oracle	`Create Table RpProfile (`
	`  Id varchar2 (50) NOT NULL,`
	`  TableName varchar2 (50) NOT NULL,`
	`  InputFolderPath varchar2 (500) NOT NULL,`
	`  ProcessedFolderPath varchar2 (100) NOT NULL,`

Database	Scripts
	ErrorFolderPath varchar2 (100) NOT NULL, TempFolderPath varchar2 (100) NOT NULL, IdleSleepTime int NOT NULL, BatchSize int NOT NULL, FileExtensionFilter varchar2 (50) NULL, FileNameFilter varchar2 (50) NULL, GracePeriod int NOT NULL, CONSTRAINT PK_RpProfile PRIMARY KEY (Id)) organization index;
Postgres SQL	Create Table RpProfile (Id varchar (50) NOT NULL CONSTRAINT PK_RpProfile PRIMARY KEY, TableName varchar (50) NOT NULL, InputFolderPath varchar (500) NOT NULL, ProcessedFolderPath varchar (100) NOT NULL, ErrorFolderPath varchar (100) NOT NULL, TempFolderPath varchar (100) NOT NULL, IdleSleepTime integer NOT NULL, BatchSize integer NOT NULL, FileExtensionFilter varchar (50) NULL, FileNameFilter varchar (50) NULL, GracePeriod integer NOT NULL);
MySQL	Create Table RpProfile (Id varchar (50) NOT NULL, TableName varchar (50) NOT NULL, InputFolderPath varchar (500) NOT NULL, ProcessedFolderPath varchar (100) NOT NULL, ErrorFolderPath varchar (100) NOT NULL, TempFolderPath varchar (100) NOT NULL, IdleSleepTime int(11) NOT NULL, BatchSize int(11) NOT NULL, FileExtensionFilter varchar (50) NULL, FileNameFilter varchar (50) NULL, GracePeriod int(11) NOT NULL, PRIMARY KEY (`Id`)) Engine = InnoDB DEFAULT CHARSET = utf8;

Table 12 – RpProfile Database Scripts for Extended CRL Status Checking

RevocationInfo

For each CA registered in ADSS Server with real time configurations, a new instance of this table is created automatically. The table name contains a count value at the end which is assigned to the table name to keep the table name unique. The name of this table is set in column "TableName" of table "RpProfile" to associate this database table with the relevant CA. For more details on database schema, click here.

5.4 Data Population

The external CA or CA utility application managing the Extended CRL Status Database needs to insert the data for all the issued certificates using one of the following means:

- Database Triggers
- Revocation Publishing Utility (RPU)

5.4.1 Database Triggers

If you wish to use the database triggers to publish the real-time revocation database then select the **Revocation database is directly populated by the CA** option in Trust Manager for the relevant CA. The below figure is an illustration of how the database triggers can be used to populate the ADSS real-time certificate status database:

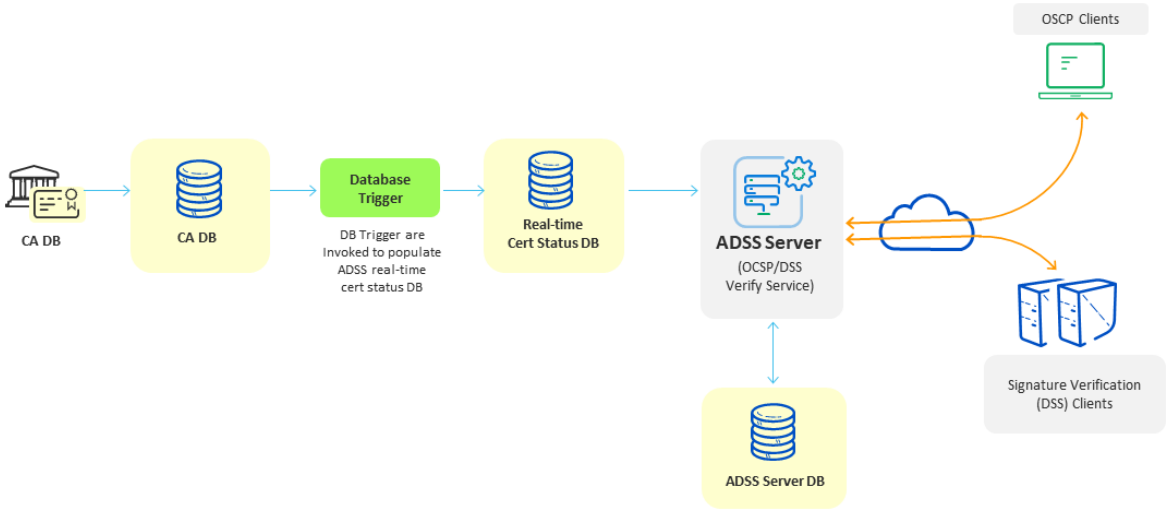

Figure 6 – Database Triggers for Extended CRL Status Checking

Once the ADSS real-time certificate status database is created and relevant configurations are made in ADSS Server along with CA registration under Trust Manager, then you are ready to write the database triggers for CA database.

Before writing triggers for CA certificate status database you should know the following:

1. The name of the RevocationInfo* table for the CA in ADSS real-time certificate status database. This can be found from the column “TableName” in the table RpProfile against the relevant CA.
2. The CA database user should have been granted with the privileges that allow CA database user to invoke the triggers which insert and update the records in a separate ADSS real-time certificate status database.
3. The following rules must be considered when writing the database triggers:

   If a certificate status is changed in the CA database since the last CRL publishing activity i.e. certificate is suspended or revoked.

   a. If serial number of certificate for which status is just changed is already listed in ADSS Server real-time certificate status database table RevocationInfo*

i. Update the revocation record in RevocationInfo* using the latest values (RevocationDate, RevocationReason, HoldInstructionCode, InvalidityDate, CreatedAt) for the already listed certificate

b. If serial number of certificate for which status just changed is not found in ADSS Server real-time certificate status database table RevocationInfo*

i. Insert a new record in RevocationInfo* using the latest values (SerialNo, RevocationDate, RevocationReason, HoldInstructionCode, InvalidityDate, CreatedAt) for the certificate

4. For details of the data types and possible values to insert/update in ADSS real-time certificate status database see the table definition for RevocationInfo* in section 4 above.

5.4.2 Revocation Publishing Utility (RPU)

The below figure is an illustration of how the Ascertia RPU utility populates the ADSS real-time certificate status database through watched folder processing of the input revocation files published by the CA. This utility has been tested with the Cybertrust UniCERT CA v5.3.2. See the installation guide within the RPU package to learn more about RPU:

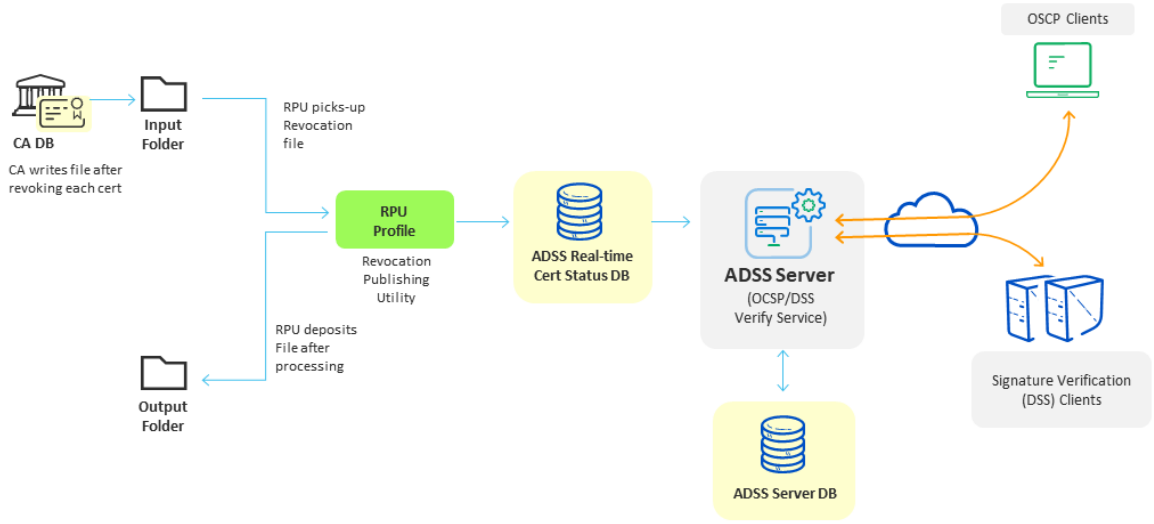

Figure 7 – Revocation Publishing Utility for Extended CRL Status Checking

6 Upgrade Instructions

If it is required to upgrade the ADSS Real-time revocation database from ADSS Server version prior to v5.6 then execute the following query on your relevant database by replacing the <TABLE_NAME> with your desired name accordingly;

- For Full Certificate Status Database execute the below script for the table **certstatusX** – X replaces with numeric
- For Extended CRL Status Database execute the below script for the table **revocationInfoX** – X replaces with numeric

Database	Scripts

SQL/Azure Server	`ALTER TABLE <TABLE_NAME> DROP CONSTRAINT PK_<TABLE_NAME>;`
	
	`ALTER TABLE <TABLE_NAME> WITH NOCHECK ADD CONSTRAINT[PK_<TABLE_NAME>] PRIMARY KEY CLUSTERED([SerialNo]) ON[PRIMARY];`

Oracle	`ALTER TABLE <TABLE_NAME> DROP CONSTRAINT PK_<TABLE_NAME>;`
	
	`ALTER TABLE <TABLE_NAME> ADD CONSTRAINT PK_<TABLE_NAME> PRIMARY KEY (SerialNo);`

Postgres SQL	`ALTER TABLE <TABLE_NAME> DROP CONSTRAINT PK_<TABLE_NAME>;`
	
	`ALTER TABLE <TABLE_NAME> ADD CONSTRAINT PK_<TABLE_NAME> PRIMARY KEY (SerialNo);`

MySQL	`ALTER TABLE <TABLE_NAME> DROP PRIMARY KEY;`
	
	`ALTER TABLE <TABLE_NAME> ADD CONSTRAINT PK_<TABLE_NAME> PRIMARY KEY (SerialNo);`

Table 13 – Upgrade Database Scripts for ADSS Server version prior to v5.6

*** End of document ***