This PDF is generated from authoritative online content, and is provided for convenience only. This PDF cannot be used for legal purposes. For authoritative understanding of what is and is not supported, always use the online content. To copy code samples, always use the online content.

eServices Deployment Guide

Advance Secure Connection Configuration for PostGreSQL

5/3/2025

Contents

Advance Secure Connection Configuration for PostGreSQL

Starting with release 8.5.300.32, UCS supports advanced secure configuration connection for PostGreSQL, including FIPS. This configuration can be achieved in any of three ways:

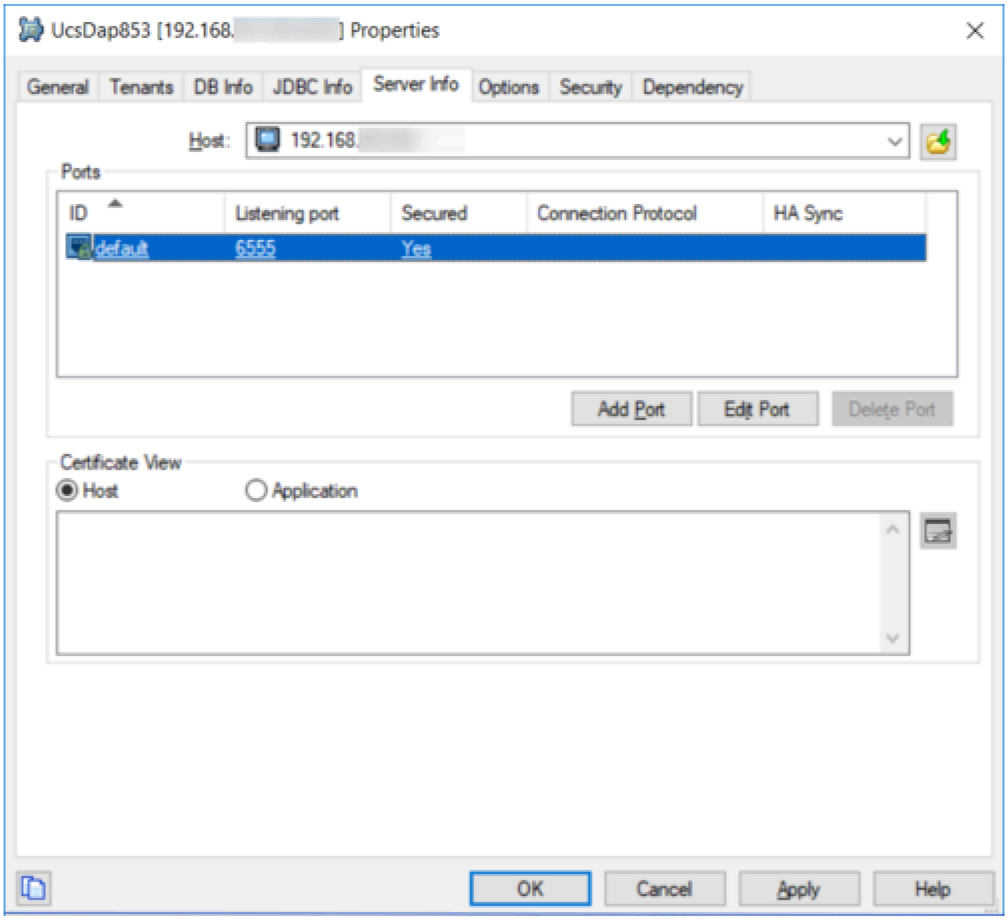
- Within the DAP application itself (two methods are available)
- On the connection link between the UCS application in Configuration Server and the Postgres DAP application
- On the host object where the Postgres database resides

Configuration approaches are described below.

Configuration within the DAP Application

Method 1

1. Enable the secure port option:



UcsDap853 [192.168.[redacted]] Properties

Host: 192.168.[redacted]

ID	Listening port	Secured	Connection Protocol	HA Sync
default	6555	Yes		

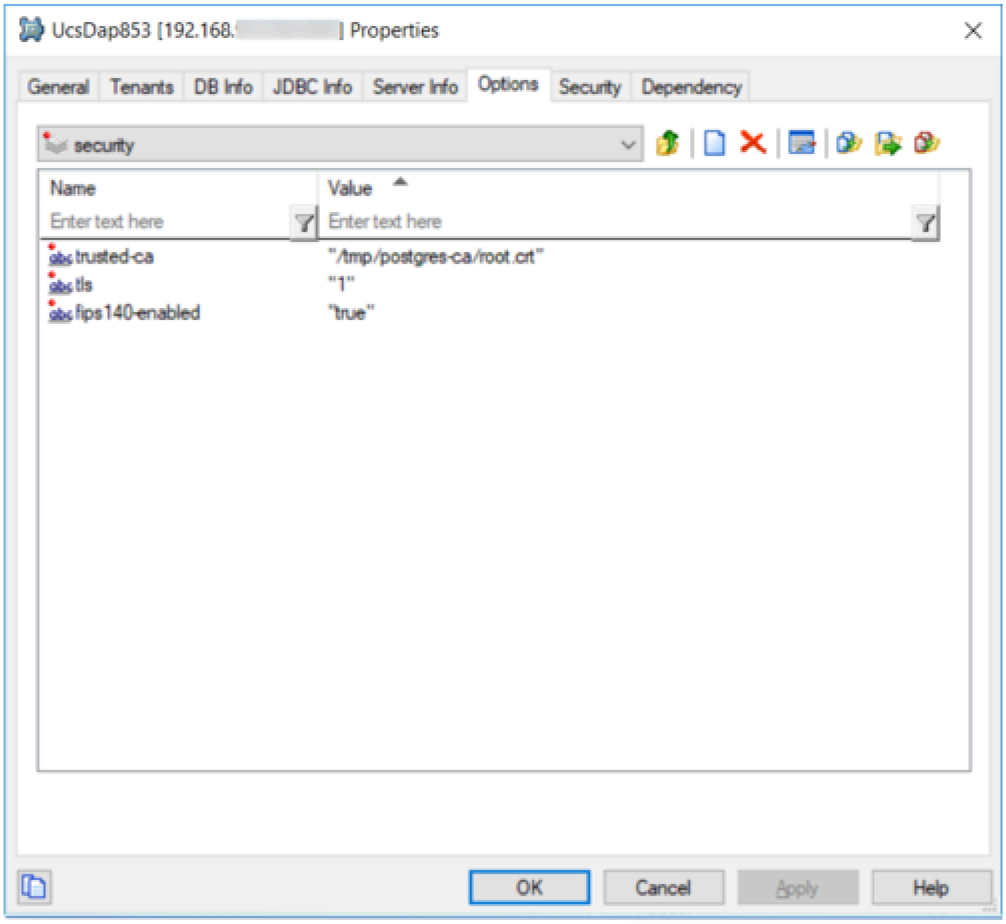
Certificate View: ● Host ○ Application

2. On the **Options** tab, create a section called **[security]**. This is because the connection is client only.
3. In the new **[security]** section:
   a. Add all TLS-related information, such as certificates information.
   b. Manually add the option **fips140-enabled** and set its value to `true` to enable FIPS mode on the database connection. Note: FIPS mode is never enabled by default and must be explicitly (manually) set.
   c. For Postgres, if the Postgres **sslmode** is set to `verify-ca` or `verify-full`, create a CA root certificate.

Example 1:

```
[security]
tls=1
fips140-enabled=true
trusted-ca=<certificate path>
```

Example 2:



UcsDap853 [192.168.[redacted]] Properties

security

Name	Value
trusted-ca	"/tmp/postgres-ca/root.crt"
tls	"1"
fips140-enabled	"true"

All parameters from Genesys TLS configuration are available but do not necessarily apply to the Postgres TLS connection. Specifically, the following options are not applicable to the Postgres database connection:

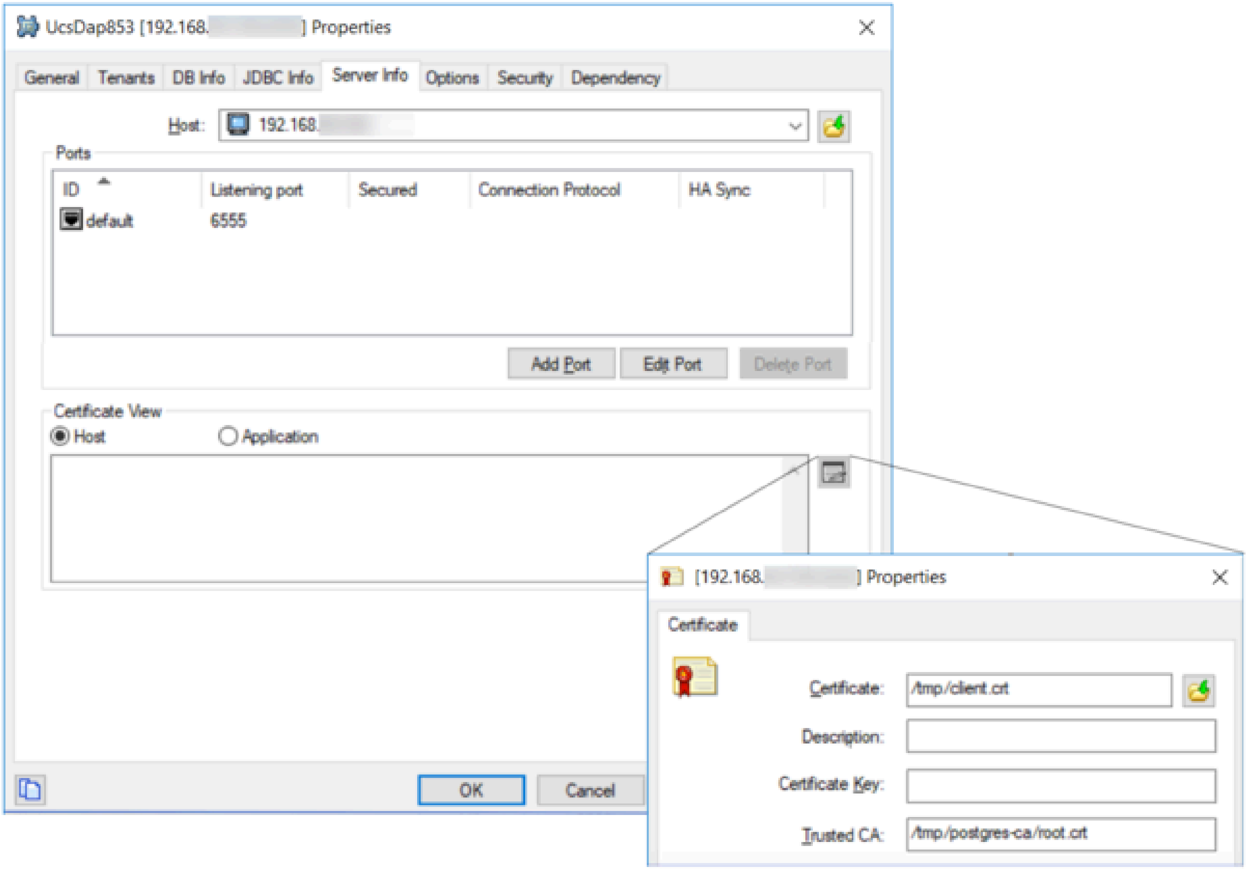
- **provider** (hard-coded to Bouncy Castle)
- **tls-mutual** (Postgres server-side configuration)
- **tls-crl** (Postgres server-side configuration)
- **cipher-list** (Postgres server-side configuration)
- **sec-protocol** (Postgres server-side configuration)
- **tls-version** (Postgres server-side configuration)
- **protocol-list** (Postgres server-side configuration)

All Postgres-specific options go in the [settings] section:

```
[settings]
 sslmode=require
```

Method 2

Alternatively, you can configure the application through the Certificate View, which synchronizes the **trusted-ca** option in the **[security]** section previously described. Note that other options still need to be manually added in the **[security]** section.



UcsDap853 [192.168.[redacted]] Properties

Host: 192.168.[redacted]

Ports

ID	Listening port	Secured	Connection Protocol	HA Sync
default	6555			

Certificate View

● Host ○ Application

[192.168.[redacted]] Properties

Certificate: /tmp/client.crt

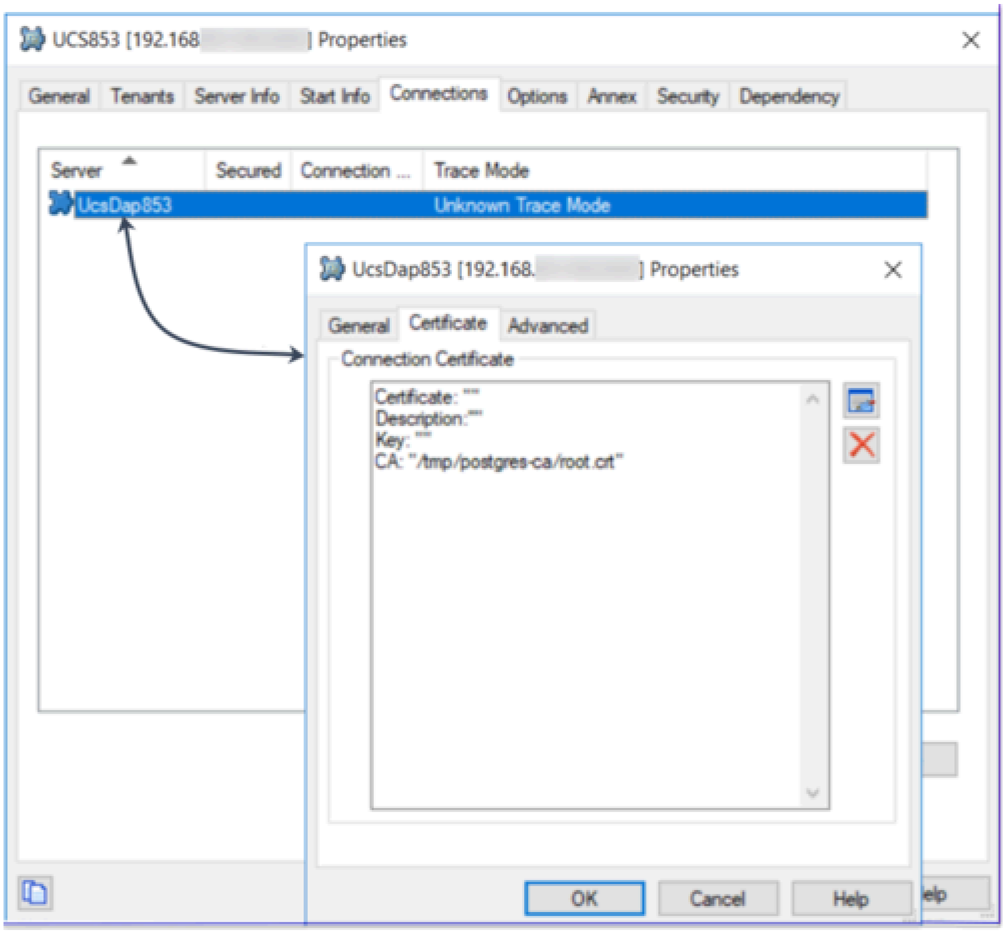
Description:

Certificate Key:

Trusted CA: /tmp/postgres-ca/root.crt

Configuration on the connection link

You can configure the connection between the UCS application and the Postgres DAP application to inject TLS configuration and its FIPS mode.



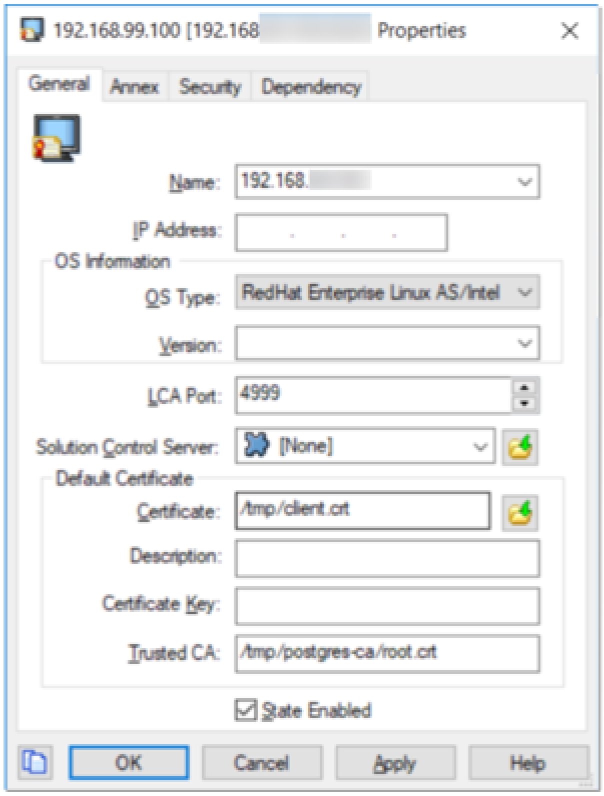
```
Certificate: ""
Description:""
Key: ""
CA: "/tmp/postgres-ca/root.crt"
```

On the **Advanced** tab:

1. Set the **fips140-enabled** option to `true` to enable FIPS mode, which must be specified manually.
2. Specify any other other required parameters.

Configuration on the Postgres host

The connection between the UCS application and the Postgres DAP application can be configured to inject TLS configuration and its FIPS mode.



192.168.99.100 [192.168[redacted] Properties

Name: 192.168.[redacted]

IP Address: . . .

OS Information

OS Type: RedHat Enterprise Linux AS/Intel

Version:

LCA Port: 4999

Solution Control Server: [None]

Default Certificate

Certificate: /tmp/client.crt

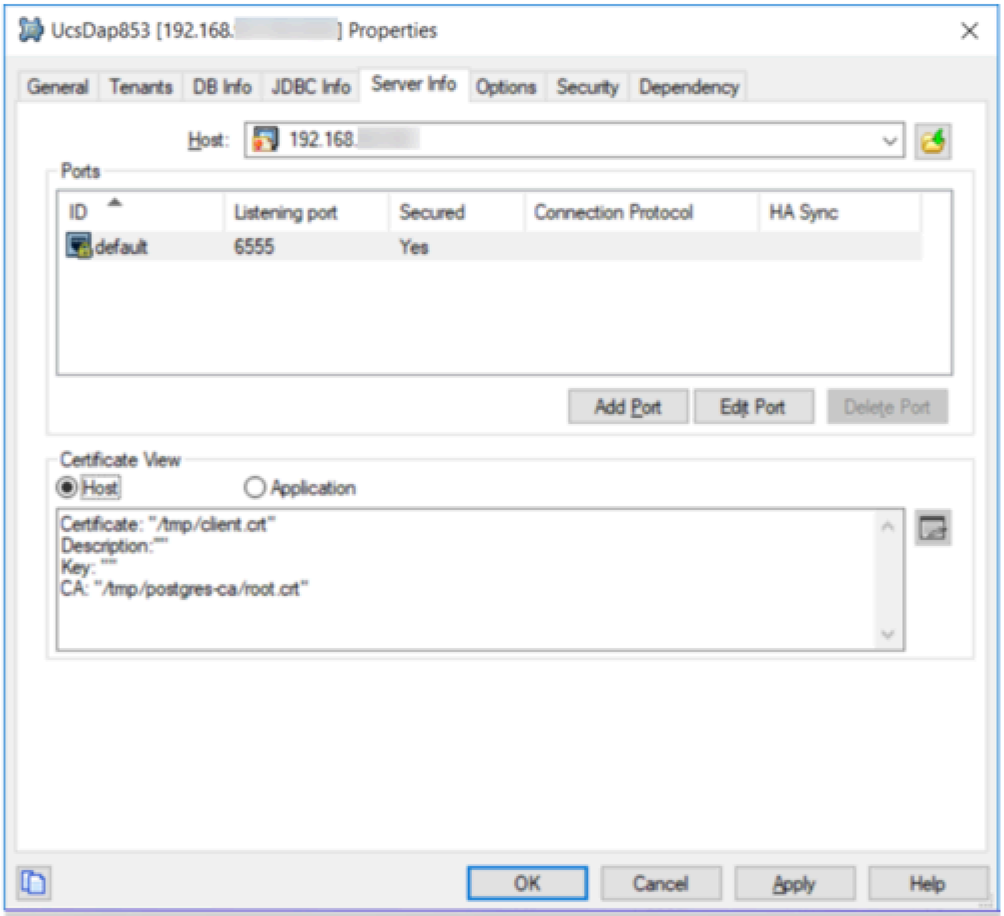
Description:

Certificate Key:

Trusted CA: /tmp/postgres-ca/root.crt

☑ State Enabled

The host information on the **Server Info** tab should look like this:


UcsDap853 [192.168.[redacted]] Properties

Host: 192.168.[redacted]

Ports

ID	Listening port	Secured	Connection Protocol	HA Sync
default	6555	Yes		

Certificate View

● Host ○ Application

Certificate: "/tmp/client.crt"
Description:""
Key: ""
CA: "/tmp/postgres-ca/root.crt"

Configuration hierarchy

UCS first looks in the UCS DAP application for TLS/FIPS configuration. If it is not found there, the connection between UCS and UCS DAP is scanned for TLS/FIPS information.

Logs

Logs contain the TLS configuration retrieved (from which object) as well as the FIPS mode set:

```
16:54:59.894 Dbg 09900 [Ucs-Main ] <[]> Registering DataAccessPoint 'UcsDap853' with role
 'Main'
 16:54:59.900 Trc 09900 [Ucs-Main ] <[]> TLS flag enabled on application UcsDap853 (155)
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Expected Hostname : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Certificate : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Certificate Key : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Cipher List : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Certificate Revocation List : null
 16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Provider : null
```

```
16:54:59.900 Dbg 09900 [Ucs-Main ] <[]> TLS Trusted CA Certificate : /tmp/postgres-ca/
root.crt
16:54:59.900 Std 21111 [Ucs-Main ] <[]> The Database Access Point 'UcsDap853' has been
configured with an infinite timeout, default 60 s will be used.
16:54:59.905 Int 09900 [Ucs-Main ] <[]> Ignoring invalid value 'null' for max-connections
```

Further reading

Please consult the following sources:

- Genesys TLS configuration - Developer's Guide
- Postgres vendor documentation