SAGE X3

Hotfix – How to manage Syracuse mutations callback URL for X3 services

For 2023R1 and 2023R2

Version control

Date	Author	Version	Comments
2024-04-03	Central CoEx	1.0	Initial version
2024-04-12	Central CoEx	1.1	Some enhancements

Table of Contents

1. Introduction

In the original design of X3-Services for Mobile Automation, the callback URL from X3-Services to Syracuse performing mutation leveraging Web Services was generated from the URL used by the mobile device to connect to Syracuse in handheld mode.

For instance, if the URL was `https://x3-mobile-automation.example.com/handheld` which may correspond to some kind of external reverse proxy / load balancer, X3-services used the same URL base `https://x3-mobile-automation.example.com` to connect to Syracuse but in a potentially different network context where:

- Ports may not be open by firewall rules:
  - This was generating **`ECONNREFUSED`** error entries in X3-services' log
- CA certificate associated with public URL being not declared in X3-services' xtrem-config.yml file
  - This was generating **`Error: self-signed certificate in certificate chain`** entries in X3-service's log.

To solve this issue, a new configuration parameter that makes the call-back URL from X3-services to Syracuse 'static' has been introduced:

- Natively in X3 2024R1 / V12.0.35 with Syracuse 12.20
- In Syracuse Hotfixes:
  - 12.18.20 (or later) for X3 2023R1 / V12.0.33
  - 12.19.14 (or later) for X3 2023R2 / V12.0.34

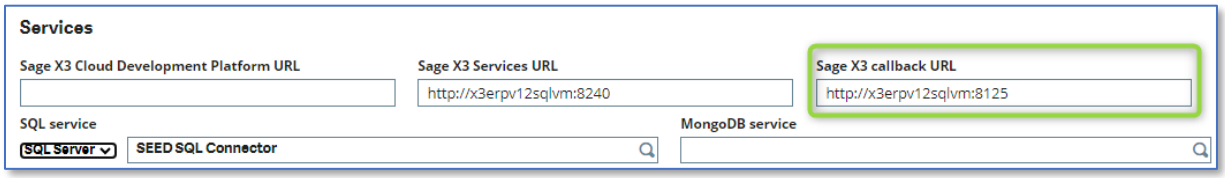
*Pay attention to the fact that if you encounter issues with X3-services – to – Syracuse connection, you need to install the relevant hotfix **AND** define the URL base that should be used between X3-Services and Syracuse as explained further.*

Just installing the hotfix without defining this parameter won't solve the issue as it will continue working the same way.

2. X3 2023R1 / V12.0.33 with Syracuse 12.18.20 hotfix

Note: this also applies to Syracuse 12.18 hotfixes later than 12.18.20.

A new parameter **Sage X3 callback URL** is present in the ***Services*** section of the X3 Solution parameters:



In this field you must enter the URL prefix (including protocol – http or https - and port in case of non-default port for protocol) that X3-services will use to build its connection strings to Syracuse web services for performing mutations.

Please understand that this has to be taken from the point of view of the host where X3-Services is installed and operating. This parameter should represent the way this host will be able to connect to Syracuse server.

In case of https use, see the relevant section at the end of this document regarding registering CA certificates in X3-services xtrem-security.yml configuration file.

3. X3 2023R2 / V12.0.34 with Syracuse 12.19.14 hotfix

Note: this also applies to Syracuse 12.19 hotfixes later than 12.19.14.

From Syracuse 12.19, the parameters for **Services** (including X3-services) can be defined at the X3 Solution level as for earlier versions but also at Endpoint (folder) level.

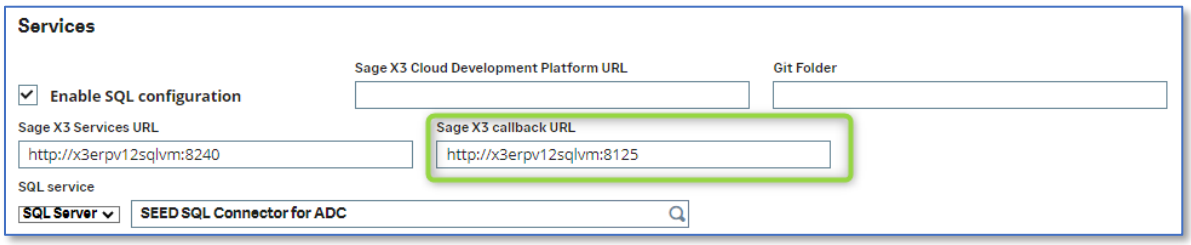
We recommend doing it at Endpoint (folder) level (and this is required if you have to configure X3-Services with multiple folders).

3.1 Parameters at Solution level

Same as for X3 2023R1 / V12.0.33 / Syracuse 12.18.20 or later, please see above.

3.2 Parameters at Endpoint level – Recommended

A new parameter called **Sage X3 callback URL** is present in the ***Services*** section of each Endpoint parameters:



In this field you must enter the URL prefix (including protocol – http or https - and port in case of non-default port for protocol) that X3-services will use to build its connection strings to Syracuse web services for performing mutations.

Please understand that this has to be taken from the point of view of the host where X3-Services is installed and operating, the parameter should represent the way this host will be able to connect to Syracuse server.

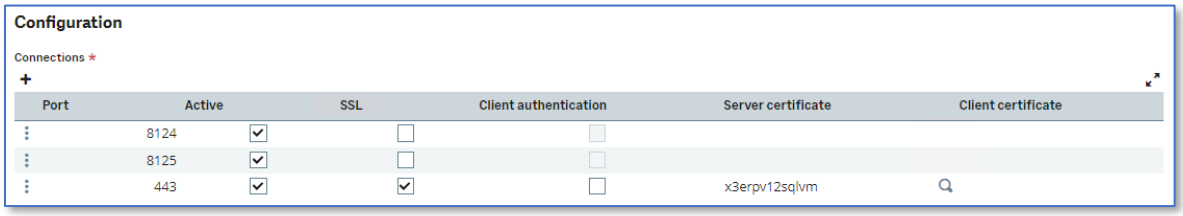
4. X3 2024R1 / V12.0.35 with Syracuse 12.20

Same as for X3 2023R2 / V12.0.34 with Syracuse 12.19.14 (and later), see above.

5. Using https for connecting X3-Services to Syracuse

In this case you will certainly need to add some extra lines in X3-Services' **`xtrem-security.yml`** file to register the CA certificate (or certificates chain) that the certificate used by Syracuse is depending on.

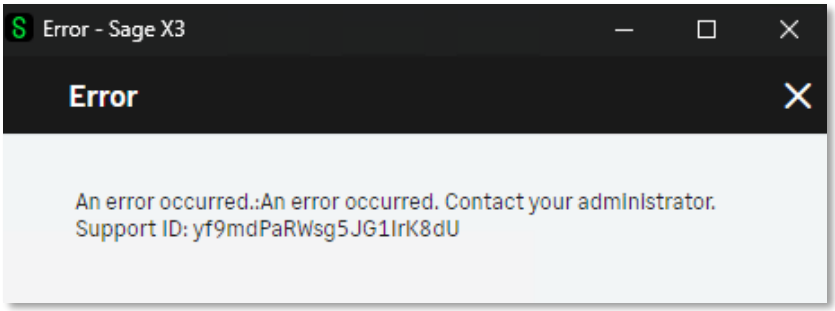
For performing this first test, Syracuse configuration is as follows:



Syracuse listens:

- For http protocol on ports:
  - 8124 (the "administration" port that should not be used to connect any user)
  - 8125 (the "unsecure users" port)
- For https protocol on port:
  - 443 (the default https port)
    Note that the server certificate presented on port 443 is the "private" one that was generated by Syracuse installer at configuration time.

If you don't register the CA certificate used to create the server certificate, you will be unable to perform mutations and you will get this kind of error in Mobile Automation:



And you will spot these error lines in the X3-Services log:

```
6388 | 000226 | 11:24:48.494 | INFO     | xtrem-x3-gateway/web-service   | Web service config loaded
from Syracuse
6388 | 000227 | 11:24:48.637 | INFO     | xtrem-x3-gateway/web-service   | SOAP Request Error: Retry
1 /3  action:  getDescription
6388 | 000228 | 11:24:49.647 | INFO     | xtrem-x3-gateway/web-service   | SOAP Request Error: Retry
2 /3  action:  getDescription
6388 | 000229 | 11:24:50.667 | INFO     | xtrem-x3-gateway/web-service   | SOAP Request Error: Retry
3 /3  action:  getDescription
6388 | 000230 | 11:24:50.670 | ERROR    | xtrem-core/graphql             | Error: self-signed
certificate in certificate chain
    at TLSSocket.onConnectSecure (node:_tls_wrap:1600:34)
    at TLSSocket.emit (node:events:517:28)
    at TLSSocket._finishInit (node:_tls_wrap:1017:8)
    at TLSWrap.onhandshakedone (node:_tls_wrap:803:12)
    at TLSWrap.callbackTrampoline (node:internal/async_hooks:130:17)
6388 | 000231 | 11:24:50.672 | ERROR    | xtrem-core/core                | self-signed certificate in
certificate chain
6388 | 000232 | 11:24:50.676 | ERROR    | xtrem-core/core                | Error: self-signed
certificate in certificate chain
    at TLSSocket.onConnectSecure (node:_tls_wrap:1600:34)
    at TLSSocket.emit (node:events:517:28)
    at TLSSocket._finishInit (node:_tls_wrap:1017:8)
    at TLSWrap.onhandshakedone (node:_tls_wrap:803:12)
    at TLSWrap.callbackTrampoline (node:internal/async_hooks:130:17)
6388 | 000233 | 11:24:50.695 | INFO     | xtrem-service/http             | HTTP response 62 200
2258ms /api
```

To avoid this, you MUST register in file **xtrem-security.yml** the CA certificate (or certificate chain) that was used to create the server certificate.

In our example case, we are using the “Private” certificate that was generated by Syracuse installer at configuration time, so we will register the path of Syracuse’s ca.cacrt as we are on the same server.

In **xtrem-security.yml** we will add a section with this generic syntax:

```
tls:
    extraCaFiles:
        - 'path_of_CA_cert#1'
        - 'path_of_CA_cert#2'
        - …
        - 'path_of_CA_cert#n'
```

In our example case this will be something like:

```
loginUrl: https://x3erpv12sqlvm

tls:
    extraCaFiles:
        - 'D:\Sage\SafeX3\SyraSrv\syracuse\certs\x3erpv12sqlvm\ca.cacrt'

# The following clientId and secret must be set with the same values in the syracuse
nodelocal.js in the section
# Both this file and nodelocal.js must be kept safe with restricted access to admin only.
# exports.config = {
#   [...]
#   etna: {
#     security: {
#       clientId: "create-your-own-client-id-uuid",
#       secret: "change-to-use-a-strong-secret-for-your-client-id"
#     }
#   }
#   [...]
# };
syracuse:
    clientId: "c9776480-c97d-11eb-b8bc-0242ac130003"
    secret: "my-more-than-20-characters-long-x3secret"
```

If Syracuse and X3-services are not hosted on the same server, you **MUST** perform a local copy of the CA certificate file used by Syracuse for the port used and register a local path.
Don't register network shared path as they cannot be resolved in most cases by X3-Services which runs as *LocalSystem*.

6. Case of public certificates or certificates generated by an external entity

If X3-services must use https to connect to Syracuse for mutations call-back, and the certificate configured on Syracuse's https port is an "external one", you must register the CA certificate (or certification chain) that was used to generate this certificate.

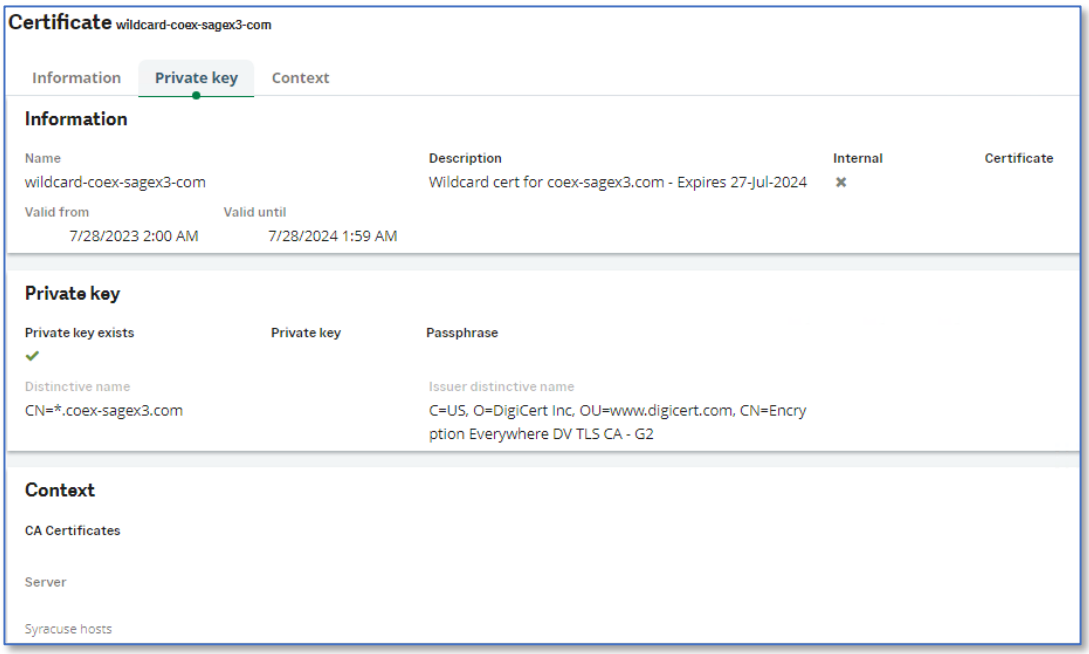
In some cases, it is difficult to have this first-hand (because usually the certificate was generated by some external entity with procedures you don't have access to).

However, you can retrieve this certificate chain by using a browser and connect to the https URL that will be used by X3-Services to perform the mutation call-backs.

Best is to perform this operation from the host where X3-services is installed, to be in the exact same network configuration that will be used by X3-services.

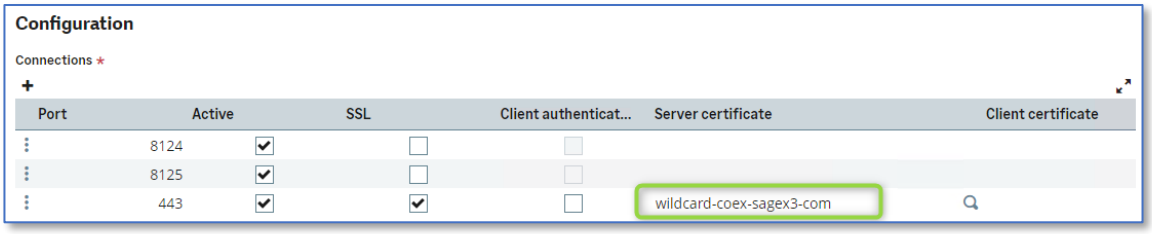
To demonstrate this, we have:

- Directly exposed Syracuse https port 443 to the Internet **(this should not be done without some reverse proxy located in a DMZ on a production environment).**
- Created a DNS entry `x3-2023r2-syra-hotfix-test.coex-sagex3.com` for that server in a test domain named `coex-sagex3.com`
- Registered a wildcard certificate for `*.coex-sagex3.com` in Syracuse:

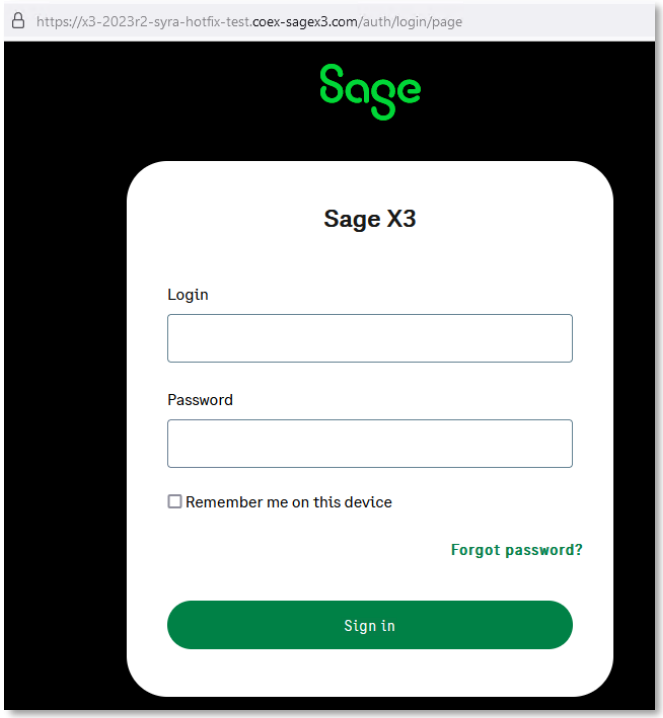


Note that there is no CA Certificate associated with this as this is something "public" issued by DigiCert / Encryption Anywhere.

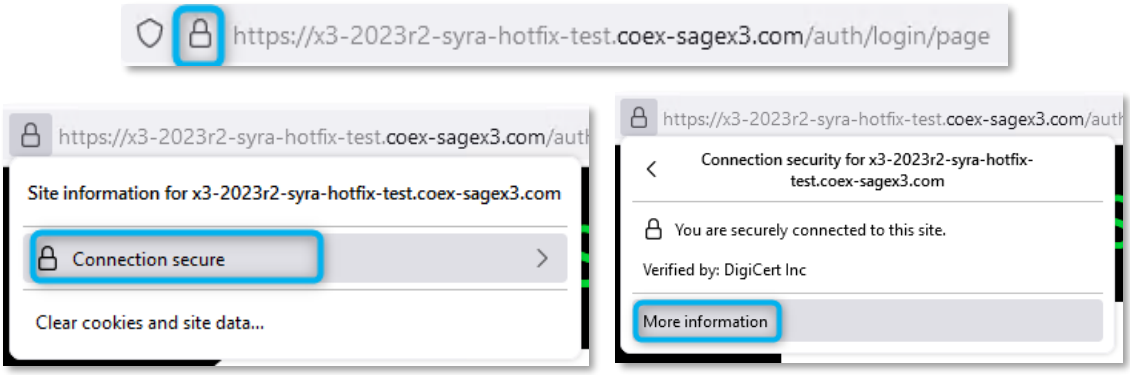
- Configured Syracuse to use this certificate as server certificate for listening on port 443:

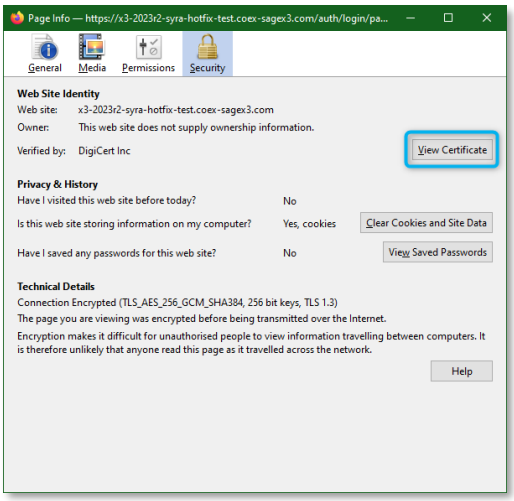


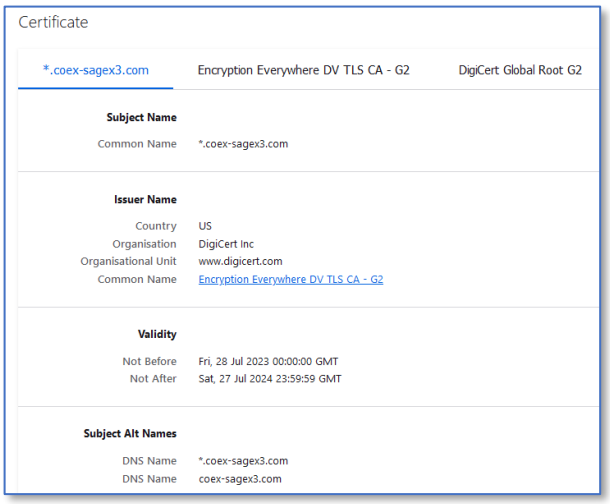
- After restarting Syracuse, when connecting a browser to **https://x3-2023r2-syra-hotfix-test.coex-sagex3.com**, connection screen is

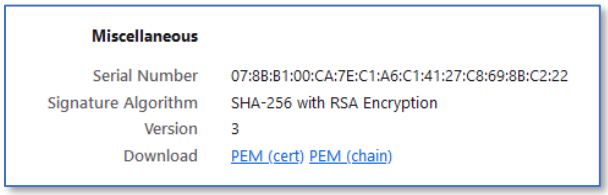


- Checking the certificate presented by the server, as seen from the browser (Firefox in this case but we'll see later with Edge):



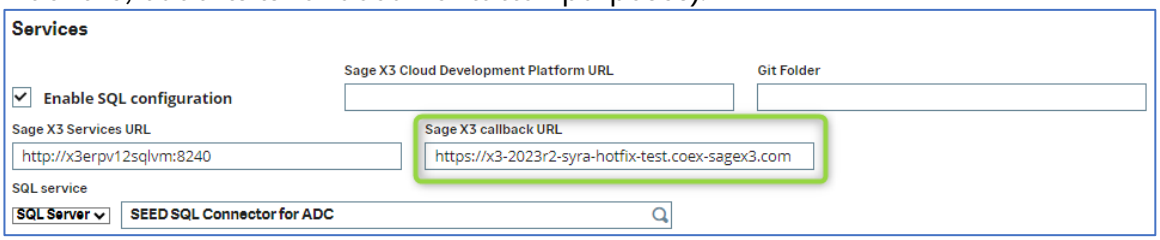






Note the fact you can download the certificate and the certificate chain from Firefox and from other browsers too.

- Configured SEED Endpoint Services section to use the public URL to connect to Syracuse (this isn't real life scenario in this case because everything is in the same machine, but this is for documentation purposes).



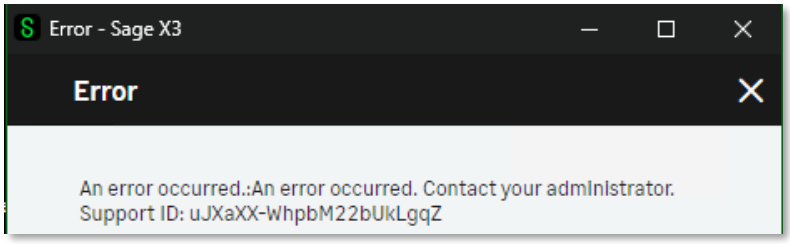
- In my **`xtrem-security.yml`** I have just changed the **`loginURL`** specification for the moment and not declared any CA.

```
loginUrl: https://x3-2023r2-syra-hotfix-test.coex-sagex3.com

# The following clientId and secret must be set with the same values in the syracuse nodelocal.js in
the section
# Both this file and nodelocal.js must be kept safe with restricted access to admin only.
# exports.config = {
#   [...]
#   etna: {
#     security: {
#       clientId: "create-your-own-client-id-uuid",
#       secret: "change-to-use-a-strong-secret-for-your-client-id"
#     }
#   }
#   [...]
# };
syracuse:
    clientId: "c9776480-c97d-11eb-b8bc-0242ac130003"
    secret: "my-more-than-20-characters-long-x3secret"
```

Now restarting “Sage X3 Services” Windows service (already done for Syracuse after changing my certificate) and testing a mutation from Mobile Automation.

I know I *may* get an error about SSL and certificates, as I’m not certain that the public CAs that were used by my Internet Provider to generate my wildcard certificate are “known” to the Node.js version used in X3-services.

Trying to do a mutation (a Stock Change), I get the same error I got earlier:



In X3-Services log, I get an error that is *not exactly* the same as the one I got earlier while using a self-generated certificate:

```
7376 | 000484 | 15:03:09.390 | INFO    | xtrem-x3-gateway/web-service   | Web service config loaded
from Syracuse
7376 | 000485 | 15:03:09.572 | INFO    | xtrem-x3-gateway/web-service   | SOAP Request Error: Retry
1 /3  action:  getDescription
7376 | 000486 | 15:03:10.581 | INFO    | xtrem-x3-gateway/web-service   | SOAP Request Error: Retry
2 /3  action:  getDescription
7376 | 000487 | 15:03:11.590 | INFO    | xtrem-x3-gateway/web-service   | SOAP Request Error: Retry
3 /3  action:  getDescription
7376 | 000488 | 15:03:11.591 | ERROR   | xtrem-core/graphql             | Error: unable to verify
the first certificate
    at TLSSocket.onConnectSecure (node:_tls_wrap:1600:34)
    at TLSSocket.emit (node:events:517:28)
    at TLSSocket._finishInit (node:_tls_wrap:1017:8)
    at TLSWrap.onhandshakedone (node:_tls_wrap:803:12)
    at TLSWrap.callbackTrampoline (node:internal/async_hooks:130:17)
7376 | 000489 | 15:03:11.592 | ERROR   | xtrem-core/core                | unable to verify the first
certificate
7376 | 000490 | 15:03:11.593 | ERROR   | xtrem-core/core                | Error: unable to verify
the first certificate
    at TLSSocket.onConnectSecure (node:_tls_wrap:1600:34)
    at TLSSocket.emit (node:events:517:28)
    at TLSSocket._finishInit (node:_tls_wrap:1017:8)
    at TLSWrap.onhandshakedone (node:_tls_wrap:803:12)
    at TLSWrap.callbackTrampoline (node:internal/async_hooks:130:17)
7376 | 000491 | 15:03:11.612 | INFO    | xtrem-service/http             | HTTP response 191 200
2329ms /api
```

Earlier error was **“self-signed certificate in certificate chain”** with a self-generated certificate, now error is **“unable to verify the first certificate”** .

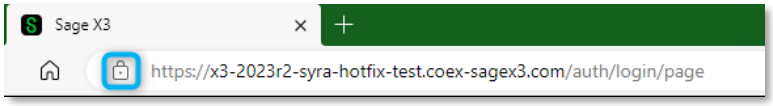
The solution is ALSO to register in X3-Services’ `xtrem-security.yml` the CA (or CA chain) that was used to generate the certificate.

First, I have to retrieve those “public” certificates.

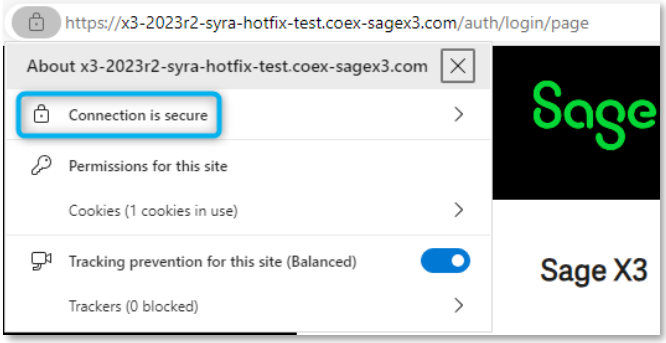
Earlier I showed you a glimpse of how to do it with Firefox (the “Download PEM (chain)” in the last Firefox screenshot above) but Firefox not being a “default” browser, I now show you how to do this with Microsoft Edge, which is standard on Windows 10, 11 and Windows Server 2022.

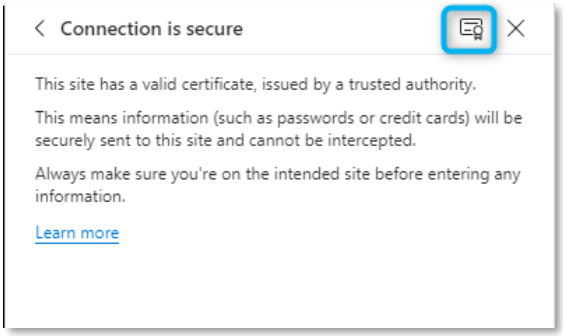
The method is very similar with Chrome.

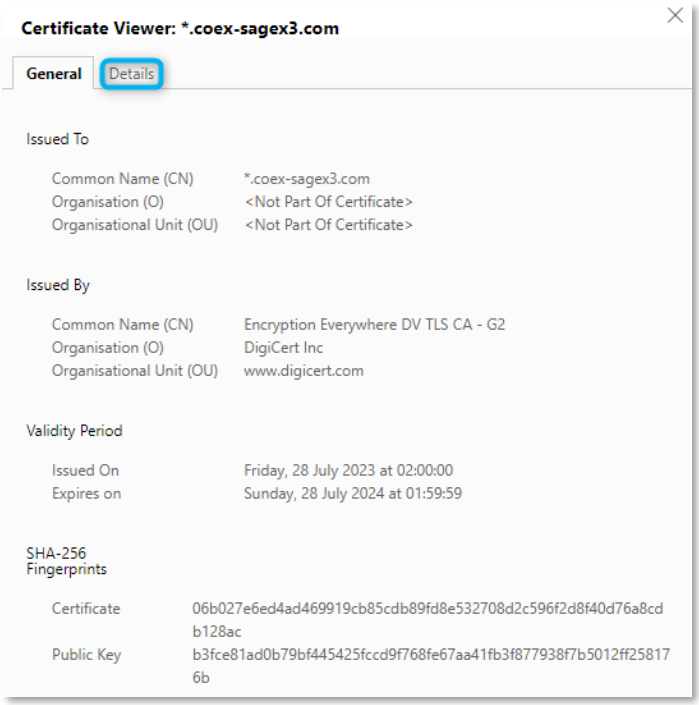
- Connect to the Syracuse https URL with Edge, better from the server where X3-Services is installed as you will be exactly in the same context as X3-Services component will run in.
  - If you can't connect you have a network / DNS / Firewall issue that has to be solved BEFORE going any further.
- When the login page is displayed, no need to log in, just click on the security icon at the left of the URL

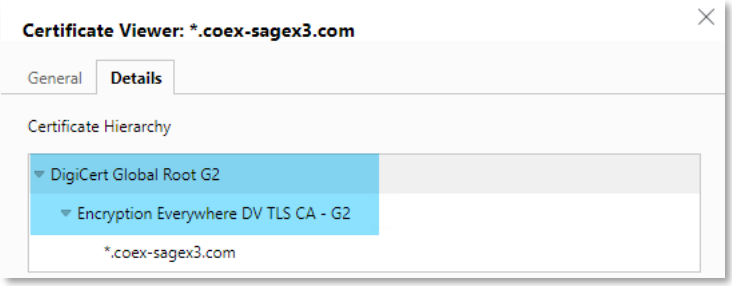


- Follow the clicks trail…

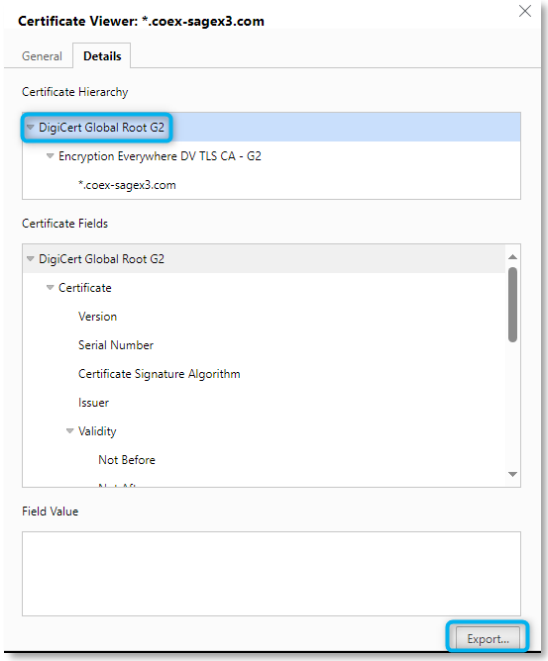


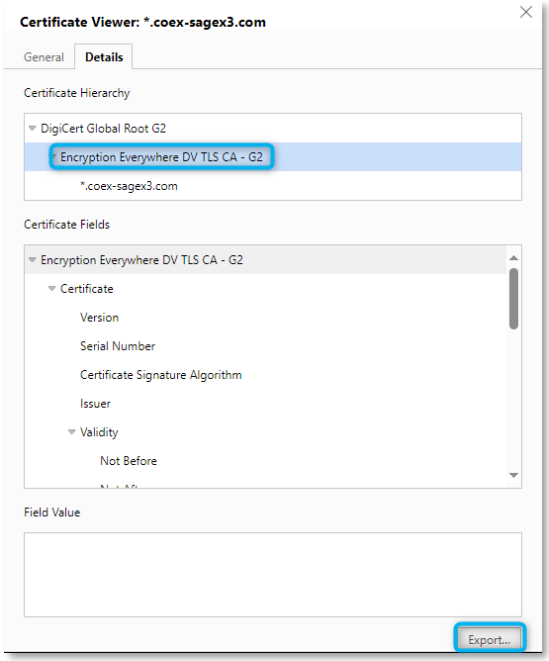






This is the interesting part, the Certificate Hierarchy: for each certificate in the hierarchy **above** the server certificate, select it and click on the [Export] button and save as a .crt file in a relevant directory. In my case, there are two levels of hierarchy:

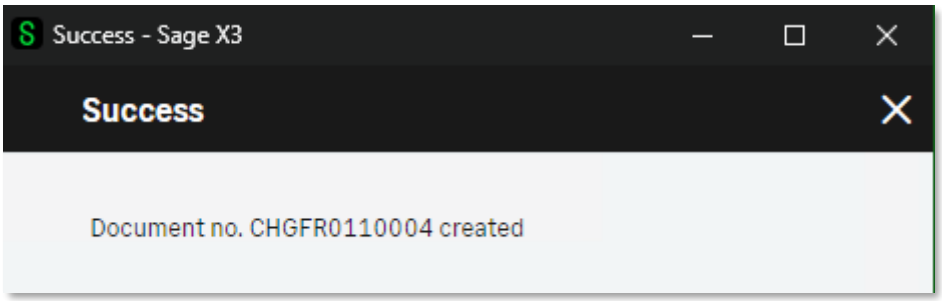




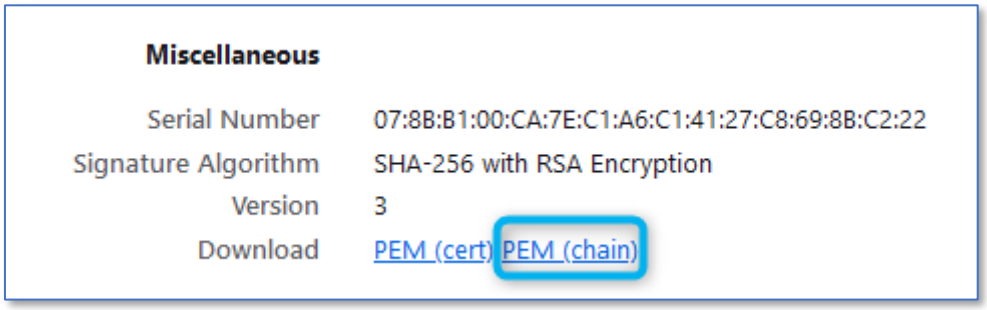
- When you have the .crt files corresponding to all levels of Certificate Hierarchy above your Syracuse connection URL certificate, you can now register them in X3-Services' **`xtrem-security.yml`** file.

```
loginUrl: https://x3-2023r2-syra-hotfix-test.coex-sagex3.com

tls:
    extraCaFiles:
        - 'D:\Sage\SafeX3\X3-Services\MyCerts\DigiCert Global Root G2.crt'
        - 'D:\Sage\SafeX3\X3-Services\MyCerts\Encryption Everywhere DV TLS CA - G2.crt'

# The following clientId and secret must be set with the same values in the syracuse nodelocal.js in
the section
# Both this file and nodelocal.js must be kept safe with restricted access to admin only.
# exports.config = {
#   [...]
#   etna: {
#     security: {
#       clientId: "create-your-own-client-id-uuid",
#       secret: "change-to-use-a-strong-secret-for-your-client-id"
#     }
#   }
#   [...]
# };
syracuse:
    clientId: "c9776480-c97d-11eb-b8bc-0242ac130003"
    secret: "my-more-than-20-characters-long-x3secret"
```

- Stop and restart "Sage X3 Services" Windows Service
- Test again something that will perform a mutation, for instance a Stock Change:



If using Firefox for retrieving your server's Certificate Hierarchy, you are able to obtain a single .pem file that is your certificate chain.



In fact, it's a single file where all individual certificates are catenated (a usual multi-certificate pem file).

In this case, you would declare something like this in **xtrem-security.yml**:

```
loginUrl: https://x3-2023r2-syra-hotfix-test.coex-sagex3.com

tls:
    extraCaFiles:
        - 'D:\Sage\SafeX3\X3-Services\MyCerts\coex-sagex3-com-chain.pem'

# The following clientId and secret must be set with the same values in the syracuse nodelocal.js in
the section
# Both this file and nodelocal.js must be kept safe with restricted access to admin only.
# exports.config = {
#   [...]
#   etna: {
#     security: {
#       clientId: "create-your-own-client-id-uuid",
#       secret: "change-to-use-a-strong-secret-for-your-client-id"
#     }
#   }
#   [...]
# };
syracuse:
    clientId: "c9776480-c97d-11eb-b8bc-0242ac130003"
    secret: "my-more-than-20-characters-long-x3secret"
```

This is exactly the same thing as registering the individual .crt certificates downloaded through Edge, because the .pem file downloaded through Firefox contains the same data plus the server certificate itself.

I could also have created a single .pem file by catenating both .crt files downloaded through Edge and register this single .pem file into `xtrem-security.yml`.