

CSTS CONFIGURATION FILES

Prepared by	Milena Bischoff
Reference	
Issue/Revision	.
Date of Issue	01.04.2019
Status	Draft

APPROVAL

Title	
Issue Number	Revision Number
Author Milena Bischoff	Date 01.04.2019
Approved By	Date of Approval

CHANGE LOG

Reason for change	Issue Nr.	Revision Number	Date

CHANGE RECORD

Issue Number	Revision Number		
Reason for change	Date	Pages	Paragraph(s)

DISTRIBUTION

Name/Organisational Unit

Table of contents:

1	INTRODUCTION.....	4
2	SHARED VALUES IN USER CONFIGURATION AND PROVIDER CONFIGURATION	5
3	USER CONFIGURATION ONLY.....	6
4	PROVIDER CONFIGURATION ONLY	7
5	ENUMERATIONS.....	8

Table of figures:

Figure 3-1: Example user config	6
Figure 4-1: Example provider config	7

Table of tables:

NO TABLE OF FIGURES ENTRIES FOUND.

1 INTRODUCTION

The CSTS framework has two configuration files - one for user side and one for provider and proxy.

The configuration files have to be passed to the framework at startup to initialise the API.

Some attributes are shared among both and some are specific to each configuration.

2 SHARED VALUES IN USER CONFIGURATION AND PROVIDER CONFIGURATION

proxy_role: the role in regards of the proxy that is taken, see Enumerations

local_id: the local id, for example the user would be 'CSTS-USER'

local_password: corresponding pw to id

startup_timer: maximum startup time as integer in seconds

non_useheartbeat: boolean to describe if heartbeat is to be used or not (proxy)

authentication_delay: maximum delay for the authentication as integer in seconds

transfer_type: the transfer type, see Enumerations

transmit_queue_size: the maximum queue size for queued pdus before overflow is triggered

list of service_types:

- list of service versions: required or offered services have to match. No check on user side if the service is in the list, but check on provider side. Check on user side need to be implemented by the application using the framework
- service version: number in integer format, at least the one being used has to be available on provider side
- service id: the OID or name of the service so it can be uniquely identified

list of remote peers:

- id: the id of the remote peer, is the connected sides local id
- authentication mode: see Enumerations
- password

port list:

- default
- not required list of port mappings:
 - responder port id
 - protocol id

3 USER CONFIGURATION ONLY

list of foreign logical ports:

- list of port data: this list will be iterated through until a connection can be made
 - hostname
 - ip address
 - port number
 - port name: name of the connection
- port heartbeat timer:
- port dead factor:
- tcp xmit buffer size: transmit buffer size
- tcp rcv buffer size: received buffer size

```
<?xml version="1.0" encoding="UTF-8"?>
- <UserConfig transfer_type="COMPLETE" xsi:noNamespaceSchemaLocation="UserConfig1.xsd" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
transmit_queue_size="10" startup_timer="180" proxy_role="INITIATOR" non_useheartbeat="true" local_password="ffdf01449809e4e5e677818892"
local_id="CSTS-USER" authentication_delay="180">
- <service_types>
- <service_type srv_id="DUMMY">
- <srv_versions>
<srv_version>1</srv_version>
<srv_version>2</srv_version>
</srv_versions>
</service_type>
- <service_type srv_id="MONITORING">
- <srv_versions>
<srv_version>1</srv_version>
</srv_versions>
</service_type>
- <service_type srv_id="[1, 3, 112, 4, 4, 1, 2]">
- <srv_versions>
<srv_version>1</srv_version>
</srv_versions>
</service_type>
</service_types>
- <remote_peers>
<remote_peer id="CSTS-PROVIDER" password="000102030405060708090a0b0c0d0e0f" authentication_mode="NONE"/>
</remote_peers>
- <foreign_logical_ports>
- <logical_port port_dead_factor="4" tcp_xmit_buffer_size="4096" port_heartbeat_timer="300" port_name="CSTS_PT1" tcp_rcv_buffer_size="4096">
<port_data ip_address="127.0.0.1" port_number="5018"/>
</logical_port>
</foreign_logical_ports>
<portlist default="ISP1"/>
</UserConfig>
```

Figure 3-1: Example user config

4 PROVIDER CONFIGURATION ONLY

cs_address

default_reporting_address: default reporting address

use_nagel: boolean if nagel should be enabled or not

min_deadfactor: integer number of min dead factor

max_deadfactor: integer number of max dead factor

min_heartbeat: integer number of min heartbeat if heartbeat enabled

max_heartbeat: integer number of max heartbeat if heartbeat enabled

list of local logical ports:

- list of port data: this list will be iterated through until a connection can be made
 - ip address
 - port number
 - port name: name of the connection
- tcp xmit buffer size: transmit buffer size
- tcp rcv buffer size: received buffer size

```
<?xml version="1.0" encoding="UTF-8"?>
- <provider_config transfer_type="COMPLETE" default_reporting_address="5510" cs_address="5500" xsi:noNamespaceSchemaLocation="ProviderConfig1.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" use_nagel="true" transmit_queue_size="10" startup_timer="180" proxy_role="RESPONDER"
  non_useheartbeat="true" min_heartbeat="20" min_deadfactor="2" max_heartbeat="600" max_deadfactor="10"
  local_password="000102030405060708090a0b0c0d0e0f" local_id="CSTS-PROVIDER" authentication_delay="180">
  - <service_types>
    - <service_type srv_id="DUMMY">
      - <srv_versions>
        <srv_version>1</srv_version>
      </srv_versions>
    </service_type>
    - <service_type srv_id="MONITORING">
      - <srv_versions>
        <srv_version>1</srv_version>
      </srv_versions>
    </service_type>
    - <service_type srv_id="[1, 3, 112, 4, 4, 1, 2]">
      - <srv_versions>
        <srv_version>1</srv_version>
      </srv_versions>
    </service_type>
  </service_types>
  - <remote_peers>
    <remote_peer id="CSTS-USER" password="ffdf01449809e4e5e677818892" authentication_mode="NONE"/>
  </remote_peers>
  - <local_logical_ports>
    - <logical_port tcp_xmit_buffer_size="4096" port_name="CSTS_PT1" tcp_rcv_buffer_size="4096">
      <port_data ip_address="*" port_number="5018"/>
    </logical_port>
  </local_logical_ports>
  <portlist default="ISP1"/>
</provider_config>
```

Figure 4-1: Example provider config

5 ENUMERATIONS

proxyRoleEnum:

INITIATOR – initiates the connection, hence is the user.

RESPONDER – responds to the connection, hence is the provider

authenticationMode:

NONE – authentication is not used

BIND_ONLY – authentication is only used for the bind

ALL – authentication is used for everything

transferType:

TIMELY – pdus that cannot be send will be discarded

COMPLETE – pdus will be queued if they cannot be send and then resend