
TERRA AC WALLBOX

OCPP 1.6

Implementation Overview

Contents

1. Revisions	3
2. Overview	4
3. OCPP server configuration	4
3.1. ABB ChargerSync server.....	4
3.2. OCPP external server configuration.....	5
3.3. WebSocket communication.....	5
3.3.1. Client request.....	5
3.3.2. Server response.....	6
3.3.3. WebSocket ping in relation to OCPP Heartbeat.....	6
4. Supported Feature profiles	6
4.1. Messages.....	7
4.2. Standard Configuration keys.....	10
4.2.1. Supported keys.....	10
4.2.2. Not supported keys.....	14
4.3. Custom Configuration keys.....	16
4.4. OCPP ABB error code definition.....	20
4.5. Security.....	20
4.5.1. Encryption.....	20
4.5.2. Authentication.....	20
4.5.3. Security event and security logs.....	21
4.5.4. Certificate management.....	21
4.6. Certification maturity.....	21
5. Additional Information	21
5.1. Listing of related documents.....	21
6. Addendum	22
6.1. Firmware upgrade from external OCPP server.....	22
6.2. Regional specific interface support.....	24
6.2.1. UK - The Electric Vehicles (Smart Charge Points) Regulations 2021.....	24
6.2.2. United States (US) specific support.....	25
6.2.3. Japanese specific support.....	27
6.3. Security digital certificate.....	27

1. Revisions

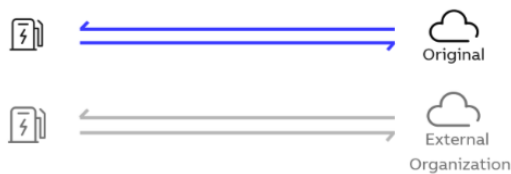
Rev.	Page (P) Chapt. (C)	Description	Date Dept./Init.
1.6.1		No technical changes from 1.6 document version(FW1.5.2). Only document restructuring. And section 5 is added for informative.	03/17 PM VJ
1.7		Firmware 1.6.3 updates. Refer release note for more details. And added a new section to capture regional specific details.	July 2022 PM/VJ
1.8		Firmware 1.6.6 updates. FreeVendEnabled, FreeVendIdTag change to FreevendEnabled, FreevendIdTag Support SecurityEventNotification, GetLog, LogStatusNotification	May 2023 PM JV
1.9		Firmware 1.8 updates. FTP support for firmware upgrade and GetDiagnostics. ChangeNFCPower support.	Feb 2024 PM JV
2.0		Firmware 1.8.2 updates. Solar support, certificate management and addition of various optional configuration keys	Jun 2024 PM JV
2.1		Firmware 1.8.32 updates. OCPP customized keys:UserSetMaxCurrent, ChargingLimitedBy,ChargingLimitedChannels, MeterValueReportOnSuspendedEV, StopChargingOnlyWithSameCard, ExternalMeterSampleInterval, AuthReqInOfflineMode, ABBIdTag etc.	Mar 2025 FW JV
2.2		Firmware 1.8.34 updates. ConnectorPhaseRotation support. OCPP customized keys: ChargeRecordUseTotalEnergy, PWMAfterStartTransaction.	Sep 2025 FW JV

2. Overview

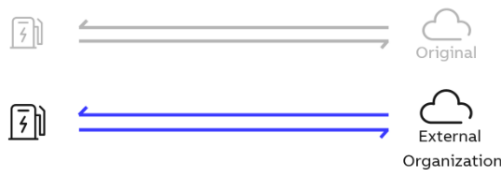
ABB Terra AC chargers support OCPP 1.6 J. This document describes OCPP 1.6 functionality supported by ABB Terra AC chargers (from firmware version 1.8.32 onwards) according to OCPP protocol specification.

ABB has implemented OCPP 1.6-J version, which means using JSON over WebSocket. The charger is connected to the ABB ChargerSync server by default. The charger is either connected to a 3rd party/external OCPP Server or to ChargerSync server. The connection to the ChargerSync portal allows for efficient remote support and enables additional features next to OCPP. This concept is referred to as below.

Link to the ABB ChargerSync server:



Link to a 3rd party OCPP backend server:



3. OCPP server configuration

Contact your local ABB sales representative to arrange access to TerraConfig App account and facilitate company creation and new OCPP configuration via the ChargerSync Portal.

For more information refer section 5.

3.1. ABB ChargerSync server

By default, charger is connected to ABB ChargerSync server. And this ChargerSync server can be accessed by ChargerSync mobile application as well as web portal.



3.2. OCPP external server configuration



Once you have access for TerraConfig App, the below steps need to be performed for each charger commissioning that needs connection to a 3rd party/external OCPP backend.

1. Download TerraConfig app and use the credentials sent in the email, generated by the creation of the account.
2. Pair the TerraConfig app with the charger and check firmware version. It's always recommended that charger has latest firmware version.
3. Make sure the charger is connected to the internet (over Wi-Fi, LAN or 4G).
4. Enable external OCPP server and choose the correct backend URL which has been preconfigured in the portal. Then press configure and afterwards OK.
5. Check via Device info that the URL is the correct one and that connection has been established. Via OCPP logs (of the 3rd party backend) validate that BootNotification is successfully sent.
6. Run some remote commands to confirm good communication between charger and backend.

3.3. WebSocket communication

For the connection between a Charge Point(charger) and a Central System (OCPP Server) using OCPP-J, the Central System acts as a WebSocket server and the Charge Point acts as a WebSocket client.

3.3.1. Client request

The following is an example of an opening HTTP request of an OCPP-J connection handshake:

GET /webServices/ocpp/CP3211 HTTP/1.1

Host: some.server.com:33033

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Key: x3JJHMbDL1EzLkh9GBhXDw==

Authorization: Basic <Base64 encoded(<ChargePointId>:<AuthorizationKey*>)>

Sec-WebSocket-Protocol: ocpp1.6, ocpp1.5

Sec-WebSocket-Version: 13

The bold parts are found as such in every WebSocket handshake request, the other parts are specific to this example. In this example, the Central System's OCPP-J endpoint URL is "ws://some.server.com:33033/webServices/ocpp". The Charge Point's unique identifier is "CP3211", so the path to request becomes "webServices/ocpp/CP3211".

Remark:

*The *AuthorizationKey* (i.e.) password was configured in TerraConfig portal and sent to charger via Bluetooth using TerraConfig App.

3.3.2. Server response

Upon receiving the Charge Point's request, the Central System has to finish the handshake with a response as described in RFC6455.

So, if the Central System accepts the above example request and agrees to using OCPP 1.6J with the Charge Point, the Central System's response will look as follows:

HTTP/1.1 101 Switching Protocols

Upgrade: websocket

Connection: Upgrade

Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

Sec-WebSocket-Protocol: ocpp1.6

The bold parts are found as such in every WebSocket handshake response, the other parts are specific to this example.

3.3.3. WebSocket ping in relation to OCPP Heartbeat

The WebSocket specification defines Ping and Pong frames that are used to check if the remote endpoint is still responsive. In practice this mechanism is also used to prevent the network operator from quietly closing the underlying network connection after a certain period of inactivity. This WebSocket feature can be used as a substitute for most of the OCPP Heartbeat messages but cannot replace all of its functionality.

Remark:

Charger supports sending PING every certain time interval. The interval range is 0, 10 to 65535 seconds, 0 disables the PING, default value is 60 seconds.

Charger will also respond with PONG to the PING request from the server.

4. Supported Feature profiles

The implementation is following OCPP 1.6 specification of Open Charge Alliance. According to OCPP 1.6 specification all of features and associated messages are grouped into Feature Profiles.

OCPP 1.6 specified following Feature profiles:

Profile name	Description	Implementation Mandatory
Core	Basic Charge Point functionality comparable with OCPP1.5 without support for firmware updates, local authorization list management and reservations.	Required
Firmware Management	Support for firmware update management and diagnostic log file download.	Optional
Local Auth List Management	Features to manage the local authorization list in Charge Points.	Optional

Profile name	Description	Implementation Mandatory
Reservation	Support for reservation of a Charge Point	Optional
Smart Charging	Support for basic Smart Charging	Optional
Remote Trigger	Support for remote triggering of Charge Point initiated messages	Optional

4.1. Messages

Please see below which messages are supported per OCPP feature profile.

Message	Supported (Y/N)	Comment
Core profile		
Authorize	Y	Example when two different online cards: card #1 swiped and starts, card #2 swipes, card #2 is checked by back-end card, if accepted charging stops. If rejected charging continues.
BootNotification	Y	
ChangeAvailability	Y	
ChangeConfiguration	Y	
ClearCache	Y	The cache is empty, while the charger received the command, it will response accept but do nothing.
DataTransfer	Y	While the charger connects to customer's own back end, customer could use this message for log transfer. NOTE: Please refer OCPP Data Transfer document. This option may become obsolete moving forward for log transfer as GetDiagnostics message is supported.
GetConfiguration	Y	Before FW 1.3 the charger will only response with support keys. After FW1.3 charger will response with all keys, in case the key not supported, the charger will response "unknown" Refer section Supported keys.
HeartBeat	Y	
MeterValues	Y	ABB supports following Measurand types for AC: <ul style="list-style-type: none"> • Energy.Active.Import.Register • Current.Import • Voltage • Power.Active.Import • Current.Offered
RemoteStartTransaction	Y	
RemoteStopTransaction	Y	
Reset	Y	Chargers support hard reset and soft reset. Hard reset fully reboot charger. The resets gracefully stop charging session if one is in progress before resetting.

Message	Supported (Y/N)	Comment
		Soft reset gracefully stops charging session if one is in progress before resetting. Then the charger gracefully disconnects from the server. After disconnection, the charger will reboot itself.
StartTransaction	Y	
StatusNotification	Y	
StopTransaction	Y	
UnlockConnector	Y	Message is supported only to socket variants, upon receiving this message, socket variants charger will release the E-lock of socket. If send the message to cable variants, the charger will response with NotSupported.
Firmware Management profile		
GetDiagnostics	Y	Supports 7-day logs with 300 lines/less than 25kb per day. Charger uploads the files by HTTP/HTTPS/FTP based on the server connection.
DiagnosticsStatusNotification	Y	Support status: Uploading, Uploaded, Upload-Failed, Idle.
FirmwareStatusNotification	Y	Charger will response the status: <ul style="list-style-type: none"> • Downloading • Downloaded • Installing • Installed • DownloadFailed • InstallationFailed
UpdateFirmware	Y	Supports HTTP/HTTPS/FTP. Please note that there are different variants of FTP upgrade methodologies and not all are supported. In section 6.1, more information can be found on the supported use cases.
Local Auth List Management		
GetLocalListVersion	Y	
SendLocalList	Y	Each list is limited to 8 ID tag, each ID tag with max 20 characters; The charger has a total limit of 16 ID tags.
Reservation*		
CancelReservation	N	Planned in roadmap.
ReserveNow	N	Planned in roadmap.
Smart charging*		
ClearChargingProfile	Y	
GetCompositeSchedule	Y	
SetChargingProfile	Y	ChargeProfileMaxStackLevel = 16 for display models. ChargeProfileMaxStackLevel = 3 for non-display models.

Message	Supported (Y/N)	Comment
		Support up to 25 periods for each stack level
Remote Trigger*		
TriggerMessage	Y	Chargers supports below MessageTrigger: <ul style="list-style-type: none"> • BootNotification • DiagnosticsStatusNotification • FirmwareStatusNotification • Heartbeat • MeterValues • StatusNotification
Improved Security*		
SecurityEventNotification	Y	Chargers supports below SecurityEventNotification: <ul style="list-style-type: none"> • FirmwareUpdated • SettingSystemTime • StartupOfTheDevice • ResetOrReboot • MemoryExhaustion • SecurityLogWasCleared • InvalidFirmwareSignature • AttemptedReplayAttacks
GetLog	Y	Chargers only supports request SecurityLog, DiagnosticsLog Charger uploads the security file by HTTP/HTTPS/FTP based on the server connection. Support to save max 128 security events, fixed 64bytes for each event Support to get diagnostics for ordinary log from FW v1.8.32.
LogStatusNotification	Y	Charger will response the status: <ul style="list-style-type: none"> • Uploading • Uploaded • UploadFailure
InstallCertificate	Y	Support to install a new certificate and allow two certificates existing in parallel.
DeleteCertificate	Y	If one of certificates doesn't valid for certain reason or needs to install a new one when two certificates existed, please use this command to delete one of them.
GetInstalledCertificateIds	Y	Using this command to inquiry the status of certificate in charger

Note: Due to system performance, please do not send a large number of OCPP commands in a short period of time. The recommended interval for sending messages is about 1 second.

4.2. Standard Configuration keys

4.2.1. Supported keys

Key Name	Required/ Optional	Description	Type	Accessibility	Default Value
Core profile					
ClockAlignedDataInterval	required	Size (in seconds) of the clock-aligned data interval. This is the size (in seconds) of the set of evenly spaced aggregation intervals per day, starting at 00:00:00 (midnight). For example, a value of 900 (15 minutes) indicates that every day should be broken into 96 15-minute intervals. The range of this value: 0, 30 – (86400-1)	Integer	RW	0
ConnectionTimeOut	required	Interval(from successful authorization) until incipient charging session is automatically canceled due to failure of EV user to (correctly) insert the charging cable connector(s) into the appropriate connector(s). The range of this value: 10 - 240	Integer	RW	120
GetConfigurationMaxKeys	required	The number of configuration keys requested in a single PDU may be limited by the Charge Point. This maximum can be retrieved by reading this configuration key.	Integer	R	20
HeartbeatInterval	required	Interval of inactivity (no OCPP exchanges) with central system after which the Charge Point should send a Heartbeat.req PDU. If the interval less than 10, the AC charger will accept but execute 10	Integer	RW	120
MeterValuesAlignedData	required	Clock-aligned measurand(s) to be included in a MeterValues.req PDU, every ClockAlignedDataInterval seconds. Supported value: Current.Import, Current.Offered, Energy.Active.Import.Register, Energy.Active.Import.Interval, Power.Active.Import, Power.Offered, Voltage.	CSL	RW	Energy.Active.Import.Register
MeterValuesAligned-DataMaxLength	required	Maximum number of items in a MeterValuesAlignedData Configuration Key.	Integer	R	4

Key Name	Required/Optional	Description	Type	Accessibility	Default Value
MeterValuesSampleInterval	required	Interval between sampling of metering (or other) data, intended to be transmitted by "MeterValues" PDUs. The range of this value: 0, 4 - 65534 If the interval less than 4, the AC charger will reject.	Integer	RW	30
LocalAuthorizeOffline	required	Controls whether a Charge Point will authorize a user when offline using the Authorization Cache and/or the Local Authorization List.	boolean	RW	TRUE
LocalPreAuthorize	required	Controls whether a Charge Point will use the Authorization Cache and/or the Local Authorization List to start a transaction without waiting for an authorization response from the Central System.	boolean	RW	FALSE
NumberOfConnectors	required	The number of physical charging connectors of this Charge Point.	Integer	R	1
SupportedFeatureProfiles	required	A list of supported Feature Profiles. Possible profile identifiers: Core, FirmwareManagement, LocalAuthListManagement, Reservation, SmartCharging and RemoteTrigger.	CSL	R	Core, FirmwareManagement, LocalAuthenticationListManagement, SmartCharging, RemoteTrigger
WebSocketPingInterval	optional	Only relevant for websocket implementations. 0 disables client side websocket Ping/Pong. In this case there is either no ping/pong or the server initiates the ping and client responds with Pong. Positive values are interpreted as number of seconds between pings. Negative values are not allowed. ChangeConfiguration is expected to return a REJECTED result. The range of this value: 0, 10 – 65535	Integer	RW	15
AllowOfflineTxForUnknownId	optional	When offline, a Charge Point may allow automatic authorization of any "unknown" identifiers that cannot be explicitly authorized by Local Authorization List or Authorization Cache entries. Identifiers with status other than "Accepted" (Invalid, Blocked, Expired) must be rejected. Now the charger will not allow any ID except in local authentication list while it is offline	boolean	RW	False

Key Name	Required/Optional	Description	Type	Accessibility	Default Value
AuthorizeRemoteTxRequests	required	Whether a remote request to start a transaction in the form of a RemoteStartTransaction.req message should be authorized beforehand like a local action to start a transaction. Now the charger will not send the authorize.req	boolean	RW	False
Local Authorization List Management					
LocalAuthListEnabled	required	Whether the Local Authorization List is enabled	boolean	RW	TRUE
LocalAuthListMaxLength	required	Maximum number of identifications that can be stored in the Local Authorization List	Integer	R	16
SendLocalListMaxLength	required	Maximum number of identifications that can be send in a single SendLocalList.req	Integer	R	8
Smart charging profile					
ChargeProfileMaxStackLevel	required	Max StackLevel of a Charging. The number defined also indicates the max allowed number of installed charging schedules per Charging Purposes.	Integer	R	Based on the charger model
ChargingScheduleAllowedChargingRateUnit	required	A list of supported quantities for use in a ChargingSchedule. Allowed values: 'Current' and 'Power'.	CSL	R	Current,Power
ChargingScheduleMaxPeriods	required	Maximum number of periods that may be defined per ChargingSchedule.	Integer	R	25
MaxChargingProfilesInstalled	required	Maximum number of Charging profiles installed at a time.	Integer	R	Same as ChargeProfileMaxStackLevel
ResetRetries	required	Number of times to retry an unsuccessful reset of the Charge Point. Charger now only supports value 0.	Integer	RW	0
TransactionMessageAttempts	required	How often the Charge Point should try to submit a transaction-related message when the Central System fails to process it. Now the transaction data will always attempt to send to central system until it response	Integer	RW	15
TransactionMessageRetryInterval	required	How long the Charge Point should wait before re-submitting a transaction related message that the Central System failed to process.	Integer	RW	10S
MaxEnergyOnInvalidId	optional	Maximum energy in Wh delivered when an identifier is invalidated by the Central System after start of a transaction.	integer	RW	1 Kwh
StopTransactionOnInvalidId	required	Whether the Charge Point will stop an ongoing transaction when it receives a non-Accepted authorization status in a StartTransaction.conf for this transaction.	Boolean	RW	TRUE

Key Name	Required/Optional	Description	Type	Accessibility	Default Value
		Default value is true.			
MeterValuesSampledData	required	Sampled measurands to be included in a MeterValues.req PDU, every MeterValueSampleInterval seconds. Supported value: Current.Import, Current.Offered, Energy.Active.Import.Register, Power.Active.Import, Voltage.	CSL	RW	Energy.Active.Import.Register, current import power active import, current.Offered Voltage
MeterValueSampleInterval	required	Interval between sampling of metering (or other) data, intended to be transmitted by "MeterValues" PDUs. For charging session data (ConnectorId>0), samples are acquired and transmitted periodically at this interval from the start of the charging transaction. A value of "0" (numeric zero), by convention, is to be interpreted to mean that no sampled data should be transmitted. (range from 1-65535 or 0)	Integer	RW	30 (range from 1-65535 or 0)
MeterValuesSampledDataMaxLength	Optional	Maximum number of items in a MeterValuesSampledData Configuration Key.	Integer	R	5
ConnectorPhaseRotation	required	For individual connector phase rotation information, the Central System may query the ConnectorPhaseRotation configuration key on the Charging Point via GetConfiguration. The Charge Point shall report the phase rotation in respect to the grid connection.	CSL	RW	Single-phase : 0.NotApplicable,1.NotApplicable Three-phase : 0.Unknown,1.Unknown
StopTxnAlignedData	required	Clock-aligned periodic measurand(s) to be included in the TransactionData element of StopTransaction.req MeterValues.req PDU for every ClockAlignedDataInterval of the charging session.	CSL	RW	"Current.Import", "Current.Offered", "Energy.Active.Import.Register", "Energy.Active.Import.Interval", "Power.Active.Import", "Power.Offered", "Voltage"
StopTxnAlignedDataMaxLength	optional	Maximum number of items in a StopTxnAlignedData Configuration Key.	integer		7
StopTxnSampledData	required	Sampled measurands to be included in the TransactionData element of StopTransaction.req PDU, every MeterValueSampleInterval seconds from the start of the charging session	CSL	RW	"Current.Import", "Current.Offered", "Energy.Active.Import.Register", "Power.Active.Import", "Voltage"

Key Name	Required/Optional	Description	Type	Accessibility	Default Value
StopTxnSampledDataMaxLength	optional	Maximum number of items in a StopTxnSampledData Configuration Key.	Integer	R	5
StopTransactionOnEVSideDisconnect	required	When set to true, the Charge Point shall administratively stop the transaction when the cable is unplugged from the EV. NOTE: this parameter is not being used, Transaction will always stop on EV disconnect or even before.	boolean	RW	True
UnlockConnectorOnEVSideDisconnect	required	When set to true, the Charge Point shall unlock the cable on Charge Point side when the cable is unplugged at the EV.	boolean	RW	True
AuthorizationKey	required	The basic authentication password is used for HTTP Basic Authentication, minimal length(As string): 16 characters (from FW v1.8.32, it supports up to 20 characters) FW supports 31 characters(From 1.8.32, it supports up to 40 characters) if using string format. If using hexadecimal format, FW supports 15 bytes and 30 characters at most. (From 1.8.32, it supports up to 20 bytes and 40 characters)	String	W	ABB default randomized key
AuthorizationCacheEnabled	optional	A Charge Point may implement an Authorization Cache that autonomously maintains a record of previously presented identifiers that have been successfully authorized by the Central System.	boolean	RW	false

4.2.2. Not supported keys

Key Name	Required/Optional	Description	Type
Core profile			
MinimumStatusDuration	optional	The minimum duration that a Charge Point or Connector status is stable before a StatusNotification.req PDU is sent to the Central System.	integer
BlinkRepeat	optional	Number of times to blink Charge Point lighting when signaling	integer
ConnectorPhaseRotationMaxLength	optional	Maximum number of items in a ConnectorPhaseRotation Configuration Key	integer

Key Name	Required/ Optional	Description	Type
LightIntensity	optional	Percentage of maximum intensity at which to illuminate Charge Point lighting	integer
SupportedFeatureProfilesMaxLength	optional	Maximum number of items in a SupportedFeatureProfiles Configuration Key.	integer
ConnectorSwitch3to1PhaseSupported	optional	If defined and true, this Charge Point support switching from 3 to 1 phase during a charging session.	boolean
Reservation profile			
ReserveConnectorZeroSupported	optional	If this configuration key is present and set to true: Charge Point support reservations on connector 0.	boolean

4.3. Custom Configuration keys

Key Name	Description	Type	Accessibility	Default Value
FreevendEnabled	In Free Vend mode authorization is disabled and charging could be started without authorization. When this mode is enabled, Authorize message will not be sent to Central System. StartTransaction message will be sent as usually at the beginning of charging session.	boolean	RW	TRUE
FreevendIdTag	In Free Vend mode, use this key to set idTag. Max length is 20 like IdToken type. Central system should be configured to accept StartTransaction with idTag configured for Free Vend mode.	String	RW	Serial number
TimeOffset	This is used for display/local time purposes only. Configured current local time offset in the UTC time offset format. Let say for US, 5 hours ahead of UTC time. So, the value to set "-05:00". If this is not set/empty, it will show in UTC format.	String	RW	"-05:00"
NextTimeOffsetTransitionDateTime	This is used for display/local time purposes only. On this date time, the clock displayed to the EV driver will be given the new offset as configured via 'TimeOffsetNextTransition'. For example, "2022-03-28T02:00:00+01:00". It is represented according to the ISO8601 standard.	String	RW	""
TimeOffsetNextTransition	This is used for display/local time purposes only. New offset that will be set on the next time offset transition as configured via 'NextTimeOffsetTransitionDateTime'. For example, "-04:00"	String	RW	""
ChangeNFCPower	This is used to enable/disable the NFC function and to change NFC power. NFC power is configurable with values 0 to 255. NFC function can be disabled (=0) or enabled (=1). Two values must be set at the same time, otherwise charger will reject the message. For example, "100,1"	CSL	RW	"10,1"

CentralSystemURL	This key is used to change the server URL(only for non-security http link)	String	RW	ABB URL
HTTPBasicAuthEnable	To control whether including "AuthorizationKey" in the field of http GET request or not.	boolean	RW	TRUE
DefaultSolarMode	<p>This is used for set/get solar work mode parameters</p> <p>For example, "1,1 "</p> <p>First parameter: 1 solar enabled 0: disabled</p> <p>Second parameter is solar working mode: 1: MAX Speed 2: Avoid injection/ECO mix (need the 3rd parameter to set a threshold) 3: Solar only</p> <p>Example of avoid injection, "1,2,4" 1: solar enable 2: avoid inject, 4: threshold, unit is amps</p> <p>The scope of threshold is from 0-6 which means to keep charging if solar generates more energy than the threshold of the value read from meter.</p>	CSL	RW	0,1
ActiveSolarMode	<p>This is used to switch solar work mode during charging, it can only be changed during charging</p> <p>For example, "1"</p> <p>1: MAX Speed 2: Avoid injection/ECO mix (need the 3rd parameter to set a threshold) 3: Solar only</p> <p>Example of avoid injection, "2,4" 2: Avoid inject , 4: threshold(only valid for avoid injection mode)</p>	CSL	RW	
ChargingLimitedBy	<p>Get the information of valid instant PWM channel and the corresponding value. Each channel is separated by a comma and the content after the last comma is the current value of the channel</p> <p>.</p> <p>Two examples: 1: DelayCharge,0A. 2: TcBleOutputCurrent,CsBleOutputCurrent,ModbusChargingCurrentLimit,16A</p>	CSL	R	

ChargingLimitedChannels	Get all the information of valid PWM channels and the corresponding current value. Each channel is separated by a comma. For example: [BleSchedule:0A],[TcBleOutputCurrent:16A],[ModbusChargingCurrentLimit:16A],[CsBleOutputCurrent:16A],[RatedOutputCurrent:16A]	CSL	R	
UserSetMaxCurrent	This is used to set the maximum current of the charger which is same with the one configured in TC app. In another word, if it is changed, the value in TC app will be changed at same time. For channel info, it shares the same channel info with TC app.	integer	RW	
MeterValueReportOnSuspendedEV	This key is used to decide whether or not upload meter values to backend when charging is in SuspendedEV status.	boolean	RW	true
StopChargingOnlyWithSameCard	This key is used to decide whether or not only allow to stop charging with same RFID card. If it is set to true, only the RFID card that was used to start the charging session can stop the same charging session.	boolean	RW	false
ExternalMeterSampleInterval	This is used to decide the interval of uploading external meter values to backend by datatransfer command when enable cloud load management.	integer	RW	30 s, range from 4s-65535
AuthReqInOfflineMode	If this key is disabled and network connection is lost, charger will automatically work in freevending mode.	boolean	RW	true
StartTransactionBeforeEnergyDelivered	This key is used to be compatible with versions on the market due to different moment of sending StartTransaction . As default true, it will send StartTransaction when the authorization is passed and the cable is plugged. (as the behavior before 1.8.x) If configured to false, it only reports StartTransaction when the energy is first time delivered(Relay is closed first time).	boolean	RW	true
ABBIdTag	This key can be used to query and configure the enabling switch of third - party RFID cards. When configured as "true", the enabling of third - party RFID cards is turned off. when configured as "false", the enabling of third - party RFID cards is turned on.	boolean	RW	true

ChargeRecordUseTotalEnergy	The function of this key is to determine whether the electricity quantity in the orders uploaded via OCPP uses cumulative electricity or starts from zero for the current order. When set to false, the starting value of the order is zero, and the ending value is the electricity consumed during the current order. When set to true, both the starting and ending values of the order are cumulative electricity quantities.	boolean	RW	false
PWMAfterStartTransaction	This key is used to control the timing of PWM . When set to FALSE, PWM wave generation will occur upon successful authentication. When set to TRUE, PWM wave generation will only take place after receiving a response to the start transaction.	boolean	RW	false

4.4. OCPP ABB error code definition

Error Code	OCPP Definition	ABB Definition
0x0000	NoError	
0x0002	OtherError	Residual current detected
0x0004	GroundFailure	
0x0008	OverVoltage	
0x0010	UnderVoltage	
0x0020	OverCurrentFailure	
0x0040	OverCurrentFailure (Severe over current)	
0x0080	HighTemperature	
0x0100	OtherError	Control pilot (-12V)
0x0200	PowerMeterFailure	
0x0400	OtherError	PowerRelay fault (RelayError)
0x0800	OtherError	Internal communication failure (CommError)
0x1000	ConnectorLockFailure	E-Lock failure (Lock/Unlock/PPLine)
0x2000	OtherError	Missing phase
0x4000	OtherError	Modbus communication lost (ModbusCommError)
0x8000	OtherError	Offline storage error

4.5. Security

4.5.1. Encryption

In addition to network level security, ABB OCPP 1.6 implementation supports OCPP-J over TLS security(security profile 2). TLS 1.2 is supported. It is up to Central System operator to decide if TLS with WebSocket (WSS) is used or not. No additional configuration changes are required to enable it. For more information on encryption with OCPP 1.6-J please see chapter “6.2.1 Encryption” of [2].

4.5.2. Authentication

ABB OCPP 1.6 implementation supports basic HTTP authentication. Username equals charge point ID and password/authorization keys can optionally be set during installation.

Charger supports write-only OCPP AuthorizationKey which is maximum of 20 bytes where central system could change the key using ChangeConfiguration.req.

For more information on this key please see chapter 4.2.1.

From FW1.8.2, FW supports a customized key “HTTPBasicAuthEnable” to control whether including “AuthorizationKey” in the field of http GET request or not.

For more information on this key please see chapter **Error! Reference source not found..**

4.5.3. Security event and security logs

In case of a security event, ABB OCPP 1.6 supports messages of security by sending SecurityEventNotification to central system and in parallel will save security logs which can be fetched by command of GetLog.

For details, refer to chapter 4.1.

4.5.4. Certificate management

ABB OCPP 1.6 implementation supports basic certificate management:

InstallCertificate, DeleteCertificate, GetInstalledCertificateIds

By this way, CPO can add a new certificate or delete one of them if needed.

For details, refer to chapter 4.1.

4.6. Certification maturity

This section provides a summary towards certification readiness of Terra AC charger.

Functionality	Full Certificate	Subset Certificate	TAC support maturity
Core	Mandatory	Mandatory	Complete
Firmware Management	Mandatory	Optional	Complete
Smart Charging	Mandatory	Optional	Complete
Reservation	Mandatory	Optional	Planning
Local Auth List Management	Mandatory	Optional	Complete
Remote Trigger	Mandatory	Optional	Complete

Remark:

Refer “Messages” section for more details.

5. Additional Information

5.1. Listing of related documents

Ref #	Document Kind, Title
1	Open Charge Point Protocol 1.6 edition 2 FINAL, 2017-09-28
2	Open Charge Point Protocol JSON 1.6, OCPP-J 1.6 Specification
3	Improved security for OCPP 1.6-J edition 2 FINAL, 2020-03-31
4	TerraConfig App & Portal
5	Accounts and 3 party OCPP backend configuration
6	OCPP Data Transfer document
7	The Electric Vehicles (Smart Charge Points) Regulations 2021 (legislation.gov.uk)

6. Addendum

6.1. Firmware upgrade from external OCPP server

ABB E-mobility will share the bin files of a new firmware release in the [ABB library](#). The bin files in this library will, for all new firmware versions that are to be released from February 2024 onwards, a specific folder naming that will make it easier for charge point operators to upgrade their installed base based upon the firmware package version of the affiliated charger models. In Appendix A, more information regarding this bin file naming strategy is available.

To upgrade our chargers, the Central System need to use “UpdateFirmware” message with the location field where firmware files are stored. Please refer OCPP1.6 specification [1] section “6.55. UpdateFirmware.req”. To start a process of download, OCPP server should send command to the charger to trigger the process.

Before firmware version 1.8.0, the charger only supports firmware upgrade using http/https mode. After FW v1.8.0, the firmware also supports FTP.

OCPP command example:

```
<MessageTypeId>, "<UniqueId>", "<Action>", {<Payload>}]
```

```
[2,"256530071089713152","UpdateFirmware",{"location":"https://down.XXX.com/TAC3Z9118906710247-MID-3P-1.8.0/?524&TAC22W910012345678","retrieveDate":"2021-12-15T13:18:55.000Z"}]
```

Note:

The location of bin files should be same as below:

https://Domain name+ port number + Relative path address of the update bin files+/?+ quantity of bin files&package code

Only this format can be identified by charger. Port number is unnecessary.

URL Example:

```
https://down.XXX.com:8089/TAC3Z9118906710247-MID-3P-1.8.0/?524&TAC22W910012345678
```

- The maximum length of domain is limited to 47
- :8089 in example is optional port number. In case here is no port number mentioned, 443 will be taken as default for https and 80 for http.
- CE-N-22-1.4.2 is a folder path on the server for bin files. The maximum length of the folder path is limited to 63.
- The number of 524 is the quantity of bin files of the updating package. It should strictly meet the quantities in the folder for the target upgrading bin files.
- &TAC22W910012345678 is the FW package code which is an optional parameter.
- The protocol should be same as server connection type such as http or https. And charger will initiate GET request to fetch files one by one.

FTP specific URL format

From firmware version 1.8 and above, the charger supports FTP to update firmware or upload diagnostic logs. FW can support FTP data transmission in 4G and LAN mode. FTP is not supported in Wi-Fi mode.

FW supports basic authentication and anonymous authentication.

FTP basic authentication format example:

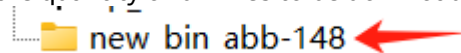
```
ftp://username:password@down.XXX.com:21/TAC3Z9118906710247-MID-3P-1.8.0/?524&TAC22W910012345678
```

FTP anonymous authentication format example:

```
ftp://down.XXX.com:21/TAC3Z9118906710247-MID-3P-1.8.0/?524&TAC22W910012345678
```

Note:

- The maximum length of username and password is up to 31.
- The maximum length of domain is up to 63.
- The maximum length of folder path is up to 63.
- Port number (:21) must be present for FTP formats (even default ones). Above FW1.8.32, it is NOT a must any more for the default port number21
- The number of 524 is the quantity of bin files of the updating package. It should strictly meet the quantities in the folder for the target upgrading bin files. Above FW1.8.32, the number in folder name is the quantity of bin files, there is no need to place parameters in the URL after question mark if using the folder provided by ABB. Like the number(148) in picture below, this is the quantity of bin files to be downloaded.



OCPP upgrade command example:

```
[2,"256530071089713152","UpdateFirmware",{"location":"ftp://down.XXX.com:21/CE-N-22-1.4.2/?524&TAC22W910012345678","retrieveDate":"2021-12-15T13:18:55.000Z"}]
```

Remarks:

- Need to be connected to ABB cloud (not 3rd party) in order to upgrade from v1.5.X(or previous version) to v1.6.X(or above) when using ABB URL link for BIN files.
- OR CPO needs to implement http(s) approach to get chargers upgraded accordingly based on ABB upgrade rules.
- Above FW1.8.32, FW upgrading speed is reduced significantly because of fewer bin files with bigger size.

6.2. Regional specific interface support

6.2.1. UK - The Electric Vehicles (Smart Charge Points) Regulations 2021

Charger provides below OCPP interface options to adhere compliance with regulation. This information facilitates customer/end user if they prefer enable/control regulation related actions. Since this is provided as an additional option, its customer/end user responsibility to configure the chargers according to their preferences. For more details about regulation refer [The Electric Vehicles \(Smart Charge Points\) Regulations 2021 \(legislation.gov.uk\)](https://www.legislation.gov.uk).

6.2.1.1. Randomised delay - Custom configuration keys

Key Name	Description	Type	Accessi- bility	Default Value
RandomDelay	<p>RandomDelay has three parameters: delay duration, random mode and enable mode. OCPP send message sequence according to {Delay duration, Random mode, Enable mode}</p> <p>Delay duration: The range is from 0 to 1800 in seconds.</p> <p>Random mode: Below mode of operation possible.</p> <ul style="list-style-type: none"> 1 - delay duration is always fixed. 2 - delay duration is randomly generated (with in the Delay duration) by charger for each transaction. <p>Enable mode: Below mode of operation possible.</p> <ul style="list-style-type: none"> 0 - Disable random delay. 1 - Enable random delay and it will take effect for each start of transactions and the start of each schedule (peak/off-peak hour). <p>Remark: Similar setting is influenced by mobile app where Random Mode to 2 and Enable mode to 1 set implicitly.</p>	CSL	RW	Delay duration :600 Random Mode:2 Enable Mode: 1
Ran- domDelayCancel	<p>If set to true, RandomDelay will be cancelled and starts charging directly. After each transaction is finished, this key will be set to default value automatically.</p> <p>Remarks: This provides similar behavior provided by Charger Sync mobile app.</p>	boolean	RW	False

6.2.1.2. Off-peak charging - Example preset charging schedule

Regulation indicates that "peak hours" means 8am to 11am on weekdays and 4pm to 10pm on weekdays. And here is an example OCPP command (one of the ways) how charger could be configured to incorporate pre-set default charging hours which are outside of peak hours.

ChargingProfile			
chargingProfileId	14		
stackLevel	14		
chargingProfilePurpose	TxDefaultProfile		
chargingProfileKind	Recurring		
recurrencyKind	Daily		

chargingSchedule			
	duration	86400	
	startSchedule	2022-06-15T00:00:00.0Z	
	chargingRateUnit	A	
	chargingSchedulePeriod		
		startPeriod	0
		limit	32
		startPeriod	28800
		limit	0
		startPeriod	39600
		limit	32
		startPeriod	57600
		limit	0
		startPeriod	79200
		limit	32
chargingProfileId	15		
stackLevel	15		
chargingProfilePurpose	TxDefaultProfile		
chargingProfileKind	Recurring		
recurrencyKind	Weekly		
chargingSchedule			
	duration	172800	
	startSchedule	2022-06-18T00:00:00.0Z	
	chargingRateUnit	A	
	chargingSchedulePeriod		
		startPeriod	0
		limit	32

Remarks:

Example is provided with following assumptions.

- stackLevel number intended to indicate hierarchy stack of profiles where higher values have precedence over lower values.
- startSchedule intended to start from 15-July-2022 and stackLevel 15 is to support exception on weekend days.

6.2.2. United States (US) specific support

Touch screen model - Custom configuration keys

Key Name	Description	Type	Accessi- bility	Default Value
ScreenDisplay- Setting	<p>It has two values of integer type.</p> <p>ScreenOffTime: Select how long you want your charger display to wait before turning the screen off.</p> <p>The possible value is 10 to 300 seconds. When value is 0, then display will be always ON. If the value is other than possible range defined, it will be 'Rejected'.</p> <p>Brightness: possible range to adjust screen brightness is 10 to 100. If the value is other than possible range defined, it will be 'Rejected'.</p>	CSL	RW	120, 100
ScreenBright- nessLevel	<p>This key change the screen brightness. The value of key is 1 or 2, other values will be rejected. Level 1 mean the brightness is 100%, level 2 mean the brightness is 50%.</p>	Integer	RW	1
CustomDisplay- CostAndPrice	<p>CTEP related key</p> <p>It means whether or not CTEP pro-protocol is supported, and False means that it is not supported.</p> <p>Central System can change this key to enable CTEP.</p>	boolean	RW	FALSE
DefaultPrice	<p>CTEP related key.</p> <p>There are three values with integer type.</p> <p>priceText: holds a text to show on the display.</p> <p>priceTextOffline: be used to display a different pricing message when the charge point is offline.</p> <p>chargingPrice: contain the pricing components that the charge point should use in case a transaction is started while the charge point is offline and cannot receive RunningCost messages.</p> <p>An example message for DefaultPrice: ChangeConfiguration.req("DefaultPrice", "{ "priceText": "0.15 \$/kWh, idle fee after charging: 1 \$/hr", "priceTextOffline": "The station is offline. Charging is possible for 0.15 \$/kWh.", "chargingPrice": { "kWhPrice": 0.15, "hourPrice": 0.00, "flatFee": 0.00 }}")</p>	CSL	RW	0
User-specific Price	<p>CTEP related function.</p> <p>When the user has been identified, Central System can send the user-specific price.</p> <p>Example: DataTransfer.req("ABB", "SetUserPrice", "{ "idToken": "12345678", "priceText": "\$0.12/kWh, no idle fee" }")</p>			
Final Cost	<p>CTEP related function.</p> <p>The following message is used to send the final cost. It is linked to the transaction and sent at the end of the transaction.</p> <p>Example: DataTransfer.req("org.openchargealliance.costmsg", "FinalCost", "{ "transactionId": 98765, "cost": 3.31, "priceText": "\$2.81 @ \$0.12/kWh, \$0.50 @ \$1/h, TOTAL KWH: 23.4 TIME: 03.50 COST: \$3.31. Visit www.cpo.com/invoices/13546 for an invoice of your session.", "qrCodeText": "https://www.cpo.com/invoices/13546" }")</p>			

RunningCost Message	<p>CTEP related function.</p> <p>The RunningCost message is used to display the price and cost during the transaction.</p> <p>Example: DataTransfer.req("vendorId": "org.openchargealliance.costmsg", "messageId": "RunningCost", "data": "{ "transactionId": 12345, "timestamp": "2021-03-19T12:00:00Z", "meterValue": 1234000, "cost": 1.00, "state": "Charging", "chargingPrice": { "kWhPrice": 0.123, "hourPrice": 0.00, "flatFee": 0.00 }, "idlePrice": { "graceMinutes": 30, "hourPrice": 1.00 } }"</p>			
----------------------------	--	--	--	--

6.2.3. Japanese specific support

Key Name	Description	Type	Accessi- bility	Default Value
FelicaAdvancedAuthorization	<p>This is used to submit additional Felica S_PAD0 and DataFormatCode to backend via DataTransfer before transferring the authentication message</p> <p>For example, "1" 1: Enable FeliCa transmission S_PAD0 0: Disable FeliCa transmission S_PAD0</p> <p>Example: DataTransfer.req({ "vendorId": "ABB", "messageId": "EneKeyData", "data": "{\"EneKeyData\":\"4020900000020539\", \"DataFormatCode\": \"0077\", \"idTag\": \"012E4CE082548F69\"}" })</p> <p>Response: {"status": "Accepted"} {"status": "Rejected"}</p> <p>FW will ignore the response and still handle the response from authorize.req</p>	boolean	RW	0

6.3. Security digital certificate

This section provides complementary information for the Terra AC charger (TAC) user to understand brief information about security/digital certificate to establish secured communi-

cation with their own OCPP server/backend/Central Systems via OCPP communication protocol. And, whether security needed or not, and different OCPP security profiles are out the scope.

1. What is certificate?

Simple analogy is Passport or University degree certificate. A well know **trusted** bodies/authorities provide **identity of the owner** which is **publicly** accepted.

Now, with the security context, certificate is used as **identity** of individual, computer, and other entities on the network and **public key**.

This certificate is intended for the following purposes:

- Proves your identity to a remote computer
- Ensures the identity of a remote computer

And a certificate contains below general information while keep aside the detailed technical information.

Issued to :

Issue from:

Valid from x to y date.

2. Why certificate needed?

To establish a secured connection, a system uses Secure Socket Layer (SSL) also known as Transfer Layer Security (TLS) protocol communication which is based on public key cryptography. In public key cryptography, a matching pair of keys is used; one for encryption and the other for decryption. One of the keys is called the **public key** (can be sent over the network to another system). The other is called the private key (kept secretly by the owner).

The purpose of digital certificate is to ascertain the public key belongs to someone who claim who own. The certificate contains the name of the person (or the organization), together with the public key.

3. Why is certificate needed for TAC?

OCPP specification provides a way to achieve security measures for charge point (TAC) and Central System (let say OCPP sever), to establish a secure connection between charger and OCPP server, also to validate/authenticate whether charger is communicating to correct OCPP sever.

The secure/encrypted communication shall be achieved by web socket secure(wss), whereas OCPP server authentication is based on TLS and public key cryptography using X.509 certificates.

4. Who issues certificate?

Certificate Authority (CA)/Third Party:

CA is a trusted organization who validate the identity of the holder and provide a certificate. For example, to name few Symantec, GoDaddy, DigiCert etc., who sign and provide a digital certificate based on applicant/user Certificate Signing Request (CSR).



Self-signed certificate:

This is created, issue, signed by themselves. A self-signed certificate can be used in private network.

A Certificate Authority (CA) signs (i.e., encrypts) the certificate using its private key. Consequently, the certificate can only be decrypted using CA's public key. CAs are considered trustworthy, and their public key are pre-installed into the browser. One single CA (or a few CAs) to sign all the certificates and CAs are organized in hierarchy. The root CA, whose public key is pre-installed inside the browser, signs the certificate of sub-CAs. The sub-CAs can sign the certificate of sub-sub-CAs. The sub-sub-CAs can then sign the certificate of end-users. Because of this hierarchical structure, the certificate verification process involves a chain of certificates, all the way back to a root CA.

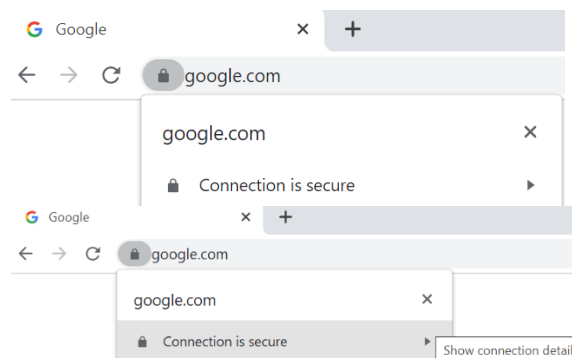
For TAC, its recommended to use CA signed certificate however it's not a mandate. And root CA is used for authentication.

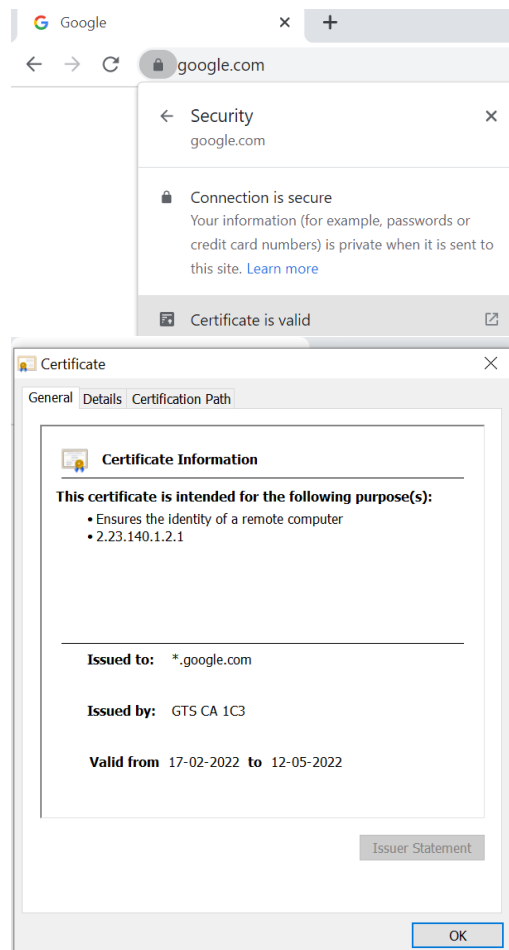
5. How to download my server root certificate?

During installation process, installer would request to provide your OCPP server URL and its certificate.

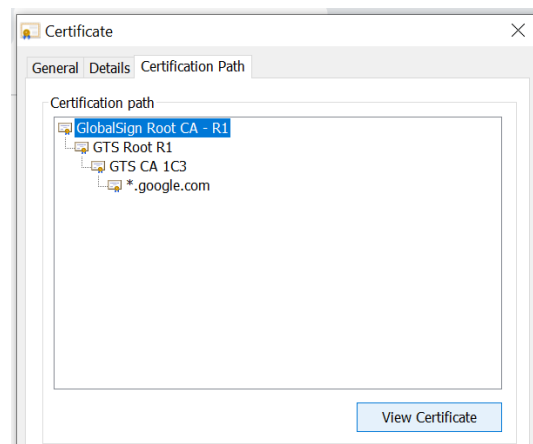
Let us consider, google.com is your server, below steps would help to get the certificate of your server in chrome browser. You will be able to relate if you are using different browser.

- a. Click the secure(lock) symbol in the server URL and navigate further to open certificate as shown in the snap.

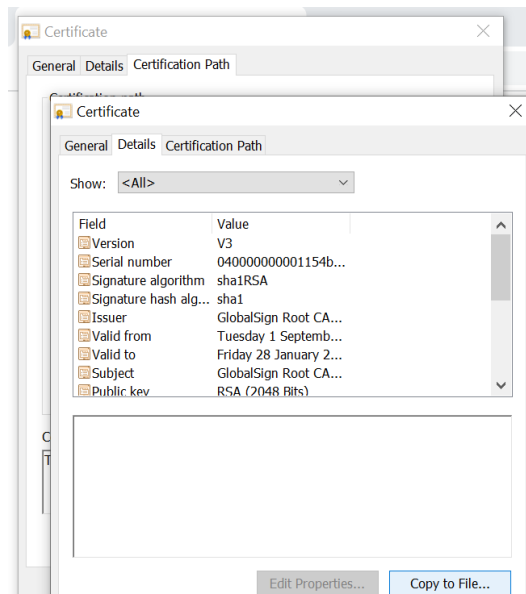




b. Go to certificate path tab and select the Root CA. And then click view certificate.



c. Go to the "Details" tab and click copy to file.



d. Click Next and continue

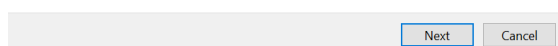


Welcome to the Certificate Export Wizard

This wizard helps you copy certificates, certificate trust lists and certificate revocation lists from a certificate store to your disk.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.



e. Select Base 64 encoded as shown, proceed to Next.



← Certificate Export Wizard

Export File Format

Certificates can be exported in a variety of file formats.

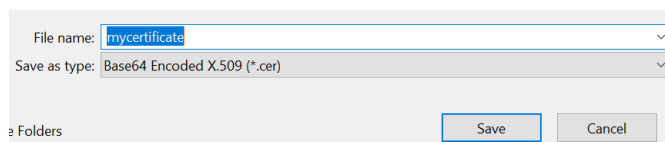
Select the format you want to use:

- DER encoded binary X.509 (.CER)
- Base-64 encoded X.509 (.CER)
- Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
 - Include all certificates in the certification path if possible
- Personal Information Exchange - PKCS #12 (.PFX)
 - Include all certificates in the certification path if possible
 - Delete the private key if the export is successful
 - Export all extended properties
 - Enable certificate privacy
- Microsoft Serialized Certificate Store (.SST)

Next

Cancel

f. Save the certificate file as below (i.e.) .cer extension.



g. Finish the wizard. Now you have certificate of your sever to share with installer to configure with TAC charger.