



OCPP 1.6

Implementation Overview

Notice

This document contains information about one or more ABB products and may include a description of or a reference to one or more standards that may be generally relevant to the ABB products. The presence of any such description of a standard or reference to a standard is not a representation that all of the ABB products referenced in this document support all of the features of the described or referenced standard. In order to determine the specific features supported by a particular ABB product, the reader should consult the product specifications for the particular ABB product.

ABB may have one or more patents or pending patent applications protecting the intellectual property in the ABB products described in this document.

The information in this document is subject to change without notice and should not be construed as a commitment by ABB. ABB assumes no responsibility for any errors that may appear in this document.

In no event shall ABB be liable for direct, indirect, special, incidental or consequential damages of any nature or kind arising from the use of this document, nor shall ABB be liable for incidental or consequential damages arising from use of any software or hardware described in this document.

This document is originally written in English. Other language versions are a translation of the original document and ABB cannot be held liable for errors in the translation.

This document and parts thereof must not be reproduced or copied without written permission from ABB, and the contents thereof must not be imparted to a third party nor used for any unauthorized purpose.

Copyrights

All rights to copyrights, registered trademarks, and trademarks reside with their respective owners.

Copyright © 2023 ABB.

All rights reserved.

Table of Contents

Table of Contents	3
Revisions	4
Overview	5
Supported Functionality	6
MeterValues.....	8
Voltage	8
Current.Offered and Power.Offered.....	9
Temperature.....	9
Sampling Moments.....	9
ABB Custom Extensions.....	10
Free Vend Mode.....	10
Smart Charging Custom Configuration Key	
DiscardTxProfileWhenConnectionLoss	11
Reference to the Payment Session for a Payment Terminal Authorized Session	
in the idTag of StartTransaction	11
Reference to the Vehicle ID for an Autocharge Authorized Session in a Custom	
DataTransfer Message	11
ABB Implementation Choices for Certain OCPP 1.6 Unspecified Areas	12
StopTransaction in Case of Power Supply Failure.....	12
StartTransaction and StopTransaction Timestamps.....	12
Date and Time Format	13
Known Issues	13
RebootRequired status in ChangeConfiguration.conf.....	13
MeterValuesSampledData, MeterValuesAlignedData Keys and Other Supported	
CSL Configuration Keys that Accept Any String	13
Configuration Keys	14
Supported Configuration Keys.....	14
Not Supported Configuration Keys.....	18
ABB Custom Configuration Keys.....	19
Security	20
Encryption	20
Authentication.....	20
Reference Documentation	20

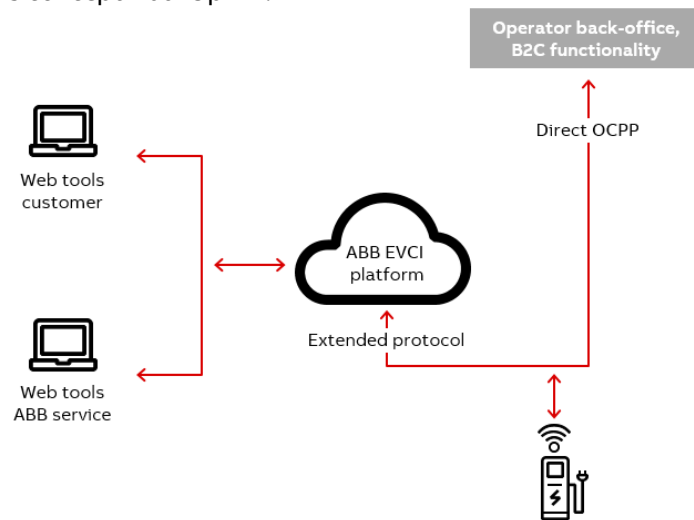
Revisions

Revision	Description	Date	Author
A	First issue.	12 February 2019	Mikhail Kireev
1.2	Added description for new messages and additional Measurands. The first part of the document is restructured.	06 January 2020	Mikhail Kireev
1.3	<ul style="list-style-type: none"> Added TriggerMessage in “Supported functionality”, Added chapter “Smart Charging custom configuration key DiscardTxProfileWhen-ConnectionLoss”, Added chapter “Reference to payment session for payment terminal authorized session in idTag of StartTransaction”, Added part “ABB implementation choices for some of OCPP 1.6 unspecified areas”, Added part “Known issues”. 	15 February 2021	Mikhail Kireev
1.4	<ul style="list-style-type: none"> Some typos and small mistakes corrected, Updated Firmware Management profile information in “Supported functionality”, Updated AC socket information in “MeterValues”, Updated chapter “Free Vend Mode”, Added chapter “Reference to Vehicle ID for autocharge authorized session in custom DataTransfer message”, Updated chapter “StopTransaction in case of power supply failure”, Added chapter “StartTransaction and StopTransaction timestamps”, Added chapter “Date and Time format”, Updated list of Supported / Not Supported keys and added new ABB custom keys in “Configuration keys” part, Added Value Range information for keys of Type ‘int’ in “Configuration keys” part. 	30 July 2023	Evangelos Kontosis

Overview

This document describes the OCPP 1.6 functionality supported by ABB DC chargers according to the OCPP protocol specification and ABB specific extensions of OCPP 1.6.

On its DC chargers, ABB implemented the OCPP 1.6-J version, that uses JSON over WebSockets. An ABB DC charger connects to an OCPP Server directly and, in parallel, it also connects to the ABB Ability™ Platform. This setup enables efficient remote support and offerings simultaneously with OCPP. ABB names this concept Dual Uplink.



Supported Functionality

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

OCPP 1.6 specifies the following Feature Profiles:

Profile name	Description	Mandatory
Core	Basic Charge Point functionality comparable with OCPP 1.5 without support for firmware updates, local authorization list management and reservations.	Yes
Firmware Management	Support for firmware update management and diagnostic log file download.	No
Local Auth List Management	Features to manage the local authorization list in Charge Points.	No
Reservation	Support for reservation of a Charge Point.	No
Smart Charging	Support for basic Smart Charging.	No
Remote Trigger	Support for remote triggering of Charge Point initiated messages.	No

For more information about Feature Profiles please refer to [1] “3.2 Feature Profiles”.

The Table below shows the mapping between the various OCPP 1.6 Feature Profiles and the software versions of the ABB DC Fast Charging products in which these profiles are supported:

Profile \ Model	Core	Firmware* Management	Local Auth List Management	Reservation	Smart Charging	Remote Trigger
Terra T53 and T54	4.0.0.x	4.0.0.x*	4.0.0.x	To be supported**	4.3.x	4.5.x (partially)
Terra High Power T175	Any version	Any version*	Any version	To be supported**	1.2.x	1.5.x (partially)
Terra HVC (Connector and OppCharge)	1.2.x	Not supported	N/A	N/A	1.2.x	1.5.x (partially)
DC Wallbox	Any version	Any version*	Any version	To be supported**	Any version	1.5.x (partially)
Terra 94/124/184	Any version	Any version*	Any version	To be supported**	Any version	1.7.0 (partially)

* Messages for the Firmware Management profile are implemented. However, the update of the charger firmware via OCPP is currently disabled due to security concerns and potential SLA violation. In the standard ABB firmware update process, the operator is informed in advance about firmware updates, including release notes. That gives the operator the possibility to influence the process and ask questions about the possible impact. ABB rolls out the new firmware in a controlled way and ensures successful completion of the firmware update process through a separate dedicated connection. In this way, the firmware updates can be more effectively controlled compared to using only OCPP. Furthermore, they are executed effortlessly for the operator.

** The reservation profile is not yet implemented in the current software version for the chargers. ABB intends to provide this functionality in the future software versions. Updating to these versions will be possible remotely, without service personnel visiting the charger.

The following table lists the messages which are supported per OCPP 1.6 Feature Profile. Note that the charger can produce WebSocket messages with a maximum size of 128KB.

Message	Supported (Y/N)	Comment
Core profile		
Authorize	Y	
BootNotification	Y	
ChangeAvailability	Y	
ChangeConfiguration	Y	
ClearCache	Y	
DataTransfer	Y	General message is supported. Custom functionality based on this command shall be discussed with ABB separately.
GetConfiguration	Y	
HeartBeat	Y	
MeterValues	Y	<p>ABB supports the following Measurand types:</p> <ol style="list-style-type: none"> 1. Current.Import, 2. Energy.Active.Import.Register, 3. Power.Active.Import, 4. SoC, 5. Voltage, 6. Current.Offered*, 7. Power.Offered*, 8. Temperature*. <p>Note: Please see the chapter on MeterValues below for more details.</p>
RemoteStartTransaction	Y	
RemoteStopTransaction	Y	
Reset	Y	Both Soft and Hard reset are supported. Soft reset restarts the charger software, Hard reset fully reboots the charger. Before resetting, both reset types gracefully stop the charging session if it is in progress.
StartTransaction	Y	
StatusNotification	Y	
StopTransaction	Y	
UnlockConnector	Y	The message is supported but does not unlock the connector on EVSE side simply because ABB DC fast charging does not normally have sockets, only cables. Upon receiving this message, the charger will only stop any active charging session on the desired connector, as specified by OCPP. ABB would like to stress out that according to the OCPP specification the UnlockConnector command should not be used to remotely stop a running transaction, RemoteStopTransaction should be used instead.
Smart Charging		
SetChargingProfile	Y	
ClearChargingProfile	Y	
GetCompositeSchedule	N	
FirmwareManagement profile*		
GetDiagnostics	Y	<p>The following types of logs can be retrieved:</p> <ol style="list-style-type: none"> 1. 1st line service, 2. Debug, 3. Binary.

		In the case of Debug logs, the downloaded file is encrypted and password protected. Only ABB service personnel has the ability to open it.
DiagnosticsStatusNotification	Y	
UpdateFirmware	N	Firmware update via OCPP is currently disabled due to security concerns and potential SLA violation.
FirmwareStatusNotification	N	Firmware update via OCPP is currently disabled due to security concerns and potential SLA violation.
Local Authorization List Management		
GetLocalListVersion	Y	
SendLocalList	Y	
Remote Trigger		
TriggerMessage (BootNotification)	Y	
TriggerMessage (StatusNotifciation)	Y	
TriggerMessage (DiagnosticsStatusNotification)	N	
TriggerMessage (FirmwareStatusNotification)	N	
TriggerMessage (HeartBeat)	N	
TriggerMessage (MeterValues)	N	

*Available from Software versions 1.2.x for Terra HP and HVC and from 4.3.x for Terra 53/4.

MeterValues

As mentioned above, the following Measurands are supported:

1. Current.Import,
2. **Energy.Active.Import.Register,**
3. Power.Active.Import,
4. **SoC,**
5. Voltage,
6. Current.Offered,
7. Power.Offered,
8. Temperature.

‘Energy.Active.Import.Register’ and ‘SoC’ are enabled by default. Other values could be enabled through the ChangeConfiguration command for the keys MeterValuesAlignedData, MeterValuesSampledData, StopTxnAlignedData, and StopTxnSampledData. See Configuration Keys chapter for the description of these keys.

Note: For an AC cable or a socket of T53 and T54 chargers only the ‘Energy.Active.Import.Register’ Measurand is currently supported.

Voltage

In the OCPP 1.6 protocol specification [1], Chapter 7.31, the Voltage value of a Measurand is specified as an “Instantaneous AC RMS supply voltage”. In ABB DC Fast charging, the Voltage field contains the instantaneous voltage measured on DC contactors of the plug during charging.

Current.Offered and Power.Offered

The Current.Offered and Power.Offered measurands are dependent on the power management software component. Thus, it is not enough to only enable these values by adding them to Measurands in the MeterValuesAlignedData and MeterValuesSampledData keys.

To enable Current.Offered and Power.Offered, either the Smart Charging profile should be activated or (if Smart Charging is not used) the power management component should be activated in the charger configuration by ABB Service.

The Current.Offered and Power.Offered values represent the actual maximum current and power, respectively, that could be given to the connected vehicle based on the maximum capabilities of the vehicle provided during the charging session setup. These Measurands do not always represent the charger's maximum capabilities without considering the context of the connected vehicle.

For example, on a 350kW High-Power charger, if during a session setup the vehicle communicates that it could charge up to a maximum of 100kW, the charger will allocate only 100kW, with 10-15% of extra power, and will communicate this number through Power.Offered. Accordingly, the Power.Offered will not contain the 350kW power value.

Note: An additional 10-15% of power is attributed to the usage of AC input current from the grid for power and current management. Therefore, some margin is added during the calculation of DC values.

Temperature

The temperature measurand is available for connectorId=0 only. To enable the sending of temperature measurements, the "Temperature" value should be added to the key MeterValuesAlignedData.

The temperature is measured inside one or several enclosures of the charging system. Therefore, in the case of multiple enclosure topology e.g., a charge post + power cabinet(s), several temperature measurands are provided, one per enclosure.

Depending on the number of enclosures, one or more temperature values are placed in a comma-separated list inside Measurands. Below is an example of a JSON structure containing two temperature measurements for a Heavy Vehicle Charger which consists of a depot box and a power cabinet. The power cabinet temperature is 33.8°C and depot box temperature is 18.6°C:

```

"MeterValues": {
  "connectorId": 0,
  "meterValue": [{
    "sampledValue": [{
      "unit": "Celsius",
      "context": "Sample.Clock",
      "measurand": "Temperature",
      "location": "Body",
      "value": "33.8,18.6"
    }],
    "timestamp": "2019-12-19T14:20:00.155Z"
  }]
}

```

Sampling Moments

Samples are taken at different moments in time, depending on the value of the interval configuration, but always only during an ongoing charging session, as specified in Section 3.16 [1].

ClockAlignedDataInterval (if greater than 0):

- At fixed moments in time during the charging session (readingContext = Sample.Clock).

MeterValueSampleInterval (if greater than 0):

- At the start of the transaction (readingContext = Transaction.Begin),
- Periodically with a specified interval during the charging session (readingContext = Sample.Periodic),
- At the end of the transaction (readingContext = Transaction.End).

Which Measurands (CSL)	When	OCPP message	ConnectorID
MeterValuesAlignedData	ClockAlignedDataInterval	MeterValues.req	0
MeterValuesSampledData	MeterValueSampleInterval	MeterValues.req	>0
StopTxnAlignedData	ClockAlignedDataInterval	StopTransaction.req	>0
StopTxnSampledData	MeterValueSampleInterval	StopTransaction.req	>0

MeterValues with different connectorId will be sent in separate MeterValues.req messages.

ABB Custom Extensions

ABB supports the following additional features for OCPP 1.6-J.

Free Vend Mode

An ABB DC Fast charger could be set to a so-called Free Vend mode. In this mode, authorization is disabled, and charging could be started without authorization. When this mode is enabled, the Authorize message will not be sent to the Central System. The StartTransaction message will be sent as usual at the beginning of a charging session.

To enable this mode, the Central System should send the ChangeConfiguration message for the FreevendEnabled key with the TRUE value (the default value for this key is FALSE). For specification of this configuration key please see Chapter ABB Custom Configuration Keys.

When the Free Vend mode is enabled and charging is started locally, the StartTransaction message will contain an idTag as configured by the custom configuration key FreevendIdTag. However, if Autocharge is enabled via the Charger Management System and the vehicle communicates its Vehicle ID to the charger, the idTag parameter will contain this Vehicle ID as value instead.

When the Free Vend mode is enabled and charging is started remotely via a RemoteStartTransaction message, the StartTransaction message will contain the idTag referenced in the received RemoteStartTransaction message.

Note: The central system should be configured to accept the StartTransaction message with the id-Tag configured for the Free Vend mode. If this idTag is not known by the Central System, and the Central System rejects StartTransaction for this idTag, charging will be stopped with the Reason = DeAuthorized, unless the configuration key StopTransactionOnInvalidId is set to FALSE.

Smart Charging Custom Configuration Key DiscardTxProfileWhenConnectionLoss

There are cases when it is desired to cancel Smart Charging limits applied to the ongoing charging session if the connection to the Central System is lost. For these cases ABB has implemented the custom configuration key DiscardTxProfileWhenConnectionLoss.

The key is Boolean, when it is set to TRUE and the connection to the Central System is lost (the Web-Socket connection is dropped), the txProfile, if such profile exists for the ongoing charging session, will be deleted immediately and all the limits applied by it will be not considered.

All other types of Smart Charging profiles will not be affected.

When the key is set to FALSE, the behavior aligns with the OCPP specification: the txProfile will be active until the end of the ongoing charging session even if the connection to the Central System is lost.

Reference to the Payment Session for a Payment Terminal Authorized Session in the idTag of StartTransaction

For chargers that have an integrated payment terminal, ABB supports a link between the charging session and the payment session.

This is implemented by putting a reference to the payment transaction into the field idTag of the StartTransaction request message preceded by a configurable prefix (e.g., "PAY:", "PT:").

The default value of the prefix is an empty String.

Example: For a payment terminal the authorized session idTag=PT:00001111.

The numbers that are communicated in the idTag of an OCPP StartTransaction message per type of supported payment terminals are the following:

- [CCV]: The "Authorization Number" as shown on the payment receipt,
- [Nayax]: The "Transaction Id" as shown on the Nayax web portal.

Reference to the Vehicle ID for an Autocharge Authorized Session in a Custom DataTransfer Message

For (CCS) charging sessions, in which the vehicle communicates its Vehicle ID to the charger, ABB support a link between the charging session and the corresponding Vehicle ID via a custom DataTransfer message regardless of the authorization method used for the session.

The DataTransfer message is sent by the charger as soon as the StartTransaction confirmation message has been received. The structure of the message is shown below:

```
"DataTransfer" {
  "data": "{
    \"VehicleID\":<Vehicle Id>,
    \"transactionID\":<Transaction Id>
  }",
  "messageId": "VehicleID",
  "vendorId": "com.abb.evci/vid/v1"
}
```

In case no Vehicle ID is received by the charger, the DataTransfer message is sent with the "VehicleID" parameter value set to empty String (i.e., "").

Sending the above described DataTransfer message can be enabled by setting the value of the configuration key SendVehicleID to TRUE.

ABB Implementation Choices for Certain OCPP 1.6 Unspecified Areas

Like any specification, OCPP 1.6 has some areas that are not explicitly specified, either deliberately or unintentionally. This section contains the description of how ABB interprets some of these areas and describes the respective implementation (or lack of such implementation) for these areas.

StopTransaction in Case of Power Supply Failure

OCPP does not specify how exactly the Charging Station should handle the situation of unexpected immediate power loss during an ongoing charging session when the charging session cannot be gracefully stopped. There is no explicit description of how StopTransaction must be sent and when it should be sent, as well as which StatusNotifications should be sent and in which order.

ABB agrees that it makes sense to at least properly finalize the charging session and at least send the StopTransaction message. Therefore, ABB has implemented this functionality in the power loss scenario right after the Charging Station has recovered and booted.

The exact sequence of messages sent by the Charging Station after it has booted is the following:

1. BootNotification request,
2. StatusNotification requests,
3. StopTransaction request (with the “reason” set to “PowerLoss”), as shown below:

```
"StopTransaction": {  
  "reason": "PowerLoss",  
  "idTag": <ID Tag>,  
  "transactionId": <Transaction Id>,  
  "meterStop": <Meter Stop Value>,  
  "timestamp": <Timestamp>  
}
```

StartTransaction and StopTransaction Timestamps

The OCPP StartTransaction and StopTransaction messages are triggered by certain events occurring in the electrical circuit of the Charging Station, as explained below.

- StartTransaction request: contactors are closed,
 - For CHAdeMO, this corresponds to the moment of setting D2 pin to High,
- StopTransaction request: contactors are opened.

Note: There is a slight time difference between the exact moment of when the event happens and the timestamp mentioned in the OCPP message due to communication latency among the software layers within the charger.

Date and Time Format

The fields in several types of OCPP messages include date- and time-related data, for example, the 'currentTime' field in the BootNotification confirmation message.

ABB chargers accept only the extended format of date and time data according to the ISO 8601 standard. This practically means that separators have to be used in the provided data (i.e., 2019-10-11T11:23:58+00:00) otherwise the relevant message is rejected by the charger.

Known Issues

RebootRequired status in ChangeConfiguration.conf

According to Chapter 5.3 "Change configuration" [1], "*If the change was applied successfully, but a reboot is needed to make it effective, the Charge Point SHALL respond with status 'RebootRequired'.*"

This type of response is not implemented in the software of ABB DC chargers. The only response implemented is "Accepted" or "Rejected".

However, the only parameter change that requires a reboot of the charging station is WebSocket-PingInterval. All the other parameters do not require a reboot and are applied immediately.

ABB plans to fix it in future versions of the software.

MeterValuesSampledData, MeterValuesAlignedData Keys and Other Supported CSL Configuration Keys that Accept Any String

According to the OCPP specification, the keys MeterValuesSampledData and MeterValuesAlignedData have the comma-separated list (CSL) type and may contain a combination of Measurands to be set for sending in MeterValues or StopTransaction messages.

Section 3.16.4 of the OCPP specifications, that was added in the last edition says "*If the comma separated list contains one or more Measurands that are not supported by this Charge Point, the Charge Point SHALL respond with: ChangeConfiguration.conf with: status = Rejected*".

ABB based its implementation on an earlier version of the OCPP document where certain aspects were not explicitly specified. ABB chose to have the charger parse CSL (Charging Station Language) with the best effort and applied the recognized Measurands. As a result, ultimately, any string will be accepted.

The same principle applies to every key of CSL type supported in the ABB implementation.

ABB agrees that specification 3.16.4 is clearer, but at the same time, it is stricter. Unfortunately, OCPP has no means to provide information to the Central System regarding why exactly the configuration change was rejected. ABB plans to address this and align it with version 3.16.4 in future software versions.

Configuration Keys

Supported Configuration Keys

Please see below which configuration keys are supported per OCPP feature profile.

Key Name	Required/Optional	Description	Type	Accessibility	Default Value	Value Range (for Type 'int')
Core profile						
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.	Boolean	RW	FALSE	N/A
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	TRUE	N/A
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.	Boolean	R	FALSE	N/A
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.	int	RW	0	0 - 3600

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).	int	RW	30	0 - 360
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	Unknown	N/A
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.	int	R	200	5 - 600
HeartbeatInterval	required	Interval of inactivity (no OCPP exchanges) with central system after which the Charge Point should send a Heartbeat.req PDU.	int	RW	240	0 - 86400
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	N/A
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	TRUE	N/A
MeterValuesAlignedData	required	Clock-aligned measurand(s) to be included in a MeterValues.req PDU, every Clock-AlignedDataInterval seconds.	CSL	RW	SoC, Energy.Active.Import.Register	N/A
MeterValuesSampledData	required	Sampled measurands to be included in a MeterValues.req PDU, every MeterValueSampleInterval seconds.	CSL	RW	SoC, Energy.Active.Import.Register	N/A

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.	int	RW	0	0 - 600
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.	int	RW	5	0 - 60
NumberOfConnectors	required	The number of physical charging connectors of this Charge Point.	int	R	Depends on charger model	0 - 10
ResetRetries	required	Number of times to retry an unsuccessful reset of the Charge Point. NOTE: this parameter is not being used, the charger will always reset on first attempt.	int	RW	5	0 - 15
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	N/A
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	SoC, Energy.Active.Import.Register	N/A
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	SoC, Energy.Active.Import.Register	N/A
SupportedFeatureProfiles	required	A list of supported Feature Profiles. Possible profile identifiers: Core, FirmwareManagement, LocalAuthListManagement, Reservation, SmartCharging and RemoteTrigger.	CSL	R	Core, LocalAuthListManagement, SmartCharging	N/A

TransactionMessageAttempts	required	How often the Charge Point should try to submit a transaction-related message when the Central System fails to process it.	int	RW	10	0 - 30
TransactionMessageRetryInterval	required	How long the Charge Point should wait before resubmitting a transaction-related message that the Central System failed to process.	int	RW	60	60 - 600
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.	int	RW	30	0 - 3600
Local Authorization List Management						
LocalAuthListEnabled	required	Whether the Local Authorization List is enabled.	Boolean	RW	TRUE	N/A
LocalAuthListMaxLength	required	Maximum number of identifications that can be stored in the Local Authorization List.	int	R	10000	5 - 1000000
SendLocalListMaxLength	required	Maximum number of identifications that can be sent in a single SendLocalList.req.	int	R	1000	5 - 1000000
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.	int	R	10	Any value
ChargingScheduleAllowedChargingRateUnit	required	A list of supported quantities for use in a ChargingSchedule. Allowed values: 'Current' and 'Power'.	CSL	R	Current, Power	N/A
ChargingScheduleMaxPeriods	required	Maximum number of periods that may be defined per ChargingSchedule.	int	R	24	Any value
MaxChargingProfilesInstalled	required	Maximum number of Charging profiles installed at a time.	int	R	10	Any value

Not Supported Configuration Keys

Following configuration keys are NOT SUPPORTED:

Key Name	Required/ Optional	Description	Type
Core profile			
BlinkRepeat	optional	Number of times to blink Charge Point lighting when signalling.	int
ConnectorPhaseRotationMaxLength	optional	Maximum number of items in a ConnectorPhaseRotation Configuration Key.	int
LightIntensity	optional	Percentage of maximum intensity at which to illuminate Charge Point lighting.	int
MaxEnergyOnInvalidId	optional	Maximum energy in Wh delivered when an identifier is invalidated by the Central System after start of a transaction.	int
StopTxnAlignedDataMaxLength	optional	Maximum number of items in a StopTxnAlignedData Configuration Key.	int
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
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. NOTE: not applicable for ABB high-power chargers, not implemented.	Boolean
SupportedFeaturesMaxLength	optional	Maximum number of items in a SupportedFeatures Configuration Key.	int
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

ABB Custom Configuration Keys

Key Name	Description	Type	Default Value
FreevendEnabled	Enables or disables "FreeMode".	Boolean	FALSE
FreevendIdTag	This tag is put in StartTransaction message in idTag field when authorization is disabled (i.e. FreevendEnabled value is set to false).	String[20]	'NOA'
DiscardTxProfileWhenConnectionLoss	When set to true and Webscoket is disconnected, TxProfile is deleted immediately without waiting for end of charging session.	Boolean	FALSE
SendVehicleID	When set to true, the Charge Point sends a DataTransfer message containing the Vehicle ID right after the StartTransaction confirmation has been received.	Boolean	FALSE

Security

Encryption

In addition to network level security ABB OCPP 1.6 implementation supports OCPP-J over TLS security. 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].

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.

Setting authorization key over OCPP after installation is not supported.
For more information on OCPP 1.6-J authentication please see chapter “6.2.2 Authentication” of [2].

Reference Documentation

[1] Open Charge Point Protocol 1.6

[2] Open Charge Point Protocol JSON 1.6, OCPP 1.6-J Specification