



OCPP 1.6 Errata sheet

v4.0 Release, 2019-10-23

Errata sheet: for OCPP version: 1.6 edition 2 and 1.6 FINAL.

Copyright © 2010 – 2019 Open Charge Alliance. All rights reserved.

This document is made available under the **Creative Commons Attribution-NoDerivatives 4.0 International Public License** (<https://creativecommons.org/licenses/by-nd/4.0/legalcode>).

Version History

VERSION	DATE	AUTHOR	DESCRIPTION
v4.0 Release	2019-10-23	Robert de Leeuw (ihomer) Franc Buve (ELaadNL)	4th release of the errata sheet New errata are marked with v4.0.
v3.0 Release	2017-09-08	Robert de Leeuw (IHomer)	3th release of the errata sheet New errata are marked with v3.0.
v2.0 Release	2017-03-27	Robert de Leeuw (IHomer) Klaas van Zuuren (ELaadNL) Brendan McMahon (ESB)	2nd release of the errata sheet Errata have been reordered to match the chronological order of the 1.6 specification.
v1.0 Release	2016-03-31	Robert de Leeuw (IHomer) Klaas van Zuuren (ELaadNL)	First release

1. Scope

This document contains errata on the OCPP 1.6 specification. Any errata added after v3.0 of this errata sheet is an errata on OCPP 1.6 edition 2, and is also applicable for OCPP 1.6 FINAL (the first edition) unless marked otherwise.

1.1. Terminology and Conventions

Bold: when needed to clarify differences, bold text might be used.

Since document version v3.0 errata are marked with a version number, indicating when an errata was added.

2. Major errata

Problems with the content/definition of the messages, class and enumerations of the protocol.

None known

3. Minor errata

Improvements to the descriptions on how the protocol (should) work.

3.1. Page 7, section: 2.2: More than IEC 15118 limits might be taken into account for Smart Charging.

The definition of "Composite Charging Schedule" states that "IEC 15118 limits might be taken into account", but also other limits might be taken into account.

Old text	Also IEC 15118 limits might be taken into account.
New text	Local Limits might be taken into account.

3.2. Page 10, section: 3.2: Feature Profiles should be normative

Added in errata sheet v4.0 The table with messages per feature profiles the columns have been mixed in edition 2.

Old text	New text
REMOTE TRIGGER	RESERVATION
RESERVATION	SMART CHARGING
SMART CHARGING	REMOTE TRIGGER

3.3. Page 10, section: 3.2: Feature Profiles should be normative

The Feature Profiles paragraph states that it is "informative", but it is "normative".

Old text	This section is informative .
New text	This section is normative .

3.4. Page 13, section: 3.5.1: Missing requirement, cache persists reset

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 16, section 3.4.1

In section 3.4.1 there is a requirement about the persistence of the cache, but it does not mention 'reset'

Old text	Cache values SHOULD be stored in non-volatile memory, and SHOULD be persisted across reboots and power outages.
New text	Cache values SHOULD be stored in non-volatile memory, and SHOULD be persisted across reboots , resets and power outages.

3.5. Page 15, section: 3.5.4: For unlock cable after invalid id in StartTransaction.conf, same identifier should be used.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 18, section 3.4.4

The text explaining what a Charge Point should do when a StartTransaction.conf is received is not 100% clear that it should be the same identifier, not an identifier of the same owner.

Old text	it SHOULD keep the Charging Cable locked until the owner presents his identifier.
New text	it SHOULD keep the Charging Cable locked until the same identifier (or an identifier with the same ParentIdTag) is used to start the transaction (StartTransaction.req) is presented.

3.6. Page 18, section 3.5: Better description how to determine start/end of Energy Transfer Period

Added in errata sheet v3.0

Description how the Central System can determine start/end of Energy Transfer Period can be improved.

Old text	A Central System MAY deduce the start and end of an Energy Transfer Period from the MeterValues that are sent during the Transaction.
New text	A Central System MAY deduce the start and end of an Energy Transfer Period from: the MeterValues that are sent during the Transaction, the status notifications: Charging, SuspendedEV and/or SuspendedEVSE. etc. Central System implementations need to take into account factors such as: Some EVs don't go to state SuspendedEV: they might continue to trickle charge. Some Charge Point don't even have a electrical meter.

3.7. Page 18, section 3.12.2: Add description of stacking without duration

Added in errata sheet v3.0

When using stacking for Smart Charging: a high stack level without a duration will cause lower profiles to be never executed.

Add note at end of section 3.12.2

New text	NOTE: If you use Stacking without a duration, on the highest stack level, the Charge Point will never fall back to a lower stack level profile."
----------	--

3.8. Page 20, section 3.13.1: Effect of updating or deleting TxDefaultProfile during a transaction not defined

Added in errata sheet v4.0

It was not clear that a new TxDefaultProfile not only applies to running transactions with a TxDefaultProfile, but also to running transactions without a ChargingProfile. Furthermore, removing a TxDefaultProfile will cause running transactions to continue without a TxDefaultProfile.

To clarify this best, two additional paragraphs are needed.

Page	Section	Additional text
20	3.13.1	<p>When an new or updated TxDefaultProfile is received and a transaction is ongoing that is not using a ChargingProfile or using the current TxDefaultProfile, the transaction SHALL continue, but switch to using the new or updated TxDefaultProfile.</p> <p>When TxDefaultProfile is removed, then running transactions, that started with that profile, SHALL continue without a TxDefaultProfile.</p>
54	5.16.3	<p>When the ongoing transaction is not using a ChargingProfile or using the TxDefaultProfile, and a new or updated TxDefaultProfile is received, the transaction SHALL continue, but switch to using the new or updated TxDefaultProfile.</p>

3.9. Page 29, section: 3.1.6: Missing advice to send meter register value if available

Added in errata sheet v4.0

When a Charge Point has a meter which allows the value of Meter Register (counter) to be read. It is seen as best to send this value instead of start every transaction at 0.

Additional text	If a Charge Point contains a Meter that has a register that can be read (instead of counting pulses or some other way of measuring the amount of energy etc.) It is RECOMMENDED to send this register value in every MeterValue send instead of starting at 0 for every transaction.
-----------------	--

3.10. Page 33, new Chapter about time notations

In addition to "3.13 Time Zones" to following should have been added about time notations:

New text	<p>3.14 Time notations</p> <p>Implementations MUST use ISO 8601 date time notation. Message receivers must be able to handle fractional seconds and time zone offsets (another implementation might use them). Message senders MAY save data usage by omitting insignificant fractions of seconds.</p>
----------	---

3.11. Page 35, section: 4.2: Boot Notification: Note on behaviour while not accepted by Central system

Old text	While not yet accepted by the Central System, the Charge Point may allow locally authorized transactions if it is configured to do so, as described in Local Authorization & Offline Behaviour. Parties who want to implement this behaviour must realize that it is uncertain if those transactions can ever be delivered to the Central System.
New text	<p>A Charge Point Operator MAY choose to configure a Charge Point to accept transactions before the Charge Point is accepted by a Central System. Parties who want choose to implement this behavior should realize that it is uncertain if those transactions can ever be delivered to the Central System.</p> <p>After a restart (for instance due to a remote reset command, power outage, firmware update, software error etc.) the Charge Point MUST again contact the Central System and SHALL send a BootNotification request. If the Charge Point fails to receive a BootNotification.conf from the Central System, and has no in-built non-volatile real-time clock hardware that has been correctly preset, the Charge Point may not have a valid date / time setting, making it impossible to later determine the date / time of transactions.</p> <p>It might also be the case (e.g. due to configuration error) that the Central System indicates a status other than Accepted for an extended period of time, or indefinitely.</p> <p>It is usually advisable to deny all charging services at a Charge Point if the Charge Point has never before been Accepted by the Central System (using the current connection settings, URL, etc.) since users cannot be authenticated and running transactions could conflict with provisioning processes.</p>

3.12. Page 37, section: 4.5: Relation between FirmwareStatusNotification.req and FirmwareUpdate.req is missing.

There is no description about the relation between FirmwareUpdate.req and FirmwareStatusNotification.req in the specification.

Additional text	The FirmwareStatusNotification.req PDUs SHALL be sent to keep the Central System updated with the status of the update process, started by the Central System with a FirmwareUpdate.req PDU.
-----------------	--

3.13. Page 38, section: 4.7: Missing description of configuration keys for MeterValues

There are a couple of required configuration keys that have a huge influence of how MeterValues work. But there is no reference or description for them in 4.7 Meter Values:

- ClockAlignedDataInterval
- MeterValuesAlignedData
- MeterValuesAlignedDataMaxLength
- MeterValuesSampledData
- MeterValuesSampledDataMaxLength
- MeterValueSampleInterval

- StopTxnAlignedData
- StopTxnAlignedDataMaxLength
- StopTxnSampledData
- StopTxnSampledDataMaxLength

New Chapter to be added to the specification:

3.14: Metering Data

This section is normative.

Extensive metering data relating to charging sessions can be recorded and transmitted in different ways depending on its intended purpose. There are two obvious use cases (but the use of meter values is not limited to these two):

- Charging Session Meter Values
- Clock-Aligned Meter Values

Both types of meter readings MAY be reported in standalone [MeterValues.req](#) messages (during a transaction) and/or as part of the *transactionData* element of the [StopTransaction.req](#) PDU.

3.14.1 Charging Session Meter Values

Frequent (e.g. 1-5 minute interval) meter readings taken and transmitted (usually in "real time") to the Central System, to allow it to provide information updates to the EV user (who is usually not at the charge point), via web, app, SMS, etc., as to the progress of the charging session. In OCPP, this is called "sampled meter data", as the exact frequency and time of readings is not very significant, as long as it is "frequent enough". "Sampled meter data" can be configured with the following configuration keys:

- MeterValuesSampledData
- MeterValuesSampledDataMaxLength
- MeterValueSampleInterval
- StopTxnSampledData
- StopTxnSampledDataMaxLength

MeterValueSampleInterval is the time (in seconds) between sampling of metering (or other) data, intended to be transmitted by "MeterValues" PDUs. 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.

MeterValuesSampledData is a comma separated list that prescribes the set of measurands to be included in a MeterValues.req PDU, every MeterValueSampleInterval seconds. The maximum amount of elements in the MeterValuesSampledData list can be reported by the Charge Point via: **MeterValuesSampledDataMaxLength**

StopTxnSampledData is a comma separated list that prescribes the sampled measurands to be included in the

TransactionData element of StopTransaction.req PDU, every MeterValueSampleInterval seconds from the start of the charging session. The maximum amount of elements in the StopTxnSampledData list can be reported by the Charge Point via: **StopTxnSampledDataMaxLength**

3.14.2 Clock-Aligned Meter Values

Grid Operator might require meter readings to be taken from fiscally certified energy meters, at specific Clock aligned times (usually every quarter hour, or half hour).

"Clock-Aligned Billing Data" can be configured with the following configuration keys:

- ClockAlignedDataInterval
- MeterValuesAlignedData
- MeterValuesAlignedDataMaxLength
- StopTxnAlignedData
- StopTxnAlignedDataMaxLength

ClockAlignedDataInterval is the size of the clock-aligned data interval (in seconds). This defines the set of evenly spaced meter data 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.

A value of "0" (numeric zero), by convention, is to be interpreted to mean that no clock-aligned data should be transmitted.

MeterValuesAlignedData is a comma separated list that prescribes the set of measurands to be included in a MeterValues.req PDU, every ClockAlignedDataInterval seconds. The maximum amount of elements in the MeterValuesAlignedData list can be reported by the Charge Point via: **MeterValuesAlignedDataMaxLength**

StopTxnAlignedData is a comma separated list that prescribes the set of clock-aligned periodic measurands to be included in the TransactionData element of StopTransaction.req PDU for every ClockAlignedDataInterval of the charging session. The maximum amount of elements in the StopTxnAlignedData list can be reported by the Charge Point via: **StopTxnAlignedDataMaxLength**

3.14.3 Multiple Locations/Phases

When a Charge Point can measure the same measurand on multiple locations or phases, all possible locations and/or phases SHALL be reported when configured in one of the relevant configuration keys.

For example: A Charge Point capable of measuring *Current.Import* on *Inlet* (all 3 phases) (grid connection) and *Outlet* (3 phases per connector on both its connectors). *Current.Import* is set in MeterValuesSampledData. MeterValueSampleInterval is set to 300 (seconds). Then the Charge Point should send:

- a MeterValue.req with: connectorId = 0; with 3 SampledValue elements, one per phase with location = Inlet.
- a MeterValue.req with: connectorId = 1; with 3 SampledValue elements, one per phase with location = Outlet.

- a MeterValue.req with: connectorId = 2; with 3 SampledValue elements, one per phase with location = Outlet.

3.14.4 Unsupported measurands

When a Central System sends a ChangeConfiguration.req to a Charge Point with one of the following configuration keys:

- MeterValuesAlignedData
- MeterValuesSampledData
- StopTxnAlignedData
- StopTxnSampledData

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. No changes SHALL be made to the currently configuration.

Added in errata sheet v3.0

3.14.5 No metering data in a Stop Transaction

When the configuration keys: **StopTxnAlignedData** and **StopTxnSampledData** are set to an empty string, the Charge Point SHALL NOT put meter values in a StopTransaction.req PDU.

3.14. Page 31, section: 3.16.5: Missing description of Start and Stop MeterValues in StopTransaction.req

Added in errata sheet v4.0

Additional Section should have been added to 3.16:

3.16.6 Start and Stop MeterValues in StopTransaction.req

When the Charge Point is configured to provide metering data in the StopTransaction.req, the Charge Point is RECOMMENDED to always provide the start (Transaction.Begin) and stop (Transaction.End) values of every measurand configured.

If the Charge Point has to drop meter values because it is running out of memory, it is RECOMMENDED not to drop the start and stop values.

3.15. Page 36, section: 4.7: make it more explicit that MeterValues are required if they are configured

Added in errata sheet v4.0

The first paragraph of section 4.7 reads:

"A Charge Point MAY sample the electrical meter or other sensor/transducer hardware to provide extra information about its meter values. It is up to the Charge Point to decide when it will send meter values. This can be configured using the ChangeConfiguration.req message to data acquisition intervals and specify data to be

acquired & reported."

The keyword "MAY" in this text suggests that it is optional to send MeterValues. However, when the value of the configuration variable MeterValueSampleInterval has been set to a value greater than zero, then it is mandatory to send MeterValues.

Additional text After first section	When the value of the configuration variable MeterValueSampleInterval and/or ClockAlignedDataInterval has been set to a value greater than zero, the Charge Point SHALL send MeterValues at the given interval(s).
--	--

3.16. Page 36, section: 4.7: Not 100% clear which transaction ID to use in a MeterValue (multiple transactions during interval)

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 38

It was not 100% clear which transactionId to put a MeterValue.req message for a Clock-Aligned meter value sample when during the last period between samples, the previous transaction has ended and a new transaction has started.

Additional text Bullet 2	The transactionId is always the ID of the Transaction that is ongoing (on that connector) at the moment the meter value sample is taken. When the TransactionId is set, only MeterValues related to that transaction can be reported. If the values are taken from the main energy meter, it is advised NOT to add a transactionId. When reporting Meter Values for connectorId 0 (the main energy meter) it is RECOMMENDED NOT to add a TransactionId.
-----------------------------	--

3.17. Page 37, section: 4.8: Unclear how to handle: StartTransaction.conf with idTagInfo not Accepted.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 40

It is not (clearly) defined what a Charge Point should do when it receives a StartTransaction.conf with authorization status other then Accepted, when it is online.

There is already text that describes what a Charge Point should do, but this text is now in the part explaining what to do when the Charge Point is offline. But because it is in that part, it is not clear that it is also they way a Charge Point should handle the same situation when it is online.

The following text should be moved from: Page 15, section: 3.5.4 (OCPP 1.6 FINAL Page 18, section 3.4.4) to: Page 37, section 4.8 (OCPP 1.6 FINAL Page 40)

"When the authorization status in the StartTransaction.conf is not Accepted, and the transaction is still ongoing, the Charge Point SHOULD:

- when StopTransactionOnInvalidId is set to true: stop the transaction normally as stated in Stop Transaction. The Reason field in the Stop Transaction request should be set to DeAuthorized. If the Charge Point has the possibility to lock the Charging Cable, it SHOULD keep the Charging Cable locked until the owner presents his identifier.

- when StopTransactionOnInvalidId is set to false: only stop energy delivery to the vehicle.

Note: In the case of an invalid identifier, an operator MAY choose to charge the EV with a minimum amount of energy so the EV is able to drive away. This amount is controlled by the optional configuration key: MaxEnergyOnInvalidId."

3.18. Page 37, section: 4.8: How to deliver Transaction related messages when no transactionId is known.

Added in errata sheet v4.0

This section mentions at the end, that failing to respond with a StartTransaction.conf will cause the Charge Point to retry several times. However, it is not specified how to deal with Transaction related messages for which no transactionId is known, when this happens.

At the end of section 4.8 add the following text:

Additional text	<p>If the Charge Point was unable to deliver the StartTransaction.req despite repeated attempts, or if the Central System was unable to deliver the StartTransaction.conf response, then the Charge Point will not receive a transactionId.</p> <p>In that case, the Charge Point SHALL send any Transaction related messages for this transaction to the Central System with a transactionId = -1. The Central System SHALL respond as if these messages refer to a valid transactionId, so that the Charge Point is not blocked by this.</p>
-----------------	--

3.19. Page 38, section: 4.9: Not defined that StatusNotification should always be send.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 40

The specification on page 38 states: "The following table depicts changes from a previous status (left column) to a new status (upper row) upon which a Charge Point MAY send a StatusNotification.req PDU to the Central System."

But the idea has always been that the Charge Point has to send a StatusNotification when the state changes.

Old text	a Charge Point MAY send a StatusNotification.req PDU to the Central System.
New text	a Charge Point SHALL send a StatusNotification.req PDU to the Central System, taking into account some minimum status duration.

3.20. Page 41, section: 4.9: Status transition from Preparing to Finishing (B6) is possible

Added in errata sheet v3.0

The original 1.6 does not allow a transition from Preparing to Finishing (B6). But there is a use case where this is

possible:

A Charge Point with 2 connectors and 1 RFID reader. Driver 1 connects his charging cable. State goes to "Preparing" User does not authorize, times out. Then the Charge Point has to go to state: "Finishing". It should NOT go to "Available". Because if the charging cable remains plugged in and another EV driver swipes his RFID (before plugging in), that would start a transaction on the already plugged in cable. By going to "Finishing" this is prevented.

Table top page 41	Add B6
Table on page 42	B6: Timed out. Usage was initiated (e.g. insert plug, bay occupancy detection), but idTag not presented within timeout.

3.21. Page 42, section: 4.9 (B1): Missing reference to configuration key: "ConnectionTimeOut"

The configuration key: "ConnectionTimeOut" is never referenced in the specification, section: 4.9 needs a note explaining how the state: "Preparing" and the configuration key: "ConnectionTimeOut" work together.

Old text	Intended usage is ended (e.g. plug removed, bay no longer occupied, second presentation of idTag, time out on expected user action)
New text	Intended usage is ended (e.g. plug removed, bay no longer occupied, second presentation of idTag, time out (configured by the configuration key: ConnectionTimeOut) on expected user action)

3.22. Page 43, section: 4.9: Not defined that StatusNotification should always be send after BootNotification accepted.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 45

There is no description of behaviour that is expected and is implemented by every known implementation of OCPP: After being accepted via BootNotification.conf(Accepted) the Charge Point has to report its current status.

Additional text	After the Central System accept a Charge Point by sending a BootNotification.conf with a status Accepted, the Charge Point SHALL send a StatusNotification.req PDU for connectorId 0 and all connectors with the current status.
-----------------	--

3.23. Page 46, section: 5.4. Clear Cache when Cache is not implemented not defined.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 49

In OCPP 1.6, the Cache is not required, but the message: ClearCache.req is required to be implemented. OCPP does not define what the expected behaviour is.

Additional text	When the Authorization Cache is not implemented and the Charge Point receives a ClearCache.req message. The Charge Point SHALL response with ClearCache.conf with the status: Rejected.
-----------------	---

3.24. Page 46, section 5.8: Missing requirement about chargingRateUnit in GetCompositeSchedule

Added in errata sheet v4.0

The following note must be added at the end of this section:



If an invalid value is used for `chargingRateUnit` in `GetCompositeSchedule.req`, e.g. when using a value for `chargingRateUnit` that is not 'A' or 'W', then Charge Point SHALL respond with RPC Framework CALLERROR: PropertyConstraintViolation (JSON) or SOAP Fault: Sender, ProtocolError (SOAP).

3.25. Page 47, section 5.5: Unclear how to match fields in a ClearChargingProfile request

Added in errata sheet v4.0

The following note must be added at the end of this section:



If no fields are specified in the `ClearChargingProfile.req` message, then Charge Point SHALL clear all `ChargingProfiles`. If one or more fields are specified in the `ClearChargingProfile.req` message, then Charge Point SHALL clear all `ChargingProfiles` that match (logical AND) all of the provided fields.

3.26. Page 63, section 6.13: Unclear how to match fields in a ClearChargingProfile request

Added in errata sheet v4.0

The following sentence in the description of `ClearChargingProfile.req` must be replaced with new text:

Old text	The Central System can use this message to clear (remove) either a specific charging profile (denoted by id) or a selection of charging profiles that match with the values of the optional <code>connectorId</code> , <code>stackLevel</code> and <code>chargingProfilePurpose</code> fields.
New text	<p>The Central System can use this message to clear (remove) either a specific charging profile (denoted by id) or a selection of charging profiles that match (logical AND) with the values of the optional <code>connectorId</code>, <code>stackLevel</code> and <code>chargingProfilePurpose</code> fields.</p> <p>If no fields are provided, then all charging profiles will be cleared.</p> <p>If id is specified, then all other fields are ignored.</p>

3.27. Page 50, section: 5.7: More than IEC 15118 might be taken into account for Smart Charging.

It is stated that "IEC 15118 limits might be taken into account", but also other limits might be taken into account.

Old text	Also IEC 15118 limits might be taken into account.
----------	--

New text	Local Limits might be taken into account.
----------	---

3.28. Page 51, section: 5.7: Get Composite Schedule: First sentences is not clear

In the description of Get Composite Schedule, it is not clear, what are the start and end points in time of the schedule that is to be sent.

Old text	Upon receipt of a GetCompositeSchedule.req, the Charge Point SHALL calculate the scheduled time intervals up to the Duration is met and send them to the central system.
New text	Upon receipt of a GetCompositeSchedule.req, the Charge Point SHALL calculate the Composite Charging Schedule intervals, from the moment the request PDU is received: Time X, up to X + Duration, and send them in the GetCompositeSchedule.conf PDU to the central system.

3.29. Page 51, section: 5.7: Get Composite Schedule: First sentences is not clear

Added in errata sheet v4.0

The description of Get Composite Schedule, can be clarified a bit more.

Old text	Upon receipt of a GetCompositeSchedule.req, the Charge Point SHALL calculate the Composite Charging Schedule intervals, from the moment the request PDU is received: Time X, up to X + Duration, and send them in the GetCompositeSchedule.conf PDU to the central system.
New text	Upon receipt of a GetCompositeSchedule.req, the Charge Point SHALL calculate the scheduled time intervals from the moment of message receipt up to the Duration (in seconds) and send them to the central system.

3.30. Page 51, section: 5.7: Improve use of connectorId '0' in GetCompositeSchedule

In the description of the use of connectorId '0' in GetCompositeSchedule it is not clear that it can also mean the Charge Point reports current instead of power.

Old text	If the ConnectorId in the request is set to '0', the Charge Point SHALL report the total expected energy flow of the Charge Point for the requested time period.
New text	If the ConnectorId in the request is set to '0', the Charge Point SHALL report the total expected power or current the Charge Point expects to consume from the grid during the requested time period.

3.31. Page 51, section: 5.9: Relation between GetDiagnostics.req and DiagnosticsStatusNotification.req is missing.

There is no description about the relation between GetDiagnostics.req and DiagnosticsStatusNotification.req in the specification.

The following diagram should replace the diagram in 5.9.

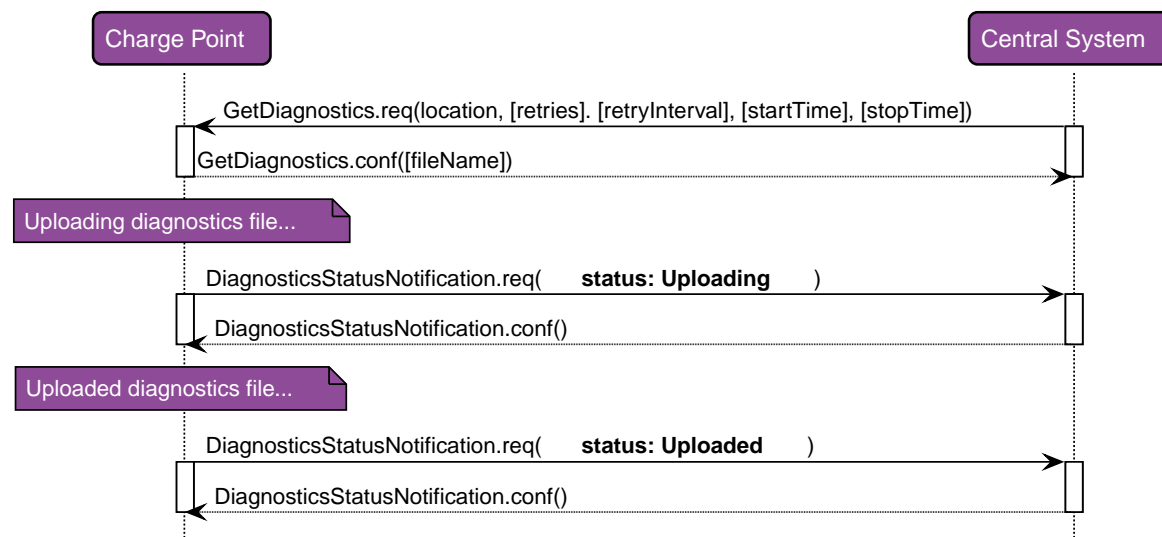


Figure 1. Sequence Diagram: get diagnostics

Additional text	During uploading of a diagnostics file, the Charge Point MUST send DiagnosticsStatusNotification.req PDUs to keep the Central System updated with the status of the upload process.
-----------------	---

3.32. Page 52, section: 5.11. Remote Start Transaction: .conf status is accepted request

Old text	Central System can request a Charge Point to start a transaction by sending a RemoteStartTransaction.req. Upon receipt, the Charge Point SHALL reply with RemoteStartTransaction.conf and a status indicating whether it is able to start a transaction or not .
New text	Central System can request a Charge Point to start a transaction by sending a RemoteStartTransaction.req. Upon receipt, the Charge Point SHALL reply with RemoteStartTransaction.conf and a status indicating whether it has accepted the request and will attempt to start a transaction.

3.33. page 54, section: 5.12. Remote Stop Transaction: .conf status is accepted request

Old text	RemoteStopTransaction.req to Charge Point with the identifier of the transaction. Charge Point SHALL reply with RemoteStopTransaction.conf to indicate whether it is indeed able to stop the transaction .
New text	RemoteStopTransaction.req to Charge Point with the identifier of the transaction. Charge Point SHALL reply with RemoteStopTransaction.conf and a status indicating whether it has accepted the request and a transaction with the given transactionId is ongoing and will be stopped .

3.34. page 54, section: 5.16. Not defined how a Charge Point should process a Charging Profile with more phases then being used.

Added in errata sheet v4.0

Additional Text	When the Charge Point receives a Charging Profile with 'numberPhases' higher then the currently used amount of phases or maximum amount of phases this Charge Point can use, The Charge Point SHALL use the 'limit' as is given, for the amount of phases that are/can be used for charging. The Charge Point SHALL NOT calculate a new 'limit' based on the amount of phases possible and given in the 'numberPhases' field.
-----------------	---

3.35. page 55, section: 5.17. TriggerMessage(BootNotification) after being accepted may be rejected

Added in errata sheet v4.0

The following scenario is not defined in OCPP 1.6: "When the Charge Point is accepted by the Central System and the Central System sends a TriggerMessage for requesting a BootNotification. The Charge Point will accept this TriggerMessage and sends a BootNotification.req to the Central System. What should be the behavior of the Charge Point when the Central System is responding with the status Pending or Rejected?"

As this can cause a lot of confusing situations, what to do with ongoing transaction etc. It is allowed to reject such a TriggerMessage.req.

Additional text	After the Charge Point has received a BootNotification.conf(Accepted), until the next reset/reboot/reconnect, the Charge Point is RECOMMENDED to Reject a TriggerMessage request for BootNotification.
-----------------	--

3.36. Page 56, section: 5.14: Improved description of Soft/Hard Reset

Added in errata sheet v3.0

The descriptions of Soft/Hard reset can be improved, not very clear what is the difference between the two.

Additional text, between par 1 & 2	After receipt of a Reset.req, The Charge Point SHALL send a StopTransaction.req for any ongoing transaction before performing the reset. If the Charge Point fails to receive a StopTransaction.conf from the Central System, it SHALL queue the StopTransaction.req.
---------------------------------------	---

Old text	At receipt of a soft reset, the Charge Point SHALL return to a state that behaves as just having been booted. If any transaction is in progress it SHALL be terminated normally, before the reset, as in Stop Transaction.
New text	At receipt of a soft reset request, the Charge Point SHALL stop ongoing transactions gracefully and send StopTransaction.req for every ongoing transaction. It should then restart the application software (if possible, otherwise restart the processor/controller).

Old text	At receipt of a hard reset the Charge Point SHALL attempt to terminate any transaction in progress normally as in StopTransaction and then perform a reboot.
New text	At receipt of a hard reset request, the Charge Point SHALL restart (all) the hardware, it is not required to gracefully stop ongoing transaction. If possible the Charge Point sends a StopTransaction.req for previously ongoing transactions after having restarted and having been accepted by the Central System via a BootNotification.conf. This is a last resort solution for a not correctly functioning Charge Point, by sending a "hard" reset, (queued) information might get lost.

3.37. Page 53, section: 5.16: Unclear ChargingProfiles should persist reboots

Added in errata sheet v4.0

There is no text that explain that it is expected of a Charge Point to store ChargingProfiles in persistent memory.

Additional text	ChargingProfiles set via SetChargingProfile.req SHALL be persistent across reboots/power cycles.
-----------------	--

3.38. Page 55, section 5.16: Missing requirement about ChargingSchedule's ChargingRateUnit

Added in errata sheet v4.0

The following note must be added at the end of this section:



If an invalid value is used for an enumeration in ChargingProfile or ChargingSchedule, e.g. when using a value for `chargingRateUnit` that is not 'A' or 'W', then Charge Point SHALL respond with RPC Framework CALLERROR: PropertyConstraintViolation (JSON) or SOAP Fault: Sender, ProtocolError (SOAP).

3.39. Page 55, section: 5.16: SetChargingProfile may be rejected

Added in errata sheet v4.0

The following notes must be added at the end of this section:



If an invalid value for `connectorId` is used in SetChargingProfile.req, then the Charge Point SHALL respond with RPC Framework CALLERROR: PropertyConstraintViolation (JSON) or SOAP Fault: Sender, ProtocolError (SOAP).



If the Charge Point does not support smart charging, then it SHALL respond with RPC Framework CALLERROR: NotSupported (JSON) or SOAP Fault: Receiver, NotSupported (SOAP).

3.40. Page 56, section: 5.18: TriggerMessage(BootNotification) not allowed after BootNotificationResponse(Accepted)

Added in errata sheet v4.0

Add the following note at the end of this section:



Once a CSMS has sent a BootNotification.conf message with status `registrationStatus = Accepted` to the Charge Point, then CSMS SHALL not send a TriggerMessage to request for a new BootNotification until the Charge Point sends a BootNotification.req message.

3.41. Page 58, section: 5.16.2: RemoteStart with ChargingProfile: TransactionId should not be set

It is not clear that in a remoteStartTransaction.req with ChargingProfile, the transactionId should not be set.

Old text	If the Central System includes a ChargingProfile, the ChargingProfilePurpose MUST be set to TxProfile.
New text	If the Central System includes a ChargingProfile, the ChargingProfilePurpose MUST be set to TxProfile and the transactionId SHALL NOT be set.

3.42. Page 58, section: 5.16.2: Meaning of note on RemoteStart with ChargingProfile is not clear

In the description of RemoteStartTransaction with a ChargingProfile there is a note, but the meaning of the note is not clear.

Old text	The Charge Point SHOULD add the TransactionId to the received profile once the transaction is reported to the central system.
New text	<p>The Charge Point SHALL apply the given profile to the newly started transaction. This transaction will get a transactionId assigned by Central System via a startTransaction.conf.</p> <p>When the Charge Point receives a setChargingProfile.req, with the transactionId for this transaction, with the same StackLevel as the profile given in the remoteStartTransaction.req, the Charge Point SHALL replace the existing charging profile, otherwise it SHALL install/stack the profile next to the already existing profile(s).</p>

3.43. Page 58, Section 5.16.4: Smart Charging fall back to default unclear

Added in errata sheet v3.0

Old text	When recurrencyKind is used in combination with a chargingSchedule duration shorter than the recurrencyKind period, the Charge Point SHALL fall back to default behavior after the chargingSchedule duration ends.
New text	When recurrencyKind is used in combination with a chargingSchedule duration shorter than the recurrencyKind period, the Charge Point SHALL fall back to default behavior after the chargingSchedule duration ends. This fall back means that the Charge Point SHALL use a ChargingProfile with a lower stackLevel if available. If no other ChargingProfile is available, the Charge Point SHALL allow charging as if no ChargingProfile is installed. If the chargingSchedulePeriod and/or duration is longer than the recurrencyKind period, the remainder periods SHALL NOT be executed.

3.44. Page 58, Section 5.16.4: Not defined what to do with a charging schedule period longer then recurrence.

Added in errata sheet v3.0

It is not defined what to do with a chargingSchedulePeriod and/or duration that is longer than the recurrence period.

Add the following note after the NOTE about "chargingSchedulePeriod longer than duration"

New text	NOTE: When recurrencyKind is used in combination with a chargingSchedulePeriod and/or duration that is longer than the recurrence period duration, the remainder periods SHALL NOT be executed.
----------	---

3.45. Page 58, Section 5.16.4: First ChargingSchedulePeriod should start with StartSchedule = 0

Added in errata sheet v3.0

The obvious is not defined: The first ChargingSchedulePeriod StartSchedule in a ChargingSchedule should be 0.

Add the following note after the NOTE about "chargingSchedulePeriod longer than duration"

New text	NOTE: The startPeriod of the first ChargingSchedulePeriod in a ChargingSchedule SHALL always be 0.
----------	--

3.46. Page 60, section: 5.17: Description TriggerMessage for MeterValues not clear

The description of what a Charge Point should do when it receives a TriggerMessage.req PDU with requestedMessage: MeterValues, is not clear.

Old text	A MeterValues message triggered in this way for instance SHOULD return the most recent measurements for all measurands configured in configuration key MeterValuesSampledData.
New text	A MeterValues message triggered in this way for instance SHALL return the most recent measurements for all measurands configured in configuration key MeterValuesSampledData.

3.47. Page 61, section: 5.19: Relation between FirmwareUpdate.req and FirmwareStatusNotification.req is missing.

There is no description about the relation between FirmwareUpdate.req and FirmwareStatusNotification.req in the specification. In paragraph 3.3 on page 15, there is a more elaborate diagram of firmware update, including the relationship between UpdateFirmware.req and FirmwareStatusNotification.req, but that paragraph is informative.

The following diagram should replace the diagram in 5.19.

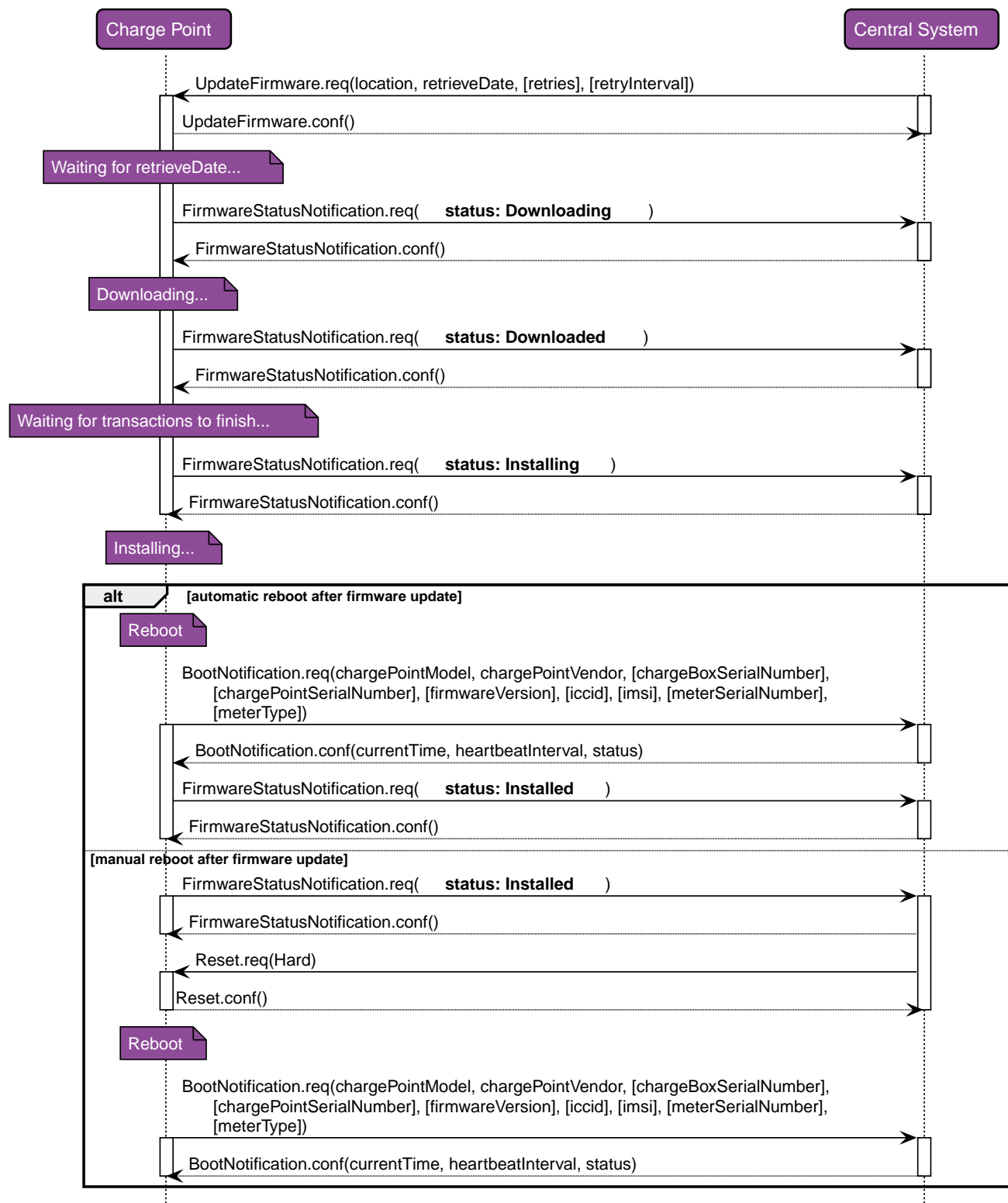


Figure 2. Sequence Diagram: firmware update

Additional text	<p>During downloading and installation of the firmware, the Charge Point MUST send FirmwareStatusNotification.req PDUs to keep the Central System updated with the status of the update process.</p> <p>The sequence diagram above is an example. It is good practice to first reboot the Charge Point to check the new firmware is booting and able to connect to the Central System, before sending the status: Installed. It is not a requirement.</p>
-----------------	--

3.48. Page 61, section: 5.19: No description new firmware should be installed.

There is no requirement on the installation of new firmware.

Additional text	The Charge Point SHALL, if the new firmware image is "valid", install the new firmware as soon as it is able to.
-----------------	--

3.49. page 61, section: 5.19. Firmware installation during charging session

It is advised to not stop charging session to install new firmware, but wait until session has ended.

Additional text	If it is not possible to continue charging during installation of firmware, it is RECOMMENDED to wait until Charging Session has ended (Charge Point idle) before commencing installation. It is RECOMMENDED to set connectors that are not in use to UNAVAILABLE while the Charge Point waits for the Session to end.
-----------------	--

This errata note is also applicable for OCPP versions 1.2 and 1.5.

3.50. Page 66, section: 6.25: Description of retries incorrect

Added in errata sheet v4.0

In the description of the field **retries** the word "try" should be "retry".

Old text	Optional. This specifies how many times Charge Point must try to upload the diagnostics before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
New text	Optional. This specifies how many times Charge Point must retry to upload the diagnostics before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.

3.51. Page 70, section: 6.22: Unclear when to use the fields: scheduleStart and chargingSchedule in GetCompositeSchedule.conf.

The message: GetCompositeSchedule.conf contains 3 optional fields, for two of these fields: "scheduleStart" and "chargingSchedule" it is not clear when they should or should not be used:

Add to the description the field: scheduleStart	If status is "Rejected", this field may be absent.
Add to the description the field: chargingSchedule	If status is "Rejected", this field may be absent.

3.52. Page 70, section: 6.41: SendLocalList list version should never be 0

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 77

In GetLocalListVersion.conf listVersion = 0 and -1 have a special meaning, so they should not be used in SetLocalList.req

Old text	Required. In case of a full update this is the version number of the full list. In case of a differential update it is the version number of the list after the update has been applied.
----------	--

New text	Required. In case of a full update this is the version number of the full list. In case of a differential update it is the version number of the list after the update has been applied. SHALL NOT be -1 or 0 as these have a special meaning in GetLocalListVersion.conf
----------	--

3.53. Page 70, section: 6.43: Description connectorId = 0 in SetChargingProfile.req causes confusion

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 78

The description of the connectorId in the SetChargingProfile.req does not take into account that it can also be used for setting a TxDefaultProfile. It seems to only take into account a ChargePointMaxProfile.

Old text	If connectorId = 0, the message contains an overall limit for the Charge Point.
New text	If connectorId = 0, and the message contains a ChargePointMaxProfile it contains an overall limit for the Charge Point. If connectorId = 0, and the message contains a TxDefaultProfile it contains limits that are to be used for any new transaction on any connector of that Charge Point.

3.54. Page 71, section: 6.46: Useful values for Transaction ID not clear

Added in errata sheet v4.0

It is not clear to some that the expected value for Transaction IDs are unique positive integers.

Old description	Required. This contains the transaction id supplied by the Central System.
New description	Required. This contains the transaction id supplied by the Central System. It is RECOMMENDED to use unique positive numbers for transactionIds. Negative numbers and zero (0) are possible, but don't have any special meaning to the Charge Point (they don't mean the transaction is rejected or something like that.) Note that the Charge Point might use transactionId = -1 in transaction related message when the Charge Point was not able to successfully deliver the StartTransaction.req.

3.55. Page 77, section: 7.7: Definition of SuspendedEV is too confusing

Added in errata sheet v4.0

The new text for SuspendedEV is too confusing. The way it is written now implies that, even if the EVSE is in charging state (C2 with contactor closed) and the EV is not consuming any power that the state should be SuspendedEV.

SuspendedEV	Old text	When the EV is connected to the EVSE and the EVSE is offering energy but the EV is not taking any energy. (Operative)
SuspendedEV	New text	When the EV is connected to the EVSE and the EVSE is willing and ready to offer energy, but EV is not asking or taking energy. For example: A Charge Point using mode 3: if the EV is in B2 this also means SuspendedEV. (Operative)

3.56. Page 79, section: 7.8: recurrencyKind only to be used with chargingProfileKind: Recurring

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 90

There is no description/explanation that the field "recurrencyKind" should only be used when "chargingProfileKind" is "Recurring"

Old description	Optional. Indicates the start point of a recurrence.
New description	Optional. Indicates the start point of a recurrence. SHALL only be used when the field: recurrencyKind is set to: Recurring.

3.57. Page 80, section: 7.13: GetCompositeScheduleResponse unclear how/when to use which start field

Added in errata sheet v4.0

The description of the field **startSchedule** must be replaced by the following.

Old text	Optional. Starting point of an absolute schedule. If absent the schedule will be relative to start of charging.
New text	Optional. Starting point of an absolute schedule. If absent the schedule will be relative to start of charging. When ChargingSchedule is used as part of a GetCompositeSchedule.conf message, then this field must be omitted.

3.58. Page 81, section: 7.15: Improved definition of string types: CiString20Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 94

Description of: CiString20Type can now be read as: "A String that has to be exactly 20 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 20 characters.
New text	A case insensitive string with a maximum length of 20 characters.

3.59. Page 81, section: 7.16: Improved definition of string types: CiString25Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 94

Description of: CiString25Type can now be read as: "A String that has to be exactly 25 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 25 characters.
New text	A case insensitive string with a maximum length of 25 characters.

3.60. Page 82, section: 7.17: Improved definition of string types: CiString50Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 94

Description of: CiString50Type can now be read as: "A String that has to be exactly 50 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 50 characters.
New text	A case insensitive string with a maximum length of 50 characters.

3.61. Page 82, section: 7.18: Improved definition of string types: CiString255Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 95

Description of: CiString255Type can now be read as: "A String that has to be exactly 255 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 255 characters.
New text	A case insensitive string with a maximum length of 255 characters.

3.62. Page 82, section: 7.19: Improved definition of string types: CiString500Type.

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 95

Description of: CiString500Type can now be read as: "A String that has to be exactly 500 characters long." This is a maximum length, not a required length.

Old text	Generic used case insensitive string of 500 characters.
New text	A case insensitive string with a maximum length of 500 characters.

3.63. Page 84, section: 6.55: Description for UpdateFirmware.req: retrieveDate is ambiguous

The description of the field: "retrieveDate" is ambiguous. It should not be "must", but: "is allowed to"

Old text	Required. This contains the date and time after which the Charge Point must retrieve the (new) firmware.
New text	Required. This contains the date and time after which the Charge Point is allowed to retrieve the (new) firmware.

3.64. Page 88, section: 7.7: Description SuspendedEVSE and SuspendedEV too strict, not all chargers have a contactor

Descriptions for SuspendedEVSE and SuspendedEV seems to imply that an EVSE has a contactor, but that is not always the case, for example: wireless charging.

SuspendedEVSE	Old text	When the contactor of a Connector opens upon request of the EVSE, e.g. due to a smart charging restriction or as the result of StartTransaction.conf indicating that charging is not allowed (Operative)
SuspendedEVSE	New text	When the EV is connected to the EVSE but the EVSE is not offering energy to the EV, e.g. due to a smart charging restriction, local supply power constraints, or as the result of StartTransaction.conf indicating that charging is not allowed etc. (Operative)
SuspendedEV	Old text	When the EVSE is ready to deliver energy but contactor is open, e.g. the EV is not ready.
SuspendedEV	New text	When the EV is connected to the EVSE and the EVSE is offering energy but the EV is not taking any energy. (Operative)

3.65. Page 90, section: 7.8: validFrom fields are allowed for TxProfiles.

In the class definition of ChargingProfile, the field: "validFrom", is defined as: "Not to be used when ChargingProfilePurpose is TxProfile." The specification denotes that the field ValidTo and ValidFrom are not to be used in combination with profiletype TxProfile. This note should have been deleted in the final version. With the decision to support stacking in combination with ProfileType TxProfile, the use of ValidFrom and ValidTo fields is unavoidable, since otherwise the profile with the highest StackLevel will be active until it is uninstalled.

Old text	Optional. Point in time at which the profile starts to be valid. If absent, the profile is valid as soon as it is received by the Charge Point. Not to be used when ChargingProfilePurpose is TxProfile.
New text	Optional. Point in time at which the profile starts to be valid. If absent, the profile is valid as soon as it is received by the Charge Point.

3.66. Page 91, section: 7.8: validTo fields are allowed for TxProfiles.

In the class definition of ChargingProfile, the field: "validTo", is defined as: "Not to be used when ChargingProfilePurpose is TxProfile." The specification denotes that the field ValidTo and ValidFrom are not to be used in combination with profiletype TxProfile. This note should have been deleted in the final version. With the decision to support stacking in combination with ProfileType TxProfile, the use of ValidFrom and ValidTo fields is unavoidable, since otherwise the profile with the highest profile will be active until it is uninstalled.

Old text	Optional. Point in time at which the profile stops to be valid. If absent, the profile is valid until it is replaced by another profile. Not to be used when ChargingProfilePurpose is TxProfile.
----------	--

New text	Optional. Point in time at which the profile stops to be valid. If absent, the profile is valid until it is replaced by another profile.
----------	--

3.67. Page 91, section: 7.10: Description of TxProfile/TxDefaultProfile in ChargingProfilePurposeType in relation with RemoteStartTransaction unclear

It is not completely clear what the correct ProfilePurpose should be in a remoteStartTransaction.req.

TxDefaultProfile	Old text	Default profile to be used for new transactions.
TxDefaultProfile	New text	Default profile that can be configured in the Charge Point. When a new transaction is started, this profile SHALL be used, unless it was a transaction that was started by a remoteStartTransaction.req with a ChargingProfile that is accepted by the Charge Point.
TxProfile	Old text	Profile with constraints to be imposed by the Charge Point on the current transaction. A profile with this purpose SHALL cease to be valid when the transaction terminates.
TxProfile	New text	Profile with constraints to be imposed by the Charge Point on the current transaction , or on a new transaction when this is started via a RemoteStartTransaction.req with a ChargingProfile. A profile with this purpose SHALL cease to be valid when the transaction terminates.

3.68. Page 92, section: 7.12: ChargingRateUnit value descriptions need more clarification

Using a ChargingRateUnit W for AC charging is potentially very complicated, and, if used, the calculation is tricky
The description of the values of the ChargingRateUnit should be improved.

VAL UE	OLD DESCRIPTION	NEW DESCRIPTION
W	Watts (power).	<p>Watts (power).</p> <p>This is the TOTAL allowed charging power.</p> <p>If used for AC Charging, the phase current should be calculated via: $\text{Current per phase} = \text{Power} / (\text{Line Voltage} * \text{Number of Phases})$. The "Line Voltage" used in the calculation is not the measured voltage, but the set voltage for the area (hence, 230 or 110 volt). The "Number of Phases" is the numberPhases from the ChargingSchedulePeriod.</p> <p>It is usually more convenient to use this for DC charging.</p> <p>Note that if numberPhases in a ChargingSchedulePeriod is absent, 3 SHALL be assumed.</p>
A	Amperes (current).	<p>Amperes (current).</p> <p>The amount of Ampere per phase, not the sum of all phases.</p> <p>It is usually more convenient to use this for AC charging.</p>

3.69. Page 93, section: 7.13: First ChargingSchedulePeriod should start with StartSchedule = 0

Added in errata sheet v3.0

There is no requirement that explains the obvious: The first ChargingSchedulePeriod should start with

StartSchedule = 0.

Updated description for the chargingSchedulePeriod field:

Old text	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time.
New text	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time. The StartSchedule of the first ChargingSchedulePeriod SHALL always be 0.

3.70. page 94, section: 7.14 ChargingSchedulePeriod Limit can also be in Watts

ChargingSchedulePeriod field: Limit can be in Watts or Ampere.

Old text	Required. Power limit during the schedule period, expressed in Amperes. Accepts at most one digit fraction (e.g. 8.1).
New text	Required. Charging rate limit during the schedule period, in the applicable chargingRateUnit, for example in Amperes or Watts. Accepts at most one digit fraction (e.g. 8.1).

3.71. page 94 - 95, section: 7.15 - 7.19 CiStringXXType should be defined as type

The CiStringXXTypes are defined as Class, that contain a Field Name, but they should have been defined as Type (without Field Name)

Paragraphs effected:

- 7.15 CiString20Type
- 7.16 CiString25Type
- 7.17 CiString50Type
- 7.18 CiString255Type
- 7.19 CiString500Type

Old definition:

7.15 CiString20Type

Class

Generic used case insensitive string of 20 characters.

FIELD NAME	FIELD TYPE	DESCRIPTION
cistring20	CiString[20]	String is case insensitive.

New definition:

7.15 CiString20Type

Type

Generic used case insensitive string of 20 characters.

FIELD TYPE	DESCRIPTION
CiString[20]	String is case insensitive.

These Changes have no effect on the WSDL and JSON definitions, they are defined correct in WSDL and JSON Schemas.

3.72. Page 98, section: 7.27: Definition of IdTagInfo misses cardinality

Every class definition in OCPP 1.6 contains a column called: "Card." (cardinality), but this column is missing in the definition of IdTagInfo.

New definition:

7.27 IdTagInfo

Class

Contains status information about an identifier. It is returned in Authorize, Start Transaction and Stop Transaction responses.

If expiryDate is not given, the status has no end date.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
expiryDate	dateTime	0..1	Optional. This contains the date at which idTag should be removed from the Authorization Cache.
parentIdTag	IdToken	0..1	Optional. This contains the parent-identifier.
status	AuthorizationStatus	1..1	Required. This contains whether the idTag has been accepted or not by the Central System.

3.73. page 98, section: 7.28. idToken: idToken field is of wrong type

The field: idToken should have been of the type CiString20Type, it is case insensitive.

Old text	String[20]
New text	CiString20Type

To prevent interoperability issues: Do NOT update WSDL files!

Note: For future version: idToken was missed when added the CiStringXXTypes. Might be removed as type, and all field of the type idToken in other Classes will then be replaced by CiString20Type.

3.74. Page 98, section: 7.28: IdToken should be defined as type

The IdToken is defined as Class, that contain a Field Name, but they should have been defined as Type (without Field Name) The WSDL and JSON Schema are correct, was only wrong in the specification. (was also wrong in OCPP 1.5)

New definition:

7.28 IdToken

Type

Contains the identifier to use for authorization. It is a case insensitive string. In future releases this may become a complex type to support multiple forms of identifiers.

FIELD TYPE	DESCRIPTION
CiString20Type	IdToken is case insensitive.

3.75. Page 98, section: 9.1.5: Incorrect text about duration in MeterValue

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 111

There is some text in the description of the configuration key: ClockAlignedDataInterval, talking about MeterValue duration. There is no duration of a MeterValue. This is old text from OCPP 1.5, which was already incorrect in OCPP 1.5. This text should have been removed.

Ignore incorrect text:

~~" and (optional) duration interval value, represented according to the ISO8601 standard"~~

3.76. Page 98, section: 9.1.6: Improved description configuration key: "ConnectionTimeOut"

Added in errata sheet v4.0

The description of the configuration key: "ConnectionTimeOut" can be improved even further

Old text	Interval from beginning of status: 'Preparing' until incipient session is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). The Charge Point SHALL go back to the original state, probably: 'Available'
----------	---

New text	Interval from beginning of status: 'Preparing' until incipient session is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). The Charge Point SHALL go back to the original state, typically: 'Available'
----------	--

3.77. Page 99, section: 7.31: No definition of .register and .interval.

There are a couple of measurands that are defined as .register or .interval. But there is no definition of what that means.

Old:

VALUE	DESCRIPTION
Energy.Active.Export.Register	Energy exported by EV (Wh or kWh)
Energy.Active.Import.Register	Energy imported by EV (Wh or kWh)
Energy.Reactive.Export.Register	Reactive energy exported by EV (varh or kvarh)
Energy.Reactive.Import.Register	Reactive energy imported by EV (varh or kvarh)
Energy.Active.Export.Interval	Energy exported by EV (Wh or kWh)
Energy.Active.Import.Interval	Energy imported by EV (Wh or kWh)
Energy.Reactive.Export.Interval	Reactive energy exported by EV. (varh or kvarh)
Energy.Reactive.Import.Interval	Reactive energy imported by EV. (varh or kvarh)

New:



Import is energy flow from the Grid to the Charge Point, EV or other load. Export is energy flow from the EV to the Charge Point and/or from the Charge Point to the Grid.

VALUE	DESCRIPTION
Energy.Active.Export.Register	Numerical value read from the "active electrical energy" (Wh or kWh) register of the (most authoritative) electrical meter measuring energy exported (to the grid).
Energy.Active.Import.Register	Numerical value read from the "active electrical energy" (Wh or kWh) register of the (most authoritative) electrical meter measuring energy imported (from the grid supply).
Energy.Reactive.Export.Register	Numerical value read from the "reactive electrical energy" (VARh or kVARh) register of the (most authoritative) electrical meter measuring energy exported (to the grid).
Energy.Reactive.Import.Register	Numerical value read from the "reactive electrical energy" (VARh or kVARh) register of the (most authoritative) electrical meter measuring energy imported (from the grid supply).

VALUE	DESCRIPTION
Energy.Active.Export.Interval	Absolute amount of "active electrical energy" (Wh or kWh) exported (to the grid) during an associated time "interval", specified by a MeterValues ReadingContext, and applicable interval duration configuration values (in seconds) for "ClockAlignedDataInterval" and "MeterValueSampleInterval".
Energy.Active.Import.Interval	Absolute amount of "active electrical energy" (Wh or kWh) imported (from the grid supply) during an associated time "interval", specified by a MeterValues ReadingContext, and applicable interval duration configuration values (in seconds) for "ClockAlignedDataInterval" and "MeterValueSampleInterval".
Energy.Reactive.Export.Interval	Absolute amount of "reactive electrical energy" (VARh or kVARh) exported (to the grid) during an associated time "interval", specified by a MeterValues ReadingContext, and applicable interval duration configuration values (in seconds) for "ClockAlignedDataInterval" and "MeterValueSampleInterval".
Energy.Reactive.Import.Interval	Absolute amount of "reactive electrical energy" (VARh or kVARh) imported (from the grid supply) during an associated time "interval", specified by a MeterValues ReadingContext, and applicable interval duration configuration values (in seconds) for "ClockAlignedDataInterval" and "MeterValueSampleInterval".



All "Register" values relating to a single charging transaction, or a non-transactional consumer (e.g. charge point internal power supply, overall supply) MUST be monotonically increasing in time.

The actual quantity of energy corresponding to a reported ".Register" value is computed as the register value in question minus the register value recorded/reported at the start of the transaction or other relevant starting reference point in time. For improved auditability, ".Register" values SHOULD be reported exactly as they are directly read from a non-volatile register in the electrical metering hardware, and SHOULD NOT be re-based to zero at the start of transactions. This allows any "missing energy" between sequential transactions, due to hardware fault, mis-wiring, fraud, etc. to be identified, by allowing the Central System to confirm that the starting register value of any transaction is identical to the finishing register value of the preceding transaction on the same connector.

3.78. Page 103, section: 7.37: RecurrencyKindType definition is ambiguous.

Added in errata sheet v3.0

The definition of RecurrencyKindType is ambiguous. It is not clear when a Charging Profile should recur.

Changes to the table

VALUE	OLD DESCRIPTION	NEW DESCRIPTION
Daily	The schedule restarts at the beginning of the next day.	The schedule restarts every 24 hours, at the same time as in the startSchedule.
Weekly	The schedule restarts at the beginning of the next week (defined as Monday morning).	The schedule restarts every 7 days, at the same time and day-of-the-week as in the startSchedule.

3.79. Page 103, section: 9.1.23: Configuration key: "StopTransactionOnEVSideDisconnect" should not be required

Added in errata sheet v4.0

OCPP 1.6 FINAL Page 116

The description of the configuration key: `StopTransactionOnEVSideDisconnect` is required. It was added to OCPP to support EVs without lock at the car side. But this is now never the case, it was only the case with the first version of the Mitsubishi Outlander PHEV.

The German eichrect does not allow this configuration key to be implemented.

Old value Required/optional	required
New value Required/optional	optional
Old value Accessibility	RW
New value Accessibility	R or RW. Choice is up to Charge Point implementation.

3.80. Page 104, section: 9.1.30: Unneeded configuration key: "SupportedFeatureProfilesMaxLength"

Added in errata sheet v4.0

The description of the configuration key: `SupportedFeatureProfilesMaxLength` has no use. `SupportedFeatureProfiles` is a readonly configuration key. `SupportedFeatureProfiles` could have been removed from the specification.

Addition al text before table	NOTE: This configuration key does not have to be implemented. It should not have been part of OCPP 1.6, "SupportedFeatureProfiles" is a readonly configuration key.
--	---

3.81. Page 105, section: 7.42: Improved description of Soft/Hard Reset

Added in errata sheet v3.0

The descriptions of Soft/Hard reset can be improved, not very clear what is the difference between the two.

Changes to the table

VALUE	OLD DESCRIPTION	NEW DESCRIPTION
Hard	Full reboot of Charge Point software.	Restart (all) the hardware, the Charge Point is not required to gracefully stop ongoing transaction. If possible the Charge Point sends a StopTransaction.req for previously ongoing transactions after having restarted and having been accepted by the Central System via a BootNotification.conf. This is a last resort solution for a not correctly functioning Charge Point, by sending a "hard" reset, (queued) information might get lost.
Soft	Return to initial status, gracefully terminating any transactions in progress.	Stop ongoing transactions gracefully and sending StopTransaction.req for every ongoing transaction. It should then restart the application software (if possible, otherwise restart the processor/controller).

3.82. Page 105, section: 9.1: Missing standard Configuration Key for Message Timeout

Added in errata sheet v4.0

OCPP does not define what the required message timeout is. As OCPP is used for a lot of different transport layers, from 3G to fiber, timing can be very different. A CPO needs to be able to configure this, based on the network used. But OCPP did not define a standard configuration key for this. So now almost every Charge Point manufacturer defines his own name for the same thing.

New definition:

9.1 Core Profile

9.1.15 MessageTimeout

Required/optional	optional
Accessibility	RW
Type	integer
Description	Defines the OCPP Message timeout in seconds. If the Charge Point has not received a response to a request within this timeout, the Charge Point SHALL consider the request timed out.

3.83. Page 105, section: 9.1.33: Unclear how to implement UnlockConnectorOnEVSideDisconnect with a Charge Point with a fixed cable

Added in errata sheet v4.0

In the table describing the variable UnlockConnectorOnEVSideDisconnect change the text of the following rows:

Accessibility	RW (RO in case of fixed cable)
Description	When set to true, the Charge Point SHALL unlock the cable on Charge Point side when the cable is unplugged at the EV. A Charge Point with a fixed cable SHALL always report this value as false and SHALL not allow this value to be changed.

3.84. Page 106, section: 7.45: No UnitOfMeasure for Measurand Frequency.

There is no UnitOfMeasure for Measurand: Frequency.

Add to the description for Frequency on page 100, section: 7.31:	OCPP 1.6 does not have a UnitOfMeasure for frequency, the UnitOfMeasure for any SampledValue with measurand: Frequency is Hertz.
--	--

3.85. Page 107, section: 7.42: UnlockConnector with unknown ConnectorId

Added in errata sheet v3.0

It has not been specified how a Charge Point should respond when a Central System request an Unlock Connector for an unknown ConnectorId.

Preferably the response would have been: "Rejected", so that will be added to OCPP 2.0. For OCPP 1.6 we cannot add extra states, so we have to use "NotSupported". "UnlockFailed" should not be used for this, "UnlockFailed" is really for when the locking mechanism detects a failed unlock attempt.

Changes to the table

VALUE	OLD DESCRIPTION	NEW DESCRIPTION
UnlockFailed	Failed to unlock the connector.	Failed to unlock the connector: The Charge Point has tried to unlock the connector and has detected that the connector is still locked or the unlock mechanism failed.
NotSupported	Charge Point has no connector lock	Charge Point has no connector lock, or ConnectorId is unknown.

3.86. Page 107, section: 9.4.2: Configuration key: "ChargingScheduleAllowedChargingRateUnit" values confusing

Added in errata sheet v4.0

The configuration key: ChargingScheduleAllowedChargingRateUnit has two allowed values: 'Current' and 'Power'. While ChargingRateUnitType (page 80, section: 7.12) has the possible values: 'A' and 'W'. This is confusing.

It would have been better if the allowed values for ChargingScheduleAllowedChargingRateUnit would also have been: 'A' and 'W'. That will be the solution for OCPP 2.0.

For OCPP 1.6 an extra explanation is added.

Addition al descripti on	'Current' means only ChargingSchedules with ChargingRateUnit: 'A' allowed 'Power' means only ChargingSchedules with ChargingRateUnit: 'W' allowed
--------------------------	--

3.87. Page 112, section: 9.1.6: Improved description configuration key: "ConnectionTimeOut"

The description of the configuration key: "ConnectionTimeOut" can be improved

Old text	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).
New text	Interval from beginning of status: 'Preparing' until incipient session is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). The Charge Point SHALL go back to the original state, probably: 'Available'

3.88. Page 121, section: 9: Missing definition for SupportedFileTransferProtocols

Added in errata sheet v4.0

In chapter 8, page 96 (OCPP 1.6 FINAL: page 109), there is a reference to a configuration key: SupportedFileTransferProtocols. This configuration key is not defined in chapter 9.

New definition:

9.5 Firmware Management

9.5.1 SupportedFileTransferProtocols

Required/optional	optional
Accessibility	R
Type	CSL
Description	This configuration key tells the Central System which file transfer protocols are supported by the Charge Point. Allowed values: 'FTP', 'FTPS', 'HTTP' and 'HTTPS'.

3.89. Page 121, section: 9.1: Central System needs to known maximum amount of meter values in StopTransaction

Added in errata sheet v4.0

When a Charge Point is configured to provide meter values in the StopTransaction via: StopTxnSampledData and/or StopTxnAlignedData, and the Transaction takes really long (think: EV parked for days at an airport) or the interval in configured to a low value in: ClockAlignedDataInterval or MeterValueSampleInterval, the Charge Point might collect more meter values than it can store or send in a StopTransaction. In this case the Charge Point needs to drop intermediate values to prevent crashes etc. But the Central System needs to known at what point this happens.

New definition:

9.1 Core Profile

9.1.23 StopTransactionMaxMeterValues

Required/optional	optional
Accessibility	R
Type	integer
Description	<p>The maximum amount of meter values that this Charge Point can report in the transactionData field of a StopTransaction.req. When the amount of meter values collected for a transaction exceeds: StopTransactionMaxMeterValues, the Charge Point MAY drop intermediate meter values, to prevent running out of memory, or being unable to send the StopTransaction.req (overrunning the transmit buffer size). The Start and Stop meter values SHALL never be dropped.</p> <p>When the Charge Point needs to store more intermediate values than: StopTransactionMaxMeterValues, it is RECOMMENDED not to start dropping messages from the start, or stop storing new values. It is better to drop intermediate messages first (1st message, 3th message, 5th message etc.), or uses a smart algorithm, for example remove duplicate values first. etc.</p>

4. Typos

Typos, fixes to incorrect links/reference, improve terms used etc. that have no impact on the description of the way the protocol works.

4.1. Generic: Typo Field Type: DateTime should be dateTime

dateTime field type is misspelled a couple of times as: DateTime (with upper-case D)

PAGE	SECTION	MESSAGE/CLASS	FIELD NAME
24	3.12.2	NOTE add the bottom	validFrom
70	6.22	GetCompositeSchedule.conf	scheduleStart
90	7.8	ChargingProfile	validFrom
91	7.8	ChargingProfile	validTo
93	7.13	ChargingProfile	startSchedule

4.2. Generic: Use of Energy Meter vs Power Meter

The terms Energy Meter and Power Meter are use throughout the specification, but they are not used consistently and the term: Electrical Meter seems to fit most cases even better.

List of all textual improvements for this:

PAGE	SECTION	OLD TEXT	NEW TEXT
8	2.2	Defines the wiring order of the phases between the energy meter (or if absent, the grid connection), and the Charge Point connector.	Defines the wiring order of the phases between the electrical meter (or if absent, the grid connection), and the Charge Point connector.
38	4.7	A Charge Point MAY sample the energy meter or other sensor/transducer hardware to provide extra information about its meter values.	A Charge Point MAY sample the electrical meter or other sensor/transducer hardware to provide extra information about its meter values.
38	4.7	The Charging Point SHALL report all phase number dependent values from the power meter (or grid connection when absent) point of view.	The Charging Point SHALL report all phase number dependent values from the electrical meter (or grid connection when absent) point of view.
63	6.3	This contains the serial number of the main power meter of the Charge Point.	This contains the serial number of the main electrical meter of the Charge Point.
63	6.3	This contains the type of the main power meter of the Charge Point.	This contains the type of the main electrical meter of the Charge Point.
87	7.6	Failure to read power meter.	Failure to read electrical/energy/ power meter.
116	9.1.21	The phase rotation per connector in respect to the connector's energy meter (or if absent, the grid connection).	The phase rotation per connector in respect to the connector's electrical meter (or if absent, the grid connection).

4.3. Page 13, section: 3.2: Typo in text about SupportedFeatureProfiles

Below the table with the mapping of messages to feature profiles, there is a typo: "charging profiles" instead of "feature profiles".

Old text	The support for the specific charging profiles is reported by the SupportedFeatureProfiles configuration key.
New text	The support for the specific feature profiles is reported by the SupportedFeatureProfiles configuration key.

4.4. Page 18, section: 3.4.4: Typo in Unknown Offline Authorization

There is a typo in the text about Unknown Offline Authorization.

Old text	When connection the the Central Server is restored
New text	When connection to the Central Server is restored

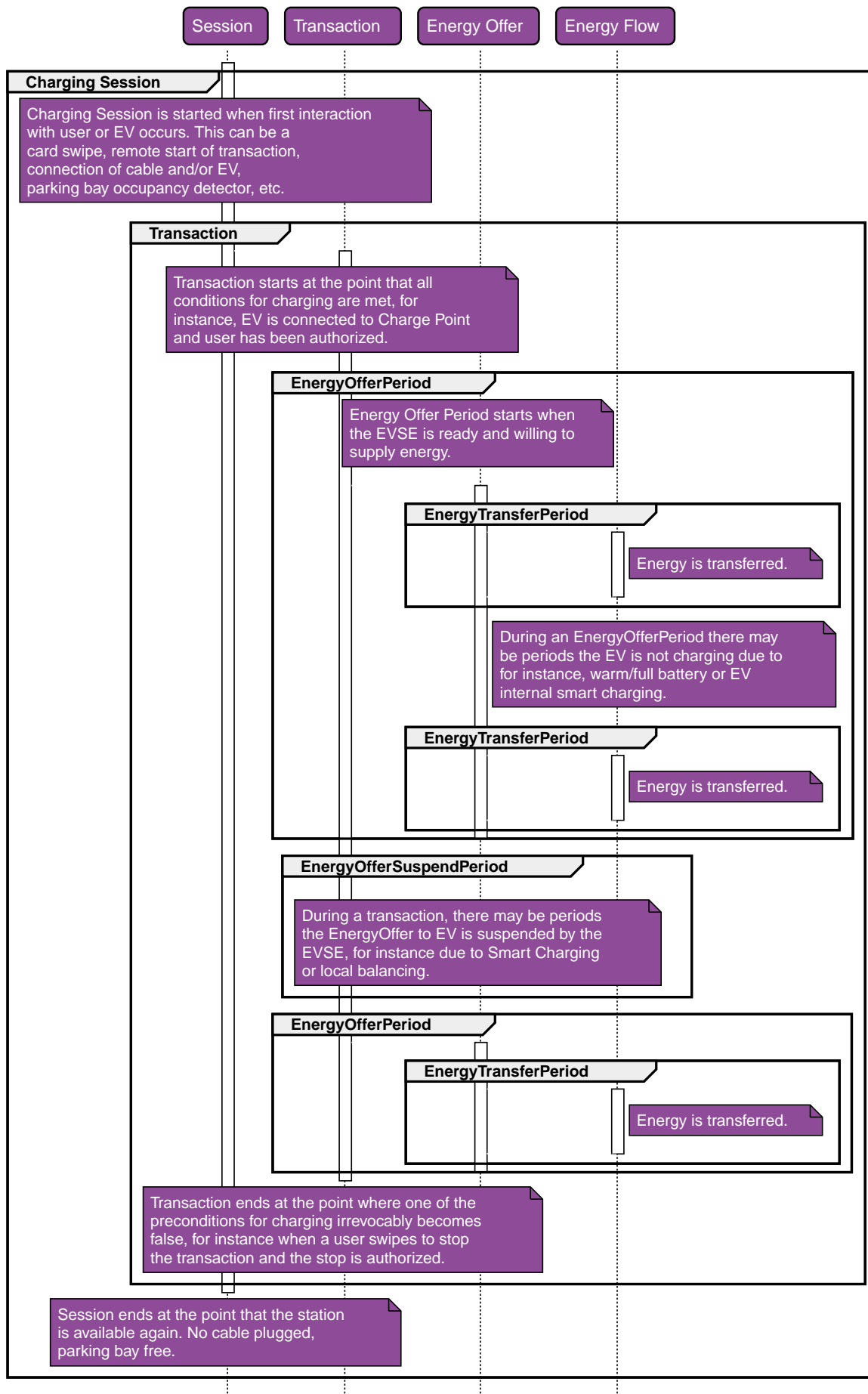
4.5. Page 19, section: 3.5: Definition of OCPP Transaction, Session, EnergyOfferPeriod etc missing.

In the table on page 19, there are some terms used that are never described in the specification.

Changes to the diagram on page 19:

OLD TEXT	NEW TEXT
Session	Charging Session
OCPP Transaction	Transaction

Update diagram for section: 3.5 on page 19:



Changes to the table of definitions in section:. 2.2 on page 7

TERM	OLD DESCRIPTION	NEW DESCRIPTION
Charging Session	Part of a transaction during which the EV is allowed to request energy	a Charging Session is started when first interaction with user or EV occurs. This can be a card swipe, remote start of transaction, connection of cable and/or EV, parking bay occupancy detector, etc.

Additions to the table of definitions in section:. 2.2 on page 7

TERM	DESCRIPTION
Energy Offer Period	Energy Offer Period starts when the EVSE is ready and willing to supply energy.
Energy Offer SuspendPeriod	During a transaction, there may be periods the EnergyOffer to EV is suspended by the EVSE, for instance due to Smart Charging or local balancing.

Changes throughout the entire specification, correcting incorrect term use.

PAGE	PAR.	OLD TEXT	NEW TEXT
42	4.9	C6: Charging session is stopped by user or a Remote Stop Transaction message and further user action is required (e.g. remove cable, leave parking bay)	C6: Transaction is stopped by user or a Remote Stop Transaction message and further user action is required (e.g. remove cable, leave parking bay)
42	4.9	D6: Charging session is stopped and further user action is required	D6: Transaction is stopped and further user action is required
43	4.9	E6: Charging session is stopped and further user action is required	E6: Transaction is stopped and further user action is required
43	4.9	F2: User restart charging session (e.g. reconnects cable, presents idTag again)	F2: User restart charging session (e.g. reconnects cable, presents idTag again), thereby creating a new Transaction
88	7.7	Preparing: When a Connector becomes no longer available for a new user but no charging session is active . Typically a Connector is occupied when a user presents a tag, inserts a cable or a vehicle occupies the parking bay	Preparing: When a Connector becomes no longer available for a new user but there is no ongoing Transaction (yet) . Typically a Connector is in preparing state when a user presents a tag, inserts a cable or a vehicle occupies the parking bay
88	7.7	Finishing: When a charging session has stopped at a Connector, but the Connector is not yet available for a new user, e.g. the cable has not been removed or the vehicle has not left the parking bay	Finishing: When a Transaction has stopped at a Connector, but the Connector is not yet available for a new user, e.g. the cable has not been removed or the vehicle has not left the parking bay.
91	7.9	such as the start of a session	such as the start of a Transaction
111	9.1.5	or partial interval, at the beginning or end of a charging session	or partial interval, at the beginning or end of a Transaction

PAGE	PAR.	OLD TEXT	NEW TEXT
112	9.1.6	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).	Interval (from successful authorization) until incipient Transaction is automatically canceled due to failure of EV user to (correctly) insert the charging cable connector(s) into the appropriate connector(s).
117	9.1.25	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	Clock-aligned periodic measurand(s) to be included in the TransactionData element of StopTransaction.req MeterValues.req PDU for every ClockAlignedDataInterval of the Transaction
117	9.1.27	Sampled measurands to be included in the TransactionData element of StopTransaction.req PDU, every MeterValueSampleInterval seconds from the start of the charging session	Sampled measurands to be included in the TransactionData element of StopTransaction.req PDU, every MeterValueSampleInterval seconds from the start of the Transaction
121	9.4.4	If defined and true, this Charge Point support switching from 3 to 1 phase during a charging session .	If defined and true, this Charge Point support switching from 3 to 1 phase during a Transaction .

4.6. Page 34, section: 4.2: Boot Notification diagram: Interval

Figure 13: "Sequence Diagram: Boot Notification" contains a typo.

Old text	BootNotification.conf(currentTime, heartbeatInterval , status)
New text	BootNotification.conf(currentTime, interval , status)

4.7. Page 35, section: 4.21: Typo in description

Added in errata sheet v4.0

This errata does not effect OCPP 1.6 Final (first edition)

The text in contains: "this such" which is just wrong.

Old text	Parties who want to implement this such behaviour must realize that it is uncertain if those transactions can ever be delivered to the Central System.
New text	Parties who want to implement this behaviour must realize that it is uncertain if those transactions can ever be delivered to the Central System.

4.8. Page 35, section: 4.5. Wrong message name for UpdateFirmware

Added in errata sheet v4.0

This errata does not effect OCPP 1.6 Final (first edition)

Last sentence of this section contains a reference to UpdateFirmware.req, but is has an incorrect name

Old text	started by the Central System with a FirmwareUpdate.req PDU
New text	started by the Central System with a UpdateFirmware.req PDU

4.9. Page 39, section: 4.9. Typo in table: Preparing

Added in errata sheet v4.0

This errata does not effect OCPP 1.6 Final (first edition)

The title of column 2 contains a typo

Old text	Prepairing
New text	Preparing

4.10. Page 47, section: 4.10: Typo description in stop transaction

There is a typo in the description of StopTransactionOnEVSideDisconnect set to true.

Old text	Setting StopTransactionOnEVSideDisconnect to true will prevent sabotage acts top stop the energy flow by unplugging not locked cables on EV side.
New text	Setting StopTransactionOnEVSideDisconnect to true will prevent sabotage acts to stop the energy flow by unplugging not locked cables on EV side.

4.11. Page 50, section: 5.5: Clear Charging Profile sequence diagram: incorrect .conf message

The sequence diagram on page 50 for Clear Charging Profile contains a typo. It contains the incorrect response: "ClearCache.conf" instead of "ClearChargingProfile.conf"

Update diagram for section: 5.5 on page 50:

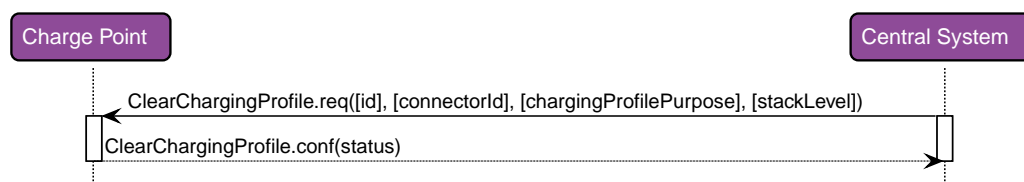


Figure 3. Updated Sequence Diagram: Clear Charging Profile

4.12. Page 55, section: 5.13: Missing MAY in description Reservation Parent idTag

The text MAY is missing in the description of getting the parent-id for a reservation via Authorize.req.

Old text	The Authorize.conf response contains the parent-id.
----------	--

New text	The Authorize.conf response MAY contain the parent-id
----------	--

4.13. Page 55, section: 5.16.4: Typo in note about first chargingSchedulePeriod

Added in errata sheet v4.0

This errata does not effect OCPP 1.6 Final (first edition)

There is a typo in the note about the first chargingSchedulePeriod

Old text	The StartSchedule of the first chargingSchedulePeriod in a chargingSchedule SHALL always be 0.
New text	The startPeriod of the first chargingSchedulePeriod in a chargingSchedule SHALL always be 0.

4.14. Page 56, section: 5.14: Typo in reset description

Added in errata sheet v3.0

There is a typo in the description of Reset response.

Old text	The response PDU SHALL include whether the Charge Point is will attempt to reset itself.
New text	The response PDU SHALL include whether the Charge Point will attempt to reset itself.

4.15. Page 60, section: 5.18: Central System sends Unlock Connector

In the paragraph about Unlock Connector, "Charge Point" and "Central Server" are mixed up.

Old text	To do so, The Charge Point SHALL send
New text	To do so, The Central System SHALL send

4.16. Page 71, section: 6.23: Typo in GetConfiguration.req

There is a typo in the text about GetConfiguration.req.

Old text	This contains the field definition of the GetConfiguration.req PDU sent by the the Central System to the Charge Point
New text	This contains the field definition of the GetConfiguration.req PDU sent by the Central System to the Charge Point

4.17. Page 81, section: 7.13: Typo in ChargingSchedule chargingSchedulePeriod description

Added in errata sheet v4.0

There is a typo in the description of chargingSchedulePeriod

Old text	The startSchedule of the first ChargingSchedulePeriod SHALL always be 0.
New text	The startPeriod of the first ChargingSchedulePeriod SHALL always be 0.

4.18. Page 91, section: 7.9: ChargingProfileKindType misses description and 'where used'

There is no 'where used' in the definition of ChargingProfileKindType

Additional text	Kind of charging profile, as used in: ChargingProfile
-----------------	---

4.19. Page 91, section: 7.10: Description of ChargePointMaxProfile in ChargingProfilePurposeType contains unused words

The description ChargePointMaxProfile in ChargingProfilePurposeType contains words that should not have been here.

Old text	Configuration for the maximum power or current available for an entire Charge Point. SetChargingProfile.req message.
New text	Configuration for the maximum power or current available for an entire Charge Point.

4.20. Page 91, section: 7.10: ChargingProfilePurposeType misses description and 'where used'

There is no 'where used' in the definition of ChargingProfilePurposeType

Additional text	Purpose of the charging profile, as used in: ChargingProfile
-----------------	--

4.21. Page 92, section: 7.13: ChargingSchedule misses description and 'where used'

There is no 'where used' in the definition of ChargingSchedule.

Additional text	Charging schedule structure defines a list of charging periods, as used in: GetCompositeSchedule.conf and ChargingProfile
-----------------	---

4.22. Page 93, section: 7.14: ChargingSchedulePeriod misses description and 'where used'

There is no 'where used' in the definition of ChargingSchedulePeriod.

Additional text	Charging schedule period structure defines a time period in a charging schedule, as used in: ChargingSchedule
-----------------	---

4.23. Page 96, section: 7.22. Typo in descriptions ConfigurationStatus

Word 'is' is missing in descriptions of ConfigurationStatus

OLD TEXT	NEW TEXT
Configuration key supported and setting has been changed.	Configuration key is supported and setting has been changed.
Configuration key supported, but setting could not be changed.	Configuration key is supported, but setting could not be changed.
Configuration key supported and setting has been changed,	Configuration key is supported and setting has been changed,

4.24. Page 101, section: 7.33: MeterValue misses used by StopTransaction.req

In the description of MeterValue there is a link to the usage of MeterValue: MeterValue.req, MeterValue it is also used in StopTransaction.req.

Old text	Collection of one or more sampled values in MeterValues.req.
New text	Collection of one or more sampled values in MeterValues.req and StopTransaction.req .

4.25. Page 102, section: 7.35: Descriptions Transaction.Begin, Transaction.End swapped

The descriptions of Transaction.Begin and Transaction.End are swapped.

Old:

VALUE	DESCRIPTION
Transaction.Begin	Value taken at end of transaction.
Transaction.End	Value taken at start of transaction.

New:

VALUE	DESCRIPTION
Transaction.Begin	Value taken at start of transaction.
Transaction.End	Value taken at end of transaction.

4.26. Page 106, section: 7.45: UnitOfMeasure links to incorrect usage

In the description of UnitOfMeasure there are links to the usage of UnitOfMeasure. There incorrectly say: MeterValues.req and StopTransaction.req, should be: SampledValue

Old text	Allowable values of the optional "unit" field of a Value element, as used in MeterValues.req and StopTransaction.req messages .
----------	--

New text	Allowable values of the optional "unit" field of a Value element, as used in SampledValue .
----------	--

4.27. Page 110, section: 9: Typo in Standard Configuration Key Names & Values

There is a typo in the text about Standard Configuration Key Names & Values.

Old text	In case the the accessibility is read-write, the Central System can also write the value for the key using ChangeConfiguration
New text	In case the accessibility is read-write, the Central System can also write the value for the key using ChangeConfiguration.

4.28. Page 121, section: 9.4.4: ConnectorSwitch3to1PhaseSupported: Type should be boolean

There is a typo in the type definition of the configuration key: ConnectorSwitch3to1PhaseSupported: Type should be boolean, is now bool.

Old type	bool
New type	boolean

4.29. Page 123, section: A.1: Power.Factor is missing in the list of new enum values

In the list of new values for the enum: Measurand, the value: Power.Factor is missing, should be added.

Old text	Added enum Current.Offered, Frequency, Power.Offered, RPM and SoC
New text	Added enum Current.Offered, Frequency, Power.Factor , Power.Offered, RPM and SoC

5. Known issues that will not be fixed

None known