
# OCPP 2.1 Edition 1

Errata 2025-09

# Table of Contents

# Disclaimer



---

**Version History**

| Version | Date | Description |
|---|---|---|
| 2025-09 | 2025-09-22 | Includes errata for Part 2 and 4 of OCPP 2.1 Edition 1. |
| 2025-06 | 2025-07-08 | Includes errata for Part 2 and 4 of OCPP 2.1 Edition 1. |

# Scope

This document contains errata on the OCPP 2.1 documentation. These errata have to be read as an addition to the release of OCPP 2.1 Edition 1.

The errata do not affect any schemas of OCPP messages. Certain errata do contain changes to requirements or even new requirements, but only in cases where a requirement contains an obvious error and would not or could not be implemented literally. New requirements are only added when they were already implicitly there. These changes have been discussed in or were proposed by the Technology Working Group of the Open Charge Alliance.

The appendices of the OCPP specification can be updated without requiring a new OCPP release. This mainly concerns the components and variables of the OCPP device model, which can be extended with new components or variables, as long as they are optional.

# Terminology and Conventions

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

The errata entries are sorted by page number of the affected section of the specification document. When an errata entry affects multiple parts of the specification, then the various changes are grouped together with subsections referring to the pages affected by those changes.

This is version 2025-09 of the errata. The errata of this version are marked with "(2025-09)" in the section title.

In some cases the issue number by which it was reported, is added in square brackets at the end of the section title, e.g. "[349]". For retrieval of the issue in the issue tracking system prefix the number with "OCPP20M", like "[OCPP20M-349]".

# 0. Part 0

*Currently no new errata for OCPP 2.1 Edition 1 part 0.*

# 1. Part 1

*Currently no new errata for OCPP 2.1 Edition 1 part 1.*

# 2. Part 2

## 2.1. Page 65 - (2025-09) - B07.FR.15 has been deleted [1007]

B07.FR.15 was added since 2.1, but it is wrong. It speaks about ReadOnly variables, but this must be WriteOnly. It is also not needed, because in 2.0.1 Errata 2025-02 requirement B07.FR.03 was already updated to exclude WriteOnly variables.

*Deleted requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| ~~B07.FR.15~~ | When the Charging Station is sending the requested information via one or more NotifyReportRequest messages to the CSMS | The Charging Station SHALL omit the *value* of readonly variables | |

## 2.2. Page 70 - (2025-09) - B09.FR.02/04/05 - Added optional reasonCode

A mention of adding an optional *reasonCode* when a SetNetworkProfileRequest is rejected, has been added.

*Changed requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| B09.FR.02 | On receipt of the SetNetworkProfileRequest | The Charging Station SHALL validate the content. If the content is invalid, the Charging Station SHALL respond by sending a SetNetworkProfileResponse message, with status *Rejected* and optional *statusInfo.reasonCode* = "InvalidNetworkConf" | Matches B09.FR.34 for NetworkConfiguration. |
| B09.FR.04 | The variable AllowSecurityProfileDowngrade is not implemented or implemented and set to false AND the Charging Station receives a SetNetworkProfileRequest AND the NetworkConnectionProfile contains a lower securityProfile than the currently active security profile | The Charging Station SHALL respond by sending a SetNetworkProfileResponse message, with status *Rejected* and optional *statusInfo.reasonCode* = "NoSecurityDowngrade" | Matches B09.FR.35 for NetworkConfiguration. |
| B09.FR.05 | When the value of *configurationSlot* in SetNetworkProfileRequest does not match an entry in *valuesList* of NetworkConfigurationPriority | The Charging Station SHALL respond by sending a SetNetworkProfileResponse message with status *Rejected* with optional *statusInfo.reasonCode* = "InvalidConfSlot" | |

## 2.3. Page 71 - (2025-09) - B09.FR.22/26/27/28 - Improved definitions

The original requirements have been rephrased so that they apply to network configurations from SetNetworkProfileRequests as well as instances of NetworkConfiguration.

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | B09.FR.22 **(2.1)** | B09.FR.10 AND<br>On receipt of a SetVariablesRequest containing any NetworkConfiguration component variable AND<br>the component instance matches any of the members in the currently configured NetworkConfigurationPriority | The Charging Station SHALL respond by sending a SetVariablesResponse with the corresponding *setVariableResult* containing status `Rejected` | It is not allowed to update any NetworkConfiguration instance that can potentially be used during a reconnection attempt. |
| New | B09.FR.22 **(2.1)** | B09.FR.10 AND<br>On receipt of a SetVariablesRequest containing any NetworkConfiguration component variable AND<br>the component instance matches any of the members in the currently configured NetworkConfigurationPriority | The Charging Station SHALL respond by sending a SetVariablesResponse with the corresponding *setVariableResult* with *attributeStatus* `Rejected` and *attributeStatusInfo.reasonCode* = "ActiveNetworkConf" | It is not allowed to update any NetworkConfiguration instance that can potentially be used during a reconnection attempt. |
| Old | B09.FR.26 **(2.1)** | B09.FR.10 AND<br>On receipt of a SetVariablesRequest containing the variable SecurityCtrlr.Identity AND<br>the mutability of this variable is read/write | The Charging Station SHALL also set the variable of the same name in all NetworkConfiguration component instances to the same value (if valid), including component instances which are contained in the currently configured NetworkConfigurationPriority. This is for backwards compatibility only. CSMS SHOULD set the NetworkConfiguration component variable instead. | |
| New | B09.FR.26 **(2.1)** | When a SetVariablesRequest changes the variable of SecurityCtrlr.Identity | The Charging Station SHALL clear the Identity from the active NetworkConnectionProfile and NetworkConfiguration (when it is writable). | The SecurityCtrlr.Identity is deprecated, and remains for backwards compatibility only. This assures that the Charging Station will use the value from the SecurityCtrlr.Identity if it is set by CSMS (See B09.FR.16). |
| Old | B09.FR.27 **(2.1)** | B09.FR.10 AND<br>On receipt of a SetVariablesRequest containing the variable SecurityCtrlr.BasicAuthPassword | The Charging Station SHALL also set the variable of the same name in all NetworkConfiguration component instances to the same value (if valid), including component instances which are contained in the currently configured NetworkConfigurationPriority. This is for backwards compatibility only. CSMS SHOULD set the NetworkConfiguration component variable instead. | |
| New | B09.FR.27 **(2.1)** | When a SetVariablesRequest changes the variable SecurityCtrlr.BasicAuthPassword | The Charging Station SHALL clear the BasicAuthPassword from the active NetworkConnectionProfile and NetworkConfiguration (when it is writable). | The SecurityCtrlr.BasicAuthPassword is deprecated, and remains for backwards compatibility only. This assures that the Charging Station will use the value from the SecurityCtrlr.BasicAuthPassword if it is set by CSMS (See B09.FR.16). |
| Old | B09.FR.28 **(2.1)** | B09.FR.10 AND<br>When Charging Station activates a new network configuration | Charging Station SHALL ensure that the values of SecurityCtrlr.Identity and SecurityCtrlr.BasicAuthPassword match the corresponding variables of NetworkConfiguration.Identity[*configurationSlot*] and NetworkConfiguration.BasicAuthPassword [*configurationSlot*] for the currently active *configurationSlot*. | |

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New | B09.FR.28 **(2.1)** | If the NetworkConnectionProfile or NetworkConfiguration used for the currently active connection includes values for the variables Identity and/or BasicAuthPassword | The Charging Station SHALL set the values of SecurityCtrlr.Identity and/or SecurityCtrlr.BasicAuthPassword accordingly. | |

## 2.4. Page 71 - (2025-09) - B09.FR.31/32 - Improved definition

The original requirements has been rephrased so that it applies to a network configuration from either SetNetworkProfileRequests or from an instance of NetworkConfiguration.

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | B09.FR.31 | The variable AllowSecurityProfileDowngrade is implemented and set to true AND The currently active 'SecurityProfile' is 3 AND The Charging Station receives a SetNetworkProfileRequest AND the NetworkConnectionProfile contains a securityProfile with a value of 2. | The Charging Station SHALL respond with SetVariablesResponse(Accepted) | |
| New | B09.FR.31 | The variable AllowSecurityProfileDowngrade is implemented and set to true AND the currently active 'SecurityProfile' is higher than 1 AND the Charging Station receives a SetNetworkProfileRequest with a NetworkConnectionProfile with *securityProfile* = 1 | The Charging Station SHALL respond with SetNetworkProfileResponse with *status* `Rejected` and optional *statusInfo.reasonCode* = "NoSecurityDowngrade" | |
| Old | B09.FR.32 | The variable AllowSecurityProfileDowngrade is implemented and set to true AND The currently active 'SecurityProfile' is higher than 1 AND The Charging Station receives a SetNetworkProfileRequest AND the NetworkConnectionProfile contains a securityProfile with a value of 1. | The Charging Station SHALL respond with SetVariablesResponse(Rejected) | |
| New | B09.FR.32 **(2.1)** | The variable AllowSecurityProfileDowngrade is implemented and set to true AND the currently active 'SecurityProfile' is higher than 1 AND the Charging Station receives a SetVariablesRequest for NetworkConfiguration.SecurityProfile with *attributeValue* = 1. | The Charging Station SHALL respond with SetVariablesResponse with the corresponding *setVariableResult* with *attributeStatus* `Rejected` and *attributeStatusInfo.reasonCode* = "NoSecurityDowngrade" | |

## 2.5. Page 71 - (2025-09) - B09.FR.33/34/35 - Added requirements to validate NetworkConfiguration

The same validations that are performed when activating a network connection profile from SetNetworkProfileRequest also need to be performed when activating an instance of NetworkConfiguration.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| B09.FR.33 **(2.1)** | B09.FR.10 AND<br>On receipt of a SetVariablesRequest containing the variable<br>NetworkConfigurationPriority AND the new value adds configuration slot(s) to the current value | The Charging Station SHALL validate the NetworkConfiguration component for instances equal to the added configuration slot(s), and if successful, the Charging Station SHALL respond by sending a SetVariablesResponse message, with status *Accepted* | |
| B09.FR.034 **(2.1)** | B09.FR.10 AND<br>On receipt of a SetVariablesRequest containing the variable<br>NetworkConfigurationPriority AND the new value adds configuration slot(s) to the current value | The Charging Station SHALL validate the NetworkConfiguration component for instances equal to the added configuration slot(s), and if not successful the Charging Station SHALL respond by sending a SetVariablesResponse message, with the corresponding *setVariableResult* with *attributeStatus* `Rejected` and *attributeStatusInfo.reasonCode* = "InvalidNetworkConf" | The field *additionalInfo* can be used to provide details about which NetworkConfiguration variable is invalid. Matches B09.FR.02 for SetNetworkProfileRequest. |
| B09.FR.35 **(2.1)** | The variable AllowSecurityProfileDowngrade is not implemented or set to false AND<br>The Charging Station receives a SetVariablesRequest for NetworkConfiguration.SecurityProfile with an *attributeValue* that has a lower value than the currently active 'SecurityProfile' | The Charging Station SHALL respond with SetVariablesResponse with the corresponding *setVariableResult* with *attributeStatus* `Rejected` and *attributeStatusInfo.reasonCode* = "NoSecurityDowngrade" | Matches B09.FR.04 for SetNetworkProfileRequest. |

# 2.6. Page 74 - (2025-06) - B10.FR.03/04/10 - Migrate to new NetworkConnectionProfile

*Changed requirements*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | B10.FR.03 | B10.FR.04 AND<br>When connecting fails | The Charging Station SHALL make the number of attempts as configured in `NetworkProfileConnectionAttempts` per entry of `NetworkConfigurationPriority`. | |
| New | B10.FR.03 | After a reboot OR<br>When connection to CSMS is lost | The Charging Station SHALL make the number of attempts as configured in `NetworkProfileConnectionAttempts` per entry of `NetworkConfigurationPriority`. | |
| Old | B10.FR.04 | B10.FR.01 OR B09.FR.01 AND<br>After a reboot | The Charging Station SHALL begin connecting to the first entry of `NetworkConfigurationPriority` | Same as A05.FR.05 |
| New | B10.FR.04 | ( B10.FR.01 OR B09.FR.01 ) AND<br>After a reboot | The Charging Station SHALL begin connecting to the first entry of `NetworkConfigurationPriority` | Same as A05.FR.05 |

The following requirement is added to make explicit that a BootNotification must be sent, or else Charging Station might connect to a new CSMS without it, in which case CSMS would respond with a CALLERROR(SecurityEvent).

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| B10.FR.10 (new) | B10.FR.03 AND<br>Charging Station successfully connected after having switched to a different NetworkConnectionProfile | Charging Station SHALL send a BootNotificationRequest to CSMS to reestablish its registration status, even if it has not rebooted since last being accepted by any CSMS. | Charging Station does not need to check whether the CSMS it connected to, is actually one that it has not connected to before. |

## 2.7. Page 75 - (2025-06) - B11 - Clarify meaning of OnIdle for Reset

The "idle state" is defined in Terminology as: "In both use cases and sequence diagrams, Idle status is referred as the state in which a Charging Station is not performing any use case related tasks. Condition during which the equipment can promptly provide a primary function but is not doing so." This is a broader concept, than having an active transaction. A remark is added to the use case to explain that.

The sentence about persistent states and ResetResponse did not belong in Remarks section.

| No. | Type | Description |
|---|---|---|
| 1 | **Name** | Reset - Without Ongoing Transaction |
| ... | ... | |
| 8 | **Remark(s)** | ~~Persistent states: for example, EVSE set to *Unavailable* SHALL persist.~~<br>~~+ [line-through]#The Charging Station responds with ResetResponse.~~<br><br>`OnIdle` refers to the "idle state" of a charging station. This is when the Charging Station is not performing any use case related tasks that might interfere with a reset process. The most obvious case is being involved in an active transaction, but there are other conditions when the Charging Station is not idle, for example, when a firmware update process is ongoing, a log file is uploaded to CSMS, a reservation is pending or a cable is still locked in the Charging Station. |

## 2.8. Page 189 - (2025-06) - E06.FR.05 for DataSigned as TxStopPoint is invalid

DataSigned cannot be used as a TxStopPoint. This requirement is therefore invalid and confusing when present.

*Deleted requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Delete | E06.FR.05 | `TxStopPoint` contains: DataSigned<br>AND<br>Charging Station can no longer retrieve signed meter values. | The Charging Station SHALL stop the transaction and send a TransactionEventRequest (eventType = Ended) to the CSMS. |

## 2.9. Page 193 - (2025-09) - E07.FR.07 - Improved precondition

The precondition of E07.FR.07 was written as text, but it is more precise to refer another requirement.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | E07.FR.07 | As part of the normal transaction termination. | The Charging Station SHALL unlock the cable (if not permanently attached). | |
| New | E07.FR.07 | E07.FR.02 | The Charging Station SHALL unlock the cable (if not permanently attached). | |

## 2.10. Page 219 - (2025-06) - E17.FR.01 Clarification of transaction state to store

Minor improvement of definition to clarify that state information needed to resume a transaction must be persisted.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | E17.FR.01 | If [configkey-tx-resumption-timeout] > 0 | Charging Station SHALL store transaction state in persistent memory | This is needed in order to resume transactions after a power loss. |

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| New | E17.FR.01 | If [configkey-tx-resumption-timeout] > 0 | Charging Station SHALL store transaction state that is needed to resume transactions in persistent memory | This includes at least, but is not limited to, the *seqNo*, *idToken*, *evse* and *transactionInfo* data of all active transactions. This ensures transactions can be restored after a power loss. |

# 2.11. Page 240 - (2025-06) - F06 Requirement for CSMS to support customTrigger [896]

A requirement for CSMS to support customTriggers was missing.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| F06.FR.20 **(2.1)** | If Charging Station reports custom triggers in CustomizationCtrlr.CustomTriggers | CSMS SHALL support sending these custom triggers as a [triggermessagerequest] with *requestedMessage* = `CustomTrigger` and *customTrigger* set to the custom trigger. | |

# 2.12. Page 260 - (2025-06) - H02 - Added missing requirements

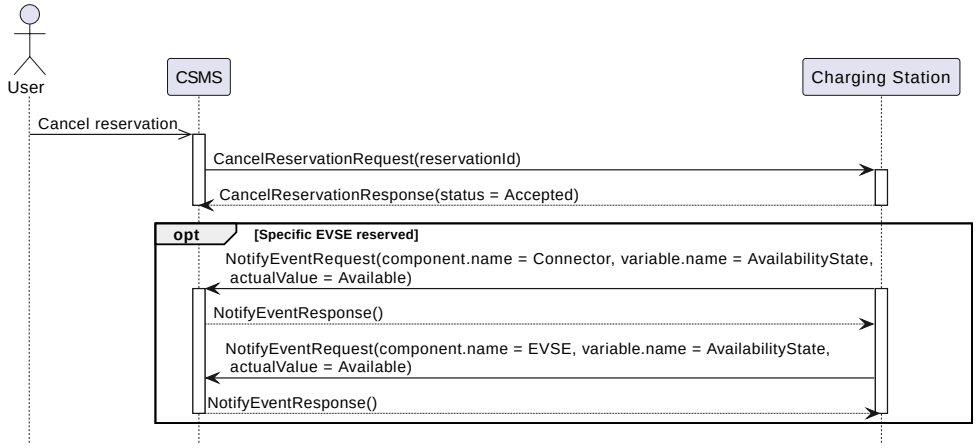
Added missing requirements explicitly specifying behaviour of Charging Station when a reservation is cancelled.

Removed details from scenario description:

| No. | Type | Description |
|---|---|---|
| | | [...] |
| | *Scenario description* | **1.** EV Driver asks the CSMS to cancel a reservation.<br>**2.** To cancel a reservation, the CSMS sends CancelReservationRequest to the Charging Station.<br>**3.** If the Charging Station has a reservation matching the reservationId in the request PDU, it returns the status *Accepted*.<br>~~**4.** If a specific EVSE was reserved for this reservation, the Charging Station sends a NotifyEventRequest with variable "AvailabilityState" set to "Available" for all the Connectors of that EVSE.~~<br>**4.** If needed, the Charging Station sends NotifyEventRequest with variable "AvailabilityState" set to "Available" for all the Connectors of EVSEs that became available.<br>**5.** The CSMS responds with a NotifyEventResponse to the Charging Station.<br>**6.** The reservation is canceled. |
| | | [...] |

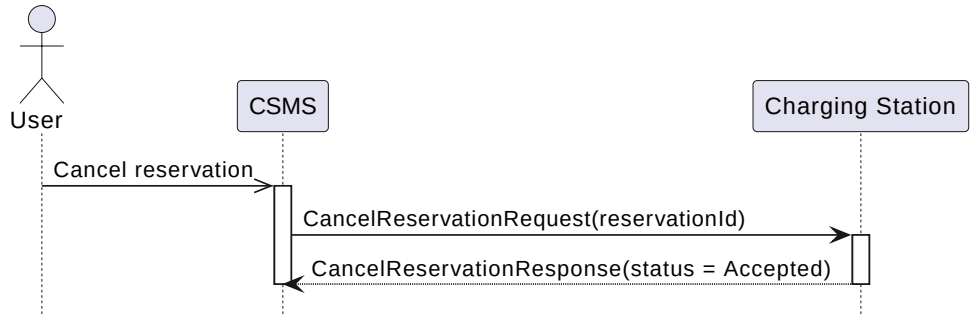
*Removed details from sequence diagram*

Old:




*Sequence Diagram: Cancel Reservation*

New:




*Sequence Diagram: Cancel Reservation*

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| H02.FR.03 | H02.FR.02 AND<br>If a specific EVSE was reserved for this reservation | The Charging Station SHALL allow charging again on this EVSE. |
| H02.FR.04 | H02.FR.03 | The Charging Station SHALL send a StatusNotificationRequest with status *Available* or a NotifyEventRequest with *AvailabilityState* set to *Available* to the CSMS for each connector, notifying the CSMS that all the connectors of this EVSE are available again for any EV Driver. |
| H02.FR.05 | H02.FR.02 AND<br>If no specific EVSE was reserved for this reservation | The Charging Station SHALL allow charging on all EVSE which were not reserved explicitly. |
| H02.FR.06 | H01.FR.05 AND<br>before cancelling the reservation the amount of EVSEs reserved was equal to the amount of reservations | The Charging Station SHALL send for all connectors of all EVSEs which were not reserved explicitly:<br>- a NotifyEventRequest with *component* = "Connector", *variable* = "AvailabilityState", *trigger* = "Delta", *actualValue* = "Available", OR<br>- a StatusNotificationRequest with *connectorStatus* = `Available`. |

# 2.13. Page 290 - (2025-09) - I08.FR.31 is a duplicate requirement number [1042]

By mistake the requirement number I08.FR.31 occurs twice in I08. This has been fixed by moving the first I08.FR.31 requirement to become I08.FR.37.

*Requirement number changed*

| ID. | Precondition | Requirements | Notes |
|---|---|---|---|
| ~~I08.FR.31~~ I08.FR.37 | I08.FR.30 AND Charging Station does not have a Delta monitor installed on TariffCostCtrlr.Problem | Charging Station SHALL send a [notifyeventrequest] with *trigger* = `Alerting`, *eventNotificationType* = `HardWiredNotification`, *component* = "TariffCostCtrlr", *variable* = "Problem", *actualValue* = "true" and *techCode* optionally set to the applicable reason code from Appendix 5, to notify CSMS that it cannot support the *tariff* in the response. | *techCode* can be, for example, one of "TooManyElements", "OutOfMemory", "InternalError", "UnsupportedParam", etc. |

## 2.14. Page 297 - (2025-09) - I12.FR.02 fails to mention that *chargingPeriods* are not sent for running cost updates [1048]

The element *chargingPeriods* in CostDetailsType is not sent for running cost updates, because that is not needed and adds a lot of data to the message. This is mentioned explicitly in CostDetailsType, but it is not formalized in the requirements.

*Changed requirement*

| | ID. | Precondition | Requirements | Notes |
|---|---|---|---|---|
| Old | I12.FR.02 | I12.FR.01 AND TariffCostCtrlr.Enabled[ RunningCost] = true | Charging Station SHALL provide a *costDetails* field of type [cmn_costdetailstype] in [transactioneventrequest] with *eventType* = `Started` and every TariffCostCtrlr.Interval[Cost] seconds during the transaction for *eventType* = `Updated`. | Providing running cost updates needs to be enabled via TariffCostCtrlr.Enabled[RunningCost]. See [configkey-running-cost-enabled] and [configkey-cost-interval]. |
| New | I12.FR.02 | I12.FR.01 AND TariffCostCtrlr.Enabled[ RunningCost] = true | Charging Station SHALL provide a *costDetails* field of type [cmn_costdetailstype] without a *chargingPeriods* field in [transactioneventrequest] with *eventType* = `Started` and every TariffCostCtrlr.Interval[Cost] seconds during the transaction for *eventType* = `Updated`. | Providing running cost updates needs to be enabled via TariffCostCtrlr.Enabled[RunningCost]. See [configkey-running-cost-enabled] and [configkey-cost-interval]. |

## 2.15. Page 354 - (2025-06) - Updated remark of K11

Added sentence to Remarks a new charging profile for an update of external limit can use the same charging profile id.

| No. | Type | Description |
|---|---|---|
| ... | ... | ... |
| **8** | **Remarks** | [...]<br><br>If the external limit is represented by an Absolute or Relative `ChargingStationExternalConstraints` charging profile, then every update of the external limit requires (K11.FR.06) that the existing `ChargingStationExternalConstraints` charging profile is replaced by a new one. This one can use the same chargingProfile.id, however. |

## 2.16. Page 327 - (2025-06) - Updated note of K01.FR.05

Note suggested that ChargingStationExternalConstraints cannot be replaced at all. Updated note to clarify that a ChargingStationExternalConstraints cannot be replaced by CSMS.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | K01.FR.05 | When a SetChargingProfileRequest with an already known ChargingProfile.*id* is received AND the existing ChargingProfile does NOT have chargingProfilePurpose = `ChargingStationExternalConstraints` | The Charging Station SHALL replace the existing ChargingProfile with the one specified. | ChargingStationExternal Constraints profile cannot be replaced. |
| New | K01.FR.05 | When a SetChargingProfileRequest with an already known ChargingProfile.*id* is received AND the existing ChargingProfile does NOT have chargingProfilePurpose = `ChargingStationExternalConstraints` | The Charging Station SHALL replace the existing ChargingProfile with the one specified. | ChargingStationExternal Constraints profile cannot be replaced by CSMS. |

## 2.17. Page 327 - (2025-06) - Add cross-references to K01.FR.06 and K01.FR.39

Requirement K01.FR.06 and K01.FR.39 are similar, but located far apart in the table. It is convenient to add a cross-reference between both.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | K01.FR.06 | When *chargingProfilePurpose* is NOT `TxProfile` | The CSMS SHALL NOT send a ChargingProfile with a *stackLevel* - *chargingProfilePurpose* - *evseId* combination that already exists in another ChargingProfile (with different *id*) on the Charging Station and has an overlapping validity period. | This is to ensure that no two charging profiles with same stack level and purpose can be valid at the same time. |
| New | K01.FR.06 | When *chargingProfilePurpose* is NOT `TxProfile` | The CSMS SHALL NOT send a ChargingProfile with a *stackLevel* - *chargingProfilePurpose* - *evseId* combination that already exists in another ChargingProfile (with different *id*) on the Charging Station and has an overlapping validity period. | This is to ensure that no two charging profiles with same stack level and purpose can be valid at the same time. (See also K01.FR.39) |
| Old | K01.FR.39 | When *chargingProfilePurpose* is `TxProfile` | The CSMS SHALL NOT send a ChargingProfile with a *stackLevel* - *transactionId* combination that already exists in another ChargingProfile (with different *id*) with purpose `TxProfile`. | This is to ensure that no two charging profiles with same stack level and purpose can be valid at the same time. |
| New | K01.FR.39 | When *chargingProfilePurpose* is `TxProfile` | The CSMS SHALL NOT send a ChargingProfile with a *stackLevel* - *transactionId* combination that already exists in another ChargingProfile (with different *id*) with purpose `TxProfile`. | This is to ensure that no two charging profiles with same stack level and purpose can be valid at the same time. (See also K01.FR.06) |

## 2.18. Page 331 - (2025-06) - K01.FR.50 requirement is a SHALL

Physics determines how to convert power to current. This cannot be a "should" requirement, but is a SHALL.

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | K01.FR.49 | When a SetChargingProfileRequest without a value for *numberPhases* is received AND the EVSE is of type AC | The Charging Station SHALL assume numberPhases = 3 as a default value. | |
| New | K01.FR.49 | When a SetChargingProfileRequest without a value for *numberPhases* is received AND the EVSE is of type AC | The Charging Station SHALL assume *numberPhases* = 3 as a default value. | Regions with a single phase network should always provide *numberPhases* = 1, otherwise 3 phases will be assumed. |
| Old | K01.FR.50 | When a SetChargingProfileRequest with a chargingRateUnit = *W* is received AND The ChargingSchedule is used for AC charging | The Charging Station SHOULD calculate the phase current limit via: Current per phase = Power / (Line Voltage * Number of Phases). | The "Line Voltage" used in the calculation is not the measured voltage, but the set voltage for the area (for example, 230 or 110 V). The "Number of Phases" is the numberPhases from the ChargingSchedulePeriod. It is usually more convenient to use chargingRateUnit = *A* for AC charging. |
| New | K01.FR.50 | When a SetChargingProfileRequest with a chargingRateUnit = *W* is received AND The charging profile is used for AC charging | The Charging Station SHALL calculate the phase current limit via: Current per phase = *limit* / (Line Voltage * *numberPhases*), in which *limit* and *numberPhases* are the values from the ChargingSchedulePeriod. | The "Line Voltage" used in the calculation is not the measured voltage, but the set voltage for the area (for example, 230 or 110 V). . ~~The *limit* and *numberPhases* are the values from the ChargingSchedulePeriod.~~ When *numberPhases* is not specified, a value of 3 is assumed (see K01.FR.49). It is usually more convenient to use chargingRateUnit = *A* for AC charging, since in that case the limit does not change depending on number of phases in use. |

## 2.19. Page 332 - (2025-09) - K01.FR.56 is too strict

K01.FR.56 attemps to limit the update rate of persistent profiles, but current requirement prohibits setting profiles on different EVSEs in quick succession.

*Updated requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | K01.FR.56 **(2.1)** | When Charging Station receives a [setchargingprofilerequest] for a [cmn_chargingprofiletype] with a *chargingProfilePurpose* that is to be stored persistently AND the previous [setchargingprofilerequest] for this *chargingProfilePurpose* was less than `ChargingProfileUpdateRateLimit` seconds ago | Charging Station MAY respond with [setchargingprofileresponse] with *status* = `Rejected` and *reasonCode* = "RateLimitExceeded" | See also K01.FR.55 and K01.FR.27. If `ChargingProfileUpdateRateLimit` does not exist, there is no rate limit. |
| New | K01.FR.56 **(2.1)** | When Charging Station receives frequent [setchargingprofilerequest] messages at a rate that threatens to wear out its persistent memory, ~~for a [cmn_chargingprofiletype] with a *chargingProfilePurpose* that is to be stored persistently~~ | Charging Station MAY respond with [setchargingprofileresponse] with *status* = `Rejected` and *reasonCode* = "RateLimitExceeded" | See K01.FR.55 and K01.FR.27 for which charging profiles are persistent. |

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.57 **(2.1)** | K01.FR.56 | Charging Station SHOULD report the duration after which a next update will be accepted, in field *statusInfo.additionalInfo* as "<xx> seconds before retry" | <xx> is the number of seconds after which the next [setchargingprofilerequest] is allowed. |
| K01.FR.58 **(2.1)** | K01.FR.56 | CSMS MAY retry the [setchargingprofilerequest] if still applicable | |
| K01.FR.59 **(2.1)** | K01.FR.57 | CSMS IS RECOMMENDED to use at least the number of seconds in *statusInfo.additionalInfo* as a delay before retrying | This will avoid unnecessary rejections |

## 2.19.1. Page 747 - `ChargingProfileUpdateRateLimit`

This variable has now been deprecated.

## 2.19.2. `ChargingProfileUpdateRateLimit`

**Deprecated**

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | SmartChargingCtrlr | |
| **Variable** | **variableName** | UpdateRateLimit | |
| | **variableAttributes** | **mutability** | ReadOnly |
| | **variableCharacteristics** | **dataType** | integer |

| | |
|---|---|
| **Description** | This configuration key limits how often a persistent charging profile can be updated. It is the minimum duration in seconds between updates of charging profiles of the same *chargingProfilePurpose*. A Charging Station may reject SetChargingProfileRequests that occur too frequently, as per K01.FR.56.<br><br>Note: This configuration variable has been deprecated, because a simple variable with number of seconds between updates does not determine when a Charging Station may reject a message or not. |

# 2.20. Page 332 - (2025-06) - CSMS requirements for useLocalTime, PriorityCharging and others [954]

Requirements have been added for the implicit assumption that CSMS has to support these new features.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| PriorityCharging | | | |
| K01.FR.72 **(2.1)** | | CSMS SHALL support sending a SetChargingProfileRequest with *chargingProfilePurpose* = `PriorityCharging`. | |
| K01.FR.73 **(2.1)** | | CSMS SHALL NOT add a *duration* to a *chargingSchedule* in a *chargingProfile* with *chargingProfilePurpose* = `PriorityCharging`. | |
| Use Local Time / Randomized Delay | | | |
| K01.FR.96 **(2.1)** | | CSMS SHALL support sending a SetChargingProfileRequest with a *chargingSchedule* that contains fields *useLocalTime* = true and/or *randomizedDelay* > 0 | |
| Limit Beyond SoC / Offline validity | | | |
| K01.FR.104 **(2.1)** | | CSMS SHALL support sending a SetChargingProfileRequest with a *maxOfflineDuration* > 0 and *invalidAfterOfflineDuration* = true or false. | |
| K01.FR.105 **(2.1)** | | CSMS SHALL support sending a SetChargingProfileRequest with a *chargingSchedule* with a *limitAtSoC* element. | |

# 2.21. Page 333 - [2025-09] - K01 New requirement for randomizedDelays larger than schedule period [1004]

Each *startPeriod* (except the first one) is increased with a random value at start of the transaction. Remember that a delay in the start of the next period, implies an increase in the length of the current period.

A requirement was missing to define how to deal with the situation where a *randomizedDelay* turns out to be longer than the duration of the *chargingSchedulePeriod* for which the start is randomized. It is important to remember that all randomized *startPeriods* are calculated before the charging profile is used (at submission or start of transaction). If a randomization of the *startPeriod* is longer than the schedule period (i.e. until the randomized start of the next period), then this period is skipped entirely.

This implies that a randomized delay can never become more than the duration between startPeriod(i) and startPeriod(i+1)$_{randomized}$, because at that point the next period is started.

**Example**

**Assume a schedule**

{start: 0, limit: 0}, {start: 300, limit: 1}, {start: 600, limit: 2}, {start: 900, limit: 3}

**Random delays create**

{start: 0, limit: 0}, {start: 300+**753**=1053, limit: 1}, {start: 600+123=723, limit: 2}, {start: 900+87=987, limit: 3}
In this case the second period will be dropped, because it exceeds start of the third period (723).

**The randomized schedule becomes**

{start: 0, limit: 0}, {start: 723, limit: 2}, {start: 987, limit: 3}

*New requirement*

| | | | |
|---|---|---|---|
| K01.FR.97 **(2.1)** | K01.FR.93 AND *startPeriod* + <random delay> of *chargingSchedulePeriod*[i] is greater/equal than *startPeriod* + <random delay> of *chargingSchedulePeriod*[i+1] or greater/equal than *chargingSchedule .duration* | *chargingSchedulePeriod*[i] is skipped from the randomized charging schedule. | *chargingSchedulePeriod*[i] is skipped because its randomized start would take place after the randomized start of *chargingSchedulePeriod*[i+1] or after end of charging schedule. This is basically means that *chargingSchedulePeriod*[i-1] continues until (randomized) start of *chargingSchedulePeriod*[i+1] or until charging schedule end if this is the last period. |

# 2.22. Page 336 - (2025-06) - K02 Updated remark of use case about merging profiles

The description of merging profiles in the remark was not complete. It has been updated to refer to the appropriate requirement.

| No. | Type | Description |
|---|---|---|
| ... | ... | ... |
| **8** | **Remark(s)** | [...]<br><br>The final schedule constraints that apply to a transaction are determined by merging the profiles ~~with purposes *ChargingStationMaxProfile* with the profile *TxProfile* or TxDefaultProfile in case no profile of purpose *TxProfile* is provided. Zero or more of the following ChargingProfile purposes MAY have been previously received from the CSMS: *ChargingStationMaxProfile* or TxDefaultProfile.~~ as described in requirement SC.01 in Chapter 4. Smart Charging Signals to a Charging Station from Multiple Actors .<br><br>[...] |

# 2.23. Page 334 - (2025-09) - Requirement for supported operationMode

A missing requirement that only supported operationModes are accepted, has been added.

*New requirement*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.115 **(2.1)** | When Charging Station receives a [setchargingprofilerequest] with a charging profile that contains a [cmn_chargingscheduleperiodtype] with a value for *operationMode* that is not `ChargingOnly` and not part of the *attributeValue* of [configkey-v2xsupportedoperationmodes] | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and *statusInfo* with *reasonCode* = "InvalidOperationMode" | |

# 2.24. Page 335 - (2025-06) - Requirements for checking operationMode and phases L2/L3

The following requirements have been made explicit from the table in paragraph 3.2 Charging Profile purpose.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| | | OperationMode | |
| K01.FR.111 **(2.1)** | When a charging profile has *chargingProfilePurpose* = `PriorityCharging` or `ChargingStationMaxProfile` | The charging profile SHALL only contain *chargingSchedulePeriods* with *operationMode* = `ChargingOnly` or without *operationMode*. | |
| K01.FR.112 **(2.1)** | When a charging profile has *chargingProfilePurpose* = `ChargingStationExternalLimits` | The charging profile SHALL only contain *chargingSchedulePeriods* with *operationMode* = `ChargingOnly`, `ExternalLimits`, `ExternalSetpoint` or without *operationMode*. | |
| K01.FR.113 **(2.1)** | When a charging profile has *chargingProfilePurpose* = `LocalGeneration` | The charging profile SHALL only contain *chargingSchedulePeriods* with *operationMode* = `ChargingOnly`, `ExternalLimits` or without *operationMode*. | |
| K01.FR.114 **(2.1)** | When Charging Station receives a [setchargingprofilerequest] with a charging profile that does not obey to K01.FR.111, K01.FR.112, K01.FR.113 | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and *statusInfo* with *reasonCode* = "InvalidOperationMode" | |

The following requirements have been added to make explicit when _L2 and _L3 fields can be used.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| | | ISO 15118-20 multi-phase support | |
| K01.FR.140 **(2.1)** | When determining the composite schedule from multiple charging profiles | Charging Station SHALL at each point in time use the lowest value of *numberPhases* for that point in time in all applicable *chargingSchedulePeriods*. | For example, if ChargingStationMaxProfile has *numberPhases* = 1 and TxProfile has *numberPhases* = 3, then the value 1 is used. |
| K01.FR.141 **(2.1)** | When Charging Station receives a [setchargingprofilerequest] that introduces a conflicting value of *phaseToUse* with the schedule periods of other applicable charging profiles | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and a *statusInfo* with *reasonCode* = "PhaseConflict". | For example, if ChargingStationMaxProfile has *phaseToUse* = 1 and TxProfile is submitted with *phaseToUse* = 3, then this will be rejected. |
| K01.FR.142 **(2.1)** | When *v2xChargingParameters* of [notifyevchargingneedsrequest] from Charging Station does not contain *maxChargePower_L2* and/or *maxChargePower_L3* | CSMS SHALL NOT provide values for *limit_L2* and/or *limit_L3* fields in a charging profile. | If EV does not report L2/L3 fields then do not provide separate limits for them. |
| K01.FR.143 **(2.1)** | When CSMS sends a [setchargingprofilerequest] for a *chargingProfilePurpose* that is not `TxProfile` | CSMS SHALL NOT provide values for *limit_L2* and *limit_L3* fields in *chargingSchedulePeriods* of the charging profile | Only a TxProfile is submitted after receiving a NotifyEVChargingNeedsRequest. |
| K01.FR.144 **(2.1)** | ( K01.FR.142 OR K01.FR.143 ) AND Charging Station receives a [setchargingprofilerequest] with values for *limit_L2* and/or *limit_L3* fields in a charging profile | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and a *statusInfo* with *reasonCode* = "PhaseConflict". | |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| K01.FR.145 **(2.1)** | When CSMS sends a [setchargingprofilerequest] of *chargingProfilePurpose* = `TxProfile` | CSMS SHALL NOT provide values for *limit_L2* and/or *limit_L3* fields in a *chargingSchedulePeriod* without providing a value for *limit*. | E.g. *limit_L2/L3* can only exist if *limit* is also provided, because in that case *limit* represents phase L1. |
| K01.FR.146 **(2.1)** | K01.FR.145 AND Charging Station receives a [setchargingprofilerequest] with values for *limit_L2* and/or *limit_L3* fields in a charging profile, but no value for *limit* | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and a *statusInfo* with *reasonCode* = "PhaseConflict". | |
| K01.FR.147 **(2.1)** | In the event that an AC ISO 15118-20 session is ongoing and Charging Station falls back to using Mode 3 PWM communication AND charging profiles are active that specify *limit_L2* and/or *limit_L3* | Charging Station SHALL use the lowest value of *limit*, *limit_L2* and/or *limit_L3* as the *limit* to use for each phase. | The phrase "*limit_L2* and/or *limit_L3*" is used to cater for both 2-phase and 3-phase situations. |

# 2.25. Page 335 - (2025-09) - K01.FR.126 corrected requirement definition

K01.FR.126 was not entirely correct, because *evseSleep* can only occur while in *operationMode* = `Idle`.

*Changed requirement*

| | ID | Precondition | Requirement definition | Note |
|---|---|---|---|---|
| Old | K01.FR.126 **(2.1)** | When Charging Station receives a [setchargingprofilerequest] with *evseSleep* = true AND [configkey-supports-evsesleep] is false or absent | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and optionally with *reasonCode* = "InvalidSchedule". | |
| New | K01.FR.126 **(2.1)** | When Charging Station receives a [setchargingprofilerequest] with a *chargingSchedulePeriod* that has *evseSleep* = true and *operationMode* != 'Idle' ~~AND [configkey-supports-evsesleep] is false or absent~~ | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and optionally with *reasonCode* = "InvalidSchedule". | A request for EVSE to sleep can only occur during *operationMode* `Idle`. See Q10 for *evseSleep* behavior. |

# 2.26. Page 376 - (2025-09) - K15 Added rule for composite schedules in case of multiple charging schedules [1002]

In the theoretical situation that 2 TxProfiles are submitted with different stack levels and multiple charging schedules (which can only be the case for an ISO 15118 session) and, because of different durations of these schedules, parts of each of these schedules will be valid at one point or another, then how is the composite schedule calculated? It is not 3 * 3 composite schedules (all possible combinations), but only 3 composite schedules, because schedule #1 is always combined with schedule #1, #2 with #2 and #3 with #3. Other *chargingProfilePurposes*, like `ChargingStationMaxProfile` need also to be taken into account when calculating the composite schedule.

A new requirement is added to define this behavior.

### 2.26.1. K15 - ISO 15118-2

*New requirement*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K15.FR.22 | When calculating CompositeSchedule(s) to create a SAScheduleList for ISO 15118-2 to send to EV AND multiple ChargingProfileTypes of *chargingProfilePurpose* = `TxProfile` with different *stackLevels* are valid AND some or all these ChargingProfileTypes have more than one *chargingSchedule* | Charging Station SHALL create up to three CompositeSchedules as defined in K08.FR.04, by combining the first *chargingSchedule* with the first *chargingSchedule* of other stack levels, the second with second (if existing), the third with the third (if existing), based on their order in the ChargingProfileTypes. | This is about a corner case when multiple TxProfiles with different stack levels and multiple charging schedules have been sent to the Charging Station. (See K18.FR.24) |

### 2.26.2. K18 - ISO 15118-20 Scheduled Control Mode

*New requirement*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| K18.FR.24 | When calculating CompositeSchedule(s) to create ScheduleTupleTypes for ISO 15118-20 to send to EV AND multiple ChargingProfileTypes of *chargingProfilePurpose* = `TxProfile` with different *stackLevels* are valid AND some or all these ChargingProfileTypes have more than one *chargingSchedule* | Charging Station SHALL create up to three CompositeSchedules as defined in K08.FR.04, by combining the first *chargingSchedule* with the first *chargingSchedule* of other stack levels, the second with the second (if existing), the third with the third (if existing), based on their order in the ChargingProfileTypes. | This is about a corner case when multiple TxProfiles with different stack levels and multiple charging schedules have been sent to the Charging Station. (See K15.FR.22) |

### 2.26.3. K19 - ISO 15118-20 Dynamic Control Mode

This issue does not affect the requirements in K19, because in Dynamic Control Mode only a single charging schedule is offered by CSMS.

## 2.27. Page 350 - (2025-06) - GetCompositeSchedule and L2/L3 values

The following clarifies that a composite schedule only needs to report L2/L3 values when they exist in the applicable charging profiles.

*New requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| K08.FR.09 **(2.1)** | K08.FR.02 AND a *chargingSchedulePeriod* in the applicable charging profiles contains *limit_L2* and/or *limit_L3* values | Charging Station SHALL report the composite value for *limit_L2* and/or *limit_L3* values in the resulting *chargingSchedulePeriod* of the [getcompositescheduleresponse]. |

## 2.28. Page 356 - (2025-06) - Updated note of K11.FR.06 with MaxExternalConstraintsId

| | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | K11.FR.06 | When an external charging limit/schedule is received | The Charging Station SHALL use purpose `ChargingStationExternalConstraints` when reporting about this limit (i.e. in a ReportChargingProfilesRequest). | It is RECOMMENDED to use negative values for the *id* of a `ChargingStationExternalConstraints` profile, to minimize the risk of a clash with an *id* that CSMS might use for a (future) charging profile.<br>See use case K29 for the use of `Dynamic` charging profiles and external limits. |
| New | K11.FR.06 | When an external charging limit/schedule is received | The Charging Station SHALL use purpose `ChargingStationExternalConstraints` when reporting about this limit (i.e. in a ReportChargingProfilesRequest). | When configuration variable `MaxExternalConstraintsId` exists, it is RECOMMENDED to use values for the *id* of a `ChargingStationExternalConstraints` profile below this value, to minimize the risk of a clash with an *id* that CSMS might use for a (future) charging profile.<br>When configuration variable `MaxExternalConstraintsId` does not exist, it is RECOMMENDED to use negative values for the *id* of a `ChargingStationExternalConstraints` profile, to minimize the risk of a clash with an *id* that CSMS might use for a (future) charging profile.<br>See use case K29 for the use of `Dynamic` charging profiles and external limits. |

## 2.29. Page 376 - (2025-09) - K16 use case description update

The use case description refers to SetChargingProfile in step 7, but that is too restricting. It is the composite schedule that is provided to EV.

| No. | Type | Description |
|---|---|---|
| ... | ... | ... |

| No. | Type | Description |
|---|---|---|
| | *Scenario description* | **1** CSMS sends a SetChargingProfileRequest to the Charging Station.<br>**2** Charging Station responds with a SetChargingProfileResponse to the CSMS.<br>**3** When EV sends the next CurrentDemandReq (for DC) or ChargingStatusReq (for AC), the Charging Station will respond with *evseNotification* = `ReNegotiation`.<br>**4** EV sends a PowerDeliveryReq with *chargeProgress* = `ReNegotiate` to confirm this.<br>**5** Charging Station responds with a PowerDeliveryRes.<br>**6** EV sends a ChargeParameterDiscoveryReq.<br>**7** Charging Station responds with a ChargeParameterDiscoveryRes with an SAScheduleList that contains the composite schedule(s) for the EVSE ~~ChargingSchedule data from the SetChargingProfileRequest~~.<br>**8** EV sends a PowerDeliveryReq with *chargeProgress* = `Start` (with an optional charging profile) to confirm this.<br>**9** Charging Station responds with PowerDeliveryRes and, if charging was suspended at start of the renegotiation, will resume power delivery.<br>**10** If EV provided a charging profile in the previous step, then Charging Station will send a NotifyEVChargingScheduleRequest to the CSMS. |
| ... | ... | ... |

## 2.29.1. Page 377

| | ID | Precondition | Requirements | NOTE |
|---|---|---|---|---|
| Old | K16.FR.02 **(2.1)** | K16.FR.01 | Charging Station SHALL initiate schedule renegotiation with EV. | In ISO 15118-**2** this is done by replying with EVSENotification=`ReNegotiation` to a CurrentDemandReq (for DC) or ChargingStatusReq (for AC) message.<br>In ISO 15118-20 this is done by replying with EVSENotification=`ScheduleRenegotiation` in ChargeLoopRes. |
| New | K16.FR.02 **(2.1)** | When the composite schedule for the EVSE changes | Charging Station SHALL initiate schedule renegotiation with EV. | This can be caused by a Set/ClearChargingProfileRequest or a change in ChargingStationExternalConstraints/LocalGeneration charging profiles.<br>In ISO 15118-**2** renegotiation is started by replying with EVSENotification=`ReNegotiation` to a CurrentDemandReq (for DC) or ChargingStatusReq (for AC) message.<br>In ISO 15118-**20** this is done by replying with EVSENotification=`ScheduleRenegotiation` in ChargeLoopRes. |
| Old | K16.FR.03 | K16.FR.02 | Charging Station SHALL provide the ChargingSchedule data to the EV. | In ISO 15118 this is done in the ChargeParameterDiscoverRes message. |
| New | K16.FR.03 | K16.FR.02 | Charging Station SHALL provide the composite schedule(s) ~~ChargingSchedule data~~ to the EV. | In ISO 15118 this is done in the ChargeParameterDiscoverRes message. |

# 2.30. Page 475 - (2025-06) - O01 - Added missing requirements

Added missing requirements explicitly specifying behaviour of Charging Station it contains one or more displays.

*New requirements*

| ID | Precondition | Requirement definition |
|---|---|---|
| *Multiple Display support* | | |

| ID | Precondition | Requirement definition |
|---|---|---|
| O01.FR.20 | When Charging Station has multiple displays<br>AND<br>Charging Station receives a [setdisplaymessagerequest] without a *display* element in its MessageInfoType | Charging Station SHOULD use the message for the main display(s) |
| O01.FR.21 | When receiving a GetBaseReportRequest AND Charging Station has one or more displays | Charging Station SHOULD include in the report a Display component for every display it contains. |
| O01.FR.22 | When Charging Station receives a [setdisplaymessagerequest] with Display element referencing an unknown Display in its MessageInfoType | Charging Station SHOULD respond with a [setdisplaymessageresponse] with *status* = `Rejected`. |
| O01.FR.23 | When Charging Station receives a [setdisplaymessagerequest] with Display element referencing a known Display in its MessageInfoType | Charging Station SHOULD use the message only for the specified display. |

# 2.31. Page 370 - (2025-06) - K27 Updated remark of use case

The remark of K27 has been improved to clarify the difference between using an Absolute or a Dynamic *chargingProfileKind* for a charging profile with a `LocalGeneration` *chargingProfilePurpose*, and to mention that *chargingProfile.id* for the updates does not change.

| No. | Type | Description |
|---|---|---|
| ... | ... | ... |
| **8** | **Remarks** | If the external system provides a limit via a protocol that is not OCPP, e.g. ModBus, then Charging Station can represent this as an Absolute charging profile, that is replaced when the limit changes, or as a Dynamic charging profile with a single charging schedule period with *operationMode* = `ExternalLimits` in which the *limit* is dynamically updated.<br>It is up to the Charging Station implementation to decide whether to represent the external limits for `LocalGeneration` as an Absolute charging profile that is replaced by a new charging profile with the same *chargingProfile.id* upon each change of the external limit, or as a Dynamic charging profile with an *operationMode* = `ExternalLimits` in which the *limit* is changed upon each change of the external limit. |

# 2.32. Page 389 - (2025-06) - K19.FR.04 Minor rephrasing

K19.FR.04 reads "If the CSMS is not **able** to provide ...". This suggests that it may be caused by an error condition, but it can be a conscious choice to not provide a charging profile. Changed "able" to "going" to make this clear.

| | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | K19.FR.04 | K19.FR.02 | If the CSMS is not able to provide a charging schedule, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to `NoChargingProfile`. | (Note, *status* value differs from K15.FR.04).<br>Charging Station will use a TxDefaultProfile or provide a schedule with unlimited power. |
| New | K19.FR.04 | K19.FR.02 | If the CSMS is not going to provide a charging schedule, it SHALL indicate this by setting the *status* field in the NotifyEVChargingNeedsResponse to `NoChargingProfile`. | (Note, *status* value differs from K15.FR.04).<br>Charging Station will use a TxDefaultProfile or provide a schedule with maximum power of EVSE. |

# 2.33. Page 395 - (2025-06) - CSMS requirement to support UsePriorityCharging

A CSMS must support priority charging. This has been added as a requirement.

*New requirement*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| K21.FR.10 | | CSMS SHALL support sending a [useprioritychargingrequest] | A Charging Station reports support for this in SmartChargingCtrlr.SupportedAdditionalPurposes. |

# 2.34. Page 396 - (2025-06) - 5.5 Dynamic Charging Profile [882]

The following paragraphs are added to clarify use of *duration* field.

## Duration in dynamic charging profiles

The field *duration* of ChargingScheduleType limits the maximum duration of a charging schedule. A dynamic charging profile consists of a charging schedule with only a single period. If no *duration* is given in the charging schedule, this period is valid indefinitely, and the limits only change when updated via an UpdateDynamicScheduleRequest or PullDynamicScheduleUpdateResponse message or by an external system.

If a value for *duration* is given, then the charging schedule will end if no update for the limits has been received for more than *duration* seconds since the last update. This mechanism can be used to ensure that a dynamic charging profile that depends on regular limit updates from CSMS or an external system, will cease to be used when no updates are received anymore, e.g. because connection to the CSMS or external system has been lost.

# 2.35. Page 397 - (2025-06) - K28 missing requirement about duration [882]

| No. | Type | Description |
|---|---|---|
| **1** | **Name** | Dynamic charging profiles from CSMS |
| … | | |
| | *Scenario description #1* | *Updates sent by CSMS*<br>…<br>5. If *chargingSchedule.duration* is set and the *setpoint/limit* is not updated by a [updatedynamicschedulerequest] after *duration* seconds, then the *chargingSchedule* ends and Charging Station will fall back to the next valid charging profile.<br>a. If *chargingSchedule.duration* has not been set, then the *chargingSchedule* is valid indefinitely, until the charging profile is cleared or replaced by CSMS. |
| | *Scenario description #2* | *Updates requested by Charging Station*<br>…<br>6. If *chargingSchedule.duration* is set and the *setpoint/limit* is not updated by [pulldynamicscheduleupdateresponse] after *duration* seconds, then the *chargingSchedule* ends and Charging Station will fall back to the next valid charging profile.<br>a. If *chargingSchedule.duration* has not been set, then the *chargingSchedule* is valid indefinitely, until the charging profile is cleared or replaced by CSMS. |
| … | | |

The following requirements have been copied from Q05 to K28, because they are generic.

*New requirements*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|

| | | | |
|---|---|---|---|
| K28.FR.13 | When a ChargingProfileType has *chargingProfileKind* = `Dynamic` AND *chargingSchedule.duration* is set AND current time > (*chargingSchedule.duration* + *dynUpdateTime*) | Charging Station SHALL consider the charging profile invalid and switch to using the next valid charging profile. | This is a fallback when CSMS is no longer responding within time set by *duration*. |
| K28.FR.14 | K28.FR.13 AND Charging Station receives an update for *limit* or *setpoint* from CSMS via [updatedynamicschedulerequest] or [pulldynamicscheduleupdateresponse] | Charging Station SHALL consider the charging profile eligible again as a valid profile. | This means the charging profile is valid again when a new update is received, assuming there is no other charging profile of higher stack level. |

Having K28.FR.07 as precondition in K28.FR.09 is not correct.

*Updated requirement*

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | K28.FR.09 | K28.FR.06 OR K28.FR.07 OR K28.FR.08 | Charging Station SHALL set *dynUpdateTime* to current time. | |
| New | K28.FR.09 | K28.FR.06 OR ~~K28.FR.07 OR~~ K28.FR.08 | Charging Station SHALL set *dynUpdateTime* to current time. | |

# 2.36. Page 399 - (2025-09) - K28.FR.10 Precondition not complete

Pulling a new schedule is, of course, only required when the *dynUpdateInterval* has elapsed.

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | K28.FR.10 | When *chargingProfileKind* = `Dynamic` and *dynUpdateInterval* > 0 in *chargingProfile* | Charging Station SHALL send a [pulldynamicscheduleupdaterequest] with *chargingProfileId* = *chargingProfile.id* to request an update of the *chargingSchedulePeriod*. | |
| New | K28.FR.10 | When *chargingProfileKind* = `Dynamic` and *dynUpdateInterval* > 0 in *chargingProfile* AND *dynUpdateTime* + *dynUpdateInterval* >= <current time> | Charging Station SHALL send a [pulldynamicscheduleupdaterequest] with *chargingProfileId* = *chargingProfile.id* to request an update of the *chargingSchedulePeriod*. | |

*New requirement*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| K28.FR.15 | When a [cmn_chargingscheduletype] of a [cmn_chargingprofiletype] with *chargingProfileKind* = `Dynamic` contains the field *duration* AND current time > (*chargingSchedule.duration* + *dynUpdateTime*) | Charging Station SHALL consider the charging profile invalid and switch to using the next valid charging profile. | Field *duration* defines how long a the charging schedule remains valid after receipt of a [updatedynamicschedulerequest] or [pulldynamicscheduleupdateresponse]. |

# 2.37. Page 400 - (2025-06) - K29 missing requirement about duration [882]

The scenarios have been updated to show how to deal with a limited *duration* of a dynamic charging profile.

| No. | Type | Description |
|---|---|---|
| **1** | **Name** | Dynamic charging profiles from external system |
| … | | |
| | *Scenario description #1* | *Charging profile from external system with dynamic updates*<br><br>…<br><br>5. If *chargingSchedule.duration* is set and the *setpoint/limit* is not updated by External System after *duration* seconds, then the *chargingSchedule* ends and Charging Station will fall back to the next valid charging profile.<br>a. If *chargingSchedule.duration* has not been set, then the *chargingSchedule* is valid indefinitely, until the charging profile is cleared or replaced by External System. |
| | *Scenario description #2* | *Charging profile from CSMS with dynamic updates from external system*<br><br>…<br><br>5. If *chargingSchedule.duration* is set and the *setpoint/limit* is not updated by External System after *duration* seconds, then the *chargingSchedule* ends and Charging Station will fall back to the next valid charging profile. #<br>a. If *chargingSchedule.duration* has not been set, then the *chargingSchedule* is valid indefinitely, until the charging profile is cleared or replaced by CSMS.<br><br>… |
| **8** | **Remarks** | …<br>It is advised to have a charging profile with a lower stack level present to fall back to, in case the dynamic charging profile is invalidated, because no update is provided within *duration* seconds. |

The following requirements have been copied from Q05 to K29, because they are generic.

*New requirements*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| K29.FR.07 | When a ChargingProfileType has *chargingProfileKind* = `Dynamic` AND *chargingSchedule.duration* is set AND current time > (*chargingSchedule.duration* + *dynUpdateTime*) | Charging Station SHALL consider the charging profile invalid and switch to using the next valid charging profile. | This is a fallback when CSMS or External System is no longer responding within time set by *duration*. |
| K29.FR.08 | K29.FR.07 AND<br>Charging Station receives an update for *limit* or *setpoint* from External System | Charging Station SHALL consider the charging profile eligible again as a valid profile. | This means the charging profile is valid again when a new update is received, assuming there is no other charging profile of higher stack level. (See also K29.FR.03) |

# 2.38. Page 401 - (2025-06) - K29.FR.04: updated precondition to using dynamic profiles

K29.FR.04 only applies when Charging Station intends to use a dynamic charging profile. This is reflected by adding K29.FR.05 in precondition.

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|

| | | | | |
|---|---|---|---|---|
| Old | K29.FR.04 | NOT K29.FR.03 AND [configkey-external-constraints-profile-disallowed] is false or absent<br>AND<br>An external system provides a current or power limit (i.e. single value, not a schedule) | Charging Station SHALL represent this as a [cmn_chargingprofiletype] with a single *chargingSchedulePeriod*, and having a *chargingProfilePurpose* = `ChargingStationExternalConstraints` with a *chargingProfileKind* = `Dynamic`. | The alternative, using a *chargingProfileKind* = `Absolute`, is described in K11.FR.06. |
| New | K29.FR.04 | NOT K29.FR.03 AND<br>K29.FR.05 AND<br>[configkey-external-constraints-profile-disallowed] is false or absent<br>AND<br>An external system provides a current or power limit (i.e. single value, not a schedule) | Charging Station SHALL represent this as a [cmn_chargingprofiletype] with a single *chargingSchedulePeriod*, and having a *chargingProfilePurpose* = `ChargingStationExternalConstraints` with a *chargingProfileKind* = `Dynamic`. | The alternative, using a *chargingProfileKind* = `Absolute`, is described in K11.FR.06. |

## 2.39. Page 402 - (2025-06) - K29.FR.05: Setpoint missing in precondition

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | K29.FR.05 | When external system updates a limit AND<br>Charging Station represents this as a `Dynamic` charging profile | Charging Station SHALL update the *limit* or *setpoint* in this charging profile. | |
| New | K29.FR.05 | When external system updates a limit or setpoint AND<br>Charging Station represents this as a `Dynamic` charging profile | Charging Station SHALL update the *limit* or *setpoint* in this charging profile. | |

## 2.40. Page 447 - (2025-09) - N01.FR.12 - Improved definition

Updated requirement definition to clarify the `AcceptedCanceled` status.

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old | N01.FR.12 | When a Charging Station is assembling or uploading the log file AND<br>the Charging Station receives a new GetLogRequest | The Charging Station SHOULD cancel the ongoing log file upload AND respond with status `AcceptedCanceled`. |
| New | N01.FR.12 | When a Charging Station is assembling or uploading the log file AND<br>the Charging Station receives a new GetLogRequest | The Charging Station SHOULD cancel the ongoing log file upload AND respond GetLogResponse with status `AcceptedCanceled`. |

## 2.41. Page 449 - (2025-09) - N02: changed empty to absent.

A number of requirements previously stated "empty" when they should have indicated "absent." For example, the phrases referring to monitoringCriteria and componentVariables being "empty" are incorrect. These arrays cannot be empty; they must be absent instead. This correction has been applied to all occurrences throughout section N02.

## 2.42. Page 450 - (2025-06) - N02.FR.13/23 monitoringCriteria DeltaMonitoring is used for TargetDelta [895]

There is no monitoring criteria TargetDeltaMonitoring. That is just DeltaMonitoring.

*Change requirement*

| | ID | Precondition | Requirement definition |
|---|---|---|---|
| Old | N02.FR.13 | If *monitoringCriteria* contains `DeltaMonitoring` | All monitors with *type* = `Delta` are reported. |
| New | N02.FR.13 | If *monitoringCriteria* contains `DeltaMonitoring` | All monitors with *type* = `Delta`, `TargetDelta` and `TargetDeltaRelative` are reported. |

*Deleted requirement*

| ID | Precondition | Requirement definition |
|---|---|---|
| ~~N02.FR.23 (2.1)~~ | If *monitoringCriteria* contains `TargetDeltaMonitoring` | All monitors with *type* = `TargetDelta` and *type* = `TargetDeltaRelative` are reported. |

# 2.43. Page 720 - (2025-06) - New configuration variable to allow TLS wildcard certificates

*New configuration key*

## `AllowCSMSTLSWildcards`

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | SecurityCtrlr | |
| **Variable** | **variableName** | AllowCSMSTLSWildcards | |
| | **variableAttributes** | **mutability** | ReadWrite |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | This variable allows a Charging Station to support non-compliant OCPP behavior and connect to a CSMS that uses a wildcard TLS server certificate for the OCPP connection.<br>If this variable is present it SHALL be ReadWrite. If this variable is not implemented or has value false, the OCPP-compliant behavior is that a Charging Station rejects a connection from a CSMS that presents a wildcard certificate.<br>It is highly RECOMMENDED to not allow wildcard certificates. | | |

# 2.44. Page 739 - (2025-09) - Error in description of AlignedData interval variables [1043]

The Interval and TxEndedInterval variables of AlignedDataCtrlr mention an incorrect time and duration format (ISO8601) that is not supported by OCPP.

## 2.44.1. `AlignedDataInterval`

| | |
|---|---|
| ... | ... |
| **Description** | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the MeterValuesRequest or TransactionEventRequest message. 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.<br>~~When clock-aligned data is being transmitted, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard.~~<br>A value of "0" (numeric zero), by convention, is to be interpreted to mean that no clock-aligned data should be transmitted. |

## 2.44.2. `AlignedDataTxEndedInterval`

| ... | ... |
|---|---|
| **Description** | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. 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.<br>~~When clock-aligned data is being collected, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard.~~ All intervals are transmitted (if so enabled) at the end of the transaction in 1 TransactionEventRequest (eventType = Ended) message.<br>This is not a recommended practice, since the size of the message can become very large. |

## 2.44.3. `AlignedDataUpstreamInterval`

*New in OCPP 2.1*

| ... | ... |
|---|---|
| **Description** | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the MeterValuesRequest message for location `Upstream` only. 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.<br>~~When clock-aligned data is being transmitted, the interval in question is identified by the start time and (optional) duration interval value, represented according to the ISO8601 standard. All "per-period" data (e.g. energy readings) should be accumulated (for "flow" type measurands such as energy), or averaged (for other values) across the entire interval, and transmitted (if so enabled) at the end of each interval, bearing the interval start time timestamp. +~~ A value of "0" (numeric zero), by convention, is to be interpreted to mean that no clock-aligned data should be transmitted. |

# 2.45. Page 492 - (2025-09) - Text instances of dischargingLimit instead of dischargeLimit

There were many instance of the word *dischargingLimit* being used in text or requirements of section Q, instead of the correct word *dischargeLimit*. This has been fixed globally.

# 2.46. Page 499 - (2025-06) - Additional V2X generic requirements

The phase-related requirement from K01 also apply to discharging and setpoint parameters in block Q.

*New requirements*

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| V2X.06 | | Any requirements for *limit*, *dischargingLimit*, *setpoint* and *setpointReactive* also apply to their equivalents with postfix _L2 and _L3 | |
| V2X.07 | | The postfixed L2 and L3 variants of *limit*, *dischargingLimit*, *setpoint* and *setpointReactive* can only occur in combination with the associated field without the postfix. | See K01.FR.145 |

| ID | Precondition | Requirement definition | Note |
|---|---|---|---|
| V2X.08 | When *v2xChargingParameters* of [notifyevchargingneedsrequest] from Charging Station does not contain *maxDischargePower_L2* and/or *maxDischargePower_L3* | CSMS SHALL NOT provide values for *dischargeLimit_L2*, *dischargeLimit_L3*, *setpoint(Reactive)_L2* or *setpoint(Reactive)_L3* fields in a charging profile. | If EV does not report L2/L3 fields then do not provide separate limits for them. See K01.FR.142. |
| V2X.09 | When CSMS sends a [setchargingprofilerequest] for a *chargingProfilePurpose* other than `TxProfile` | CSMS SHALL NOT provide values for *dischargeLimit_L2*, *dischargeLimit_L3*, *setpoint(Reactive)_L2* or *setpoint(Reactive)_L3* fields in *chargingSchedulePeriods* of the charging profile | Only a TxProfile is submitted after receiving a NotifyEVChargingNeedsRequest. See K01.FR.143. |
| V2X.10 | ( V2X.08 OR V2X.09 )<br>AND<br>Charging Station receives a [setchargingprofilerequest] with values for *dischargeLimit_L2*, *dischargeLimit_L3*, *setpoint(Reactive)_L2* or *setpoint(Reactive)_L3* fields in a charging profile | Charging Station SHALL respond with [setchargingprofileresponse] with *status* = `Rejected` and a *statusInfo* with *reasonCode* = "PhaseConflict". | See K01.FR.144. |

# 2.47. Page 503 - (2025-06) - Q01.FR.05 Precondition needs to refer to ISO15118.ServiceRenegotiationSupport

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q01.FR.05 | Q01.FR.04 | Charging Station SHOULD start a service renegotiation with EV for a different energy transfer service | This situation should not occur when an energy transfer is selected from the *allowedEnergyTransfer* list in the AuthorizeResponse. |
| New | Q01.FR.05 | Q01.FR.04 AND ISO15118.ServiceRenegotiationSupport = true | Charging Station SHALL start a service renegotiation with EV for a different energy transfer service and send a new NotifyEVChargingNeedsRequest | This situation should not occur when an energy transfer is selected from the *allowedEnergyTransfer* list in the AuthorizeResponse. |

# 2.48. Page 504 - (2025-09) - Q01.FR.02 Enhanced precondition to apply only for V2X

The need to send an EVCCID only applies for stations that are able to switch to V2X. This has been added to precondition.

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q01.FR.02 | When Charging Station starts an ISO 15118-20 transaction | Charging Station SHALL add EVCCID to *idToken* in [transactioneventrequest](*eventType*=`Started`) in *idToken.additionalInfo.additionalIdToken* and with *idToken.additionalInfo.type* set to "EVCCID". | This transaction may become bidirectional. This is needed in case CSMS uses the EVCCID of vehicle to decide whether to allow V2X. |

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| New | Q01.FR.02 | When Charging Station's ISO15118Ctrlr.Enabled = true and V2XChargingCtrlr.Enabled = true AND ~~When~~ Charging Station starts an ISO 15118-20 transaction | Charging Station SHALL add EVCCID to *idToken* in [transactioneventrequest](*eventType*=`Started`) in *idToken.additionalInfo.additionalIdToken* and with *idToken.additionalInfo.type* set to "EVCCID". | This transaction may become bidirectional. This is needed in case CSMS uses the EVCCID of vehicle to decide whether to allow V2X. |

# 2.49. Page 504 - (2025-09) - Q01.FR.07 Clarified difference Accepted and Processing

The difference between status Accepted and Processing was not mentioned in requirement, because that is described in use cases K18 and K19. As such, the requirement Q01.FR.07 could have been omitted in Q01, but since it is so essential to the use case flow, it was added to the requirements table. This errata clarifies the difference between Accepted and Processing and refers to K18 and K19.

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q01.FR.07 | If CSMS accepts the *requestedEnergyTransfer* | CSMS SHALL respond with a [notifyevchargingneedsresponse] with *status* = `Accepted` or `Processing`. | Charging station can expect to receive a charging profile immediately or soon. |
| New | Q01.FR.07 | If CSMS accepts the *requestedEnergyTransfer* | CSMS SHALL respond with a [notifyevchargingneedsresponse] with *status* = `Accepted` if able to provide a charging profile immediately or `Processing` if more time is needed to provide a charging profile. | See requirements K18/19.FR.03 and K18/19.FR.05. Charging station can expect to receive a charging profile immediately or soon. |

# 2.50. Page 504 - (2025-09) - Q01.FR.08 Improved precondition

Q01.FR.08 had Q01.FR.01 as precondition, but they were contradicting each other about being able and not able to determine *allowedEnergyTransfer*. Relevant part of Q01.FR.01 precondition has been added to Q01.FR.08.

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q01.FR.08 | Q01.FR.01 AND CSMS is not able to determine a list of *allowedEnergyTranfer* before sending the [authorizeresponse] | CSMS SHALL omit *allowedEnergyTransfer* from [authorizeresponse]. | This can happen if it could not be determined within the short time span before the response has to be returned, e.g. because a third party has to be requested for permission. |
| New | Q01.FR.08 | ~~Q01.FR.01 AND~~ Charging Station's ISO15118Ctrlr.Enabled = true and V2XChargingCtrlr.Enabled = true AND CSMS receives an [authorizerequest] AND CSMS is not able to determine a list of *allowedEnergyTranfer* before sending the [authorizeresponse] | CSMS SHALL omit *allowedEnergyTransfer* from [authorizeresponse]. | This can happen if it could not be determined within the short time span before the response has to be returned, e.g. because a third party has to be requested for permission. |

# 2.51. Page 503 - (2025-06) - Q01.FR.09 Wrong precondition

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q01.FR.09 | Q01.FR.20 | Charging Station SHALL send a [notifyevchargingneedsrequest] with *evseId* set to the EVSE used for this transaction and *requestedEnergyTransfer* set to its default energy transfer (charging only AC/DC) and *availableEnergyTransfer* set to the supported energy transfers. | Depending on type of EVSE this will be AC_single_phase, AC_two_phase, AC_three_phase or DC, DC_ACDP |
| New | Q01.FR.09 | Q01.FR.08 | Charging Station SHALL send a [notifyevchargingneedsrequest] with *evseId* set to the EVSE used for this transaction and *requestedEnergyTransfer* set to its default energy transfer (charging only AC/DC) and *availableEnergyTransfer* set to the supported energy transfers. | Depending on type of EVSE this will be AC_single_phase, AC_two_phase, AC_three_phase or DC, DC_ACDP |

## 2.52. Page 504 - (2025-06) - Q02 Use case text not in line with Q02.FR.03

| No. | Type | Description |
|---|---|---|
| **1** | **Name** | Starting in operationMode ChargingOnly before enabling V2X |
| ... | | |
| | *Scenario description* | 1. The Charging Station sends a [authorizerequest] with EVCCID of EV in *additionalInfo* of *idToken*.<br>a. The CSMS cannot (yet) allow V2X and returns an [authorizeresponse] with *idTokenInfo.status* = `Accepted` and omits the field *allowedEnergyTransfer*.<br>... |
| ... | | |

## 2.53. Page 513 - (2025-06) - Q05 add requirement about duration [822]

Requirement from K28 has been added that charging profile becomes valid again after an update of limit is received.

*New requirement*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| Q05.FR.08 | Q05.FR.07 AND<br>Charging Station receives an update for *limit*, *dischargingLimit* or *setpoint* from External System | Charging Station SHALL consider the charging profile eligible again as a valid profile. | This means the charging profile is valid again when a new update is received, assuming there is no other charging profile of higher stack level. |

## 2.54. Page 514 - (2025-06) - Prerequisite in use case Q06 updated

The use case Q06 contains a prerequisite about TxProfile or TxDefaultProfile which does not belong here.

| No. | Type | Description |
|---|---|---|
| **1** | **Name** | External V2X control with a charging profile from an External System |
| ... | | |

| | | |
|---|---|---|
| 5 | **Prerequisites** | ~~For discharging, at least one of the active charging sessions must have an active `TxProfile` or `TxDefaultProfile` for V2X operations.~~<br>Configuration variable [configkey-external-control-signals-enabled] = true.<br>Configuration variable [configkey-external-constraints-profile-disallowed] = false or absent. |
| … | | |

# 2.55. Page 516 - (2025-06) - Q06.FR.11/12 can be combined [883]

Requirements Q06.FR.11 and Q06.FR.12 are overlapping and can be combined.

| | ID | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q06.FR.11 | Q06.FR.02 OR Q06.FR.04 | Charging Station SHALL send a [notifycharginglimitrequest] with *chargingLimitSource* = `EMS`, *isDynamic* = true and with the received schedule in *chargingSchedule* to CSMS. | This *chargingSchedule* will only have a single period. |
| New | Q06.FR.11 | ( Q06.FR.02 OR Q06.FR.04 ) AND The value of *limit* , *dischargingLimit* , *setpoint* , *setpointReactive* changes more than SmartChargingCtrlr.LimitChangeSignificance | Charging Station SHALL send a [notifycharginglimitrequest] with *chargingLimitSource* = `EMS`, *isDynamic* = true and with the received schedule in *chargingSchedule* to CSMS. | This *chargingSchedule* will only have a single period. Also applies to L2 and L3 values. |

*Deleted requirement*

| ID | Precondition | Requirements | Note |
|---|---|---|---|
| ~~Q06.FR.12~~ | Q06.FR.07 AND<br>The value of *limit*, *dischargingLimit*, *setpoint*, *setpointReactive* changes more than SmartChargingCtrlr.LimitChangeSignificance | Charging Station SHALL send a [notifycharginglimitrequest] with *chargingLimitSource* = `EMS`, *isDynamic* = true and a schedule with the new values in *chargingSchedule* to CSMS. | Also applies to L2 and L3 values. |

# 2.56. Page 517 - (2025-06) - Q06 add requirement about duration [822]

Requirement from K29 has been added that charging profile becomes valid again after an update of limit is received.

*New requirement*

| **ID** | **Precondition** | **Requirements** | **Note** |
|---|---|---|---|
| Dynamic duration | | | |
| Q06.FR.40 | When a ChargingProfileType has *chargingProfileKind* = `Dynamic` AND *chargingSchedule.duration* is set AND current time > (*chargingSchedule.duration* + *dynUpdateTime*) | Charging Station SHALL consider the charging profile invalid and switch to using the next valid charging profile. | This is a fallback when External System is no longer responding within time set by *duration*. (Same as K29.FR.07) |
| Q06.FR.41 | Q06.FR.40 AND<br>Charging Station receives an update for *limit, dischargingLimit* or *setpoint* from External System | Charging Station SHALL consider the charging profile eligible again as a valid profile. | This means the charging profile is valid again when a new update is received, assuming there is no other charging profile of higher stack level. |

# 2.57. Page 519 - (2025-09) - Q07 Added requirements

Q07 is a special case of CentralSetpoint, but some requirements were added for completeness.

*New requirements*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| OperationMode CentralFrequency | | | |
| Q07.FR.01 | When Charging Station supports centrally controlled frequency support | Charging Station SHALL report the operation mode `CentralFrequency` in [configkey-v2xsupportedoperationmodes] | |
| Q07.FR.02 | When CSMS is providing centrally controlled frequency support via setpoints | CSMS SHALL send a [setchargingprofilerequest] message with *chargingProfileKind* = `Dynamic` and a single *chargingSchedulePeriod* that has *operationMode* = `CentralFrequency`. | |
| Q07.FR.03 | Q07.FR.02 | CSMS SHALL NOT include fields *limit* and *dischargeLimit* in the [cmn_chargingscheduleperiodtype]. | This also includes the L2 and L3 variants of those fields. |
| Q07.FR.04 | Q07.FR.02 | CSMS IS RECOMMENDED to set a *duration* of the *chargingSchedule* to prevent that the schedule remains active indefinitely when CSMS is unable to send any [updatedynamicscheduleRequest] for whatever reason. | |

# 2.58. Page 522 - (2025-09) - Q08.FR.02/12 Requirement updates for aFRR

*Changed requirement*

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | Q08.FR.02 | Q08.FR.01 | The [cmn_chargingscheduleperiodtype] SHALL have a *v2xFreqWattCurve* with at least two [cmn_v2xfreqwattpointtype] elements, and a value for *v2xBaseline*. | |
| New | Q08.FR.02 | Q08.FR.01 | The [cmn_chargingscheduleperiodtype] SHALL have a *v2xFreqWattCurve* with at least two [cmn_v2xfreqwattpointtype] elements, and a value for *v2xBaseline*, and optionally a *v2xSignalWattCurve* with at least to [cmn_v2xsignalwattpointtype] elements. | |

*New requirements*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| Q08.FR.12 | When CSMS receives an aFRR signal from an external actor | CSMS SHALL send a [afrrsignalrequest] with *timestamp* set to current time and *signal* set to value received from external actor. | External actor is, for example, a TSO. |
| Configuration | | | |
| Q08.FR.20 | When Charging Station supports local frequency support | Charging Station SHALL report the operation mode `LocalFrequency` in [configkey-v2xsupportedoperationmodes] | |

# 2.59. Page 550 - (2025-09) - R04 extra requirements to SetDERControlRequest [997]

A default control cannot have a *startTime* or *duration*. Requirements have been added to make this explicit. The mapping of *controlType* to control fields has been made explicit.

*New requirements*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| R04.FR.12 | | CSMS SHALL not send a [setdercontrolrequest] that has *isdefault* = true for a control that has a *startTime* and/or *duration* field | All controls except *enterService* and *gradient* have optional *startTime*, *duration*. |

| | | | |
|---|---|---|---|
| R04.FR.13 | Charging Station receives a [setdercontrolrequest] with *isDefault* = true AND a control that has a *startTime* and/or *duration* field | Charging Station SHALL respond with [setdercontrolresponse] with *status* = `Rejected` | Default controls cannot have a *startTime* or *duration*. |
| R04.FR.14 | | CSMS SHALL not send [setdercontrolrequest] that has *isdefault* = false for *controlType* = `EnterService` or `Gradients` | These only exist as default controls. |
| R04.FR.15 | Charging Station receives a [setdercontrolrequest] with *isDefault* = false AND *controlType* = `EnterService` or `Gradients` | Charging Station SHALL respond with [setdercontrolresponse] with *status* = `Rejected` | |
| R04.FR.16 | | CSMS SHALL only provide in [setdercontrolrequest] the control field that is related to *controlType*, according to the following mapping:<br><br>*fixedPFAbsorb* for `FixedPFAbsorb`,<br>*fixedPFInject* for `FixedPFInject`,<br>*fixedVar* for `FixedVar`,<br>*limitMaxDischarge* for `LimitMaxDischarge`,<br>*freqDroop* for `FreqDroop`,<br>*enterService* for `EnterService`,<br>*gradient* for `Gradients`,<br>*curve* for all other *controlTypes* | |
| R04.FR.17 | Charging Station receives a [setdercontrolrequest] with multiple controls or with a control that does not match *controlType* | Charging Station SHALL respond with [setdercontrolresponse] with *status* = `Rejected` | See R04.FR.16 for mapping of control fields to *controlType*. |

# 2.60. Page 551 - (2025-09) - R04 extra requirements to GetDERControlRequest [998]

Requirement added for the case where only *isDefault* is provided as a parameter.

*New requirement*

| ID. | Precondition | Requirements | Note |
|---|---|---|---|
| R04.FR.37 | NOT R04.FR.30 AND Charging Station receives a [getdercontrolrequest] with a value for *isDefault* and no *controlType* and no *controlId* | Charging Station SHALL return a *status* = `Accepted` and send one or more [reportdercontrolrequest] messages for all controls that match the value of *isDefault*. | This is used to request all default or all scheduled controls at once. |

# 2.61. Page 551 - (2025-09) - R04 updated requirements to ClearDERControlRequest [999]

Requirements R04.FR.41 and R04.FR.42 have been simplified. The way that requirement R04.FR.45 was phrased, it always returns Accepted — even when no matching controls exist. Requirement R04.FR.46 has been added for clearing based on *controlId*.

*Changed requirements*

| | ID. | Precondition | Requirements | Note |
|---|---|---|---|---|
| Old | R04.FR.41 | Charging Station receives a [cleardercontrolrequest] with no *controlId* and with a *controlType* that it supports, but that has not been set at the Charging Station for the specified value of *isDefault* | Charging Station returns a [cleardercontrolresponse] with *status* = `NotFound`. | |
| New | R04.FR.41 | Charging Station receives a [cleardercontrolrequest] ~~with no *controlId* and~~ with a *controlType* ~~that it supports, but~~ that has not been set at the Charging Station for the specified value of *isDefault* | Charging Station returns a [cleardercontrolresponse] with *status* = `NotFound`. | |
| Old | R04.FR.42 | Charging Station receives a [cleardercontrolrequest] with no *controlType* and with a *controlId* that has not been set for the given value of *isDefault* | Charging Station SHALL respond with [cleardercontrolresponse] with *status* = `NotFound`. | |
| New | R04.FR.42 | Charging Station receives a [cleardercontrolrequest] ~~with no *controlType* and~~ with a *controlId* that has not been set for the given value of *isDefault* | Charging Station SHALL respond with [cleardercontrolresponse] with *status* = `NotFound`. | |
| Old | R04.FR.45 | Charging Station receives a [cleardercontrolrequest] with no *controlId* and with a *controlType* that it supports and that is in use | Charging Station SHALL clear all controls that match the value of *isDefault* and *controlType* in the request, and return a [cleardercontrolresponse] with *status* = `Accepted`. | Return default or scheduled messages for *controlType* based on value of *isDefault*. |
| New | R04.FR.45 | Charging Station receives a [cleardercontrolrequest] with no *controlId* and with a *controlType* ~~that it supports and that is in use~~ that has been set for the given value of *isDefault* | Charging Station SHALL clear all controls that match the value of *isDefault* and *controlType* in the request, and return a [cleardercontrolresponse] with *status* = `Accepted`. | Clear default or scheduled messages for *controlType* based on value of *isDefault*. |

*New requirement*

| | | | |
|---|---|---|---|
| R04.FR.46 | Charging Station receives a [cleardercontrolrequest] with a *controlId* that has been set for the given value of *isDefault* | Charging Station SHALL clear the control that matches the value of *controlId* in the request, and return a [cleardercontrolresponse] with *status* = `Accepted`. | |

# 2.62. Page 620 - (2025-06) - ChargingSchedulePeriodType.limit description update

The sentence about allowing negative values has been removed, because that is not in line with requirements.
The meaning of this field in case *chargingRateUnit* = A, was missing.

| Field Name | Field Type | Card. | Description |
|---|---|---|---|
| … | | | |
| **limit** | decimal | 0..1 | Optional. Optional only when not required by the *operationMode*, as in CentralSetpoint, ExternalSetpoint, ExternalLimits, LocalFrequency, LocalLoadBalancing. Charging rate limit during the schedule period, in the applicable *chargingRateUnit*. ~~This SHOULD be a non-negative value; a negative value is only supported for backwards compatibility with older systems that use a negative value to specify a discharging limit.~~ The value is zero or positive.<br>When using *chargingRateUnit* = W, this field represents the sum of the power of all phases, unless values are provided for L2 and L3, in which case this field represents phase L1.<br>When using *chargingRateUnit* = A, this field represents the current on each phase, unless values are provided for L2 and L3, in which case the field represents phase L1. |
| … | | | |

# 2.63. Page 703 - (2025-06) - Controller component PaymentCtrlr added to list

The PaymentCtrlr has been added to the list of controller components.

| Controller Component | Description |
|---|---|
| … | … |
| PaymentCtrlr **(2.1)** | Responsible for configuration relating to a payment terminal. |

# 2.64. Page 750 - (2025-09) - TariffCostCtrlr.Enabled can be ReadOnly [934]

There are good reasons to allow TariffCostCtrlr.Enabled to be a ReadOnly variable. TariffCostCtrlr.Enabled[Cost] (`CostEnabled`) can be set to ReadOnly false when Local Cost Calculation is not supported, and only cost calculation from the CSMS is supported. In this case TariffCostCtrlr.Available[Cost] will also be false.

TariffCostCtrlr.Enabled[Tariff] (`TariffEnabled`) can be set to ReadOnly true when the Charging Station only supports OCPP standardized tariff structures and no proprietary tariff structures. In this case TariffCostCtrlr.Available[Tariff] will be true.

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | TariffCostCtrlr | |
| **Variable** | **variableName** | Enabled | |
| | **variableInstance** | Tariff | |
| | **variableAttributes** | **mutability** | ReadWrite /ReadOnly |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | … | | |

| | | | |
|---|---|---|---|
| **Required** | no | | |
| **Component** | **componentName** | TariffCostCtrlr | |
| **Variable** | **variableName** | Enabled | |
| | **variableInstance** | Cost | |
| | **variableAttributes** | **mutability** | ReadWrite /ReadOnly |
| | **variableCharacteristics** | **dataType** | boolean |
| **Description** | … | | |

# 2.65. Appendix Page 51 - (2025-09) - Added connector type BatterySlot

The generic connector type BatterySlot for battery swap stations has been added to ConnectorEnumStringType.

*New connector type*

| Value | Description |
|---|---|
| bBatterySlot | Slot of a battery swap station to accept battery cartridges (type unspecified) |

*Example representation of a battery swap station in device model*

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Connector** | **1** | **1** | | **ConnectorType** | | **Actual** | **bBatterySlot** | **string** | | **false** |
| Connector | 2 | 1 | | ConnectorType | | Actual | bBatterySlot | string | | false |

# 3. Part 3

*Currently no new errata for OCPP 2.1 part 3.*

# 4. Part 4

## 4.1. Page 16 - (2025-06) - 5.4 Reconnecting - reset backoff wait timer

The RetryBackOffWaitMinimum timer is to be used the first time it tries to connect. A sentence has been added to below paragraph to make it explicit that it needs to be reset after successful connection.

> The first reconnection attempts SHALL be after a back-off time of: `RetryBackOffWaitMinimum` seconds, plus a random value with a maximum of `RetryBackOffRandomRange` seconds. After every failed reconnection attempt the Charging Station SHALL double the previous back-off time, with a maximum of `RetryBackOffRepeatTimes`, adding a new random value with a maximum of `RetryBackOffRandomRange` seconds to every reconnection attempt. After `RetryBackOffRepeatTimes` reconnection attempts, the Charging Station SHALL keep reconnecting with the last back-off time, not increasing it any further. After a successful connection the backoff wait timer SHALL be reset to `RetryBackOffWaitMinimum` seconds.

## 4.2. Page 21 - (2025-06) - 6.3 Connection loss - Allow Local Controller to keep connection open

The sentence in this section was too strict about requiring a Local Controller to always close the connection with its charging stations when the connection with CSMS is lost. The sentence has been updated in order to allow for Local Controller implementations that are able to manage the local charging stations locally (for a limited time) when the connection with CSMS is down.

| | |
|---|---|
| Old text | Whenever one or more WebSocket connections between CSMS and the Local Controller are lost, the Local Controller SHALL close all corresponding WebSockets to the Charging Stations that are connected to it. |
| New text | Whenever one or more WebSocket connections between CSMS and the Local Controller are lost, the Local Controller SHALL close all corresponding WebSockets to the Charging Stations that are connected to it, unless the Local Controller is capable of responding to Charging Station requests, and forwards transaction-related requests to the CSMS once the connection is restored. |