



Table of Contents

Disclaimer	1
Scope	2
Terminology and Conventions	3
0. Part 0	4
1. Part 1	5
2. Part 2	6
2.1. Page 13 - (2026-02) - Updated reference 17 and added new references	6
2.2. Page 24 - (2026-02) - Inclusion of RFC 8441 for HTTP/2	6
2.3. Page 24 - (2026-02) - A00.FR.308 - Additional requirement references	6
2.4. Page 26 - (2026-02) - A00.FR.403 - Additional requirement references	7
2.5. Page 82 - (2026-04) - B12 - Reset with ongoing transaction allows wait for cable disconnection.	7
2.6. Page 27 - (2026-02) - A00.FR.411 - Additional requirement references	9
2.7. Page 37 - (2026-02) - A02.FR.02 - Requirement reference correction	9
2.8. Page 41 - (2026-02) - A03.FR.02 - Requirement reference correction	9
2.9. Page 41 - (2026-02) - A03.FR.02 Only applies in security profile 3 [1167]	9
2.10. Page 140 - (2026-06) - C22.FR.01/02 - clarify "without receipt information" [1286]	10
2.11. Page 283 - (2026-02) - I06.FR.03 - added note about privacy [715]	10
2.12. Page 227 - (2026-06) - New F01.FR.27 and F02.FR.28 for <i>evseld</i> in RequestStartTransactionRequest [1281]	11
2.13. Page 415 - (2026-04) - L01.FR.04 - removed ECSchnorr algorithm option and clarify the signature verification the Charging Station must do	11
2.14. Page 392 - (2026-06) - K18.FR.17/K19.FR.13 InvalidMessageSequence must be InvalidMessageSeq	11
2.15. Page 502 - (2026-02) - Improved description of LocalLoadBalancing	12
2.16. Page 505 - (2026-06) - Update table in Q2.2 Limits and setpoints per operation mode [1180]	15
2.17. Page 531 - (2026-02) - Use case Q09 has improved description of LocalLoadBalancing	16
2.18. Page 533 - (2026-02) - Use case Q09 has improved sequence diagram	17
2.19. Page 534 - (2026-02) - Use case Q09 has improved requirements	19
2.20. Page 554 - (2026-06) - Typo in <i>derChargingParameters</i> field in R02 and R03. [1231]	20
2.21. Page 655 - (2026-06) - LimitMaxDischargeType <i>pctMaxDischargePower</i> should be required [1270]	20
2.22. Page 666 - (2026-02) - Updated description of StreamDataElementType [1160]	21
2.23. Page 687 - (2026-06) - DERControlEnumType value PowerMonitoringMustTrip is not a DER control [1270]	21
2.24. Page 782 - (2026-06) - New optional ISO15118ProtocolEnabled variable [1252]	21
2.24.1. ISO15118ProtocolEnabled	21
2.25. Page 786 - (2026-06) - Updated values ConnectedEV for CHAdeMO [1192]	22
2.26. Page 792 - (2026-02) - New configuration variable AutoThresholds	23
3. Part 3	24
4. Part 4	25
4.1. Page 10 - (2026-02) - Update handling of unknown message type [1163]	25
4.2. Page 14 - (2026-02) - Wrong example of NotifyPeriodicEventStream [1159]	25
4.3. Page 16 - (2026-02) - MessageTypeNotSupported is deprecated [1161]	25
4.4. Page 16 - (2026-02) - Section 4.4 Extension fallback mechanism removed [1163]	25
5. Part 5	26
5.1. Page 20 - (2026-02) - TC_C_100_CS - Test case removed	26
5.2. Page 28 - (2026-06) - TC_E_39_CS - Only applicable for TxStopPoint Authorized	26
5.3. Page 60 - (2026-02) - TC_Q_102_CSMS - Test case removed	26
5.4. Page 61 - (2026-02) - TC_R_101_CS - Test case removed	26
6. Part 6	27
6.1. General	27
6.2. Charging Station	28
6.2.1. Page 23 - (2026-02) - TC_A_20_CS - Added note that the testcase will end when the SetNetworkProfile is rejected ..	28
6.2.2. Page 57 - (2026-04) - TC_B_21_CS - Clarification of charging station reboot behaviour with ongoing transactions. .	28
6.2.3. Page 65 - (2026-04) - TC_B_41_CS - Clarification of charging station reboot behaviour with ongoing transactions. .	29
6.2.4. Page 91 - (2026-02) - TC_B_100_CS - Use active securityProfile level	31
6.2.5. Page 1006 - 1112 - (2026-02) - TC_B_105_CS/TC_B_107_CS/TC_B_111_CS/TC_B_113_CS - Identity included in SetVariablesRequest on component *NetworkConfiguration	31
6.2.6. Page 125 - (2026-02) - TC_C_14_CS - Clarifications and Additional tool validations	32
6.2.7. Page 103/104 - (2026-02) - TC_B_108_CS - Added SetNetworkProfileRequest main steps	32

6.2.8. Page 98 - 113 - (2026-06) - TC_B_105_CS/TC_B_111_CS/TC_B_113_CS -Ensure NetworkConfigurationPriority contains only active configurationSlot	34
6.2.9. Page 105 - 107 - (2026-02) - TC_B_109_CS - Testcase updated.	34
6.2.10. Page 130 - (2026-02) - TC_C_17_CS - Corrected wrong triggerReason [1113].	37
6.2.11. Page 119 - 161 - (2026-04) - C Test cases - Updated configuration state	38
6.2.12. Page 143 - 167 - (2026-04) - TC_C_14_CS/TC_C_45_CS/TC_C_47_CS/TC_C_48_CS/TC_C_49_CS - Removed configuration variable masterPassGroupId	38
6.2.13. Page 147 - (2026-02) - TC_C_38_CS - Removed memory state IdTokenCached [1164]	39
6.2.14. Page 173/174 - (2026-02) - TC_C_26_CS - Clarification of main test and tool validation steps.	39
6.2.15. Page 184 - (2026-02) - TC_C_100_CS - Test case removed	40
6.2.16. Page 185 - (2026-02) - TC_C_103_CS - Test updated	40
6.2.17. Page 209 - (2026-02) - TC_C_121_CS - Added tool validations for triggerReason LimitSet	41
6.2.18. Page 182 - (2026-02) - TC_C_55_CS -Explicit mention of offline transactions only	42
6.2.19. Page 214 - (2026-02) - TC_C_124_CS and TC_C_128_CS to TC_C_130_CS - Corrected requirement reference and removed note.	43
6.2.20. Page 282 - (2026-06) - TC_E_41_CS, TC_E_42_CS, TC_E_50_CS and TC_E_51_CS - retry mechanism applied to TransactionEventRequest after suspending energy transfer on EV side	43
6.2.21. Page 354 - (2026-04) - TC_G_11_CS - Clarity provided regarding sending status notifications of connectors not in use.	44
6.2.22. Page 358 - (2026-04) - TC_G_14_CS - Clarity provided regarding sending status notifications of connectors not in use.	46
6.2.23. Page 363 - (2026-06) - TC_G_17_CS - Updated tool and post scenario validations.	47
6.2.24. Page 292 - (2026-06) - TC_E_101_CS - Reusable state EnergyTransferStarted set in preparations	47
6.2.25. Page 294/295/296 - (2026-02) - TC_E_102_CS - Test case and tool validations corrected.. . . .	48
6.2.26. Page 297/298 - (2026-02) - TC_I_119_CS - Added tool validations for totalCost.total	50
6.2.27. Page 297/298 - (2026-02) - TC_E_103_CS - Test case and tool validation corrections.	51
6.2.28. Page 301/302 - (2026-02) - TC_E_105_CS - Test case and tool validation corrections.	52
6.2.29. Page 303/304 - (2026-02) - TC_E_106_CS - Test case and tool validation corrections	54
6.2.30. Page 305/306 - (2026-02) - TC_E_107_CS - Tool validation clarifications.	54
6.2.31. Page 313/314 - (2026-02) - TC_E_114_CS/TC_E_115_CS - Added main step to check the transaction is not resumed	55
6.2.32. Page 610 - (2026-02) - TC_L_07_CS - Path to non-existent firmware updated	56
6.2.33. Page 421 - (2026-02) - TC_I_107_CS - Used reusable state Authorized	56
6.2.34. Page 424/425 - (2026-02) - TC_I_109_CS - Added reusable state EnergyTransferStarted	57
6.2.35. Page 436 - (2026-02) - TC_I_115_CSMS - Added main steps TransactionEvent TariffChanged	58
6.2.36. Page 455 - (2026-02) - TC_I_121_CS - Incorrect minCost / maxCost validations.	58
6.2.37. Page 484 - 840 - (2026-06) - Renamed Configuration variable	64
6.2.38. Page 525 - (2026-02) - TC_N_52_CS - NotifyEvent steps added.	64
6.2.39. Page 658 - (2026-02) - TC_M_24_CS - GetCertificateStatusRequest is not intended to be sent for the V2G leaf certificate	65
6.2.40. Page 783 - (2026-02) - TC_O_24_CS - Additions to tool validations.	65
6.2.41. Page 567/568/569 - (2026-02) - TC_K_116_CS - Updated limits to more practical ones and aligned chargingProfileIds.	66
6.2.42. Page 587 - (2026-02) - TC_K_125_CS Updated dynUpdateTime tool validations.	68
6.2.43. Page 588 - (2026-02) - TC_K_129_CS to TC_K_135_CS improvements.	68
6.2.44. Page 596 - (2026-02) - TC_K_136_CS - Fixed incorrect timezone usage.	68
6.2.45. Page 590 - (2026-02) - TC_K_130_CS to TC_K_135_CS - Reusable state Booted removed	68
6.2.46. Page 596 - (2026-02) - TC_K_136_CS - Remove timezone configuration	68
6.2.47. Page 664/665 - (2026-02) - TC_M_100_CS - Reduced amount of certificates sent.	69
6.2.48. Page 666 - (2026-02) - TC_M_101_CS - Removed AdditionalRootCertificateCheck note	70
6.2.49. Page 749-752 - (2026-02) - TC_N_105_CS - Use existing monitoring configurations and check the disabled monitoring events are not sent	70
6.2.50. Page 753/754 - (2026-02) - TC_N_106_CS - Use existing monitoring configurations and check the disabled monitoring events are not sent	74
6.2.51. Page 755/756 - (2026-02) - TC_N_108_CS - Use existing monitoring configurations and check the disabled monitoring events are not sent	76
6.2.52. Page 757/758 - (2026-02) - TC_N_109_CS - Use existing monitoring configurations	78
6.2.53. Page 805/806 - (2026-02) - TC_O_101_CS - Manual action and reusable state Authorized added	80
6.2.54. Page 814/815 - (2026-02) - TC_Q_102_CS - Made the testcase more flexible regarding the phases used and	

added step for NotifyEVChargingScheduleRequest	81
6.2.55. Page 820/821 - (2026-02) - TC_Q_107_CS - Added missing main steps to complete the renegotiation flow	82
6.2.56. Page 863/864 - (2026-02) - TC_Q_125_CS - Aligned with other V2X ISO 15115-20 / CHAdeMO testcases	84
6.2.57. Page 809-878 - (2026-02) - TC_Q_1XX_CS - NotifyEVChargingNeeds also needs to be used for V2X with CHAdeMO	86
6.2.58. Page 877 - (2026-02) - TC_Q_130_CS - Updated tool validation requestedEnergyTransfer	88
6.2.59. Page 879 - (2026-06) - TC_R_100_CS - Used reusable state booted	88
6.2.60. Page 881 - (2026-02) - TC_R_101_CS - Test case removed	90
6.2.61. Page 899 - (2026-02) - TC_R_108_CS - Use static curves to trigger, instead of configurable ones	90
6.2.62. Page 902/903 - (2026-02) - TC_S_104_CS - Resolved editorial issues	92
6.2.63. Page 904/905 - (2026-02) - TC_S_105_CS - Updated test case	94
6.2.64. Page 916 - (2026-02) - Reusable state Booted now accepts StatusNotification Unavailable [1162].	95
6.2.65. Page 949/952 - (2026-02) - EVConnectedPreSessionBatterySwapping/EVDisconnectedBatterySwapping - Updated tool validations.	96
6.2.66. Page 1207 - (2026-02) - TC_K_102_CS - Made the testcase useable with a real EV	97
6.3. CSMS	99
6.3.1. Page 1022 - 1024 - (2026-04) - TC_C_47_CSMS/TC_C_48_CSMS/TC_C_49_CSMS - Removed configuration variable masterPassGroupId	99
6.3.2. Page 877/878 - (2026-02) - TC_Q_130_CS - BPT manual actions updated.	99
6.3.3. Page 944 - (2026-02) - Reusable state Iso1511820V2xControlBpt - BPT manual actions updated.	100
6.3.4. Page 1100/1101 - (2026-02) - TC_E_117_CSMS - Updated prerequisites and SetChargingProfileRequest tool validations	101
6.3.5. Page 1006 - (2026-02) - TC_B_105_CSMS - Manual action added.	101
6.3.6. Page 1002 - (2026-02) - TC_B_100_CSMS - Manual action added and tool validations updated.	101
6.3.7. Page 1007/1008 - (2026-02) - TC_B_116_CSMS -	102
6.3.8. Page 1026 - (2026-06) - TC_C_51_CSMS - Corrected OSCP response and removed post scenario validations	103
6.3.9. Page 1038 - (2026-02) - TC_C_119_CSMS / TC_C_120_CSMS / TC_C_125_CSMS / TC_C_126_CSMS - Added missing triggerReason and stoppedReason	104
6.3.10. Page 1046/1047 - (2026-02) - TC_C_131_CSMS - Use idToken from the RequestStartTransactionRequest, which contains the PspRef	104
6.3.11. Page 1046/1048/1049 - (2026-02) - TC_C_131_CSMS/TC_C_132_CSMS/TC_C_133_CSMS - Sharedsecret configuration variable added.	105
6.3.12. Page 1094/1095 - (2026-02) - TC_E_110_CSMS - Added missing Authorization request to initiate transaction. ...	105
6.3.13. Page 1096/1097 - (2026-02) - TC_E_111_CSMS - Added missing Authorization request to initiate transaction. ...	106
6.3.14. Page 1333 - (2026-02) - TC_O_101_CSMS - Updated tool validations.	107
6.3.15. Page 1210 - (2026-02) - TC_K_109_CSMS - Additional test case steps added	107
6.3.16. Page 1304-1306 - (2026-02) - TC_N_102_CSMS/TC_N_103_CSMS - added without userinfo.	108
6.3.17. Page 1336 - (2026-02) - TC_Q_102_CSMS - Test case removed	108
6.3.18. Page 1309 - (2026-02) - TC_N_105_CSMS - Install usable monitor and send a NotifyPeriodicEventStream	109
6.3.19. Page 1339/1340 - (2026-02) - TC_Q_107_CSMS - Test case updated	110
6.3.20. Page 1359 - (2026-02) - TC_R_107_CSMS - Removed NotifyDERStartStopRequest and added ReportDERControlRequest.	112
6.3.21. Page 1365 - 1367 - (2026-02) - TC_S_103_CSMS - Test case updated	113
6.3.22. Page 1383 - (2026-04) - Additional validation added for certificateStatus	116

Disclaimer

Copyright © 2010 – 2026 Open Charge Alliance. All rights reserved.

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

Version History

Version	Date	Description
2026-06	2026-07-13	Includes errata for Part 2, 4, 5 and 6 of OCPP 2.1 Edition 2.
2026-04	2026-05-06	Includes errata for Part 2, 4, 5 and 6 of OCPP 2.1 Edition 2.
2026-02	2026-03-11	Includes errata for Part 2, 4, 5 and 6 of OCPP 2.1 Edition 2.
2025-11	2025-12-03	Includes errata for Part 2 and 4 of OCPP 2.1 Edition 1.
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 2.

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 2026-06 of the errata. The errata of this version are marked with "(2026-06)" 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 "OCP20M", like "[OCP20M-349]".

0. Part 0

Currently no new errata for OCPP 2.1 Edition 2 part 0.

Chapter 1. Part 1

Currently no new errata for OCPP 2.1 Edition 2 part 1.

Chapter 2. Part 2

2.1. Page 13 - (2026-02) - Updated reference 17 and added new references

Special Publication 800-57 updated with latest revision

	Reference	Description	Note
Old	[17]	National Institute of Standards and Technology. Special Publication 800-57 Part 1 Rev. 4, Recommendation for Key Management. January 2016. https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-4/final	
New	[17]	National Institute of Standards and Technology. Special Publication 800-57 Part 1 Rev. 5, Recommendation for Key Management. May 2020. https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final	

RFC 6818 defines certificate validation rules in A00.FR.308, A00.FR.403, and A00.FR.413.

Reference	Description
[26]	RFC 6818. Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. https://tools.ietf.org/html/rfc6818
[27]	European Cybersecurity Certification Group Sub-group on Cryptography, Agreed Cryptographic Mechanisms, Version 2.0, April 2025 https://certification.enisa.europa.eu/document/download/a845662b-aea0-484e-9191-890c4cfa7aaa_en?filename=ECCG%20Agreed%20Cryptographic%20Mechanisms%20version%202.pdf

2.2. Page 24 - (2026-02) - Inclusion of RFC 8441 for HTTP/2

No.	Type	Description
1	Name	TLS with Basic Authentication
...		
7	Remark(s)	TLS allows a number of configurations, not all of which provide sufficient security. The requirements below describe the configurations allowed for OCPP. The Charging Station should include the same header as used in Basic Auth RFC 2617, while requesting to upgrade the http connection to a websocket connection as described in RFC 6455 and RFC 8441 (HTTP/2). The server first needs to validate the Authorization header before upgrading the connection.

2.3. Page 24 - (2026-02) - A00.FR.308 - Additional requirement references

	ID	Precondition	Requirement definition	Note
Old	A00.FR.308		The Charging Station SHALL verify the certification path of the CSMS's certificate according to the path validation rules established in Section 6 of [3].	
New	A00.FR.308		The Charging Station SHALL verify the certification path of the CSMS's certificate according to the path validation rules established in Section 6 of [3] and Section 4 of [26]	

2.4. Page 26 - (2026-02) - A00.FR.403 - Additional requirement references

	ID	Precondition	Requirement definition	Note
Old	A00.FR.403		The CSMS SHALL verify the certification path of the Charging Station's certificate according to the path validation rules established in Section 6 of [3]	
New	A00.FR.403		The CSMS SHALL verify the certification path of the Charging Station's certificate according to the path validation rules established in Section 6 of [3] and Section 4 of [26]	

2.5. Page 82 - (2026-04) - B12 - Reset with ongoing transaction allows wait for cable disconnection

New requirements have been added to clarify the expected behavior when a reset is requested during an ongoing transaction, including the requirement to allow waiting for cable disconnection before performing the reset. Requirements were further updated for clarity.

New requirement

ID	Precondition	Requirement definition	Note
B12.FR.11	B12.FR.03 AND The transaction(s) on the EVSE(s) are terminated	The Charging Station MAY wait for cable disconnection before initiating a reboot	
B12.FR.12	B12.FR.07 AND The transaction on the EVSE is terminated	The Charging Station MAY wait for cable disconnection before initiating a reset of the EVSE	
B12.FR.13	B12.FR.11	The Charging Station SHALL reboot	
B12.FR.14	B12.FR.12	The Charging Station SHALL reset the EVSE	

Changed requirement

	ID	Precondition	Requirement definition	Note
Old	B12.FR.01	When the Charging Station receives a ResetRequest(OnIdle) AND a transaction is ongoing	The Charging Station SHALL respond with a ResetResponse(Scheduled) , to indicate whether the Charging Station will attempt to reset itself or EVSE after all transactions on Charging Station or EVSE have ended.	
New	B12.FR.01	When the Charging Station receives a ResetRequest(OnIdle) AND a transaction is ongoing	The Charging Station SHALL respond with a ResetResponse(Scheduled)	
Old	B12.FR.02	When the Charging Station receives a ResetRequest(Immediate) AND a transaction is ongoing	The Charging Station SHALL respond with a ResetResponse(Accepted) , to indicate whether the Charging Station will attempt to reset itself or EVSE.	
New	B12.FR.02	When the Charging Station receives a ResetRequest(Immediate) AND a transaction is ongoing	The Charging Station SHALL respond with a ResetResponse(Accepted)	

	ID	Precondition	Requirement definition	Note
Old	B12.FR.03	If no <i>evseld</i> is supplied AND If any transaction is in progress and an OnIdle reset is received.	The transaction of the Charging Station SHALL be terminated normally, before the reboot, e.g. as in E06 - Stop Transaction .	
New	B12.FR.03	B12.FR.01 AND If no <i>evseld</i> is supplied	The transaction of the Charging Station SHALL be terminated normally, before the reboot, e.g. as in E06 - Stop Transaction .	
Old	B12.FR.04	If no <i>evseld</i> is supplied AND If any transaction is in progress and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate any transaction in progress and send a TransactionEventRequest (eventType = Ended) message with <i>triggerReason</i> = <i>ResetCommand</i> and <i>transactionInfo.stoppedReason</i> = <i>ImmediateReset</i> for each terminated transaction before performing a reboot.	
New	B12.FR.04	B12.FR.02 AND If no <i>evseld</i> is supplied	The Charging Station SHALL attempt to terminate any transaction in progress and send a TransactionEventRequest (eventType = Ended) message with <i>triggerReason</i> = <i>ResetCommand</i> and <i>transactionInfo.stoppedReason</i> = <i>ImmediateReset</i> for each terminated transaction before performing a reboot.	
Old	B12.FR.05	If an Immediate Reset without <i>evseld</i> is received and the TransactionEventResponse is not received within timeout.	The Charging Station SHALL queue the TransactionEventRequest , reboot and resend the TransactionEventRequest after the reboot.	
New	B12.FR.05	B12.FR.04 AND the TransactionEventResponse is not received within timeout.	The Charging Station SHALL queue the TransactionEventRequest , reboot and resend the TransactionEventRequest after the reboot.	
Old	B12.FR.07	If an <i>evseld</i> is supplied AND If a transaction is in progress on the EVSE and an OnIdle reset is received.	The transaction on the EVSE SHALL be terminated normally, before the reset, e.g. as in E06 - Stop Transaction .	
New	B12.FR.07	B12.FR.01 AND If an <i>evseld</i> is supplied	The transaction on the EVSE SHALL be terminated normally, before the reset, e.g. as in E06 - Stop Transaction .	
Old	B12.FR.08	If an <i>evseld</i> is supplied AND If a transaction is in progress on the EVSE and an Immediate Reset is received.	The Charging Station SHALL attempt to terminate the transaction in progress on the EVSE and send a TransactionEventRequest (eventType = Ended) message with <i>triggerReason</i> = <i>ResetCommand</i> and <i>transactionInfo.stoppedReason</i> = <i>ImmediateReset</i> before resetting the EVSE.	
New	B12.FR.08	B12.FR.02 AND If an <i>evseld</i> is supplied	The Charging Station SHALL attempt to terminate the transaction in progress on the EVSE and send a TransactionEventRequest (eventType = Ended) message with <i>triggerReason</i> = <i>ResetCommand</i> and <i>transactionInfo.stoppedReason</i> = <i>ImmediateReset</i> before resetting the EVSE.	

2.6. Page 27 - (2026-02) - A00.FR.411 - Additional requirement references

	ID	Precondition	Requirement definition	Note
Old	A00.FR.411		The Charging Station SHALL verify the certification path of the CSMS's certificate according to the path validation rules established in Section 6 of [3]. (Same as A00.FR.308)	
New	A00.FR.411		The Charging Station SHALL verify the certification path of the CSMS's certificate according to the path validation rules established in Section 6 of [3] and Section 4 of [26]. (Same as A00.FR.308)	

2.7. Page 37 - (2026-02) - A02.FR.02 - Requirement reference correction

	ID	Precondition	Requirement definition	Note
Old	A02.FR.02		The Charging Station SHALL generate a new public / private key pair using one of the key generation functions described in Section 4.2.1.3 of [16].	
New	A02.FR.02		The Charging Station SHALL generate a new public / private key pair using one of the key generation functions described in Section 8.1 of [27].	

2.8. Page 41 - (2026-02) - A03.FR.02 - Requirement reference correction

	ID	Precondition	Requirement definition	Note
Old	A03.FR.02		The Charging Station SHALL generate a new public / private key pair using one of the key generation functions described in Section 4.2.1.3 of [16].	
New	A03.FR.02		The Charging Station SHALL generate a new public / private key pair using one of the key generation functions described in Section 8.1 of [27].	

2.9. Page 41 - (2026-02) - A03.FR.02 Only applies in security profile 3 [1167]

Creation of a new client certificate must only be done while operating with security profile 3, otherwise CSMS might refuse to sign a client certificate.

Changed requirement

	ID	Precondition	Requirement definition	Note
Old	A03.FR.02	When the Charging Station detects that the current Charging Station certificate will expire in one month.	The Charging Station SHALL generate a new public / private key pair using one of the key generation functions described in Section 4.2.1.3 of [16].	

	ID	Precondition	Requirement definition	Note
New	A03.FR.02	When the Charging Station is connected with security profile 3 AND detects that the current Charging Station certificate will expire in one month.	The Charging Station SHALL generate a new public / private key pair using one of the key generation functions described in Section 4.2.1.3 of [16].	

2.10. Page 140 - (2026-06) - C22.FR.01/02 - clarify "without receipt information" [1286]

Requirements C22.FR.01/02 mention that when settlement fails or is rejected, that Charging Station sends a `NotifySettlementRequest` without receipt information. C21.FR.03, however, always requires a `receiptUrl`, either from Charging Station or CSMS.

The intention of C22.FR.01/02 is that the receipt is not a real receipt because there was no settlement. It is, however, allowed to provide a `receiptUrl` that points to a page that tells that the settlement has failed or was rejected. This is clarified by the following change of wording.

Changed requirements

	ID	Precondition	Requirements	Note
Old	C22.FR.01	When Charging Station receives a settlement status from payment terminal that is Rejected	Charging Station SHALL send a <code>[notifysettlementrequest]</code> to CSMS with <code>status = Rejected</code> , <code>settlementAmount</code> , <code>settlementTime</code> , <code>transactionId</code> and <code>pspRef</code> without receipt information.	This means the payment service provider has rejected the settlement.
New	C22.FR.01	When Charging Station receives a settlement status from payment terminal that is Rejected	Charging Station SHALL send a <code>[notifysettlementrequest]</code> to CSMS with <code>status = Rejected</code> , <code>settlementAmount</code> , <code>settlementTime</code> , <code>transactionId</code> , <code>pspRef</code> and a <code>receiptUrl</code> for a rejected settlement.	This means the payment service provider has rejected the settlement. Information about this may be provided in receiptUrl
Old	C22.FR.02	When Charging Station receives a settlement status from payment terminal that is Failed	Charging Station SHALL send a <code>[notifysettlementrequest]</code> to CSMS with <code>status = Failed</code> , <code>settlementAmount</code> , <code>settlementTime</code> , <code>transactionId</code> and <code>pspRef</code> without receipt information.	This means that there are technical reasons why the settlement failed, e.g. a failure to communicate.
New	C22.FR.02	When Charging Station receives a settlement status from payment terminal that is Failed	Charging Station SHALL send a <code>[notifysettlementrequest]</code> to CSMS with <code>status = Failed</code> , <code>settlementAmount</code> , <code>settlementTime</code> , <code>transactionId</code> , <code>pspRef</code> and a <code>receiptUrl</code> for a failed settlement.	This means that there are technical reasons why the settlement failed, e.g. a failure to communicate. Information about this may be provided in receiptUrl

2.11. Page 283 - (2026-02) - I06.FR.03 - added note about privacy [715]

A note has been added about protecting privacy when showing messages.

	ID.	Precondition	Requirements
Old	I06.FR.03	I06.FR.02	The Charging Station SHALL display the updated tariff information to the EV Driver.
New	I06.FR.03	I06.FR.02	The Charging Station SHALL display the updated tariff information to the EV Driver. Note: Charging Station should take measures to protect the privacy of the EV Driver, for example by ensuring that only the authenticated user can view the message.

2.12. Page 227 - (2026-06) - New F01.FR.27 and F02.FR.28 for *evseId* in RequestStartTransactionRequest [1281]

There has always been an implicit (silent) requirement that a CSMS is able to provide an EVSE ID in a RequestStartTransactionRequest. It is only for flexibility that the option was given to let the charging station decide which EVSE to use if it is able to do so. A charging station that cannot do this can reject the request. In that case CSMS will have to send a new RequestStartTransaction with an EVSE ID.

Unfortunately, due to the *evseId* being an optional field, a CSMS could be implemented that is never sending an *evseId* and still comply to the requirements. Therefore, a new requirement is added to close this gap.

New requirement

ID	Precondition	Requirement definition	Note
F01.FR.27		CSMS SHALL be able to provide an <i>evseId</i> in a RequestStartTransactionRequest for interoperability with Charging Stations that do not accept RequestStartTransactionRequest without it.	F01.FR.07 allows a charging station to reject a request without <i>evseId</i> . F01.FR.06 allows CSMS to send a request without <i>evseId</i> , but in case of F01.FR.07 it must be able to provide a <i>evseId</i> .

The same requirement is also for F02.

New requirement

ID	Precondition	Requirement definition	Note
F02.FR.28		CSMS SHALL be able to provide an <i>evseId</i> in a RequestStartTransactionRequest for interoperability with Charging Stations that do not accept RequestStartTransactionRequest without it.	(Same as F01.FR.27) F02.FR.15 allows a charging station to reject a request without <i>evseId</i> . F02.FR.14 allows CSMS to send a request without <i>evseId</i> , but in case of F02.FR.15 it must be able to provide a <i>evseId</i> .

2.13. Page 415 - (2026-04) - L01.FR.04 - removed ECSchnorr algorithm option and clarify the signature verification the Charging Station must do

	ID.	Precondition	Requirements	Note
Old	L01.FR.04	When the Charging Station has successfully downloaded the new firmware	The signature SHALL be validated, by calculating the signature over the entire firmware file using the RSA-PSS or ECSchnorr algorithm for signing, and the SHA256 algorithm for calculating hash values.	
New	L01.FR.04	When the Charging Station has successfully downloaded the new firmware	The charging station SHALL verify the signature from the UpdateFirmwareRequest with the hash over the entire received firmware file computed using the signature hash algorithm.	The RSA-PSS or ECDSA algorithm is used for signing. Please refer to section [certificate_properties] for more details about signature algorithms and computing hash values.

2.14. Page 392 - (2026-06) - K18.FR.17/K19.FR.13 InvalidMessageSequence must be InvalidMessageSeq

Requirements K18.FR.17 and K19.FR.13 refer to a *reasonCode* InvalidMessageSequence. This must be InvalidMessageSeq.

ID	Precondition	Requirements	Note
K18.FR.17	When Charging Station receives a SetChargingProfileRequest immediately after the transaction has started and before it has sent the NotifyEVChargingNeedsRequest to CSMS	The Charging Station SHOULD respond with SetChargingProfileResponse with status = Rejected and a <i>statusInfo</i> with <i>reasonCode</i> = "InvalidMessageSeq".	CSMS sent profile too early. It does not harm if CS accepts the charging profile instead of rejecting it, as long as it sends a charging profile again when it receives the NotifyEVChargingNeedsRequest .
K19.FR.13	When Charging Station receives a SetChargingProfileRequest immediately after the transaction has started and before it has sent the NotifyEVChargingNeedsRequest to CSMS	The Charging Station SHOULD respond with SetChargingProfileResponse with status = Rejected and a <i>statusInfo</i> with <i>reasonCode</i> = "InvalidMessageSeq".	CSMS sent profile too early. It does not harm if CS accepts the charging profile instead of rejecting it, as long as it sends a charging profile again when it receives the NotifyEVChargingNeedsRequest .

2.15. Page 502 - (2026-02) - Improved description of LocalLoadBalancing

The section about OperationMode "LocalLoadBalancing" is replaced in its entirety by the following text.

OperationMode LocalLoadBalancing

OperationMode LocalLoadBalancing can be used to limit grid import and export of a home by charging or discharging an EV.

Only one measurement is assumed: the active power at the grid connection point (PCC), where import from the grid is positive and export to the grid is negative. This measured value already includes the EV power, as shown in below diagram.

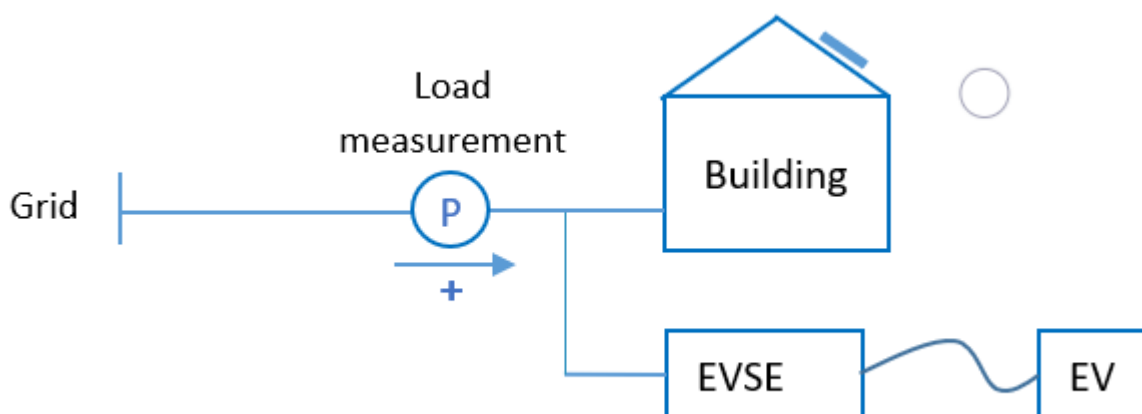


Figure 1. Diagram of a local load-balancing setup

Objective

The goal is to keep the measured grid power within an allowed range by using the EV as a buffer:

- Too much grid import ⇒ EV discharges
- Too much grid export ⇒ EV charges
- Inside the allowed range ⇒ EV does nothing

Because the EV power is already included in the grid measurement, the controller must work incrementally to avoid oscillations.

Thresholds and offsets

Two thresholds define the minimum EV action:

- **LowerThreshold** (negative): minimum discharging rate of the EV
- **UpperThreshold** (positive): minimum charging rate of the EV

Between these two values, the EV is not required to compensate the home load.

Offsets are used to shift or widen this neutral zone:

- **LowerOffset** is added to LowerThreshold
- **UpperOffset** is added to UpperThreshold

This results in the effective limits:

- Lower limit = LowerThreshold + LowerOffset
- Upper limit = UpperThreshold + UpperOffset

These limits define the allowed grid power band.

Controller behavior (conceptual)

At each control step:

1. Measure the grid power at the PCC (including EV).
2. Compare the measured value with the lower and upper limits.
3. Apply the following logic:
 - If grid power is below the lower limit (too much export): the EV increases charging power to raise grid power.
 - If grid power is above the upper limit (too much import): the EV increases discharging power to lower grid power.
 - If grid power is inside the limits: the EV setpoint is left unchanged.

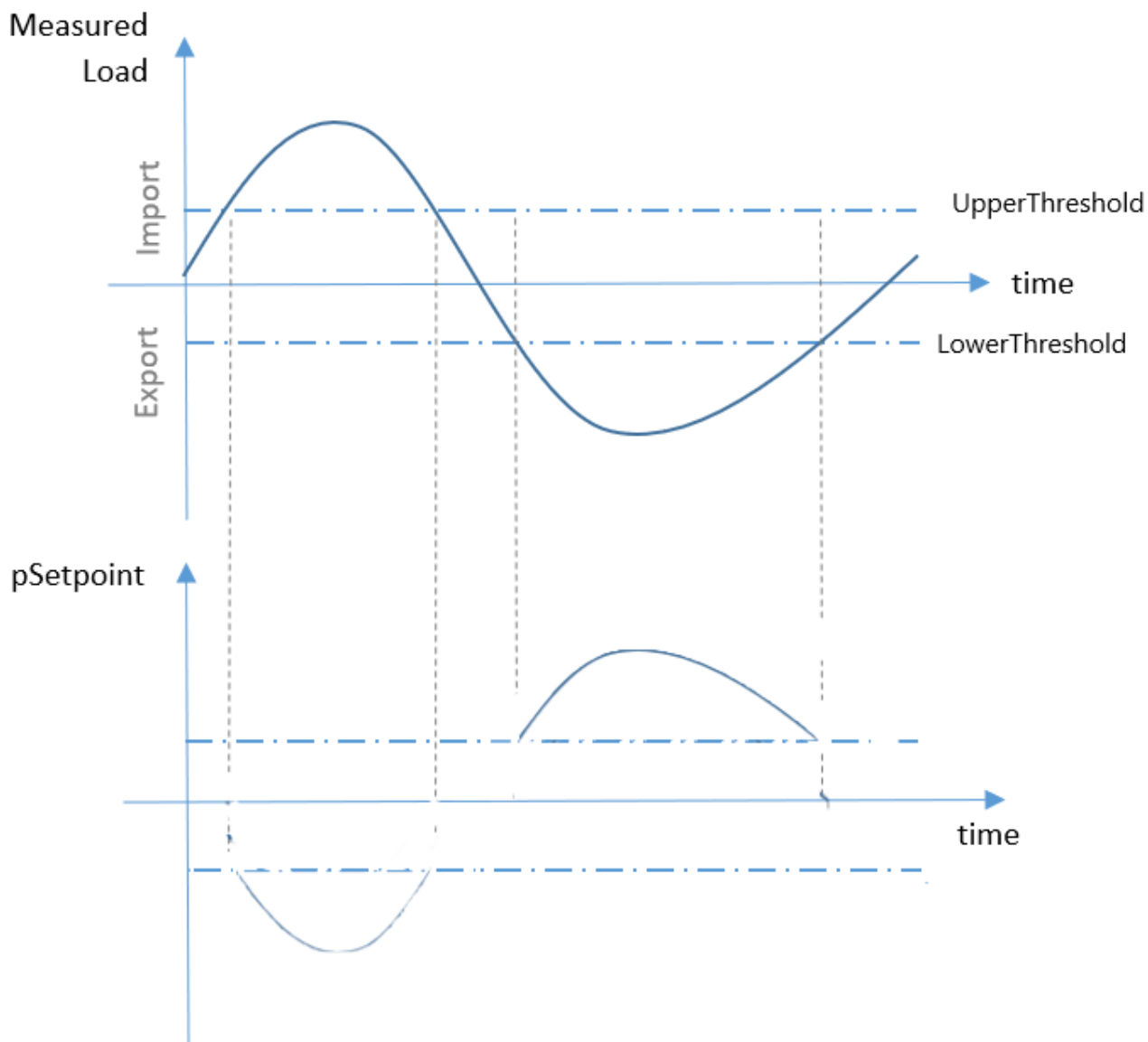


Figure 2. Graphs of measured load and compensating setpoint

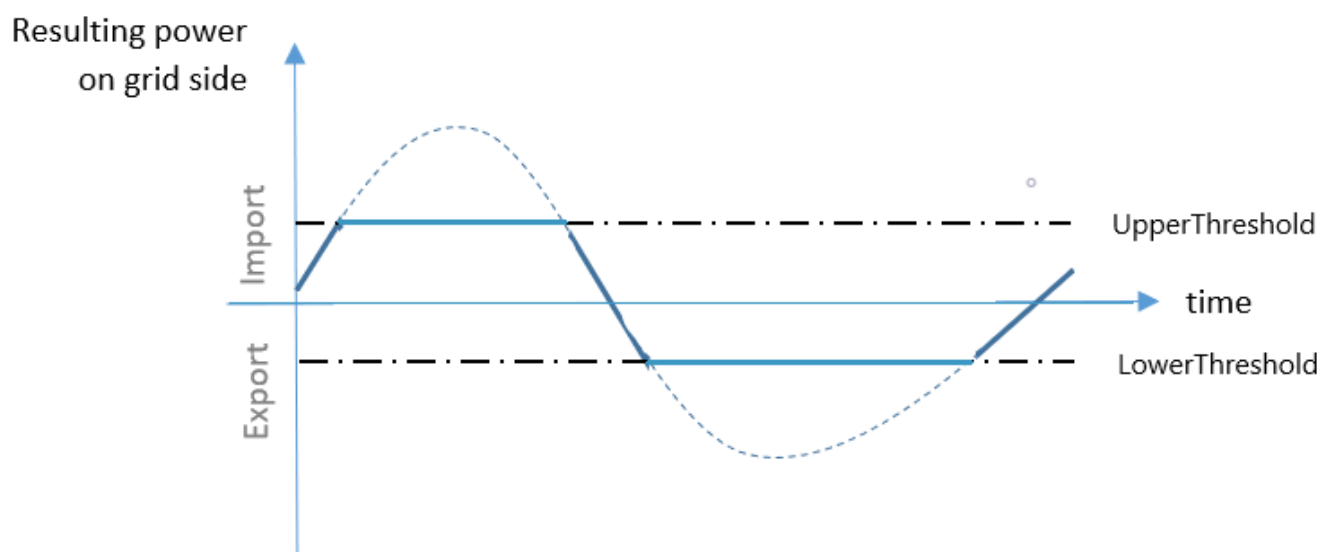


Figure 3. Graph of resulting power import/export

Formula for a deadband controller

```
Load = get latest load power reading ( )
```

```

If Load - Setpoint > UpperThreshold + UpperOffset - LowerThreshold
    Setpoint_new = Setpoint + UpperThreshold + UpperOffset - Load
Else If Load - Setpoint < LowerThreshold + LowerOffset - UpperThreshold
    Setpoint_new = Setpoint + LowerThreshold + LowerOffset - Load
Else
    Setpoint_new = 0

```

The EV setpoint is updated by adding the required correction to the previous setpoint, not by calculating an absolute value.

If the controller were to compute an absolute EV power directly from the grid measurement, it would continuously over-correct and oscillate. By adjusting the EV setpoint relative to its previous value, the controller converges in one step and remains stable.

Zero-import / zero-export operation

Although the thresholds define the **minimum** charging and discharging rates, the controller is not limited to those values:

- If grid power exceeds the limits by a large amount, the EV can compensate with more power than the threshold values.
- The further the grid power is outside the band, the stronger the correction.

This is what makes the EV an effective buffer for larger household loads.

Full compensation to 0 W grid power is possible by choosing the offsets as:

- LowerOffset = -LowerThreshold
- UpperOffset = -UpperThreshold

This moves both limits to 0 W.

Resulting behavior:

- Any grid import causes the EV to discharge.
- Any grid export causes the EV to charge.
- The controller continuously drives the net grid power toward zero, limited only by the EV's charging and discharging capabilities.

Relation with IEC 61850 DWFL

The LocalLoadBalancing control logic relates directly to the IEC 61850-7-420 logical node **DWFL** (Active Power Following).

From a functional perspective, the EV is following the grid power in order to counteract deviations beyond configured limits. This is exactly the problem DWFL was designed to solve. In IEC 61850-7-420, the logical node DWFL describes an active power following function.

Conceptually, DWFL:

- Observes a reference active power (what is actually happening)
- Has a target active power (what is desired)
- When the reference deviates beyond a threshold, it produces a compensation request to drive the system back toward the target

DWFL does not directly command a device; instead, it expresses how much power should change to follow the target.

2.16. Page 505 - (2026-06) - Update table in Q2.2 Limits and setpoints per operation mode [1180]

The following table has been updated to allow (discharge)limits for LocalLoadBalancing.

Limits and setpoints per operation mode

<i>limits and setpoints per operation mode</i>	limit	dischargeLimit	setpoint	setpoint-Reactive
ChargingOnly	required	-	-	-
CentralSetpoint	optional	optional	required	optional
CentralFrequency	optional	optional	required	-
LocalFrequency	-	-	calculated from freq/signal curves	-
ExternalSetpoint	optional	optional	set externally	-
ExternalLimits	set externally	set externally	-	-
LocalLoadBalancing	optional	optional	calculated from load and thresholds/offsets	-

2.17. Page 531 - (2026-02) - Use case Q09 has improved description of LocalLoadBalancing

No.	Type	Description
1	Name	Local V2X control for load balancing
2	ID	Q09
3	Objective	...
4	Description	...
	Actors	EV, Charging Station, CSMS
	Scenario description	1. The CSMS has sent a setchargingprofilerequest message, with a <i>chargingSchedule</i> that has one or more periods with <i>operationMode</i> set to <i>LocalLoadBalancing</i>. 2. The Charging Station will continually get the latest load power reading to calculate a new <i>setpoint</i> that will offset the measured power generation or consumption, such that it stays between the configured upper and lower limit. Configuration variables are instances of <i>V2XChargingCtrlr.LocalLoadBalancing</i>: - The upper limit is defined as: $UpperThreshold + UpperOffset$. - The lower limit is defined as: $LowerThreshold + LowerOffset$.
5	Prerequisites	- Charging Station is able to read the upstream meter. - The configuration variable <i>UpperThreshold</i> (a positive value) should be equal or above the minimum charging rate of the EV. - The configuration variable <i>LowerThreshold</i> (a negative value) should be equal or below the minimum (as in "closest to zero") discharging rate of the EV.
6	Post conditions	ChargingSchedulePeriod ends
7	Error Handling	
8	Remarks	By setting the configuration variable <i>UpperOffset</i> to $-UpperThreshold$ and <i>LowerOffset</i> to $-LowerThreshold$ the nett power will be regulated down to 0 W as soon as the building power exceeds the thresholds. See section intro-localloadbalancing in the introduction of this chapter for more detailed information on how local load-balancing works. This use case only works for one Charging Station, but can work for multiple EVSEs when Charging Station has the capability to combine setpoints from multiple EVSEs to compensate the measured load at the upstream meter. Local load-balancing by multiple Charging Stations would have to be coordinated by a CSMS or Local Controller using the Central Setpoint use case.

2.18. Page 533 - (2026-02) - Use case Q09 has improved sequence diagram

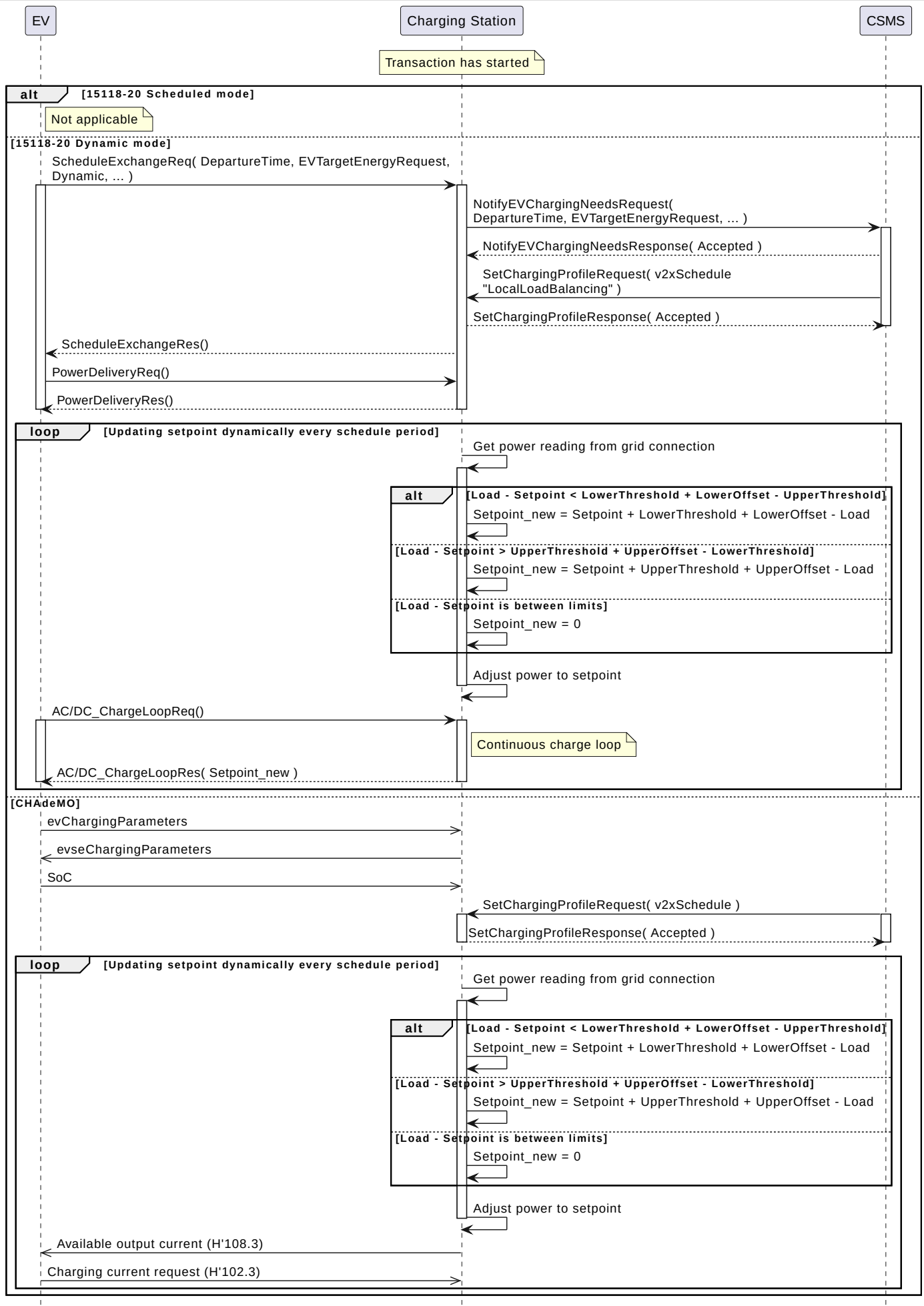


Figure 4. V2X Operation Mode LocalLoadBalancing

2.19. Page 534 - (2026-02) - Use case Q09 has improved requirements

Changed requirements

	ID.	Precondition	Requirements	Note
Old	Q09.FR.02	When Charging Station receives a [setchargingprofilerequest] with a [cmn_chargingscheduleperiodtype] with <i>operationMode</i> = LocalLoadBalancing AND ((UpperThreshold or LowerThreshold are not set) OR (UpperOffset or LowerOffset are not set) OR (UpperThreshold is not greater than LowerThreshold))	Charging Station SHALL respond with [setchargingprofileresponse] with <i>status</i> = Rejected and <i>statusInfo</i> = "MissingDevModelInfo"	
New	Q09.FR.02	When Charging Station receives a [setchargingprofilerequest] with a [cmn_chargingscheduleperiodtype] with <i>operationMode</i> = LocalLoadBalancing AND AutoThresholds = <i>false</i> or not set AND ((UpperThreshold or LowerThreshold are not set) OR (UpperOffset or LowerOffset are not set) OR (UpperThreshold is not greater than LowerThreshold))	Charging Station SHALL respond with [setchargingprofileresponse] with <i>status</i> = Rejected and <i>statusInfo</i> = "MissingDevModelInfo"	
Old	Q09.FR.04	Q09.FR.03	Charging Station SHALL calculate the power setpoint as follows: <pre> If Load > UpperThreshold DeltaSetpoint = UpperThreshold + UpperOffset - Load Else If Load < LowerThreshold DeltaSetpoint = LowerThreshold + LowerOffset - Load Else DeltaSetpoint = 0 setpoint = setpoint + DeltaSetpoint </pre>	"Load" is power from upstream meter. Positive values for consumption of energy.
New	Q09.FR.04	Q09.FR.03	Charging Station SHALL calculate the power setpoint for EV such that the power reported by the upstream grid meter in steady state is reduced to a value between UpperThreshold + UpperOffset and LowerThreshold + LowerOffset .	See section [intro-localloadbalancing] for an example algorithm that can be used for this.

New requirement

ID.	Precondition	Requirements	Note
Q09.F R.06	If <code>AutoThresholds = true</code>	Charging Station SHALL set * <code>UpperThreshold = minChargePower,</code> * <code>LowerThreshold = -minDischargePower,</code> * <code>UpperOffset = -UpperThreshold,</code> * <code>LowerOffset = -LowerThreshold</code> where <code>minChargePower</code> and <code>minDischargePower</code> are values from <code>[notifyevchargingneedsrequest]</code>	This will automatically regulate grid power down to 0 W while taking into account the minimum charging/discharging rates of the EV.

2.20. Page 554 - (2026-06) - Typo in `derChargingParameters` field in R02 and R03. [1231]

The requirements R02.FR.01 and R03.FR.01 refer to the parameter `derChargeParameters`. This must be changed to `derChargingParameters`.

Changed requirement

ID.	Precondition	Requirements	Note
R02.FR.01	Charging Station starts AC bidirectional power transfer session with EV via ISO 15118-20 AC_BPT_DER service	Charging Station SHALL expose the DER capabilities of the DC inverter in EV, also referred to as nameplate information, in the field <code>derChargingParameters</code> of a <code>[notifyevchargingneedsrequest]</code> .	

ID.	Precondition	Requirements	Note
R03.FR.01	Charging Station start AC bidirectional power transfer session with EV via ISO 15118-20 AC_BPT_DER service	Charging Station SHALL expose the DER capabilities of the DC inverter in EV, also referred to as nameplate information, in the field <code>derChargingParameters</code> of a <code>[notifyevchargingneedsrequest]</code> .	(Same as R02.FR.01)

2.21. Page 655 - (2026-06) - `LimitMaxDischargeType` `pctMaxDischargePower` should be required [1270]

The field `pctMaxDischargePower` is marked as optional in `LimitMaxDischargeType`, but that is not correct. It should be treated as a required field and it is not only for use in combination with a `powerMonitoringMustTrip` field.

Update field description

LimitMaxDischargeType

Field Name	Field Type	Card.	Description
priority	integer, 0 <= val	1..1	Required. Priority of setting (0=highest)
pctMaxDischargePower	decimal	0..1	Optional. Only for PowerMonitoring. Should be treated as required. The value specifies a percentage (0 to 100) of the rated maximum discharge power of EV. If present, the PowerMonitoring MonitoringMustTrip curve becomes active when power exceeds this percentage.
startTime	dateTime	0..1	Optional. Time when this setting becomes active
duration	decimal	0..1	Optional. Duration in seconds that this setting is active

Field Name	Field Type	Card.	Description
powerMonitoringMustTrip	DERCurveType	0..1	Optional. The curve is an interpolation of data points where the x-axis values are time in seconds and the y-axis values refer to the percentage value of the rated EVMaximumDischargePower, reported in the ChargeParameterDiscoveryRequest message. The value lies between 0 and 100. The curve is activated when the power value measured via the ExternalMeter value reported in the ChargeLoopRes is higher than the <i>pctMaxDischargePower</i> defined above. If the power does not stay within the defined curve for the respective time period, the EV must trip.

2.22. Page 666 - (2026-02) - Updated description of StreamDataElementType [1160]

Added the unit (seconds) to description of *t*.

StreamDataElementType

Field Name	Field Type	Card.	Description
t	decimal	1..1	Required. Offset in seconds relative to <i>basetime</i> of this message. <i>basetime</i> + <i>t</i> is timestamp of recorded value.
v	string[0..2500]	1..1	Required.

2.23. Page 687 - (2026-06) - DERControlEnumType value PowerMonitoringMustTrip is not a DER control [1270]

The DER control *PowerMonitoringMustTrip* is not a valid DER control. This curve only exists are part of the DER Control *LimitMaxDischarge*.

It cannot be removed from the existing enumeration, but it has been marked as a value that cannot be used.

Value	Description
...	...
PowerMonitoringMustTrip	Not valid as a DER control! The <i>PowerMonitoringMustTrip</i> DER control only exists as part of a <i>LimitMaxDischarge</i> DER control. It cannot be defined stand-alone.
...	...

2.24. Page 782 - (2026-06) - New optional ISO15118ProtocolEnabled variable [1252]

The variable *ISO15118Ctrlr.ProtocolSupported[]* lists the ISO 15118 versions that the SECC supports.

A new (optional) configuration variable *ISO15118ProtocolEnabled* has been added to allow the selective enabling/disabling of these protocol versions. The variable *ProtocolEnabled[]* in the *ISO15118Ctrlr* specifies for each matching instance in *ProtocolSupported[]* whether it is enabled or not.

New configuration variable

2.24.1. ISO15118ProtocolEnabled

New in OCPP 2.1

Required	no
-----------------	----

Component	componentName	ISO15118Ctrlr	
	evse	*	
Variable	variableName	ProtocolEnabled	
	variableInstance	<instance number in ProtocolSupported variable>	
	variableAttributes	mutability	ReadWrite
	variableCharacteristics	dataType	boolean
Description	By default, a protocol listed in ISO15118Ctrlr.ProtocolSupported is enabled, unless an instance of ISO15118Ctrlr.ProtocolEnabled with the same instance name exists that has a value of false. In that case the protocol mentioned in ISO15118Ctrlr.ProtocolSupported with the same instance is explicitly disabled and cannot be used. ProtocolEnabled shall always have the same number of instances and the same instance names as ProtocolSupported, such that instance "1" of ProtocolEnabled matches with instance "1" of ProtocolSupported and instance "2" of ProtocolEnabled matches with instance "2" of ProtocolSupported, and so on. Below example shows how the ISO 15118-20:DC protocol can be disabled, even though it is supported: Example: Charging Station supports ISO 15118-2 and ISO 15118-20 DC - ISO15118Ctrlr.ProtocolSupported["1"] = "urn:iso:15118:2:2013:MsgDef,2,0" - ISO15118Ctrlr.ProtocolSupported["2"] = "urn:iso:std:iso:15118:-20:DC,1,0" CSO disables ISO 15118-20 DC: - ISO15118Ctrlr.ProtocolEnabled["1"] = true - ISO15118Ctrlr.ProtocolEnabled["2"] = false		

2.25. Page 786 - (2026-06) - Updated values ConnectedEV for CHAdeMO [1192]

The highlighted rows have been added to the ConnectedEV for CHAdeMO in 2.16.2 in chapter Referenced Components and Variables.

Variable	Unit	CHAdeMO value
Voltage and current values:		
ACCurrent(MinSet)	A	-
ACCurrent(MaxSet)	A	-
ACVoltage(MaxSet)	V	-
DCCurrent(MinSet)	A	Minimum charge current (H'100.0)
DCCurrent(MaxSet)	A	Maximum charge current (H'102.3)
DCCurrent(Target)	A	Charging current request (H'108.3) If HighCurrentControl is true, use the value from Charging current request (extended) (H'110.1,2).
DCVoltage(MinSet)	V	Minimum battery voltage (H'100.2,3)
DCVoltage(MaxSet)	V	Maximum battery voltage (H'100.4,5)
DCVoltage(Target)	V	Target battery voltage (H'102.1,2)
DischargeDCCurrent(MaxSet)	A	Maximum discharge current (H'200.0)
DischargeDCCurrent(Target)	A	Discharging current request (H'208.0)
DischargeDCVoltage(MinSet)	V	Minimum battery discharging voltage (H'200.4,5)
Power, energy and time values:		

Variable	Unit	CHAdEMO value
Power(MaxSet)	W	-
EnergyImport(MaxSet)	Wh	Total capacity of traction battery * 100 (H'101.5,6)
EnergyImport(Target)	Wh	-
DepartureTime	dateTime	-
RemainingTimeBulk	s	-
RemainingTimeFull(MaxSet)	s	Maximum charging time * 60 (H'101.2)
RemainingTimeFull	s	Estimated charging time * 60 (H'101.3)
StateOfChargeBulk	%	-
StateOfCharge(MaxSet)	%	Charged rate reference constant (H'100.6)
StateOfCharge	%	State of charge (H'102.6)
ChargingCompleteBulk	boolean	-
ChargingCompleteFull	boolean	-
Error status values:		
ChargingState with a memberlist consisting of the following values:		
* BatteryOvervoltage		Battery overvoltage (H'102.4.0)
* BatteryUndervoltage		Battery undervoltage (H'102.4.1)
* ChargingCurrentDeviation		Battery current deviation (H'102.4.2)
* BatteryTemperature		High battery temperature (H'102.4.3)
* VoltageDeviation		Battery voltage deviation (H'102.4.4)
* ChargingSystemError		Charging system error (H'102.5.2)
* VehicleShiftPosition		Vehicle shift position (H'102.5.1)
* VehicleChargingEnabled		Vehicle charging enabled (H'102.5.0)
* ChargingSystemIncompatibility		-
* ChargerConnectorLockFault		-
NOTE: ChargingState variable reports an error status and is not related to the <i>chargingState</i> in a TransactionEventRequest.		

2.26. Page 792 - (2026-02) - New configuration variable AutoThresholds

V2XLocalLoadBalancing[AutoThresholds]

New in OCPP 2.1

Required	no		
Component	componentName	V2XChargingCtrlr	
	evse	*	
Variable	variableName	LocalLoadBalancing	
	variableInstance	AutoThresholds	
	variableAttributes	mutability	ReadWrite
	variableCharacteristics	dataType	boolean
Description	When set to <i>true</i> , this will automatically set the value of UpperThreshold to the minimum charging power of the EV (minChargePower) and the value of LowerThreshold to the minus the minimum discharging rate (minDischargePower) from the NotifyEVChargingNeedsRequest message. This ensures that the load balancing algorithm will not provide setpoints that cannot be achieved by EV.		

Chapter 3. Part 3

Currently no new errata for OCPP 2.1 part 3.

Chapter 4. Part 4

4.1. Page 10 - (2026-02) - Update handling of unknown message type [1163]

The sentence about ignoring unknown message types has been improved slightly. It mentioned to ignore the message payload, but the intention was to ignore the entire message.

Old paragraph	When a system receives a message with a Message Type Number not in this list, it SHALL ignore the message payload. Each message type may have additional required fields.
New paragraph	When a system receives a message with a Message Type Number not in this list, it SHALL ignore the message payload . Each message type may have additional required fields.

4.2. Page 14 - (2026-02) - Wrong example of NotifyPeriodicEventStream [1159]

Chapter 4.2.4 on page 14 shows an example RPC of the "NotifyPeriodicEventStream" action - they payload of which does not match the JSON schema.

The correct example is:

```
[6,
  "19223201",
  "NotifyPeriodicEventStream",
  {
    "id": 123,
    "pending": 0,
    "basetime" = "2024-08-27T12:30:40Z",
    "data": [ {"t": 0, "v": "230.4"}, {"t": 5, "v": "230.2"} ]
  }
]
```

4.3. Page 16 - (2026-02) - MessageTypeNotSupported is deprecated [1161]

As of OCPP 2.1 a message type number that is not supported is silently ignored, as described in section 4.4. This is therefore no longer a valid error code.

Valid Error Codes

ErrorCode	Description
...	...
MessageTypeNotSupported	(deprecated) A message with a Message Type Number that is not supported by this implementation.
...	...

4.4. Page 16 - (2026-02) - Section 4.4 Extension fallback mechanism removed [1163]

Section 4.4 "Extension fallback mechanism" mentions the same as section 4.1, which states to ignore unknown message types. It has therefore been removed.

Chapter 5. Part 5

5.1. Page 20 - (2026-02) - TC_C_100_CS - Test case removed

This testcases has been removed, because it duplicates TC_E_05_CS.

5.2. Page 28 - (2026-06) - TC_E_39_CS - Only applicable for TxStopPoint Authorized

We already have two other testcases testing the EVConnectionTimeout; TC_E_05_CS for Local and TC_F_04_CS for Remote. TC_E_39_CS was specifically meant to test it in combination with TxStopPoint Authorized.

		Stop transaction options					
Old	TC_E_39 (2.1)	Deauthorized - timeout	C	M		NOT HFS-13	
New	TC_E_39 (2.1)	Deauthorized - timeout	C	M		C-10.2 AND NOT HFS-13	

5.3. Page 60 - (2026-02) - TC_Q_102_CSMS - Test case removed

This testcases has been removed, because as a CSMS cannot be forced to omit the energyModes. In addition, there is also no added benefit to test this.

5.4. Page 61 - (2026-02) - TC_R_101_CS - Test case removed

This testcases has been removed, because TC_B_53_CS will be responsible for performing all device model validations.

Chapter 6. Part 6

6.1. General

Currently no errata for Part 6 General.

6.2. Charging Station

6.2.1. Page 23 - (2026-02) - TC_A_20_CS - Added note that the testcase will end when the SetNetworkProfile is rejected

Test case name	Upgrade Charging Station Security Profile - No valid CSMSRootCertificate installed
Test case Id	TC_A_20_CS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetNetworkProfileResponse	1. The Test System sends a SetNetworkProfileRequest with - configurationSlot is <Configured configurationSlot2> or <Configured configurationSlot> (the one currently not used for the active connection) - connectionData.messageTimeout <Configured messageTimeout2> - connectionData.ocppCsmsUrl <ocppCsmsUrl that is not currently active> - connectionData.ocppInterface <Configured ocppInterface2> - connectionData.ocppVersion OCPP20 - connectionData.securityProfile <Configured securityProfile2>
Note: <i>If the Charging Station responds with a SetNetworkProfileResponse with status Rejected, then step 3/4 will not be executed.</i>	
4. The Charging Station responds with a SetVariablesResponse	3. The Test System sends a SetVariablesRequest with variable.name is "NetworkConfigurationPriority" component.name is "OCPPCommCtrlr" attributeValue is <configurationSlot set at step 1>,<previous configurationSlot>

6.2.2. Page 57 - (2026-04) - TC_B_21_CS - Clarification of charging station reboot behaviour with ongoing transactions

The test case has been updated to provide clarity on when the charging station is allowed to reboot. The charging station is expected to reboot either after the transaction has ended or if the transaction ends before the cable is disconnected, once the cable is disconnected.

Test case name	Reset Charging Station - With Ongoing Transaction - OnIdle
Test case Id	TC_B_21_CS

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): State is <i>EnergyTransferStarted</i>

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a ResetResponse	1. The Test System sends a ResetRequest with type <i>OnIdle</i>
3. Execute Reusable State <i>StopAuthorized</i>	

Main (Test scenario)
Notes(s): If the TxStopPoint contains EnergyTransfer, PowerPathClosed, or Authorized: The test system will wait the <Configured Long Operation Time Out> to determine if the charging station resets because the transaction has ended. If the charging station resets proceed to step 7. Else proceed with step 4. 4. Execute Reusable State <i>EVConnectedPostSession</i> 5. Execute Reusable State <i>EVDisconnected</i> Notes(s): Steps 4 and 5 will only be executed if TxStartPoint does not contain: EnergyTransferStarted, DataSigned, PowerPathClosed, or Authorized Notes(s): Step 6 will only be executed if TxStartPoint does not contain: EnergyTransferStarted, DataSigned, PowerPathClosed, Authorized, or EVConnected is ParkingBayOccupancy ...

Tool validations
* Step 2: Message ResetResponse - status <i>Scheduled</i> * Step 7: Message BootNotificationRequest - reason <i>ScheduledReset</i> * Step 9: Message: StatusNotificationRequest - If the transaction was stopped charging station reset at step 3, then – for the connector involved in the transaction: connectorStatus <i>Occupied</i> – for other connectors of this EVSE: connectorStatus <i>Available</i> or <i>Unavailable</i> Else connectorStatus <i>Available</i> Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> - If the transaction was stopped charging station reset at step 3, then – for the connector involved in the transaction: eventData[0].actualValue <i>Occupied</i> – for other connectors of this EVSE: eventData[0].actualValue <i>Available</i> or <i>Unavailable</i> - Else eventData[0].actualValue <i>"Available"</i> * Step 11: Message: SecurityEventNotificationRequest - type <i>StartupOfTheDevice</i> or <i>ResetOrReboot</i> Post scenario validations: - A message to report the state of a connector has been received for all connectors.

6.2.3. Page 65 - (2026-04) - TC_B_41_CS - Clarification of charging station reboot behaviour with ongoing transactions

The test case has been updated to provide clarity on when the charging station is allowed to reboot. The charging station is expected to reboot either after the transaction has ended or if the transaction ends before the cable is disconnected, once the cable is disconnected.

Test case name	Reset Charging Station - With multiple ongoing transactions - OnIdle
Test case Id	TC_B_41_CS

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. The Charging Station responds with a ResetResponse	1. The Test System sends a ResetRequest with type OnIdle
3. Execute Reusable State StopAuthorized for EVSE.id = 1	
4. Execute Reusable State EVConnectedPostSession for EVSE.id = 1	
5. Execute Reusable State EVDisconnected for EVSE.id = 1	
6. Execute Reusable State ParkingBayUnoccupied for EVSE.id = 1	
Notes(s): The test system will wait the configured max timeout period to ensure the charging station does not reboot at this point.	
7. Execute Reusable State StopAuthorized for EVSE.id = 2	
Notes(s): If the TxStopPoint contains EnergyTransfer, PowerPathClosed, or Authorized: The test system will wait the <Configured Long Operation Time Out> to determine if the charging station resets because the transaction has ended.	
If the charging station resets proceed to step 11. Else proceed with step 8.	
8. Execute Reusable State EVConnectedPostSession for EVSE.id = 2	
Note(s): If TxStopPoint contains one of the following values; Authorized, EnergyTransfer, PowerPathClosed, DataSigned. Then the transaction will have ended at the EVConnectedPostSession state AND the Charging Station will proceed with resetting itself. Proceed to step 11. Else proceed with step 9.	
9. Execute Reusable State EVDisconnected for EVSE.id = 2	
Note(s): If TxStopPoint contains the value EVConnected. Then the transaction will have ended at the EVDisconnected state AND the Charging Station will proceed with resetting itself. Proceed to step 11. Else proceed with step 10.	
10. Execute Reusable State ParkingBayUnoccupied for EVSE.id = 2	
Note(s): The transaction will end at this state, if it was not ended at an earlier state. Proceed to step 11.	
Notes(s): Step 10 will only be executed if TxStopPoint is ParkingBayOccupancy	
11. The Charging Station sends a BootNotificationRequest	12. The Test System responds with a BootNotificationResponse
13. The Charging Station notifies the CSMS about the current state of all connectors.	14. The Test System responds accordingly.

Tool validations
* Step 2: Message ResetResponse - status <i>Scheduled</i> * Step 11: Message BootNotificationRequest - reason <i>ScheduledReset</i> * Step 13: Message: StatusNotificationRequest - If the transaction was stopped charging station reset at step 3 7, then — for the connector involved in the transaction: connectorStatus <i>Occupied</i> — for other connectors of this EVSE: connectorStatus <i>Available</i> or <i>Unavailable</i> - Else connectorStatus <i>Available</i> Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> - If the transaction was stopped charging station reset at step 3 7, then — for the connector involved in the transaction: eventData[0].actualValue <i>Occupied</i> — for other connectors of this EVSE: eventData[0].actualValue <i>Available</i> or <i>Unavailable</i> - Else eventData[0].actualValue <i>"Available"</i>
Post scenario validations: - A message to report the state of a connector has been received for all connectors.

6.2.4. Page 91 - (2026-02) - TC_B_100_CS - Use active securityProfile level

Test case name	Set new NetworkConnectionProfile - Identity and password
Test case Id	TC_B_100_CS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetNetworkProfileResponse	1. The Test System sends a SetNetworkProfileRequest with: - configurationSlot is <i><Configured non-active configurationSlot></i> - connectionData.ocppVersion <i>OCPP21</i> - connectionData.ocppInterface <i>Any</i> - connectionData.ocppTransport <i>JSON</i> - connectionData.messageTimeout <i><Configured messageTimeout></i> - connectionData.ocppCsmsUrl <i><Configured ocppCsmsUrl></i> - connectionData.securityProfile <Active SecurityProfile> - connectionData.identity <i><Configured identity></i> - connectionData.basicAuthPassword <i><Configured basicAuthPassword></i>
...	

6.2.5. Page 1006 - 1112 - (2026-02) - TC_B_105_CS/TC_B_107_CS/TC_B_111_CS/TC_B_113_CS - Identity included in SetVariablesRequest on component *NetworkConfiguration

The **SetVariablesRequest** on component **NetworkConfiguration***<Configured non-active configurationSlot>* was updated to include the required field **Identity**.

The note relating to the DeviceDataCtrlr.ItemsPerMessage was updated to be 10 or more items with the additional item for **Identity**

being sent.

Main (Test scenario)	
Charging Station	CSMS
...	
Note: If the value of DeviceDataCtrlr.ItemsPerMessage[GetVariables] from step #1 is 9 or more 10 or more , then the following can all be set in one SetVariablesRequest, otherwise the variables are set one-by-one.	
...	

6.2.6. Page 125 - (2026-02) - TC_C_14_CS - Clarifications and Additional tool validations

Test case name	Authorization through authorization cache - GroupID equal to MasterPassGroupId.
Test case Id	TC_C_14_CS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
Note(s): Step 3, 4, and 5 are only executed if the transaction is still running as determined by the <Configured TxStopPoint> .	

Tool validations
* Step 2: Message TransactionEventRequest - triggerReason StopAuthorized - idToken.idToken <Configured masterPass idToken> - idToken.type <Configured masterPass idTokenType> If eventType Ended, then: - transactionInfo.stoppedReason MasterPass * Step 3, 4, 5: Message TransactionEventRequest If eventType Ended, then: - transactionInfo.stoppedReason MasterPass
Post scenario validations: - N/a

6.2.7. Page 103/104 - (2026-02) - TC_B_108_CS - Added SetNetworkProfileRequest main steps

Test case name	Set new NetworkConnectionProfile - Prevent overwriting configured Network Profile slot
Test case Id	TC_B_108_CS
...	
Prerequisite(s)	NetworkConfiguration component variables are writable. The Charging Station supports having at least two NetworkProfiles installed at the same time.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with GetVariablesResponse	1. Test System sends GetVariablesRequest for - OCPPCommCtrlr.NetworkConfigurationPriority
Note: If step 2 attributeValue contains a list of at least two values, then skip to step 7	

Main (Test scenario)	
4. The Charging Station responds with a SetNetworkProfileResponse	3. The Test System sends a SetNetworkProfileRequest with: - configurationSlot is <Configured non-active configurationSlot> - connectionData.ocppVersion OCPP21 - connectionData.ocppInterface Any - connectionData.ocppTransport JSON - connectionData.messageTimeout <Configured messageTimeout> - connectionData.ocppCsmsUrl <Configured ocppCsmsUrl> - connectionData.securityProfile <Active SecurityProfile> - connectionData.basicAuthPassword <Configured basicAuthPassword>
6. The Charging Station responds with SetVariablesResponse	5. Test System sends SetVariablesRequest with - setVariableData[0].component.name OCPPCommCtrlr - setVariableData[0].variable.name NetworkConfigurationPriority - setVariableData[0].attributeValue <Active configuration slot>,<Non-active configuration slot>
<i>Note: For each value in NetworkConfigurationPriority, try to set a NetworkConfiguration instance:</i> FOR slot in attributeValue of OCPPCommCtrlr.NetworkConfigurationPriority DO	
4. 8. Charging Station responds with SetVariablesResponse	3. 7. Test System sends SetVariablesRequest with: - setVariableData[0].component.name NetworkConfiguration - setVariableData[0].component.instance <slot> - setVariableData[0].variable.name MessageTimeout - setVariableData[0].attributeValue 123
END FOR	

Tool validations
* Step 2: Message GetVariablesResponse with: - getVariableResult[0].component = OCPPCommCtrlr - getVariableResult[0].variable = NetworkConfigurationPriority - getVariableResult[0].attributeValue = <a list of at least two values> - getVariableResult[0].attributeStatus = Accepted
* Step 4: Message SetNetworkProfileResponse with: - status = Accepted
* Step 6: Message SetVariablesResponse with: - setVariableResult[0].component = OCPPCommCtrlr - setVariableResult[0].variable = NetworkConfigurationPriority - setVariableResult[0].attributeStatus = Accepted
* Step 4 Step 8: FOR slot in OCPPCommCtrlr.NetworkConfigurationPriority DO Message SetVariablesResponse with: - setVariableResult[0].component.name NetworkConfiguration - setVariableResult[0].component.instance <slot> - setVariableResult[0].variable.name MessageTimeout - setVariableResult[0].attributeStatus Rejected - setVariableResult[0].attributeStatusInfo.reasonCode = PriorityNetworkConf
END FOR

Tool validations
Post scenario validations: - N/a

6.2.8. Page 98 - 113 - (2026-06) - TC_B_105_CS/TC_B_111_CS/TC_B_113_CS -Ensure NetworkConfigurationPriority contains only active configurationSlot

At the start of the test cases (TC_B_105_CS, TC_B_111_CS, and TC_B_113_CS), the NetworkConfigurationPriority is configured to include only the currently active configurationSlot. This setup ensures that the non-active configuration slot can be modified and activated as intended during the execution of the test cases.

In addition, the previously included note was removed because it was no longer required.

Main (Test scenario)	
Charging Station	CSMS
<u>Note:</u> In the following steps <configurationSlot> equals <Configured non-active configurationSlot>	
2. Charging Station responds with SetVariablesResponse	1. Test System sends SetVariablesRequest with - setVariableData[0].component.name OCPPCommCtrlr - setVariableData[0].variable.name NetworkConfigurationPriority - setVariableData[0].attributeValue <Active configuration slot>
...	

Tool validations
* Step 2: Message SetVariablesResponse - setVariableResult.attributeStatus must be Accepted ...

6.2.9. Page 105 - 107 - (2026-02) - TC_B_109_CS - Testcase updated

Added descriptions for WebSocket disconnection and reconnection. NetworkConfiguration.Identity and NetworkConfiguration.BasicAuthPassword are now set using SetNetworkProfileRequest. To ensure SecurityCtrlr.BasicAuthPassword is correctly applied during reconnection, a new valid password is generated and used instead of the originally configured basicAuthPassword.

Test case name	Set new NetworkConnectionProfile - When changing SecurityCtrlr.Identity/BasicAuthPassword the NetworkProfiles.Identity/BasicAuthPassword must be cleared
Test case Id	TC_B_109_CS
...	
Prerequisite(s)	NetworkConfiguration component variables are writable. NetworkConfiguration.Identity and NetworkConfiguration.BasicAuthPassword are set.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with GetVariablesResponse	1. Test System sends GetVariablesRequest for - OCPPCommCtrlr.ActiveNetworkProfile
<u>Note:</u> The result of Step #2 is called <the active configuration>	
4. The Charging Station responds with GetVariablesResponse	3. Test System sends GetVariablesRequest for - NetworkConfiguration[<the active configuration>].Identity - NetworkConfiguration [<the active configuration> .SecurityProfile
<u>Note:</u> The security profile of step 4 is used in the SetNetworkProfileRequest	

Main (Test scenario)	
6. The Charging Station responds with SetNetworkProfileResponse	5. The Test System sends a SetNetworkProfileRequest with: - configurationSlot is <the active configuration> - connectionData.ocppVersion OCPP21 - connectionData.ocppInterface Any - connectionData.ocppTransport JSON - connectionData.messageTimeout <Configured messageTimeout> - connectionData.ocppCsmsUrl <Configured ocppCsmsUrl> - connectionData.securityProfile <security profile from step 4> - connectionData.identity = <Configured ChargingStationId> - connectionData.basicAuthPassword <Configured basicAuthPassword>
6- 8. The Charging Station responds with SetVariablesResponses	5- 7. Test System sends SetVariablesRequests for - SecurityCtrlr.Identity = <configured ChargingStationId> - SecurityCtrlr.BasicAuthPassword = <Configured basicAuthPassword> <generated valid password>
8- 10. The Charging Station responds with GetVariablesResponse	7- 9. Test System sends GetVariablesRequest for - NetworkConfiguration[<the active configuration>].Identity
10. Charging Station reconnects	9. Test System closes the websocket connection
Note: Test System reconnects using values from SecurityCtrlr.Identity/BasicAuthPassword	
11. The Test System closes the WebSocket connection.	
12. The Charging Station sends a HTTP upgrade request with an Authorization header, containing a username/password combination (with the <generated valid password>). Note(s) : - The Authorization header is formatted as follows: AUTHORIZATION: Basic <Base64 encoded(<Configured ChargingStationId>:<generated valid password>)>	13. The Test System validates the username/password combination AND upgrades the connection to a (secured) WebSocket connection.
12- 15. The Charging Station responds with GetVariablesResponse	11- 14. Test System sends GetVariablesRequest for - OCPPCommCtrlr.ActiveNetworkProfile
Note: Restoring the Identity and BasicAuthPassword via SetNetworkProfileRequest and reconnecting.	
14. The Charging Station responds with GetVariablesResponse	13. Test System sends GetVariablesRequest for - NetworkConfiguration[<the active configuration>].SecurityProfile
16- 17. The Charging Station responds with SetNetworkProfileResponse	15- 16. The Test System sends a SetNetworkProfileRequest with: - configurationSlot is <the active configuration> - connectionData.ocppVersion OCPP21 - connectionData.ocppInterface Any - connectionData.ocppTransport JSON - connectionData.messageTimeout <Configured messageTimeout> - connectionData.ocppCsmsUrl <Configured ocppCsmsUrl> - connectionData.securityProfile <security profile from step 14 4> - connectionData.identity = <Configured ChargingStationId> - connectionData.basicAuthPassword <Configured basicAuthPassword>
18. Charging Station reconnects	17. Test System closes the websocket connection
18. The Test System closes the WebSocket connection.	

Main (Test scenario)	
19. The Charging Station sends a HTTP upgrade request with an Authorization header, containing a username/password combination (with the <code><configured BasicAuthPassword></code>). Note(s) : - The Authorization header is formatted as follows: AUTHORIZATION: Basic <Base64 encoded(<Configured ChargingStationId>:<configured BasicAuthPassword>)>	20. The Test System validates the username/password combination AND upgrades the connection to a (secured) WebSocket connection.
20. 22. The Charging Station responds with GetVariablesResponse	19. 21. Test System sends GetVariablesRequest for - OCPPCommCtrlr.ActiveNetworkProfile

Tool validations
* Step 2: Messages GetVariableResponse with: - getVariableResult[0].component = <i>OCPPCommCtrlr</i> - getVariableResult[0].variable = <i>ActiveNetworkProfile</i> - getVariableResult[0].attributeValue = <i><any configuration slot></i> - getVariableResult[0].attributeStatus = <i>Accepted</i>
* Step 4: Messages GetVariableResponse with: - getVariableResult[0].component = <i>NetworkConfiguration</i> - getVariableResult[0].instance = <i><the active configuration></i> - getVariableResult[0].variable = <i>Identity</i> - getVariableResult[0].attributeValue = <i><identity of charging station></i> - getVariableResult[0].variable = <i>SecurityProfile</i> - getVariableResult[0].attributeValue not <i><omitted></i> - getVariableResult[0].attributeStatus = <i>Accepted</i>
* Step 6: Messages SetNetworkConfigurationResponse with: - status = <i>Accepted</i>
* Step 6 8: Messages SetVariableResponse with: - setVariableResult[0].component.name = <i>SecurityCtrlr</i> - setVariableResult[0].variable.name = <i>Identity</i> - setVariableResult[0].attributeStatus = <i>Accepted</i> - setVariableResult[1].component.name = <i>SecurityCtrlr</i> - setVariableResult[1].variable.name = <i>BasicAuthPassword</i> - setVariableResult[1].attributeStatus = <i>Accepted</i>
* Step 8 10: Messages GetVariableResponse with: - getVariableResult[0].component = <i>NetworkConfiguration</i> - getVariableResult[0].variable = <i>Identity</i> - getVariableResult[0].attributeValue = <i><empty></i> - getVariableResult[0].attributeStatus = <i>Accepted</i>
* Step 12 15: Messages GetVariableResponse with: - getVariableResult[0].component = <i>OCPPCommCtrlr</i> - getVariableResult[0].variable = <i>ActiveNetworkProfile</i> - getVariableResult[0].attributeValue = <i><the active configuration></i> - getVariableResult[0].attributeStatus = <i>Accepted</i>

Tool validations
* Step 14: Messages GetVariableResponse with: -getVariableResult[0].component = NetworkConfiguration -getVariableResult[0].variable = SecurityProfile -getVariableResult[0].attributeValue not <omitted> -getVariableResult[0].attributeStatus = Accepted
* Step 16 17: Messages SetNetworkConfigurationResponse with: - status = Accepted
* Step 20 22: Messages GetVariableResponse with: - getVariableResult[0].component = OCPPCommCtrlr - getVariableResult[0].variable = ActiveNetworkProfile - getVariableResult[0].attributeValue = <the active configuration> - getVariableResult[0].attributeStatus = Accepted
Post scenario validations: - N/a

6.2.10. Page 130 - (2026-02) - TC_C_17_CS - Corrected wrong triggerReason [1113]

Wrong mention of triggerReason in step 5 has been changed to chargingState. Tool validation steps clarified.

Main (Test scenario)	
Charging Station	CSMS
1. The Test System closes the WebSocket connection AND does not accept a reconnect.	
<u>Manual Action</u> : Present valid idToken which is already configured in the Authorization Cache	
<u>Note(s)</u> : The Test System will wait for 5 seconds	
2. The Test System accepts reconnection attempt from the Charging Station.	
<u>Note(s)</u> : - The Charging Station will empty its Transaction message queue. This will contain one or more TransactionEventRequest messages	
3. The Charging Station sends a TransactionEventRequest	4. The Test System responds with a TransactionEventResponse with idTokenInfo.status Invalid (if idToken is not omitted)
3. The Charging Stations empties its Transaction message queue: TransactionEventRequest	4. The Test System responds with a TransactionEventResponse
<u>Note(s)</u> : - This will contain one or more TransactionEventRequest messages	<u>Note(s)</u> : - The Test System will respond to the TransactionEventRequest containing the idToken, with idtokenInfo.status Invalid
5. The Charging Station sends a TransactionEventRequest with triggerReason SuspendedEVSE transactionInfo.chargingState SuspendedEVSE	6. The Test System responds with a TransactionEventResponse
<u>Note(s)</u> : Steps 5 & 6 can occur before or during the sending of messages from the offline queue (steps 3 & 4).	

Tool validations
* Step 5: Step 3: Message TransactionEventRequest A message with: All Message(s): TransactionEventRequest - offline must be true One of the Message(s): TransactionEventRequest - triggerReason <i>Authorized</i> - idToken.idToken <i><Configured valid_idtoken_idtoken></i> - idToken.type <i><Configured valid_idtoken_type></i> - offline True true * Step 5: A message with: - offline <i>false</i> or omitted - transactionInfo.chargingState <i>SuspendedEVSE</i> - triggerReason <i>ChargingStateChanged</i>

6.2.11. Page 119 - 161 - (2026-04) - C Test cases - Updated configuration state

An additional **Configuration State** setting **AuthorizeRemoteStart** to *true* was added to the following test cases:

TC_C_32, TC_C_33, TC_C_34, TC_C_36, TC_C_46, TC_C_37, TC_C_08, TC_C_09, TC_C_12, TC_C_10, TC_C_11, TC_C_13, TC_C_57, TC_C_27, TC_C_28, TC_C_31, TC_C_29, TC_C_30, TC_C_58

This ensures that the test case does not skip the authorization step when executed with a remote **RequestStartTransactionRequest**.

Before (Preparations)
Configuration State: ... AuthorizeRemoteStart is <i>true</i> (If ReadWrite) ...

6.2.12. Page 143 - 167 - (2026-04) - TC_C_14_CS/TC_C_45_CS/TC_C_47_CS/TC_C_48_CS/TC_C_49_CS - Removed configuration variable masterPassGroupId

The *<Configured MasterPassGroupId>* variable has been removed from the OCTT. The MasterPassGroupId can be generated by the OCTT when the *<Configured masterpass_idtoken_idtoken>* is received and returned to the charging station as part of the standard **AuthorizeResponse**.

Configuration states have been updated to ensure the Masterpass idToken is configured when necessary.

Before (Preparations)
Configuration State: TxCtrlr.TxStartPoint should contain <i>Authorized</i> or <i>PowerPathClosed</i> and not contain <i>ParkingBayOccupancy</i> or <i>EVConnected</i> AuthCtrlr.MasterPassGroupId is <i><Configured MasterPassGroupId></i> Masterpass idToken is configured.

Where applicable the **AuthorizeResponse** and **TransactionEventResponse** have been updated to return the Generated masterPassGroupId.

Main (Test scenario)
2. The Test System responds with an AuthorizeResponse with - idTokenInfo.groupIdToken.idToken <i><Configured masterPassGroupId></i> <i><Generated masterPassGroupId></i>

Main (Test scenario)	
	4. The Test System responds with a TransactionEventResponse with idTokenInfo.groupIdToken.idToken <i><Configured masterPassGroupId></i> <Generated masterPassGroupId>

6.2.13. Page 147 - (2026-02) - TC_C_38_CS - Removed memory state IdTokenCached [1164]

There is no need to call memory state IdTokenCached if the Authorization Cache has been disabled. This has therefore been removed.

Test case name	Clear Authorization Data in Authorization Cache - Rejected
Test case Id	TC_C_38_CS
...	

Before (Preparations)
Configuration State: AuthCacheEnabled is <i>false</i> (If implemented)
Memory State: <i>IdTokenCached</i> for <i><Configured valid IdToken fields></i>
Reusable State(s): N/a

[...]

NOTE: If the Charging Station supports ISO15118, this testcase needs to be executed using EIM.

6.2.14. Page 173/174 - (2026-02) - TC_C_26_CS - Clarification of main test and tool validation steps.

Main (Test scenario)	
Charging Station	CSMS
1. The Charging Station notifies the CSMS about the current state of all connectors.	2. The Test System responds accordingly.
3. The Charging Station sends a TransactionEventRequest . The Charging Station empties its Transaction message queue: TransactionEventRequest	4. The Test System responds with a TransactionEventResponse
Note(s): - The Charging Station will empty its Transaction message queue. This will contain one or more TransactionEventRequest messages	Note(s): - The Test System will respond to the TransactionEventRequest containing the idToken, with idtokenInfo.status Invalid
5. The Charging Station sends a TransactionEventRequest	6. The Test System responds with a TransactionEventResponse
Note(s): - This will contain chargingState SuspendedEVSE	
Note(s): Steps 5 & 6 can occur before or during the sending of messages from the offline queue (steps 3 & 4).	
7. Execute Reusable State <i>StopAuthorized</i>	
8. Execute Reusable State <i>EVConnectedPostSession</i>	
9. Execute Reusable State <i>EVDisconnected</i>	

Tool validations
* Step 1: Message: StatusNotificationRequest - connectorStatus must be <i>Occupied</i> Message: NotifyEventRequest - eventData[0].trigger must be <i>Delta</i> - eventData[0].actualValue must be <i>Occupied</i> - eventData[0].component.name must be <i>Connector</i> - eventData[0].variable.name must be <i>AvailabilityState</i> * Step 3: All Message(s): TransactionEventRequest - offline must be <i>true</i> One of the Message(s): TransactionEventRequest - chargingState must be <i>Charging</i> * Step 4: One of the Message(s): TransactionEventRequest - chargingState must be <i>SuspendedEVSE</i> * Step 5: TransactionEventRequest - chargingState must be <i>SuspendedEVSE</i>
Post scenario validations: N/a

6.2.15. Page 184 - (2026-02) - TC_C_100_CS - Test case removed

This testcases has been removed, because it duplicates TC_E_05_CS.

6.2.16. Page 185 - (2026-02) - TC_C_103_CS - Test updated

This test case has been updated to always execute until **transactionInfo.chargingState** *Charging* has been reached in accordance with C17.FR.04. Additionally, reusable states were implemented where applicable.

Test case name	Authorization with prepaid card - success
Test case Id	TC_C_103_CS
...	

Main (Test scenario)	
Charging Station	CSMS
<u>Manual Action:</u> EV driver presents prepaid card	
1. The CS sends a AuthorizeRequest +	2. The Test System responds with an AuthorizeResponse with + - idTokenInfo.status <i>Accepted</i> - idTokenInfo.cacheExpiryDateTime <i><current date/time></i>
1. Execute Reusable State <i>Authorized</i> (prepaid card) with AuthorizeResponse : - idTokenInfo.status <i>Accepted</i> - idTokenInfo.cacheExpiryDateTime <i><current date/time></i>	
<u>Note:</u> Only when TxStartPoint does not contain ParkingBayOccupancy or Authorized: Manual action: <i>Plug cable in</i>	
2. Execute Reusable State <i>EVConnectedPreSession</i>	
3. Execute Reusable State <i>EnergyTransferStarted</i>	

Main (Test scenario)	
3- 4. The Charging Station sends a TransactionEventRequest	4- 5. The Test System responds with a TransactionEventResponse Note(s) : <i>The Test System will respond to the TransactionEventRequest containing the idToken, with:</i> - idTokenInfo.status <i>Accepted</i> - idTokenInfo.cacheExpiryDateTime <i><current date/time></i> - transactionLimit.maxCost = 123.32

Tool validations
* Step 1: Message AuthorizeRequest idToken.idToken <i><prepaid card id></i> idToken.type = "ISO14443" or "ISO15693" * Step 3: Step 4: Message TransactionEventRequest - eventType <i>Updated (for TxStartPoint ParkingBayOccupancy) or Started (for all other TxStartPoints)</i> - idToken.idToken <i><prepaid card id></i> - idToken.type = <i><idTokenType from step 1></i> Post scenario validations: - transactionInfo.chargingState must transition to <i>Charging</i>

6.2.17. Page 209 - (2026-02) - TC_C_121_CS - Added tool validations for triggerReason LimitSet

Test case name	Incremental authorization - increasing enabled
Test case Id	TC_C_121_CS
...	

Tool validations
* Step 2: Message SetDefaultTariffResponse - status must be <i>Accepted</i>
* Step 3 and 4 and 5 combined: (First transactionEventRequest with idToken) Message TransactionEventRequest with: - idToken must be <i><Not empty></i> - idToken.type must be <i>DirectPayment</i> - transactionInfo.transactionId must be <i><transactionId></i> - transactionInfo.transactionLimit.maxCost must be <i>10.00</i>
* Step 6: Message TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionId must be <i><transactionId></i> - transactionInfo.transactionLimit.maxCost must be <i>15.00</i>
* Step 8: Message TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionId must be <i><transactionId></i> - transactionInfo.transactionLimit.maxCost must be <i>20.00</i>
* Step 10: Message TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionId must be <i><transactionId></i> - transactionInfo.transactionLimit.maxCost must be <i>25.00</i>
Post scenario validations: N/a

6.2.18. Page 182 - (2026-02) - TC_C_55_CS -Explicit mention of offline transactions only

Test case name	Authorization using Contract Certificates 15118 - Offline - ContractValidationOffline is false
Test case Id	TC_C_55_CS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Test System closes the WebSocket connection AND does not accept a reconnect.	
<u>Manual Action</u> : Drive EV into parking bay.	
<u>Manual Action</u> : Connect the EV and EVSE.	
<u>Note(s)</u> : The tool will wait for <Configured Transaction Duration> 61 seconds (ISO15118-2 timeout) .	
2. The Test System accepts the reconnection attempt from the Charging Station after <Configured Transaction Duration> 61 seconds (ISO15118-2 timeout) .	

Main (Test scenario)
...
Tool validations
...
Post scenario validations: EV is not authorized and shall not charge: Charging Station does not send TransactionEventRequest with: While offline, the EV is not authorized and shall not charge: Charging Station does not send an TransactionEventRequest with: - offline = true - triggerReason = Authorized or chargingState = Charging

6.2.19. Page 214 - (2026-02) - TC_C_124_CS and TC_C_128_CS to TC_C_130_CS - Corrected requirement reference and removed note

Corrected the requirement reference to E16.FR.03 and removed C24.FR.03 from TC_C_124_CS. The second note was incorrect and has been removed, as the first note sufficiently covers the test case.

6.2.20. Page 282 - (2026-06) - TC_E_41_CS, TC_E_42_CS, TC_E_50_CS and TC_E_51_CS - retry mechanism applied to TransactionEventRequest after suspending energy transfer on EV side

Due to the complexity of having different TxStartPoints and Charging Station manufacturers also have the option to combine TransactionEvent triggers in some specific cases, it has been decided that it would be easier to test this testcase by using the TransactionEventRequest after suspending energy transfer on EV side.

All:

Before (Preparations)
...
Reusable State(s): State is Authorized State is EVConnectedPreSession State is EnergyTransferStarted
Main (Test scenario)
...
Tool validations
* Step 1: -triggerReason SignedDataReceived * Step 3: -triggerReason ChargingStateChanged -chargingState SuspendedEVSE * Step 5: - Needs to be sent a number of times equal to <Configured message_attempts_transaction_event> with an interval of (<Configured message_attempts_transaction_event_interval> * the number of preceding transmissions of this same message) + OCPPCommCtrlr.MessageTimeout.Default. - The Test System waits an additional MessageAttemptsTransactionEvent iteration where the interval is multiplied again, to validate if the Charging Station stops resending the TransactionRequest message(s).
Post scenario validations: N/a

TC_E_41_CS:

Main (Test scenario)	
Charging Station	CSMS
Manual Action: <i>The EV suspends the energy transfer.</i>	
Note(s): <i>Step 1 is repeated for the configured number of times</i>	
1. The Charging Stations sends a TransactionEventRequest	

TC_E_42_CS:

Main (Test scenario)	
Charging Station	CSMS
Manual Action: <i>The EV suspends the energy transfer.</i>	
Note(s): <i>The tool will ignore the first TransactionEventRequest and respond to the second TransactionEventRequest</i>	
1. The Charging Stations sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse

TC_E_50_CS:

Main (Test scenario)	
Charging Station	CSMS
Manual Action: <i>The EV suspends the energy transfer.</i>	
Note(s): <i>Step 1 and 2 is repeated for the configured number of times</i>	
1. The Charging Stations sends a TransactionEventRequest	2. The Test System responds with a CallError with errorCode InternalError

TC_E_51_CS:

Main (Test scenario)	
Charging Station	CSMS
Manual Action: <i>The EV suspends the energy transfer.</i>	
Note(s): <i>The tool will send a CallError with errorCode InternalError to the first TransactionEventRequest and respond to the second TransactionEventRequest with a TransactionEventResponse .</i>	
1. The Charging Stations sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse

6.2.21. Page 354 - (2026-04) - TC_G_11_CS - Clarity provided regarding sending status notifications of connectors not in use

Additional note and optional test case steps were added to provide clarity regarding when status notifications of the connectors not in use can be sent.

Test case name	Change Availability EVSE - With ongoing transaction
Test case Id	TC_G_11_CS

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a ChangeAvailabilityResponse	1. The Test System sends a ChangeAvailabilityRequest with operationalStatus Inoperative and evse.id <Configured evseld>
3. The Charging Station notifies the CSMS about the current state of the connectors of the configured evse, that are NOT in use.	4. The Test System responds accordingly.
Note(s): <i>Step 3 and 4 are optional and can be sent by the charging station if it chooses to immediately report the status of connectors of the same EVSE that are NOT in use .</i>	
Note(s): Wait for <Configured Transaction Duration>	

Main (Test scenario)	
5. 3. Execute Reusable State <i>StopAuthorized</i>	
6. 4. The Charging Station notifies the CSMS about the current state of the connectors of the configured evse.	7. 5. The Test System responds accordingly.
8. 6. Execute Reusable State <i>EVConnectedPostSession</i>	
9. 7. The Charging Station notifies the CSMS about the current state of the connectors of the configured evse.	10. 8. The Test System responds accordingly.
11. 9. Execute Reusable State <i>EVDisconnected</i>	
12. 10. The Charging Station notifies the CSMS about the current state of the connectors of the configured evse.	13. 11. The Test System responds accordingly.
14. 12. Execute Reusable State <i>ParkingBayUnoccupied</i>	
15. 13. The Charging Station notifies the CSMS about the current state of the connectors of the configured evse.	16. 14. The Test System responds accordingly.
Note(s): Steps 4, 5, 7, 8, 10, 11, 13, and 14 6, 7, 9, 10, 12, 13, 15, and 16 will only be executed if the previous step ended the transaction	
17. The test system will wait the <configured max timeout period>	
Note(s) : If all connectors of the configured evse have not yet reported their status, the test system will wait for the configured max timeout period to receive them.	

Tool validations
* Step 2: Message ChangeAvailabilityResponse - status <i>Scheduled</i> * Step 3: For all connectorId's of the Configured evseld not in use: Message: StatusNotificationRequest - connectorStatus <i>Unavailable</i> - evseld <Configured evseld> - connectorId Not <Configured connectorId> Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Unavailable"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> - eventData[0].component.evse.id <Configured evseld> - eventData[0].component.evse.connectorId Not <Configured connectorId> * Step 6, 9, 12, 15: Step 4, 7, 10, 13: Message: StatusNotificationRequest - connectorStatus <i>Unavailable</i> - evseld <Configured evseld> - connectorId <Configured connectorId> - when applicable this is also sent for other connectors of the same EVSE Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Unavailable"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> - eventData[0].component.evse.id <Configured evseld> - eventData[0].component.evse.connectorId <Configured connectorId> - when applicable this is also sent for other connectors of the same EVSE Post scenario validations: - A message to report the state of a connector has been received for all connectors of the configured EVSE.

6.2.22. Page 358 - (2026-04) - TC_G_14_CS - Clarity provided regarding sending status notifications of connectors not in use

Additional note added to provide clarity regarding when status notifications of the connectors not in use can be sent.

Test case name	Change Availability Charging Station - Operative to operative
Test case Id	TC_G_12_CS

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a ChangeAvailabilityResponse	1. The Test System sends a ChangeAvailabilityRequest with operationalStatus Inoperative
3. The Charging Station notifies the CSMS about the current state of the connectors of the EVSEs that do not have an active transaction.	4. The Test System responds accordingly.
Note(s) : Step 3 and 4 are optional and can be sent by the charging station if it chooses to immediately report the status of connectors of the EVSEs that are NOT in use.	
Note(s) :- Wait for <Configured Transaction Duration>	
5. Execute Reusable State StopAuthorized	
...	
Note(s) : Steps 6, 7, 9, 10, 12, 13, 15, and 16 will only be executed if the previous step ended the transaction	
17. The test system will wait the <configured max timeout period>	
Note(s) : If all connectors of the configured evse have not yet reported their status, the test system will wait for the configured max timeout period to receive them.	

Tool validations
* Step 2: Message ChangeAvailabilityResponse - status <i>Scheduled</i> * Step 3: For all connectorId's and evseld's not in use: Message: StatusNotificationRequest - connectorStatus <i>Unavailable</i> - evseld Not <configured evseld> AND connectorId Not <configured connectorId> Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Unavailable"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> - eventData[0].component.evse.id Not <configured evseld> AND eventData[0].component.evse.connectorId Not <configured connectorId> * Step 7, 10, 13, 16: Message: StatusNotificationRequest - connectorStatus <i>Unavailable</i> - evseld not 0 - connectorId not 0 Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Unavailable"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i>

6.2.23. Page 363 - (2026-06) - TC_G_17_CS - Updated tool and post scenario validations

The post scenario validation is not applicable for change availability at a connector level and has therefore been removed.

Step 7 validations for other status notifications on the same EVSE have been removed. The charging station will **not** report *Unavailable* for the other connectors of the same EVSE, when a specific connector receives a change availability request.

Test case name	Change Availability Connector - With ongoing transaction
Test case Id	TC_G_17_CS
...	

Tool validations
* Step 2: Message ChangeAvailabilityResponse - status <i>Scheduled</i> * Step 7: Message: StatusNotificationRequest - connectorStatus <i>Unavailable</i> - evseld <i><Configured evseld></i> - connectorId <i><Configured connectorId></i> - when applicable this is also sent for other connectors of the same EVSE Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Unavailable"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].component.evse <i>not omit</i> - eventData[0].component.evse.id <i><Configured evseld></i> - eventData[0].component.evse.connectorId <i><Configured connectorId></i> - eventData[0].variable.name <i>"AvailabilityState"</i> - when applicable this is also sent for other connectors of the same EVSE
Post scenario validations: - A message to report the state of a connector has been received for all connectors. N/a

6.2.24. Page 292 - (2026-06) - TC_E_101_CS - Reusable state EnergyTransferStarted set in preparations

The reusable state **EnergyTransferStarted** is initiated in the preparations to avoid potentially receiving a **TransactionEvent** with **triggerReason** *ChargingStateChanged* while the test system is expecting the **triggerReason** *LimitSet*

Test case name	Transactions with fixed cost, energy or time - CSMS calculates costs (through CostUpdatedRequest) and CS specifies limit
Test case Id	TC_E_101_CS
...	

Before (Preparations)
...
Reusable State(s): State is <i>Authorized</i> State is <i>EVConnectedPreSession</i> State is <i>EnergyTransferStarted</i>

6.2.25. Page 294/295/296 - (2026-02) - TC_E_102_CS - Test case and tool validations corrected.

Step 11 was corrected to align with E16.FR.04. Tool validations for **transactionInfo.transactionLimit.maxTime** removed as they are not applicable.

Test case name	Transactions with fixed cost, energy or time - CSMS and CS both specify limits
Test case Id	TC_E_102_CS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Charging Station sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse with transactionLimit.maxEnergy 20000
1. Execute Reusable State EnergyTransferStarted with: (At the first TransactionEventResponse) - transactionLimit.maxEnergy 20000	
3. 2. The Charging Station sends a TransactionEventRequest	4. 3. The Test System responds with a TransactionEventResponse
5. 4. The Charging Station sends a TransactionEventRequest	6. 5. The Test System responds with a TransactionEventResponse
Note: Steps 5 4 and 6 5 may repeat	
Manual Action: Enter energy limit 10 kWh for transaction	
7. 6. The Charging Station sends a TransactionEventRequest	8. 7. The Test System responds with a TransactionEventResponse
9. 8. The Charging Station sends a TransactionEventRequest	10. 9. The Test System responds with a TransactionEventResponse
Note: Steps 9 8 and 10 9 may repeat	
Manual Action: Enter energy limit 40 kWh for transaction Note: CS should not allow this or at least not send a higher limit than 20000 as set by CSMS	
11. The Charging Station sends a TransactionEventRequest	12. The Test System responds with a TransactionEventResponse
10. The Charging Station SHALL NOT send a TransactionEventRequest with triggerReason is LimitSet	
13. 11. The Charging Station sends a TransactionEventRequest	14. 12. The Test System responds with a TransactionEventResponse with transactionLimit.maxEnergy 2000
15. 13. The Charging Station sends a TransactionEventRequest	16. 14. The Test System responds with a TransactionEventResponse
17. 15. The Charging Station sends a TransactionEventRequest	18. 16. The Test System responds with a TransactionEventResponse
Note: Steps 17 15 and 18 16 may repeat	
19. 17. The Charging Station sends a TransactionEventRequest	20. 18. The Test System responds with a TransactionEventResponse

Main (Test scenario)
<u>Note: Step 49 17 should be triggered when 2kWh has been charged</u>

Tool validations
* Step 1: Message: TransactionEventRequest - transactionInfo.transactionLimit must be <omitted> * Step 3 2: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be 20000 - transactionInfo.transactionLimit.maxTime must be 3600 <omitted> - transactionInfo.transactionLimit.maxCost must be <omitted> * Step 5 4: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionLimit must be <omitted> * Step 7 6: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be 10000 - transactionInfo.transactionLimit.maxTime must be 3600 <omitted> - transactionInfo.transactionLimit.maxCost must be <omitted>

Tool validations
* Step 8: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionLimit must be <i><omitted></i> * Step 10: No TransactionEventRequest with triggerReason <i>LimitSet</i> is received Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must NOT be <i>LimitSet</i> IF triggerReason is <i>LimitSet</i> THEN - transactionInfo.transactionLimit.maxEnergy must be <i>20000</i> - transactionInfo.transactionLimit.maxTime must be <i>3600</i> - transactionInfo.transactionLimit.maxCost must be <i><omitted></i> ENDIF * Step 11: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionLimit must be <i><omitted></i> * Step 13: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be <i>2000</i> - transactionInfo.transactionLimit.maxTime must be <i>3600</i> <i><omitted></i> - transactionInfo.transactionLimit.maxCost must be <i><omitted></i> * Step 15: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionLimit must be <i><omitted></i> * Step 17: Message: TransactionEventRequest - eventType must be <i>Updated</i> (or <i>Ended</i> if TxStopPoint contains "EnergyTransfer") - triggerReason must be <i>EnergyLimitReached</i> - transactionInfo.transactionLimit must be <i><omitted></i> - transactionInfo.chargingState must be <i>SuspendedEVSE</i> (or <i>EVConnected</i> if TxStopPoint contains "EnergyTransfer")
Post scenario validations: N/a

6.2.26. Page 297/298 - (2026-02) - TC_I_119_CS - Added tool validations for totalCost.total

Test case name	Local Cost Calculation - Cost Details of Transaction - with tariff conditions
Test case Id	TC_I_119_CS
...	

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. The Charging Station responds with SetDefaultTariffResponse	1. The Test System sends a SetDefaultTariffRequest with evseld 0 tariff.tariffId <i>Test System1</i> tariff.currency <i>EUR</i> tariff.reservationFixed.prices[0].priceFixed <i>20.00</i> tariff.reservationFixed.taxRates[0].type <i>t5_0</i> tariff.reservationFixed.taxRates[0].tax <i>5.0</i> tariff.fixedFee.prices[0].priceFixed <i>10.00</i> tariff.fixedFee.prices[2 1].priceFixed <i>5.00</i> tariff.fixedFee.taxRates[0].type <i>t5_0</i> tariff.fixedFee.taxRates[0].tax <i>5.0</i> tariff.chargingTime.prices[0].priceMinute <i>6.00</i> tariff.chargingTime.prices[0].conditions.minTime <i>30</i> tariff.chargingTime.prices[1].priceMinute <i>60.00</i> tariff.chargingTime.taxRates[0].type <i>t20_0</i> tariff.chargingTime.taxRates[0].tax <i>20.0</i>

Tool validations
...
Post scenario validations: <u>Note:</u> The value of TransactionEventRequest.timestamp of step 6 where TransactionEventRequest.eventType is Started is called <TxStartDateTime>. * A message TransactionEventRequest must have been received with: ... - costDetails.totalCost.total.exclTax must be <sum of all exclTax subtotals after rounding> - costDetails.totalCost.total.inclTax must be <sum of all inclTax subtotals after rounding> ..

6.2.27. Page 297/298 - (2026-02) - TC_E_103_CS - Test case and tool validation corrections

Updated reusable state usage to include **transactionLimit.maxCost** set through CSMS and removed **transactionLimit.maxCost** from step 5. Incorrect references to elapsed time in the note after step 9 and to **EnergyLimitsReached** in step 8 tool validations have been corrected.

Test case name	Transactions with fixed cost, energy or time - CS calculates costs and CSMS specifies limit
Test case Id	TC_E_103_CS
...	

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. The Charging Station responds with a SetDefaultTariffResponse	1. The Test System sends a SetDefaultTariffRequest with evseld 0 tariff.tariffId <i>Tariff1</i> tariff.currency <i>EUR</i> tariff.chargingTime.taxRate[0].type <i>MyTax1</i> tariff.chargingTime.taxRate[0].tax 0 tariff.chargingTime.prices[0].priceMinute 60 tariff.idleTime.taxRate[0].type <i>MyTax2</i> tariff.idleTime.taxRate[0].tax 0 tariff.idleTime.prices[0].priceMinute 60
3. Execute Reusable State <i>EnergyTransferStarted</i> with: (At the first <i>TransactionEventResponse</i>) - transactionLimit.maxCost 120.00 Manual Action: Enter cost limit 120.00 for transaction	
4. The Charging Station sends a TransactionEventRequest	5. The Test System responds with a TransactionEventResponse with transactionLimit.maxCost 120.00
6. The Charging Station sends a TransactionEventRequest	7. The Test System responds with a TransactionEventResponse
<i>Note: Steps 6 and 7 may repeat</i>	
8. The Charging Station sends a TransactionEventRequest	9. The Test System responds with a TransactionEventResponse
<i>Note: Step 10 should trigger automatically after 120 seconds have been elapsed since start of transaction</i> <i>Note: Step 8 should trigger automatically after 120.00 euro is charged.</i>	

Tool validations
* Step 2: Message SetDefaultTariffResponse - status must be <i>Accepted</i> ... * Step 8: Message: TransactionEventRequest - eventType must be <i>Updated</i> (or <i>Ended</i> if TxStopPoint contains "EnergyTransfer") - triggerReason must be <i>EnergyLimitReached</i> CostLimitReached

6.2.28. Page 301/302 - (2026-02) - TC_E_105_CS - Test case and tool validation corrections

Reusable state *EnergyTransferStarted* used and maxTime reduced to 180. Step 10 **triggerReason** tool validation corrected.

Test case name	Transactions with fixed cost, energy or time - CSMS specifies time limit
Test case Id	TC_E_105_CS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Charging Station sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse with transactionLimit.maxTime 3600

Main (Test scenario)	
1. Execute Reusable State <i>EnergyTransferStarted</i> with: (At the first <i>TransactionEventResponse</i>) transactionLimit.maxTime 3600	
3. 2. The Charging Station sends a TransactionEventRequest	4. 3. The Test System responds with a TransactionEventResponse with
5. 4. The Charging Station sends a TransactionEventRequest	6. 5. The Test System responds with a TransactionEventResponse with transactionLimit.maxTime 300 180
7. 6. The Charging Station sends a TransactionEventRequest	8. 7. The Test System responds with a TransactionEventResponse
9. 8. The Charging Station sends a TransactionEventRequest	10. 9. The Test System responds with a TransactionEventResponse
Note: Steps 9 8 and 10 9 may repeat	
11. 10. The Charging Station sends a TransactionEventRequest	12. 11. The Test System responds with a TransactionEventResponse
Note: Step 11 10 should be triggered when 300 180 seconds have elapsed since start of transaction	

Tool validations
* Step 1: Message: TransactionEventRequest - transactionInfo.transactionLimit must be <omitted> * Step 3: Step 2: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be <omitted> - transactionInfo.transactionLimit.maxTime must be 3600 - transactionInfo.transactionLimit.maxCost must be <omitted> * Step 5: Step 4: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionLimit must be <omitted> * Step 7: Step 6: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be <omitted> - transactionInfo.transactionLimit.maxTime must be 300 180 - transactionInfo.transactionLimit.maxCost must be <omitted> * Step 9: Step 8: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionLimit must be <omitted> * Step 11: Step 10: Message: TransactionEventRequest - eventType must be <i>Updated</i> (or <i>Ended</i> if TxStopPoint contains "EnergyTransfer") - triggerReason must be EnergyLimitReached TimeLimitReached - transactionInfo.transactionLimit must be <omitted> - transactionInfo.chargingState must be <i>SuspendedEVSE</i> (or <i>EVConnected</i> if TxStopPoint contains "EnergyTransfer")

6.2.29. Page 303/304 - (2026-02) - TC_E_106_CS - Test case and tool validation corrections

Clarification of note and tool validations.

Test case name	Transactions with fixed cost, energy or time - CS specifies energy limit
Test case Id	TC_E_106_CS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
<u>Note:</u> Steps 3 and 4 may repeat while entering new limit	

Tool validations
* Step 1: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> or <i>MeterValuePeriodic</i> - transactionInfo.transactionLimit.maxEnergy must be 20000 - transactionInfo.transactionLimit.maxTime must be <omitted> - transactionInfo.transactionLimit.maxCost must be <omitted> * Step 3: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>MeterValuePeriodic</i> - transactionInfo.transactionLimit must be <omitted> * Step 5: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> or <i>MeterValuePeriodic</i> - transactionInfo.transactionLimit.maxEnergy must be 2000 - transactionInfo.transactionLimit.maxTime must be <omitted> - transactionInfo.transactionLimit.maxCost must be <omitted> * Step 7: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>MeterValuePeriodic</i> - transactionInfo.transactionLimit must be <omitted> ...

6.2.30. Page 305/306 - (2026-02) - TC_E_107_CS - Tool validation clarifications.

Clarification of tool validations.

Test case name	Transactions with fixed cost, energy or time - CS specifies time limit
Test case Id	TC_E_107_CS
...	

Tool validations
* Step 1: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be <i><omitted></i> - transactionInfo.transactionLimit.maxTime must be <i>3600</i> - transactionInfo.transactionLimit.maxCost must be <i><omitted></i> * Step 3: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>MeterValuePeriodic</i> - transactionInfo.transactionLimit must be <i><omitted></i> * Step 5: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>LimitSet</i> - transactionInfo.transactionLimit.maxEnergy must be <i><omitted></i> - transactionInfo.transactionLimit.maxTime must be <i>20</i> - transactionInfo.transactionLimit.maxCost must be <i><omitted></i> * Step 7: Message: TransactionEventRequest - eventType must be <i>Updated</i> - triggerReason must be <i>MeterValuePeriodic</i> - transactionInfo.transactionLimit must be <i><omitted></i> ...

6.2.31. Page 313/314 - (2026-02) - TC_E_114_CS/TC_E_115_CS - Added main step to check the transaction is not resumed

Added main step to check the transaction is not resumed, but Ended.

Test case Id	TC_E_114_CS / TC_E_115_CS
...	
Main (Test scenario)	
Charging Station	CSMS
<u>Manual Action:</u> Cut the power to the Charging Station, wait 30 seconds and restore power to the Charging Station	
1. The Charging Station sends a TransactionEventRequest	2. The Test System responds with a TransactionEventResponse

Tool validations
* Step 1: Message: TransactionEventRequest - eventType must be <i>Ended</i> - triggerReason must NOT be <i>TxResumed</i> - stoppedReason must be <i>PowerLoss</i> or <i>Reboot</i>
Post scenario validations (TC_E_114_CS): Charging Station boots normally, but no Transaction related messages are sent to the CSMS with <i>triggerReason</i> = <i>TxResumed</i>, since resuming is not supported. Note: It is allowed that some TransactionEventRequests are sent to report that the running transactions have been ended, e.g. via <i>triggerReason</i> = <i>AbnormalCondition</i>.
Post scenario validations (TC_E_115_CS): Charging Station boots normally, but no Transaction related messages are sent to the CSMS with <i>triggerReason</i> = <i>TxResumed</i>, since resuming is disabled.

6.2.32. Page 610 - (2026-02) - TC_L_07_CS - Path to non-existent firmware updated

Test case name	Secure Firmware Update - DownloadFailed
Test case Id	TC_L_07_CS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a UpdateFirmwareResponse	1. The Test System sends a UpdateFirmwareRequest with firmware.installDateTime <Current DateTime - 2 hours> firmware.location "does_not_exist_download" + <Configured firmware location> + "_does_not_exist" firmware.retrieveDateTime <Current DateTime - 2 hours> firmware.signingCertificate <Configured signingCertificate> firmware.signature <Configured signature> Note(s) : - The firmware location is mutated such that "does_not_exist_download" is pre-appended to the file path.
...	

6.2.33. Page 421 - (2026-02) - TC_I_107_CS - Used reusable state Authorized

Test case name	Receive Driver Tariff - CS cannot process tariff - UseDefault/CentralCost
Test case Id	TC_I_107_CS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with SetDefaultTariffResponse	1. The Test System sends a SetDefaultTariffRequest with evseld 0 tariff.tariffId Test System1A tariff.currency EUR tariff.energy.prices[0].priceKwh 1.00
Note: Test system returns a TariffType in the AuthorizeResponse that is too large to handle, i.e. more than MaxElements number of fixedFee prices.	
3. Execute Reusable State Authorized with AuthorizeResponse with a tariff element that contains more than TariffCostCtrlr.MaxElements[Tariff] TariffFixedType elements: idTokenInfo.status Accepted tariff.tariffId Test System1 tariff.currency EUR tariff.fixedFee.prices[0].priceFixed 0.00 tariff.fixedFee.prices[0].conditions.startTimeOfDay 00:00 tariff.fixedFee.prices[1].priceFixed 0.01 tariff.fixedFee.prices[1].conditions.startTimeOfDay 00:01 ... tariff.fixedFee.prices[<max elements + 1>].priceFixed <increasing amount> tariff.fixedFee.prices[<max elements + 1>].conditions.startTimeOfDay <increasing time>	
Option A : If AuthCtrlr.AuthorizeRemoteStart = false (when variable is readOnly): Manual Action: Present idToken.	

Main (Test scenario)	
Option B: If AuthCtrlr.AuthorizeRemoteStart = true: 4. The Charging Station sends an RequestStartTransactionResponse	3. The Test System sends an RequestStartTransactionRequest with idToken = <configured valid_idtoken_idtoken>
Note: It is allowed that Charging Station sends a CALLRESULTERROR if it cannot process the tariff in the response.	
Note 2: A transaction may have been started already when TxStartPoint contains ParkingBayOccupancy or EVConnected. Energy shall be delivered, because idToken is Accepted and Charging Station continues with default tariff or CSMS does central cost calculation	
4. Execute Reusable State EnergyTransferStarted	
5. The Charging Station sends a NotifyEventRequest	6. The Test System sends a NotifyEventResponse

Tool validations
* Step 2: Message SetDefaultTariffResponse - status Accepted * Step 5: Message NotifyEventRequest - trigger = Alerting or Delta - component = TariffCostCtrlr - variable = Problem - actualValue = true Note: It is allowed to also receive other NotifyEventRequests from the Charging Station.
Post scenario validations: - Charging Station SHALL deliver energy.

6.2.34. Page 424/425 - (2026-02) - TC_I_109_CS - Added reusable state EnergyTransferStarted

Main (Test scenario)	
Charging Station	CSMS
...	
2. The Charging Station sends a TransactionEventRequest	3. The Test System responds with a TransactionEventResponse
2. Execute Reusable State EnergyTransferStarted	
5. 4. The Charging Station responds with GetTariffsResponse	4. 3. The Test System sends a GetTariffsRequest with evseld <Configured evseld>
6. 5. Execute Reusable State StopAuthorized	
8. 7. The Charging Station responds with GetTariffsResponse	7. 6. The Test System sends a GetTariffsRequest with evseld <Configured evseld>
9. 8. Execute Reusable State Authorized	
10. 9. The Charging Station sends a TransactionEventRequest Note(s): - This step needs to be executed after the <Configured ev_connection_timeout> expires, if the transaction has been started. So in the case TxStartPoint contains ParkingBayOccupancy OR Authorized	11. 10. The Test System responds with a TransactionEventResponse

Main (Test scenario)	
13 12 . The Charging Station responds with GetTariffsResponse	12 11 . The Charging Station responds with GetTariffsResponse The Test System sends a GetTariffsRequest with evseld <Configured evseld>

Tool validations
* Step 2: Message TransactionEventRequest - transactionInfo.tariffId must be <i>Test System1</i> * Step 5 4: Message GetTariffsResponse - status must be <i>Accepted</i> - tariffAssignments[0].tariffId must be <i>Test System1</i> - tariffAssignments[0].tariffKind must be <i>DriverTariff</i> - tariffAssignments[0].idTokens[0] must be <Configured valid_idtoken_idtoken> * Step 8 7: Message GetTariffsResponse - status must be <i>NoTariff</i> * Step 10 9: Message: TransactionEventRequest - triggerReason must be <i>EVConnectTimeout</i> - transactionInfo.stoppedReason must be <i>Timeout</i> * Step 13 12: Message GetTariffsResponse - status must be <i>NoTariff</i>

6.2.35. Page 436 - (2026-02) - TC_I_115_CSMS - Added main steps TransactionEvent TariffChanged

Test case name	Local Cost Calculation - Change transaction tariff - TariffConditionsSupported is true
Test case Id	TC_I_115_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
4 . The Test System sends a TransactionEventRequest with eventType <i>Updated</i> triggerReason <i>TariffChanged</i> transactionInfo.tariffId <tariff.tariffId from step 2>	5 . The CSMS responds with a TransactionEventResponse

6.2.36. Page 455 - (2026-02) - TC_I_121_CS - Incorrect minCost / maxCost validations

Test case name	Local Cost Calculation - Cost Details of Transaction - minCost/maxCost
Test case Id	TC_I_121_CS
Use case Id(s)	I12
Requirement(s)	I12.FR.17, I12.FR.38, I12.FR.39, I12.FR.39, I12.FR.40
System under test	Charging Station
Description	Charging Station calculates cost of the transaction locally and returns a break-down of the cost at end of the transaction for every charging period for which a different tariff element was active. CSMS can use this to generate invoices or Charge Detail Records for EMSPs.
Purpose	To verify if the Charging Station correctly reports the cost and usage details of a transaction.

Test case name	Local Cost Calculation - Cost Details of Transaction - minCost/maxCost
Prerequisite(s)	- The Charging Station supports local cost calculation using the Tariff mechanism (TariffCostCtrlr.Enabled[Tariff] exists and <i>ReadWrite</i> or is <i>true</i> and <i>ReadOnly</i>). - The Charging Station supports tariff conditions (TariffCostCtrlr.ConditionsSupported[Tariff] is <i>true</i>) - This test cases expects that reusable state <i>StopAuthorized</i> does not introduce an additional delay.

Before (Preparations)	
Configuration State: - TariffCostCtrlr.Enabled[Tariff] is <i>true</i> - TariffCostCtrlr.Enabled[RunningCost] is <i>true</i> - TariffCostCtrlr.Enabled[Cost] is <i>true</i> - TariffCostCtrlr.Interval[Cost] is 15 - SampledDataCtrlr.TxUpdatedInterval is 0 - AlignedDataCtrlr.Interval is 0	
Memory State: N/a	
Reusable State(s): N/a	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with SetDefaultTariffResponse	1. The Test System sends a SetDefaultTariffRequest with evseld 0 tariff.tariffId <i>Test System1</i> tariff.currency <i>EUR</i> tariff.minCost.exclTax <i>16.00</i> tariff.minCost.inclTax <i>19.20</i> tariff.minCost.taxRates[0].type <i>t10_0</i> tariff.minCost.taxRates[0].tax <i>10.0</i> tariff.maxCost.exclTax <i>35.00</i> tariff.maxCost.inclTax <i>42.00</i> tariff.minCost.taxRates[0].type <i>t10_0</i> tariff.minCost.taxRates[0].tax <i>10.0</i> tariff.chargingTime.prices[0].priceMinute <i>60.00</i> tariff.chargingTime.taxRates[0].type <i>t20_0</i> tariff.chargingTime.taxRates[0].tax <i>20.0</i>
3. Execute Reusable State <i>Authorized</i>	
4. Execute Reusable State <i>EnergyTransferStarted</i>	
Note: The Test System waits 5 seconds, calculated cost < minCost	
5. Execute Reusable State <i>StopAuthorized</i> Execute Reusable State <i>EVDIsconnected</i> Execute Reusable State <i>ParkingBayUnoccupied</i>	
6. The Charging Station sends a TransactionEventRequest	7. The Test System responds with a TransactionEventResponse
8. Execute Reusable State <i>Authorized</i>	
9. Execute Reusable State <i>EnergyTransferStarted</i>	
Note: The Test System waits 20 seconds, calculated cost between minCost and maxCost	
10. Execute Reusable State <i>StopAuthorized</i> Execute Reusable State <i>EVDIsconnected</i> Execute Reusable State <i>ParkingBayUnoccupied</i>	

Main (Test scenario)	
9. 11. The Charging Station sends a TransactionEventRequest	10. 12. The Test System responds with a TransactionEventResponse
13. Execute Reusable State <i>Authorized</i>	
11. 14. Execute Reusable State <i>EnergyTransferStarted</i>	
Note: <i>Reporting costDetails every 15 s, eventually cost > maxCost</i>	
12. 15. The Charging Station sends a TransactionEventRequest	13. 16. The Test System responds with a TransactionEventResponse
14. 17. The Charging Station sends a TransactionEventRequest	15. 18. The Test System responds with a TransactionEventResponse
16. 19. The Charging Station sends a TransactionEventRequest	17. 20. The Test System responds with a TransactionEventResponse
18. 21. The Charging Station sends a TransactionEventRequest	19. 22. The Test System responds with a TransactionEventResponse
20. 23. Execute Reusable State <i>StopAuthorized</i> Execute Reusable State <i>EVDDisconnected</i> Execute Reusable State <i>ParkingBayUnoccupied</i>	
21. 24. The Charging Station sends a TransactionEventRequest	22. 25. The Test System responds with a TransactionEventResponse

Tool validations
Note: <i>In the validation steps the value of TransactionEventRequest.timestamp of:</i> - last TransactionEventRequest.eventType = Started is called: TxStartDateTime, - the first TransactionEventRequest with chargingState = Charging is called: StartChargingTime, - the next TransactionEventRequest in which chargingState <> Charging is called: EndChargingTime. + * Step 5 Step 6: (MinCost) Message TransactionEventRequest with eventType Ended from reusable states leading up to <i>ParkingBayUnoccupied</i> must have: - costDetails.failureToCalculate must be <omitted> - costDetails.failureReason must be <omitted> - costDetails.totalCost.currency must be EUR - costDetails.totalCost.typeOfCost must be MinCost - costDetails.totalUsage.chargingTime must be <EndChargingTime> - <StartChargingTime> - costDetails.totalCost.chargingTime.exclTax must be 16.00 - costDetails.totalCost.chargingTime.inclTax must be 19.20 - costDetails.totalCost.chargingTime.taxRates[0].type t20_0 - costDetails.totalCost.chargingTime.taxRates[0].tax 20.0 - costDetails.totalCost.energy must be <omitted> - costDetails.totalCost.fixed must be <omitted> - costDetails.totalCost.idleTime must be <omitted> - costDetails.totalCost.reservation must be <omitted> - costDetails.totalCost.total.exclTax must be costDetails.totalCost.chargingTime.exclTax - costDetails.totalCost.total.inclTax must be costDetails.totalCost.chargingTime.inclTax - costDetails.chargingPeriods[0].tariffId must be Test System1 - costDetails.chargingPeriods[0].startPeriod must be <TxStartDateTime> - costDetails.chargingPeriods[0].dimensions[0].type must be ChargingTime - costDetails.chargingPeriods[0].dimensions[0].volume must be costDetails.totalUsage.chargingTime

Tool validations

* ~~Step 9~~ Step 11: (NormalCost)

~~Note: The value of TransactionEventRequest.timestamp of last TransactionEventRequest.eventType is Started is called TxStartDateTime.~~

Message **TransactionEventRequest** with **eventType** *Ended* from reusable states leading up to **ParkingBayUnoccupied** must have:

- **costDetails.failureToCalculate** must be <omitted>
- **costDetails.failureReason** must be <omitted>
- **costDetails.totalCost.currency** must be *EUR*
- **costDetails.totalCost.typeOfCost** must be *NormalCost*
- **costDetails.totalUsage.chargingTime** must be <EndChargingTime> - <StartChargingTime>
- **costDetails.totalCost.chargingTime.exclTax** must be **costDetails.totalUsage.chargingTime**
- **costDetails.totalCost.chargingTime.inclTax** must be **costDetails.totalCost.chargingTime.exclTax** * 1.20
- **costDetails.totalCost.chargingTime.taxRates[0].type** *t20_0*
- **costDetails.totalCost.chargingTime.taxRates[0].tax** *20.0*
- ~~**costDetails.totalUsage.chargingTime** must be 30~~
- **costDetails.totalCost.energy** must be <omitted>
- **costDetails.totalCost.fixed** must be <omitted>
- **costDetails.totalCost.idleTime** must be <omitted>
- **costDetails.totalCost.reservation** must be <omitted>
- **costDetails.totalCost.total.exclTax** must be **costDetails.totalCost.chargingTime.exclTax**
- **costDetails.totalCost.total.inclTax** must be **costDetails.totalCost.chargingTime.inclTax**
- **costDetails.chargingPeriods[0].tariffId** must be *Test System1*
- **costDetails.chargingPeriods[0].startPeriod** must be <TxStartDateTime>
- **costDetails.chargingPeriods[0].dimensions[0].type** must be *ChargingTime*
- **costDetails.chargingPeriods[0].dimensions[0].volume** must be **costDetails.totalUsage.chargingTime**

Tool validations

* ~~Step 12~~ Step 15: (NormalCost)

Note: ~~The value of TransactionEventRequest.timestamp of last TransactionEventRequest.eventType is Started is called TxStartDateTime.~~

Message TransactionEventRequest with ~~eventType Ended~~ from reusable states leading up to ~~ParkingBayUnoccupied~~ must have:

- ~~eventType Updated~~
- ~~triggerReason must be RunningCost~~
- costDetails.failureToCalculate must be <omitted>
- costDetails.failureReason must be <omitted>
- costDetails.totalCost.currency must be EUR
- costDetails.totalCost.typeOfCost must be NormalCost
- costDetails.totalCost.chargingTime.exclTax must be 15.00
- costDetails.totalCost.chargingTime.inclTax must be 18.00
- costDetails.totalCost.chargingTime.taxRates[0].type t20_0
- costDetails.totalCost.chargingTime.taxRates[0].tax 20.0
- costDetails.totalUsage.chargingTime must be 15
- ~~costDetails.chargingPeriods[0].tariffId must be Test System1~~
- ~~costDetails.chargingPeriods[0].startPeriod must be <txStartDateTime>~~
- ~~costDetails.chargingPeriods[0].dimensions[0].type must be ChargingTime~~
- ~~costDetails.chargingPeriods[0].dimensions[0].volume must be 15~~
- costDetails.chargingPeriods must be <omitted>

* ~~Step 14~~ Step 17: (NormalCost)

Note: ~~The value of TransactionEventRequest.timestamp of last TransactionEventRequest.eventType is Started is called TxStartDateTime.~~

Message TransactionEventRequest with ~~eventType Ended~~ from reusable states leading up to ~~ParkingBayUnoccupied~~ must have:

- ~~eventType Updated~~
- ~~triggerReason must be RunningCost~~
- costDetails.failureToCalculate must be <omitted>
- costDetails.failureReason must be <omitted>
- costDetails.totalCost.currency must be EUR
- costDetails.totalCost.typeOfCost must be NormalCost
- costDetails.totalCost.chargingTime.exclTax must be 30.00
- costDetails.totalCost.chargingTime.inclTax must be 36.00
- costDetails.totalCost.chargingTime.taxRates[0].type t20_0
- costDetails.totalCost.chargingTime.taxRates[0].tax 20.0
- costDetails.totalUsage.chargingTime must be 30
- ~~costDetails.chargingPeriods[0].tariffId must be Test System1~~
- ~~costDetails.chargingPeriods[0].startPeriod must be <txStartDateTime>~~
- ~~costDetails.chargingPeriods[0].dimensions[0].type must be ChargingTime~~
- ~~costDetails.chargingPeriods[0].dimensions[0].volume must be 30~~
- costDetails.chargingPeriods must be <omitted>

Tool validations

* ~~Step 16~~ Step 19: (MaxCost)

Note: ~~The value of TransactionEventRequest.timestamp of last TransactionEventRequest.eventType is Started is called TxStartDateTime.~~

Message TransactionEventRequest

- eventType must be Updated
- triggerReason must be RunningCost
- costDetails.failureToCalculate must be <omitted>
- costDetails.failureReason must be <omitted>
- costDetails.totalCost.currency must be EUR
- costDetails.totalCost.typeOfCost must be MaxCost
- costDetails.totalCost.chargingTime.exclTax must be 35.00
- costDetails.totalCost.chargingTime.inclTax must be 42.00
- costDetails.totalCost.chargingTime.taxRates[0].type t20_0
- costDetails.totalCost.chargingTime.taxRates[0].tax 20.0
- costDetails.totalUsage.chargingTime must be 45
- ~~- costDetails.totalCost.energy must be <omitted>~~
- ~~- costDetails.totalCost.chargingTime must be <omitted>~~
- ~~- costDetails.totalCost.idleTime must be <omitted>~~
- ~~- costDetails.totalCost.reservation must be <omitted>~~
- ~~- costDetails.totalCost.total.exclTax must be 35.00~~
- ~~- costDetails.totalCost.total.inclTax must be 38.50~~
- costDetails.chargingPeriods must be <omitted> ~~- costDetails.chargingPeriods[0].tariffId must be Test System1~~
- ~~- costDetails.chargingPeriods[0].startPeriod must be <txStartDateTime>~~
- ~~- costDetails.chargingPeriods[0].dimensions[0].type must be ChargingTime~~
- ~~- costDetails.chargingPeriods[0].dimensions[0].volume must be 45~~

* ~~Step 18~~ Step 21: (MaxCost)

Message TransactionEventRequest

- eventType must be Updated
- triggerReason must be RunningCost
- costDetails.failureToCalculate must be <omitted>
- costDetails.failureReason must be <omitted>
- costDetails.totalCost.currency must be EUR
- costDetails.totalCost.typeOfCost must be MaxCost
- costDetails.totalCost.chargingTime.exclTax must be 35.00
- costDetails.totalCost.chargingTime.inclTax must be 42.00
- costDetails.totalCost.chargingTime.taxRates[0].type t20_0
- costDetails.totalCost.chargingTime.taxRates[0].tax 20.0
- costDetails.totalUsage.chargingTime must be 60
- costDetails.chargingPeriods must be <omitted>
- ~~- costDetails.totalCost.energy must be <omitted>~~
- ~~- costDetails.totalCost.chargingTime must be <omitted>~~
- ~~- costDetails.totalCost.idleTime must be <omitted>~~
- ~~- costDetails.totalCost.reservation must be <omitted>~~
- ~~- costDetails.totalCost.total.exclTax must be 35.00~~
- ~~- costDetails.totalCost.total.inclTax must be 38.50~~
- ~~- costDetails.chargingPeriods[0].tariffId must be Test System1~~
- ~~- costDetails.chargingPeriods[0].startPeriod must be <txStartDateTime>~~
- ~~- costDetails.chargingPeriods[0].dimensions[0].type must be ChargingTime~~
- ~~- costDetails.chargingPeriods[0].dimensions[0].volume must be 60~~

Tool validations

* Step 21 Step 24: (MaxCost)

Note: The value of **TransactionEventRequest.timestamp** of last **TransactionEventRequest.eventType** is Started is called **TxStartDateTime**.

Message **TransactionEventRequest** with **eventType** *Ended* from reusable states leading up to **ParkingBayUnoccupied** must have:

- **costDetails.failureToCalculate** must be <omitted>
- **costDetails.failureReason** must be <omitted>
- **costDetails.totalCost.currency** must be *EUR*
- **costDetails.totalCost.typeOfCost** must be *MaxCost*
- **costDetails.totalUsage.chargingTime** must be <EndChargingTime> - <StartChargingTime>
- **costDetails.totalCost.chargingTime.exclTax** must be *35.00*
- **costDetails.totalCost.chargingTime.inclTax** must be *42.00*
- **costDetails.totalCost.chargingTime.taxRates[0].type** *t20_0*
- **costDetails.totalCost.chargingTime.taxRates[0].tax** *20.0*
- **costDetails.totalCost.energy** must be <omitted>
- **costDetails.totalCost.fixed** must be <omitted>
- **costDetails.totalCost.idleTime** must be <omitted>
- **costDetails.totalCost.reservation** must be <omitted>
- **costDetails.totalCost.total.exclTax** must be **costDetails.totalCost.chargingTime.exclTax**
- **costDetails.totalCost.total.inclTax** must be **costDetails.totalCost.chargingTime.inclTax**
- **costDetails.chargingPeriods[0].tariffId** must be *Test System1*
- **costDetails.chargingPeriods[0].startPeriod** must be <TxStartDateTime>
- **costDetails.chargingPeriods[0].dimensions[0].type** must be *ChargingTime*
- **costDetails.chargingPeriods[0].dimensions[0].volume** must be **costDetails.totalUsage.chargingTime**

Post scenario validations:

N/a

6.2.37. Page 484 - 840 - (2026-06) - Renamed Configuration variable

The configuration variable <Configured max time deviation> was renamed to <Configured max clock deviation>.

The following test cases were updated:

TC_K_09_CS, TC_K_21_CS, TC_K_22_CS, TC_K_23_CS, TC_K_28_CS, TC_K_39_CS, TC_K_40_CS, TC_K_41_CS, TC_K_56_CS, TC_Q_110_CS, TC_Q_117_CS

6.2.38. Page 525 - (2026-02) - TC_N_52_CS - NotifyEvent steps added

Test case name	Set Variable Monitoring - Removing a VariableMonitor
Test case Id	TC_N_52_CS
...	

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>EnergyTransferStarted</i> or manually trigger the monitor. <i>Notes: If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentVariable is chosen a manual action is needed to trigger the monitor.</i>	
2. Charging Station sends a NotifyEventRequest	3. Test System responds with a NotifyEventResponse
3. 5. The Charging Station responds with a ClearVariableMonitoringResponse	2. 4. The Test System sends a ClearVariableMonitoringRequest with id <monitoringId of monitor set in Memory State>

Main (Test scenario)	
5. 7. The Charging Station responds with a GetMonitoringReportResponse	4. 6. The Test System sends a GetMonitoringReportRequest with componentVariable.component <Configured threshold monitor component variable> componentVariable.variable <Configured threshold monitor component variable> monitoringCriteria ThresholdMonitoring
8. Execute Reusable State StopAuthorized or manually trigger the monitor. Notes: If componentVariable is set to "Power" or "Current" EnergyTransferStarted will trigger the monitor. If another componentvariable is chosen a manual action is needed to trigger the monitor.	
Note: The Charging Station should not send a request NotifyEventRequest for the cleared monitor	

Tool validations
* Step 1: Step 2: Message NotifyEventRequest - eventData[0].trigger Alerting - eventData[0].component <Configured threshold monitor component variable> - eventData[0].variable <Configured threshold monitor component variable> - eventData[0].variableMonitoringId <monitoringId of monitor set in Memory State>
* Step 3: Step 5: Message ClearVariableMonitoringResponse - clearMonitoringResult[0].status Accepted AND - clearMonitoringResult[0].id <monitoringId of monitor set in Memory State>
* Step 5: Step 7: Message GetMonitoringReportResponse - getMonitoringResult[0].status EmptyResultSet

6.2.39. Page 658 - (2026-02) - TC_M_24_CS - GetCertificateStatusRequest is not intended to be sent for the V2G leaf certificate

Updated note

	Text
Old	Note(s): Step 1/2 are repeated for the V2G Charging Station (leaf), the subCA1 and subCA2 certificates.
New	Note(s): - Step 1/2 are executed for both subCA1 and subCA2. - Step 1/2 are not intended to be executed for the V2G Charging Station (leaf), but it is not disallowed.

6.2.40. Page 783 - (2026-02) - TC_O_24_CS - Additions to tool validations

Test case name	Set Display Message - Second Alwaysfront priority
Test case Id	TC_O_24_CS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetDisplayMessageResponse	1. The Test System sends a SetDisplayMessageRequest with message.id <Generated displayMessageId> message.priority AlwaysFront
4. The Charging Station responds with a SetDisplayMessageResponse	3. The Test System sends a SetDisplayMessageRequest with message.id <Configured displayMessage2Id> message.priority AlwaysFront

Main (Test scenario)	
6. The Charging Station responds with a GetDisplayMessagesResponse	5. The Test System sends a GetDisplayMessagesRequest with id <Configured displayMessageId>
6. 8. The Charging Station responds with a GetDisplayMessagesResponse	5. 7. The Test System sends a GetDisplayMessagesRequest with id <Configured displayMessage2Id>
7. 9. The Charging Station sends a NotifyDisplayMessagesRequest	8. 10. The Test System responds with a NotifyDisplayMessagesResponse .
Note(s): - If tbc is True at Step 7 9 then step 7 9 and 8 10 will be repeated - The message from step 1 is NOT displayed anymore and is replaced by the message from step 5 7 .	

Tool validations
* Step 2: Message SetDisplayMessageResponse - status Accepted * Step 4: Message SetDisplayMessageResponse - status Accepted * Step 6: Message GetDisplayMessagesResponse - status Unknown * Step 6: Step 8: Message GetDisplayMessagesResponse - status Accepted * Step 7: Step 9: Message NotifyDisplayMessagesRequest - requestId <Generated requestId>
Post scenario validations: - N/a

6.2.41. Page 567/568/569 - (2026-02) - TC_K_116_CS - Updated limits to more practical ones and aligned chargingProfileIds

Aligned the chargingProfileIds, so they are replacing each other.

Test case name	Renegotiating a Charging Schedule ISO 15118-20 - Adjusting charging schedule when energy needs change
Test case Id	TC_K_116_CS
...	

Main (Test scenario)	
Charging Station	CSMS
Manual Action: Use EV to trigger renegotiation requesting: departureTime <Current DateTime + 2 hours> evTargetEnergyRequest* 5000 6000	
1. The Charging Station sends a NotifyEVChargingNeedsRequest .	2. The Test System responds with a NotifyEVChargingNeedsResponse . With status Accepted

Main (Test scenario)	
4. The Charging Station responds with a SetChargingProfileResponse	3. The Test System sends a SetChargingProfileRequest with chargingProfile.chargingProfilePurpose TxProfile chargingProfile.transactionId <Provided transactionId from before> chargingProfile.chargingSchedule[0].id 10 chargingProfile.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit 2.5 6 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>
...	
10. The Charging Station responds with a SetChargingProfileResponse	9. The Test System sends a SetChargingProfileRequest with chargingProfile.id <Same id as step 3> chargingProfile.chargingProfilePurpose TxProfile chargingProfile.transactionId <Provided transactionId from before> chargingProfile.chargingSchedule[0].id 10 chargingProfile.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit 5 10 * <limit multiplier> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>
...	
16. The Charging Station responds with a SetChargingProfileResponse	15. The Test System sends a SetChargingProfileRequest with chargingProfile.id <Same id as step 3> chargingProfile.chargingProfilePurpose TxProfile chargingProfile.transactionId <Provided transactionId from before> chargingProfile.chargingSchedule[0].id 10 chargingProfile.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit 2 10 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>

Tool validations
* Step 1: Message: NotifyEVChargingNeedsRequest - evseld must be <Configured evseld> - chargingNeeds.acChargingParameters must be <omitted> - chargingNeeds.dcChargingParameters must be <omitted> - chargingNeeds.v2xChargingParameters.evTargetEnergyRequest must be 5000 6000 - chargingNeeds.v2xChargingParameters.evTargetEnergyRequest must be - chargingNeeds.controlMode must be ScheduledControl - chargingNeeds.departureTime must be <Current DateTime + 2 hours> ...

6.2.42. Page 587 - (2026-02) - TC_K_125_CS Updated dynUpdateTime tool validations

Test case name	Dynamic charging profiles by external system - Dynamic charging profile configured
Test case Id	TC_K_125_CS
...	

Tool validations
...
* Step 7:
Message ReportChargingProfilesRequest
- requestId must be <i>Same Id as in the GetChargingProfilesRequest in step 5</i>
- EVSEId must be <i>0</i>
There must be one ChargingProfile be reported with
- chargingProfile[0].id must be <i><Configured chargingProfileId></i>
- chargingProfile[0].stackLevel must be <i>0</i>
- chargingProfile[0].chargingProfilePurpose must be <i>ChargingStationMaxProfile</i>
- chargingProfile[0].chargingProfileKind must be <i>Dynamic</i>
- chargingProfile[0].dynUpdateTime must be <i><Datetime of moment of step 3 between sending manual action for limit 10 and current dateTime></i>
- chargingProfile[0].chargingSchedule size must be <i>1</i>
- chargingProfile[0].chargingSchedule[0].chargingSchedulePeriod size must be <i>1</i>
- chargingProfile[0].chargingSchedule[0].chargingSchedulePeriod[0].startPeriod must be <i>0</i>
- chargingProfile[0].chargingSchedule[0].chargingSchedulePeriod[0].operationMode <i>ExternalSetpoint</i>
- chargingProfile[0].chargingSchedule[0].chargingSchedulePeriod[0].limit must be <i><omitted></i>
- chargingProfile[0].chargingSchedule[0].chargingSchedulePeriod[0].setpoint must be <i>10 * <limit multiplier></i>
There must be one ChargingProfile be reported with
- chargingProfile[1].id must be <i><Configured chargingProfileId2></i>
- chargingProfile[1].stackLevel must be <i>1</i>
- chargingProfile[1].chargingProfilePurpose must be <i>ChargingStationMaxProfile</i>
- chargingProfile[1].chargingProfileKind must be <i>Dynamic</i>
- chargingProfile[1].dynUpdateTime must be <i><Datetime of moment of step 3 between sending manual action for limit 8 and current dateTime></i>
...

6.2.43. Page 588 - (2026-02) - TC_K_129_CS to TC_K_135_CS improvements

The "Absolute" charging profiles that were used have been replaced by "Relative" charging profiles for easier testing.

6.2.44. Page 596 - (2026-02) - TC_K_136_CS - Fixed incorrect timezone usage

TC_K_136_CS was using the value "UTC+02:00" for variable TimeZone, but that is incorrect. For a time offset of 2 hours the variable TimeOffset must be used with a value of "+02:00".

6.2.45. Page 590 - (2026-02) - TC_K_130_CS to TC_K_135_CS - Reusable state Booted removed

Reusable state [booted](#) removed from preparations.

6.2.46. Page 596 - (2026-02) - TC_K_136_CS - Remove timezone configuration

Timezone configuration was removed from preparations configuration state.

Before (Preparations)
Configuration State: ClockCtrlr.TimeZone is "" (if present) N/a

6.2.47. Page 664/665 - (2026-02) - TC_M_100_CS - Reduced amount of certificates sent

Test case name	Certificate Installation EV - ISO 15118-20 - Success
Test case Id	TC_M_100_CS
...	

Before (Preparations)
Configuration State: -The test case calls <i>SendISO15118AuthorizationMethod</i> method with parameter <i>PnC</i> in order to inform the EV emulator about the expected authorization method. -The test case calls <i>SendInstallISO15118CertificateMethod</i> method in order to trigger the EV emulator to initiate installing at most 5 EV certificates.

Main (Test scenario)	
Charging Station	CSMS
Note: EV emulator accepts at most 5 contracts.	
1. The Charging Station sends a Get15118EVCertificateRequest	2. The Test System responds with a Get15118EVCertificateResponse with status <i>Accepted</i> exiResponse <i><Raw CertificateInstallationRes response for the EV, Base64 encoded.></i> remainingContracts <i><Configured</i> <i>numberOfSupportedEvCertificates> - 1</i> 1
3. The Charging Station sends a Get15118EVCertificateRequest	4. The Test System responds with a Get15118EVCertificateResponse with status <i>Accepted</i> exiResponse <i><Raw CertificateInstallationRes response for the EV, Base64 encoded.></i> remainingContracts <i><Configured</i> <i>numberOfSupportedEvCertificates> - 2</i> 0
5. The Charging Station sends a Get15118EVCertificateRequest SHALL NOT send another	6. The Test System responds with a Get15118EVCertificateResponse with status <i>Accepted</i> exiResponse <i><Raw CertificateInstallationRes response for the EV, Base64 encoded.></i> remainingContracts <i><Configured</i> <i>numberOfSupportedEvCertificates> - 3</i>
6. Execute Reusable State <i>Authorized15118</i>	
7. Execute Reusable State <i>EnergyTransferStarted</i>	

Tool validations
* Step 1: Message: Get15118EVCertificateRequest - action must be <i>Install</i> - maximumContractCertificateChains must be 5 NOT be omitted * Step 3: Message: Get15118EVCertificateRequest - action must be <i>Install</i> - maximumContractCertificateChains must be 5 NOT be omitted * Step 5: Message: Get15118EVCertificateRequest - action must be <i>Update</i> - maximumContractCertificateChains must be 5

6.2.48. Page 666 - (2026-02) - TC_M_101_CS - Removed AdditionalRootCertificateCheck note

Test case name	Install CA certificate - OEMRootCertificate
Test case Id	TC_M_101_CS
...	

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State <i>CertificateInstalled</i> for certificateType <i>OEMRootCertificate</i> (Root 2) Note(s): When the Charging Station has the following configuration; AdditionalRootCertificateCheck implemented with value true, then a custom OEMRootCertificate should be used.	
2. Execute Reusable State <i>GetInstalledCertificates</i> for certificateType <i>OEMRootCertificate</i>	

749

6.2.49. Page 749-752 - (2026-02) - TC_N_105_CS - Use existing monitoring configurations and check the disabled monitoring events are not sent

Test case name	Set Variable Monitoring - Set Frequent Periodic Variable Monitoring - Periodic
Test case Id	TC_N_105_CS
...	

Before (Preparations)
Configuration State: N/a
Memory State: This test requires the Monitoring Base to be set to <i>All</i> . - SetMonitoringBaseRequest with monitoringBase = <i>All</i> . This test requires the existence of a <i>Periodic</i> monitor on a (numerical) variable. It is not mandated which variables are required to be monitored. This test uses the variable "Power" of component "EVSE", which is most likely to be available for monitoring.
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. Charging Station responds with a SetVariableMonitoringResponse	1. Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].value 1 <Configured Clock Aligned MeterValues Interval> setMonitoringData[0].type Periodic setMonitoringData[0].severity <Configured severity> setMonitoringData[0].component.name <Configured threshold monitor component> setMonitoringData[0].component.evse.id <Configured evseId> setMonitoringData[0].variable.name <Configured threshold monitor component variable> setMonitoringData[0].periodicEventStream.interval 10 <Configured Clock Aligned MeterValues Interval> * 2 setMonitoringData[0].periodicEventStream.values 30 <Configured Clock Aligned MeterValues Interval> * 4
3. Charging Station sends a OpenPeriodicEventStreamRequest	4. Test System responds with a OpenPeriodicEventStreamResponse with status Accepted
6. Charging Station response with a GetPeriodicEventStreamResponse	5. Test System sends a GetPeriodicEventStreamRequest
Note: Test System waits for 20 seconds	
Note : Waits <Configured Clock Aligned MeterValues Interval> * 2 + 1 seconds.	
7. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND AND SHALL NOT send a NotifyEventRequest with - eventData[0].trigger must be Periodic - eventData[0].variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - eventData[0].component.name must be <Configured threshold monitor component> - eventData[0].variable.name must be <Configured threshold monitor component variable> +	After 10 seconds
Note : Waits <Configured Clock Aligned MeterValues Interval> * 2 + 1 seconds.	
8 Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND AND SHALL NOT send a NotifyEventRequest with - eventData[0].trigger must be Periodic - eventData[0].variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - eventData[0].component.name must be <Configured threshold monitor component> - eventData[0].variable.name must be <Configured threshold monitor component variable> +	After another 10 seconds

Main (Test scenario)	
10. Charging Station responds with a SetVariableMonitoringResponse	9. Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].id <code><SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2></code> setMonitoringData[0].value 4 <code><Configured Clock Aligned MeterValues Interval></code> setMonitoringData[0].type <code>Periodic</code> setMonitoringData[0].severity <code><Configured severity></code> setMonitoringData[0].component.name <code><Configured threshold monitor component></code> setMonitoringData[0].component.evse.id <code><Configured evseld></code> setMonitoringData[0].variable.name <code><Configured threshold monitor component variable></code> setMonitoringData[0].periodicEventStream <code><omitted></code>
11 Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	This step may be omitted if buffer already empty This step is optional. It is not sent if the buffer is already empty.
12. Charging Station sends a ClosePeriodicEventStreamRequest	13. Test System responds with a ClosePeriodicEventStreamResponse
14. Charging Station sends a NotifyEventRequest AND SHALL NOT send a NotifyPeriodicEventStream with - id set to <code><OpenPeriodicEventStreamRequest.constantStreamData.id of step 3></code>	15. Test System responds with a NotifyEventResponse
Note: Step 14 and 15 will repeat every second	
Note: The Test System waits 20 seconds, before ending the test case.	
Note: The Test System waits <code><Configured Clock Aligned MeterValues Interval> * 2 + 1</code> seconds, before ending the test case.	

Tool validations

* Step 2:

Message: **SetVariableMonitoringResponse**

- **setMonitoringResult[0].id** must be *<not omitted>*
- **setMonitoringResult[0].status** must be *Accepted*
- **setMonitoringResult[0].type** must be *Periodic*
- **setMonitoringResult[0].severity** must be *<Configured severity>*
- **setMonitoringResult[0].component.name** must be *<Configured threshold monitor component>*
- **setMonitoringResult[0].component.evse.id** must be *<Configured evseld>*
- **setMonitoringResult[0].variable.name** must be *<Configured threshold monitor component variable>*

* Step 3:

Message: **OpenPeriodicEventStreamRequest**

- **constantStreamData.id** must be *<not omitted>*
- **constantStreamData.variableMonitoringId** must be *<SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2>*
- **constantStreamData.params.interval** must be ~~40~~ *<Configured Clock Aligned MeterValues Interval> * 2*
- **constantStreamData.params.values** must be ~~30~~ *<Configured Clock Aligned MeterValues Interval> * 4*

* Step 6:

Message: **GetPeriodicEventStreamResponse**

contains at least

- **constantStreamData[0].id** must be *<not omitted>*
- **constantStreamData[0].variableMonitoringId** must be *<SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2>*
- **constantStreamData[0].params.interval** must be ~~40~~ *<Configured Clock Aligned MeterValues Interval> * 2*
- **constantStreamData[0].params.values** must be ~~30~~ *<Configured Clock Aligned MeterValues Interval> * 4*

* Step 7:

Message: **NotifyPeriodicEventStream**

- **id** must be *<OpenPeriodicEventStreamRequest.constantStreamData.id of step 3>*
- **data** array must contain ~~9 or 10~~ *1 or 2* elements

* Step 8:

Message: **NotifyPeriodicEventStream**

- **id** must be *<OpenPeriodicEventStreamRequest.constantStreamData.id of step 3>*
- **data** array must contain ~~9 or 10~~ *1 or 2* elements

Tool validations
* Step 10: Message: SetVariableMonitoringResponse - setMonitoringResult[0].id must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - setMonitoringResult[0].status must be <i>Accepted</i> - setMonitoringResult[0].type must be <i>Periodic</i> - setMonitoringResult[0].severity must be <Configured severity> - setMonitoringResult[0].component.name must be <Configured threshold monitor component> - setMonitoringResult[0].component.evse.id must be <Configured evseld> - setMonitoringResult[0].variable.name must be <Configured threshold monitor component variable> * Step 11: Message: NotifyPeriodicEventStream - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 12: Message: ClosePeriodicEventStreamRequest - id <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 14: Message: NotifyEventRequest - generatedAt must be <not omitted> - eventData[0].trigger must be <i>Periodic</i> - eventData[0].actualValue must be <not omitted> - eventData[0].variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - eventData[0].severity must be <Configured severity> - eventData[0].component.name must be <Configured threshold monitor component> - eventData[0].component.evse.id must be <Configured evseld> - eventData[0].variable.name must be <Configured threshold monitor component variable> Post scenario validations: * After step 12 no message of type NotifyPeriodicEventStream may be sent with - id set to <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> N/a

6.2.50. Page 753/754 - (2026-02) - TC_N_106_CS - Use existing monitoring configurations and check the disabled monitoring events are not sent

Test case name	Set Variable Monitoring - Set Frequent Periodic Variable Monitoring - CSMS rejects stream
Test case Id	TC_N_106_CS
...	

Before (Preparations)
Configuration State: N/a Memory State: This test requires the Monitoring Base to be set to All. - SetMonitoringBaseRequest with monitoringBase = <i>All</i>. Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. Charging Station responds with a SetVariableMonitoringResponse	1. Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].value <Configured Clock Aligned MeterValues Interval> setMonitoringData[0].type Periodic setMonitoringData[0].severity <Configured severity> setMonitoringData[0].component.name <Configured threshold monitor component variable> setMonitoringData[0].component.evse.id <Configured evseld> setMonitoringData[0].variable.name <Configured threshold monitor component variable> setMonitoringData[0].periodicEventStream.interval 10 <Configured Clock Aligned MeterValues Interval> * 2 setMonitoringData[0].periodicEventStream.values 30 <Configured Clock Aligned MeterValues Interval> * 4
3. Charging Station sends a OpenPeriodicEventStreamRequest	4. Test System responds with a OpenPeriodicEventStreamResponse with status Rejected
5. Charging Station sends a NotifyEventRequest AND SHALL NOT send a NotifyPeriodicEventStream with - id set to <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2>	6. Test System responds with a NotifyEventResponse
Note: Step 5 and 6 will repeat every second	
Note: The Test System waits 11 seconds, before ending the testcase.	
Note: The Test System waits <Configured Clock Aligned MeterValues Interval> * 2 + 1 seconds, before ending the testcase.	

Tool validations
* Step 2: Message: SetVariableMonitoringResponse - setMonitoringResult[0].id must be <not omitted> - setMonitoringResult[0].status must be <i>Accepted</i> - setMonitoringResult[0].type must be <i>Periodic</i> - setMonitoringResult[0].severity must be <Configured severity> - setMonitoringResult[0].component.name must be <Configured threshold monitor component variable> - setMonitoringResult[0].component.evse.id must be <Configured evseld> - setMonitoringResult[0].variable.name must be <Configured threshold monitor component variable> No message of type NotifyPeriodicEventStream shall be sent with - id set to <SetVariableMonitoringResponse.setMonitoringResult[0].id> * Step 3: Message: OpenPeriodicEventStreamRequest - constantStreamData.id must be <not omitted> - constantStreamData.variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - constantStreamData.params.interval must be 10 <Configured Clock Aligned MeterValues Interval> * 2 - constantStreamData.params.values must be 30 <Configured Clock Aligned MeterValues Interval> * 4 * Step 5: Message: NotifyEventRequest - generatedAt must be <not omitted> - eventData[0].trigger must be <i>Periodic</i> - eventData[0].actualValue must be <not omitted> - eventData[0].variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - eventData[0].severity must be <Configured severity> - eventData[0].component.name must be <Configured threshold monitor component variable> - eventData[0].component.evse.id must be <Configured evseld> - eventData[0].variable.name must be <Configured threshold monitor component variable> Post scenario validations: No message of type NotifyPeriodicEventStream shall be sent with - id set to <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> N/a

6.2.51. Page 755/756 - (2026-02) - TC_N_108_CS - Use existing monitoring configurations and check the disabled monitoring events are not sent

Test case name	Set Variable Monitoring - Close Periodic Event Streams
Test case Id	TC_N_108_CS
...	

Before (Preparations)
Configuration State: N/a Memory State: This test requires the Monitoring Base to be set to All. - SetMonitoringBaseRequest with monitoringBase = All. This test requires the existence of a <i>Periodic</i> monitor on a (numerical) variable. It is not mandated which variables are required to be monitored. This test uses the variable "Power" of component "EVSE", which is most likely to be available for monitoring. Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
2. Charging Station responds with a SetVariableMonitoringResponse	1. Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].value 1 <Configured Clock Aligned MeterValues Interval> setMonitoringData[0].type Periodic setMonitoringData[0].severity <Configured severity> setMonitoringData[0].component.name <Configured threshold monitor component> setMonitoringData[0].component.evse.id <Configured evseld> setMonitoringData[0].variable.name <Configured threshold monitor component variable> setMonitoringData[0].periodicEventStream.interval 10 <Configured Clock Aligned MeterValues Interval> * 2 setMonitoringData[0].periodicEventStream.values 30 <Configured Clock Aligned MeterValues Interval> * 4
3. Charging Station sends a OpenPeriodicEventStreamRequest	4. Test System responds with a OpenPeriodicEventStreamResponse with status Accepted
6. Charging Station response with a ClearVariableMonitoringResponse	5. Test System sends a ClearVariableMonitoringRequest with id <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2>
7. Charging Station sends a ClosePeriodicEventStreamRequest	8. Test System responds with a ClosePeriodicEventStreamResponse
Note: The Test System waits 20 seconds, before ending the testcase.	
9. Charging Station SHALL NOT send a NotifyPeriodicEventStream with	
- id set to <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3>	
Note: The Test System waits <Configured Clock Aligned MeterValues Interval> * 2 + 1 seconds, before ending the testcase.	

Tool validations
* Step 2: Message: SetVariableMonitoringResponse - setMonitoringResult[0].id must be <not omitted> - setMonitoringResult[0].status must be Accepted - setMonitoringResult[0].type must be Periodic - setMonitoringResult[0].severity must be <Configured severity> - setMonitoringResult[0].component.name must be <Configured threshold monitor component> - setMonitoringResult[0].component.evse.id must be <Configured evseld> - setMonitoringResult[0].variable.name must be <Configured threshold monitor component variable> * Step 3: Message: OpenPeriodicEventStreamRequest - constantStreamData.id must be <not omitted> - constantStreamData.variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - constantStreamData.params.interval must be 10 <Configured Clock Aligned MeterValues Interval> * 2 - constantStreamData.params.values must be 30 <Configured Clock Aligned MeterValues Interval> * 4 * Step 6: Message: ClearVariableMonitoringResponse - clearMonitoringResult[0].status must be Accepted - clearMonitoringResult[0].id must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> * Step 7: Message: ClosePeriodicEventStreamRequest - id <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3>

Tool validations
Post scenario validations: * After step 9 no message of type NotifyPeriodicEventStream may be sent with id set to <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> N/a

6.2.52. Page 757/758 - (2026-02) - TC_N_109_CS - Use existing monitoring configurations

Test case name	Set Variable Monitoring - Adjust Periodic Event Streams
Test case Id	TC_N_109_CS
...	

Before (Preparations)
Configuration State: N/a
Memory State: This test requires the Monitoring Base to be set to <i>All</i> . - SetMonitoringBaseRequest with monitoringBase = <i>All</i> . This test requires the existence of a <i>Periodic</i> monitor on a (numerical) variable. It is not mandated which variables are required to be monitored. This test uses the variable "Power" of component "EVSE", which is most likely to be available for monitoring.
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
2. Charging Station responds with a SetVariableMonitoringResponse	1. Test System sends a SetVariableMonitoringRequest with setMonitoringData[0].value $\neq$ <Configured Clock Aligned MeterValues Interval> setMonitoringData[0].type <i>Periodic</i> setMonitoringData[0].severity <i><Configured severity></i> setMonitoringData[0].component.name <i><Configured threshold monitor component></i> setMonitoringData[0].component.evse.id <i><Configured evseld></i> setMonitoringData[0].variable.name <i><Configured threshold monitor component variable></i> setMonitoringData[0].periodicEventStream.interval $\neq$ <Configured Clock Aligned MeterValues Interval> * 2
3. Charging Station sends a OpenPeriodicEventStreamRequest	4. Test System responds with a OpenPeriodicEventStreamResponse with status <i>Accepted</i>
5. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	<i>After 10 seconds</i>
6. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	<i>After 10 seconds</i>
7. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	<i>After 10 seconds</i>

Main (Test scenario)	
9. Charging Station responds with a AdjustPeriodicEventStreamResponse	8. Test System sends a AdjustPeriodicEventStreamRequest with id <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> params.interval 15 <Configured Clock Aligned MeterValues Interval> * 3
10. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	After 15 seconds
11. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	After 15 seconds
12. Charging Station sends a NotifyPeriodicEventStream using RFC framework SEND	After 15 seconds

Tool validations
* Step 2: Message: SetVariableMonitoringResponse - setMonitoringResult[0].id must be <not omitted> - setMonitoringResult[0].status must be Accepted - setMonitoringResult[0].type must be Periodic - setMonitoringResult[0].severity must be <Configured severity> - setMonitoringResult[0].component.name must be <Configured threshold monitor component> - setMonitoringResult[0].component.evse.id must be <Configured evseld> - setMonitoringResult[0].variable.name must be <Configured threshold monitor component variable> * Step 3: Message: OpenPeriodicEventStreamRequest - constantStreamData.id must be <not omitted> - constantStreamData.variableMonitoringId must be <SetVariableMonitoringResponse.setMonitoringResult[0].id of step 2> - constantStreamData.params.interval must be 10 <Configured Clock Aligned MeterValues Interval> * 2 * Step 5: Message: NotifyPeriodicEventStream - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 6: Message: NotifyPeriodicEventStream shall be sent about 10 seconds after step 5 - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 7: Message: NotifyPeriodicEventStream shall be sent about 10 seconds after step 6 - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 9: Message: AdjustPeriodicEventStreamResponse - status must be Accepted * Step 10: Message: NotifyPeriodicEventStream - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 11: Message: NotifyPeriodicEventStream shall be sent about 15 seconds after step 10 - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3> * Step 12: Message: NotifyPeriodicEventStream shall be sent about 15 seconds after step 11 - id must be <OpenPeriodicEventStreamRequest.constantStreamData.id of step 3>

Tool validations
Post scenario validations: The interval of steps 5, 6, 7 shall be equal to <Configured Clock Aligned MeterValues Interval> * 2 The interval of steps 10, 11, 12 shall be equal to <Configured Clock Aligned MeterValues Interval> * 3

6.2.53. Page 805/806 - (2026-02) - TC_O_101_CS - Manual action and reusable state Authorized added

User language selection manual action added and used reusable state Authorized instead.

Main (Test scenario)	
Charging Station	CSMS
1. Execute Reusable State Authorized with AuthorizeResponse : idTokenInfo.language1 <Configured language>	
<u>Manual Action:</u> Present idToken.	
1. The Charging Station sends an AuthorizeRequest Note 1: This step needs to be executed: unless (AuthEnabled is implemented with mutability ReadOnly AND the value is set to false) OR a start button as described at Use case C02 is used (This must be configured at the Test System) OR the idToken is cached. Note 2: Alternatively, EV Driver may select <Configured language1> using button or display on charging station.	2. The Test System responds with an AuthorizeResponse with idTokenInfo.status Accepted idTokenInfo.language1 <Configured language1> <Configured language>
<u>Manual action:</u> User (customer) selects desired language.	
<u>Note(s):</u> - The Charging Station will use the language preference of the EV Driver to display messages.	
3. The Charging Station responds with a SetDisplayMessageResponse	2. The Test System sends a SetDisplayMessageRequest with message.id <Generated displayMessageId> message.message[0].language <Configured supported languages> message.message.language <Configured language> message.message[0].content contentSupported message.message.content contentSupported message.messageExtra[0].language <Configured language1> <Configured customer language> message.messageExtra[0].content ContentInCustomerLanguage1 ContentInCustomerLanguage

Tool validations
* Step 1: Message: AuthorizeRequest - idToken.idToken <Configured valid_idtoken_idtoken> - idToken.type <Configured valid_idtoken_type> * Step 4: Visual inspection Message: SetDisplayMessageResponse - The Message ContentInCustomerLanguage1 is shown on display.
Post scenario validations: - After step 2 the user has selected their preferred language, the Charging Station will show messages with language preference (<Configured language1 customer language>) of the EV Driver, including the SetDisplayMessage of step 2.

6.2.54. Page 814/815 - (2026-02) - TC_Q_102_CS - Made the testcase more flexible regarding the phases used and added step for NotifyEVChargingScheduleRequest

Test case name	V2X Authorisation - ISO15118-20 - Charging only (V2X control) before starting V2X - Allowed Energy Transfer modes omitted
Test case Id	TC_Q_102_CS
...	

Before (Preparations)
Configuration State: - ISO15118Ctrlr.Enabled is true - V2XChargingCtrlr.SupportedEnergyTransferMode contains [<AC_single_phase OR AC_two_phase OR AC_three_phase>, AC_BPT] or [DC, DC_BPT]
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
Communicate that default charging station energy transfer mode will be used	
1. Execute Reusable State Iso1511820V2xControlBpt with controlMode ScheduledControl	
2. Execute Reusable State EVConnectedPreSession	
3. Execute Reusable State [authorized15118] (for 15118-20)	
Note(s): - The Test System omits the allowedEnergyTransfer from the AuthorizeResponse.	
4. The Charging Station sends a NotifyEVChargingNeedsRequest Note(s): - This step is captured by [authorized15118] (for 15118-20), but validated on testcase level.	5. Test System responds with a NotifyEVChargingNeedsResponse with status Accepted
6. Execute Reusable State [energytransferstarted]	
6. The Charging station sends a NotifyEVChargingScheduleRequest	7. The Test System responds with a NotifyEVChargingScheduleResponse with status Accepted
8. Execute Reusable State [energytransferstarted]	

Tool validations
* Step 4: Message: NotifyEVChargingNeedsRequest - evseld must be <Configured evseld> For AC charging station: - chargingNeeds.requestedEnergyTransfer AC_three_phase - chargingNeeds.availableEnergyTransfer [AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER] - chargingNeeds.requestedEnergyTransfer AC_three_phase or AC_single_phase or AC_two_phase - chargingNeeds.availableEnergyTransfer : <Contains; AC_single_phase OR AC_two_phase OR AC_three_phase> <Contains; AC_BPT> <Other allowed supported values; AC_BPT_DER, AC_DER> For DC charging station: - chargingNeeds.requestedEnergyTransfer DC - chargingNeeds.availableEnergyTransfer [DC, DC_BPT, DC_ACDP, DC_ACDP_BPT] - chargingNeeds.availableEnergyTransfer : <Contains; DC AND DC_BPT> <Other allowed supported values; DC_ACDP, DC_ACDP_BPT> - chargingNeeds.controlMode must be <i>ScheduledControl</i> - chargingNeeds.v2xChargingParameters.maxChargePower must be 10000 (W) - chargingNeeds.v2xChargingParameters.maxDischargePower must be 5000 (W) Post scenario validations: N/a

6.2.55. Page 820/821 - (2026-02) - TC_Q_107_CS - Added missing main steps to complete the renegotiation flow

Test case name	V2X Authorisation - ISO15118-20 - Charging only (V2X control) before starting V2X
Test case Id	TC_Q_107_CS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Charging Station sends a NotifyEVChargingNeedsRequest <u>Note(s):</u> - This step is captured by [authorized15118] (for 15118-20), but validated on testcase level.	2. The Test System responds with a NotifyEVChargingNeedsResponse with - status <i>Accepted</i>
Setting ChargingOnly charging profile	

Main (Test scenario)	
3. 4. The Charging Station responds with a SetChargingProfileResponse	4. 3. The Test System sends a SetChargingProfileRequest with evseld <Configured evseld> chargingProfile.chargingProfilePurpose TxProfile chargingProfile.chargingProfileKind Dynamic chargingProfile.transactionId <transactionId> chargingProfile.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].limit 6 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].operationMode ChargingOnly
5. The Charging station sends a NotifyEVChargingScheduleRequest	6. The Test System responds with a NotifyEVChargingScheduleResponse with status Accepted
7. Execute Reusable State EnergyTransferStarted	
6. 9. The Charging Station responds with a NotifyAllowedEnergyTransferResponse	5. 8. The Test System sends a NotifyAllowedEnergyTransferRequest with allowedEnergyTransfer [AC_single_phase, AC_BPT, DC, DC_BPT]
7. 10. The Charging Station sends a NotifyEVChargingNeedsRequest	8. 11. The Test System responds with a NotifyEVChargingNeedsResponse with - status Accepted
13. The Charging Station responds with a SetChargingProfileResponse	12. The Test System sends a SetChargingProfileRequest with chargingProfile.id <Same chargingProfileId as step 3> evseld <Configured evseld> chargingProfile.chargingProfilePurpose TxProfile chargingProfile.chargingProfileKind Dynamic chargingProfile.transactionId <transactionId> chargingProfile.chargingSchedule[0].chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].setpoint -7 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier> chargingProfile.chargingSchedule[0].chargingSchedulePeriod[0].operationMode CentralSetpoint
14. The Charging station sends a NotifyEVChargingScheduleRequest	15. The Test System responds with a NotifyEVChargingScheduleResponse with status Accepted

Tool validations
* Step 1: Message: NotifyEVChargingNeedsRequest - evseld must be <Configured evseld> - chargingNeeds.requestedEnergyTransfer must be one of [AC_single_phase, AC_two_phase, AC_three_phase, DC, DC_ACDP, WPT] - chargingNeeds.availableEnergyTransfer must be <not omitted> * Step 3 Step 4: Message: SetChargingProfileResponse - status must be Accepted * Step 6 Step 9: Message: NotifyAllowedEnergyTransferResponse - status must be Accepted * Step 7 Step 10: Message: NotifyEVChargingNeedsRequest - evseld must be <Configured evseld> - chargingNeeds.requestedEnergyTransfer must be one of [AC_BPT, DC_BPT] - chargingNeeds.availableEnergyTransfer must be <not omitted> * Step 13: Message: SetChargingProfileResponse - status must be Accepted
Post scenario validations: N/a

6.2.56. Page 863/864 - (2026-02) - TC_Q_125_CS - Aligned with other V2X ISO 15115-20 / CHAdeMO testcases

Test case name	Idle operationMode - Idle, minimizing energy consumption - Idle with EvseSleep
Test case Id	TC_Q_125_CS
...	

Before (Preparations)
Configuration State: - SampledDataCtrlr.TxUpdatedInterval = 10 - SampledDataCtrlr.TxUpdatedMeasurands = Energy.Active.Import.Register
Memory State: N/a
Reusable State(s): State is [evconnectedpresession] IF <Configured EVProtocol> is CHAdeMO THEN State is CHAdeMOV2xControlBpt State is [authorized] State is [evconnectedpresession] ELSE IF <Configured EVProtocol> is ISO15118 THEN State is Iso1511820V2xControlBpt State is [evconnectedpresession] State is [authorized15118] (for 15118-20) END IF State EnergyTransferStarted

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
1. The Charging Station sends a NotifyEVChargingNeedsRequest Note(s) : - This step is captured by the preparation steps, but validated on testcase level.	2. Test System responds with a NotifyEVChargingNeedsResponse with status Accepted
Without EVSE sleeping	
2. 4. The Charging Station responds with a SetChargingProfileResponse with status must be <i>Accepted</i>	1. 3. The Test System sends a SetChargingProfileRequest with evseld <Configured evseld> chargingProfile - transactionId <transactionId> - chargingProfilePurpose TxProfile - chargingSchedule.chargingRateUnit W - chargingSchedule.chargingSchedulePeriod.operationMode Idle - chargingSchedule.chargingSchedulePeriod.evseSleep false
5. The Charging station sends a NotifyEVChargingScheduleRequest Note(s) : - This step is required if <Configured EVProtocol> is ISO15118 , otherwise this step is optional.	6. The Test System responds with a NotifyEVChargingScheduleResponse with status Accepted
7. Execute Reusable State <i>EnergyTransferStarted</i>	
3. 8. The Charging Station sends following TransactionEventRequest	4. 9. The Test System responds with a TransactionEventResponse
5. 10. The Charging Station sends following TransactionEventRequest	6. 11. The Test System responds with a TransactionEventResponse
With EVSE sleeping	
8. 13. The Charging Station responds with a SetChargingProfileResponse	7. 12. The Test System sends a SetChargingProfileRequest with evseld <Configured evseld> chargingProfile - transactionId <transactionId> - chargingProfilePurpose TxProfile - chargingSchedule.chargingRateUnit W - chargingSchedule.chargingSchedulePeriod.operationMode Idle - chargingSchedule.chargingSchedulePeriod.evseSleep true
9. 14. The Charging Station sends following TransactionEventRequest	10. 15. The Test System responds with a TransactionEventResponse
11. 16. The Charging Station sends following TransactionEventRequest	12. 17. The Test System responds with a TransactionEventResponse

Tool validations * Step 2 Step 4: Message: SetChargingProfileResponse - status must be <i>Accepted</i> * Step 3 Step 8: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionId must be <i><transactionId></i> - evseSleep must be <i>false</i> <i><omitted></i> - meterValue[0].sampledValue[0].measurand must be <i>Energy.Active.Import.Register</i> * Step 5 Step 10: Message: TransactionEventRequest - eventType must be <i>Updated</i> - transactionInfo.transactionId must be <i><transactionId></i> - evseSleep must be <i>false</i> <i><omitted></i> * Step 8 Step 13: Message: SetChargingProfileResponse - status must be <i>Accepted</i> * Step 9 Step 14: Message: TransactionEventRequest - eventType must be <i>Updated</i> - eventData.transactionData.transactionId must be <i><transactionId></i> - evseSleep must be <i>true</i> - meterValue[0].sampledValue[0].measurand must be <i>Energy.Active.Import.Register</i> * Step 11 Step 16: Message: TransactionEventRequest - eventType must be <i>Updated</i> - eventData.transactionData.transactionId must be <i><transactionId></i> - evseSleep must be <i>true</i> - meterValue[0].sampledValue[0].measurand must be <i>Energy.Active.Import.Register</i> - meterValue[0].sampledValue[0].value must be identical to value in Step 9 Post scenario validations: N/a

6.2.57. Page 809-878 - (2026-02) - TC_Q_1XX_CS - NotifyEVChargingNeeds also needs to be used for V2X with CHAdeMO

Added reusable state for BPT with CHAdeMO:

State	CHAdeMOV2xControlBpt
System under test	Charging Station
Description	Configures the EV to use <i>AC_BPT</i> or <i>DC_BPT</i> energy transfer mode.
Exposes	<i><transactionId></i>

Before (Preparations)
Configuration State: - CHAdeMOCtrlr.Enabled is <i>true</i> - V2XChargingCtrlr.V2XEnabled is <i>true</i>
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
Agree on energy transfer mode	
Manual Action: Use EV to communicate EV charging limits: BPT true controlMode DynamicControl maxChargePower 10000 (W) maxDischargePower 5000W (W)	

Tool validations
N/a
Post scenario validations: N/a

Applied new BPT reusable state at testcases:

Test case name	...
Test case Id	TC_Q_1XX_CS
...	

Before (Preparations)
Configuration State: ...
Memory State: ...
Reusable State(s): IF <Configured EVProtocol> is CHAdeMO THEN State is CHAdeMOV2xControlBpt State is Authorized State is [evconnectedpresession] ELSE IF <Configured EVProtocol> is ISO15118 THEN State is Iso1511820V2xControlBpt State is [evconnectedpresession] State is [authorized15118] (for 15118-20) END IF

Updated note - NotifyEVChargingNeedsRequest also needs to be sent in case of CHAdeMO:

	Text
Old	<u>Note(s):</u> - This step is applicable if <Configured EVProtocol> is ISO15118 and is captured by [authorized15118] (for 15118-20), but validated on testcase level.
New	<u>Note(s):</u> - This step is captured by the preparation steps, but validated on testcase level.

Updated note - NotifyEVChargingScheduleRequest is allowed to be sent in case of CHAdeMO:

	Text
Old	<u>Note(s):</u> - This step is applicable if <Configured EVProtocol> is ISO15118
New	<u>Note(s):</u> - This step is required if <Configured EVProtocol> is ISO15118, otherwise this step is optional.

Preparation steps updated - TC_Q_126_CS & TC_Q_127_CS:

	Text
Old	State is [csIso1511820V2xControlBpt]
New	State is [authorized15118] (for 15118-20)

6.2.58. Page 877 - (2026-02) - TC_Q_130_CS - Updated tool validation requestedEnergyTransfer

Test case name	V2X Authorisation - ISO15118-20 - has ISO15118ServiceRenegotiationSupport - Charging needs rejected
Test case Id	TC_Q_130_CS
...	

Tool validations
* Step 1: Message: NotifyEVChargingNeedsRequest - evseld must be <Configured evseld> For AC charging station: - chargingNeeds.requestedEnergyTransfer AC_BPT - chargingNeeds.availableEnergyTransfer [AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER] For DC charging station: - chargingNeeds.requestedEnergyTransfer DC_BPT - chargingNeeds.availableEnergyTransfer [DC, DC_BPT, DC_ACDP, DC_ACDP_BPT] - chargingNeeds.controlMode must be <i>ScheduledControl</i> - chargingNeeds.v2xChargingParameters.maxChargePower must be 10000 (W) - chargingNeeds.v2xChargingParameters.maxDischargePower must be 5000 (W) * Step 3: Message: NotifyEVChargingNeedsRequest - evseld must be <Configured evseld> For AC charging station: - chargingNeeds.requestedEnergyTransfer <Not a bi-directional energyTransferMode> - chargingNeeds.availableEnergyTransfer [AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER] For DC charging station: - chargingNeeds.requestedEnergyTransfer DC - chargingNeeds.availableEnergyTransfer [DC, DC_BPT, DC_ACDP, DC_ACDP_BPT] - chargingNeeds.controlMode must be <i>ScheduledControl</i> - chargingNeeds.v2xChargingParameters.maxChargePower must be 10000 (W) - chargingNeeds.v2xChargingParameters.maxDischargePower must be <omitted> Post scenario validations: N/a

6.2.59. Page 879 - (2026-06) - TC_R_100_CS - Used reusable state booted

The test case was updated to make use of the reusable state booted.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station sends a ClearDERControlResponse	1. The Test System sends a ClearDERControlRequest with isDefault true
4. The Charging Station sends a ClearDERControlResponse	3. The Test System sends a ClearDERControlRequest with isDefault false
6. The Charging Station sends a SetDERControlResponse	5. The Test System sends a SetDERControlRequest with controlId control1 isDefault true controlType <CS supported controlType1> *.priority 6
8. The Charging Station sends a SetDERControlResponse	7. The Test System sends a SetDERControlRequest with controlId control2 isDefault false controlType <CS supported controlType2> *.priority 6 *.startTime <Current DateTime + 24 hours> *.duration 300
10. The Charging Station responds with a ResetResponse	9. The Test System sends a ResetRequest with type Immediate
9. Execute Reusable State Booted	
12. 11. The Charging Station sends a GetDERControlResponse	11. 10. The Test System sends a GetDERControlRequest with isDefault true
13. 12. The Charging Station sends a ReportDERControlRequest	14. 13. The Test System sends a ReportDERControlResponse
Note(s): - If tbc is True at Step 13 12 then step 13 12 and 14 13 will be repeated	
16. 15. The Charging Station sends a GetDERControlResponse	15. 14. The Test System sends a GetDERControlRequest with isDefault false
17. 16. The Charging Station sends a ReportDERControlRequest	18. 17. The Test System sends a ReportDERControlResponse
Note(s): - If tbc is True at Step 17 16 then step 17 16 and 18 17 will be repeated	

Tool validations
... * Step 42 11: Message: GetDERControlResponse - status must be <i>Accepted</i> * Step 43 12: Message: ReportDERControlRequest Field corresponding to configuring <CS supported controlType1> - *[0].id must be <i>control1</i> Field corresponding to configuring <CS supported controlType2> - <i>Corresponding field should be omitted.</i> * Step 46 15: Message: GetDERControlResponse - status must be <i>Accepted</i> * Step 47 16: Message: ReportDERControlRequest Field corresponding to configuring <CS supported controlType2> - *[0].id must be <i>control2</i> Field corresponding to configuring <CS supported controlType1> - <i>Corresponding field should be omitted.</i>
Post scenario validations: N/a

6.2.60. Page 881 - (2026-02) - TC_R_101_CS - Test case removed

This testcases has been removed, because TC_B_53_CS will be responsible for performing all device model validations.

6.2.61. Page 899 - (2026-02) - TC_R_108_CS - Use static curves to trigger, instead of configurable ones

Test case name	Charging station reporting a DER event
Test case Id	TC_R_108_CS
Use case Id(s)	R05
Requirement(s)	R05.FR.01, R05.FR.02, R05.FR.03, R05.FR.04
System under test	Charging Station
Description	Configure a DER Control which will always trigger and affect the charging rate of the running transaction, then add another DER Control to test overriding the previous one. After that, clearing all DER controls to test it reports that the alert is cleared. This test case uses DER controls that, when installed, trigger during normal operation: a HVMustTrip curve (configured to trip at nominal voltage) and a LFMustTrip curve (configured to trip at nominal frequency). Curves that are a function of power (WattPF, WattVar) or voltage (HV/LVMustTrip, VoltVar, VoltWatt) are usually easier to trigger than frequency-based curves.
Purpose	To check if the CS reports when DER controls are taking over.
Prerequisite(s)	CS supports DER Control (DCDERCtrlr or ACDERCtrlr is present for EVSE) CS is a DC charging station CS supports ISO15118-20

Main (Test scenario)	
Charging Station	CSMS
Note: Curve HVCURVE is defined as: {(1.00, 5.0), (1.20, 0.16)} – this should trip at nominal voltage (1.00) after 5 seconds.	
Note: Curve LFCURVE is defined as: {(50.5, 5.0), (60.5, 10.0)} – this should trip after 5 seconds below 50.5 Hz (EU, and trip after 10 seconds below 60.5 Hz (US).	
Note: Wait 10 seconds after start of energy transfer before setting DER control	

Main (Test scenario)	
2. The Charging Station sends a SetDERControlResponse	1. The Test System sends a SetDERControlRequest with controlId <i>control1</i> isDefault <i>true</i> controlType <i>HVMustTrip</i> curve <i>HVCURVE</i> *.priority 6
3. The Charging Station sends a NotifyDERAlarmRequest	4. The Test System responds with a NotifyDERAlarmResponse
6. The Charging Station sends a ClearDERControlResponse +	5. The Test System sends a ClearDERControlRequest with isDefault <i>true</i>
7. The Charging Station sends a NotifyDERAlarmRequest +	8. The Test System responds with a NotifyDERAlarmResponse
10. The Charging Station sends a SetDERControlResponse	9. The Test System sends a SetDERControlRequest with controlId <i>control2</i> isDefault <i>true</i> controlType <i>LFMustTrip</i> curve <i>LFCURVE</i> *.priority 5
11. The Charging Station sends a NotifyDERAlarmRequest	12. The Test System responds with a NotifyDERAlarmResponse
14. The Charging Station sends a ClearDERControlResponse	13. The Test System sends a ClearDERControlRequest with isDefault <i>true</i>
15. The Charging Station sends a NotifyDERAlarmRequest	16. The Test System responds with a NotifyDERAlarmResponse

Tool validations
* Step 2: Message: SetDERControlResponse - status must be <i>Accepted</i> * Step 3: Message: NotifyDERAlarmRequest - controlType must be <i>HVMustTrip</i> - gridEventFault must be <i>OverVoltage</i> - alarmEnded must be <i>false</i> or <absent> - timestamp must be <Current DateTime> * Step 6: Message: ClearDERControlResponse - status must be <i>Accepted</i> * Step 7: Message: NotifyDERAlarmRequest - controlType must be <i>HVMustTrip</i> - gridEventFault must be <i>OverVoltage</i> - alarmEnded must be <i>true</i> - timestamp must be <Current DateTime> * Step 10: Message: SetDERControlResponse - status must be <i>Accepted</i> * Step 11: Message: NotifyDERAlarmRequest - controlType must be <i>LFMustTrip</i> - gridEventFault must be <i>UnderFrequency</i> - alarmEnded must be <i>false</i> or <absent> - timestamp must be <Current DateTime> * Step 14: Message: ClearDERControlResponse - status must be <i>Accepted</i> * Step 15: Message: NotifyDERAlarmRequest - controlType must be <i>LFMustTrip</i> - gridEventFault must be <i>UnderFrequency</i># - alarmEnded must be <i>true</i> - timestamp must be <Current DateTime>
Post scenario validations: No other NotifyDERAlarmRequest shall be received, except those mentioned in the tool validation steps.

6.2.62. Page 902/903 - (2026-02) - TC_S_104_CS - Resolved editorial issues

Steps 7 to 14 were incorrectly swapped, and at the tool validations the elements incorrectly refer to Component for some of the entries. Like baseReportData[0].*component*.variable.name

Test case name	Battery Swap - Charging - Variables validation
Test case Id	TC_S_104_CS
Use case Id(s)	S04
Requirement(s)	S04.FR.08, S04.FR.09, S04.FR.10
System under test	Charging Station
Description	Charging station is able to support battery swapping.
Purpose	To verify whether the Charging Station is able to be configured correctly.

Test case name	Battery Swap - Charging - Variables validation
Prerequisite(s)	Charging Station supports battery swapping (BatterySwapCtrlr.Available is <i>true</i>)

Before (Preparations)
Configuration State: BatterySwapCtrlr.TargetSoc is 80 BatterySwapCtrlr.MaxSoc is 90
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with GetBaseReportResponse	1. The Test System sends GetBaseReportRequest with requestId = <Generated requestId> reportBase = <i>FullInventory</i>
3. The Charging Station sends NotifyReportRequest	4. The Test System responds with NotifyReportResponse
5. The Charging Station sends NotifyReportRequest	6. The Test System responds with NotifyReportResponse
<u>Note:</u> If NotifyReportRequest.tbc is true in step 3 then steps 5 and 6 repeat until NotifyReportRequest.tbc is false in step 5.	
8. The Charging Station responds with SetVariablesResponse	7. The Test System sends a SetVariablesRequest with setVariableData[0].component.name = <i>BatterySwapCtrlr</i> setVariableData[0].variable.name = <i>TargetSoc</i> setVariableData[0].attributeValue = "91"
10. The Charging Station responds with SetVariablesResponse	9. The Test System sends a SetVariablesRequest with setVariableData[0].component.name = <i>BatterySwapCtrlr</i> setVariableData[0].variable.name = <i>MaxSoc</i> setVariableData[0].attributeValue = "79"
12. The Charging Station responds with SetVariablesResponse	11. The Test System sends a SetVariablesRequest with setVariableData[0].component.name = <i>BatterySwapCtrlr</i> setVariableData[0].variable.name = <i>TargetSoc</i> setVariableData[0].attributeValue = "79"
14. The Charging Station responds with SetVariablesResponse	13. The Test System sends a SetVariablesRequest with setVariableData[0].component.name = <i>BatterySwapCtrlr</i> setVariableData[0].variable.name = <i>MaxSoc</i> setVariableData[0].attributeValue = "79"

Tool validations
* Step 2: Message: GetBaseReportResponse - status must be <i>Accepted</i> - statusInfo is absent or statusInfo.reasonCode = "NoError" * Step 3: Message: NotifyReportRequest - requestId must be <Generated requestId> - generatedAt must be <timestamp at charging station> - seqNo must be 0 * Step 5: Message: NotifyReportRequest - requestId must be <Generated requestId> - generatedAt must be <timestamp at charging station> - seqNo must be <incremented seqNo> * Step 8: Message: SetVariablesResponse - setVariableResult[0].attributeStatus must be <i>Rejected</i> * Step 10: Message: SetVariablesResponse - setVariableResult[0].attributeStatus must be <i>Rejected</i> * Step 12: Message: SetVariablesResponse - setVariableResult[0].attributeStatus must be <i>Accepted</i> * Step 14: Message: SetVariablesResponse - setVariableResult[0].attributeStatus must be <i>Accepted</i>
Post scenario validations: In the collected data of all NotifyReportRequest.reportData it must contain: - baseReportData[0].component.name must be <i>TxCtrlr</i> - baseReportData[0].component.variable.name must be <i>TxStartPoint</i> - baseReportData[0].component.variableAttribute.type is absent or must be <i>Actual</i> - baseReportData[0].component.variableAttribute.value must be <i>EVConnected</i> - baseReportData[0].component.variableAttribute.mutability must be <i>ReadOnly</i> - baseReportData[0].component.variableCharacteristics.dataType must be <i>MemberList</i> - baseReportData[0].component.variableCharacteristics.valuesList must be <i>EVConnected</i> - baseReportData[1].component.name must be <i>TxCtrlr</i> - baseReportData[1].component.variable.name must be <i>TxStopPoint</i> - baseReportData[1].component.variableAttribute.type is absent or must be <i>Actual</i> - baseReportData[1].component.variableAttribute.value must be <i>EVConnected</i> - baseReportData[1].component.variableAttribute.mutability must be <i>ReadOnly</i> - baseReportData[1].component.variableCharacteristics.dataType must be <i>MemberList</i> - baseReportData[1].component.variableCharacteristics.valuesList must be <i>EVConnected</i>

6.2.63. Page 904/905 - (2026-02) - TC_S_105_CS - Updated test case

Removed Target SoC test steps. This will be covered in a separate test case.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with GetVariablesResponse	1. Test System sends GetVariablesRequest with: - variable.name = "SoC" - component.name = "BatteryCartridge" - component.evse.id = "<i>" - attributeType is omitted

Main (Test scenario)	
<u>Note:</u> Step 1 and 2 request the SoC for each configured EVSE	
4. The Charging Station responds with GetVariablesResponse	3. Test System sends SetVariablesRequest with: - variable.name = "TargetSoC" - component.name = "BatterySwapCtrlr" - attributeValue = "<Highest returned SoC value from step 2, that is lower than 96 percent + 2>" - attributeType is omitted
6. 4. The Charging Station responds with GetVariablesResponse SetVariablesResponse	5. 3. Test System sends SetVariablesRequest with: - variable.name = "MaxSoC" - component.name = "BatterySwapCtrlr" - attributeValue = "<Highest returned SoC value from step 2, that is equal to or lower than 96 percent + 4>" - attributeType is omitted
7. The Charging Station notifies the CSMS about the status change of the slot.	8. The Test System responds accordingly.
<u>Note:</u> Step 7 shall be triggered when the battery's SoC reaches TargetSoC	
9. 5. The Charging Station sends a TransactionEventRequest	10. 6. The Test System responds with a TransactionEventResponse
<u>Note:</u> Step 9 5 shall be triggered when the battery's SoC reaches the MaxSoC	

6.2.64. Page 916 - (2026-02) - Reusable state Booted now accepts StatusNotification Unavailable [1162]

Booted

State	Booted
System under test	Charging Station
Description	This state will reset or power cycle the Charging Station, depending on the testcase. The charging station ends in a state where it is booted back up and is in idle mode.

Before (Preparations)
Configuration State: N/a
Memory State: N/a
Reusable State(s): If Product subtype is <i>Battery Swapping Station</i> , then execute <i>BootedBatterySwapping</i> instead

Main (Test scenario)	
Charging Station	CSMS
<u>Manual Action:</u> Power cycle the Charging Station. OR execute step 1 and 2, depending on the testcase.	
2. The Charging Station responds with a ResetResponse with status Accepted	1. The Test System sends a ResetRequest
3. The Charging Station notifies the CSMS about the unavailability of all connectors. <u>Note:</u> This step is optional.	4. The Test System responds accordingly.
5. The Charging Station sends a BootNotificationRequest	6. The Test System responds with a BootNotificationResponse with status Accepted

Main (Test scenario)	
7. The Charging Station notifies the CSMS about the current state of all connectors.	8. The Test System responds accordingly.
9 The Charging Station sends a SecurityEventNotificationRequest	10 The Test System responds with a SecurityEventNotificationResponse

Tool validations
* Step 2: Message: ResetResponse - status <i>Accepted</i> * Step 3: Message: StatusNotificationRequest - connectorStatus <i>Unavailable</i> - evseld not 0 - connectorId not 0 Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Unavailable"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> * Step 7: Message: StatusNotificationRequest - connectorStatus <i>Available</i> - evseld not 0 - connectorId not 0 Message: NotifyEventRequest - eventData[0].trigger <i>Delta</i> - eventData[0].actualValue <i>"Available"</i> - eventData[0].component.name <i>"Connector"</i> - eventData[0].variable.name <i>"AvailabilityState"</i> * Step 9: Message: SecurityEventNotificationRequest - type must be <i>StartupOfTheDevice</i> OR <i>ResetOrReboot</i> Post scenario validations: State is <i>Booted</i>

6.2.65. Page 949/952 - (2026-02) - EVConnectedPreSessionBatterySwapping/EVDisconnectedBatterySwapping - Updated tool validations

Tool validations for batteryData required fields removed (validated at schema level). Validations added to ensure the array length matches the number of batteries inserted.

Tool validations
... * Step 5: Message: BatterySwapRequest - eventType must be <i>BatteryIn</i> - requestId must be <i><Configured requestId></i> - idToken.idToken must be <i><Configured valid_idtoken_idtoken></i> For each inserted battery: - batteryData.evseId must be <i><not omitted></i> - batteryData.serialNumber must be <i><not omitted></i> - batteryData.soC must be <i><not omitted></i> - batteryData.soH must be <i><not omitted></i> - batteryData[] length must be equal to <i><number of inserted batteries></i>

6.2.66. Page 1207 - (2026-02) - TC_K_102_CS - Made the testcase useable with a real EV

Test case name	Set Charging Profile - limitAtSoc
Test case Id	TC_K_102_CS
...	
Prerequisite(s)	- Charging station supports receiving SoC from the EV. e.g. by CHaDeMo, ISO15118-2 or ISO15118-20 - SmartChargingCtrlr.SupportsFeature[LimitAtSoc] is <i>true</i> EV must have a SoC level of between 1 and 98 percent.

Main (Test scenario)	
Charging Station	CSMS
2. The Charging Station responds with a SetChargingProfileResponse	1. The Test System sends a SetChargingProfileRequest with evseId <i>0</i> chargingProfile.id <i><Configured chargingProfileId></i> evseId <i><Configured evseId></i> chargingProfile.chargingProfilePurpose <i>TxDefaultProfile</i> chargingProfile.chargingSchedule.chargingRateUnit <i><Configured chargingRateUnit></i> chargingProfile.chargingSchedule.chargingSchedulePeriod[0].startPeriod <i>0</i> chargingProfile.chargingSchedule.chargingSchedulePeriod[0].limit <i>8 * <limit multiplier></i> Note: Check [csKSmartChargingChargingProfileLimits] for <i><limit multiplier></i> chargingProfile.chargingSchedule.chargingSchedulePeriod[0].numberPhases <i><Configured numberPhases></i> chargingProfile.chargingSchedule.limitAtSoc.soc <i><Configured limitAtSocSoc></i> 99 chargingProfile.chargingSchedule.limitAtSoc.limit <i>6 * <limit multiplier></i> Note: Check [csKSmartChargingChargingProfileLimits] for <i><limit multiplier></i>
3. Execute Reusable State <i>EnergyTransferStarted</i>	
Manual Action: Use EV to send out a SoC value less than <i><Configured limitAtSocSoc></i>	
5. The Charging Station responds with a GetCompositeScheduleResponse	4. The Test System sends a GetCompositeScheduleRequest with evseId <i><Configured evseId></i> duration is <i>900</i> chargingRateUnit <i><Configured chargingRateUnit></i>
Manual Action: Use EV to send out a SoC value greater or equal to <i><Configured limitAtSocSoc></i>	

Main (Test scenario)

7. The Charging Station responds with a GetCompositeScheduleResponse SetChargingProfileResponse	6. The Test System sends a GetCompositeScheduleRequest with evseld <Configured evseld> duration is 900 chargingRateUnit <Configured chargingRateUnit> The Test System sends a SetChargingProfileRequest with evseld <Configured evseld> chargingProfile.id <chargingProfile.id from step 1> chargingProfile.chargingProfilePurpose TxDefaultProfile chargingProfile.chargingSchedule.chargingRateUnit <Configured chargingRateUnit> chargingProfile.chargingSchedule.chargingSchedulePeriod[0].startPeriod 0 chargingProfile.chargingSchedule.chargingSchedulePeriod[0].limit 8 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier> chargingProfile.chargingSchedule.chargingSchedulePeriod[0].numberPhases <Configured numberPhases> chargingProfile.chargingSchedule.limitAtSoc.soc 1 chargingProfile.chargingSchedule.limitAtSoc.limit 6 * <limit multiplier> Note: Check [csKSmartChargingChargingProfileLimits] for <limit multiplier>
Manual Action: Use EV to send out a SoC value less than <Configured limitAtSocSoc>	
9. The Charging Station responds with a GetCompositeScheduleResponse	8. The Test System sends a GetCompositeScheduleRequest with evseld <Configured evseld> duration is 900 chargingRateUnit <Configured chargingRateUnit>

Tool validations

...

* Step 7:
Message: **GetCompositeScheduleResponse** **SetChargingProfileResponse**
- **status** must be Accepted
- ~~**schedule.chargingRateUnit** must be <Configured chargingRateUnit>~~
- ~~**schedule.chargingSchedulePeriod[0].startPeriod** must be 0~~
- ~~**schedule.chargingSchedulePeriod[0].limit** must be 6 * <limit multiplier>~~
Note: Check [\[csKSmartChargingChargingProfileLimits\]](#) for <limit multiplier>

* Step 9:
Message: **GetCompositeScheduleResponse**
- **status** must be Accepted
- **schedule.chargingRateUnit** must be <Configured chargingRateUnit>
- **schedule.chargingSchedulePeriod[0].startPeriod** must be 0
- **schedule.chargingSchedulePeriod[0].limit** must be 8 * **6** * <limit multiplier>

6.3. CSMS

6.3.1. Page 1022 - 1024 - (2026-04) - TC_C_47_CSMS/TC_C_48_CSMS/TC_C_49_CSMS - Removed configuration variable masterPassGroupId

The <Configured MasterPassGroupId> variable has been removed from the OCTT. The MasterPassGroupId can be provided to the OCTT by the CSMS through a **SetVariableRequest**.

Main (Test scenario)	
Charging Station	CSMS
Manual action: <i>trigger CSMS to set the masterPassGroupId</i>	
2. The Charging Station responds with SetVariablesResponse with attributeStatus = <i>Accepted</i>	1. Test System sends SetVariablesRequest with: - variable.name = <i>"MasterPassGroupId"</i> - component.name = <i>"AuthCtrlr"</i> - attributeValue = <i><Specified masterPassGroupId></i>

Where applicable tool validations mentioning <Configured masterPassGroupId> were replace with the masterPassGroupId specified through the **SetVariablesRequest**

Tool validations
* Step 4: Message AuthorizeResponse - idTokenInfo.groupIdToken.idToken <i><Configured masterPassGroupId></i> <i><masterPassGroupId from step 1></i>
* Step 6: Message TransactionEventResponse - idTokenInfo.groupIdToken.idToken <i><Configured masterPassGroupId></i> <i><masterPassGroupId from step 1></i>
Post scenario validations: - N/a

6.3.2. Page 877/878 - (2026-02) - TC_Q_130_CS - BPT manual actions updated

The AC and DC BPT parameters were removed from the initial manual action because they are no longer necessary. Instead, a single BPT flag is sufficient, as the AC/DC configuration is determined by the vehicle. The manual action after renegotiation was also removed because this step is not truly a manual action. It represents a fallback to the standard operational mode rather than an explicit user intervention.

Main (Test scenario)	
Charging Station	CSMS
<i>Agree on energy transfer mode</i>	
Manual Action: <i>Use EV to communicate EV charging needs:</i> For AC charging station: requestedEnergyTransfer <i>AC_BPT</i> availableEnergyTransfer <i>[AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER]</i> For DC charging station: requestedEnergyTransfer <i>DC_BPT</i> availableEnergyTransfer <i>[DC, DC_BPT, DC_ACDP, DC_ACDP_BPT]</i> BPT <i>true</i> controlMode <i>ScheduledControl</i> maxChargePower <i>10000 (W)</i> maxDischargePower <i>5000 (W)</i>	

Main (Test scenario)	
1. The Charging Station sends a NotifyEVChargingNeedsRequest	2. The Test System responds with a NotifyEVChargingNeedsResponse with - status <i>Rejected</i>
<i>Renegotiate</i>	
<u>Manual Action:</u> Use EV to communicate EV charging needs: For AC charging station: requestedEnergyTransfer AC_two_phase availableEnergyTransfer [AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER] For DC charging station: requestedEnergyTransfer DC availableEnergyTransfer [DC, DC_BPT, DC_ACDP, DC_ACDP_BPT] controlMode ScheduledControl maxChargePower 10000 (W)	
3. The Charging Station sends a NotifyEVChargingNeedsRequest	4. The Test System responds with a NotifyEVChargingNeedsResponse with - status <i>NoChargingProfile</i>

6.3.3. Page 944 - (2026-02) - Reusable state Iso1511820V2xControlBpt - BPT manual actions updated

The AC and DC BPT parameters were removed from the initial manual action because they are no longer necessary. Instead, a single BPT flag is sufficient, as the AC/DC configuration is determined by the vehicle.

Iso1511820V2xControlBpt

State	Iso1511820V2xControlBpt
System under test	Charging Station
...	

Main (Test scenario)	
Charging Station	CSMS
<i>Agree on energy transfer mode</i>	
<u>Manual Action:</u> Use EV to communicate EV charging limits: For AC charging station: requestedEnergyTransfer AC_BPT availableEnergyTransfer [AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER] For DC charging station: requestedEnergyTransfer DC_BPT availableEnergyTransfer [DC, DC_BPT, DC_ACDP, DC_ACDP_BPT] BPT true controlMode <i>DynamicControl</i> or <i>ScheduledControl</i> maxChargePower 10000 (W) maxDischargePower 5000W (W)	

6.3.4. Page 1100/1101 - (2026-02) - TC_E_117_CSMS - Updated prerequisites and SetChargingProfileRequest tool validations

The configuration *SmartChargingCtrlr.ChargingProfilePersistence* is *<absent>* was moved to Prerequisites and SetChargingProfileRequest tool validations were updated.

Test case name	Set Charging Profile - Resuming transaction after interruption - TxResumptionTimeout not expired - TxAllowEnergyTransferResumption is true
Test case Id	TC_E_117_CSMS
...	
Prerequisite(s)	SmartChargingCtrlr.ChargingProfilePersistence is <i><absent></i> TxCtrlr.TxResumptionTimeout is 999 TxCtrlr.TxAllowEnergyTransferResumption is <i>true</i>

Before (Preparations)
Configuration State: SmartChargingCtrlr.ChargingProfilePersistence is <i><absent></i> N/a
...

Main (Test scenario)	
Charging Station	CSMS
...	
7. The Test System sends a SecurityEventNotificationRequest - type must be StartupOfTheDevice OR ResetOrReboot	8. The CSMS responds with a SecurityEventNotificationResponse
...	

Tool validations
...
* Step 15: Message: SetChargingProfileRequest must contain an identical <i>chargingProfile</i> as in Step 1 or identical as of time of Step 15. Message: SetChargingProfileRequest - chargingProfile.chargingProfilePurpose must be <i>TxProfile</i>

6.3.5. Page 1006 - (2026-02) - TC_B_105_CSMS - Manual action added

The note before step 1. has been changed to indicate a manual action is required.

Main (Test scenario)	
Charging Station	CSMS
...	
Note: Request CSMS to set a network configuration that includes APN connection details, but no VPN details.	
Manual action: Request CSMS to set a network configuration that includes APN connection details, but no VPN details.	
2. The Charging Station Test system responds with SetVariablesResponse(s) with - setVariableResult.attributeStatus = Accepted for each variable	1. CSMS sends a SetVariablesRequest on component NetworkConfiguration[<Configured non-active configurationSlot>]

6.3.6. Page 1002 - (2026-02) - TC_B_100_CSMS - Manual action added and tool validations updated

The note was replaced with a manual action and tool validations were updated.

Main (Test scenario)	
Charging Station	CSMS
Note: Request CSMS to send a network profile including identity and password. Manual Action : Request CSMS to send a network profile including identity and password.	

Tool validations
* Step 1: Message SetNetworkProfileRequest - configurationSlot is <Configured configurationSlot> - connectionData.messageTimeout <Configured messageTimeout> - connectionData.ocppCsmsUrl <Configured ocppCsmsUrl> - connectionData.ocppInterface <Configured ocppInterface> - connectionData.ocppTransport JSON - connectionData.ocppVersion OCPP20 OCPP21 - connectionData.securityProfile <Configured securityProfile> <Active securityProfile> - connectionData.identity <Configured identity> - connectionData.basicAuthPassword <Configured basicAuthPassword>

6.3.7. Page 1007/1008 - (2026-02) - TC_B_116_CSMS -

Added SecurityEventNotificationRequest steps and updated tool validations.

Test case name	Reset ImmediateAndResume - With Ongoing Transaction and SmartCharging - resuming energytransfer
Test case Id	TC_B_116_CSMS
...	
Description	This test case covers how the CSMS can remotely request the Charging Station to reset itself by sending a ResetRequest during a transaction. When ResetRequest "Immediate" "ImmediateAndResume" is sent the charging stations will try to stop all transactions before rebooting.

Main (Test scenario)	
Charging Station	CSMS
...	
9. The Test System notifies the CSMS about the current state of all connectors. For <Configured connectorId>: Message: StatusNotificationRequest - connectorStatus Occupied Message: NotifyEventRequest - trigger Delta - actualValue "Occupied" - component.name "Connector" - variable.name "AvailabilityState" For <Other connector(s)>: Message: StatusNotificationRequest - connectorStatus Available Message: NotifyEventRequest - trigger Delta - actualValue "Available" - component.name "Connector" - variable.name "AvailabilityState"	10. The CSMS responds accordingly.

Main (Test scenario)	
11. The Test System sends a SecurityEventNotificationRequest with - type <i>ResetOrReboot</i>	12. The CSMS responds with a SecurityEventNotificationResponse
14. 13. The Test System sends a TransactionEventRequest . - eventType <i>Updated</i> - triggerReason <i>TxResumed</i> - idToken is omitted	12. 14. The CSMS responds with a TransactionEventResponse .
14. 16. The Test System responds with a SetChargingProfileResponse with status <i>Accepted</i>	13. 15. The CSMS sends a SetChargingProfileRequest

Tool validations
* Step 1: ... * Step 13 15: Message: SetChargingProfileRequest must either contain an identical <i>chargingProfile</i> as in Step 1 or identical from the time of step 11 onwards. Message: SetChargingProfileRequest - chargingProfile.chargingProfilePurpose <i>TxProfile</i>

6.3.8. Page 1026 - (2026-06) - TC_C_51_CSMS - Corrected OCSP response and removed post scenario validations

The OCSP response was previously incorrectly marked as valid, this has now been corrected to indicate a revoked status.

Post-scenario validations have been removed, as they are not applicable to the CSMS.

Test case name	Authorization using Contract Certificates 15118 - Online - Local contract certificate validation - Rejected
Test case Id	TC_C_51_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Test System sends an AuthorizeRequest With idToken.idToken <Configured valid_idtoken_idtoken> idToken.type <Configured valid_idtoken_type> iso15118CertificateHashData contains <hashes from configured (V2G) certificate chain	2. The CSMS sends an OCSP request to responder URL of iso15118CertificateHashData to check validity
3. The Test System OCSP service responds that certificate is valid revoked .	4. The CSMS responds with a AuthorizeResponse

Tool validations
...
Post scenario validations: EV is not authorized and shall not charge: Charging Station does not send TransactionEventRequest with: - triggerReason = <i>Authorized</i> or chargingState = <i>Charging</i> N/a

6.3.9. Page 1038 - (2026-02) - TC_C_119_CSMS / TC_C_120_CSMS / TC_C_125_CSMS / TC_C_126_CSMS - Added missing triggerReason and stoppedReason

Test case Id	TC_C_119_CSMS / TC_C_120_CSMS / TC_C_125_CSMS / TC_C_126_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
X. The Test System sends a TransactionEventRequest with eventType <i>Ended</i> idToken.type <i>DirectPayment</i> ... transactionInfo.stoppedReason <i>Local</i> triggerReason <i>StopAuthorized</i> 	4. The CSMS responds with a TransactionEventResponse

6.3.10. Page 1046/1047 - (2026-02) - TC_C_131_CSMS - Use idToken from the RequestStartTransactionRequest, which contains the PspRef

Test case name	Ad hoc payment via static or dynamic QR code - success
Test case Id	TC_C_131_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
3. The Test System sends a TransactionEventRequest with eventType <i>Started</i> triggerReason <i>RemoteStart</i> idToken.idToken <PspRef> <idToken.idToken from step 1> idToken.idToken idToken.type <i>DirectPayment</i> transactionInfo.transactionId <i><transactionId></i> transactionInfo.remoteStartId <i><remoteStartId></i> meterValue[0].timestamp <i><not omitted></i> meterValue[0].sampledValue[0].value <i>10000</i> meterValue[0].sampledValue[0].context <i>Transaction.Begin</i>	4. The CSMS responds with a TransactionEventResponse
...	

Tool validations
* Step 1: Message RequestStartTransactionRequest idToken.idToken must be <PspRef> NOT be omitted ...

6.3.11. Page 1046/1048/1049 - (2026-02) - TC_C_131_CSMS/TC_C_132_CSMS/TC_C_133_CSMS - Sharedsecret configuration variable added

The hardcoded value used for WebPaymentsCtrlr.SharedSecret ("mysharedsecret") has been replaced with a configurable variable <Configured sharedsecret>.

6.3.12. Page 1094/1095 - (2026-02) - TC_E_110_CSMS - Added missing Authorization request to initiate transaction

An explicit authorization request was introduced for improved clarity, and the configured energy limit was standardized to a fixed value.

Test case name	Transactions with fixed cost, energy or time - CSMS specifies energy limit
Test case Id	TC_E_110_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Test System sends an AuthorizeRequest With idToken.idToken <Configured valid_idtoken_idtoken> idToken.type <Configured valid_idtoken_type>	2. The CSMS responds with an AuthorizeResponse
Manual action: In CSMS, configure energy limit of 10000 (Wh) for this transaction	
3. The Test System sends a TransactionEventRequest with With eventType is Started triggerReason is ChargingStateChanged idToken.idToken <Configured valid_idtoken_idtoken> idToken.type <Configured valid_idtoken_type> evse.id is <Configured evseld> evse.connectorId is <Configured connectorId> transactionInfo.transactionId is <transactionId> transactionInfo.chargingState is Charging transactionInfo.transactionLimit <omitted>	4. The CSMS responds with a TransactionEventResponse
5. The Test System sends a TransactionEventRequest with triggerReason is LimitSet transactionInfo.transactionLimit.maxEnergy is <Configured energy limit> 10000	6. The CSMS responds with TransactionEventResponse
7. The Test System sends a TransactionEventRequest with eventType is Updated triggerReason is EnergyLimitReached evse.id is <Configured evseld> evse.connectorId is <Configured connectorId> transactionInfo.transactionId is <transactionId> transactionInfo.transactionLimit.maxEnergy <Configured energy limit> 10000 transactionInfo.transactionLimit.maxTime <omitted> transactionInfo.transactionLimit.maxCost <omitted> transactionInfo.chargingState SuspendedEVSE	8. The CSMS responds with a TransactionEventResponse

Tool validations
* Step 2 4: Message: TransactionEventResponse idTokenInfo.status must be <i>Accepted</i> transactionLimit.maxEnergy must be <i><Configured energy limit></i> 10000 transactionLimit.maxTime must be <i><omitted></i> transactionLimit.maxCost must be <i><omitted></i>
* Step 4 6: Message: TransactionEventResponse transactionLimit must be <i><omitted></i>
Post scenario validations: N/a

6.3.13. Page 1096/1097 - (2026-02) - TC_E_111_CSMS - Added missing Authorization request to initiate transaction

An explicit authorization request was introduced for improved clarity, and the configured time limit was standardized to a fixed value.

Test case name	Transactions with fixed cost, energy or time - CSMS specifies time limit
Test case Id	TC_E_111_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
1. The Test System sends an AuthorizeRequest With idToken.idToken <i><Configured valid_idtoken_idtoken></i> idToken.type <i><Configured valid_idtoken_type></i>	2. The CSMS responds with an AuthorizeResponse
Manual action: In CSMS, configure time limit of duration <i><Configured time limit></i> 120 for this transaction	
1. 3. The Test System sends a TransactionEventRequest with With eventType is <i>Started</i> triggerReason is <i>ChargingStateChanged</i> idToken.idToken <i><Configured valid_idtoken_idtoken></i> idToken.type <i><Configured valid_idtoken_type></i> evse.id is <i><Configured evseld></i> evse.connectorId is <i><Configured connectorId></i> transactionInfo.transactionId is <i><transactionId></i> transactionInfo.chargingState is <i>Charging</i> transactionInfo.transactionLimit <i><omitted></i>	2. 4. The CSMS responds with a TransactionEventResponse
3. 5. The Test System sends a TransactionEventRequest with eventType is <i>Updated</i> triggerReason is <i>LimitSet</i> evse.id is <i><Configured evseld></i> evse.connectorId is <i><Configured connectorId></i> transactionInfo.transactionId is <i><transactionId></i> transactionInfo.transactionLimit.maxEnergy <i><omitted></i> transactionInfo.transactionLimit.maxTime <i><Configured time limit></i> 120 transactionInfo.transactionLimit.maxCost <i><omitted></i>	4. 6. The CSMS responds with a TransactionEventResponse

Tool validations
* Step 2 4: Message: TransactionEventResponse idTokenInfo.status must be <i>Accepted</i> transactionLimit.maxEnergy must be <i><omitted></i> transactionLimit.maxTime must be <i><Configured time limit></i> 120 transactionLimit.maxCost must be <i><omitted></i> * Step 4 6: Message: TransactionEventResponse transactionLimit must be <i><omitted></i>
Post scenario validations: N/a

6.3.14. Page 1333 - (2026-02) - TC_O_101_CSMS - Updated tool validations

Tool validations
* Step 1: Message SetDisplayMessageRequest - message.id <i><Generated Id></i> - message.priority <i><Configured Priority></i> - message.message.language <i><Not omitted default language></i> <Not omitted> - message.message.content <i><Not omitted default language content></i> - message.messageExtra[0].language <i><Not omitted other language></i> <Different from message.language> - message.messageExtra[0].content <i><Not omitted other language content></i>

6.3.15. Page 1210 - (2026-02) - TC_K_109_CSMS - Additional test case steps added

The main test scenario was updated to include an initial SetVariablesRequest for MaxExternalConstraintsId to ensure the CSMS is accurately tested for updating the chargingProfileId.

Test case name	EMS Control - Set Charging Profile - MaxExternalConstraintsId
Test case Id	TC_K_109_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
2. The Test System responds with a GetBaseReportResponse with status <i>Accepted</i>	1. The CSMS sends a GetBaseReportRequest
3. The Test System sends a NotifyReportRequest containing requestId <i><requestId></i> reportData[0].component.name <i>SmartChargingCtrlr</i> reportData[0].variable.name <i>MaxExternalConstraintsId</i> reportData[0].variableAttribute.value <i>2147400000</i>	4. The CSMS responds with a NotifyReportResponse
Manual Action : Configure variable MaxExternalConstraintsId on the CSMS	
2. The Test System responds with SetVariablesResponse	1. The CSMS sends SetVariablesRequest with: - variable.name <i>MaxExternalConstraintsId</i> - component.name <i>SmartChargingCtrlr</i> - attributeValue 10
Manual Action : Request the CSMS to send a TxDefaultProfile Charging Profile	

Main (Test scenario)	
6. 4. The Test System responds with a SetChargingProfileResponse with status <i>Accepted</i>	5. 3. The CSMS sends a SetChargingProfileRequest
Manual Action : Request the CSMS to clear the charging profile with <chargingProfileId from step 3>	
6. The Test System responds with a ClearChargingProfileResponse with status <i>Accepted</i>	5. The CSMS sends a ClearChargingProfileRequest with chargingProfileId <chargingProfileId from step 3> AND chargingProfileCriteria <i>omit</i>
Manual Action : Configure variable <i>MaxExternalConstraintsId</i> on the CSMS	
8. The Test System responds with SetVariablesResponse	7. The CSMS sends SetVariablesRequest with: - variable.name <i>MaxExternalConstraintsId</i> - component.name <i>SmartChargingCtrlr</i> - attributeValue <i>2147400000</i>
Manual Action : Request the CSMS to send a <i>TxDefaultProfile Charging Profile</i>	
10. The Test System responds with a SetChargingProfileResponse with status <i>Accepted</i>	9. The CSMS sends a SetChargingProfileRequest

Tool validations
* Step 3: Message SetChargingProfileRequest chargingProfile.id must be greater than <i>10</i>
* Step 5: Message ClearChargingProfileRequest chargingProfile.id must be <chargingProfileId from step 1>
* Step 9: Message SetChargingProfileRequest chargingProfile.id must be greater than <i>2147400000</i>
Post scenario validations: - N/a

6.3.16. Page 1304-1306 - (2026-02) - TC_N_102_CSMS/TC_N_103_CSMS - added without userinfo

The first attempt needs to be without to check it is not accepted.

Test case Id	TC_N_102_CSMS / TC_N_103_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
3. The Test System uploads log file to <GetLogRequest.log.remoteLocation of step 1 without userinfo > using HTTP POST	4. The CSMS responds using HTTP
...	

6.3.17. Page 1336 - (2026-02) - TC_Q_102_CSMS - Test case removed

This testcases has been removed, because as a CSMS cannot be forced to omit the energyModes. In addition, there is also no added benefit to test this.

6.3.18. Page 1309 - (2026-02) - TC_N_105_CSMS - Install usable monitor and send a NotifyPeriodicEventStream

Test case name	Set Variable Monitoring - Set Frequent Periodic Variable Monitoring - Periodic
Test case Id	TC_N_105_CSMS
...	

Before (Preparations)
Configuration State: CSMS must be configured to accept opening the periodic event stream. N/a
Memory State: N/a
Reusable State: N/a

Main (Test scenario)	
Charging Station	CSMS
Manual Action: Request CSMS to install a monitor on: component.name EVSE EVSE.id <Configured evseld> variable.name Power type Periodic severity 8 value 5 periodicEventStream.interval 5 periodicEventStream.values 10	
2. Test System responds with: Message: SetVariableMonitoringResponse with setMonitoringResult[0].status = Accepted	1. CSMS sends a SetVariableMonitoringRequest
1. 3. Test System sends a OpenPeriodicEventStreamRequest with constantStreamData.id 2 constantStreamData.variableMonitoringId 3 <monitor id from step 1> constantStreamData.params.interval 40 5 constantStreamData.params.values 30 10	2. 4. CSMS responds with a OpenPeriodicEventStreamResponse
Note: The Test System waits 5 seconds.	
5. Test System sends a NotifyPeriodicEventStream using RFC framework SEND	
Manual Action: Request the CSMS to reconfigure the periodic event stream to an interval of 15s and values 45	
4. 7. Test System responds with a AdjustPeriodicEventStreamResponse with status Accepted	3. 6. CSMS sends a AdjustPeriodicEventStreamRequest
Note: The Test System waits 10 seconds.	
5. 8. Test System sends a NotifyPeriodicEventStream using RFC framework SEND	
6. 9. Test System sends a ClosePeriodicEventStreamRequest with constantStreamData.id 2	7. 10. CSMS responds with a ClosePeriodicEventStreamResponse
Note: The Test System waits 10 seconds, before ending the test case.	

Tool validations
* Step 1: Message SetVariableMonitoringRequest with: - setMonitoringData[0].type = <i>Periodic</i>, - setMonitoringData[0].severity = 8, - setMonitoringData[0].component.name = "EVSE" - setMonitoringData[0].component.evse.id = <Configured evseld> - setMonitoringData[0].variable.name = <i>Power</i> - setMonitoringData[0].value = 5 - setMonitoringData[0].periodicEventStream.interval 5 - setMonitoringData[0].periodicEventStream.values 10 * Step 2: Step 4: Message OpenPeriodicEventStreamResponse - status must be <i>Accepted</i> * Step 3: Step 6: Message AdjustPeriodicEventStreamRequest id must be 2 params.interval must be 45 10 params.values must be 45 20

6.3.19. Page 1339/1340 - (2026-02) - TC_Q_107_CSMS - Test case updated

Test case updated to reflect current use cases and functional requirements.

Test case name	V2X Authorisation - ISO15118-20 - Charging only (V2X control) before starting V2X
Test case Id	TC_Q_107_CSMS
...	

Before (Preparations)
Configuration State: - CSMS is configured to start transactions first in a non-V2X mode before upgrading to V2X N/a
Memory State: N/a
Reusable State(s): State is ISO15118-20 transaction just started exposing: - <allowedEnergyTransferModes> - <transactionId>

Main (Test scenario)	
Charging Station	CSMS
Agree on energy transfer mode	
1. The Test System sends a NotifyEVChargingNeedsRequest with evseld <Configured evseld> chargingNeeds.requestedEnergyTransfer one of non-bidirectional mode in <allowedEnergyTransferModes> received in AuthorizeResponse or if omitted <i>AC_three_phase</i> chargingNeeds.availableEnergyTransfer [<i>AC_single_phase</i>, <i>AC_two_phase</i>, <i>AC_three_phase</i>, <i>AC_BPT</i>, <i>AC_BPT_DER</i>, <i>AC_DER</i>] chargingNeeds.controlMode <i>ScheduledControl</i> chargingNeeds.v2xChargingParameters.maxChargePower 1234W chargingNeeds.v2xChargingParameters.maxDischargePower 1234W	2. The CSMS responds with a NotifyEVChargingNeedsResponse

Main (Test scenario)	
Setting charging profile	
4. The Test System responds with a SetChargingProfileResponse with status <i>Accepted</i>	3. The CSMS sends a SetChargingProfileRequest
<u>Manual Action:</u> <i>Trigger CSMS to allow V2X after initial start</i>	
6. The Test System sends a NotifyAllowedEnergyTransferResponse with status <i>Accepted</i>	5. The CSMS sends a NotifyAllowedEnergyTransferRequest
7. The Test System sends a NotifyEVChargingNeedsRequest with evseld <i><Configured evseld></i> chargingNeeds.requestedEnergyTransfer one of <i>bidirectional</i> mode in <i><allowedEnergyTransferModes></i> received in NotifyAllowedEnergyTransferRequest chargingNeeds.availableEnergyTransfer <i>[AC_single_phase, AC_two_phase, AC_three_phase, AC_BPT, AC_BPT_DER, AC_DER]</i> chargingNeeds.controlMode <i>ScheduledControl</i> chargingNeeds.v2xChargingParameters.maxChargePower <i>1234W</i> chargingNeeds.v2xChargingParameters.maxDischargePower <i>1234W</i>	8. The CSMS responds with a NotifyEVChargingNeedsResponse
Setting charging profile	
10. The Test System responds with a SetChargingProfileResponse with status <i>Accepted</i>	9. The CSMS sends a SetChargingProfileRequest
11. The Test System sends a TransactionEventRequest with eventType <i>Updated</i> triggerReason <i>OperationModeChanged</i> transactionInfo.transactionId <i><transactionId></i> transactionInfo.operationMode <i><Operation mode from step 9></i>	12. The CSMS responds with a TransactionEventResponse

Tool validations
* Step 2: Message: NotifyEVChargingNeedsResponse - status must be <i>Accepted</i> or <i>Processing</i> * Step 3: Message: SetChargingProfileRequest - evseld must be <i><Configured evseld></i> - chargingProfile.transactionId must be <i><transactionId></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].operationMode must be <i>ChargingOnly</i> <i><omitted></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].Limit must NOT be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].dischargeLimit must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].dischargeLimit_L2 must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].dischargeLimit_L3 must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].setpoint must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].setpoint_L2 must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].setpoint_L3 must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].setPointReactive must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].setPointReactive_L2 must be <i><omitted></i> <i><empty></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].setPointReactive_L3 must be <i><omitted></i> <i><empty></i> * Step 5: Message: NotifyAllowedEnergyTransferRequest - transactionId must be <i><transactionId></i> - allowedEnergyTransfer must contain one of <i>[DC_BPT, AC_BPT, AC_BPT_DER]</i> * Step 7: Message: NotifyEVChargingNeedsResponse - status must be <i>Accepted</i> or <i>Processing</i> * Step 9: Message: SetChargingProfileRequest - evseld must be <i><Configured evseld></i> - chargingProfile.transactionId must be <i><transactionId></i> - chargingProfile.chargingSchedule[*].chargingSchedulePeriod[*].operationMode must NOT be <i>ChargingOnly</i> <i><empty></i>
Post scenario validations: N/a

6.3.20. Page 1359 - (2026-02) - TC_R_107_CSMS - Removed NotifyDERStartStopRequest and added ReportDERControlRequest

Test case name	Configure DER control settings at CS
Test case Id	TC_R_107_CSMS
...	

Main (Test scenario)	
Charging Station	CSMS
...	
<u>Manual action:</u> trigger CSMS to send a GetDERControlRequest with controlType VoltVar	
20. The Test System responds with a GetDERControlResponse with status NotFound	19. The CSMS sends a GetDERControlRequest
<u>Manual action:</u> trigger CSMS to send a GetDERControlRequest with controlId from step 7	
22. The Test System responds with a GetDERControlResponse with status Accepted	21. The CSMS sends a GetDERControlRequest

Main (Test scenario)	
23. The Test System sends a NotifyDERStartStopRequest with controlId <SetDERControlRequest.controlId of step 11> started = true timestamp <Current date/time> supersededId <SetDERControlRequest.controlId of step 9>	24. The CSMS responds with a NotifyDERStartStopResponse
23. The Test System sends a ReportDERControlRequest with - curve : - curve.id = "freqwatt_1" - curve.curveType = "FreqWatt" - curve.isDefault = false - curve.isSuperseded = false - curve.curve.priority = 4 - curve.curve.yUnit = "PctMaxW" - curve.curve.curvedata[0].x = 49 - curve.curve.curvedata[0].y = 75 - curve.curve.curvedata[1].x = 49.5 - curve.curve.curvedata[1].y = 90 - curve.curve.curvedata[2].x = 50.5 - curve.curve.curvedata[2].y = 100 - curve.curve.curvedata[3].x = 51 - curve.curve.curvedata[3].y = 100 - curve.curve.startTime = <current time> - curve.curve.duration = 900	24. The CSMS sends a ReportDERControlResponse

Tool validations
... * Step 21: Message: GetDERControlRequest - controlId must be <controlId from step 7>
Post scenario validations: N/a

6.3.21. Page 1365 - 1367 - (2026-02) - TC_S_103_CSMS - Test case updated

Test case name	Battery Swap - Remote Start - enough batteries available
Test case Id	TC_S_103_CSMS
Prerequisite(s)	Charging Station supports battery swapping. There are enough batteries available for swapping. BatterySwapCtrlr.Idtoken is <not set>

Before (Preparations)
Configuration State: BatterySwapCtrlr.Idtoken is <not set> N/a
Memory State: N/a
Reusable State(s): N/a

Main (Test scenario)	
Charging Station	CSMS

Main (Test scenario)	
<u>Manual Action:</u> Let CSMS trigger a request for swapping batteries	
2. The Test System responds with a RequestBatterySwapResponse with status <i>Accepted</i>	1. The CSMS sends a RequestBatterySwapRequest
<u>Note:</u> Set of two (virtual) batteries are inserted at this point in scenario.	
3. The Test System notifies the CSMS about the status change of the slot. connectorStatus <i>Occupied</i>	4. The CSMS responds accordingly.
5. The Test System sends a TransactionEventRequest with eventType <i>Started</i> triggerReason <i>CablePluggedIn</i> idToken.idToken "" idToken.type <i>NoAuthorization</i> evse <Configured evseld> <id> evse.connectorId 1 transactionInfo.transactionId 111-222-333-444-3 <transactionId> transactionInfo.chargingState <i>EVConnected</i>	6. The CSMS responds with a TransactionEventResponse
7. The Test System notifies the CSMS about the status change of the slot.	8. The CSMS responds accordingly.
9. The Test System sends a TransactionEventRequest with eventType <i>Started</i> triggerReason <i>CablePluggedIn</i> idToken.idToken "" idToken.type <i>NoAuthorization</i> evse <Configured evseld2> evse.connectorId 1 transactionInfo.transactionId 111-222-333-444-4 transactionInfo.chargingState <i>EVConnected</i>	10. The CSMS responds with a TransactionEventResponse
<u>Note:</u> Steps 3 through 6 are repeated for each battery inserted	
11. 7. The Test System sends a BatterySwapRequest with eventType <i>BatteryIn</i> requestId <requestId> idToken.idToken <Configured valid_idtoken_idtoken> batteryData[0].evseld <Configured evseld> 1 batteryData[0].serialNumber 1234 batteryData[0].soC 23 batteryData[0].soH 85 batteryData[1].evseld <Configured evseld2> 2 batteryData[1].serialNumber 5678 batteryData[1].soC 45 batteryData[1].soH 87	12. 8. The CSMS responds with a BatterySwapResponse
<u>Note:</u> Set of two batteries are extracted at this point in scenario.	
13. The Charging Station sends a TransactionEventRequest with eventType <i>Ended</i> triggerReason <i>EnergyLimitReached</i> idToken.idToken "" idToken.type <i>NoAuthorization</i> transactionInfo.transactionId 111-222-333-444-1 transactionInfo.chargingState <i>Idle</i> transactionInfo.stoppedReason <i>EVDisconnected</i>	14. The Test System responds with a TransactionEventResponse

Main (Test scenario)	
15. 9. The Test System notifies the CSMS about the status change of the slot. connectorStatus <i>Available</i>	16. 10. The CSMS responds accordingly.
17. 11. The Charging Station sends a TransactionEventRequest with eventType <i>Ended</i> triggerReason <i>EnergyLimitReached</i> EVCommunicationLost idToken.idToken "" idToken.type <i>NoAuthorization</i> transactionInfo.transactionId <i>111-222-333-444-2</i> <transactionId> transactionInfo.chargingState <i>Idle</i> transactionInfo.stoppedReason <i>EVDisconnected</i>	18. 12. The Test System responds with a TransactionEventResponse
19. The Test System notifies the CSMS about the status change of the slot.	20. The CSMS responds accordingly.
Note: Steps 9 through 12 are repeated for each battery taken out	
21. 13. The Test System sends a BatterySwapRequest with eventType <i>BatteryOut</i> requestId <i><requestId></i> idToken.idToken <i><Configured valid_idtoken_idtoken></i> batteryData[0].evseld <i><Configured evseld3></i> 3 batteryData[0].serialNumber <i>4321</i> batteryData[0].soC <i>80</i> batteryData[0].soH <i>95</i> batteryData[1].evseld <i><Configured evseld4></i> 4 batteryData[1].serialNumber <i>8765</i> batteryData[1].soC <i>85</i> batteryData[1].soH <i>78</i>	22. 14. The CSMS responds with a BatterySwapResponse
Note: Steps 13 and 14 are repeated for each battery taken out	

Tool validations
* Step 1: Message: RequestBatterySwapResponse RequestBatterySwapRequest - requestId must be <i><not omitted></i> - idToken.idToken must be <i><Configured valid_idtoken_idtoken></i> * Step 6: Message: TransactionEventResponse - idTokenInfo.status must be <i>Accepted</i> * Step 10: Message: TransactionEventResponse - idTokenInfo.status must be <i>Accepted</i> * Step 12: Message: BatterySwapResponse * Step 14: Step 12: Message: TransactionEventResponse - idTokenInfo.status must be <i>Accepted</i> * Step 18: Message: TransactionEventResponse - idTokenInfo.status must be <i>Accepted</i> * Step 22: Message: BatterySwapResponse

6.3.22. Page 1383 - (2026-04) - Additional validation added for certificateStatus

ISO15118-20 transaction just started

The **certificateStatus** should be included in **AuthorizeResponse** when certificate or iso15118CertificateHashData is included in the **AuthorizeRequest**.

State	ISO15118-20 transaction just started
System under test	CSMS
...	

Tool validations
* Step 2: Message: AuthorizeResponse - idTokenInfo.status must be <i>Accepted</i> - certificateStatus must be <i>Accepted</i> * Step 4: Message: TransactionEventResponse - idTokenInfo.status must be <i>Accepted</i> Post scenario validations: N/a