

OCPP 2.0.1

Part 2 - Appendices

v1.3 FINAL, 2022-12-15

# Table of Contents

[Disclaimer](#_bookmark0) [1](#_bookmark0)

[Version History](#_bookmark1) [2](#_bookmark1)

1. [Security Events](#_bookmark2) [3](#_bookmark2)
2. [Standardized Units of Measure](#_bookmark3) [4](#_bookmark3)
3. [Standardized Components](#_bookmark4) [5](#_bookmark4)
   1. [Controller Components](#_bookmark5) [5](#_bookmark5)
   2. [Physical Components](#_bookmark6) [13](#_bookmark6)
   3. [Summary List of Standardized Components](#_bookmark7) [29](#_bookmark7)
4. [Standardized Variables](#_bookmark8) [32](#_bookmark8)
5. [Reason Codes](#_bookmark9) [36](#_bookmark9)

# Disclaimer

Copyright © 2010 – 2022 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

|  |  |  |  |
| --- | --- | --- | --- |
| **Appendix version** | **Date** | **OCPP Version** | **Description** |
| 1.3 | 2022-12-15 | OCPP 2.0.1 | Appendix version for Errata 2 (2022)  Updated parts are marked with "*(Updated in v1.3)*". |
| 1.2 | 2021-10-01 | OCPP 2.0.1 | Appendix version for Errata 1 (2021)  Appendix 3: Updated components are marked with "*(Updated in v1.2)*".  Appendix 3: Added ConnectedEV component for info from ISO15118 and CHAdeMO.  Appendix 5: Added reason MissingDeviceModelInfo |
| 1.1 | 2020-03-23 | OCPP 2.0.1 | Update for OCPP 2.0.1 |
| 1.0 | 2018-04-11 | OCPP 2.0 | First release of this Appendix for OCPP 2.0 |

**Appendix 1. Security Events**

The table below provides a list of security events. Security events that are implemented SHALL be stored at the security log and security events that are implemented and marked as critical SHALL also be pushed to the CSMS.

This is a non-exhaustive list of security events, when a security event matches the *description* of one of the Security Events in this section, for interoperability reasons, the Security Event from this section SHALL be used, instead of adding a new (proprietary) Security Event. Some security events like; *InvalidCsmsCertificate*, *InvalidChargingStationCertificate*, etc. are mandatory to be implemented. Please refer to Part 2 - Specification for which security events are mandatory to be implemented.

*(Updated in v1.3)*

|  |  |  |
| --- | --- | --- |
| **Security Event** | **Description** | **Critical** |
| FirmwareUpdated | The Charging Station firmware is updated | Yes |
| FailedToAuthenticateAtCsms | The authentication credentials provided by the Charging Station were rejected by the CSMS | No |
| CsmsFailedToAuthenticate | The authentication credentials provided by the CSMS were rejected by the Charging Station | No |
| SettingSystemTime | The system time on the Charging Station was changed more than  ClockCtrlr.TimeAdjustmentReportingThreshold seconds | Yes |
| StartupOfTheDevice | The Charging Station has booted | Yes |
| ResetOrReboot | The Charging Station was rebooted or reset | Yes |
| SecurityLogWasCleared | The security log was cleared | Yes |
| ReconfigurationOfSecurityParameters | Security parameters, such as keys or the security profile used, were changed | No |
| MemoryExhaustion | The Flash or RAM memory of the Charging Station is getting full | Yes |
| InvalidMessages | The Charging Station has received messages that are not valid OCPP messages, if signed messages, signage invalid/incorrect | No |
| AttemptedReplayAttacks | The Charging Station has received a replayed message (other than the CSMS trying to resend a message because it there was for example a network problem) | No |
| TamperDetectionActivated | The physical tamper detection sensor was triggered | Yes |
| InvalidFirmwareSignature | The firmware signature is not valid | No |
| InvalidFirmwareSigningCertificate | The certificate used to verify the firmware signature is not valid | No |
| InvalidCsmsCertificate | The certificate that the CSMS uses was not valid or could not be verified | Yes |
| InvalidChargingStationCertificate | The certificate sent to the Charging Station using the CertificateSignedRequest message is not a valid certificate | Yes |
| InvalidTLSVersion | The TLS version used by the CSMS is lower than 1.2 and is not allowed by the security specification | Yes |
| InvalidTLSCipherSuite | The CSMS did only allow connections using TLS cipher suites that are not allowed by the security specification | Yes |
| MaintenanceLoginAccepted | Successful login to the local maintenance interface. It is recommended to include information like the user identification and the origin of the login attempt, which can be an ip-address or a touch screen for example, to the techInfo field. For this the following format is strongly recommended: "{\"user\": \"…\", \"origin\": \"…\"}" | Yes |
| MaintenanceLoginFailed | Failed login attempt to the local maintenance interface. It is recommended to include information like the user identification and the origin of the login attempt, which can be an ip-address or a touch screen for example, to the techInfo field. For this the following format is strongly recommended: "{\"user\": \"…\", \"origin\": \"…\"}" | Yes |

# Appendix 2. Standardized Units of Measure

The standardized values for Unit of Measure. Default value of "unit" is always "Wh".

|  |  |
| --- | --- |
| **Value** | **Description** |
| **A** | Amperes (current) |
| **ASU** | Arbitrary Strength Unit (Signal Strength) |
| **B** | Bytes |
| **Celsius** | Degrees (temperature) |
| **dB** | Decibel (for example Signal Strength) |
| **dBm** | Power relative to 1mW (10log(P/1mW)) |
| **Deg** | Degrees (angle/rotation) |
| **Fahrenheit** | Degrees (temperature) |
| **Hz** | Hertz (frequency) |
| **K** | Degrees Kelvin (temperature) |
| **lx** | Lux (Light Intensity) |
| **m** | Meter (length) |
| **ms2** | m/s2 (Acceleration) |
| **N** | Newtons (Force) |
| **Ohm** | Ohm (Impedance) |
| **kPa** | kiloPascal (Pressure) |
| **Percent** | Percentage |
| **RH** | Relative Humidity% |
| **RPM** | Revolutions per Minute |
| **s** | Seconds (Time) |
| **V** | Voltage (DC or r.m.s. AC) |
| **VA** | Volt-Ampere (apparent power) |
| **kVA** | kiloVolt-Ampere (apparent power) |
| **VAh** | Volt-Ampere-hours (apparent energy) |
| **kVAh** | kiloVolt-Ampere-hours (apparent energy) |
| **var** | vars (reactive power) |
| **kvar** | kilovars (reactive power) |
| **varh** | var-hours (reactive energy) |
| **kvarh** | kilovar-hours (reactive energy) |
| **W** | Watts (power) |
| **kW** | kilowatts (power) |
| **Wh** | Watt-hours (energy). Default |
| **kWh** | kilowatt-hours (energy) |

# Appendix 3. Standardized Components

This appendix provides a list of all standardized component names for OCPP 2.0.1 for controller components and for physical components. A summary table listing just all components without variables is provided at the end of this appendix in [Summary List](#_bookmark7) [of Standardized Components](#_bookmark7).

## Controller Components

This is the list of Standardized Controller Components for OCPP 2.0.1. and typical Variables that might be associated with them.

IMPORTANT

This list does not imply that these Components are required, nor does it imply that the listed Variables are required for a Component or no other Variables are allowed to be associated with a Component.

* + 1. **AlignedDataCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the reporting of clock-aligned meter data. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | If this variable reports a value of true, Aligned Data is enabled. |
| Available | boolean | If this variable reports a value of true, Aligned Data is supported. |
| Interval | integer | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the MeterValuesRequest message. |
| Measurands | MemberList | Clock-aligned measurand(s) to be included in MeterValuesRequest, every AlignedDataInterval seconds. |
| SendDuringIdle | boolean | If set to true, the Charging Station SHALL not send clock aligned meter values when a transaction is ongoing. |
| SignReadings | boolean | If set to true, the Charging Station SHALL include signed meter values in the TransactionEventRequest to the CSMS. |
| TxEndedInterval | integer | Size (in seconds) of the clock-aligned data interval, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. |
| TxEndedMeasurands | MemberList | Clock-aligned periodic measurand(s) to be included in the meterValues element of TransactionEventRequest (eventType = Ended) for every TxEndedAlignedDataInterval of the transaction. |

* + 1. **AuthCtrlr** *(Updated in v1.2)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the use of authorization for Charging Station use. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | If set to *false*, then no authorization is done before starting a transaction or when reading an idToken. If an idToken was provided, then it will be put in the *idToken* field of the TransactionEventRequest. If no idToken was provided, then *idToken* in TransactionEventRequest will be left empty and type is set to NoAuthorization. |
| AdditionalInfoItemsPerMessag e | integer | Maximum number of AdditionalInfo items that can be sent in one message. |
| AuthorizeRemoteStart | boolean | Whether a remote request to start a transaction in the form of RequestStartTransactionRequest message should be authorized beforehand like a local action to start a transaction. |
| DisableRemoteAuthorization | boolean | When set to *true* this instructs the Charging Station to not issue any AuthorizationRequests, but only use Authorization Cache and Local Authorization List to determine validity of idTokens. |
| LocalAuthorizeOffline | boolean | Whether the Charging Station, when Offline, will start a transaction for locally- authorized identifiers. |
| LocalPreAuthorize | boolean | Whether the Charging Station, when online, will start a transaction for locally- authorized identifiers without waiting for or requesting an AuthorizeResponse from the CSMS. |

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| MasterPassGroupId | string | IdTokens that have this id as groupId belong to the Master Pass Group. |
| OfflineTxForUnknownIdEnabled | boolean | If this key exists, the Charging Station supports Unknown Offline Authorization. |

* + 1. **AuthCacheCtrlr** *(Updated in v1.2)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the use of a local cache for authorization for Charging Station use. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | If this variable exists, the Charging Station supports an Authorization Cache. |
| Available | boolean | If this variable reports a value of true, Authorization Cache is supported. |
| LifeTime | integer | Indicates in seconds how long it takes until a token expires in the authorization cache since it is last used. |
| Policy | OptionList | Cache Entry Replacement Policy: (LRU,LFU) LeastRecentlyUsed or LeastFrequentlyUsed. Allowed values: LRU, LFU. |
| DisablePostAuthorize | boolean | When set to *true* this variable disables the behavior to request authorization for an idToken that is stored in the cache with a status other than Accepted, as stated in C10.FR.03 and C12.FR.05. |

* + 1. **CHAdeMOCtrlr** *(Updated in v1.2)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| A CHAdeMO Controller component communicates with an EV using the wired CANbus protocol to exchange information and control charging using the CHAdeMO protocol | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | CHAdeMO controller enabled |
| Active | boolean | Connected |
| Complete | boolean | Protocol session ended normally |
| Tripped | boolean | CHAdeMO protocol terminated abnormally |
| Problem | boolean | CHAdeMO controller fault |
| SelftestActive(Set) | boolean | Start self-test by setting to true |
| SelftestActive | boolean | Self-test running when reported as true |
| ***Specific CHAdeMO interface data from vehicle:*** | | |
| CHAdeMOProtocolNumber | integer | CHAdeMO protocol number (H'102.0) |
| VehicleStatus | boolean | Vehicle status (H'102.5.3) |
| DynamicControl | boolean | Vehicle is compatible with dynamic control (H'110.0.0) |
| HighCurrentControl | boolean | Vehicle is compatible with high current control (H'110.0.1) |
| HighVoltageControl | boolean | Vehicle is compatible with high voltage control (H'110.1.2) |
| AutoManufacturerCode | integer | Auto manufacturer code (H'700.0)  *A single byte manufacturer code assigned by CHAdeMO association* |

* + 1. **ClockCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Provides a means to configure management of time tracking by Charging Station. | | |
| **Variables** | **Type** | **Description** |
| DateTime | dateTime | Contains the current date and time. |
| NtpServerUri | string | This contains the address of the NTP server. Multiple NTP servers can be configured as backups, etc. If the NTP client supports it, it can also connect to multiple NTP servers simultaneous to get a more reliable time source. Variable instance value is single digit NTP priority (1=highest). |

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| NtpSource | string | When an NTP client is implemented, this variable can be used to configure the client: Use the NTP server provided via DHCP, or use the manually configured NTP server. |
| TimeOffset | string | Configured local time offset in the format: "+01:00", "-02:00" etc. |
| NextTimeOffsetTransitionDateT ime | dateTime | Date time of the next time offset transition. |
| TimeSource | string | Via this variable, the Charging Station provides the CSMS with the option to configure a clock source, if more than 1 are implemented. |
| TimeZone | string | Configured current local time zone in the format: "Europe/Oslo", "Asia/Singapore" etc. |
| TimeAdjustmentReportingThres hold | integer | If set, then time adjustments with an absolute value in seconds larger than this need to be reported as a security event SettingSystemTime. |

* + 1. **CustomizationCtrlr** *(New in v1.2)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to custom vendor-specific implementations, using the DataTransfer message and CustomData extensions. | | |
| **Variables** | **Type** | **Description** |
| CustomImplementationEnabled | boolean | This standard configuration variable can be used to enable/disable custom implementations that the Charging Station supports. The instance name of the variable matches the *vendorId* of the customization in CustomData or DataTransfer messages. |

* + 1. **DeviceDataCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the exchange and storage of Charging Station Device Model data. | | |
| **Variables** | **Type** | **Description** |
| BytesPerMessage | integer | Message Size (in bytes) - maxLimit used to report constraint on message size. Which message is specified in the instance. |
| ItemsPerMessage | integer | Maximum number of entries that can be sent in one message. Which entries in which message is specified in the instance. |
| ValueSize | integer | Can be used to limit the following fields: SetVariableData.attributeValue, GetVariableResult.attributeValue, VariableAttribute.value, VariableCharacteristics.valueList and EventData.actualValue. |

* + 1. **DisplayMessageCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the display of messages to Charging Station users. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | Whether Display Message is enabled. |
| Available | boolean | Whether Display Message is supported. |
| DisplayMessages | integer | Amount of different messages that are currently configured in this Charging Station, via SetDisplayMessageRequest. |
| PersonalMessageSize | integer | Max size (in characters) of the personal message element of the IdTokenInfo data (0 specifies no personal data may be stored). |
| SupportedFormats | MemberList | List of message formats supported by this Charging Station. Possible values: See MessageFormatEnumType. |

* + 1. **ISO15118Ctrlr** *(Updated in v1.3)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Communicates with an EV to exchange information and control charging using the ISO 15118 protocol. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | ISO15118 controller enabled |
| Active | boolean | Connected |
| Tripped | boolean | ISO15118 communication session aborted |
| Complete | boolean | ISO15118 communication session ended |
| Problem | boolean | ISO15118 controller fault |
| SeccId | string | The name of the SECC in the string format as required by ISO 15118. |
| SelftestActive(Set) | boolean | Start self-test by setting to true |
| SelftestActive | boolean | Self-test running when reported as true |
| ContractValidationOffline | boolean | Supports validation of a contract certificate when offline |
| CentralContractValidationAllow ed | boolean | Contract certificates can be validated by the CSMS |
| PnCEnabled | boolean | If this variable is *true*, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is enabled.  If this variable is *false*, then ISO 15118 plug and charge as described by use case C07 - Authorization using Contract Certificates is disabled. |
| V2GCertificateInstallationEnabl ed | boolean | If this variable is *true*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is enabled.  If this variable is *false*, then ISO 15118 V2G Charging Station certificate installation as described by use case A02 - Update Charging Station Certificate by request of CSMS and A03 - Update Charging Station Certificate initiated by the Charging Station is disabled. |
| ContractCertificateInstallationE nabled | boolean | If this variable is *true*, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is enabled.  If this variable is *false*, then ISO 15118 contract certificate installation/update as described by use case M01 - Certificate installation EV and M02 - Certificate Update EV is disabled. |
| RequestMeteringReceipt | boolean | If this variable is *true*, then Charging Station shall request a metering receipt from EV before sending a fiscal meter value to CSMS. |
| OrganizationName | string | The organizationName of the CSO operating the charging station. It is used as the **organizationName** (O) of the SECC leaf certificate. Example: "John Doe Charging Services Ltd"  Note: This value will usually be identical to SecurityCtrlr.OrganizationName, but it does not have to be. |
| CountryName | string | The countryName of the SECC in the ISO 3166-1 format.  It is used as the **countryName** (C) of the SECC leaf certificate. Example: "DE" |
| ***Specific ISO15118 interface data from vehicle:*** | | |
| MaxScheduleEntries | integer | MaxEntriesSAScheduleType (15118-2)  or MaximumSupportingPoints (15118-20) *Number of allowed schedule periods* |
| RequestedEnergyTransferMode | OptionList | RequestedEnergyTransferMode  *"AC\_single\_phase\_core", "AC\_three\_phase\_core", "DC\_core, "DC\_extended", "DC\_combo\_core", "DC\_unique"* |

* + 1. **LocalAuthListCtrlr** *(Updated in v1.2)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the use of Local Authorization Lists for Charging Station use. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | Whether Local Authorization List is enabled. |
| Entries | integer | Amount of IdTokens currently in the Local Authorization List. The maxLimit of this variable SHALL be provided to report the maximum number of IdTokens that can be stored in the Local Authorization List. |

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Available | boolean | Whether Local Authorization List is supported. |
| ItemsPerMessage | integer | Maximum number of identifications that can be sent in a single SendLocalListRequest. |
| BytesPerMessage | integer | Message Size (in bytes) - puts a constraint on SendLocalListRequest message size. |
| Storage | integer | Indicates the number of bytes currently used by the Local Authorization List. MaxLimit indicates the maximum number of bytes that can be used by the Local Authorization List. |
| DisablePostAuthorize | boolean | When set to *true* this variable disables the behavior to request authorization for an idToken that is stored in the local authorization list with a status other than Accepted, as stated in C14.FR.03. |

* + 1. **MonitoringCtrlr** *(Updated in v1.3)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the exchange of monitoring event data. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | Whether Monitoring is enabled. |
| Available | boolean | Whether Monitoring is supported. |
| ItemsPerMessage | integer | Maximum number of items. |
| BytesPerMessage | integer | Message Size (in bytes) - puts constraint on message size. |
| MonitoringBase | optionList | Currently used MonitoringBase. (readonly) |
| MonitoringLevel | integer | Currently use MonitoringLevel (readonly) |
| OfflineQueuingSeverity | integer | When set and the Charging Station is offline, the Charging Station shall queue any notifyEventRequest messages triggered by a monitor with a severity number equal to or lower than the severity configured here. Value ranging from 0 (Emergency) to 9 (Debug). |
| ActiveMonitoringBase | OptionList | Shows the currently used MonitoringBase. Valid values according MonitoringBaseEnumType: All, FactoryDefault, HardwiredOnly. (readonly) |
| ActiveMonitoringLevel | integer | Shows the currently used MonitoringLevel. Valid values are severity levels of SetMonitoringLevelRequest: 0-9. (readonly) |

* + 1. **OCPPCommCtrlr** *(Updated in v1.3)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to information exchange between Charging Station and CSMS. | | |
| **Variables** | **Type** | **Description** |
| ActiveNetworkProfile | string | Indicates the configuration profile the station uses at that moment to connect to the network. |
| FileTransferProtocols | MemberList | List of supported file transfer protocols. |
| HeartbeatInterval | integer | Interval in seconds of inactivity (no OCPP exchanges) with CSMS after which the Charging Station should send HeartbeatRequest. |
| MessageAttempts | integer | How often the Charging Station should try to submit a TransactionEventRequest message when the CSMS fails to process it. |
| MessageAttemptInterval | integer | How long in seconds the Charging Station should wait before resubmitting a TransactionEventRequest message that the CSMS failed to process. |
| MessageTimeout | integer | Message timeout in seconds. The message timeout setting in a Charging Station can be configured in the messageTimeout field in the NetworkConnectionProfile. |
| MinimumStatusDuration | integer | Minimum duration that a Charging Station or EVSE status is stable before StatusNotificationRequest is sent to the CSMS. |
| NetworkConfigurationPriority | string | A comma separated ordered list of the priority of the possible Network Connection Profiles. |
| NetworkProfileConnectionAtte mpts | integer | Specifies the number of connection attempts the Charging Station executes before switching to a different profile. |

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| OfflineThreshold | integer | When the offline period in seconds of a Charging Station exceeds the OfflineThreshold it is recommended to send a StatusNotificationRequest for all its Connectors when the Charging Station is back online. |
| PublicKeyWithSignedMeterValu e | boolean | This Configuration Variable can be used to configure whether a public key needs to be sent with a signed meter value. |
| QueueAllMessages | boolean | When this variable is set to true, the Charging Station will queue all message until they are delivered to the CSMS. |
| RetryBackOffRepeatTimes | integer | When the Charging Station is reconnecting, after a connection loss, it will use this variable for the amount of times it will double the previous back-off time. |
| RetryBackOffRandomRange | integer | When the Charging Station is reconnecting, after a connection loss, it will use this variable as the maximum value for the random part of the back-off time. |
| RetryBackOffWaitMinimum | integer | When the Charging Station is reconnecting, after a connection loss, it will use this variable as the minimum back-off time, the first time it tries to reconnect. |
| UnlockOnEVSideDisconnect | boolean | When set to true, the Charging Station SHALL unlock the cable on the Charging Station side when the cable is unplugged at the EV. For an EVSE with only fixed cables, the mutability SHALL be ReadOnly and the actual value SHALL be false. For a charging station with fixed cables and sockets, the variable is only applicable to the sockets. |
| WebSocketPingInterval | integer | Number of seconds between pings. |
| FieldLength | integer | This variable is used to report the length of <field> in <message> when it is larger than the length that is defined in the standard OCPP message schema. |

* + 1. **ReservationCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to reservations. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | Whether Reservation is enabled. |
| Available | boolean | Whether Reservation is supported. |
| NonEvseSpecific | boolean | If this configuration variable is present and set to true: Charging Station supports Reservation without specifying an EVSE. |

* + 1. **SampledDataCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to the reporting of sampled meter data. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | If this variable reports a value of true, Sampled Data is enabled. |
| Available | boolean | If this variable reports a value of true, Sampled Data is supported. |
| SignReadings | boolean | If set to true, the Charging Station includes signed meter values in the MeterValuesRequest to the CSMS. |
| TxEndedMeasurands | MemberList | Sampled measurands to be included in the meterValues element of TransactionEventRequest (eventType = Ended), every TxEndedSampleInterval seconds from the start of the transaction. |
| TxEndedInterval | integer | Interval in seconds between sampling of metering (or other) data, intended to be transmitted in the TransactionEventRequest (eventType = Ended) message. |
| TxStartedMeasurands | MemberList | Sampled measurand(s) to be taken at the start of any transaction to be included in the meterValues field of the first TransactionEventRequest message send at the start of a transaction (eventType = Started). |
| TxUpdatedMeasurands | MemberList | Sampled measurands to be included in the meterValues element of every TransactionEventRequest (eventType = Updated), every SampledDataTxUpdatedInterval seconds from the start of the transaction. |
| TxUpdatedInterval | integer | Interval in seconds between sampling of metering (or other) data, intended to be transmitted via TransactionEventRequest (eventType = Updated) messages. |

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| RegisterValuesWithoutPhases | boolean | If this variable reports a value of *true*, then meter values of measurand Energy.Active.Import.Register will only report the total energy over all phases without reporting the individual phase values.  If this variable is absent or *false*, then the value for each phase is reported, possibly also with a total value (depending on the meter). |

* + 1. **SecurityCtrlr** *(Updated in v1.3)*

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to security of communications between Charging Station and CSMS. | | |
| **Variables** | **Type** | **Description** |
| BasicAuthPassword | passwordStrin g | The basic authentication password that is used for HTTP Basic Authentication. The datatype is passwordString consisting of minimum 16 and maximum 40 characters (alpha-numeric characters and the special characters allowed by passwordString). The password SHALL be sent as a UTF-8 encoded string (NOT encoded into octet string or base64). This configuration variable is write-only, so that it cannot be accidentally stored in plaintext by the CSMS when it reads out all configuration variables.  This configuration variable is required unless only "security profile 3 - TLS with client side certificates" is implemented. |
| Identity | identifierString | The Charging Station identity. identity is an identifierString string, so it SHALL only contain characters that are allowed for identifierString. Maximum length was chosen to ensure compatibility with EVSE ID from [EMI3] "Part 2: business objects". |
| OrganizationName | string | This configuration variable is used to set the organization name of the CSO or an organization trusted by the CSO. It is used to set the O (organizationName) RDN in the subject field of the client certificate. |
| CertSigningWaitMinimum | integer | Seconds to wait before generating another CSR in case CSMS does not return a signed certificate. |
| CertSigningRepeatTimes | integer | Number of times to resend a SignCertificateRequest when CSMS does nor return a signed certificate. |

* + 1. **SmartChargingCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to smart charging. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | Whether Smart Charging is enabled. |
| Available | boolean | Whether Smart Charging is supported. |
| ACPhaseSwitchingSupported | boolean | If defined and true, this EVSE supports the selection of which phase to use for 1 phase AC charging. |
| ProfileStackLevel | integer | Maximum acceptable value for *stackLevel* in a ChargingProfile. Since the lowest *stackLevel* is 0, this means that if SmartChargingCtrlr.ProfileStackLevel = 1, there can be at most 2 valid charging profiles per Charging Profile Purpose per EVSE. |
| RateUnit | MemberList | A list of supported quantities for use in a ChargingSchedule. Allowed values: 'A' and 'W'. |
| PeriodsPerSchedule | integer | Maximum number of periods that may be defined per ChargingSchedule. |
| ExternalControlSignalsEnabled | boolean | Indicates whether a Charging Station should respond to external control signals that influence charging. |
| NotifyChargingLimitWithSched ules | boolean | Indicates if the Charging Station should include the externally set charging limit/schedule in the message when it sends a NotifyChargingLimitRequest message. This might increase the data usage significantly, especially when an external system sends new profiles/limits with a short interval. Default is false when omitted. |
| Phases3to1 | boolean | If defined and true, this Charging Station supports switching from 3 to 1 phase during a transaction. |
| Entries | integer | Amount of Charging profiles currently installed on the Charging Station. MaxLimit used to limit number of Charging profiles installed at any time. |

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| LimitChangeSignificance | integer | If at the Charging Station side a change in the limit in a ChargingProfile is lower than this percentage, the Charging Station MAY skip sending a NotifyChargingLimitRequest or a TransactionEventRequest message to the CSMS. It is RECOMMENDED to set this key to a low value. See Smart Charging signals to a Charging Station from multiple actors. |

* + 1. **TariffCostCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to tariff and cost display. | | |
| **Variables** | **Type** | **Description** |
| Enabled | boolean | Whether Tariff/cost is enabled. |
| Available | boolean | Whether Tariff/cost is supported. |
| TariffFallbackMessage | string | Message (and/or tariff information) to be shown to an EV Driver when there is no driver specific tariff information available. |
| TotalCostFallbackMessage | string | Message to be shown to an EV Driver when the Charging Station cannot retrieve the cost for a transaction at the end of the transaction. |
| Currency | string | Currency used by this Charging Station in a ISO 4217 formatted currency code. |

* + 1. **TxCtrlr**

|  |  |  |
| --- | --- | --- |
| **Description** | | |
| Logical Component responsible for configuration relating to transaction characteristics and behaviour. | | |
| **Variables** | **Type** | **Description** |
| EVConnectionTimeOut | integer | Interval in seconds from between "starting" of a transaction until incipient transaction is automatically canceled, due to failure of EV driver to (correctly) insert the charging cable connector(s) into the appropriate socket(s). The Charging Station SHALL go back to the original state, probably: 'Available'. "Starting" might be the swiping of the RFID, pressing a start button, a RequestStartTransactionRequest being received etc. |
| TxBeforeAcceptedEnabled | boolean | With this configuration variable the Charging Station can be configured to allow charging before having received a BootNotificationResponse with RegistrationStatus: Accepted. See: Transactions before being accepted by a CSMS. |
| TxStartPoint | MemberList | Defines when the Charging Station starts a new transaction: first transactioneventRequest: eventType = Started.  When any event in the given list occurs, the Charging Station SHALL start a transaction The Charging Station SHALL only send the Started event once for every transaction.  It is advised to put all events that should be part of a transaction in the list, in case the start event never occurs. Because the possible events don’t always have to come in the same order it is possible to provide a list of events. Which ever comes first will then cause a transaction to be started.  For example: EVConnected, Authorized would mean that a transaction is started when an EV is detected (Cable is connected), or when an EV Driver swipes his RFID card en the CSMS successfully authorizes the ID for charging. |
| TxStopPoint | MemberList | Defines when the Charging Station ends a transaction: last transactioneventRequest: eventType = Ended.  When any event in the given list is no longer valid, the Charging Station SHALL end the transaction. The Charging Station SHALL only send the Ended event once for every transaction. |
| MaxEnergyOnInvalidId | integer | Maximum amount of energy in Wh delivered when an identifier is deauthorized by the CSMS after start of a transaction. |
| StopTxOnInvalidId | boolean | whether the Charging Station will stop an ongoing transaction when it receives a non- Accepted authorization status in TransactionEventResponse for this transaction. |
| StopTxOnEVSideDisconnect | boolean | When set to true, the Charging Station SHALL administratively stop the transaction when the cable is unplugged from the EV. |

## Physical Components

This is a non-exhaustive list of Standardized Physical Components that SHALL be used when mapping a real Charging Station to the Device Model (for monitoring purposes).

When the physical component that is to be mapped, matches the *description* of one of the Standardized Components in this section, for interoperability reasons, the Standardized Component from this section SHALL be used, instead of adding a new (proprietary) component.

The list of typically used variables that is given for each Component is also non-exhaustive and all variables are optional. See also Part 1, paragraph 4.5. If a description of a variable is empty, please refer to the description in [Standardized Variables](#_bookmark8).

* + 1. **AccessBarrier**

|  |  |
| --- | --- |
| **Description** | |
| Allows physical access of vehicles to a charging site to be controlled. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Open |
| Problem |  |

* + 1. **AcDcConverter**

|  |  |
| --- | --- |
| **Description** | |
| Provides a variable DC current source to force energy directly into an EV battery stack, under tight control of the EV’s battery management system. | |
| **Typically used variables** | **Description** |
| Enabled | (not commanded Out of Service) |
| Problem | some problem/fault exists |
| Tripped | A problem requiring intervention has occurred |
| Overload | Excessive current/power consumption |
| DCVoltage | measured DC voltage |
| DCCurrent | measured DC current |
| Power | measured power |
| Temperature | temperature of converter |
| FanSpeed | Speed of cooling fan(s) |

* + 1. **AcPhaseSelector**

|  |  |
| --- | --- |
| **Description** | |
| Allows a specific AC phase to be selected (typically at EVSE tier) for single phase vehicle charging in order to lower overall (e.g. site) phase imbalance. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Changing |
| Problem |  |
| PhaseRotation |  |

* + 1. **Actuator**

|  |  |
| --- | --- |
| **Description** | |
| A general purpose electro-mechanical output system, with optional completion tracking sensing. Each output should use a Variable instance key indicating the nature of the output. | |
| **Typically used variables** | **Description** |
| Enabled |  |

|  |  |
| --- | --- |
| **Description** | |
| Active | Non-Default |
| Problem |  |
| State |  |

* + 1. **AirCoolingSystem**

|  |  |
| --- | --- |
| **Description** | |
| Fans (or equivalent devices) used to provide cooling. | |
| **Typically used variables** | **Description** |
| Enabled | Cooling system enabled to run |
| Active | Cooling |
| Problem | fault: e.g. fan stalled/slow |
| FanSpeed | Speed of cooling fan(s) |

* + 1. **AreaVentilation**

|  |  |
| --- | --- |
| **Description** | |
| Fans (or equivalent devices) used to ensure that EVs that require ventilation during charging | |
| **Typically used variables** | **Description** |
| Enabled | Area ventilation enabled |
| Active | Ventilating |
| Problem | fault: e.g. fan stalled/slow |
| FanSpeed | Speed of cooling fan(s) |

* + 1. **BayOccupancySensor**

|  |  |
| --- | --- |
| **Description** | |
| Sensor (optical, ground loop, ultrasonic, etc.) to detect whether the associated parking/charging bay is physically vacant, or is occupied by a vehicle or other obstruction | |
| **Typically used variables** | **Description** |
| Enabled | Sensor is sensing for occupancy |
| Active | Occupied |
| Percent | percentage obstruction (for analogue sensors). |

* + 1. **BeaconLighting**

|  |  |
| --- | --- |
| **Description** | |
| Beacon Lighting to help EV drivers to locate nearby charging places, and/or to determine charging availability state, usually by color variation. | |
| **Typically used variables** | **Description** |
| Enabled | Beacon Lighting operational |
| Enabled(Set)=0 | Disable beacon lighting |
| Active | On |
| Problem | Beacon lighting fault |
| Percent | Lighting Level (% of maximum) |
| Percent(Set)=x% | Lighting Level (% of maximum) |
| Power | Lighting Wattage |
| Color | Displayed color/intensity |

* + 1. **CableBreakawaySensor**

|  |  |
| --- | --- |
| **Description** | |
| A sensor that detects when a charging cable (captive or removable) has been forcibly pulled from the Charging Station. | |
| **Typically used variables** | **Description** |
| Enabled | Breakaway sensor operational |
| Active | Tripped |
| Tripped | Breakaway detected: manual check/fix required |

* + 1. **CaseAccessSensor**

|  |  |
| --- | --- |
| **Description** | |
| Reports when an access door/panel is open | |
| **Typically used variables** | **Description** |
| Enabled | Access sensor is enabled to detect/report opening/closing of access door/panel |
| Enabled(Set)=0 | Disable reporting of access |
| Active | Open |
| Tripped | An access door/panel that needs manual reset action has been activated |
| Problem | A fault exists in the Sensor mechanism itself |

* + 1. **ChargingStation**

|  |  |
| --- | --- |
| **Description** | |
| The entire Charging Station as a logical entity | |
| **Typically used variables** | **Description** |
| Enabled | Available for use (not commanded Out of Service) |
| Problem | Some problem/fault exists |
| Identity | Charging Station identity |
| Tripped | A problem requiring local/manual intervention has occurred. |
| Overload | Excessive current/power consumption |
| SupplyPhases | Number of AC supply phases connected |
| SupplyPhases(MaxLimit) | Number of AC supply phases supported |
| PhaseRotation | AC wiring phase rotation |
| ACVoltage | Measured incoming AC voltage [per phase] |
| ACVoltage(MaxLimit) | Designed maximum operating AC voltage |
| ACCurrent | Measured total AC current [per phase] |
| Power | Measured/calculated total power being consumed, including standby/ancilliary loads |
| Power(MaxLimit) | Designed total operating load power, including standby/ancilliary loads |
| VoltageImbalance | voltage imbalance in three phase supply |
| CurrentImbalance | current imbalance in three phase supply |
| VendorName | Charging Station vendor name (as reported in BootNotification) |
| Model | Charging Station model (as reported in BootNotification) |
| ECVariant | Engineering Change Variant |
| SerialNumber | Charging Station serial number |
| OperatingTimes | recurring operating times |
| ChargeProtocol | Charging Control Protocol applicable to the Charging Station |
| AvailabilityState | Indicates if the Charging Station is available or not (replaces the Charging Station Status values reported by the StatusNotification) |
| AllowNewSessionsPendingFirmware Update | Indicates whether new sessions can be started on EVSEs, while Charging Station is waiting for all EVSEs to become Available in order to start a pending firmware update. |

* + 1. **ChargingStatusIndicator**

|  |  |
| --- | --- |
| **Description** | |
| The Charging Status Indicator, provides visible feedback to the user about the connection and charging status of an EVSE/Connector. This is commonly in the form of multi-colored lighting. | |
| **Typically used variables** | **Description** |
| Active | Lighted |
| Color | Displayed color |

* + 1. **ConnectedEV** *(updated in v1.3)*

ConnectedEV is a component that represents a connected vehicle for which data is received via an ISO 15118 or CHAdeMO interface. The generic information that is received, is represented as variables of ConnectedEV. Any protocol-specific information is represented in variables of the ISO15118Ctrlr or CHAdeMOCtrlr component.

**Description**

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Unit** | **ISO 15118-2 value** | **CHAdeMO value** |
| Available | boolean | Is true when an EV is connected | |
| ***Vehicle:*** | | | |
| VehicleId | string | EVCCID (from SessionSetupReq)  *Six bytes, represented as hexbinary encoded string, e.g. "010203040A0B"* | Vehicle ID (H'710 + H'711 + H'712)  *Three times 8 bytes, represented as hexbinary encoded string, e.g. "****010203040A0B0C0D****111213141A1B1C1D*  ***212223242A2B2C2D****". A concatenation of*  ***H'710*** *+ H'711 +* ***H'712****.* |
| ProtocolAgreed | string | A string with the following comma- separated items: “<uri>,<major>,<minor>”.  This is the protocol uri and version information that was agreed upon between EV and EVSE in the supportedAppProtocolReq handshake from ISO 15118.  Example: "urn:iso:15118:2:2013:MsgDef,2,0" | Lowest of Chademo protocol number from EV (H'102.0) and charger (H'109.0) |
| ProtocolSupportedByEV | string | A string with the following comma- separated items: “<uri>,<major>,<minor>”.  This is information from the supportedAppProtocolReq message from ISO 15118.  Variable has multiple instances, one for each priority.  Example: "urn:iso:15118:2:2013:MsgDef,2,0" | Chademo protocol number (H'102.0) |
| ***Voltage and current values:*** | | | |
| ACCurrent.minSet | A | EVMinCurrent | - |
| ACCurrent.maxSet | A | EVMaxCurrent | - |
| ACVoltage.maxSet | V | EVMaxVoltage | - |
| DCCurrent.minSet | A | - | Minimum charge current (H'100.0) |
| DCCurrent.maxSet | A | EVMaximumCurrentLimit | - |
| DCCurrent.target | A | EVTargetCurrent | Charging current request (H'102.3)  *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 | EVMaximumVoltageLimit | Maximum battery voltage (H'100.4,5) |
| DCVoltage.target | V | EVTargetVoltage | Target battery voltage (H'102.1,2) |
| ***Power, energy and time values:*** | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Variable** | **Unit** | **ISO 15118-2 value** | **CHAdeMO value** |
| Power.maxSet | W | EVMaximumPowerLimit | - |
| EnergyCapacity | Wh | EVEnergyCapacity | Total capacity of traction battery \* 100 (H'101.5,6) |
| EnergyImport.target | Wh | EVEnergyRequest (DC) EAmount (AC) | - |
| DepartureTime | dateTime | DepartureTime  *Provided as seconds since message receipt. Convert to absolute time.* | - |
| RemainingTimeBulk | s | RemainingTimeToBulkSoC | - |
| RemainingTimeFull.maxSet | s | - | Maximum charging time \* 60 (H'101.2) |
| RemainingTimeFull.actual | s | RemainingTimeToFullSoc | Estimated charging time \* 60 (H'101.3) |
| StateOfChargeBulk | % | BulkSoC | - |
| StateOfCharge.maxSet | % | FullSoC | Charged rate reference constant (H'100.6) |
| StateOfCharge.actual | % | DC\_EVStatus.EVRESSSOC | State of charge (H'102.6) |
| ChargingCompleteBulk | boolean | BulkChargingComplete | - |
| ChargingCompleteFull | boolean | ChargingComplete | - |
| ***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* | | FAILED\_ChargingCurrentDifferential | Battery current deviation (H'102.4.2) |
| \* *BatteryTemperature* | | FAILED\_RESSTemperatureInhibit | High battery temperature (H'102.4.3) |
| \* *VoltageDeviation* | | FAILED\_ChargingVoltageOutOfRange | Battery voltage deviation (H'102.4.4) |
| \* *ChargingSystemError* | | FAILED\_EVRESSMalfunction | Charging system error (H'102.5.2) |
| \* *VehicleShiftPosition* | | FAILED\_EVShiftPosition | Vehicle shift position (H'102.5.1) |
| \* *VehicleChargingEnabled* | | - | Vehicle charging enabled (H'102.5.0) |
| \* *ChargingSystemIncompatibility* | | FAILED\_ChargingSystemIncompatibility | - |
| \* *ChargerConnectorLockFault* | | FAILED\_ChargerConnectorLockFault | - |

* + 1. **Connector**

|  |  |
| --- | --- |
| **Description** | |
| A means to connect an EV to a Charging Station with either a socket, an attached cable & inline connector, or any wireless power transfer device. | |
| **Typically used variables** | **Description** |
| Enabled | Connector available for use (not commanded Out of Service) |
| Problem | problem/fault exists (e.g. over-temperature) |
| Tripped | A problem requiring intervention has occurred. |
| ConnectorType | Type of connector as defined by ConnectorEnumType in "Part 2 - Specification" plus additionally: cGBT, cChaoJi, OppCharge. |
| SupplyPhases | AC phases connected |
| SupplyPhases(MaxLimit) | AC phases Max |
| PhaseRotation | AC wiring phase rotation |
| ChargeProtocol | Charging Control Protocol applicable to the Connector |
| AvailabilityState | Indicates if the Connector is available or not (replaces the Status values reported by the StatusNotification) |

* + 1. **ConnectorHolsterRelease**

|  |  |
| --- | --- |
| **Description** | |
| A mechanism present in a connector holster to prevent the connector from being removed inappropriately: typically unlocks connector after authorization. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Unlocked for removal/return |
| Problem |  |
| State |  |

* + 1. **ConnectorHolsterSensor**

|  |  |
| --- | --- |
| **Description** | |
| A mechanism to report when a tethered cable connector has been removed from its normal stowage position. May be used for detection of connectors left un-holstered, and possible penalty billing. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Un-Holstered |
| Problem |  |

* + 1. **ConnectorPlugRetentionLock**

|  |  |
| --- | --- |
| **Description** | |
| Locking mechanism to retain an inserted plug, both to prevent on-load disconnection, and to prevent theft of charging cables | |
| **Typically used variables** | **Description** |
| Enabled | Retention mechanism enabled |
| Active | Locked |
| Problem | Locking Failed |
| Tripped | Stall protection fuse blown, etc. |
| Tries | (Re)tries taken on last attempt |
| Tries(SetLimit) | Configured auto retry count |
| Tries(MaxLimit) | Maximum auto retry count |

* + 1. **ConnectorProtectionRelease**

|  |  |
| --- | --- |
| **Description** | |
| External protective mechanism (e.g. an external shutter or a connector holster lock mechanism) to prevent contact with conductors that may become "live" under other failure modes | |
| **Typically used variables** | **Description** |
| Enabled | Protection in effect (locked except when in use) |
| Active | Unlocked |
| Problem | Lock/Unlock mechanism fault |
| Tripped | protective mechanism triggered (fuse) |

* + 1. **Controller**

|  |  |
| --- | --- |
| **Description** | |
| An embedded logic controller | |
| **Typically used variables** | **Description** |
| Active | Running |
| Problem | Controller fault |
| Interval[Heartbeat] | Heartbeat interval |

|  |  |
| --- | --- |
| **Description** | |
| Manufacturer | Controller manufacturer name |
| Model | Controller model number |
| ECVariant | Engineering Change variant |
| SerialNumber | Controller hardware serial number |
| VersionNumber | Hardware version number |
| VersionDate | Hardware version date |
| FirmwareVersion | Firmware version number (as reported in BootNotification) |
| MaxMsgElements | Array of implementation-defined limits to the number of elements of specific type that the Charging Station can accept in one message. |
| SelftestActive(Set) | Start self-test |
| SelftestActive | Self-test running |

* + 1. **ControlMetering**

|  |  |
| --- | --- |
| **Description** | |
| Energy, Power, Electricity meter, used to measure energy, current, voltages etc. | |
| **Typically used variables** | **Description** |
| Power | Measured power |
| ACCurrent | Measured AC current [per phase] |
| DCCurrent | Measured DC current |
| DCVoltage | Measured DC voltage |

* + 1. **CPPWMController**

|  |  |
| --- | --- |
| **Description** | |
| Control Pilot PWM Controller: provides and senses the IEC 61851-1 / SAE J1772 low voltage DC and PWM signalling between an EVSE and EV over a control pilot line. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Connected |
| Problem | CP PWM controller fault |
| DCVoltage | Control Pilot wire DC Voltage (0-12V) |
| State | IEC 61851-1 states ("A" to "E") |
| Percentage | 1kHz Duty Cycle |
| SelftestActive(Set) | Start self-test |
| SelftestActive | Self-test running |

* + 1. **DataLink**

|  |  |
| --- | --- |
| **Description** | |
| Provides a communications link from a Charging Station to a CSMS. It may use fixed infrastructure, mobile telephony data services, WiFi, or other connectivity channels. | |
| **Typically used variables** | **Description** |
| Enabled | Data link enabled |
| Active | Connected |
| Fallback | Using Backup SIM/Network Preference |
| Complete | Link connection terminated |
| Problem | Communications module or link connection fault |
| IMSI | International Mobile Subscriber Identity number of mobile data SIM card |
| ICCID | Integrated Circuit Card IDentifier of mobile data SIM card. |

|  |  |
| --- | --- |
| **Description** | |
| NetworkAddress | Current network address |
| SignalStrength | Data signal strength/quality |

* + 1. **Display**

|  |  |
| --- | --- |
| **Description** | |
| Provides information and feedback to the user. | |
| **Typically used variables** | **Description** |
| Enabled | Display configured to show information |
| Problem | Display fault |
| Color | Display color (monochrome/backlighting) |
| Count[HeightInChars] | Display height (characters) |
| Count[WidthInChars] | Display width (characters) |
| DataText[Visible] | Current Display Contents |
| State | Alphanumeric code indicating current message purpose |

* + 1. **DistributionPanel**

|  |  |
| --- | --- |
| **Description** | |
| Defines the Distribution Panel, with it’s fuses and connections to both Charging Stations and other Distribution Panel’s. | |
| **Common Variables** | **Description** |
| InstanceName | Name of the distribution box |
| Fuse | Fuse (index n) is the fuse for phase Ln in Ampere. |
| ChargingStation | The Identity of Charging Station (index n) which is connnected to this DistributionPanel.  Note: this is an indexed list of Charging Station Identities, not to be confused by the Charging Station component. |
| ChargingStation | List of Charging Stations Identities connected to this LocalController. (not to be confused with the ChargingStation Component) |
| DistributionPanel | List of Distribution Panels InstanceNames connected to this LocalController. (not to be confused with the DistributionPanel Component)  See the LocalController component for an example. |

* + 1. **ElectricalFeed**

|  |  |
| --- | --- |
| **Description** | |
| Represents an incoming electrical connection to a Charging Station, that may be a grid/distribution network connection, of a connection to local power generation and/or storage. Each electrical feed can record the electrical and other characteristics of that feed, including power rating, fusing, upstream metering, etc. When a Charging Station has more than one electrical feed, it must represent which feed supplies each EVSE, and which feed supplies the house load of the Charging Station itself. Simple Charging Stations with only a single electrical feed may omit all electrical feed information, in which case it is inferred that all power is supplied from a single feed, and what would otherwise be ElectricalFeed data (Variables) may be reported as being associated with the ChargingStation component. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Connected |
| Problem |  |
| PowerType |  |
| Power |  |
| Energy |  |
| DCVoltage |  |
| SupplyPhases |  |
| PhaseRotation |  |

|  |  |
| --- | --- |
| **Description** | |
| ACVoltage |  |

* + 1. **ELVSupply**

|  |  |
| --- | --- |
| **Description** | |
| Represents the low voltage power supply (typically 12V DC and often other ELV voltages) that provides operating power for controllers, relays, and other electrical components. | |
| **Typically used variables** | **Description** |
| EnergyImportRegister | Standby/house energy meter register reading |
| Power | instantaneous standby power consumption |
| Power(MaxLimit) | Design maximum standby power consumption |
| Fallback | Running on backup energy; |
| Fallback(MaxLimit): =1 | has backup |
| StateOfCharge | backup battery SOC |
| Time | (estimated) operating time on backup energy |

* + 1. **EmergencyStopSensor**

|  |  |
| --- | --- |
| **Description** | |
| An "Emergency Stop" button that should be pressed by the user or other nearby persons if serious faulty behavior is observed (e.g. smoke/flames from EV or Charging Station). | |
| **Typically used variables** | **Description** |
| Enabled | Emergency Stop action armed |
| Active | Pressed/Latched |
| Tripped | Needs manual reset |

* + 1. **EnvironmentalLighting**

|  |  |
| --- | --- |
| **Description** | |
| Provides reporting/control of general illumination lighting in use at Charging Station. | |
| **Typically used variables** | **Description** |
| Enabled | Environmental Lighting operational |
| Enabled(Set)=0 | Disable Environmental lighting |
| Active | On |
| Problem | Environmental lighting fault |
| Percent | Lighting Level (% of maximum) |
| Percent(Set)=x% | Lighting Level (% of maximum) |
| Power | Lighting Wattage |
| Color | Displayed color/intensity |

* + 1. **EVRetentionLock**

|  |  |
| --- | --- |
| **Description** | |
| A locking mechanism on the EV side as a safety measure to prevent it being disconnected while high currents are flowing. | |
| **Typically used variables** | **Description** |
| Enabled | Retention locking detection in effect |
| Active | Locked to EV |
| Complete | Has unlocked |
| Problem | Lock Problem (e.g. failed to lock/unlock) |

* + 1. **EVSE**

|  |  |
| --- | --- |
| **Description** | |
| The entire chain of components responsible for transporting energy from the incoming supply to the electric vehicle (or vice versa) | |
| **Typically used variables** | **Description** |
| Enabled | Ready for use (not commanded Out of Service) |
| Problem | some problem/fault exists |
| Tripped | A problem requiring intervention has occurred |
| Overload | Excessive current/power consumption |
| EvseId | The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2. |
| SupplyPhases | AC phases connected |
| PhaseRotation | AC wiring phase rotation |
| AllowReset | When true: EVSE can be reset individually |
| ACVoltage | Measured total AC voltage [per phase] |
| ACCurrent | Measured total AC current [per phase] |
| DCVoltage | Measured total DC voltage [per phase] |
| DCCurrent | Measured total DC current [per phase] |
| Power | Measured Power |
| VoltageImbalance | voltage imbalance in three phase supply |
| CurrentImbalance | current imbalance in three phase supply |
| ChargeProtocol | Charging Control Protocol applicable to the EVSE |
| ChargingTime | Total time duration that EV is taking energy from an EVSE. Short pauses in charging (e.g. battery pre-, post-conditioning) are included |
| PostChargingTime | Total time since EV has taken energy from EVSE |
| Count[ChargingProfiles] | Charging Profiles present |
| Count[ChargingProfiles](MaxLimit) | Maximum Charging Profiles supported |
| ISO15118EvseId | The name of the EVSE in the string format as required by ISO 15118 and IEC 63119-2. Example: "DE\*ICE\*E\*1234567890\*1" |

* + 1. **ExternalTemperatureSensor**

|  |  |
| --- | --- |
| **Description** | |
| Reports ambient air temperature | |
| **Typically used variables** | **Description** |
| Active | Temperature above MaxSet or MinSet |
| Problem | Temperature sensor fault |
| Temperature | Ambient temperature |

* + 1. **FiscalMetering**

|  |  |
| --- | --- |
| **Description** | |
| Provides energy transfer readings that are the basis for billing. | |
| **Typically used variables** | **Description** |
| Problem | Metering Fault (e.g. read error) |
| EnergyImport | Energy transferred to EV during session |
| EnergyImportRegister | Cumulative import reading |
| EnergyExport | Energy transferred from EV during session |
| EnergyExportRegister | Cumulative export reading |
| Manufacturer[Meter] | Meter manufacturer name |
| Manufacturer[CT] | Current transformer manufacturer name |
| Model[Meter] | Meter model number |

|  |  |
| --- | --- |
| **Description** | |
| Model[CT] | CT model number |
| ECVariant | Meter engineering change variant |
| SerialNumber[Meter] | Meter serial number |
| SerialNumber[CT] | CT serial number(s) |
| Certificate |  |
| OptionsSet [MeterValueAlignedData] | Set of measurands to read and report at clock-aligned time intervals while charging. |
| OptionsSet [TxnStoppedAlignedData] | Set of measurands to be read at clock-aligned time intervals while charging and reported in TransactionStopped |

* + 1. **FloodSensor**

|  |  |
| --- | --- |
| **Description** | |
| A sensor reporting whether the Charging Station is experiencing water ingress/pooling. | |
| **Typically used variables** | **Description** |
| Enabled | Water presence/level sensing in effect |
| Active | Flooding |
| Tripped | Water level safety sensor tripped |
| Height | Absolute water height above reference (ground) level. |
| Percent | Height as percentage between reference minimum (0%) and maximum allowable (100%). Values below 0% and above 100% are possible. |

* + 1. **GroundIsolationProtection**

|  |  |
| --- | --- |
| **Description** | |
| An Isolation Tester as part of their own self-test mechanisms, to confirm the isolation of floating circuitry when no Evs are connected | |
| **Typically used variables** | **Description** |
| Enabled | Electrical isolation testing enabled |
| Active | Leakage |
| Complete | Isolation test completed |
| Problem | Isolation fault |
| Impedance | Isolation resistance/impedance |

* + 1. **Heater**

|  |  |
| --- | --- |
| **Description** | |
| Heater to ensure reliable operation in cold environments | |
| **Typically used variables** | **Description** |
| Enabled | Heater hardware operation enabled |
| Active | Heating |
| Problem | Heater fault |
| Tripped | Heater equipment permanent fault |
| Power | Instantaneous heater power level |
| Power(MaxLimit) | Maximum heater power |
| Power(MaxSet) | Configured heater power |
| Temperature(MinSet) | Cut-in temperature |
| Temperature(MaxSet) | Cut-out temperature |

* + 1. **HumiditySensor**

|  |  |
| --- | --- |
| **Description** | |
| Reports relative air humidity | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Problem | Humidity sensor fault |
| Humidity | RH(%) |

* + 1. **LightSensor**

|  |  |
| --- | --- |
| **Description** | |
| Reports ambient light levels. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Problem | Lighting sensor fault |
| Light | The ambient light level |

* + 1. **LiquidCoolingSystem**

|  |  |
| --- | --- |
| **Description** | |
| A liquid based cooling system, typically used to cool the connector cables of very high power Charging Stations. | |
| **Typically used variables** | **Description** |
| Enabled | Cooling system enabled to run |
| Active | Liquid circulating |
| Problem |  |
| Temperature |  |

* + 1. **LocalAvailabilitySensor**

|  |  |
| --- | --- |
| **Description** | |
| Accepts local signal inputs controlling whether new Charging Sessions can start and/or whether ongoing sessions should continue. Typically connected to a site/building power supply, to automatically report unavailability when closed. | |
| **Typically used variables** | **Description** |
| Enabled | Local Availability input sensing in operation |
| Active | Out of Service |
| Problem | Local Availability sensing circuit error |

* + 1. **LocalController**

|  |  |
| --- | --- |
| **Description** | |
| The entire Local Controller as a logical entity | |
| **Common Variables** | **Description** |
| Enabled | Available for use (not commanded Out of Service) |
| Problem | Some problem/fault exists |
| Identity | Local Controller identity |
| Tripped | A problem requiring local/manual intervention has occurred. |
| Manufacturer | Local Controller manufacturer name |
| Model | Local Controller manufacturer model |
| ECVariant | Engineering Change Variant |
| SerialNumber | Local Controller serial number |

|  |  |
| --- | --- |
| **Description** | |
| ChargingStation | List of Charging Stations Identities connected to this LocalController. (not to be confused with the ChargingStation Component) |
| DistributionPanel | List of Distribution Panels InstanceNames connected to this LocalController. (not to be confused with the DistributionPanel Component)  This can be used to describes the electrical connections in the site controlled by the Local Controller.  An example. The incoming fuses are all 120A. Each floor has a set of 80A fuses. On the first floor, there’s also a group of Charging Stations that are behind a set of 32A fuses.  DistributionPanel.Fuse[1] = 120  DistributionPanel.Fuse[2] = 120  DistributionPanel.Fuse[3] = 120 DistributionPanel.DistributionPanel[0] = "Level-1" DistributionPanel.DistributionPanel[1] = "Level-2"  DistributionPanel["Level-1"].Fuse[1] = 80  DistributionPanel["Level-1"].Fuse[2] = 80  DistributionPanel["Level-1"].Fuse[3] = 80 DistributionPanel["Level-1"].ChargingStation[0] = "NLCP013" DistributionPanel["Level-1"].ChargingStation[1] = "NLCP014" DistributionPanel["Level-1"].ChargingStation[2] = "NLCP015" DistributionPanel["Level-1"].DistributionPanel[0] = "Level-1a"  DistributionPanel["Level-1a"].Fuse[1] = 32  DistributionPanel["Level-1a"].Fuse[2] = 32  DistributionPanel["Level-1a"].Fuse[3] = 32 DistributionPanel["Level-1a"].ChargingStation[0] = "NLCP130" DistributionPanel["Level-1a"].ChargingStation[1] = "NLCP136" DistributionPanel["Level-1a"].ChargingStation[2] = "NLCP132"  DistributionPanel["Level-2"].Fuse[1] = 80  DistributionPanel["Level-2"].Fuse[2] = 80  DistributionPanel["Level-2"].Fuse[3] = 80 DistributionPanel["Level-2"].ChargingStation[0] = "NLCP023" DistributionPanel["Level-2"].ChargingStation[1] = "NLCP024" |

* + 1. **LocalEnergyStorage** *(updated in v1.3)*

|  |  |
| --- | --- |
| **Description** | |
| Local energy storage device | |
| **Typically used variables** | **Description** |
| EnergyCapacity | Maximum storage capacity |
| Identity | Local Energy Storage identity |

* + 1. **OverCurrentProtection**

|  |  |
| --- | --- |
| **Description** | |
| Protects equipment by disconnecting the electrical supply when the current drawn (on any phase) exceeds the rated value to a substantial degree. | |
| **Typically used variables** | **Description** |
| Active | Tripped. Trip when over MaxSet/MaxLimit. |
| Operated | Breaker opened and auto-reclosed |
| ACCurrent | Measured total AC current [per phase] |

* + 1. **OverCurrentProtectionRecloser**

|  |  |
| --- | --- |
| **Description** | |
| Recloser mechanism of an OverCurrentProtection to perform re-arm retries after a trip, or may be set for remotely controlled re- arming on command. | |
| **Typically used variables** | **Description** |
| Enabled | Auto reclosing enabled |
| Active | Reclosing |
| Active(Set) | Initiate manual reclose |
| Complete | Reclose cycle completed |
| Problem | Recloser Fault |
| Mode | Reclose Mode (None, Auto, Commanded) |
| Tries | (Re)tries taken on last attempt |
| Tries(SetLimit) | Configured auto retry count |
| Tries(MaxLimit) | Maximum auto retry count |

* + 1. **PowerContactor**

|  |  |
| --- | --- |
| **Description** | |
| Switches on and off the power to the EV after all authorization and safety requirements have been met. May have secondary contacts to report closure state. | |
| **Typically used variables** | **Description** |
| Active | Closed |
| Tripped | Mirror contact protection tripped |
| Problem | Close/Open failed |

* + 1. **RCD**

|  |  |
| --- | --- |
| **Description** | |
| A Residual Current Device (US: ground fault breaker) protects human life and/or downstream equipment by quickly detecting abnormal current flows (usually indicative in earth faults) in the Charging Station, cable, or EV during charging. | |
| **Typically used variables** | **Description** |
| Tripped | Breaker opened (manual reset required) |
| Operated | Breaker opened and auto-reclosed |

* + 1. **RCDRecloser**

|  |  |
| --- | --- |
| **Description** | |
| A motorized recloser mechanism of an RCD that may be configured to perform re-arm retries after a trip, or may be set for remotely controlled re-arming on command. | |
| **Typically used variables** | **Description** |
| Enabled | Auto reclosing enabled |
| Active | Reclosing in progress |
| Active(Set) | Initiate manual reclose |
| Complete | Reclose cycle completed |
| Problem | Recloser Fault |
| Tries | (Re)tries taken on last attempt |
| Tries(SetLimit) | Configured auto (re)try count |
| Tries(MaxLimit) | Maximum auto (re)try count |

* + 1. **RealTimeClock**

|  |  |
| --- | --- |
| **Description** | |
| Represents realtime clock hardware that can maintain accurate date & time information in a Charging Station, even in the case of simultaneous CSMS uncontactability and power outages or resets. | |
| **Typically used variables** | **Description** |
| Active | RTC running OK |
| DCVoltage | Battery voltage |
| Fallback | Battery failing |
| Fallback(MaxLimit) | RTC has backup-power. MaxLimit = 1 |
| Problem | RTC fault |

* + 1. **ShockSensor**

|  |  |
| --- | --- |
| **Description** | |
| Measures impact forces/accelerations experienced, indicative of possible damage. | |
| **Typically used variables** | **Description** |
| Enabled | Shock sensing enabled |
| Active | Shock |
| Force | detected force (vector) |

* + 1. **SpacesCountSignage**

|  |  |
| --- | --- |
| **Description** | |
| Electronic signage allowing a charging controller for a large charging facility to advertise counts of available spaces to passing traffic. | |
| **Typically used variables** | **Description** |
| Enabled | Spaces count signage enabled |
| Active | Not Blank |
| Count |  |

* + 1. **Switch**

|  |  |
| --- | --- |
| **Description** | |
| A general purpose electromechanical input device, with optional remote defaulting/resetting of values. Each input should use a Variable instance key indicating the nature of the input. | |
| **Typically used variables** | **Description** |
| Enabled |  |
| Active | Non-Default |
| State |  |

* + 1. **TemperatureSensor**

|  |  |
| --- | --- |
| **Description** | |
| Temperature sensor at a point inside the Charging Station; multiple sensing points for a single sensing controller. Multiple sensing points for a single sensing controller may be reported using distinct Variable instance keys. | |
| **Typically used variables** | **Description** |
| Active | High temperature (over MaxSet) |
| Problem | Internal temperature sensor fault |
| Temperature | Enclosure temperature |

* + 1. **TiltSensor**

|  |  |
| --- | --- |
| **Description** | |
| Measures Tilt angle from normal reference position (normally 90 degree vertical). | |
| **Typically used variables** | **Description** |
| Enabled | Tilt sensing enabled |
| Active | Tilted |
| Angle | Measured tilt (vector) from vertical |

* + 1. **TokenReader**

|  |  |
| --- | --- |
| **Description** | |
| An authorization token reader (e.g. RFID) | |
| **Typically used variables** | **Description** |
| Enabled | Token reader enabled |
| Enabled(Set)=0 | Token reader disabled: allow charging without token authentication/authorization |
| Operated | token data read event |
| Problem | token reader fault |
| Token | String read by TokenReader |
| TokenType | Type of token as IdTokenEnumType |

* + 1. **UpstreamProtectionTrigger**

|  |  |
| --- | --- |
| **Description** | |
| Circuitry designed to trigger the disconnection of power to the structure by an upstream protection device after a severe problem has been detected | |
| **Typically used variables** | **Description** |
| Enabled | Upstream protection enabled |
| Active(Set) | Force triggering of upstream protection |
| Tripped | Upstream protection triggered |
| Problem | Upstream protection fault |

* + 1. **UIInput**

|  |  |
| --- | --- |
| **Description** | |
| A logical input mechanism (e.g. set of buttons) that is part of a UI whose use may be communicated to the CSMS (in near real time). May support momentary inputs ("Operated") or modal state ("Active"). Multiple input sources should use explicit Variable instance keys (where the input function is key name). | |
| **Typically used variables** | **Description** |
| Enabled | UI input enabled |
| Operated |  |
| Active |  |

* + 1. **VehicleIdSensor**

|  |  |
| --- | --- |
| **Description** | |
| Reports an identifier associated with a vehicle occupying a charging bay. The identifier may be a vehicle registration number via ANPR hardware, a VIN, or other local identifier of the vehicle based on medium range/active RFID, or any other relevant technology and result. | |
| **Typically used variables** | **Description** |
| Enabled | VehicleIdSensor enabled |
| Active | Processing |

## Summary List of Standardized Components

Following is a list that sums up all above-mentioned standardized component names.

|  |  |
| --- | --- |
| **Component** | **Description** |
| AlignedDataCtrlr | Logical Component responsible for configuration relating to the reporting of clock-aligned meter data. |
| AuthCtrlr | Logical Component responsible for configuration relating to the use of authorization for Charging Station use. |
| AuthCacheCtrlr | Logical Component responsible for configuration relating to the use of a local cache for authorization for Charging Station use. |
| CHAdeMOCtrlr | A CHAdeMO Controller component communicates with an EV using the wired CANbus protocol to exchange information and control charging using the CHAdeMO protocol |
| ClockCtrlr | Provides a means to configure management of time tracking by Charging Station. |
| DeviceDataCtrlr | Logical Component responsible for configuration relating to the exchange and storage of Charging Station Device Model data. |
| DisplayMessageC trlr | Logical Component responsible for configuration relating to the display of messages to Charging Station users. |
| ISO15118Ctrlr | Communicates with an EV to exchange information and control charging using the ISO 15118 protocol. |
| LocalAuthListCtrl r | Logical Component responsible for configuration relating to the use of Local Authorization Lists for Charging Station use. |
| MonitoringCtrlr | Logical Component responsible for configuration relating to the exchange of monitoring event data. |
| OCPPCommCtrlr | Logical Component responsible for configuration relating to information exchange between Charging Station and CSMS. |
| ReservationCtrlr | Logical Component responsible for configuration relating to reservations. |
| SampledDataCtrlr | Logical Component responsible for configuration relating to the reporting of sampled meter data. |
| SecurityCtrlr | Logical Component responsible for configuration relating to security of communications between Charging Station and CSMS. |
| SmartChargingCt rlr | Logical Component responsible for configuration relating to smart charging. |
| TariffCostCtrlr | Logical Component responsible for configuration relating to tariff and cost display. |
| TxCtrlr | Logical Component responsible for configuration relating to transaction characteristics and behaviour. |
| AccessBarrier | Allows physical access of vehicles to a charging site to be controlled. |
| AcDcConverter | Provides a variable DC current source to force energy directly into an EV battery stack, under tight control of the EV’s battery management system. |
| AcPhaseSelector | Allows a specific AC phase to be selected (typically at EVSE tier) for single phase vehicle charging in order to lower overall (e.g. site) phase imbalance. |
| Actuator | A general purpose electro-mechanical output system, with optional completion tracking sensing. Each output should use a Variable instance key indicating the nature of the output. |
| AirCoolingSystem | Fans (or equivalent devices) used to provide cooling. |
| AreaVentilation | Fans (or equivalent devices) used to ensure that EVs that require ventilation during charging |
| BayOccupancySe nsor | Sensor (optical, ground loop, ultrasonic, etc.) to detect whether the associated parking/charging bay is physically vacant, or is occupied by a vehicle or other obstruction |
| BeaconLighting | Beacon Lighting to help EV drivers to locate nearby charging places, and/or to determine charging availability state, usually by color variation. |
| CableBreakawayS ensor | A sensor that detects when a charging cable (captive or removable) has been forcibly pulled from the Charging Station. |
| CaseAccessSens or | Reports when an access door/panel is open |
| ChargingStation | The entire Charging Station as a logical entity |
| ChargingStatusIn dicator | The Charging Status Indicator, provides visible feedback to the user about the connection and charging status of an EVSE/Connector. This is commonly in the form of multi-colored lighting. |
| ConnectedEV | ConnectedEV is a component that represents a connected vehicle for which data is received via an ISO 15118 or CHAdeMO interface. The generic information that is received, is represented as variables of ConnectedEV. Any protocol-specific information is represented in variables of the ISO15118Ctrlr or CHAdeMOCtrlr component. |
| Connector | A means to connect an EV to a Charging Station with either a socket, an attached cable & inline connector, or any wireless power transfer device. |

|  |  |
| --- | --- |
| **Component** | **Description** |
| ConnectorHolster Release | A mechanism present in a connector holster to prevent the connector from being removed inappropriately: typically unlocks connector after authorization. |
| ConnectorHolster Sensor | A mechanism to report when a tethered cable connector has been removed from its normal stowage position. May be used for detection of connectors left un-holstered, and possible penalty billing. |
| ConnectorPlugRe tentionLock | Locking mechanism to retain an inserted plug, both to prevent on-load disconnection, and to prevent theft of charging cables |
| ConnectorProtect ionRelease | External protective mechanism (e.g. an external shutter or a connector holster lock mechanism) to prevent contact with conductors that may become 'live' under other failure modes |
| Controller | An embedded logic controller |
| ControlMetering | Energy, Power, Electricity meter, used to measure energy, current, voltages etc. |
| CPPWMControlle r | Control Pilot PWM Controller: provides and senses the IEC 61851-1 / SAE J1772 low voltage DC and PWM signalling between an EVSE and EV over a control pilot line. |
| DataLink | Provides a communications link from a Charging Station to a CSMS. It may use fixed infrastructure, mobile telephony data services, WiFi, or other connectivity channels. |
| Display | Provides information and feedback to the user. |
| DistributionPanel | Defines the Distribution Panel, with it’s fuses and connections to both Charging Stations and other Distribution Panel’s. |
| ElectricalFeed | Represents an incoming electrical connection to a Charging Station, that may be a grid/distribution network connection, of a connection to local power generation and/or storage. Each electrical feed can record the electrical and other characteristics of that feed, including power rating, fusing, upstream metering, etc. When a Charging Station has more than one electrical feed, it must represent which feed supplies each EVSE, and which feed supplies the house load of the Charging Station itself. Simple Charging Stations with only a single electrical feed may omit all electrical feed information, in which case it is inferred that all power is supplied from a single feed, and what would otherwise be ElectricalFeed data (Variables) may be reported as being associated with the ChargingStation component. |
| ELVSupply | Represents the low voltage power supply (typically 12V DC and often other ELV voltages) that provides operating power for controllers, relays, and other electrical components. |
| EmergencyStopS ensor | An 'Emergency Stop' button that should be pressed by the user or other nearby persons if serious faulty behavior is observed (e.g. smoke/flames from EV or Charging Station). |
| EnvironmentalLig hting | Provides reporting/control of general illumination lighting in use at Charging Station. |
| EVRetentionLock | A locking mechanism on the EV side as a safety measure to prevent it being disconnected while high currents are flowing. |
| EVSE | The entire chain of components responsible for transporting energy from the incoming supply to the electric vehicle (or vice versa) |
| ExternalTemperat ureSensor | Reports ambient air temperature |
| FiscalMetering | Provides energy transfer readings that are the basis for billing. |
| FloodSensor | A sensor reporting whether the Charging Station is experiencing water ingress/pooling. |
| GroundIsolationP rotection | An Isolation Tester as part of their own self-test mechanisms, to confirm the isolation of floating circuitry when no Evs are connected |
| Heater | Heater to ensure reliable operation in cold environments |
| HumiditySensor | Reports relative air humidity |
| LightSensor | Reports ambient light levels. |
| LiquidCoolingSys tem | A liquid based cooling system, typically used to cool the connector cables of very high power Charging Stations. |
| LocalAvailabilityS ensor | Accepts local signal inputs controlling whether new Charging Sessions can start and/or whether ongoing sessions should continue. Typically connected to a site/building power supply, to automatically report unavailability when closed. |
| LocalController | The entire Local Controller as a logical entity |
| LocalEnergyStora ge | Energy storage |
| OverCurrentProte ction | Protects equipment by disconnecting the electrical supply when the current drawn (on any phase) exceeds the rated value to a substantial degree. |
| OverCurrentProte ctionRecloser | Recloser mechanism of an OverCurrentProtection to perform re-arm retries after a trip, or may be set for remotely controlled re-arming on command. |

|  |  |
| --- | --- |
| **Component** | **Description** |
| PowerContactor | Switches on and off the power to the EV after all authorization and safety requirements have been met. May have secondary contacts to report closure state. |
| RCD | A Residual Current Device (US: ground fault breaker) protects human life and/or downstream equipment by quickly detecting abnormal current flows (usually indicative in earth faults) in the Charging Station, cable, or EV during charging. |
| RCDRecloser | A motorized recloser mechanism of an RCD that may be configured to perform re-arm retries after a trip, or may be set for remotely controlled re-arming on command. |
| RealTimeClock | Represents realtime clock hardware that can maintain accurate date & time information in a Charging Station, even in the case of simultaneous CSMS uncontactability and power outages or resets. |
| ShockSensor | Measures impact forces/accelerations experienced, indicative of possible damage. |
| SpacesCountSign age | Electronic signage allowing a charging controller for a large charging facility to advertise counts of available spaces to passing traffic. |
| Switch | A general purpose electromechanical input device, with optional remote defaulting/resetting of values. Each input should use a Variable instance key indicating the nature of the input. |
| TemperatureSens or | Temperature sensor at a point inside the Charging Station, multiple sensing points for a single sensing controller. Multiple sensing points for a single sensing controller may be reported using distinct Variable instance keys. |
| TiltSensor | Measures Tilt angle from normal reference position (normally 90 degree vertical). |
| TokenReader | An authorization token reader (e.g. RFID) |
| UpstreamProtecti onTrigger | Circuitry designed to trigger the disconnection of power to the structure by an upstream protection device after a severe problem has been detected |
| UIInput | A logical input mechanism (e.g. set of buttons) that is part of a UI whose use may be communicated to the CSMS (in near real time). May support momentary inputs ('Operated') or modal state ('Active'). Multiple input sources should use explicit Variable instance keys (where the input function is key name). |
| VehicleIdSensor | Reports an identifier associated with a vehicle occupying a charging bay. The identifier may be a vehicle registration number via ANPR hardware, a VIN, or other local identifier of the vehicle based on medium range/active RFID, or any other relevant technology and result. |

# Appendix 4. Standardized Variables

This is a non-exhaustive list of Standardized Variables that SHALL be used when the Charging Station and CSMS want to exchange information about a Variable. See also Part 1, paragraph 4.5.

Variables that are specific to a Controller Component are not included in this list, but are part of section 3.1 Controller Components.

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **DataType** | **Unit** | **Description** |
| ACCurrent | decimal | A | RMS AC Current (in amperes). For 3-phase circuits, each phase (and optional neutral) is represented by a Variable instance equal to a value of the PhaseEnumType (e.g. L1,N). Unkeyed values reported for a Component declared to be multi-phase are assumed to be an average of all per-phase readings and written values are common per-phase settings. Example(s): ChargingStation: Total AC current consumption (all EVSE’s, ancillaries), EVSE: Total current consumed by EVSE: includes losses (AC→DC) and EVSE specific ancillaries (e.g. fans), ElectricalFeed: Inflow AC current on feed |
| Active | boolean |  | Component is in its non-resting / active state: e.g: On, Engaged, Locked. Some Components may have secondary functions that have corresponding Active Variables with an explicit Variable instance., Note: Monitoring of changes in the Active state of any Component can be specified by setting Delta monitoring on the boolean value with a delta values of 1. Setting/clearing an Active Variable activates/stops the associated functionality, where remotely controllable. Only components that are Available and Enabled can be in the Active state. |
| ACVoltage | decimal | V | RMS AC Voltage (in volts). For 3-phase circuits, each phase (and optional neutral) is represented by a Variable instance equal to a value of the PhaseEnumType (e.g. L1,N). Unkeyed values reported for a Component declared to be multi-phase are assumed to be an average ofall per-phase readings and written values are common per-phase settings. Example(s): ElectricalFeed: Input Voltage |
| AllowReset | boolean |  | Component can be reset. Can be used to announce that an EVSE can be reset individually. |
| Angle | decimal | Deg | Angle(s) relative to normal/design idle position. Multiple Variable instance values may be used to indicate angular position in multiple axes (e.g. Left-Right, Forward-Back). |
| Attempts | integer |  | Number of attempts (INCLUDING the original attempt) in the last successful or attempted, cycle of operation. Applies typically to self- monitoring motorized electro-mechanical equipment, etc. {Null}: Unknown, 0: Not Attempted/Not allowed, 1: Single attempt/No retries [allowed], 2-N: [up to] N tries [allowed] |
| AvailabilityState | OptionList |  | A value of ConnectorStatusEnumType (See part 2): replicates ConnectorStatus values reported in StatusNotification messages. |
| Available | boolean |  | The Component exists and is locally configured/wired for use, but might not be (remotely) Enabled. |
| Certificate | string |  | Digital Certificate (in Base64 encoding) |
| ChargeProtocol |  |  | The Charging Control Protocol applicable to a Connector. CHAdeMO: CHAdeMO protocol, ISO15118: ISO15118 V2G protocol (wired or wireless) as used with CCS, CPPWM: IEC61851-1 / SAE J1772 protocol (ELV DC & PWM signalling via Control Pilot wire), Uncontrolled: No charging power management applies (e.g. Schuko socket), Undetermined: Yet to be determined (e.g. before plugged in), Unknown: Not determinable, NOTE: ChargeProtocol is distinct from and orthogonal to connectorType. |
| ChargingCompleteBulk | boolean |  | Charging up to StateOfChargeBulk has completed. |
| ChargingCompleteFull | boolean |  | Charging up to StateOfCharge.maxSet has completed. |
| ChargingTime | decimal | s | Time from earliest to latest substantive energy transfer |
| Color | string |  | Standard 24 bit hexadecimal RGB values. Reg Green Blue color intensity, expressed as standard 24 bit hexadecimal RGB values: 3 00- FF (0-255), in order RRGGBB). E.g. 000000: Black, FF0000: Red, 00FF00: Green, 0000FF: Blue, FFFF00:Yellow, FFFFFF: White, 008000:  Medium intensity green. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **DataType** | **Unit** | **Description** |
| Complete | boolean |  | Component’s operation cycle has completed. Used only in event notifications, where it is always true. |
| ConnectedTime | decimal | s | Time since logical connection established |
| ConnectorType | OptionList |  | A value of ConnectorEnumType (See part 2) plus additionally: cGBT, cChaoJi, OppCharge. Specific type of connector, including sub- variant information. Note: Distinct and orthogonal to Charging Protocol, Power Type, Phases. |
| Count | integer |  | General purpose integer count variable for Component state reporting |
| Currency | string |  | Currency in a ISO 4217 formatted currency code. |
| CurrentImbalance | decimal | Percent | Percentage current imbalance in an AC three phase supply. |
| DataText | string |  | Text associated with a Component, e.g. a Display. |
| DateTime | dateTime |  | Point in time value, in [RFC3339] datetime format. Time zone optional. |
| DCCurrent | decimal | A | DC Current (in amperes). May be an instantaneous measurement, or a period average, depending on context/equipment. |
| DCVoltage | decimal | V | DC Voltage (volts). May be an instantaneous measurement, or a period average, depending on context/equipment. |
| DepartureTime | dateTime |  | Time in [RFC3339] datetime format, when an EV intends to leave the charging station. |
| ECVariant | string |  | Production series variants reflecting internal design changes or sub- component substitutions not affecting external functionality. |
| Enabled | boolean |  | The Component is Enabled for operation. For Available components that cannot be selectively (remotely) enabled / disabled, this value is always true. Note: Available cannot be false of Enabled is true, so during inventory reporting, Enabled=1 also logically states Available=true |
| Energy | decimal | Wh | Energy quantity (in Wh) for reporting/configuring values related to stored energy (i.e. not transferred energy). |
| EnergyCapacity | decimal | Wh | Energy capacity in Wh of an energy storage device. |
| EnergyExport | decimal | Wh | Total energy transferred: e.g. from EV during (ongoing or terminated) charging session (in wH by default) |
| EnergyExportRegister | decimal | Wh | Cumulative export kWh register value, such as from a (certified) fiscal energy meter. |
| EnergyImport | decimal | Wh | Total energy transferred. |
| EnergyImportRegister | decimal | Wh | Cumulative export kWh register value, such as from a (certified) fiscal energy meter. |
| Entries | integer |  | General purpose variable for reporting/managing numbers of entries in repetitive data structures. maxLimit characteristic reports maximum possible entries. |
| EvseId | string |  | EVSE ID in string format as used in ISO 15118 and IEC 63119-2 |
| Fallback | boolean |  | Component is operating in a fallback, or backup mode. In inventory reports, a Value of 1 for the maxLimit characteristic indicates that the component can enter a fallback state (i.e. a fallback mode is present). |
| FanSpeed | decimal | RPM | Fan Speed (in RPM). A value of 0 represents stopped/stalled. An empty value indicates that fan speed cannot be read. |
| FirmwareVersion | string |  | Version number of firmware. |
| Force | decimal | N | Reports (impact) force/ acceleration values (estimates) in one or more directions, in units of Newtons or “g”. Multiple force readings in different (orthogonal) dimensions may be reported using Variable instance values, such as Down, Right, Forward. |
| Formats | MemberLi st |  | List of message formats supported by this Charging Station. Possible values: ASCII, HTML, URI, UTF-8. |
| Frequency | decimal | Hz | Frequency of AC power, signal, or component operation. |
| FuseRating | decimal | A | Current rating of a fuse/breaker. Variable instances keyed by phase identifier (L1/L2/L3/N). |
| Height | decimal | m | Height above(+)/below(-) reference level (ground level unless context demands otherwise). |
| Humidity | decimal | RH | The relative humidity in %. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **DataType** | **Unit** | **Description** |
| Hysteresis | decimal | Percent | Specifies the width of a 'dead band' (as a percentage of the threshold) around the central value of a threshold setting (e.g. MinSet, MaxSet, monitor thresholds) to avoid repeated triggering when the measured quantity lies close to the threshold and is subject to small variations. |
| ICCID | string |  | ICCID (Integrated Circuit Card IDentifier) of mobile data SIM card. |
| Impedance | decimal | Ohm | Impedance: Primary value is real (resistive only) impedance. Where a complex impedance is to be reported, the imaginary part (reactance) must be represented with a separate Variable instance value of 'reactance'. Reactance values are expressed at the (nominal) relevant operating frequency of the Component (e.g. 50/60Hz for mains electricity feed). |
| IMSI | string |  | IMSI (International Mobile Subscriber Identity) number of mobile data SIM card |
| Interval | integer | s | Minimum Interval (in seconds) between (attempted) operations. |
| Length | decimal | m | General Purpose linear distance measure. |
| Light | decimal | lx | (Ambient) light level. The value is in Lux. |
| Manufacturer | string |  | Component Manufacturer name |
| Message | string |  | Specific stored message for display. |
| MinimumStatusDuration | integer | s | Minimum duration that a Charging Station or EVSE status is stable before StatusNotificationRequest is sent to the CSMS. |
| Mode | string |  | Operating mode string from among valid options (communicated by OptionList, etc. during capability/configuration discovery). |
| Model | string |  | Manufacturer’s Model code/number of Component, including suffixes etc. to identify functional, regional or linguistic variation, but NOT engineering change level internal variation not affecting external behaviour, etc. |
| NetworkAddress | string |  | Current network address of a Component. |
| Operated | boolean |  | The Component operated in an instantaneous, transient, or immediately self-resetting pattern. Used only in event notifications, where it is always true. |
| OperatingTimes | string |  | Recurring operating times in iCalendar RRULE format. |
| Overload | boolean |  | Component is in Overload state. |
| Percent | decimal | Percent | Generic dimensionless value reporting/setting value. |
| PhaseRotation | OptionList |  | The phase wiring of Component, relative to it’s upstream feed Component/device. This variable describes the phase rotation of a Component relative to its parent Component, using a three letter string consisting of the letters: R, S, T and x. The letter 'R' can be identified as phase 1 (L1), 'S' as phase 2 (L2), 'T' as phase 3 (L3). The lower case 'x' is used to designate a phase that is not connected. An empty string means that phase rotation is not applicable or not known. |
| PostChargingTime | decimal | s | Elapsed time in seconds since last substantive energy transfer |
| Power | decimal | W,kW | Instantaneous (real) Power (measured/calculated, including power factor for AC). Where a component (e.g. AC to DC Power Converter) has multiple power measurements, the default (unkeyed) instance is “input” power. |
| Problem | boolean |  | Component itself has a 'Problem' condition that impacts in any significant way on its normal operation. By definition, 'Problem' state includes (logical OR) 'Fault' state. 'Problem' specifically INCLUDES inability to operate that is propagated (up/down/sideways) from any other associated/connected/containing/contained Component. |
| Protecting | boolean |  | Applies to 'sensor' type Components that have an associated protection capability, whereby they can disconnect power (e.g. using the main PowerContactor) if the sensed quantity is outside preset/configured limits. If Protecting is true, the Component is actively preventing/interrupting charging. |
| RemainingTimeBulk | integer | s | Number of seconds remaining to charge to bulk state of charge, given by StateOfChargeBulk. |
| RemainingTimeFull | integer | s | Number of seconds remaining to charge to 100% state of charge. |
| SeccId | string |  | The name of the SECC in the string format as required by ISO 15118. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **DataType** | **Unit** | **Description** |
| SerialNumber | string |  | Serial number of Component. |
| SignalStrength | decimal | dBm | (Radio/Wired/Optical) data signal strength, in ASU (typically 0-31 or 99 for unknown). Or dbmW (typically -140 to -50). |
| State | string |  | A state code or name identifier string, to allow the internal state of components to be reported and/or controlled |
| StateOfCharge | decimal | Percent | Energy Storage Device (e.g. battery) state of charge, expressed as a percentage of nominal design 0-100% operating range. The value of StateOfCharge.maxSet represents the maximum state of charge for a full battery and is usually at or near 100%. |
| StateOfChargeBulk | decimal | Percent | Energy Storage Device (e.g. battery) state of charge up to which fast charging is possible. Above this percentage charging speed will drop significantly. |
| Storage | integer | B | In bytes. Amount of storage occupied. Storage(maxLimit) specifies absolute limit Storage(MaxSet) restricts usage to specified Max, if supported. |
| SupplyPhases | integer |  | Number of alternating current phases connected/available. 1 or 3 for AC, 0 means DC (no alternating phases). Null value indicates that the number of phases (e.g. in use) is unknown. |
| Suspending | boolean |  | If Suspending is true, the Component can is currently suspending charging. |
| Suspension | boolean |  | Applies to 'sensor' type Components that have a charging suspension capability, typically for safety or equipment protection reasons. If Suspension is true, the component can suspend charging when the sensed quantity is outside preset/configured limits. |
| Temperature | decimal | Celsius, Fahrenheit | Temperature(s) of component (in Celsius, by default). Components may have multiple indexed temperature sensors. |
| Time | dateTime |  | Point in time value, in ISO 8601 datetime format. Time zone optional. |
| TimeOffset | string |  | A Time Offset with respect to Coordinated Universal Time (aka UTC or Greenwich Mean Time) in the form of an [RFC3339] time (zone) offset suffix, including the mandatory “+” or “-“ prefix. |
| Timeout | decimal | s | Generic timeout value for Component operation (in seconds). |
| Token | string |  | String of bytes representing an ID token. |
| TokenType | OptionList |  | Type of Token. Value is one of IdTokenEnumType. |
| Tries | integer |  | Number of attempts done by a Component. |
| Tripped | boolean |  | Single-shot device requires explicit intervention to re-prime/activate to normal. |
| VehicleId | string |  | ID that EV provides to charging station. Encoded as a hexbinary string. In ISO 15118 the EVCCID is 6 bytes (MAC address), in CHAdeMO the vehicle id can be 24 bytes. |
| VersionDate | dateTime |  | [RFC3339] |
| VersionNumber | string |  | Version number of hardware |
| VoltageImbalance | decimal | Percent | Percentage voltage imbalance in three phase supply. |

# Appendix 5. Reason Codes

The table below provides a list of standardized reason codes that can be used in the optional StatusInfo element of a response.

For each reason code, some messages that might typically return them are shown. This is not an exhaustive list and only indicative.

StatusInfo is optional. Any implementation should be able to function properly without StatusInfo, because every message has the response code values that are essential to perform the function. The *reasonCode* and *additionalInfo* in StatusInfo are meant to provide more insight on what is happening and maybe allow for some automatic recovery.

### IMPORTANT

The existence of a reason code in this table does not imply a requirement to use it nor does it imply a requirement to any of the mentioned messages.

|  |  |  |
| --- | --- | --- |
| **Reason Code** | **Description** | **Typically used for** |
| CSNotAccepted | BootNotification of Charging Station has not (yet) been accepted by CSMS. | RequestStartTransaction, RequestStopTransaction |
| DuplicateProfile | A charging profile with same *stackLevel - chargingProfilePurpose* combination already exists on the Charging Station and has an overlapping validity period. | SetChargingProfile |
| DuplicateRequestId | A *requestId* is provided, that has already been used for this type of request. | UpdateFirmware, PublishFirmware and requests for reports. |
| FixedCable | The connector has its own fixed cable that cannot be unlocked. | UnlockConnector |
| FwUpdateInProgress | Operation is not possible, because a firmware update is in progress. | Reset |
| InternalError | Operation cannot be completed due to an internal error. | (generic) |
| InvalidCertificate | Provided certificate is invalid. | CertificateSigned, InstallCertificate |
| InvalidCSR | Provided CSR is invalid | SignCertificate |
| InvalidIdToken | Provided *idToken* is not valid. | RequestStartTransaction |
| InvalidMessageSequence | Message should not be sent at this moment in current scenario. | (generic), SetChargingProfile with ISO15118 |
| InvalidProfile | Provided *chargingProfile* contains invalid elements. | SetChargingProfile, RequestStartTransaction |
| InvalidSchedule | Provided *chargingSchedule* contains invalid elements. | SetChargingProfile, RequestStartTransaction |
| InvalidStackLevel | Provided value for *stackLevel* is invalid. | SetChargingProfile |
| InvalidURL | Provided URL is invalid. | UpdateFirmware, PublishFirmware |
| InvalidValue | An invalid value has been provided. | (generic) |
| MissingDeviceModelInfo | Information needed for operation is missing from Device Model | (generic) |
| MissingParam | A parameter that is required for the request is missing. | (generic) |
| NoCable | No cable is connected at this time. | UnlockConnector |
| NoError | No error has occurred, but some extra information is in  *additionalInfo* . | (generic) |
| NotEnabled | Feature is not enabled. | ClearCache |
| NotFound | No object(s) found that match a provided ID or criteria. | ClearVariableMonitoring, CustomerInformation, GetChargingProfiles, GetDisplayMessages, GetInstalledCertificateIds, GetReport |
| OutOfMemory | Operation not possible, because system does not have enough memory. | (generic) |
| OutOfStorage | Operation not possible, because system does not have enough storage. | (generic) |
| ReadOnly | Targeted variable is read-only and cannot be set. | SetVariables |

|  |  |  |
| --- | --- | --- |
| **Reason Code** | **Description** | **Typically used for** |
| TooLargeElement | Provided element is too large to handle. | CertificateSigned, InstallCertificate |
| TooManyElements | Too many elements have been provided. | SetChargingProfile, SetVariables, SendLocalList |
| TxInProgress | A transaction is in progress. | ChangeAvailability, Reset, RequestStartTransaction |
| TxNotFound | There is no such transaction. | RequestStopTransaction, SetChargingProfile |
| TxStarted | A transaction had already started (e.g. due to cable being plugged in). | RequestStartTransaction |
| UnknownConnectorId | Connector Id is not known on EVSE | ChangeAvailability, UnlockConnector |
| UnknownConnectorType | Connector type is not known on EVSE | ReserveNow |
| UnknownEvse | EVSE is not known on Charging Stations | ChangeAvailability, ReserveNow, RequestStartTransaction |
| UnknownTxId | Provided *transactionId* is not known. | RequestStopTransaction |
| Unspecified | No reason is specified, but some extra information is in  *additionalInfo* | (generic) |
| UnsupportedParam | A parameter was provided that is not supported. | (generic) |
| UnsupportedRateUnit | A *chargingRateUnit* is provided that is not supported. | SetChargingProfile |
| UnsupportedRequest | This request is not supported. | (generic) |
| ValueOutOfRange | Provided value is out of range. | SetVariables, SetVariableMonitoring |
| ValuePositiveOnly | Provided value is not greater than zero. | (generic) |
| ValueTooHigh | Provided value is too high. | (generic) |
| ValueTooLow | Provided value is too low. | (generic) |
| ValueZeroNotAllowed | Provided value cannot be zero. | (generic) |
| WriteOnly | Targeted variable is write-only and cannot be read. | GetVariables |