6.45. StartTransaction.req

This section contains the field definition of the StartTransaction.req PDU sent by the Charge Point to the Central System. See also Start Transaction

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
connectorId	integer connectorId > 0	1..1	Required. This identifies which connector of the Charge Point is used.
idTag	IdToken	1..1	Required. This contains the identifier for which a transaction has to be started.
meterStart	integer	1..1	Required. This contains the meter value in Wh for the connector at start of the transaction.
reservationId	integer	0..1	Optional. This contains the id of the reservation that terminates as a result of this transaction.
timestamp	dateTime	1..1	Required. This contains the date and time on which the transaction is started.

6.46. StartTransaction.conf

This contains the field definition of the StartTransaction.conf PDU sent by the Central System to the Charge Point in response to a StartTransaction.req PDU. See also Start Transaction

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
idTagInfo	IdTagInfo	1..1	Required. This contains information about authorization status, expiry and parent id.
transactionId	integer	1..1	Required. This contains the transaction id supplied by the Central System.

6.47. StatusNotification.req

This contains the field definition of the StatusNotification.req PDU sent by the Charge Point to the Central System. See also Status Notification

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
connectorId	integer connectorId >= 0	1..1	Required. The id of the connector for which the status is reported. Id '0' (zero) is used if the status is for the Charge Point main controller.
errorCode	ChargePointErrorCode	1..1	Required. This contains the error code reported by the Charge Point.
info	CiString50Type	0..1	Optional. Additional free format information related to the error.
status	ChargePointStatus	1..1	Required. This contains the current status of the Charge Point.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
timestamp	dateTime	0..1	Optional. The time for which the status is reported. If absent time of receipt of the message will be assumed.
vendorId	CiString255Type	0..1	Optional. This identifies the vendor-specific implementation.
vendorErrorCode	CiString50Type	0..1	Optional. This contains the vendor-specific error code.

6.48. StatusNotification.conf

This contains the field definition of the StatusNotification.conf PDU sent by the Central System to the Charge Point in response to an StatusNotification.req PDU. See also Status Notification

No fields are defined.

6.49. StopTransaction.req

This contains the field definition of the StopTransaction.req PDU sent by the Charge Point to the Central System. See also Stop Transaction

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
idTag	IdToken	0..1	Optional. This contains the identifier which requested to stop the charging. It is optional because a Charge Point may terminate charging without the presence of an idTag, e.g. in case of a reset. A Charge Point SHALL send the idTag if known.
meterStop	integer	1..1	Required. This contains the meter value in Wh for the connector at end of the transaction.
timestamp	dateTime	1..1	Required. This contains the date and time on which the transaction is stopped.
transactionId	integer	1..1	Required. This contains the transaction-id as received by the StartTransaction.conf.
reason	Reason	0..1	Optional. This contains the reason why the transaction was stopped. MAY only be omitted when the Reason is "Local".
transactionData	MeterValue	0..*	Optional. This contains transaction usage details relevant for billing purposes.

6.50. StopTransaction.conf

This contains the field definition of the StopTransaction.conf PDU sent by the Central System to the Charge Point in response to a StopTransaction.req PDU. See also Stop Transaction

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
idTagInfo	IdTagInfo	0..1	Optional. This contains information about authorization status, expiry and parent id. It is optional, because a transaction may have been stopped without an identifier.

6.51. TriggerMessage.req

This contains the field definition of the TriggerMessage.req PDU sent by the Central System to the Charge Point. See also Trigger Message

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
requestedMessage	MessageTrigger	1..1	Required.
connectorId	integer connectorId > 0	0..1	Optional. Only filled in when request applies to a specific connector.

6.52. TriggerMessage.conf

This contains the field definition of the TriggerMessage.conf PDU sent by the Charge Point to the Central System in response to a TriggerMessage.req PDU. See also Trigger Message

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	TriggerMessageStatus	1..1	Required. Indicates whether the Charge Point will send the requested notification or not.

6.53. UnlockConnector.req

This contains the field definition of the UnlockConnector.req PDU sent by the Central System to the Charge Point. See also Unlock Connector

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
connectorId	integer connectorId > 0	1..1	Required. This contains the identifier of the connector to be unlocked.

6.54. UnlockConnector.conf

This contains the field definition of the UnlockConnector.conf PDU sent by the Charge Point to the Central System in response to an UnlockConnector.req PDU. See also Unlock Connector

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
status	UnlockStatus	1..1	Required. This indicates whether the Charge Point has unlocked the connector.

6.55. UpdateFirmware.req

This contains the field definition of the UpdateFirmware.req PDU sent by the Central System to the Charge Point. See also Update Firmware

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
location	anyURI	1..1	Required. This contains a string containing a URI pointing to a location from which to retrieve the firmware.
retries	integer	0..1	Optional. This specifies how many times Charge Point must try to download the firmware before giving up. If this field is not present, it is left to Charge Point to decide how many times it wants to retry.
retrieveDate	dateTime	1..1	Required. This contains the date and time after which the Charge Point is allowed to retrieve the (new) firmware.
retryInterval	integer	0..1	Optional. The interval in seconds after which a retry may be attempted. If this field is not present, it is left to Charge Point to decide how long to wait between attempts.

6.56. UpdateFirmware.conf

This contains the field definition of the UpdateFirmware.conf PDU sent by the Charge Point to the Central System in response to a UpdateFirmware.req PDU. See also Update Firmware

No fields are defined.

7. Types

7.1. AuthorizationData

Class

Elements that constitute an entry of a Local Authorization List update.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
idTag	IdToken	1..1	Required. The identifier to which this authorization applies.
idTagInfo	IdTagInfo	0..1	Optional. (Required when UpdateType is Full) This contains information about authorization status, expiry and parent id. For a Differential update the following applies: If this element is present, then this entry SHALL be added or updated in the Local Authorization List. If this element is absent, than the entry for this idtag in the Local Authorization List SHALL be deleted.

7.2. AuthorizationStatus

Enumeration

Status in a response to an Authorize.req.

VALUE	DESCRIPTION
Accepted	Identifier is allowed for charging.
Blocked	Identifier has been blocked. Not allowed for charging.
Expired	Identifier has expired. Not allowed for charging.
Invalid	Identifier is unknown. Not allowed for charging.
ConcurrentTx	Identifier is already involved in another transaction and multiple transactions are not allowed. (Only relevant for a StartTransaction.req.)

7.3. AvailabilityStatus

Enumeration

Status returned in response to ChangeAvailability.req.

VALUE	DESCRIPTION
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.
Scheduled	Request has been accepted and will be executed when transaction(s) in progress have finished.

7.4. AvailabilityType

Enumeration

Requested availability change in ChangeAvailability.req.

VALUE	DESCRIPTION
Inoperative	Charge point is not available for charging.
Operative	Charge point is available for charging.

7.5. CancelReservationStatus

Enumeration

Status in CancelReservation.conf.

VALUE	DESCRIPTION
Accepted	Reservation for the identifier has been cancelled.
Rejected	Reservation could not be cancelled, because there is no reservation active for the identifier.

7.6. ChargePointErrorCode

Enumeration

Charge Point status reported in StatusNotification.req.

VALUE	DESCRIPTION
ConnectorLockFailure	Failure to lock or unlock connector.
EVCommunicationError	Communication failure with the vehicle, might be Mode 3 or other communication protocol problem. This is not a real error in the sense that the Charge Point doesn’t need to go to the faulted state. Instead, it should go to the SuspendedEVSE state.
GroundFailure	Ground fault circuit interrupter has been activated.
HighTemperature	Temperature inside Charge Point is too high.
InternalError	Error in internal hard- or software component.
LocalListConflict	The authorization information received from the Central System is in conflict with the LocalAuthorizationList.
NoError	No error to report.

VALUE	DESCRIPTION
OtherError	Other type of error. More information in vendorErrorCode.
OverCurrentFailure	Over current protection device has tripped.
OverVoltage	Voltage has risen above an acceptable level.
PowerMeterFailure	Failure to read electrical/energy/power meter.
PowerSwitchFailure	Failure to control power switch.
ReaderFailure	Failure with idTag reader.
ResetFailure	Unable to perform a reset.
UnderVoltage	Voltage has dropped below an acceptable level.
WeakSignal	Wireless communication device reports a weak signal.

7.7. ChargePointStatus

Enumeration

Status reported in StatusNotification.req. A status can be reported for the Charge Point main controller (connectorId = 0) or for a specific connector. Status for the Charge Point main controller is a subset of the enumeration: *Available*, *Unavailable* or *Faulted*.

States considered Operative are: *Available*, *Preparing*, *Charging*, *SuspendedEVSE*, *SuspendedEV*, *Finishing*, *Reserved*. States considered Inoperative are: *Unavailable*, *Faulted*.

STATUS	CONDITION
Available	When a Connector becomes available for a new user (Operative)
Preparing	When a Connector becomes no longer available for a new user but there is no ongoing Transaction (yet). Typically a Connector is in preparing state when a user presents a tag, inserts a cable or a vehicle occupies the parking bay (Operative)
Charging	When the contactor of a Connector closes, allowing the vehicle to charge (Operative)
SuspendedEVSE	When the EV is connected to the EVSE but the EVSE is not offering energy to the EV, e.g. due to a smart charging restriction, local supply power constraints, or as the result of StartTransaction.conf indicating that charging is not allowed etc. (Operative)
SuspendedEV	When the EV is connected to the EVSE and the EVSE is offering energy but the EV is not taking any energy. (Operative)

STATUS	CONDITION
Finishing	When a Transaction has stopped at a Connector, but the Connector is not yet available for a new user, e.g. the cable has not been removed or the vehicle has not left the parking bay (Operative)
Reserved	When a Connector becomes reserved as a result of a Reserve Now command (Operative)
Unavailable	When a Connector becomes unavailable as the result of a Change Availability command or an event upon which the Charge Point transitions to unavailable at its discretion. Upon receipt of a Change Availability command, the status MAY change immediately or the change MAY be scheduled. When scheduled, the Status Notification shall be send when the availability change becomes effective (Inoperative)
Faulted	When a Charge Point or connector has reported an error and is not available for energy delivery . (Inoperative).

7.8. ChargingProfile

Class

A ChargingProfile consists of a ChargingSchedule, describing the amount of power or current that can be delivered per time interval.

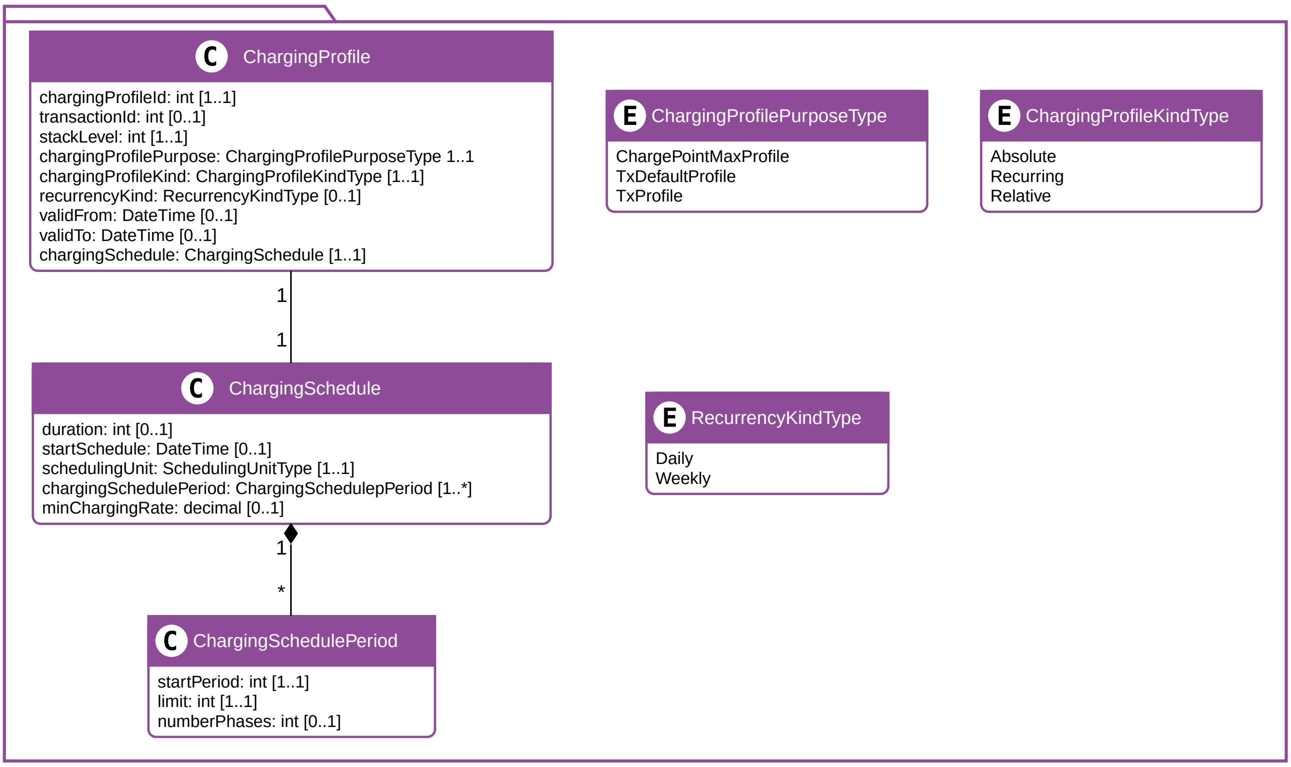
C ChargingProfile
chargingProfileId: int [1..1]
transactionId: int [0..1]
stackLevel: int [1..1]
chargingProfilePurpose: ChargingProfilePurposeType 1..1
chargingProfileKind: ChargingProfileKindType [1..1]
recurrencyKind: RecurrencyKindType [0..1]
validFrom: DateTime [0..1]
validTo: DateTime [0..1]
chargingSchedule: ChargingSchedule [1..1]
E ChargingProfilePurposeType
ChargePointMaxProfile
TxDefaultProfile
TxProfile
E ChargingProfileKindType
Absolute
Recurring
Relative
1
1
C ChargingSchedule
duration: int [0..1]
startSchedule: DateTime [0..1]
schedulingUnit: SchedulingUnitType [1..1]
chargingSchedulePeriod: ChargingSchedulepPeriod [1..*]
minChargingRate: decimal [0..1]
E RecurrencyKindType
Daily
Weekly
1
*
C ChargingSchedulePeriod
startPeriod: int [1..1]
limit: int [1..1]
numberPhases: int [0..1]

Figure 42. Class Diagram: ChargingProfile

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
chargingProfileId	integer	1..1	Required. Unique identifier for this profile.
transactionId	integer	0..1	Optional. Only valid if ChargingProfilePurpose is set to TxProfile, the transactionId MAY be used to match the profile to a specific transaction.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
stackLevel	integer >=0	1..1	Required. Value determining level in hierarchy stack of profiles. Higher values have precedence over lower values. Lowest level is 0.
chargingProfilePurpose	ChargingProfilePurposeType	1..1	Required. Defines the purpose of the schedule transferred by this message.
chargingProfileKind	ChargingProfileKindType	1..1	Required. Indicates the kind of schedule.
recurrencyKind	RecurrencyKindType	0..1	Optional. Indicates the start point of a recurrence.
validFrom	dateTime	0..1	Optional. Point in time at which the profile starts to be valid. If absent, the profile is valid as soon as it is received by the Charge Point.
validTo	dateTime	0..1	Optional. Point in time at which the profile stops to be valid. If absent, the profile is valid until it is replaced by another profile.
chargingSchedule	ChargingSchedule	1..1	Required. Contains limits for the available power or current over time.

7.9. ChargingProfileKindType

Enumeration

Kind of charging profile, as used in: ChargingProfile.

VALUE	DESCRIPTION
Absolute	Schedule periods are relative to a fixed point in time defined in the schedule.
Recurring	The schedule restarts periodically at the first schedule period.
Relative	Schedule periods are relative to a situation-specific start point (such as the start of a Transaction) that is determined by the charge point.

7.10. ChargingProfilePurposeType

Enumeration

Purpose of the charging profile, as used in: ChargingProfile.

VALUE	DESCRIPTION
ChargePointMaxProfile	Configuration for the maximum power or current available for an entire Charge Point.
TxDefaultProfile	Default profile *that can be configured in the Charge Point. When a new transaction is started, this profile SHALL be used, unless it was a transaction that was started by a RemoteStartTransaction.req with a ChargeProfile that is accepted by the Charge Point.

VALUE	DESCRIPTION
TxProfile	Profile with constraints to be imposed by the Charge Point on the current transaction, or on a new transaction when this is started via a RemoteStartTransaction.req with a ChargeProfile. A profile with this purpose SHALL cease to be valid when the transaction terminates.

7.11. ChargingProfileStatus

Enumeration

Status returned in response to SetChargingProfile.req.

VALUE	DESCRIPTION
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.
NotSupported	Charge Point indicates that the request is not supported.

7.12. ChargingRateUnitType

Enumeration

Unit in which a charging schedule is defined, as used in: GetCompositeSchedule.req and ChargingSchedule

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

7.13. ChargingSchedule

Class

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

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
duration	integer	0..1	Optional. Duration of the charging schedule in seconds. If the duration is left empty, the last period will continue indefinitely or until end of the transaction in case startSchedule is absent.
startSchedule	dateTime	0..1	Optional. Starting point of an absolute schedule. If absent the schedule will be relative to start of charging.
chargingRateUnit	ChargingRateUnitType	1..1	Required. The unit of measure Limit is expressed in.
chargingSchedulePeriod	ChargingSchedulePeriod	1..*	Required. List of ChargingSchedulePeriod elements defining maximum power or current usage over time. The startSchedule of the first ChargingSchedulePeriod SHALL always be 0.
minChargingRate	decimal	0..1	Optional. Minimum charging rate supported by the electric vehicle. The unit of measure is defined by the chargingRateUnit. This parameter is intended to be used by a local smart charging algorithm to optimize the power allocation for in the case a charging process is inefficient at lower charging rates. Accepts at most one digit fraction (e.g. 8.1)

7.14. ChargingSchedulePeriod

Class

Charging schedule period structure defines a time period in a charging schedule, as used in: ChargingSchedule.

FIELD NAME	FIELD TYPE	CARD.	DESCRIPTION
startPeriod	integer	1..1	Required. Start of the period, in seconds from the start of schedule. The value of StartPeriod also defines the stop time of the previous period.
limit	decimal	1..1	Required. Charging rate limit during the schedule period, in the applicable chargingRateUnit, for example in Amperes or Watts. Accepts at most one digit fraction (e.g. 8.1).
numberPhases	integer	0..1	Optional. The number of phases that can be used for charging. If a number of phases is needed, numberPhases=3 will be assumed unless another number is given.

7.15. CiString20Type

Type

Generic used case insensitive string of 20 characters.

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

7.16. CiString25Type

Type

Generic used case insensitive string of 25 characters.

FIELD TYPE	DESCRIPTION
CiString[25]	String is case insensitive.

7.17. CiString50Type

Type

Generic used case insensitive string of 50 characters.

FIELD TYPE	DESCRIPTION
CiString[50]	String is case insensitive.

7.18. CiString255Type

Type

Generic used case insensitive string of 255 characters.

FIELD TYPE	DESCRIPTION
CiString[255]	String is case insensitive.

7.19. CiString500Type

Type

Generic used case insensitive string of 500 characters.

FIELD TYPE	DESCRIPTION
CiString[500]	String is case insensitive.

7.20. ClearCacheStatus

Enumeration

Status returned in response to ClearCache.req.

VALUE	DESCRIPTION
Accepted	Command has been executed.

VALUE	DESCRIPTION
Rejected	Command has not been executed.

7.21. ClearChargingProfileStatus

Enumeration

Status returned in response to ClearChargingProfile.req.

VALUE	DESCRIPTION
Accepted	Request has been accepted and will be executed.
Unknown	No Charging Profile(s) were found matching the request.

7.22. ConfigurationStatus

Enumeration

Status in ChangeConfiguration.conf.

VALUE	DESCRIPTION
Accepted	Configuration key is supported and setting has been changed.
Rejected	Configuration key is supported, but setting could not be changed.
RebootRequired	Configuration key is supported and setting has been changed, but change will be available after reboot (Charge Point will not reboot itself)
NotSupported	Configuration key is not supported.

7.23. DataTransferStatus

Enumeration

Status in DataTransfer.conf.

VALUE	DESCRIPTION
Accepted	Message has been accepted and the contained request is accepted.
Rejected	Message has been accepted but the contained request is rejected.
UnknownMessageId	Message could not be interpreted due to unknown messageId string.

VALUE	DESCRIPTION
UnknownVendorId	Message could not be interpreted due to unknown vendorId string.

7.24. DiagnosticsStatus

Enumeration

Status in DiagnosticsStatusNotification.req.

VALUE	DESCRIPTION
Idle	Charge Point is not performing diagnostics related tasks. Status Idle SHALL only be used as in a DiagnosticsStatusNotification.req that was triggered by a TriggerMessage.req
Uploaded	Diagnostics information has been uploaded.
UploadFailed	Uploading of diagnostics failed.
Uploading	File is being uploaded.

7.25. FirmwareStatus

Enumeration

Status of a firmware download as reported in FirmwareStatusNotification.req.

VALUE	DESCRIPTION
Downloaded	New firmware has been downloaded by Charge Point.
DownloadFailed	Charge point failed to download firmware.
Downloading	Firmware is being downloaded.
Idle	Charge Point is not performing firmware update related tasks. Status Idle SHALL only be used as in a FirmwareStatusNotification.req that was triggered by a TriggerMessage.req
InstallationFailed	Installation of new firmware has failed.
Installing	Firmware is being installed.
Installed	New firmware has successfully been installed in charge point.

7.26. GetCompositeScheduleStatus

Enumeration

Status returned in response to GetCompositeSchedule.req.

VALUE	DESCRIPTION
Accepted	Request has been accepted and will be executed.
Rejected	Request has not been accepted and will not be executed.

7.27. IdTagInfo

Class

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

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

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

7.28. IdToken

Type

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

FIELD TYPE	DESCRIPTION
CiString20Type	IdToken is case insensitive.

7.29. KeyValue

Class

Contains information about a specific configuration key. It is returned in GetConfiguration.conf.

NAME	FIELD TYPE	CARD.	DESCRIPTION
key	CiString50Type	1..1	Required.