



ENGAGE – JSON Data Structures

Version 2.18

Contents

5	Introduction	46	Partial User Database Update
6	JSON Data Structures	48	Example of database with minimum required tags necessary to send a database to an ENGAGE device
6	Device Messages	48	Delete Holiday Records
6	Database Overview	49	Delete Auto Unlock Event Records
13	Device Parameters	49	WiFi Commissioning
36	Audit Trail Reporting	49	WiFi Check-in Retry Parameters
36	Asynchronous Events/Alerts Reporting	50	Audit Trail
37	WiFi Device Commissioning	50	Asynchronous Events
38	Keyed-Hashing for Message Authentication (HMAC)	50	Device Settings
39	Examples of User Records (JSON)	52	Persistent Connection JSON Data Structures and IP API
39	deleteAll Tag Set	52	Persistent Connection Edge Device JSON Data structures
39	deleteAll Tag Cleared	54	Gateway Edge Device JSON Data Structures
40	deleteAll Tag Cleared, No User Records to Update	57	Gateway JSON Data Structures
40	partialDb Tag Set, More Than 5000 User Records to Update	66	Appendix A: Credential Tag Name Definitions
41	Example of Holiday Records (JSON)	67	Credential Tags Supported by Device
42	Example of Credential Schedule Records (JSON)	68	Appendix B: Device Specific Characteristics
42	Example of Credential Schedule Records with Two-Factor Authentication (JSON)	69	Appendix C: Type Tags Supported by Device
43	Example of Extended Credential Schedule Records (JSON)	70	Appendix D: Model Type Supported by Device
44	Example of Auto Unlock Records (JSON)		
44	Example of Configuration Records (JSON)		
45	Example of Configuration Records with GPIO Configuration (JSON)		

REVISION CONTROL RECORD			
VER	Date	DESCRIPTION OF CHANGE	AUTHOR
1.86	06/19/2019	- Updates for Reader-Controller; includes changes to: User Records, Credential Schedules, Modifiable Parameters, Retrievable Parameters, Device Control, and Device Status.	C. MacCrindle
		- Changed Gateway JSON Data Structures and IP API section to Persistent Connection JSON Data Structures and IP API	
1.87	09/06/2019	- Updated edgeDeviceFwUrls, extendedStatus and site_survey mdl and model tags with ndeb, leb, controlb, cte and rmrU.	R. Verde
		- Updated fwurl lengths for firmware upgrade messages to Gateway	
		- Minor updates to configuration examples.	
1.88	09/19/2019	- Additional changes for Reader-Controller in retrievable and modifiable tables	C. MacCrindle
1.89	10/31/2019	- Added updateEdgeDeviceTime tag to Gateway device Info. Minor updates to activation/expiration dates.	R. Verde
1.90	03/02/2020	- Added clarification on Change Lock State commands are separate commands and nextLockState momentaryUnlock uses internal relock delay	J. Everson
1.91	03/05/2020	- Updated "pol" note to clarify main relay polarity.	C. MacCrindle
		- Added "ethernetProperties" to retrievable parameters with "assignedIpAddr", "defGatewayIpAddr", and "netmask" child tags	
		- Added "REXwoNotif" tag, Added note to "noc14443" tag. Added note that "auditIDen" defaults to "T" for RC-05	
1.92	05/08/2020	- Added extended schedule "schedulesExt" and associated child tags, added "extSchedEn" configuration tag, updated "gpo" notes, added tmpRRebaseline to lockControl	C. MacCrindle
1.93	06/13/2020	- Updated MTB, MTKB in exclusion table 11, Added MTK, MTKB to support "jsonHmac" extension, Added "null" to Secondary Credential Type, Removed previous versions from revision control record	R. Verde
		- Removed "clsrM" and "dorm" from lock types as value not supported, Updated retrievable parameter dbDwnldTm as the tag doesn't support seconds	
		- Updated Device Exclusions column for various tags, mainly SCB and RC	
1.94	08/17/2020	- Updated RC models, PACS tag values, and increased RC ipServerURL and caServerURL to 128 characters	C. MacCrindle
1.95	08/31/2020	- Updated Device Specific Characteristics table to show RC extended schedules capabilities	C. MacCrindle
1.96	09/21/2020	- Minor updates and additional information of tags. Included Appendix D table on device model type and added example db with minimum required tags for download	R. Verde
1.97	02/02/2021	- Update RC05 changes for new clear override command, activation/expiration credential time resolution and add info on default ASM Key	S. Thaker
1.98	03/02/2021	- Added RMRU to exclusion table for risRtryTm, rsiSubqtDly, and resiFrstDly tags. Updated device name (deviceName, name) to reflect 28 character limit.	R. Verde
		- Added autoHolAudEn tag (New Style of Holidayt and Auto Event Audits) to Modifiable and Retrievable Parameters, Removed "ble" and "biomtrc" as value for prCrTyp and scndCrTyp	
1.99	07/01/2021	- Added note for lockState values supported by Control and ControlB locks.	M. Bhadsalkar
2.0	07/23/2021	- Added Device Profile Tag to indicate HID Secure Read support. Also added Modifiable and Retrievable Parameter for HID Secure Read (i.e. "Smart Card Secure iClass, iClass SE, SEOS")	S. Hegde
2.01	08/02/2021	- Added "Chassis core Parameters" to retrievable parameters for XE360.	Pradyuman Agarwal
2.02	10/11/2021	- Deleted "Smart Card 15693 iClass SE" parameter and refined other credential parameters.	B. Gilchrist
2.03	02/08/2022	- Changed version numbers to start with 2.0, Added autoWBadge tag for RC	R. Verde
2.04	03/07/2022	- Removed device exclusions for "uid15693" and "iClsUID40b" for XE360.	B. Gilchrist

REVISION CONTROL RECORD			
		- Added device exclusions for “configCrdPrsntd” and “customKeyPrsntd” for xe360, added notes for “hwVer” for lockPrmtrs, rdrPrmtrs, and chassisPrmtrs for XE360.	
2.05	03/30/2022	- Added ControlBM Model numbers FE410BM and BE467BM	Chowdappa K.V
2.06	06/02/2022	- Added device profile to indicate support for dynamic MTU	Pradyuman
2.07	10/13/2022	- Removed device exclusions for “uid15693” and “iClsUID40b” for XE360 in modifiable parameter.	
2.08	05/18/2023	- Add mechanical override detection enable/disable config for XE360.	
2.09	08/04/2023	- Add CTE new config items and device file item for main strike/aux relay control	Jane L.
2.10	10/30/2023	- Added blue and amber to LED blink colors for RC11/15. Added note for how the “nxtDbVerTS” tag is used in RC11/15. Added note for GPI and GPO configuration.	R. Dinoso
		- Added row for Device Event in Table 15: Device Status JSON Tags.	
2.11	11/03/2023	- Added XE360 model types in Retrievable parameters.	Pradyuman
2.12	01/17/2024	- Added new config for “credential brute force attack prevention enable” for XE360 devices	Pradyuman
2.13	04/03/2024	- For RC11/15, replaced configurable parameter autoWBadge with autowbadgeSchedIDs. Replaced retrievable parameter autoWBadge with autoWBadgeEn.	R. Dinoso
2.14	06/28/2024	- Added information for nxtDbVerTS tag, included delAuditCmd in the extension key table	R. Verde
2.15	03/24/2025	- Added information for the wifiAlertEn tag related to 410 IP Mode.	Rajesh Vanteru
2.16	11/10/2025	- Added information for wifiCmSh object.	Vimal Jain
2.17	12/04/2025	- Updated for GWE2 model id (Artemis)	Deepa Teggi
2.18	01/05/2026	- Added information on WiFi Check-in Retry parameters	Ram Perumanam

Introduction

Abbreviations Used In This Document	
410-IP	Device, linked to a Gateway equipped with an Ethernet connection
BLE	Bluetooth Low Energy
db	Database
Dbm	Database Manager
ENGAGE™	Connectivity Platform Technology
JSON	Java Script Object Notation
NDE, N	Schlage ENGAGE Wireless Cylindrical Lock
NDEB	Schlage ENGAGE Wireless Cylindrical Lock with BLE Credential capability
LE, L	Schlage ENGAGE Wireless Mortise Lock
LEB	Schlage ENGAGE Wireless Mortise Lock with BLE Credential capability
SC	Schlage Control, Schlage Deadbolt / Interconnected Lock (with ENGAGE)
SCB, be467b, fe410b, be467bm, fe410bm	Schlage Control, Schlage Deadbolt / Interconnected Lock (with ENGAGE) with BLE Credential capability
MTB, MTKB	Schlage Multi-Tech Credential Readers with BLE Credential capability
RC	Reader-Controller
RMRU, R	Von Duprin E-Dog Exit Device (with ENGAGE) Base Device Type
RM	Remote Monitoring – RMRU Variant Supporting Sensors Only
RU	Remote Undogging – RMRU Variant Supporting Sensors and Undogging
CTE, C	Schlage ENGAGE Controller
RTAC	Real-Time Access Control Service
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
XE360	Schlage ENGAGE Wireless(exit trim, tubular, mortise) devices with BLE credential and no tour capability
TBD	To Be Determined

JSON Data Structures

The communication protocol that is used to transfer data directly between the ENGAGE family of devices and the database host device is based on the JSON file format. The database host can be any device that connects to an ENGAGE device through an Ethernet, WiFi, or Bluetooth Low Energy (BLE) connection.

This section defines all possible JSON tags that are used to configure an ENGAGE device, however tags are not applicable to all system configurations or devices. Tags which are utilized only in either the ENGAGE Gateway or ENGAGE API may be defined only in the *ENGAGE – SAM API Integration* documents.

Within the data structure tables, there are parent tags which do not include any data elements. Parent tags are implemented because they allow for the re-use of common tag names within a different parent tag context. An example of a common tag name is the tag “days”. This tag is common to the credential schedule and auto unlock parent tags. Another example is within the user records data structure. The “add”, “update” and “delete” parent tags share parameters that are specific elements with an individual user record. Parent tags within the tables are shown in **Bold Text** to differentiate from Child records.

For information regarding the JSON standard, the web site www.json.org contains many helpful links. The JSON specification can also be found at this link: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>.

Device Messages

There are four (4) main categories for JSON data structures that are sent to and from a host system (access control panel or SW system) and the security device located at the entryway (lock or other device). These categories are:

1. Database
2. Configuration
3. Audit Trail
4. Asynchronous Events / Alerts

Data transfers between a host and device will occur within a “session”. The session contains data that is formatted to the JSON specification. The JSON “files” are sent over BLE, WiFi, or Ethernet to an embedded device that has limited RAM. As a result, the JSON data files are written to the lock’s flash memory and parsed after the data transfer has completed.

➔ **NOTE:** ENGAGE Edge devices support only standard English language ASCII characters less than 0x7F.

Database Overview

During a connected event when a device is either requesting a database update from the host directly over WiFi/Ethernet or when the database is being updated via BLE, the device can receive a database update. The database received by the device can contain user records, credential schedules, holidays, and auto unlock data sent by the host.

The database consists of several different sub-components. The sub-components are user records, holidays, user schedules and auto unlock events. The parent tag of the database is expressed with the JSON “db” tag.

From these sub-components, only individual user records can be modified during a database update. This is accomplished through the “deleteAll”, “delete”, “update” and “add” tags.

If the holidays, user schedules or auto unlock records need to be added or updated, the records should be sent as a whole. For these records, no individual records can be deleted or updated. When the holidays or auto unlock records need to be completely erased from the lock, the tag is sent with an empty “[]” collection. (i.e. “holidays”: [] or “autoUnlock”: [])

When a database update occurs for devices with user credential capabilities, at least one credential schedule must always be sent with the default schedule of access 24 hrs x 7 days / week. An empty collection for credential schedules is not possible. If the host fails to send the default 24x7 credential schedule, even valid credentials will not gain access.

Configurations can also be included with the database but they use a separate JSON parent tag structure.

Database Request Records

The “nxtDbVerTS” tag is used to keep the host synchronized with the ENGAGE device. This timestamp is inserted into the URL by the ENGAGE device on each database request. When the ENGAGE device requests a database for the very first time, it does so with respect to a value of “0” for the “nxtDbVerTS” tag. The host responds by sending an initial database, and it includes an updated value for the “nxtDbVerTS” tag. When the lock makes the next database request, it includes the updated nxtDbVerTS in the URL. The value of this field is essential when updating the device database as it keeps track of database state. The “nxtDbVerTS” tag needs to be included at the end of the database file. This process continues until a device reset occurs.

If the ENGAGE device is a networked offline device (210 WiFi Mode), the host must include in the database records the specific time that the device will request the next database. This is accomplished through the “dbDwnLdTm” database download time. See **Table 1**.

Table 1: Database Download Records

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Database Download Time	dbDwnLdTm	String/14	YYYYMMDDHHMMSS – “20140704030000” July 4, 2014, 3:00:00 am	SC, SCB, MTB, MTKB
Next Database Version Time Stamp	nxtDbVerTS	String/18	Hexadecimal value, that starts with 0x, followed by 16 hex digits. (i.e. “0x08d16357eef73a69”)	MTB, MTKB

➔ NOTE: For RC11/15, the “nxtDbVerTS” tag is used differently from other ENGAGE devices. The value associated with the tag is a sequential number that starts at 0. The tag-value pair can be sent by the access control software to the RC during a database update. If the tag-value pair is provided, the RC stores the value in non-volatile memory. The value can be polled by the access control software by issuing the command /edgeDevices/[device name]/params. The value provides an indication to the host access control software whether a previous database update was successful or not.

User Records

User records come to the device through a “new” or an “updated” database. This ability is controlled through the “deleteAll” tag. When the “deleteAll” tag value is set to “1”, the device gets a new user database. When this value is set to “0”, the device gets an update to the existing user database.

For the reader-controller, a user database of up to 64,000 records is required. For the device to support 64,000 records, a mechanism for multiple transfers of up to 1000 records at a time is required. This is accomplished using the “partialDB” tag. The “partialDB” value starts at 1 and is incremented for each group of up to 1000 records. While groups of 1000 users can be sent, 300 users per partial DB file is recommended. Exceeding 1000 users per partial DB file can result in long processing and response times. The final group of records is sent with a “partialDB” value of “65535” denoting that this is the final transfer.

➔ Note: when the “partialDB” tag is used, it takes priority over the “deleteAll” tag (ie “deleteAll” is ignored if “multi” is non-zero). A “partialDB” value of “0” indicates that there is not a multiple transfer database being sent.

There are instances when a completely new database is sent to the device. This happens under two conditions. When the:

- ENGAGE device is programmed for the very first time.
- Host device determines that the differences between the device’s existing database and the new database are too numerous.

When the host sends user records to the device, the records are grouped in terms of user records that will be deleted, updated, or added to the device. The order in which these records are sent from the host must be in that same order: delete, update and add. All four of the tags (deleteAll, delete, update, add) must be included inside the db tag, even if there is no applicable user record changes to the respective tag. When user records need to be updated within the ENGAGE device, it is logical that the device has knowledge of which records need to be deleted before updating and adding new user records.

For the purposes of updating a user record, the entire user record is sent from the host. The record includes data that needs to be updated along with data that does not need to be updated.

When an individual user record needs to be deleted, only the primary credential ID and primary credential type is sent as the identifier for that record.

A user record describes the parameters associated with each user of the access control device. A complete user record is delimited with several different JSON tags. The individual tags within a User Record are described in **Table 2**.

The order defined in the usrRcrd tag defines the searching order for the lock. As user records are modified, these are appended to the devices database.

The credential records must be encrypted as specified in the **Schlage ENGAGE Sort & Encryption** document.

Table 2: User Record

Tag	JSON Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Database Block	db	String/2	N/A	
User Record Block	usrRcrd	String/7	N/A	RMRU, MTB, MTKB
Delete All	deleteAll	Int/1	1 = Delete entire existing database (For SC and SCB, deleteAll does not delete blocked credentials, blocked credentials must be unblocked first before they can be deleted with deleteAll)	RMRU, MTB, MTKB
			0 = Do not delete entire existing database	
Partial Database Transfer Counter	partialDb	Int/5	0 = ignore, 1-65534 = incrementing counter, 65535 = final transfer	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Delete User Record	delete	String/6	N/A	RMRU, MTB, MTKB
Update User Record	update	String/6	N/A	RMRU, MTB, MTKB
Add User Record	add	String/3	N/A	RMRU, MTB, MTKB
UserID	usrID	Unsigned Int/7	1-1,048,575 record number identifier	RMRU, MTB, MTKB
ADA enable	adaEn	Int/1	0 = ADA Disabled 1 = ADA Enabled	SC, ,SCB, RMRU, MTB, MTKB
Credential Function	fnctn	String/8	"norm", "freeze", "toggle", "passThru", "lockDown", "oneTm", "dogd", "suprv", "delAlrm", "ctAux", "ctMnAux", "ctMnAuxToggle", "ctMnAuxPassThru", "ctAuxToggle", "updtFrmSvr", "updtFrmSvrNrm", "blocked"	RMRU, MTB, MTKB See Appendix A, Section 5.1 for per device compatibility
Credential Schedule	crSch	Array of Unsigned Int/2s	1 through N, where N is the maximum number of credential schedules the device allows. At a minimum, a user record will have a credential schedule of "[1]", which means the credential will have 24-7 access.	RMRU, MTB, MTKB
Activation Date, Time	actDtTm	String/14	YYYYMMDDHHMMSS – "20140706130000" July 6, 2014, 1:00:00 pm. Minimum date/time is 20100101000000 Note: All devices have resolution to hours, except Control (seconds) The absolute maximum value for the date is "21351231235959"	RMRU, MTB, MTKB
Expiration Date, Time	expDtTm	String/14	YYYYMMDDHHMMSS – "20140924150000" Sept 24, 2014, 3:00:00 pm. Minimum date/time is 20100101000000 Note: All devices have resolution to hours, except Control (seconds) The absolute maximum value for the date is "21351231235959"	RMRU, MTB, MTKB
Primary Credential	primeCr	String/32	Credential data in hexadecimal format. The contents of this tag must be encrypted. See ENGAGE Sort & Encryption document for more information. PIV credentials may need 200 bits (25 bytes) of data. However, PIV is not supported by ENGAGE devices.	RMRU, MTB, MTKB
Primary Credential Type	prCrTyp	String/7	"card", "pin", "null" Note The Schlage Mobile Credential mimics a physical credential, use "card" as the credential type for the Mobile Credential.	RMRU, MTB, MTKB
Secondary Credential Type	scndCrTyp	String/7	"card", "pin", "null" Note The Schlage Mobile Credential mimics a physical credential, use "card" as the credential type for the Mobile Credential.	SC, ,SCB, RMRU, MTB, MTKB

➔ **NOTE:** All credential attributes can be **updated** except for the primary credential value. Changing primary credential value requires **deletion** of user record with the old primary credential value followed by **addition** of new user record with the new primary credential value.

Credential Schedule Records

Credential Schedule Records are used to describe a set of times when an individual credential can be granted access. These times are described by a specific day of the week and time, not by a specific calendar date. These records are stored in the device and are applied to individual user records through the “Credential Schedule” tag within individual user records.

Table 3 describes each of the fields associated with a credential schedule.

Devices with Credential Readers

In order for a credential to have access to a device, at least one credential schedule must be programmed into the device.

If an existing credential schedule needs to be deleted from the lock, then the host must send at least one credential schedule that allows access 24 hrs / day, 7 days a week.

The credential schedules must be explicitly defined for every database structure provided to the lock, even if previously defined.

Table 3: Credential Schedule Record				
Tag	JSON Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Database Block	db	String/2	N/A	
Credential Schedule Block	schedules	String/9	N/A	RMRU, MTB, MTKB
Days	days	Array of String/2s	[“Su”, “Mo”, “Tu”, “We”, “Th”, “Fr”, “Sa”]	RMRU, MTB, MTKB
Start Hour	strtHr	Unsigned Int/2	Hour, where a start hour (Hr) can be 0-23.	RMRU, MTB, MTKB
Start Minute	strtMn	Unsigned Int/2	Minute, where Minute can be 0-59.	RMRU, MTB, MTKB
Length	lngth	Unsigned Int/4	Duration of the schedule, in minutes (decimal), (1 – 1439).	RMRU, MTB, MTKB
			Value 1439 represents 23:59:59. All other values represent the minute. Value 1438 represents 23:58:00. Value 479 represents 7:59:00.	
Authentication Mode	auth	String/4	“1f” (default), “2f”, “1fx2”, “2fx2”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Extended Credential Schedule Block	schedulesExt	String/12	N/A. Note: The device will use either standard credential schedules or extended credential schedules. See extSchedEn tag to enable/disable extended credential schedules.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Extended Schedule ID	extScheduleID	Unsigned Int/2	1-32 extended schedule identification number	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Interval ID	intvID	Unsigned Int/2	1-32 interval identification number. Each extended schedule record can have up to 32 intervals.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Sunday Start Time	suStrt	String/4	“HHMM” HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Sunday End Time	suEnd	String/4	“HHMM” HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Sunday Authentication Mode	suAuth	String/4	“1f” (default), “2f”, “1fx2”, “2fx2”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 3: Credential Schedule Record

Tag	JSON Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Monday Start Time	moStrt	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Monday End Time	moEnd	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Monday Authentication Mode	moAuth	String/4	"1f" (default), "2f", "1fx2", "2fx2"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Tuesday Start Time	tuStrt	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Tuesday End Time	tuEnd	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Tuesday Authentication Mode	tuAuth	String/4	"1f" (default), "2f", "1fx2", "2fx2"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Wednesday Start Time	weStrt	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Wednesday End Time	weEnd	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Wednesday Authentication Mode	weAuth	String/4	"1f" (default), "2f", "1fx2", "2fx2"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Thursday Start Time	thStrt	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Thursday End Time	thEnd	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Thursday Authentication Mode	thAuth	String/4	"1f" (default), "2f", "1fx2", "2fx2"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Friday Start Time	frStrt	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Friday End Time	frEnd	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 3: Credential Schedule Record

Tag	JSON Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Friday Authentication Mode	frAuth	String/4	"1f" (default), "2f", "1fx2", "2fx2"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Saturday Start Time	saStrt	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Saturday End Time	saEnd	String/4	"HHMM" HH = Hour (00 – 24) MM = Minutes (00 – 59)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Saturday Authentication Mode	saAuth	String/4	"1f" (default), "2f", "1fx2", "2fx2"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Holiday Records

Holiday records are events that describe when a device should be put into a special state based upon a certain calendar date and time, overriding an auto-unlock schedule. A secured holiday puts the device into the normally secured state. A passage holiday puts the device into a normal passage state. The restricted secured holiday puts the device into a secured state, but the device only accepts 2 card types in this state, pass-through and freeze.

When in a restricted secured state if a freeze credential is presented, the device transitions from restricted secured to a normal secured state – allowing the expanded credential types to be allowed at the door. To restore the device to the restricted secured state, a freeze credential must be presented two more times; (the first takes the device to frozen secured state, the second triggers a schedule look-up causing the device to resume restricted secured state). The pass-through credential functions as expected, momentarily unlocking the door.

In the event that the existing holiday record needs to be deleted from the device, and not replaced with a new set of holidays, then an ID empty tag of "[]" is sent from the host.

Holiday records must be explicitly defined for every database structure provided to the device, even if previously defined.

Table 4 describes each of the fields associated with a holiday record.

Table 4: Holiday Record Fields

Tag	JSON Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Database Block	db	String/2	N/A	SC, SCB, MTB, MTKB
Holiday Block	holidays	String/8	N/A	SC, SCB, MTB, MTKB
Start Date / Time	strtDtTm	String/14	YYYYMMDDHHMMSS – "20131224000000", Dec 24, 2013, 00:00:00 (Midnight) NOTE: NDE and RC has resolution to the minute.	SC, ,SCB, MTB, MTKB
End Date / Time	endDtTm	String/14	YYYYMMDDHHMMSS – "20131227080000" Dec 27, 2013, 8:00:00 am NOTE: NDE and RC has resolution to the minute.	SC, SCB, MTB, MTKB
Action	action	String/9	"pass", "sec", "rstrctSec"*	SC, SCB, MTB, MTKB rstrctSec not supported on RMRU

Auto Unlock Records

Auto unlock records are device specific events that describe when the device should be put into secure or passage mode based upon a specific day of the week and time. An auto-unlock record only describes when the device should change its current state. In order to put the device back to its original state, an additional auto unlock record is required. For many doors, a good “rule of thumb” is to have an auto-lock schedule, scheduled for midnight. This way if someone toggles a device unlocked during the day, the door will relock itself at midnight.

In the event that the existing auto unlock record needs to be deleted from the device, and not replaced with a new set of auto unlock events, then an ID empty tag of “[]” is sent from the host.

Auto unlock must be explicitly defined for every database structure provided to the device, even if previously defined.

Table 5 describes each of the fields associated with an auto unlock record.

Table 5: Auto Unlock Record Fields				
Tag	JSON Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Database Block	db	String/2	N/A	SC, SCB, MTB, MTKB
Auto Unlock Block	autoUnlock	String/10	N/A	SC, SCB, MTB, MTKB
Action	action	String/4	“pass” “sec”	SC, ,SCB, MTB, MTKB
Start Hour	strtHr	Unsigned Int/2	Hr, where HR can be 0 - 23	SC, ,SCB, MTB, MTKB
Start Min	strtMn	Unsigned Int/2	Min, where Min can be 0 - 59	SC, ,SCB, MTB, MTKB
Day	days	Array of String/2s	[“Su”, “Mo”, “Tu”, “We”, “Th”, “Fr”, “Sa”]	SC, ,SCB, MTB, MTKB

Device Parameters

This section describes the data structures for modifying and retrieving device parameters.

Modifiable Parameters

Some device parameters can be changed by a host device. When modifying (configuring) parameters, a host sends a JSON data structure to the device that contains a set of configuration parameters. Table 6 shows a complete list of the JSON data tags and tag values that can be modified by a host. Some of the parameters are device specific and may not apply to all devices.

Modifiable (configurable) parameters vary by each device's capability or the specific configuration of the device.

To support a standard of JSON configurations moving into the future, the standard has been established to use "T" / "F" for configuration tags/items (to enable or disable a feature). However, because some of the currently established tags do not use this standard, the previously implemented tag types maintain their current implementation and the standard will be established for new tags implemented in the future.

See the **Example of Configuration Records (JSON)** section below for an example of this data structure.

Table 6 shows a listing of Modifiable (Configurable) Parameters.

Table 6: Modifiable (Configurable) Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Configuration Block	config	String/6	N/A	
Name	name	String/28	"xxxx.....xxx"	
Type	type	String/6	"strm" "office" "apt" "prvcy"	SC, ,SCB, RMRU, MTB, MTKB
Relock	relock	Unsigned Int/3	Delay time in seconds (1-60s for SC, 1-30 for all other supported products)	RMRU, MTB, MTKB
Central Decision Mode Timeout	cntrlDecTimeout	Unsigned Int/4	Delay time in deciseconds (0-6540) NOTE: 0=central decision mode disabled (local decision mode enabled)	SC, SCB, NDE, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, RC, XE360
Credential Sector Number	credSectNum	Unsigned Int/2	Credential Sector Number (1-15)	NDE, LE, SC, ,SCB, CTE, RMRU, NDEB, LEB, MTB, MTKB, RC, XE360
Jaguar Sectors	jagSectors	Unsigned Int/5	Jaguar Sector Bitmask (0-65535)	RMRU, MTB, MTKB, RC
Lock ID	lockId	Unsigned Int/5	Door ID (0-65534) 65535 disables no tour on LE locks	RMRU, MTB, MTKB, RC
Group IDs	groupIds	UnsignedInt/5	Group ID (0-65535) array with up to 16 elements	RMRU, MTB, MTKB, RC
Door Prop Delay	doorProp	Unsigned Int/3	Delay time in seconds (1-255)	SC, ,SCB, MTB, MTKB
Propped Door Delay Enable	doorPropEn	String/1	"T","F"	NDE, SC, ,SCB, LE, RMRU, MTB, MTKB, NDEB, LEB, XE360
ADA Delay	ada	Unsigned Int/3	Delay time in seconds (1-255)	SC, ,SCB, RMRU, MTB, MTKB
First Man In	firstManIn	Int/1	0=No First Man In 1= First Man In	SC, ,SCB, RMRU, MTB, MTKB
Dog On Next Exit	dneEn	Int/1	0=Do not use Dog on Next Exit 1=Schedules Dog on Next Exit	NDE, SC, ,SCB, LE, CTE, MTB, MTKB, NDEB, LEB, RC, XE360

Table 6: Modifiable (Configurable) Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Battery Fail State	battFail	String/4	“asls”, “safe”, “sec”	SC, ,SCB, RMRU can only operate as sec, CTE, MTB, MTKB, RC
Blink Interior LED when locked	blinkIntLed	String/1	“T” = enabled , “F” = disabled (default)	NDE, SC, ,SCB, CTE, MTB, MTKB, RC
Privacy Rapid blink	rapidBlink	String/1	“T” = enabled , “F” = disabled (default)	NDE, SC, ,SCB, RMRU, CTE, MTB, MTKB, RC
IPB Audit Enable / Disable	ipbAuditEnable	String/1	“T” = enabled , “F” = disabled (default)	NDE, SC, ,SCB, LE, RMRU, CTE, MTB, MTKB, RC, NDEB, LEB, XE360
Daylight Savings Time	dstEnable	Int/1	0=DST not observed 1=DST observed	MTB, MTKB
DST Start	dstStart	String/4	Hour that clocks are moved ahead by 1 hr. 3=standard month start (1=Jan,...) 0=standard day of week start, 0 = Sun.... 2=standard week # start (1=1st,..5=last) 2=standard hour start US default is 3022	MTB, MTKB
DST End	dstEnd	String/4	Hour that clocks are moved behind by 1 hour. B=daylight month start 0=daylight day of week start 1=daylight week # start 2=daylight hour start US default is B012	MTB, MTKB
Clock Time	rtcTime	String/14	“YYYYMMDDHHMMSS” YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) HH = Hour (00 – 23) MM = Minutes (00 – 59) SS = Seconds (00 – 59) Example: “20140710145030” = July 10, 2014, 2:50:30 pm. NOTE 1: this message affects any date/time event (including FW upgrade) so it needs to precede the fwDownldTm and fwImplTm tags. NOTE 2: in order to limit flash write operations, the current rctTime of a device will not be changed if it is already within +/- 5mins of the specified time. For ENGAGE 6.0 Firmware and later this value has been changed to 1 minute for all ENGAGE products.	MTB, MTKB
Immediate WiFi Alert Enable	wifiAlertEn	String/1	“T” = enabled (default) , “F” = disabled, “S” = limit immediate alert reporting to the selection made in wifiAlertSel. NOTE 1: Conflicting settings between this tag and wifiAlertSel (e.g. a list of alerts specified in wifiAlertSel but wifiAlertEn is “F”) will result in the last tag processed being used. NOTE 2: WiFi-related parameters are not applicable in 410 IP mode.	SC, SCB, MTB, MTKB, RC, XE360
Individual WiFi Immediate Alert Selection	wifiAlertSel	Array of String/6s	<i>See separate document for Audit & Alert Definitions (Alert Type Only)</i>	SC, SCB, MTB, MTKB, RC, XE360

Table 6: Modifiable (Configurable) Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Firmware Address	fwurl	String/64	Max 64 characters NOTE 1: The config PUT from the lock presents a different tag “fwUrl” than what it consumes “fwurl”. NOTE 2: NDE Firmware revisions < 02.06.12 only support a fixed url endpoint: http://<host>/api/firmware/latest It is suggested that a symbolic link or copy of the latest NDE binary firmware be placed at this endpoint if supporting older NDE firmware versions <02.06.12. When using an ENGAGE mobile application to connect to an NDE lock running these older firmware versions, the ‘Update Firmware’ option will be visible and will command these locks to the fixed endpoint for ease of upgrade and to allow updating of a lock to a firmware revision that supports root certificate change.	SC, SCB, MTB, MTKB
Firmware download Time	fwDwnldTm	String/14	YYYYMMDDHHMMSS – “20140527040000” May 27, 2014, 4:00:00 am Value of “0” will result in immediate download. NOTE: All revisions of NDE, LE, and SC only support a value of “0” (immediate download) currently.	SC, SCB, MTB, MTKB
Firmware Implement/ Update Time	fwImplTm	String/14	YYYYMMDDHHMMSS – “20140603030000” June 3, 2014, 3:00:00 am Value of “0” will result in immediate update/implement. NOTE: Prior to ENGAGE 6.0 revisions of NDE, LE, and SC only support a value of “0” (implement immediately) currently. ENGAGE 6.0 and later process this tag as expected.	SC, SCB, MTB, MTKB
DPS Audits Enable	dpsEn	String/1	“T”, “F”	SC, SCB, MTB, MTKB
Beeper En	bprEn	String/1	“T”, “F”	
Reader Sensitivity	rdrSense	String/4	“norm”, “high”, “max”	SC, SCB, RMRU, CTE, MTB, MTKB, RC, XE360
Prox Config HID	proxConfHID	String/1	“T”, “F”	SC, SCB, RMRU, XE360
Prox Config GECASI	proxConfGECASI	String/1	“T”, “F”	SC, SCB, RMRU, XE360
Prox Config GE4001	proxConfGE4001	String/1	“T” = 4001, “F” = 4002	SC, SCB, RMRU, XE360
Prox Config GE4002	proxConfGE4002	String/1	“T” = 4002, “F” = 4001	SC, SCB, RMRU, XE360
Prox Config AWID	proxConfAWID	String/1	“T”, “F”	SC, SCB, RMRU, XE360
Smart Card 14443 UID	uid14443	String/1	“T”, “F”	SC, SCB, RMRU
Smart Card 14443 MiFare	mi14443	String/1	“T”, “F”	SC, SCB, RMRU
Smart Card 14443 MiFare Plus	mip14443	String/1	“T”, “F”	SC, SCB, RMRU, XE360
Smart Card 14443 Mifare DESFire EV1/2/3	noc14443	String/1	“T”, “F” NOTE: This setting enables/disables DESFire EV1, EV2, and EV3 credentials.	SC, SCB, RMRU
Smart Card 15693 UID	uid15693	String/1	“T”, “F”	SC, SCB, RMRU

Table 6: Modifiable (Configurable) Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Smart Card iClass UID 40 bit	iClsUID40b	String/1	"T","F" NOTE: "Smart Card iClass UID 64 bit" must be set to "F" to enable this parameter to be "T".	SC, SCB, RMRU
PIV Configuration	pivCnfg	String/11	"disbld","75bPIV", "58bTWIC_CAC", "14443PIV200b","64bBCD_TWIC","83bTWIC_CAC", "66bTWIC_CAC", "64bTWIC", "91bTWIC_CAC", "40bBCD", "40bRevBCD", "64bBCD", "64bRevBCD", "128bBCD", "128bRevBCD", "58bAMAG"	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Smart Card iClass UID 64 bit	iClsFrmt	String/11	"disbld", "64bCSN" NOTE: "Smart Card iClass UID 40 bit" must be set to "F" to enable this parameter to be "64bCSN".	SC, SCB, RMRU, XE360
Smart Card Secure iClass, iClass SE, SEOS	iClsSec	String/1	"T" = enabled (default), "F" = disabled NOTE: Setting applicable only for HID variant of NDEB and LEB locks	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
BLE Credential Enable	bleCredEn	String/1	"T", "F"	RMRU, NDE, LE, SC
BLE Credential Range	bleCredRng	String/5	"short", "long"	RMRU, NDE, LE, SC
BLE Performance	blePerf	String/4	"norm", "max"	RMRU, NDE, LE, SC, MTB, MTKB, CTE, RC
Mag Card Reader Track 1 Enable	magtrk1	String/1	"T","F"	NDE, SC, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, SCB, RC, XE360
Mag Card Reader Track 2 Enable	magtrk2	String/1	"T","F"	NDE, SC, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, SCB, RC, XE360
Mag Card Reader Track 3 Enable	magtrk3	String/1	"T","F"	NDE, SC, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, SCB, RC, XE360
Mag Card Reader Low Power Enable	magLwPwrEn	String/1	"T","F"	NDE, SC, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, SCB, RC, XE360
Standby LED Color	ledStbyColor	String/5	"off", "red", "green", "amber"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, XE360
Standby LED State	ledStbyState	String/5	"solid", "flash"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, XE360
Credential Read LED	ledCredRd	String/5	"none", "green"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, CTE, XE360
Backlight Timeout	bckLghtTmOut	Unsigned Int/2	0-15, 0 = disabled	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, CTE, MTB, MTKB, XE360

Table 6: Modifiable (Configurable) Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Keypad Output Format	keypadOutput	String/13	"4BitNP", "8BitNP", "Galaxy26A", "PCSC4", "PCSC5"	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, CTE, RC, XE360
Keypad facility code	kpFclCode	Unsigned Int/3	0-255	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, CTE, RC, XE360
Number of key presses to buffer/cache	kpBuffCache	Unsigned Int/2	0-15	NDE, SC, SCB, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, RC, XE360
Keystroke Timeout	keystrokeTO	Unsigned Int/2	0-15, 0 = disabled	NDE, SC, SCB, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, XE360
PIN Length	pinLength	Unsigned Int/2	3-16, default 6; RC: 0-9, 0 is variable PIN length (user must enter start (*) and stop (#)), acceptable PIN lengths are 1-9	NDE, SC, SCB, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, XE360
PIN enable (ignore keypad)	pinEn	String/1	"T","F"	NDE, SC, SCB, LE, RMRU, CTE, MTB, MTKB, NDEB, LEB, XE360
Clock and Data Format	rdrOutFmt	String/8	"dsbld", "format1", "format2", "format3", "format4", "format5", "format6", "format7", "format8", "format9", "format10", "format11", "format12"	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, MTKB, CTE, RC, XE360
Reader output Format	rdrOutput	String/8	"wiegand", "clkndata"	NDE, NDEB, SC, SCB, LE, LEB, RMRU, CTE, RC, XE360
Anti-Tailgate (immediate relock on door close)	immRelockEn	String/1	"T" = enabled, "F" = disabled (default)	NDE, SC, SCB, LE, RMRU, NDEB, LEB, MTB, MTKB, XE360
410-IP Communication Failsafe Mode	gatewayCommFail	String/4	"asls", "sec", "safe"	NDE, SC, SCB, LE, CTE, RMRU does not support "safe", MTB, MTKB, NDEB, LEB, RC, XE360
Tamper Cover Removal Failsafe Mode	tamperFail	String/4	"asls", "sec", "safe"	NDE, SC, SCB, LE, CTE, RMRU does not support "safe", MTB, MTKB, NDEB, LEB, RC does not support "safe", XE360

Table 6: Modifiable (Configurable) Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Invalid Card Presented Audit	invCrdAudEn	String/1	“F”=Do not create an audit for invalid card presentations (Default). “T”= Create an audit for invalid card presentations. NOTE: if this configuration is enabled the edge device audits should be checked regularly to ensure that the edge device audit buffer is not inadvertently filled with invalid card presentation audits. NOTE: The default for RC-05 is “T”. Since the device reports audits in real time it is not necessary to check the device audits regularly.	RMRU, MTB, MTKB
WiFi Download Retry Time Interval	wifiDwnldRtyTmIntv	Unsigned Int/4	5-1440 Default Value Setting is 1440 If WiFi event fails, this setting is the number of minutes after the failure that the device should re-try the WiFi event. The retry will continue wifiDwnldRtyCnt times every wifiDwnldRtyTmIntv minutes.	SC, SCB, MTB, MTKB, RC, XE360
WiFi Download Retry Counter	wifiDwnldRtyCnt	Unsigned Int/3	1-250 Default Value Setting is 1 If WiFi event fails, this setting is the number of times after the failure that the device should re-try the WiFi event. The retry will continue wifiDwnldRtyCnt times every wifiDwnldRtyTmIntv minutes.	SC, SCB, MTB, MTKB, RC, XE360
Audit ID Enable	auditIDEn	String/1	“F”= The lock will not send the new unique audit identifier with the audit json (default). “T”= The lock will send the new unique audit identifier with the audit JSON. NOTE: The default for RC-05 is “T”.	MTB, MTKB
New Style of Holiday and Auto Event Audits Enable	autoHolAudEn	String/1	“F”= The lock will continue to generate Engage legacy style of holiday and auto event audits (default). “T”= The lock will generate new style of holiday and auto events audits. NOTE: The default for RC-05 is “F”.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Prevent Construction Mode	blkCnstr	String/1	“T” = enables prevent construction mode. “F” = disables prevent construction mode (default option).	RMRU, SC, SCB, MTB, MTKB
REX Without Notification	REXwoNotif	String/1	“T” = RC does not change LED state for REX, “F” = RC changes LED state for REX	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Advanced Security Module (ASM) Enable	asmEn	String/1	“T”, “F”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
ASM Key	asmKey	String/8	4 byte key, default key 0x36353136	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Beep on Tamper	tmprBp	String/1	“T”, “F”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Tamper Sensitivity	tmprSens	String/4	“low”, “med”, “high”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 6: Modifiable (Configurable) Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Network Timeout	ntwkTO	UnsignedInt/3	Time in seconds (0-30), 0 (default) = decision at door (DaD); NOTE: A non-zero value prescribes Decision at Host (DaH) and the value specifies the amount of time since the last successful communication with the host before the device transitions to DaD.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Extended Credential Schedules Enable	extSchedEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Two-Factor Timeout	twoFctrTO	UnsignedInt/3	Time in seconds (0-30), 10s (default)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Auto-unlock With Badge	autowbadgeSchedIDs	Array	[] = disabled, [{1, x, y, z, ..., 32}] = array of integers representing schedules which support Auto-unlock with Badge (Any number of integers between 1 to 32 are accepted.)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Mechanical Override detection Enable	mechOverrideEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB
Credential Brute Force Attack Prevention Enable	credBFAPrevEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB
General Purpose Inputs	gpi	Array	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Input Type	iType	String/5	"rex", "auxIn", "dps"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Input Enable	iEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Signal Polarity	pol	String/4	"high", "low"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
General Purpose Outputs	gpo	Array	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Output Type	oType	String/7	"mainRly", "ttl1", "ttl2"; NOTE: The main relay is included here to specify whether the main relay will be activated on a REX or Aux Input.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Output Enable	oEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 6: Modifiable (Configurable) Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Signal Polarity	pol	String/4	“high”, “low” NOTE: Main relay is included to allow for control of fail safe and fail secure devices.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Mode	mode	String/4	“puls” (default), “hold” NOTE: When the mainRly source is auxIn, mode can be pulse, where the duration is the rellock, or hold. When the mainRly source is rex the mainRly unlocks for the duration of the rellock.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Source	src	Array of Strings/9	[“mirrorRly”, “rex”, “auxIn”, “auxCred”, “tamper”, “frcdDoor”, “prppdDoor”] NOTE: The mirrorRly does not apply to the mainRly	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
New Access Point Config	apCfgNew	String/6	N/A	SC, SCB, MTB, MTKB, RC, XE360
New AP Config Enable	apCfgNewEn	String/1	“T” = enables new Access Point Config. “F” = disables new Access Point Config (default option).	SC, SCB, MTB, MTKB, RC, XE360
New SSID	ssidNew	String/32	Max 32 characters	SC, SCB, MTB, MTKB, RC, XE360
New AP Password	psswdNew	String/64	Max 63 characters	SC, SCB, MTB, MTKB, RC, XE360
New User Name	usrNmNew	String/16	Max 16 characters	SC, SCB, MTB, MTKB, RC, XE360
New WiFi Security	wifiSecNew	String/7	“prsnl”, “entrprs”, “open”, “wep”	SC, SCB, MTB, MTKB, RC, XE360
Config Start Time	apCfgNewStrtTm	String/14	“YYYYMMDDHHMMSS”	SC, SCB, MTB, MTKB, RC, XE360
Ethernet Configuration	ethConfig	String/9	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Host Connection Type	hostProtocol	String/9	“rs485Rsi”, “ipEngage”, “ipClient”; NOTE: RC only supports ipClient	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Interface ID	interfaceld	String/4	“eth0”, “wlan0”; NOTE: RC only supports eth0	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Interface Enable	interfaceEnable	String/1	“T” = enables interface operation, “F” = disables interface operation	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Discovery Method	discoveryMethod	String/6	“dhcp”, “staticIP”, “zeroConf”; NOTE: RC supports dhcp and staticIP	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Fixed IP Address, (only applies if discovery method is set to staticIP)	fixedIpAddr	String/15	“xxx.xxx.xxx.xxx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Default Gateway IP Address (only applies if discovery method is set to staticIP)	defGatewayAddr	String/15	“xxx.xxx.xxx.xxx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Netmask (only applies if discovery method is set to staticIP)	netmask	String/15	“xxx.xxx.xxx.xxx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
IP DNS Service	ipDnsAddr	String/15	“xxx.xxx.xxx.xxx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Alternate IP DNS Service	altDnsAddr	String/15	“xxx.xxx.xxx.xxx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
IP Server URL (only applies if hostProtocol is ipClient)	ipServerURL	String/128	Max 128 characters	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
CA Server URL (only applies if hostProtocol is ipClient)	caServerURL	String/128	Max 128 characters	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Keep Alive Time (only applies if hostProtocol is ipClient)	wsKeepAlive	String/4	“1” to “3600” seconds	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Activate Aux Relay	“auxRly”	String/7	“nothing”, “autoOp”, “mntrSys”, “other”	LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Auto Operate Aux Delay Time (MS)	“autoOpDlyTm”	Number	500-1500 (default 1500)	LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Main Strike Select	“strkSel”	String/10	“elctrcStrk” (default), “elctrcExit”, “elctrfdLck”, “magLck”, “other”	LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
DPS mask time (Seconds)	“dpsShntTm”	Number	0 - 254 (2 - default, 255 - reserved)	LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Retrievable Parameters

When a host makes a request to the device for information that resides in the device, it receives all of the retrievable parameters for that device at that time.

Device Profile

Later firmware versions of ENGAGE devices include a section that defines the type and any special type variances (called extensions) supported by the device as part of a dvcProfile block contained at the beginning of the config block.

To support a standard of JSON configs moving into the future, the standard is for all device profiles to be implemented as integers. However, because some of the currently established tags do not use this standard, the previously implemented tag types maintain their current implementation and the standard will be established for new tags implemented in the future.

Table 7: Currently Established Non-Integer Tags

Tag	Short Tag	Type/Length (ASCII bytes)	Value
Device Profile	dvcProfile	Section Heading	N/A
Base Type	baseType	String/8	"rmru", "nde", "le", "mt20w", "cte", "be467", "gwe", "gwe2", "ndeb", "leb", "be467b", "be467bm", "mtb", "mtkb15", "rc05", "XE360"
Extensions	ext	Array	Array of structures of following types:
Extension's name	key	String/15	<i>See table below for Extension key, custom argument (value) combinations</i>
Extension custom argument	value	String/15	

Extensions and their supported custom arguments.

Extension key	Extension value	Device Associated
undog	<i>empty</i>	rmru
WiFiTest	0 = unsupported; 1 = Full Update Server; 2 = Fast Connection Attempt	nde, le, rmru, mt20w, cte, ndeb, leb
jsonHmac	0 = unsupported; 1 = HMAC Rev 0 Supported	nde, le, sc, rmru, cte, ndeb, leb, scb, rc, mtb, mtkb, mtkb15, XE360
dupAuditDet	0 = unsupported; 1 = Duplicate Audit Detection supported (auditId provided with audit JSON)	nde, le, sc, rmru, cte, ndeb, leb, scb, rc, XE360
fwUrlMethod	0 = unsupported; 1 = Full URL Sent (Pre-ENGAGE 6.2 firmware will continue support of "URL Path only"); 2 = URL Path only (Server name or address removed)	mt20w
getAuditNoErase	0 = unsupported; 1 = Retrieve Audits without Erase supported	nde, le, sc, rmru, cte, ndeb, leb, scb, rc, XE360
gweSiteSurvey	0 = unsupported; 1 = Site Survey JSON Definition Version 1	gwe, gwe2
newAPCfg	0 = unsupported; 1 = New Access Point Configuration Supported	nde, le, rmru, cte, ndeb, leb
usbMode	0 = unsupported; 1 = supports USB communication mode	mt20w
crdRdr	0 = unsupported; 1 = supports Card Reader configs	cte, rc
bleRdrCfg	0 = unsupported; 1 = supports BLE Credential reader configs	cte, rc
bleCred	0 = unsupported; 1 = supports BLE Credential Configuration	ndeb, leb, mtb, mtkb15, scb, rc, XE360
ipbEnabled	0 = unsupported; 1 = supports IPB Configuration Supported	ndeb NOTE: LE, XE360 use model type to detect IPB

Extension key	Extension value	Device Associated
keyPad	0 = unsupported; 1 = supports Keypad configuration	mtkb, rc
cnstrBlock	0 = unsupported; 1 = supports prevent construction mode	nde, le, cte, ndeb, leb, rc, XE360
iClassHID	0 = unsupported; 1 = supports HID iClass, iClass SE, SEOS Secure Reads	ndeb, leb (HID variants only)
dynMtuDataBufSz	“xxxx” where xxxx indicates value in multiple of 256 (2048, 4096, etc), default 2048. Presence of this device profile indicates that device is capable of Dynamic MTU with different data buffer size.	xe360
mainAuxCtrl	0 = unsupported; 1 = support Aux relay delay in Auto Operator mode; support Rex shunt DPS feature when electrified strike selected	cte
delAuditCmd	0 = unsupported; 1 = supports deleting audits	nde, ndeb, sc, scb, le, leb, cte, XE360

External Device Parameters

Table 8 shows a complete list of all the JSON data tags and values that can be read by an external device. Some of the retrievable parameters are device specific and may not apply to all devices.

ENGAGE devices report all retrievable parameters in one large block, and include some parameters that may not directly apply to the device or its mode of operation. For example; RSI protocol capable devices might report RSI parameters when the device is not configured for RSI protocol operation.

See the **Device Settings** section for an example of this data structure.

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Battery Voltages	battV	String/5 (parent = config)	N/A	MTB, MTKB, RC
Main	main	String/5	“x.x” to “xx.xx”	CTE, MTB, MTKB, RC
Lithium	li	String/5	“x.x” to “xx.xx” (This is always reported as a value of 0 for NDE, NDEB, LE, LEB, and RMRU) when reported from the GWE.	CTE, SC, SCB, MTB, MTKB, RC
Line Voltages	lineV	String/5 (parent = config)	N/A	
Main	main	String/5	“x.x” to “xx.xx”	NDE, LE, SC, SCB, RMRU, NDEB, LEB, XE360
Power input	power	String/4	“poe”, “poe+”, “line”	NDE, LE, SC, SCB, RMRU, NDEB, LEB, MTB, MTKB, RC does not support “poe+”, XE360
Firmware Versions	fwVer	String/5 (parent = config)	N/A NOTE: XE360 will report FW version in xx.xx.xxxxxx Format	
Lock	lock	String/8	“xx.xx.xx”	RC
Main	main	String/8	“xx.xx.xx”	
Credential Reader	credRdr	String/8	“xx.xx.xx”	SC, SCB, RMRU
Bluetooth Module	ble	String/8	“xx.xx.xx”	
WiFi Module	wifi	String/8	“xx.xx.xx”	SC, SCB, MTB, MTKB, RC, XE360

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Main Bootloader	mainBl	String/8	“xx.xx.xx”	
Credential Reader Bootloader	credRdrBl	String/8	“xx.xx.xx”	SC, SCB, RMRU, CTE
WiFi MAC Address	macAdr	String/17	“xx.xx.xx.xx.xx.xx”	SC, SCB, MTB, MTKB, RC, XE360
Safe Mode Recovery FW version	recoveryFwVer	String/8	“xx.xx.xx”	SC, SCB, MTB, MTKB, RC, XE360
Ethernet module	ethernet	String/8	“xx.xx.xx”	SC, SCB, RMRU, LE, NDE, NDEB, LEB, MTB, MTKB, RC, XE360
Ethernet module bootloader	ethernetBl	String/8	“xx.xx.xx”	NDE, SC, SCB, RMRU, LE, NDEB, LEB, MTB, MTKB, RC, XE360
Ethernet MAC Address	ethernetMacAdr	String/17	“xx.xx.xx.xx.xx.xx”	NDE, SC, SCB, RMRU, LE, NDEB, LEB, MTB, MTKB, XE360
Reader Main	rdrMain	String/8	“xx.xx.xx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB
Reader Main Bootloader	rdrMainBl	String/8	“xx.xx.xx”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB
WiFi Settings	wifiStngs	String/8 (parent = config)	N/A	SC, SCB, MTB, MTKB, RC, XE360
SSID	ssid	String/32	Max 32 characters	SC, SCB, MTB, MTKB, RC, XE360
AP Password / Security Key	psswd	String/64	Max 63 characters	SC, SCB, MTB, MTKB, RC, XE360
User Name	usrNm	String/16	Max 16 characters	SC, SCB, MTB, MTKB, RC, XE360
WiFi Security	wifiSec	String/7	“prsnl”, “entrprs”, “open”, “wep”	SC, SCB, MTB, MTKB, RC, XE360
Key Index	kyIndx	Unsigned Int/1	1,2,3,4	SC, SCB, MTB, MTKB, RC, XE360
Discovery Type	dscvryTyp	String/6	“ipAddr”, “dmnNm”, “svrcNm”	SC, SCB, MTB, MTKB, RC, XE360
Secure Connection	scrCnn	String/5	“http”, “https”	SC, SCB, MTB, MTKB, RC, XE360
IP	ip	String/15	“xxx.xxx.xxx.xxx”	SC, SCB, MTB, MTKB, RC, XE360
IP DNS Service	ipDNS	String/64	Max 64 characters	SC, SCB, MTB, MTKB, RC, XE360
Alternate IP DNS Service	altDNS	String/64	Max 64 characters	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
New Access Point Config	apCfgNew	String/6 (parent = config)	N/A	SC, SCB, MTB, MTKB, RC, XE360

Table 8: Retrievable Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
New AP Config Enable	apCfgNewEn	String/1	"T" = enables new Access Point Config. "F" = disables new Access Point Config (default option).	SC, SCB, MTB, MTKB, RC, XE360
New SSID	ssidNew	String/32	Max 32 characters	SC, SCB, MTB, MTKB, RC, XE360
New AP Password	psswdNew	String/64	Max 63 characters	SC, SCB, MTB, MTKB, RC, XE360
New User Name	usrNmNew	String/16	Max 16 characters	SC, SCB, MTB, MTKB, RC, XE360
New WiFi Security	wifiSecNew	String/7	"prsnl", "entrprs", "open", "wep"	SC, SCB, MTB, MTKB, RC, XE360
Config Start Time	apCfgNewStrtTm	String/14	"YYYYMMDDHHMMSS"	SC, SCB, MTB, MTKB, RC, XE360
Ethernet Configuration	ethConfig	String/9 (parent = config)	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Host Connection Type	hostProtocol	String/9	"rs485Rsi", "ipEngage", "ipClient"; NOTE: RC only supports ipClient	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Interface ID	interfaceId	String/4	"eth0", "wlan0"; NOTE: RC only supports eth0	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Interface Enable	interfaceEnable	String/1	"T" = enables interface operation, "F" = disables interface operation	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Discovery Method	discoveryMethod	String/6	"dhcp", "staticIP", "zeroConf"; NOTE: RC supports dhcp and staticIP	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Fixed IP Address, (only applies if discovery method is set to staticIP)	fixedIpAddr	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Default Gateway IP Address (only applies if discovery method is set to staticIP)	defGatewayAddr	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Netmask (only applies if discovery method is set to staticIP)	netmask	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
IP DNS Service	ipDnsAddr	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Alternate IP DNS Service	altDnsAddr	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
IP Server URL (only applies if hostProtocol is ipClient)	ipServerURL	String/128	Max 128 characters	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
CA Server URL (only applies if hostProtocol is ipClient)	caServerURL	String/128	Max 128 characters	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Keep Alive Time (only applies if hostProtocol is ipClient)	wsKeepAlive	String/4	"1" to "3600" seconds	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
IP Interfaces	ipv4Interfaces	String/14 (parent = config)	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Ethernet Interfaces	ethernetProperties	String/18 (parent = ipv4Interfaces)	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Assigned IP Address	assignedIpAddr	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Default Gateway IP Address	defGatewayIpAddr	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Netmask	netmask	String/15	"xxx.xxx.xxx.xxx"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Lock Parameters	lockPrmtrs	String/8 (parent = config)	N/A	
Name	name	String/28	"xxxx.....xxxx"	
Model	mdl	String/6	"nde", "be467", "fe410", "lems", "lmb", "lemd", "rm", "ru", "cte", "ndeb", "lebms", "lemb", "lembd", "be467b", "fe410b", "be467bm", "fe410bm", "mtb11", "mtb15", "mtkb15", "rc11", "rc15", "rck15", "xe360i", "xe360b", "xe360d"	
Lock Operation Mode	lckMode	String/8	"null", "200", "210", "310", "410rsi", "410ip"	MTB, MTKB, 200 mode not supported on RMRU
Lock Serial Number	lckSn	String/16	"xxxx.....xxxx" NOTE: XE360 reports chassis core sn with this	
Main Serial Number	mainSn	String/16	"xxxx.....xxxx"	
Mfg Date	mfgDt	String/8	"YYYYMMDD" YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) Example: "20140527" means May 27, 2014	
Days In Use	daysInUse	Unsigned Int/5	0 – 65536	SC, SCB

Table 8: Retrievable Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Warranty Activation date	wtyAct	String/14	“YYYYMMDDHHMMSS” YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) HH = Hour (00 – 23) MM = Minutes (00 – 59) SS = Seconds (00 – 59) Example: “20140710145030” means July 10, 2014, 2:50:30 pm	NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Hardware Version	hwVer	String/6	“xxxxxx”	
Clock Time	rtcTime	String/14	“YYYYMMDDHHMMSS” YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) HH = Hour (00 – 23) MM = Minutes (00 – 59) SS = Seconds (00 – 59) Example: “20140710145030” means July 10, 2014, 2:50:30 pm	MTB, MTKB
Type	type	String/6	“strm” “office” “apt” “prvcy”	SC, SCB, RMRU, MTB, MTKB
Relock	relock	Unsigned Int/3	Delay time in seconds (0-254)	RMRU, MTB, MTKB
Door Prop Delay	doorProp	Unsigned Int/3	Delay time in seconds (0-254)	SC, SCB, MTB, MTKB
Door Prop Delay Enable	doorPropEn	String/1	“T” = enabled, “F” = disabled (default)	NDE, LE, SC, SCB, RMRU, NDEB, LEB, MTB, MTKB, XE360
ADA Delay	ada	Unsigned Int/3	Delay time in seconds (0-254)	SC, SCB, RMRU, MTB, MTKB
First Man In	firstManIn	Int/1	0=No First Man In; 1= First Man In	SC, SCB, RMRU, MTB, MTKB
Dog On Next Exit	dneEn	Int/1	0=Do not use Dog on Next Exit; 1=Schedules Dog on Next Exit	NDE, SC, SCB, LE, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Battery Fail State	battFail	String/4	“asls”, “safe”, “sec”	SC, SCB, RMRU can only operate as sec, CTE, MTB, MTKB, RC
Daylight Saving Time	dstEnable	Int/1	0=DST not observed 1=DST observed	MTB, MTKB
DST Start	dstStart	String/4	Hour that clocks are moved ahead by 1 hr. 3=standard month start (1=Jan,...) 0=standard day of week start, 0 = Sun.... 2=standard week # start (1=1st,..5=last) 2=standard hour start US default is 3022	MTB, MTKB
DST End	dstEnd	String/4	Hour that clocks are moved behind by 1 hour. B=daylight month start 0=daylight day of week start 1=daylight week # start 2=daylight hour start US default is B012	MTB, MTKB

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Next Database Download Time	dbDwnldTm	String/14	“YYYYMMDDHHMM” YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) HH = Hour (00 – 23) MM = Minutes (00 – 59) Example: “201405271541” means May 27, 2014, 3:41 pm	SC, SCB, MTB, MTKB
Firmware Address	fwUrl	String/64	Max 64 characters NOTE: this is fwUrl though it conveys the same information as fwurl.	SC, SCB, MTB, MTKB
Firmware download Time	fwDwnLdTm	String/14	YYYYMMDDHHMMSS – “20140527040000” May 27, 2014, 4:00:00 am. Value of “0” will result in immediate download.	SC, SCB, MTB, MTKB
Firmware Implement/ Update Time	fwImpTm	String/14	YYYYMMDDHHMMSS – “20140603030000” June 3, 2014, 3:00:00 am. Value of “0” will result in immediate update/implement.	SC, SCB, MTB, MTKB
BLE Credential Enable	bleCredEn	String/1	“T”, “F”	RMRU, NDE, LE, SC
BLE Credential Range	bleCredRng	String/5	“short”, “long”	RMRU, NDE, LE, SC
BLE Performance	blePerf	String/4	“norm”, “max”	RMRU, NDE, LE, SC, MTB, MTKB, CTE, RC
Next Database Version Time Stamp	nxtDbVerTS	String/18	Hexadecimal value, that starts with 0x, followed by 16 hex digits. (i.e. “0x08d16357eef73a69”)	MTB, MTKB
Immediate WiFi Alert Enable	wifiAlertEn	String/1	“T” = enabled (default), “F” = disabled	SC, SCB, MTB, MTKB, RC, XE360
Individual WiFi Immediate Alert Selection	wifiAlertSel	Array of Strings/6	See separate document for Audit & Alert Definitions (Alert Type Only)	SC, SCB, MTB, MTKB, RC, XE360
Blink Interior LED when locked	blinkIntLed	String/1	“T” = enabled, “F” = disabled	NDE, SC, SCB, CTE, MTB, MTKB, RC
Privacy Rapid blink	rapidBlink	String/1	“T” = enabled, “F” = disabled	NDE, SC, SCB, RMRU, CTE, MTB, MTKB, RC
IPB Audit Enable / Disable	ipbAuditEnable	String/1	“T” = enabled, “F” = disabled	NDE, SC, SCB, LE, RMRU, CTE, MTB, MTKB, RC, NDEB, LEB, XE360
Hardware Configuration	hwCfg	String/6	“ipb”, “dbolt”, “strm”	NDE, SC, SCB, RMRU, NDEB, MTB, MTKB, RC
Lock ID	lockId	Unsigned Int/5	Door ID (0-65534) 65535 no tour disabled on LE locks	NDE, RMRU, MTB, MTKB, RC
Number of motor cycle counts	numCycles	Unsigned Int/32	0 - 4294967295	NDE, LE, RMRU, NDEB, LEB, MTB, MTKB, RC, XE360
DPS Audit Enable	dpsEn	String/1	“T” = enabled (default - LE, NDE, LEB, NDEB, RC, XE360), “F” = disabled (default - CTE)	SC, SCB, MTB, MTKB

Table 8: Retrievable Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Database Update Status	dbUpdateStatus	Unsigned Int/3	0=Reset/download not started. 1=JSON data Download completed. 2=DB Update Completed.	SC, RMRU, MTB, MTKB
PIN Length	pinLength	Unsigned Int/2	3-16, RC: 0-9, 0 is variable PIN length (user must enter start (*) and stop(#)), acceptable PIN lengths are 1-9	NDE, SC, SCB, LE, RMRU, NDEB, LEB, MTB, MTKB, XE360
PIN Enable	pinEn	String/1	"T" or "F"	NDE, SC, SCB, LE, RMRU, NDEB, LEB, MTB, MTKB, XE360
Anti-Tailgate (Immediate relock on door close)	immRelockEn	String/1	"T" or "F"	NDE, SC, SCB, LE, RMRU, NDEB, LEB, MTB, MTKB, XE360
410-IP Communication Failsafe Mode	gatewayCommFail	String/4	"asls", "sec", "safe"	NDE, SC, SCB, LE, CTE, RMRU does not support "safe", NDEB, LEB, MTB, MTKB, XE360
Tamper Cover Removal Failsafe Mode	tamperFail	String/4	"asls", "sec", "safe"	NDE, SC, SCB, LE, CTE, RMRU does not support "safe", NDEB, LEB, MTB, MTKB, RC does not support "safe", XE360
Invalid Card Presented Audit	invCrdAudEn	String/1	"F"= Do not create an audit for invalid card presentations (Default). "T"= Create an audit for invalid card presentations. NOTE: if this configuration is enabled the edge device audits should be checked regularly to ensure that the edge device audit buffer is not inadvertently filled with invalid card presentation audits. NOTE: The default for RC-05 is "T". Since the device reports audits in real time it is not necessary to check the device audits regularly.	RMRU, MTB, MTKB
Audit ID Enable	auditIDEn	String/1	"F"= The lock will not send the new unique audit identifier with the audit json (default). "T"= The lock will send the new unique audit identifier with the audit JSON. NOTE: The default for RC-05 is "T".	MTB, MTKB
New Style of Holiday and Auto Event Audits Enable	autoHolAudEn	String/1	"F"= The lock will continue to generate Engage legacy style of holiday and auto event audits (default). "T"= The lock will generate new style of holiday and auto events audits. NOTE: The default for RC-05 is "F".	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 8: Retrievable Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
WiFi Download Retry Time Interval	wifiDwnldRtyTmIntv	Unsigned Int/4	5-1440 Default Value Setting is 1440. If WiFi event fails, this setting is the number of minutes after the failure that the device should re-try the WiFi event. The retry will continue wifiDwnldRtyCnt times every wifiDwnldRtyTmIntv minutes.	SC, SCB, MTB, MTKB, RC, XE360
WiFi Download Retry Counter	wifiDwnldRtyCnt	Unsigned Int/3	1-250 Default Value Setting is 1. If WiFi event fails, this setting is the number of times after the failure that the device should re-try the WiFi event. The retry will continue wifiDwnldRtyCnt times every wifiDwnldRtyTmIntv minutes.	SC, SCB, MTB, MTKB, RC, XE360
Communication Mode	commMode	String/4	“usb” or “wifi”	NDE, LE, SC, SCB, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Initial TimeStamp	initTS	String/14	“YYYYMMDDHHMMSS”– “20140704030000” means July 4, 2014, 3:00:00 am	NDE, LE, SC, SCB, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Prevent Construction Mode	blkCnstr	String/1	“T” or “F”	RMRU, SC, SCB, MTB, MTKB
Advanced Security Module (ASM) Enable	asmEn	String/1	“T”, “F”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Beep on Tamper	tmprBp	String/1	“T”, “F”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Tamper Sensitivity	tmprSens	String/4	“low”, “med”, “high”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Network Timeout	ntwkTO	UnsignedInt/3	Time in seconds (0-30), 0 (default) = decision at door (DaD); NOTE: A non-zero value prescribes Decision at Host (DaH) and the value specifies the amount of time since the last successful communication with the host before the device transitions to DaD.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Two-Factor Timeout	twoFctrTO	UnsignedInt/3	Time in seconds (0-30), 10s (default)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
PACS	pacs	String/7	“Schlage”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
REX Without Notification	REXwoNotif	String/1	“T” = RC does not change LED state for REX, “F” = RC changes LED state for REX	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Extended Credential Schedules Enable	extSchedEn	String/1	“T”, “F”	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Mechanical Override detection Enable	mechOverrideEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB
Credential Brute Force Attack Prevention Enable	credBFAPrevEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB
Reader Parameters	rdrPrmtrs	String/7 (parent = config)	N/A	RMRU
BLE Reader Name	bleRdrName	String/25	"xxxx.....xxx"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, MTB, MTKB, RC, XE360
Beeper Enable	bprEn	String/1	"T" or "F"	
Reader Sensitivity	rdrSense	String/4	"norm", "high", "max"	SC, SCB, RMRU, CTE, MTB, MTKB, RC, XE360
Days In Use	daysInUse	Unsigned Int/5	0 - 65536	SC, SCB, RMRU
Serial Number	sn	String 10	"xxxxxxxxxx"	SC, SCB, RMRU
Hardware Version	hwVer	String/6	"xxxxxx"	SC, SCB, RMRU
Manufacture Date	mfgDt	String/8	"YYYYMMDD" YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) Example: "20140527" means May 27, 2014	SC, SCB, RMRU
Config Card Presented Status (Read Only)	configCrdPrsntd	String/1	"T", "F"	RMRU, SCB, SC, CTE, XE360
Prox Config HID	proxConfHID	String/1	"T", "F"	SC, RMRU, SCB, XE360
Prox Config GECASI	proxConfGECASI	String/1	"T", "F"	SC, RMRU, SCB, XE360
Prox Config GE4001	proxConfGE4001	String/1	"T" = 4001, "F" = 4002	SC, RMRU, SCB, XE360
Prox Config GE4002	proxConfGE4002	String/1	"T" = 4002, "F" = 4001	SC, RMRU, SCB, XE360
Prox Config AWID	proxConfAWID	String/1	"T", "F"	SC, RMRU, SCB, XE360
Smart Card 14443 UID	uid14443	String/1	"T", "F"	RMRU, SCB, SC
Smart Card 14443 MiFare	mi14443	String/1	"T", "F"	RMRU, SCB, SC
Smart Card 14443 MiFare Plus	mip14443	String/1	"T", "F"	SC, RMRU, SCB, XE360
Smart Card 14443 Mifare DESFire EV1/2/3	noc14443	String/1	"T", "F" NOTE: This setting enables/disables DESFire EV1, EV2, and EV3 credentials.	SC, RMRU
Smart Card 15693 UID	uid15693	String/1	"T", "F"	RMRU, SCB, SC

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Smart Card iClass UID 40 bit	iClsUID40b	String/1	"T","F" NOTE: "Smart Card iClass UID 64 bit" must be set to "F" to enable this parameter to be "T".	RMRU, SCB, SC
PIV Configuration	pivCnfg	String/11	"disbld","75bPIV","58bTWIC_CAC" "14443PIV200b","64bBCD_TWIC","83bTWIC_CAC","66bTWIC_CAC" "64bTWIC","91bTWIC_CAC" "40bBCD","40bRevBCD","64bBCD","64bRevBCD","128bBCD","128bRevBCD","58bAMAG"	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Smart Card iClass UID 64 bit	iClsFrmt	String/11	"disbld","64bCSN" NOTE: "Smart Card iClass UID 40 bit" must be set to "F" to enable this parameter to be "64bCSN".	SC, RMRU, SCB, XE360
Smart Card Secure iClass, iClass SE, SEOS	iClsSec	String/1	"T" = enabled (default), "F" = disabled NOTE: Setting applicable only for HID variant of NDEB and LEB locks	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Mag Card Reader Track 1 Enable	magtrk1	String/1	"T","F"	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Mag Card Reader Track 2 Enable	magtrk2	String/1	"T","F"	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Mag Card Reader Track 3 Enable	magtrk3	String/1	"T","F"	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Mag Card Reader Low Power Enable	magLwPwrEn	String/1	"T","F"	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Standby LED Color	ledStbyColor	String/5	"off", "red", "green", "amber"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, XE360
Standby LED State	ledStbyState	String/5	"solid", "flash"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, XE360
Credential Read LED	ledCredRd	String/5	"none", "green"	NDE, NDEB, LE, LEB, RMRU, SC, SCB, CTE, XE360
Backlight Timeout	bckLghtTmOut	Unsigned Int/2	0-15, 0 = disabled	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, MTKB, CTE, XE360
Keypad Output Format	keypadOutput	String/10	"4BitNP", "8BitNP", "Galaxy26A", "PCSC4", "PCSC5"	NDE, SC, LE, RMRU, LEB, NDEB, SCB, MTB, CTE, RC, XE360
Keypad facility code	kpFclCode	Unsigned Int/3	0-255	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, RC, XE360
Number of key presses to buffer/cache	kpBuffCache	Unsigned Int/2	0-15	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360
Keystroke Timeout	keystrokeTO	Unsigned Int/2	0-15, 0 = disabled	NDE, SC, SCB, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, XE360

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Custom Key Installation Status	customKeyPrsntd	String/1	"T","F"	RMRU, SC, SCB, CTE, XE360
Clock and Data Format	rdrOutFmt	String/8	"dsbld", "format1", "format2", "format3", "format4", "format5", "format6", "format7", "format8", "format9", "format10", "format11", "format12"	NDE, NDEB, SC, SCB, LE, LEB, RMRU, MTB, MTKB, CTE, RC, XE360
Reader Output Format	rdrOutput	String/8	"wiegand", "clkndata"	NDE, NDEB, SC, SCB, LE, LEB, RMRU, RC, XE360
Auto-unlock with Badge Enable	autoWBadgeEn	Unsigned Int/2	0 = disabled, 1 = enabled (0 means autowbadgeSchedIDs has a value of an empty array []. 1 means autowbadgeSchedIDs has a value of an array with at least one valid element.)	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
General Purpose Inputs	gpi	Array (parent = config)	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Input Type	iType	String/5	"rex", "auxIn", "dps"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Input Enable	iEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Signal Polarity	pol	String/4	"high", "low"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
General Purpose Outputs	gpo	Array (parent = config)	N/A	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Output Type	oType	String/7	"mainRly", "ttl1", "ttl2": NOTE: The main relay is included here to specify whether the main relay will be activated on a REX or Aux Input.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Output Enable	oEn	String/1	"T", "F"	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Signal Polarity	pol	String/4	"high", "low" NOTE: Main relay is included to allow for control of fail safe and fail secure devices.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Mode	mode	String/4	"puls" (default), "hold" NOTE: When the mainRly source is auxIn, mode can be pulse, where the duration is the relock, or hold. When the mainRly source is rex the mainRly unlocks for the duration of the relock.	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360

Table 8: Retrievable Parameters				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Source	src	Array of Strings/9	["mirrorRly", "rex", "auxIn", "auxCred", "tamper", "frcdDoor", "prppdDoor"] NOTE: The mirrorRly does not apply to the mainRly	CTE, LE, LEB, MTB, MTKB, NDE, NDEB, RMRU, SC, SCB, XE360
Device Parameters	rsiPrmtrs	String/8 (parent = config)	N/A	
Deny Access	dnyAccs	String/1	"T","F"	RMRU, CTE, MTB, MTKB, RC, XE360
Relock Method	relockMethod	String/9	"tmr", "drOpn", "drCls"	SC, SCB, RMRU, CTE, MTB, MTKB, RC, XE360
Heartbeat Time	rsiHrtbtTm	Unsigned Int/5	0 – 65535	CTE, MTB, MTKB, RC, XE360
Retry times	rsiRtryTm	Unsigned Int/2	1-15	CTE, MTB, MTKB, RC, RMRU, XE360
Subsequent Delay	rsiSubqtDly	Unsigned Int/2	1-15	CTE, MTB, MTKB, RC, RMRU, XE360
First Delay	rsiFrstDly	Unsigned Int/2	1-15	CTE, MTB, MTKB, RC, RMRU, XE360
Cache memory bits per card	cacheMemBts	Unsigned Int/3	0-255	RMRU, CTE, MTB, MTKB, RC, XE360
Clear cache entries	clrCache	String/1	"T","F"	RMRU, CTE, MTB, MTKB, RC, XE360
Auto purge	autoPrg	String/1	"T","F"	RMRU, CTE, MTB, MTKB, RC, XE360
Communication Failsafe mode	rsiCommFailSfMd	String/6	"disbld","sec","safe"	CTE, MTB, MTKB, RC, XE360
Cache memory mode	cacheMemMd	String/7	"full","fcltyCd"	RMRU, CTE, MTB, MTKB, RC, XE360
Maximum number of cache entries	maxCacheEntries	Unsigned Int/3	0-255	RMRU, CTE, MTB, MTKB, RC, XE360
PIN-Required Mode	pinReqMode	String/12	"disbld", "GrnRedNoBeep", "GrnRedBeep2"	RMRU, CTE, MTB, MTKB, RC, XE360
Request to Enter Inquiry Enable	renInquiry	String/1	"T","F"	RMRU, CTE, MTB, MTKB, RC, XE360
IPB LED Blink Enable	ipbLedBlink	String/1	"T","F"	NDE, SC, SCB, CTE, MTB, MTKB, RC, XE360
IPB Online Configuration Enable Lock/Unlock	ipbOnlineCfg	String/8	"disbld","lckUnlk"	NDE, SC, SCB, RMRU, CTE, MTB, MTKB, RC, XE360
IPB Offline Configuration	ipbOfflineCfg	String/8	"disbld","lckUnlk","dsbRdr"	NDE, SC, SCB, RMRU, CTE, MTB, MTKB, RC, XE360
IPB Long Press	ipbLongPress	Unsigned Int/2	0-15	NDE, SC, SCB, RMRU, CTE, MTB, MTKB, RC, XE360

Table 8: Retrievable Parameters

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
RSI APM Type	rsiApmType	Unsigned Int/2	00 to 13 reserved (see RSI doc); 14 = NDE; 15 = ADE; 16 = BE467/FE410/BE467B/FE410B/BE467BM/FE410BM; 17 = LE/LEB; 18 = RMRU; 19 = NDEB	CTE, MTB, MTKB, RC, XE360
RSI Reader Type	rsiRdrType	Unsigned Int/3	00 to 66 reserved (see RSI doc); 67 = NDE_MT2_NO_KPD; 68 = CONTROL_SM2_NO_KPD; 69 = LE_MT2_NO_KPD; 255 = No Model Information	RMRU, CTE, MTB, MTKB, RC, XE360
Chassis Core Parameters	chassisPrmtrs	String/6 (parent = config)	N/A	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC
Mfg Date	mfgDt	String/8	“YYYYMMDD” YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) Example: “20140527” means May 27, 2014	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC
Days In Use	daysInUse	Unsigned Int/5	0 – 65536	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC
Hardware Version	hwVer	String/6	“xxxxxx”	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC
Mechanical Variant	mechVariant	String/8	“wideExit”, “mortise”, “tubular”	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC
Activate Aux Relay	“auxRly”	String/7	“nothing”, “autoOp”, “mntrSys”, “other”	
Auto Operate Aux Delay Time (MS)	“autoOpDlyTm”	Number	500-1500 (default 1500)	
Main Strike Select	“strkSel”	String/10	“elctrcStrk” (default), “elctrcExit”, “elctrfdLck”, “magLck”, “other”	
DPS mask time (Seconds)	“dpsShntTm”	Number	0 - 254 (2 - default, 255 - reserved)	

➔ **NOTE:** For ENGAGE 6.1 and later devices, devices configured for IP or stand-alone communication may not transmit their RSI parameters (rsiPrmtrs) or any children tags under the RSI parameters as they are not relevant to IP or stand-alone operation. The device accepts any changes to the RSI parameters if they are provided to the device, however the device does not provide the updated information back to the host.

Audit Trail Reporting

To minimize the amount of data transfer time, the device only sends audits that have never been successfully transferred to the host. This is applicable when receiving audits via BLE or WiFi directly from edge device. Appended to the end of an audit trail transmission is the device's current value for the "nxtDbVerTS". Appending this time stamp to the end of the audit trail block also ensures that the device and host are synchronized. If the JSON configuration tag "lockId" is changed, this tag will also be appended to the end of the audit trail transmission.

Table 9: Audit Trail Record				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Audits	audits	String/6	N/A	
Audit Trail Event	event	String/8	<i>See separate document for Audit & Alert Definitions.</i>	MTB, MTKB
Time Stamp	time	String/14	"YYYYMMDDHHMMSS" YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) HH = Hour (00 – 23) MM = Minutes (00 – 59) SS = Seconds (00 – 59) Example: "20140809134522" means August 9, 2014, 1:45:22 pm	MTB, MTKB
Audit ID	auditId	String/3	0 – 255 (If "auditIDEn" tag = "T")	MTB, MTKB
Next Database Version Time Stamp	nxtDbVerTS	String/18	Hexadecimal value, which starts with 0x, followed by 16 hex digits. (i.e. "0x08d16357eef73a69")	MTB, MTKB
Lock ID	lockId	Unsigned Int/5	Door ID (0-65534) 65535 disables no tour on LE locks	RMRU ,MTB, MTKB

Asynchronous Events/Alerts Reporting

Table 10: Asynchronous Event Record				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Asynchronous Event	asyncEvent	String/10	N/A	
Asynchronous Event	event	String/8	<i>See separate document for Audit & Alert Definitions.</i>	MTB, MTKB
Time Stamp	time	String/14	"YYYYMMDDHHMMSS" YYYY = Year MM = Month (01 – 12) DD = Day (01 – 31) HH = Hour (00 – 23) MM = Minutes (00 – 59) SS = Seconds (00 – 59) Example: "20140809134522" means August 9, 2014, 1:45:22 pm	MTB, MTKB

An alert is an elevated audit used in WiFi mode and is sent to the host server when the condition is met by the device. For a complete list of audits/events, reference the ENGAGE Audit Spreadsheet.

WiFi Device Commissioning

If the device is to be configured with WiFi communication directly to the host, the device requires several different parameters that are associated with the ability to login to a host system. See **Table 11**.

This data is sent from the mobile application during the commissioning process.

Table 11: WiFi Commissioning Record				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
WiFiCommision	wifiCmsh	String/8	N/A	SC, SCB, MTB, MTKB, RC, XE360
Save Configurations	saveCnfgs	String/1	"T" = enables Wifi operation "F" = disables Wifi operation	SC, SCB, MTB, MTKB, RC, XE360
Access Point Config	apCnfg	String/6	N/A	SC, SCB, MTB, MTKB, RC, XE360
SSID	ssid	String/32	Max 32 characters	SC, SCB, MTB, MTKB, RC, XE360
AP Password	psswd	String/64	Max 63 characters	SC, SCB, MTB, MTKB, RC, XE360
User Name	usrNm	String/16	Max 16 characters	SC, SCB, MTB, MTKB, RC, XE360
WiFi Security	wifiSec	String/7	"prsnl", "entrprs", "open", "wep"	SC, SCB, MTB, MTKB, RC, XE360
Key Index	kyIndx	Unsigned Int/1	1,2,3,4	SC, SCB, MTB, MTKB, RC, XE360
Host Config	hstCnfg	String/7	N/A	SC, SCB, MTB, MTKB, RC, XE360
Discovery Type	dscvryTyp	String/6	"ipAddr", "dmnNm", "srvNm"	SC, SCB, MTB, MTKB, RC, XE360
Secure Connection	scrCnn	String/5	"http", "https"	SC, SCB, MTB, MTKB, RC, XE360
IP	ip	String/15	"xxx.xxx.xxx.xxx"	SC, SCB, MTB, MTKB, RC, XE360
IP DNS Service	ipDNS	String/64	Max 64 characters	SC, SCB, MTB, MTKB, RC, XE360
Alternate IP DNS Service	altDNS	String/64	Max 64 characters	SC, SCB, NDE, LE, RMRU, CTE, NDEB, LEB, MTB, MTKB, RC, XE360

* Only NDE supports "kyIndx" = 1 till 4 (if configured through host). Usage of a Key Index other than 1 is **NOT** recommended. For LE, RMRU, MT20W, CTE and ENGAGE Mobile Application "kyIndx" = 1.

Keyed-Hashing for Message Authentication (HMAC)

Keyed-Hashing for Message Authentication (HMAC) Authentication, by an IP Host, BLE enabled mobile application, or ENGAGE™ device, requires JSON structures that allows the use of a securing method for JSON General Parameter communications and methods to allow reception device targeting and contextual time relevance. HMAC is recommended as it offers extra message authentication during communication with the devices. For example, many networks routinely change their WiFi passwords as a security requirement. Sites in 210 Mode (WiFi) can use HMAC to send an updated WiFi network password and the device can authenticate that it came from the host server.

General Parameters JSON Object

To isolate the general parameters of a device, the general parameters JSON object is required and this object and all of its content are required when JSON Message Authentication is enabled for the site. See **Table 12**.

Table 12: General Parameters JSON Object				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
HMAC Message Serial Number	mainSN	String/16	The lower 8 bytes, in hexadecimal format, of the device serial number which the message is intended for	RC
HMAC Message Time Stamp	msgTS	String/17	The first two byte must be 0x, followed by 15 Hexi-decimal digits	RC
HMAC Enable	hmacEnable	String/6	"T" = HMAC validation is required on the received message "F" = HMAC validation is not required on the received message	RC

Hash Message Authentication Code (HMAC) Data JSON Object

To isolate the HMAC data parameters of a device the HMAC data JSON object was created. Each tag within the HMAC Data JSON Object is required in messages that have the HMAC-based JSON Message Authentication enabled for a site. See **Table 13**.

Table 13: Hash Message Authentication Code (HMAC) Data JSON Object				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
HMAC Type	hmacType	String/16	Instruct the device which type of HMAC hashing algorithm to use	RC
			0= not valid	
			1= SHA1-type hashing	
			2-7= reserved	
HMAC Digest	hmac	String/64	40 hexadecimal digit result of the HMAC hashing algorithm. The input to the HMAC hashing algorithm is the entire message excluding the HMAC value itself.	RC

Examples of User Records (JSON)

deleteAll Tag Set

The following JSON data structure excerpt represents how the user record data structure might look if a user list was being sent for the very first time or when the current user list needs to be erased and populated with a new user list. Note the value of “deleteAll” is set to 1, which signifies that the database should replace the existing database.

```
{
  "db": {
    "usrRcrd": {
      "deleteAll": 1,
      "delete": [],
      "update": [],
      "add": [
        {
          "usrID": 64,
          "adaEn": 0,
          "actDtTm": "20140101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1, 3],
          "primeCr": "01020304050607080910111213141516",
          "prCrTyp": "card"
        },
        {
          "usrID": 146,
          "adaEn": 1,
          "actDtTm": "20100101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1, 3],
          "primeCr": "16151413121110090807060504030201",
          "prCrTyp": "card"
        }
      ]
    }
  }
}
```

deleteAll Tag Cleared

The following JSON data structure excerpt represents how the user record data structure might look if an updated user list was being sent to the device. This list might contain a set of users to delete, update and add.

➔ **NOTE:** The value of “deleteAll” is set to 0, which signifies that the existing database in the device should be updated.

```
{
  "db": {
    "usrRcrd": {
      "deleteAll": 0,
      "delete": [
        {
          "primeCr": "02030405060708091011121314151617",
          "prCrTyp": "card"
        },
        {
          "primeCr": "17161514131211100908070605040302",
          "prCrTyp": "card"
        },
        {
          "primeCr": "03040506070809101112131415161718",
          "prCrTyp": "card"
        }
      ],
      "update": [
        {

```

```

    "usrID": 149,
    "adaEn": 1,
    "actDtTm": "20100101000000",
    "expDtTm": "20200101000000",
    "fnctn": "toggle",
    "crSch": [1],
    "primeCr": "16151413121110090807060504030201",
    "prCrTyp": "card"
  }
],
"add": [
  {
    "usrID": 148,
    "adaEn": 0,
    "actDtTm": "20100101000000",
    "expDtTm": "20200101000000",
    "fnctn": "norm",
    "crSch": [1],
    "primeCr": "01020304050607080910111213141516",
    "prCrTyp": "card"
  }
]
},

```

deleteAll Tag Cleared, No User Records to Update

The following JSON data structure excerpt represents how the user record data structure might look if an updated database is sent that contains no updates to the user records. Note the delete, update and add records are empty ([]), which signifies that there are no records to delete, add, or update.

```

{
  "db": {
    "usrRcrd": {
      "deleteAll": 0,
      "delete": [],
      "update": [],
      "add": []
    }
  },

```

partialDb Tag Set, More Than 5000 User Records to Update

The following JSON data structure excerpt represents how the user record data structure might look for when a user list of greater than 5000 records was being sent. Note the value of "partialDb" is incremented, which signifies that the database should append the user records to the previously sent database. Note the partialDb tag only applies to reader-controllers. Reader-controllers are limited to 1000 user record transfers at a time.

First set of user records, see deleteAll Tag Set example.

Second set of user records:

```

{
  "db": {
    "usrRcrd": {
      "deleteAll": 0,
      "partialDb": 2,
      "delete": [],
      "update": [],
      "add": [
        {
          "usrID": 5001,
          "adaEn": 0,
          "actDtTm": "20140101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1, 3],

```


ENGAGE - JSON Data Structures | Examples of User Records (JSON)

```
    "primeCr": "01020304050607080910111213141516",
    "prCrTyp": "card"
  },
  {
    ...
  },
  {
    "usrID": 10000,
    "adaEn": 1,
    "actDtTm": "20100101000000",
    "expDtTm": "20200101000000",
    "fnctn": "norm",
    "crSch": [1, 3],
    "primeCr": "16151413121110090807060504030201",
    "prCrTyp": "card"
  }
]
```

Last set of user records:

```
{
  "db": {
    "usrRcrd": {
      "deleteAll": 0,
      "partialDb": 65535,
      "delete": [],
      "update": [],
      "add": [
        {
          "usrID": 10001,
          "adaEn": 0,
          "actDtTm": "20140101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1, 3],
          "primeCr": "01020304050607080910111213141516",
          "prCrTyp": "card"
        },
        {
          ...
        },
        {
          "usrID": 12154,
          "adaEn": 1,
          "actDtTm": "20100101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1, 3],
          "primeCr": "16151413121110090807060504030201",
          "prCrTyp": "card"
        }
      ]
    }
  }
}
```

Example of Holiday Records (JSON)

The following JSON data structure represents how the data structure might look if the holiday records were being sent to the device.

```
{
  "db": {
```

```
{
  "holidays": [
    {
      "strtDtTm": "20141223000000",
      "endDtTm": "20141227000000",
      "action": "pass"
    },
    {
      "strtDtTm": "20140703000000",
      "endDtTm": "20140705000000",
      "action": "rstrctSec"
    }
  ],
}
```

Example of Credential Schedule Records (JSON)

The following JSON data structure represents how the data structure might look if the credential schedule records were being sent to the device.

```
{
  "db": {
    "schedules": [
      {
        "days": ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"],
        "strtHr": 0,
        "strtMn": 0,
        "lngth": 1439
      },
      {
        "days": ["Mo", "Tu", "We", "Th", "Fr"],
        "strtHr": 8,
        "strtMn": 0,
        "lngth": 540
      },
      {
        "days": ["Mo", "We", "Fr"],
        "strtHr": 8,
        "strtMn": 0,
        "lngth": 240
      }
    ],
  }
}
```

Example of Credential Schedule Records with Two-Factor Authentication (JSON)

The following JSON data structure represents how the data structure might look if the credential schedule records were being sent to the device with various types of two-factor authentication. 1f is single factor. 2f is two factor (ie single user with a card and pin). 2fx2 is two factor for two unique users (ie user 1 with card and pin and user 2 with card and pin). Note the auth tag only applies to reader-controllers.

```
{
  "db": {
    "schedules": [
      {
        "days": ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"],
        "strtHr": 0,
        "strtMn": 0,
        "lngth": 1439,
        "auth": "1f"
      },
      {
        "days": ["Mo", "Tu", "We", "Th", "Fr"],

```

```

    "strtHr": 8,
    "strtMn": 0,
    "lngth": 540,
    "auth": "2f"
  },
  {
    "days": ["Mo", "We", "Fr"],
    "strtHr": 8,
    "strtMn": 0,
    "lngth": 240,
    "auth": "2fx2"
  }
],
}
}

```

Example of Extended Credential Schedule Records (JSON)

Note the Extended Credential Schedules only apply to reader-controllers.

```

{
  "db": {
    "usrRcrd": {
      "deleteAll": 0,
      "delete": [],
      "update": [],
      "add": []
    },
    "schedulesExt ": [
      {
        "extScheduleID": "1",
        "intvlID": "1",
        "moStrt": "0800",
        "moEnd": "1700",
        "moAuth": "1f"
      },
      {
        "extScheduleID": "1",
        "intvlID": "2",
        "tuStrt": "0000",
        "tuEnd": "2400",
        "tuAuth": "2f"
      },
      {
        "extScheduleID": "23",
        "intvlID": "1",
        "tuStrt": "0000",
        "tuEnd": "2400",
        "tuAuth": "2f"
      }
    ]
  }
}

```

Example of Auto Unlock Records (JSON)

The following JSON data structure represents how the data structure might look if the auto event records were being sent to the device.

```
{
  "db": {
    "autoUnlock": [
      {
        "action": "pass",
        "strtHr": 8,
        "strtMn": 0,
        "days": ["Mo", "Tu", "We", "Th", "Fr" ]
      },
      {
        "action": "sec",
        "strtHr": 17,
        "strtMn": 0,
        "days": ["Mo", "Tu", "We", "Th", "Fr" ]
      }
    ]
  }
}
```

Example of Configuration Records (JSON)

The following JSON data structure represents how the data structure might look if the configuration records were being sent to the device.

```
{
  "config":{
    "battFail":"sec",
    "proxConfGE4001":"T",
    "rtcTime":"20140424152753",
    "iClsUID40b":"T",
    "bprEn":"T",
    "proxConfHID":"T",
    "uid15693":"T",
    "mil4443":"T",
    "uid14443":"F",
    "proxConfGECASI":"T",
    "dstStart":"12b0",
    "ada":"30",
    "proxConfAWID":"T",
    "proxConfGE4002":"F",
    "doorProp":20,
    "relock":3,
    "noc14443":"T",
    "dstEnd":"2230",
    "firstManIn":0,
    "dstEnable":0,
    "fwUrl":""
  },
  "apCfgNew":
  {
    "apCfgNewEn":"T",
    "ssidNew":"02.04.12",
    "psswdNew":" G10N\@113",
    "usrNmNew":"Cr649",
    "wifiSecNew":"entrprs",
    "apCfgNewStrtTm":"20140527141029"
  }
}
```

Example of Configuration Records with GPIO Configuration (JSON)

The following JSON data structure represents how the data structure might look if the configuration records including GPIO configurations were being sent to the device.

```
{
  "config": {
    "proxConfGE4001": "T",
    "rtcTime": "20140424152753",
    "iClsUID40b": "T",
    "bprEn": "T",
    "proxConfHID": "T",
    "uid15693": "T",
    "name": "sw555",
    "mil4443": "T",
    "uid14443": "F",
    "proxConfGECASI": "T",
    "dstStart": "12b0",
    "ada": "30",
    "proxConfAWID": "T",
    "proxConfGE4002": "F",
    "doorProp": 20,
    "relock": 3,
    "noc14443": "T",
    "dstEnd": "2230",
    "firstManIn": "false",
    "dstEnable": "false",
    "fwUrl": "",
    "tmprBp": "T",
    "tmprSens": "low",
    "gpi": [
      {
        "iType": "rex",
        "iEn": "T",
        "pol": "high"
      },
      {
        "iType": "dps",
        "iEn": "T",
        "pol": "high"
      }
    ],
    "gpo": [
      {
        "oType": "ttl1",
        "oEn": "T",
        "pol": "low",
        "mode": "puls",
        "src": ["rex"]
      }
    ]
  }
}
```

➔ Note: For gpi configuration, the iType parameter must come first, before all the other parameters. Similarly, for gpo configuration, the oType parameter must come first.

Partial User Database Update

Putting all the records together, the following might represent the JSON data structure for a sample database that could be sent to update an existing device database.

```
{
  "db": {
    "usrRcrd": {
      "deleteAll": 0,
      "delete": [],
      "update": [],
      "add": [
        {
          "usrID": 64,
          "adaEn": 0,
          "actDtTm": "20140101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1,3],
          "primeCr": "01020304050607080910111213141516",
          "prCrTyp": "card"
        },
        {
          "usrID": 146,
          "adaEn": 1,
          "actDtTm": "20100101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1,3],
          "primeCr": "16151413121110090807060504030201",
          "prCrTyp": "card"
        },
        {
          "usrID": 149,
          "adaEn": 1,
          "actDtTm": "20100101000000",
          "expDtTm": "20200101000000",
          "fnctn": "toggle",
          "crSch": [1],
          "primeCr": "02030405060708091011121314151617",
          "prCrTyp": "card"
        },
        {
          "usrID": 148,
          "adaEn": 0,
          "actDtTm": "20100101000000",
          "expDtTm": "20200101000000",
          "fnctn": "norm",
          "crSch": [1],
          "primeCr": "14161514131211100908070605040302",
          "prCrTyp": "card"
        }
      ]
    },
    "schedules": [
      {
        "days": ["Su", "Mo", "Tu", "We", "Th", "Fr", "Sa"],
        "strtHr": 0,
        "strtMn": 0,
        "lngth": 1439
      },
      {
        "days": ["Mo", "Tu", "We", "Th", "Fr"],
        "strtHr": 8,

```

```

    "strtMn": 0,
    "lngth": 540
  },
  {
    "days": ["Mo", "We", "Fr"],
    "strtHr": 8,
    "strtMn": 0,
    "lngth": 240
  }
],
"holidays": [
  {
    "strtDtTm": "20141223000000",
    "endDtTm": "20141227000000",
    "action": "pass"
  },
  {
    "strtDtTm": "20140703000000",
    "endDtTm": "20140705000000",
    "action": "rstrctSec"
  }
],
"autoUnlock": [
  {
    "action": "pass",
    "strtHr": 8,
    "strtMn": 0,
    "days": ["Mo", "Tu", "We", "Th", "Fr"]
  },
  {
    "action": "sec",
    "strtHr": 8,
    "strtMn": 0,
    "days": ["Mo", "Tu", "We", "Th", "Fr"]
  }
],
"config": {
  "relock": 3,
  "doorProp": 20,
  "ada": 30,
  "firstManIn": 0,
  "dstEnable": 0,
  "dstStart": "3022",
  "dstEnd": "B012",
  "battFail": "asIs",
  "bprEn": "T",
  "rtcTime": "20140701170122"
},
"dbDwnIdTm": "20140702210122",
"nxtdbVerTS": "0x08d16386bfaec212"
}

```

Example of database with minimum required tags necessary to send a database to an ENGAGE device

```
{
  "db": {
    "usrRcrd": {
      "deleteAll": 1,
      "delete": [],
      "update": [],
      "add": [
        {
          "usrID": 20020,
          "adaEn": 0,
          "fnctn": "norm",
          "crSch": 1,
          "actDtTm": "20000101000000",
          "expDtTm": "21350101000000",
          "primeCr": "9242dc6bb724865b563fa9756eca421f",
          "prCrTyp": "card",
          "scndCrTyp": "null"
        }
      ]
    },
    "schedules": [
      {
        "days": [
          "Su",
          "Mo",
          "Tu",
          "We",
          "Th",
          "Fr",
          "Sa"
        ],
        "strtHr": 0,
        "strtMn": 0,
        "lngth": 1440
      }
    ],
    "holidays": [],
    "autoUnlock": []
  },
  "dbDwnLdTm": "",
  "nxtDbVerTS": "0x123456789012345"
}
```

Delete Holiday Records

There may be instances when holiday records need to be deleted from the device. The host must be able to provide a JSON data structure that permits this operation.

The following JSON structure represents how holiday records would be erased from the device during a database update.

```
{
  "db": {
    "holidays": []
  }
}
```


Delete Auto Unlock Event Records

There may be instances when auto unlock records need to be deleted from the device. The host must be able to provide a JSON data structure that permits this operation.

The following JSON structure represents how auto unlock records would be erased from the device during a database update.

```
{
  "db":{
    "autoUnlock": []
  }
}
```

WiFi Commissioning

The following structure represents a JSON data structure that contains WiFi commissioning data.

```
{
  "wifiCmsh":
  {
    "hstCnfg":
    {
      "ipDNS":"orcainegration.cloudapp.net",
      "altDNS":"orcabackup.cloudapp.net",
      "scrCnn":"https",
      "dscvryTyp":"ipAddr"
    },
    "apCnfg":
    {
      "ssid":"test",
      "psswd":"password123",
      "usrNm":"default",
      "wifiSec":"prsnl"
    }
  }
}
```

➔ **Note:** The **wifiCmSh** object requires both **hstCnfg** and **apCnfg** tag to be present in order to update Wi-Fi configurations.

WiFi Check-in Retry Parameters

Whenever a lock receives a new database with a valid future dbDwnLdTm tag, it sets a timer with the dbDwnLdTm timestamp. Once this time is reached, the lock attempts to connect to the server using the saved SSID and password. If the connection fails due to a network issue, the lock will retry at regular intervals, as defined by the wifiDwnldRtyTmIntv tag. The number of retries is determined by the wifiDwnldRtyCnt tag.

If all retry attempts fail and the lock reaches the maximum retry count, it will cease further retries. The next retry attempt will then be scheduled after 24 hours. This 24-hour retry mechanism is initiated by the lock automatically.

Audit Trail

The following structure represents a JSON data structure for an audit trail block that contains 3 audits. Note that this data structure does not contain a parent tag. This is because audit trails are being requested by the host, therefore the data type is not needed.

```
{
  "audits": [
    {
      "event": "1122334455",
      "time": 20131113091752
    },
    {
      "event": "008A0000",
      "time": 20131113092022
    },
    {
      "event": "00860000",
      "time": 20131113092513
    }
  ],
  "nxtDbVerTS": "0x0873828198485981"
}
```

Asynchronous Events

The following structure represents a JSON data structure that contains an asynchronous event.

```
{
  "asyncEvent": {
    "event": "00830000",
    "time": 20131113091752
  }
}
```

Device Settings

This section gives an example of how the data is structured by the device when a request is made for the device parameters. These parameters are broken down into 5 different components so that redundant tags can be used.

Example Data Structure

When a mobile device connects to a device, the device reports a set of parameters to the mobile device. This is an example of the data structure that would be returned by the device.

```
{
  "dvcProfile": {
    {
      "baseType": "nde",
      "key": "NewAPCfg",
      "value": "1"
    },
    "battV": {
      "main": "5.87",
      "li": "2.99"
    },
    "fwVer": {
      "main": "01.03.11",
      "credRdr": "01.02.10",
      "ble": "01.02.10",
      "wifi": "01.05.10",
      "mainBl": "00.01.03",
      "credRdrBl": "00.01.01"
    }
  },
}
```

```

"wifiStngs":
{
    "ssid":"01.03.11",
    "psswd":"\@113G10N",
    "usrNm":"Br549",
    "wifiSec":"entrprs",
    "encrypt":"wep",
    "kyIndx":4,
    "dscvryTyp":"ipAddr",
    "scrCnn": "http",
    "ip":"196.23.43.113",
    "dns":"firstChoice",
    "altDNS":"secondChoice"
},
"apCfgNew":
{
    "apCfgNewEn":"T",
    "ssidNew":"02.04.12",
    "psswdNew":" G10N\@113",
    "usrNmNew":"Cr649",
    "wifiSecNew":"entrprs",
    "apCfgNewStrtTm":"20140527160629"
},
"lckPrmtrs":
{
    "nm":"corner office",
    "mdl": "nde",
    "lckSn":"1122334455",
    "mnSn":"2233445566",
    "mfgDt":"20140607",
    "daysInUse":145,
    "hwVer":"c1",
    "type":"strm",
    "relock":1,
    "doorProp":20,
    "ada":30,
    "firstManIn":0,
    "dstEnable":0,
    "dstStart": "3022",
    "dstEnd": "B012",
    "battFail": "sec",
    "rtcTime": "20140527140629",
    "dbDwnldTm":"20140528020000"
},
"rdrPrmtrs":
{
    "bprEn": "T",
    "sn":"4455667788",
    "mfgDt":"20140607",
    "daysInUse":145,
    "hwVer":"c1"
}
}

```

Persistent Connection JSON Data Structures and IP API

The following section defines JSON structures that are used in communication with the ENGAGE Gateway and reader-controllers.

None of the following JSON structures should be used to directly communicate with an ENGAGE device except for reader-controllers.

Persistent Connection Edge Device JSON Data structures

Device Control

These device JSON structures are only accessible via the Gateway or reader-controller. **These data structures cannot be written directly to a device with the exception of reader-controller devices. Additionally, the device control JSON structures do not apply to the Gateway itself. Rather they apply to the devices linked to the Gateway.**

➔ **NOTE:** lockLEDFlash and lockState are independent. The client issuing the command can send only lockState, only lockLEDFlash or both. If the lock state and lock are combined the lock will first flash default.

Table 14: Device Control

Tag	JSON Short Tag	Type/Max-Length (ASCII bytes)	Valid Values	Device Exclusions
Device Control	lockControl	Object (parent = lockData)		GWE , GWE2
Tamper Clear	tmprClr	String/1	"T", "F"	GWE , GWE2
Tamper Re-Baseline	tmprRebaseline	String/1	"T", "F"	GWE , GWE2
Device Reset	dvcReset	String/1	"T", "F"	GWE , GWE2
Change Lock State (decision at door only - separate commands)	lockState	Object (parent = lockControl)		GWE , GWE2
Next Lock State	nextLockState	String/15	"secure", "passage", (uses device's internal relock delay) "momentaryUnlock", (not on RMRU) "frozenSecure", "frozenPassage", "auxSecure", "auxPassage", "auxMomntUnlck", "mainAuxMomUnlck", "mainAuxPassage", "mainAuxSecure", "clrOverride" NOTE: Clear override command reverts the lock state to a previously scheduled event in progress or to a secured state if no scheduled events in progress (Supported on rc11,rc15,rck15 only)	GWE, GWE2
Momentary Unlock	momUnlckTm	Unsigned Int/3	0 - 255	GWE , GWE2
User Interface	lockLedFlash	Object (parent = lockControl)	N/A	GWE , GWE2
LED Control Enable	ledName	String/16	"reader", (Primary LED on RMRU) "insidePushButton" (not on RMRU)	GWE , GWE2
LED Pattern Parameters	pattPrmtrs	Object (parent = lockLedFlash)		GWE , GWE2
LED Blink Colors	ledColors	String/12	"green", "red", "amber(RC11/15)", "blue(RC11/15)", "greenThenRed", "redThenGreen", "redThenAmber"	GWE , GWE2
LED On Interval	onTime	String/6	"100", "500", "1000", "always"	GWE , GWE2
LED Off Interval	offTime	String/6	"100", "500", "1000", "always"	GWE , GWE2
LED Repeat Parameters	repeatPrmts	Object (parent = lockLedFlash)		GWE , GWE2
Pattern Repeat	repeatPatt	String/20	"stop", "once", "indefiniteIncDelay", "indefiniteConstDelay"	GWE , GWE2

Tag	JSON Short Tag	Type/Max-Length (ASCII bytes)	Valid Values	Device Exclusions
Number Times to Repeat Pattern	repeatNum	Unsigned Int/1	1,2,3, or 4	GWE , GWE2
LED Interval Time	repeatInterval	Unsigned Int/2	0,20,40,60	GWE , GWE2
Beeper Parameters	bprPrmtrs	Object (parent = lockControl)		GWE, GWE2
Beeper On Interval	onTime	String/6	"100", "500", "1000", "always"	GWE , GWE2
Beeper Off Interval	offTime	String/6	"100", "500", "1000", "always"	GWE , GWE2
Number of Times to Repeat Pattern	repeatNum	UnsignedInt/1	1,2,3, or 4	GWE , GWE2
General Purpose Outputs	gpo	Object (parent - lockControl)		GWE , GWE2
Output Type	oType	String/4	"ttl1", "ttl2"	GWE , GWE2
Output Active	oActive	String/3	"on", "off"	GWE , GWE2

Device Status

The Device Status JSON structures are used only to represent a device status from a Gateway or reader-controller to an IP host. This JSON cannot be queried directly from a device with the exception of reader-controllers.

Individual device status varies based on the device model type. The JSON structures shown in **Table 15** are a master list of possible status tags, but are dependent on the hardware of the device.

Tag	Short Tag	Type/Length (ASCII bytes)	Value
Device Status	lockStatus	Object (parent = lockData)	
Device State	currentLockState	String/9	"sec", "pass", "momUnlck", "holRstrct", "frznSec", "frznPass", "frznDne", "dne", "dvcFault", "prvcySecd" NOTE: State is set for privacy secure or deadbolt secure. NOTE: Control, ControlB and ControlBM locks support only "frznSec", "momUnlck", "sec" state values
Device Event	currentDeviceEvent	String/25	"Standard-momentaryUnlock", "Standard-toggle", "Standard-onetime", "Standard-lockdown", "Standard-freeze", "Standard-passthru", "Override-momentaryUnlock", "Override-passage", "Override-secure", "Override-frozenPassage", "Override-frozenSecure", "Override-Clearoverride", "Schedule-holidayPass", "Schedule-grantAccess", "Schedule-holidaySec", "Schedule-holidayRstrctSec", "Schedule-autoUnlock", "Schedule-autoLock", "Schedule-autoLock-Badge", "Standard-normal-badge", "Standard-passthru-badge", "Sched-clrovr-holPass", "Sched-clrovr-holSec", "Sched-clrovr-holRstrctSec", "Sched-clrovr-autUnlck", "Sched-clrovr-autLck", "Sched-clrovr-autLck-Bdge", "Aux-In-Press", "Rex-In-Press", "Rex-In-Press_No_UI", "Tamper-secure", "Tamper-End", "Unsupport_event" NOTE: For RC11/15 only.
Sensor Status	sensorStatus	Object (parent = lockStatus)	N/A
Lock Status	locked	String/1	"T", "F"
Aux Status	auxPosition	String/1	"T", "F"
Alarm Status	alarmPosition	String/1	"T", "F"
Lock Clutch Position (AD and RMRU Only)	lockClutched	String/1	"T", "F"
Battery Critical	primaryBatteryStatus	String/12	"normal", "low", "critical", "notInstalled"

Tag	Short Tag	Type/Length (ASCII bytes)	Value
Real Time Clock Backup Battery	rtcBatteryStatus	String/12	"normal","low", "critical", "notInstalled" Deprecated for ENGAGE 6.0 and later as no ENGAGE devices have an rtc battery (coin cell back up)
Door Position	doorOpen	String/1	"T", "F"
IPB Active	ipbActive	String/1	"T", "F"
Request To Exit Active	rexActive	String/1	"T", "F"
Request to Enter Active	renActive	String/1	"T", "F"
Remote Release	relActive	String/1	"T", "F"
Deadbolt Position	deadBoltExtended	String/1	"T", "F"
Key Override Active	keyOverrideActive	String/1	"T", "F"
FDR Button Active	fdrActive	String/1	"T", "F"
Tamper Lid Open	tamperOpen	String/1	"T", "F"
Magnetic Tamper Detected	magTamperDetected	String/1	"T", "F"
Reader Tamper Detected	rdrTamperDetected	String/1	"T", "F"
TTL1 Active	ttl1Active	String/1	"T", "F"
TTL2 Active	ttl2Active	String/1	"T", "F"
Beeper Active	bprActive	String/1	"T", "F"
LED Active	ledActive	String/1	"T", "F"

Gateway Edge Device JSON Data Structures

Gateway Edge Device Link List

The Gateway link list is an array object which contains an array entry for each of the Gateway linked devices.

Table 15: Gateway Edge Device Link List				
Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Access Point Link List	linkList	Array (parent)	Each entry contains child JSON structures	
Unique Identifier that a host/mapp uses to reference a lock (or other device linked to Gateway)	linkId	String/32	Host system device identifier value – alpha-numeric 32 characters	
			NOTE: For RSI mode linkId is numeric ONLY and limited to 0 to 254, excluding 170	
Status of communication between gateway and device	linkCommStatus	String/16	"disconnected", "connected", "advertising" (reserved for future use)	
Device Name	deviceName	String/28	Max 28 characters	
Model Type of Linked Device (self-reported by the linked device)	modelType	String/32	"nde", "control", "enrollment reader", "ade", "gateway", "3rd party", "le", "rmru", "cte", "ndeb", "controlb", "controlbm", "leb"	MTB, MTKB

Table 15: Gateway Edge Device Link List

Tag	Short Tag	Type/Length (ASCII bytes)	Value	Device Exclusions
Device Id	deviceId	String/32	Lower 8 bytes of edge device serial number: "A0B1000000000005"	
Edge Device BLE MAC address	macAddr	String/32	MAC address of edge device: "00:07:80:78:5B:EF"	
MAC Type	macType	String/16	"public", "random"	
Battery Voltage	battV	String	JSON object "battV" defined in Retrievable Lock parameters	CTE
Line Voltage	lineV	String	JSON object "lineV" defined in Retrievable Lock parameters	R,N,L,SC,SCB
Signal Quality	signalQuality	String/8	"Low", "Med", "High", "Unknown"	

Example: Linked Device Data

```
{
  "linkList": [
    {
      "linkId": "dev00000",
      "linkCommStatus": "connected",
      "deviceName": "Front Door",
      "modelType": "nde",
      "deviceId": "a000000000000123",
      "macAddr": "00:07:80:78:2F:88",
      "macType": "public",
      "battV": {
        "main": " 5.52",
        "li": "0"
      }
    }
  ]
}
```

Linking Edge Devices

When linking an Edge Device to a specific Gateway, the Gateway responds with the parent tag linkInfo as defined in **Table 17**.

Table 16: Linking Edge Devices

Tag	Short Tag	Type/Length (ASCII bytes)	Value
Access Point Linked Device Info	linkInfo	Object	
Unique Identifier that a host/mapp uses to reference a lock (or other device linked to Gateway)	linkId	String/32	Host system device identifier value – alpha-numeric 32 characters
			NOTE: For RSI mode linkId is numeric ONLY and limited to 0 to 254, excluding 170
Device Id	deviceId	String/32	Lower 8 bytes of edge device serial number: "A0B1000000000005"
Device Name	deviceName	String/28	Max 28 characters
Status of communication between gateway and device	linkCommStatus	String/16	"disconnected", "connected", "advertising" (reserved for future use)
Model Type	modelType	String/16	"nde", "control", "enrollment reader", "ade", "gateway", "3rd party", "le", "rmru", "cte", "ndeb", "controlb", "controlbm", "leb"

Example: JSON body response from POST /edgeDevices

```
{
  "linkInfo": {
    "linkId": "dev00000",
    "deviceId": "a0000000000000123",
    "deviceName": "Front Door",
    "linkCommStatus": "connected",
    "modelType": "nde"
  }
}
```

Linked Edge Device Data

To configure and control an online edge device that is linked to an ENGAGE Gateway, the host may need to know the current device parameters. This data structure builds on device parameters that are provided in the **Device JSON Messages Section**, and the real-time access control structures that follow in the **Gateway JSON Data Structures Section**. To configure and control an online edge device that is linked to an ENGAGE Gateway, the Gateway provides the following edge device information:

Table 17: Linked Edge Device Data			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Access Control Device Data	edgeDevice	Array (parent) of device data objects with linkId	
Unique Identifier that a host/mapp uses to reference a lock (or other device linked to a Gateway)	linkId	String/32	Host system device identifier value – alpha-numeric 32 characters
			NOTE: For RSI mode linkId is numeric ONLY and limited to 0 to 254, excluding 170
JSON data structures, (Insert device JSON blocks such as database, configs, settings, audits, or state change	“db”, “config”, “battV”, “fwVer”, “lockPrmtrs”, “rdrPrmtrs”, “rsiPrmtrs” “audits”, “asyncEvent”, “lockControl”, “lockStatus”	Parent object corresponding to device JSON structure.	See the Device JSON Messages Section for definitions of JSON tags. NOTE: lockControl and lockStatus are defined in the next section.

➔ **NOTE:** For ENGAGE 6.1 and later devices, devices configured for IP or stand-alone communication may not transmit their RSI parameters (rsiPrmtrs) or any children tags under the RSI parameters as they are not relevant to IP or stand-alone operation. The device will accept any changes to the RSI parameters if they are provided to the device, however the device will not provide the updated information back to the host.

Gateway JSON Data Structures

Gateway Scan List

The scan list object contains a report detailing ENGAGE devices that are within BLE wireless distance of the Gateway.

Table 18: Gateway Scan List			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Scan for uncaptured or unlinked devices (no linked devices)	gatewayScanList	Array of objects (parent)	Each array entry is an object containing the following objects
Main Serial Number	mainSn	String/16	Max 16 characters
Device Name	deviceName	String/28	"xxxx.....xxxx"
Signal Quality	signalQuality	String/8	"Low", "Med", "High"
Model Type	modelType	String/16	"nde", "control", "enrollment reader", "ade", "gateway", "3rd party", "le", "rmru", "cte", "ndeb", "controlb", "controlbm", "leb"

Gateway Configuration

Table 19: Gateway Configuration			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Configuration Block	gatewayConfig	Object (top level parent)	N/A
Gateway Config	genGatewayConfig	Object (parent = gatewayConfig)	
Gateway Name	deviceName	String/28	Max 28 characters
Host Connection Type	hostProtocol	String/9	"rs485Rsi", "ipEngage" "ipClient"
Enable Shell Debug Port	sshEnable	String/1	"true","false"
Clock Time	rtcTime	String/14	"YYYYMMDDHHMMSS", YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59). Example: "20140710145030" means July 10, 2014, 2:50:30 pm.
Daylight Saving Time	dstEnable	String/5	"false"=DST not observed, "true"=DST observed
DST Start	dstStart	String/4	Hour that clocks are moved ahead by 1 hour. 3=daylight month start, 0=daylight day of week start, 2=daylight week # start, 2=daylight hour start, US default is 3022.
DST End	dstEnd	String/4	Hour that clocks are moved behind by 1 hour. B=standard month start (1=Jan,...). 0=standard day of week start. 0 = Sun. 1=standard week # start. (1=1st, 5=last). 2=standard hour start. US default is B012.
Firmware Address	fwurl	String/64	Max 256 characters NOTE:FW Version 1.53.18 and earlier - Max 64 characters
Firmware Download Time	fwDwnldTm	String/14	YYYYMMDDHHMMSS – "20140527040000", May 27, 2014, 4:00:00 am, Value of "0" results in immediate download.

Table 19: Gateway Configuration			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Firmware Implement/ Update Time	fwImplTm	String/14	YYYYMMDDHHMMSS – “20140603030000”, June 3, 2014, 3:00:00 am, Value of “0” results in immediate update/implement.
IP Mode Settings	gatewayIpModeConfig	Object (parent = gatewayConfig)	N/A
Interface ID	interfaceId	String/4	“eth0”, “wlan0”
Interface Enable	interfaceEnable	String/1	“T” = enables interface operation. “F” = disables interface operation.
Gateway Discovery Method	discoveryMethod	String/6	“dhcp”, “staticIP”, “zeroConf”
Fixed IP Address, (only applies if discovery method is set to staticIP)	fixedIpAddr	String/15	“xxx.xxx.xxx.xxx”
Default Gateway IP Address (only applies if discovery method is set to staticIP)	defGatewayIpAddr	String/15	“xxx.xxx.xxx.xxx”
Netmask (only applies if discovery method is set to staticIP)	netmask	String/15	“xxx.xxx.xxx.xxx”
IP DNS Service (only applies if discovery method is set to staticIP)	ipDnsAddr	String/15	“xxx.xxx.xxx.xxx”
Alternate IP DNS Service (only applies if discovery method is set to staticIP)	altDnsAddr	String/15	“xxx.xxx.xxx.xxx”
IP Server URL (only applies if hostProtocol is ipClient)	ipServerURL	String/64	Max 64 characters
CA Server URL (only applies if hostProtocol is ipClient)	caServerURL	String/64	Max 64 characters
Keep Alive Time (only applies if hostProtocol is ipClient)	wsKeepAlive	String/4	“1” to “3600” seconds
EdgeDevice FW Update related configs	edgeDeviceFwUrls	Array of Objects (parent = gatewayConfig)	N/A
Firmware Address	fwurl	String/64	Max 256 characters
Device Model Type	mdl	String/15	“nde”, “le”, “control”, “ndeb”, “leb”, “controlb”, “controlbm”, cte, rmr
Firmware Download Time	fwDwnldTm	String/14	YYYYMMDDHHMMSS – “20140527040000”, May 27, 2014, 4:00:00 am, Value of “0” results in immediate download of the device’s firmware image to the Gateway.
Firmware Transfer Time	fwImplTm	String/14	YYYYMMDDHHMMSS – “20140603030000”, June 3, 2014, 3:00:00 am, Value of “0” results in immediate start of transfer of the image to the device.
Firmware Version	fwVer	String/9	“xx.xx.xx” Example: “02.08.05”

Edge Device Firmware Update via Gateway

Gateway firmware version 01.49.12 and later supports updating linked edge devices’ firmware for sites which are running 410-IP mode with the edgeDeviceFwUrls JSON tag as defined above in Section 4.2.2. All linked edge devices matching the Device

Model Type tag will be updated to the requested firmware. Please note that if the specified Firmware Version (fwVer) does not match the actual version downloaded from the Firmware Address (fwurl), the edge device will not be updated.

Status of the firmware update can be monitored through the resource GET /gateway/deviceInfo. The children tags, edgeDeviceFwUrls, status, and extendedStatus contain all relevant information regarding the firmware update.

Gateway firmware versions 1.52.16 and later support firmware download of a secure connection. In this case the host must provide the fwurl specified with "https://" and the required port (ex. "https://10.200.136.198:444/gw_image.bin") If no port is specified in the request, it is assumed that the user intends to use port 443.

When completing a firmware download from the host over HTTPS the gateway will utilize the port specified (or 443) and complete the download via TLS, however the gateway will not download or validate the certificate from the host.

The example below illustrates initiating a firmware update for all linked NDE edge devices to take effect immediately.

Example: application/json

```
{
  "gatewayConfig": {
    "edgeDeviceFwUrls": [
      {
        "fwurl": "http://192.168.43.24/nde_02.08.13_Releasepkg.02.08.13.bin",
        "mdl": "nde",
        "fwDwnldTm": "0",
        "fwImplTm": "0",
        "fwVer": "02.08.13"
      }
    ]
  }
}
```

Device Information Records (Gateway)

ENGAGE devices report all retrievable parameters in one large block and includes some parameters that may not directly apply to the device or its mode of operation. For example; RSI protocol capable devices might report RSI parameters when the device is not configured for RSI protocol operation.

Table 20: Device Information Records (Gateway)

Tag	Short Tag	Type/Length (ASCII bytes)	Value
Gateway Device Info	gatewayDeviceInfo	Object (top level parent)	
Device Profile	dvcProfile	Object (parent = gatewayDeviceInfo)	N/A
Base Type	baseType	String/3	"gwe", "gwe2"
Extensions	ext	Array of objects (parent = dvcProfile)	N/A
Extension's name	Key	String/15	"gweSiteSurvey"
Extension's custom argument	Value	String/15	"1"
Firmware Versions	fwVer	Object (parent = gatewayDeviceInfo)	N/A
Main	main	String/8	"xx.xx.xx"
Bluetooth Module Secondary	ble2	String/8	"xx.xx.xx"
Main Bootloader (uboot)	mainBl	String/8	"xx.xx.xx"
WiFi MAC Address	macAdr	String/17	"xx.xx.xx.xx.xx.xx"
Ethernet MAC Address	ethernetMacAdr	String/17	"xx.xx.xx.xx.xx.xx"

Table 20: Device Information Records (Gateway)			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
WiFi Settings	wifiStngs	Object (parent = gatewayDeviceInfo)	N/A
WiFi enable	wifiEnable	String/1	“T” = enables WiFi operation, “F” = disables WiFi operation
SSID	ssid	String/32	Max 32 characters
AP Password / Security Key	psswd	String/64	Max 64 characters
User Name	usrNm	String/16	Max 16 characters
WiFi Security	wifiSec	String/7	“prsnl”, “entrprs”, “open”, “wep”
Key Index	kyIndx	String/1	1,2,3,4
Gateway Parameters	gatewayParameters	Object (parent = gatewayDeviceInfo)	N/A
Host Connection Type	hostProtocol	String/8	“rs485Rsi”, “ipEngage”, “ipClient”
Name	deviceName	String/28	“xxxx.....xxxx”
Model	mdl	String/4	“gwe”, “gwe2”
Gateway Main Serial Number	mainSn	String/16	Max 16 characters
Hardware Version	hwVer	String/6	“xxxxxx”
Days In Use	daysInUse	String/5	0-65536
Gateway Uptime (time since boot)	gwUpTime	String/3	xxx (days)
Clock Time	rtcTime	String/14	“YYYYMMDDHHMMSS”, YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59), Example: “20140710145030” means July 10, 2014, 2:50:30 pm
Enable Shell Debug Port	sshEnable	String/5	“true”, “false”
Daylight Saving Time	dstEnable	String/5	“false”=DST not observed, “true”=DST observed
DST Start	dstStart	String/4	Hour that clocks are moved ahead by 1 hour. B=standard month start (1=Jan). 0=standard day of week start. 0 = Sun. 1=standard week # start. (1=1st, 5=last). 2=standard hour start. US default is 3022.
DST End	dstEnd	String/4	Hour that clocks are moved behind by 1 hour. 3=daylight month start. 0=daylight day of week start. 2=daylight week # start. 2=daylight hour start. US default is B012.
Firmware Address	fwUrl	String/64	Max 64 characters
Firmware download Time	fwDwnldTm	String/14	YYYYMMDDHHMMSS – “20140527040000” = May 27, 2014, 4:00:00 am
Firmware Implement Time	fwImplTm	String/14	YYYYMMDDHHMMSS – “20140603030000” = June 3, 2014, 3:00:00 am
Update Edge Device Time	updateEdgeDeviceTime	String/5	“true”, “false”
IP Interfaces	ipv4Interfaces	Object (parent = gatewayDeviceInfo)	N/A

Table 20: Device Information Records (Gateway)			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Ethernet Interfaces	ethernetProperties	Object (parent = ipv4Interfaces)	N/A
Assigned IP Address	assignedIpAddr	String/15	“xxx.xxx.xxx.xxx”
Default Gateway IP Address	defGatewayIpAddr	String/15	“xxx.xxx.xxx.xxx”
Netmask	netmask	String/15	“xxx.xxx.xxx.xxx”
WiFi Client Properties	wifiClientProperties	Object (parent = ipv4Interfaces)	N/A
Gateway IP Address	Assigned IP Address	String/15	assignedIpAddr
Default Gateway IP Address	Default Gateway IP Address	String/15	defGatewayIpAddr
Netmask	Netmask	String/15	netmask
RSI Settings	gatewayRsiModeConfig	Object (parent = gatewayDeviceInfo)	N/A
RSD Address	rsdAddress	Unsigned Int/3	0 to 254, where 170 is not allowed
Low Door Address	lowDoor	Unsigned Int/3	0 to 254, where 170 is not allowed, must be less than high door setting; default of 0
High Door Address	highDoor	Unsigned Int/3	0 to 254, where 170 is not allowed, must be greater than Low Door setting; default of 15
IP Mode Settings	gatewayIpModeConfig	Object (parent = gatewayDeviceInfo)	N/A
Gateway Discovery Method	discoveryMethod	String/8	“dhcp”, “staticIP”, “zeroConf”
Static IP Address, (only applies if discovery method is set to staticIP)	fixedIpAddr	String/15	“xxx.xxx.xxx.xxx”
Default Gateway IP Address (only applies if discovery method is set to staticIP)	defGatewayIpAddr	String/15	“xxx.xxx.xxx.xxx”
Netmask (only applies if discovery method is set to staticIP)	netmask	String/15	“xxx.xxx.xxx.xxx”
IP DNS Service (only applies if discovery method is set to staticIP)	ipDnsAddr	String/15	“xxx.xxx.xxx.xxx”
Alternate IP DNS Service (only applies if discovery method is set to staticIP)	altDnsAddr	String/15	“xxx.xxx.xxx.xxx”
IP Server URL (only applies if hostProtocol is ipClient)	ipServerURL	String/64	Max 64 characters
CA Server URL (only applies if hostProtocol is ipClient)	caServerURL	String/64	Max 64 characters
Keep Alive Time (only applies if hostProtocol is ipClient)	wsKeepAlive	String/4	“1” to “3600” seconds

Table 20: Device Information Records (Gateway)			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Gateway Feature List	gatewayFeatureList	Array of strings (parent = gatewayDeviceInfo)	["rsiMode", "ipServerMode", "ipClientMode", "lockFwUpgrade"]
EdgeDevice Firmware Update Status	edgeDeviceFwUrls	Array of Objects (parent = gatewayDeviceInfo)	N/A
Firmware Address	fwurl	String/64	Max 64 characters
Model Type	mdl	String/15	"nde", "le", "control", "ndeb", "leb", "controlb", "controlbm", "cte", "rmru"
Firmware Download Time	fwDwnldTm	String/14	"YYYYMMDDHHMMSS", YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59), Example: "20140710145030" means July 10, 2014, 2:50:30 pm
Firmware Transfer Time	fwImplTm	String/14	"YYYYMMDDHHMMSS", YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59), Example: "20140710145030" means July 10, 2014, 2:50:30 pm
Firmware Version	fwVer	String/9	"xx.xx.xx" Example: "02.08.05"
Firmware Update Status (for the devices of model type indicated in mdl above)	status	Objects (parent = edgeDeviceFwUrls)	N/A
Firmware Download Status (Firmware image download to the Gateway)	fwDwnld	String/24	"pendingDwnld" = Device firmware image download to the Gateway from the server is pending. "inProgress" = Device firmware image download to the Gateway from the server is in progress. "completed" = Device firmware is downloaded to the Gateway. "failed" = Device firmware image download to the Gateway from the server is failed. "canceled" = Device firmware image download to the Gateway from the server is canceled. "" = no status available.

Table 20: Device Information Records (Gateway)			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Firmware Update Status (for the model type indicated in the mdl above)	fwUpdate	String/24	“pendingUpdate” = Firmware update to the devices (of the model type indicated in the mdl above) is pending. “inProgress” = Firmware update to the devices (of the model type indicated in the mdl above) is in progress. “suspended” = Firmware update to the devices (of the model type indicated in the mdl above) is suspended, will be resumed later. “completed” = Firmware update to the devices (of the model type indicated in the mdl above) is completed successfully. “pendingVerification” = Firmware is transferred to all the devices (of the model type indicated in the mdl above) successfully, one or more devices are installing the firmware and Gateway is waiting for verification of the firmware update. “pendingReschedule” = Gateway has received a new firmware update request while still in the process of completing a firmware update to the edge device. “pendingReschedule” will be set momentarily while the current update is canceled and the new update is started. “pendingCancel” = Gateway has received a request to cancel the current firmware update. “pendingCancel” will be set momentarily while the current update is canceled. “failed” = Firmware update to one or more devices (of the model type indicated in the mdl) is failed. “canceled” = Firmware update to the devices (of the model type indicated in the mdl) is canceled. “” = no status available.
Extended Firmware Update Status for each Device (for the model type indicated in mdl above)	extendedStatus	Array of objects (parent = edgeDeviceFwUrls)	N/A
Unique Identifier that a host/mapp uses to reference a lock (or other device linked to the Gateway)	linkId	String/32	Device identifier value – alpha-numeric 32 characters
Device Id	deviceId	String/16	Lower 8 bytes of edge device serial number: “A0B1000000000005”
Device name	deviceName	String/28	
Device model type	mdl	String/5	“nde”, “le”, “control”, “ndeb”, “leb”, “controlb”, “controlbm”, “cte”, “rmru”
Device firmware version	fwVer	String/9	“xx.xx.xx” Example: “02.08.05”

Table 20: Device Information Records (Gateway)

Tag	Short Tag	Type/Length (ASCII bytes)	Value
Device firmware update status	fwUpdateStatus	String/24	“pendingUpdate” = Firmware image transfer to the device is pending. “inProgress” = Firmware image transfer to the device is in progress. “suspended” = Firmware transfer is suspended, will be resumed later. “suspended-lowbatt” = Firmware transfer is suspended due to device being in low battery condition, will be resumed later once battery is replaced. “suspended-busy” = Firmware transfer is suspended due to other JSON data (such as configs, database) being transferred to the device, will be resumed later. “completed” = Firmware update to the device is completed successfully. “pendingVerification” = Firmware is transferred to the device successfully, device is installing the firmware and Gateway is waiting for verification of the firmware update. “pendingVerification-lowbatt” = Firmware is transferred to the device successfully, device has not installed the firmware due to low battery. Replace the battery to continue installing the firmware. “failed” = Firmware update to the device is failed. “canceled” = Firmware update to the device is canceled. “” = no status available.
Firmware transfer start time for the device (Gateway Time)	fwUpdateStartTm	String/14	“YYYYMMDDHHMMSS”, YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59), Example: “20140710145030” means July 10, 2014, 2:50:30 pm
Firmware transfer complete percentage.	fwTransferCmpltPercentage	Unsigned Int/3	0 to 100
Firmware update complete time (Gateway Time), (last successful update via Gateway)	fwUpdateCmpltTm	String/14	“YYYYMMDDHHMMSS”, YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59), Example: “20140710145030” means device was last upgraded successfully via Gateway on July 10, 2014, 2:50:30 pm
Firmware update cancel time (Gateway Time)	fwUpdateCancelTm	String/14	“YYYYMMDDHHMMSS”, YYYY = Year, MM = Month (01 – 12), DD = Day (01 – 31), HH = Hour (00 – 23), MM = Minutes (00 – 59), SS = Seconds (00 – 59), Example: “20140710145030” means July 10, 2014, 2:50:30 pm
No. of firmware update retries made by the Gateway (in case of update failure)	retries	Unsigned Int/3	0 to 2

Gateway IP Host Authentication Setup

Used between a Gateway and IP host during initialization of the Gateway using default logins. This object is returned to the IP Host during a GET from /gateway/newCredentials.

Table 21: Gateway IP Host Authentication

Tag	Short Tag	Type/Length (ASCII bytes)	Value
New Login Credentials	credentials	Object (top level parent)	
Unique host login to gateway	usr	String / 32	gateway_<serial number>
Randomly generated password for host login to gateway	pwd	String / 32	randomly generated password

Site Survey Results

Gateway reports the results of a site survey once the site survey is completed.

Table 22: Site Survey Results			
Tag	Short Tag	Type/Length (ASCII bytes)	Value
Time	Time	String/21	"[2018/03/05 12:19:11]"
Version	Version	String/8	"1.0"
Site survey	site_survey	Array of objects	N/A
Link ID	linkId	String/15 (parent = site_survey)	"dev00000"
Device name	deviceName	String/28 (parent = site_survey)	"xxx...xxx"
Model	model	String/15 (parent = site_survey)	"nde", "le", "control", "ndeb", "leb", "controlb", "controlbm", "cte", "rmru"
Signal Quality	signalQuality	Unsigned int/3 (parent = site_survey)	75
Stop light mapping	stoplightMapping	Object	N/A
Red	red	String/3 (parent = stoplightMapping)	"red"
low	low	Unsigned int/3 (parent = red)	0 - 100
high	high	Unsigned int/3 (parent = red)	0 - 100
Yellow	yellow	String / 8 (parent = stoplightMapping)	"yellow"
low	low	Unsigned int/3 (parent = yellow)	0 -100
high	high	Unsigned int/3 (parent = yellow)	0 -100
Green	green	String / 8 parent = stoplightMapping)	"green"
low	low	Unsigned int/3 (parent = green)	0 - 100
high	high	Unsigned int/3 (parent = green)	0 - 100

Appendix A: Credential Tag Name Definitions

“norm” (Normal)

A normal credential opens a door for a specified time. The time span is defined by the relock delay. The normal function works on all devices.

“freeze” (Freeze)

A freeze credential disables the credential reader. After a freeze credential has been used on a lock, only a pass through credential will operate the lock. Present a freeze credential to return the lock to an operational state.

“toggle” (Toggle)

A toggle credential opens a door and leaves it open until it is closed again by a toggle credential. It toggles a door between locked and unlocked.

“passThru” (Pass Through)

A pass through credential will always unlock a lock that is in secured lockout mode, regardless of how the lock was put in secured lockout mode.

“lockDown” (Lock Down)

A lockdown credential will always put the lock into the secured mode plus freeze credential functionality.

“oneTm” (One Time Use)

A onetime use credential opens a door only once with the normal function. Once a onetime use credential has been used on a door, it will no longer work on that door. It can still work on other doors, to which is has been assigned.

“dogd” (Dogged)

Toggle credential functionality for a crash bar exit device.

“suprv” (Supervised)

Supervised credentials allow access only when a second supervised credential is presented.

“delAlarm” (Delete with Alarm)

A delete with audit/alarm credential will log an audit entry that the credential was used. However, the credential will NOT unlock the door. If an alarm is available on the lock, then the alarm will sound. This functionality is the same as the “blocked” credential type.

“ctAux” (Ct Auxiliary)

A CT Aux credential operates the auxiliary relay of a CT Controller, but not the main relay. The time span the relay is activated is specified by the relock delay.

“ctMnAux” (Ct Main and Aux)

A CT Main and Aux credential operates both the auxiliary relay and main relay of a CT controller. The time span the relays are activated as specified by the relock delay.

“ctAuxToggle” (Ct Auxiliary Toggle)

A CT Auxiliary Toggle credential changes the state of the auxiliary relay from locked to unlocked, or vice versa, unless in a Freeze state.

“ctMnAuxToggle” (Ct Main + Aux Toggle)

A CT Main + Aux Toggle credential changes the state of both the main and auxiliary relays from locked to unlocked, or vice versa, unless in a Freeze state.

“ctMnAuxPassThru” (Ct Main + Aux Pass Through)

A CT Main + Aux pass through credential unlocks both the main and auxiliary relays regardless of state, for the time of the relock delay.

“updtFrmSvr” (Update from server)

Update from server credential will trigger the update from server in SF210 mode irrespective of lock state. In Sf200 mode this credential will be treated as invalid credential.

“updtFrmSvrNrm” (Update from server+Normal)

Update from server+Normal credential will behave as normal credential and then trigger the update from server in SF210 mode irrespective of lock state.

In SF200 mode this credential will be treated as normal credential only.

“runDiag” (run internal diagnostics and report audits)

The run diagnostics credential will trigger the built-in self-test routines of the device and initiate an update from server in order to report the results in SF210 mode irrespective of lock state.

In 410IP mode, resultant audits will be sent immediately to the Gateway.

In SF200 and 410RSI mode this credential will be treated as invalid.

“calDPS” (calibrate door position sensor)

The calibrate DPS credential will establish a new closed “home” position for the device. This calibration performs the same functionality as executed during commissioning for home position.

“blocked” (Blocked)

The functionality of the blocked credential type is identical to the “delAlarm” credential type as defined above.

Credential Tags Supported by Device

Table 23: Credential Tags Supported by Device	
Device	Tags supported
NDE	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”, “updtFrmSvr”, “updtFrmSvrNrm”, “blocked”
NDEB	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”, “updtFrmSvr”, “updtFrmSvrNrm”, “blocked”
SC	“norm”, “freeze”, “toggle” (functions as “norm”), “delAlarm”, “passThru”, “blocked”
SCB	“norm”, “freeze”, “toggle” (functions as “norm”), “delAlarm”, “passThru”, “blocked”
LE	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”, “updtFrmSvr”, “updtFrmSvrNrm”, “blocked”
LEB	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”, “updtFrmSvr”, “updtFrmSvrNrm”, “blocked”
CTE	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”, “updtFrmSvr”, “updtFrmSvrNrm”, “blocked”, “ctAux”, “ctAuxToggle”, “ctMnAux”, “ctMnAuxToggle”, “ctMnAuxPassThru”
RC	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”
XE360	“norm”, “freeze”, “toggle”, “passThru”, “lockDown”, “oneTm”, “delAlarm”, “blocked”

Appendix B: Device Specific Characteristics

Table 24: Device Specific Characteristics			
Feature	Holidays	Auto Event	Credential Schedule
NDE	32	16	16
NDEB	32	16	16
SC	N/A	N/A	16
SCB	N/A	N/A	16
LE	32	16	16
LEB	32	16	16
RMRU	32	16	N/A
CTE	32	16	16
RC	32	448	16
RC (extended)	32	448	32
XE360	32	16	16

Appendix C: Type Tags Supported by Device

Table 25: Type Tags Supported by Device	
Device	Tags supported
NDE	“strm”
NDEB	“apt”, “office”, “prvcy”, “strm”
LE	“apt”, “office”, “prvcy”, “strm”
LEB	“apt”, “office”, “prvcy”, “strm”
CTE	“strm”
RC	“strm”
XE360	“apt”, “office”, “prvcy”, “strm”

Appendix D: Model Type Supported by Device

Table 26: Model Type (mdl) Supported by Device		
Lock Parameter Model	Description	Firmware Update Device Model type used with GWE and GWE2
nde	NDE	nde
ndeb	NDE with BLE credential support and IPB	ndeb
lems	LE (Mortise with LED indicator)	le
lemb	LE (Mortise with IPB and LED indicator)	le
lemd	LE (Mortise deadbolt with LED indicator)	le
lebms	LE (Mortise with LED indicator) with BLE credential support	leb
lembmb	LE (Mortise with IPB and LED indicator) with BLE credential support	leb
lebmd	LE (Mortise deadbolt with LED indicator) with BLE credential support	leb
be467	Control Deadbolt	control
fe410	Control Interconnected Deadbolt	control
be467b	Control Deadbolt with BLE credential support	controlb
fe410b	Control Interconnected Deadbolt with BLE credential support	controlb
be467bm	Control Deadbolt with BLE credential support	controlbm
fe410bm	Control Interconnected Deadbolt with BLE credential support	controlbm
cte	Single Door Controller	cte
rm	Exit device with Remote Monitoring	rmru
ru	Exit device with Remote Undogging	rmru
mtb11	Mullion mount mobile enabled multi-technology reader	n/a
mtb15	Single gang mount mobile enabled multi-technology reader	n/a
mtkb15	Single gang mount mobile enabled multi-technology with keypad reader	n/a
rc11	Mullion reader controller	n/a
rc15	Single gang reader controller	n/a
rck15	Single gang reader controller with keypad reader	n/a
xe360i	XE360 (with LED indicator)	n/a
xe360b	XE360 (with IPB)	n/a
xe360d	XE360 (with Deadbolt)	n/a

About Allegion

Allegion (NYSE: ALLE) is a global pioneer in seamless access, with leading brands like CISA®, Interflex®, LCN®, Schlage®, SimonsVoss® and Von Duprin®. Focusing on security around the door and adjacent areas, Allegion secures people and assets with a range of solutions for homes, businesses, schools and institutions. Allegion had \$2.9 billion in revenue in 2019 and sells products in almost 130 countries.

For more, visit www.allegion.com

KRYPTONITE ■ LCN ■  ■ STEELCRAFT ■ VON DUPRIN