RISE V2G Advanced I – Master the Communication For Charging Electric Vehicles

2.1 Adjusting Global Values for EVCC and SECC

Welcome to Module 2! Today, we embark on the second part of this comprehensive course on future-proof communication technology for charging electric vehicles. In the first module, you gained practical insight into the inner mechanics of the ISO 15118 standard and its application in the RISE V2G software framework.

In video seven of Module 1, you also learned that Java classes with common functionalities for EVCC and SECC can be found in the RISE-V2G-Shared project.

In this module, you'll learn more about the auxiliary functions that both the EV and charging station need to successfully establish and run a charging session.

- For example, where can you adjust timing values for all messages the EV and charging station exchange during a communication session?
- What is the operating principle behind the EXI codec that converts and compresses an XML message into binary XML and vice versa?
- What utility classes do you need to know to fully understand message handling and the security features needed for Plug & Charge?

All these questions and more will be answered once you've finished Module 2. Without further ado, let's get to know the global settings that the RISE V2G project uses.

Navigate to the RISE-V2G-Shared project, expand the package "com.v2gclarity.risev2g.shared.enumerations" and double-click the Java file GlobalValues. This file holds various enumeration values. In computer programming, the enumerated data type, or enum, consists of a set of named values. These enumerator names are usually identifiers that behave as constants, meaning: they have been defined and cannot be changed somewhere else in the code. The GlobalValues get their name because these enumeration values are used globally throughout the RISE V2G project.

Specifying enum values makes particular sense if you want to assign an identifier to either a String value or to a numeric value that could change over time. The benefit is that you don't need to search for each and every occurrence of that String or numeric value in the code to change it everywhere. Instead, you can change the value for the enum entry once because the code looks up the unique identifier and then dynamically reads the underlying value that is associated with this enumeration name.

Let's go through the enumeration values in this GlobalValues class step by step.

First, we have two enum entries that store the paths to the configuration files for the EVCC and the SECC. We discussed the various configuration options in video three of the free RISE V2G Basics course. The current values for EVCC_CONFIG_PROPERTIES_PATH and SECC_CONFIG_PROPERTIES_PATH point to the root level of each project. When deploying RISE V2G on a charging station, for example, you might need to modify this setting to adapt the location of these properties files relative to the location of the JAR file. JAR is short for Java Archive, a single file that holds the RISE V2G code for either the EVCC or the SECC. If the properties file for the SECC lies in the same directory as the SECC JAR file, you can leave the values as is. The same holds for the EVCC.

Next, we have an enum value that stores the password for accessing a keystore. A keystore is a data container in Java that stores both cryptographic key material like private keys and digital certificates that hold the associated public keys. This data is protected against manipulation by using a password. This setting is extremely useful when testing the RISE V2G software. For example, at every ISO 15118 & CCS Testing Symposium, the organizers provide a pre-generated set of certificates and associated private keys. This ensures that all test symposium participants use the same cryptographic key material to run a Plug & Charge communication session. All participants are also given the password that grants access to the private keys and certificates. By modifying this setting once, you can easily change the password as needed without having to search for every occurrence in the code that needs access to a keystore; and there are quite a few.

Be careful about keystore passwords when deploying RISE V2G into a production environment like a charging station that will be sold to your customers. Setting the password in a central configuration file that can be read by anyone who gets access to this file would pose a huge security risk. During that stage, you'll need to find other ways to securely store the password. Usually, you would use what are called Hardware Security Modules, or HSMs, to store cryptographic key material and possibly even passwords.

The following two enumeration values, ALIAS_CONTRACT_CERTIFICATE and ALIAS_OEM_PROV_CERTIFICATE, are also related to digital certificates. It's not necessary to change these values; they are used to properly store and access the certificates in a keystore and do not need alteration.

The enum value CERTIFICATE_EXPIRES_SOON_PERIOD is used to update a contract certificate that is about to expire. In video three of Module 1, we went over the message sequence for AC and DC charging sessions. There you learned that the EV can opt to send a CertificateUpdateReq message if a set expiration interval is reached and if Plug & Charge has been chosen as its identification method. The value is given in number of days and here it is set to two weeks. This value is one suggestion listed in the ISO 15118-2 document, but you are free to change the value to any reasonable number of days without violating a requirement.

The next four entries in this enumeration list address the path to the keystore and truststore files for both EVCC and SECC. Keystores and truststores are represented by the same data type in Java, namely KeyStore. But they have different purposes: the keystore is used to store a side's own certificates, while the truststore is used to store the certificates of trusted third parties.

In an EV, for example, the contract certificate and OEM provisioning certificate would be in the keystore; we would use the truststore to store the V2G Root certificates, needed to verify the charging station's certificates during the setup of a TLS connection, also called a TLS handshake. In video four of the RISE V2G Basics course, we briefly touched on the various types of certificates that are applied for a Plug & Charge communication session. The RISE V2G Advanced Course

II, "Data Security and Plug & Charge in RISE V2G", will address the topic in much more detail.

It is important to note that the keystore and truststore files must not remain inside the Java Archive file when deploying RISE V2G onto an embedded controller inside a charging station or EV. The JAR file is read-only, so you cannot modify the files inside it. Yet keystore and truststore files need to be modifiable because certificates may be updated or new certificates installed during a communication session. For this reason, you'll need to make sure to place these files outside the JAR file, in the same way you did with the properties files.

The next four enumeration entries should remain as is. These are fixed values defined in the ISO 15118-2 requirements. They entail the multicast network address to send SECC Discovery Protocol messages, also known as SDP messages, from the EV to the charging station. These request messages use the UDP port 15118. The maximum size of each payload message and the protocol version of V2GTP messages are strictly defined by the standard and should not be changed.

Keep in mind that the V2GTP version is not the same as the ISO 15118 version. Edition 2 of ISO 15118 may use the same V2GTP version; however, it will change its unique protocol namespace to distinguish different versions of ISO 15118 in the SupportedAppProtocol messages.

The following ten enum values, related to the schema information, were also thoroughly discussed in video 11 of Module 1, when we went over the steps to adapt RISE V2G to the German DIN SPEC 70121 standard.

Next up are four values used in the SDP request and response messages. The first two values indicate whether an unsecured TCP connection will be used for the communication session or whether a cryptographically-secured TLS connection is required. The second two values define the reliability level of the transport layer protocol used to send and receive further messages. In principle, you can choose between the reliable TCP and the unreliable yet faster UDP protocol. However, ISO 15118 requirements prohibit the use of UDP for all V2G messages. UDP is used solely for the SDP request and response messages.

Last but not least, we have three additional entries that address the allowable V2GTP payload types, as defined in table 10 of ISO 15118-2. These are: EXI-encoded V2G messages and SDP request and response messages.

As you can see, some of these values are given in a textual String representation; others are defined as byte or numerical values. The remainder of this class makes sure that the respective value of each enum type is properly returned when accessing that enumeration value in code.

And that's it for the GlobalValues class. Now you know where to look for and change the predefined values used throughout the entire RISE V2G project, both on the EVCC and the SECC side.

In Module 1, you learned that there are certain timer values associated with messages and message sequences. These timer values are not stored in the GlobalValues class, but in a class called TimeRestrictions. We'll dive into this in video two, called Timings and Timeouts.

See you there!

Abbreviations Used in Advanced Course I

A Ampere
AC Alternating Current
AVLN Audio Video Logical Network

BPT Bi-directional Power Transfer

CCS Combined Charging System
CharIN Charging Interface Initiative e.V.
CI Communication Interface
CP Control Pilot

DC Direct Current
DHCP Dynamic Host Configuration Protocol

EIM External Identification Means
Enum Enumerated Data Type
EV Electric Vehicle
EVCC Electric Vehicle Communication Controller
EVCCID Electric Vehicle Communication Controller ID
EVSE Electric Vehicle Supply Equipment
EXI Efficient XML Interchange
EXIG EXI Grammar

GB Guobiao, National Standard in Chinese

HLC High-Level Communication
HMI Human-Machine Interface
HPC High Power Charger
HSM Hardware Security Module

ID Identifier
IDE Integrated Development Environment
IEC International Electrotechnical Commission
IP Internet Protocol
ISO International Organization for Standardization

JAR Java Archive
JAXB Java Architecture for XML Binding
JVM Java Virtual Machine

L1, L2, L3 Phases 1, 2, and 3 in an electrical installation

MAC address Media Access Control address

N Neutral (in an electrical installation)
NID Network ID

NMK Network Membership Key

OCPP Open Charge Point Protocol
OFDM Orthogonal Frequency-Division Multiplexing
OSI Open Systems Interconnection

PE Protective Earth
PLC Powerline Communication
PnC Plug & Charge
PWM Pulse Width Modulation

QR Quick Response

RCD Residual-Current Device
RFID Radio-Frequency IDentification

SA Secondary Actor
SAAC Stateless Address Autoconfiguration
SDP SECC Discovery Protocol
SECC Supply Equipment Communication Controller
SLAC Signal Level Attenuation Characterization
SUT System Under Test

TCP Transmission Control Protocol
TLS Transport Layer Security

UDP User Datagram Protocol
URL Uniform Resource Locator
URN Uniform Resource Name

V Voltage
V2GTP Vehicle-to-Grid Transfer Protocol

XML Extensible Markup Language
XSD XML Schema Definition