

fnv::soa: (h) Run-time Version Checking for API Messages (from ECG)

- 1. New APIs
 - 1.1. [SoaMessage::setPayloadWithVersion](#)
 - 1.2. [SoaMessage::parsePayloadWithVersion](#)
 - 1.3. [Code reference](#)

1. New APIs

SOA team provided C++ [APIs](#) to support [SOA API versioning](#).

```
class SoaMessage {
public:
    /**
     * Serializes the payload and sets the payload version to the message header
     * @param payload The payload
     * @return SoaErrorCode::NO_ERROR in case of success
     *         SoaErrorCode::ERROR_PARAM_NULL_REFERENCE if the payload is NULL
     *         SoaErrorCode::ERROR_SOA_MESSAGE_SERIALIZATION_FAILED if the payload cannot be serialized
     */
    template <class Proto>
    SoaErrorCode setPayloadWithVersion(const Proto& payload);

    /**
     * @brief Parses the payload and performs the payload version check.
     * The method compares version in the message header with the current Proto version.
     * The API is in permissive mode: does not return an error in case of version mismatch
     * @param[out] payload an instance of a payload to parse the message to
     * @return SoaErrorCode::NO_ERROR in case of success
     *         SoaErrorCode::ERROR_SOA_MESSAGE_PARSING_FAILED payload parsing failed while the version is
compatible
     *         SoaErrorCode::ERROR_UNSUPPORTED_VERSION payload parsing failed and version is not compatible
     *         SoaErrorCode::WARNING_UNSUPPORTED_VERSION - version is not supported, but parsing succeed
     */
    template <class Proto>
    SoaErrorCode parsePayloadWithVersion(Proto& payload);
}
```

The API allows verifying that protobufs are compatible at both the publisher and the subscriber ends.

1.1. [SoaMessage::setPayloadWithVersion](#)

Use `setPayloadWithVersion` method to add a payload to a message. The method implicitly extracts API version from the protobuf and updates the message header. Do not use `SoaMessage::setServiceVersion()` as soon it will override the version.

1.2. [SoaMessage::parsePayloadWithVersion](#)

Use `parsePayloadWithVersion` to deserialize payload.

The method returns:

`SoaErrorCode::NO_ERROR`. The payload was successfully deserialized and version delivered in the message header is compatible with the protobuf version.

`SoaErrorCode::ERROR_SOA_MESSAGE_PARSING_FAILED`. Version delivered in the message is compatible with the protobuf version however, the payload deserialization failed.

`SoaErrorCode::ERROR_UNSUPPORTED_VERSION`: version delivered in the message is **not** compatible with the protobuf version however, the payload deserialization failed.

`SoaErrorCode::WARNING_UNSUPPORTED_VERSION` version delivered in the message is not compatible with the protobuf version however, the payload was successfully deserialized.

1.3. [Code reference](#)

