

VSS Resources at a Glance

This page is an aggregation of links to various Vehicle Signal Specifications resources. It is intended to provide a quick reference access to everything VSS. If you do not find what you are looking for or see things that are missing, please let us know.

If you want to dive right into the spec or documentation go to [GitHub](#)



Vehicle Signal Specification

Table of Contents

- [Table of Contents](#)
- [COVESA VSS Wiki](#)
- [Calendar and Collaboration Channels \(mailing lists and Slack\)](#)
- [GitHub](#)
- [Presentations and Decks](#)
- [How and where VSS is used?](#)
 - [OEM and Supplier Adoption](#)
 - [Open Source Implementations](#)
 - [Spotlight Series](#)
- [Joint Projects with W3C](#)

COVESA VSS Wiki

The VSS page on this wiki is a good starting point that leads/points to all of the other locations.

- [VSS Overview](#) - High level overview of VSS
- [Introduction to VSS Sessions](#) - Every Quarter, starting Q1 2023, there is an Intro to VSS. Presentation and recording of the session are available.
- [Understanding VSS](#) - Presentations and decks starting with an introduction to those unfamiliar with VSS followed by some deeper dives into VSS and related VSS Ontology
- [Vehicle Data Models - Overview and Gap Analysis](#) - Survey of vehicle data models
- [Weekly Meetings](#) - See the [Community Calendar](#)
- [Weekly meeting: Agenda and Notes](#)

Calendar and Collaboration Channels (mailing lists and Slack)

- [COVESA Community Calendar](#)
- [COVESA Communication Channels](#)

GitHub

GitHub is where almost all of the VSS project work is maintained. The [main page](#) contains links to

- [Documentation](#)
 - A straight forward set of topics that give a good overview of VSS
- [Releases](#)
 - Release links with lists of what has changed
- [Specification/Catalog](#)
 - VSS specification refers to the standard catalog. It is maintained in YAML. It is fully expected that not everyone will implement all data elements/signals and they will likely implement some number of proprietary elements/signals. VSS is still a useful methodology/rule set for defining interoperable models.
- [Tools](#)
 - Tools are maintained in a separate repository
- [GitHub Project Wiki](#)
 - VSS project team uses the [VSS Wiki](#) for documenting a variety of things including releases.
- [Pull Requests](#)
 - All edits and updates are presented to the team through Pull Requests and are discussed in weekly meetings.
- [Issues](#)
 - All [issues](#) are tracked through GitHub issues

Presentations and Decks

- Introduction to Vehicle Signal Specification at COVESA AMM October 2022 ([deck](#))
- Introduction to Vehicle Signal Specification ([presentation](#)) at COVESA AMM April 2022 ([deck](#))
- VSS and VSS Ontology ([presentation](#)) at COVESA AMM Spring 2022 ([deck](#))

How and where VSS is used?

There are various implementations where VSS is used as data model for vehicle data.

OEM and Supplier Adoption

- BMW
 - Mapped proprietary descriptions of vehicle data to VSS, for millions of cars.
 - Employing GraphQL to create a backend single access layer for querying data coming from various storage and streaming systems. Demonstrated backend application based on this solution. While for data streaming use cases, other technologies that are supporting VSS might be in place.
- Volvo
 - VISS implementation, exposing VSS within the vehicle
 - [Ideas for Vehicle Signal Set Matching](#)
 - [Dynamic weight generation for vehicle signal set matching.pdf](#)
- Ford
 - [EV Charging Event Data Aggregation](#) project led by Ford is actively updating the VSS model for EV Charging
 - Internally working with VSS as a starting point for a variety of data related and development projects
- Mercedes
 - Internally working with VSS as a starting point for a variety of data related and development projects
- BlackBerry
 - Used for signal normalization in the BlackBerry IVY product.

Open Source Implementations

- [playground.digital.auto](#) - playground for ideation and collaboration between digital.auto participants using VSS and python.
- [Aos Edge](#) - A software first approach to the connected vehicle.
 - Provides a protected VSS data server that uses the VISS protocol, VISS client library and a 'getting started' example using VSS as the data model.
 - [Source repositories](#)
- [Eclipse Kuksa](#) - Building blocks for the Software Defined Vehicle
 - [Repositories and other Resources](#)
- [AWS IoT Fleetwise](#) - Fleet Management software
 - [Repositories](#)
- [W3C Vehicle Information Service Specification](#) - data service for serving VSS data on multiple topologies

Spotlight Series

COVESA [Spotlight](#) is a series where member organizations from the COVESA Community present their use/implementation of COVESA's [Vehicle Signal Specification](#) answering the following questions:

- Why VSS was used? / The value of using VSS?
- How VSS was used?
- What could be improved in VSS?

You can find the past presentations from the series [here](#). If you are interested in presenting your VSS implementation, please contact the [Community Director](#).

Joint Projects with W3C



VISS and VISSR are now maintained in COVESA [GitHub](#). This section is for historical purposes and will eventually be removed.

Covesa has joint projects to develop and standardize certain technology through the W3C process.

- Vehicle Information Service Specification (VISS)
 - [VISS Core](#)
 - [VISS Transport](#)
- Vehicle Signal Specification Ontology
 - [VSSo Core](#)
 - [VSSo](#)