

Conventions used

CAR 2 CAR Communication Consortium



CAR 2 CAR

COMMUNICATION CONSORTIUM

About the C2C-CC

Enhancing road safety and traffic efficiency by means of Cooperative Intelligent Transport Systems and Services (C-ITS) is the dedicated goal of the CAR 2 CAR Communication Consortium. The industrial driven, non-commercial association was founded in 2002 by vehicle manufacturers affiliated with the idea of cooperative road traffic based on Vehicle-to-Vehicle Communications (V2V) and supported by Vehicle-to-Infrastructure Communications (V2I). Today, the Consortium comprises 61 members, with 11 vehicle manufacturers, 31 equipment suppliers and 29 research organisations.

Over the years, the CAR 2 CAR Communication Consortium has evolved to be one of the key players in preparing the initial deployment of C-ITS in Europe and the subsequent innovation phases. CAR 2 CAR members focus on wireless V2V communication applications based on ITS-G5 and concentrate all efforts on creating standards to ensure the interoperability of cooperative systems, spanning all vehicle classes across borders and brands. As a key contributor, the CAR 2 CAR Communication Consortium works in close cooperation with the European and international standardisation organisations such as ETSI and CEN.

Disclaimer

The present document has been developed within the CAR 2 CAR Communication Consortium and might be further elaborated within the CAR 2 CAR Communication Consortium. The CAR 2 CAR Communication Consortium and its members accept no liability for any use of this document and other documents from the CAR 2 CAR Communication Consortium for implementation. CAR 2 CAR Communication Consortium documents should be obtained directly from the CAR 2 CAR Communication Consortium.

Copyright Notification: No part may be reproduced except as authorized by written permission. The copyright and the foregoing restriction extend to reproduction in all media. © 2020, CAR 2 CAR Communication Consortium.

Document information

Number:	2086	Version:	n.a.	Date:	16/12/2020
Title:	Conventions used		Document Type:	TR	
Release	1.5.2				
Release Status:	Public				
Status:	Final				

Table 1: Document information

Changes since last version

Title:	Conventions used		
Date	Changes	Edited by	Approved
16/12/2020	No changes	Release Management	Steering Committee
31/07/2020	Initial release	Release Management	Steering Committee

Table 2: Changes since last version

Table of contents

About the C2C-CC	1
Disclaimer	1
Document information	2
Changes since last version.....	3
Table of contents.....	4
List of tables	4
List of figures.....	4
1 Introduction	5
2 Scope.....	6
3 Conventions on release types	7
4 Conventions on documents	8
4.1 Life Cycle Status	8
4.2 Document types	8
4.3 File names	8
5 Conventions on document content	9
5.1 Modal verbs terminology	9
5.2 Requirement schema	9
5.3 Provisions from referenced documents	9
5.4 Requirements quality.....	10

List of tables

Table 1: Document information.....	2
Table 2: Changes since last version	3

List of figures

Figure 1: Release types and mapping to product phases	7
---	---

1 Introduction

Other (informational)

TR_ConV_147

Conventions on documents and their content are essential for a high quality of specification work as done by the CAR 2 CAR Communication Consortium.

2 Scope

Other (informational)

TR_ConV_146

This document provides the conventions used for the specification work as done by CAR 2 CAR Communication Consortium. It covers conventions on:

- Release types;
- Documents and
- Document content.

3 Conventions on release types

Other (informational)

TR_ConV_157

Release types are closely related to product phase. They support the CAR 2 CAR Communication Consortium product phases schema and allow to deliver appropriate results in each life cycle phase.

C2C-CC applies three release types:

- Major Release:
 - Newly developed from scratch or
 - Major changes compared to previous release
 - Including bug fixing
- Minor release:
 - Introduction of new features, which
 - Shall not break backward compatibility
 - If mitigation is possible by the C2C-CC partner and was agreed exceptions might be made.
 - Including bug fixing
- Revisions
 - Bug fixing
 - Which shall not break backward compatibility

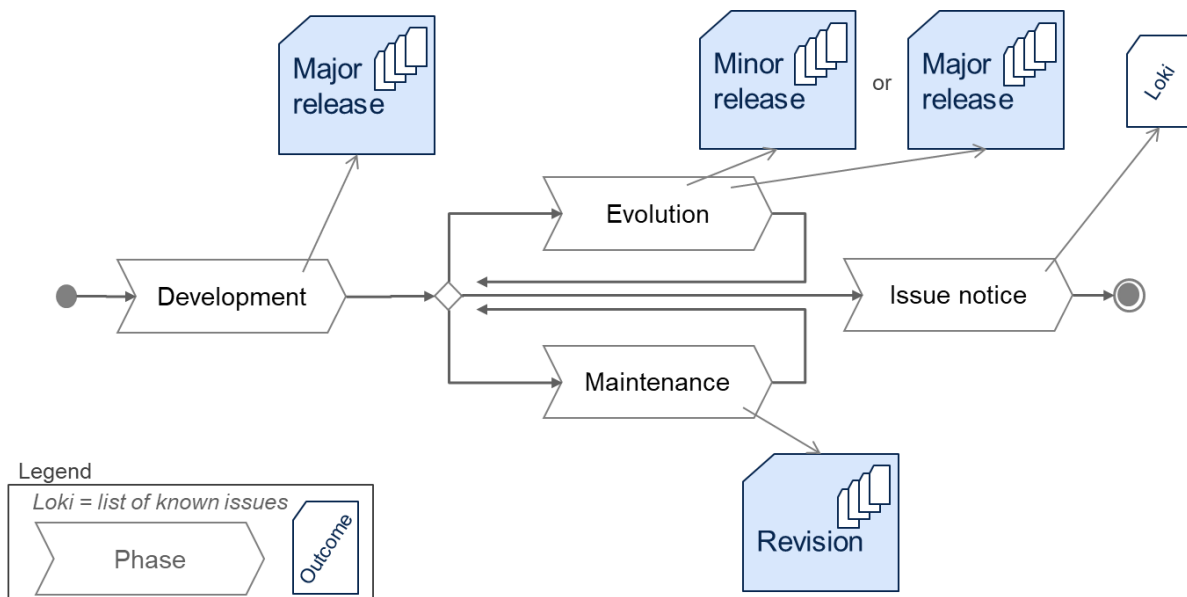


Figure 1: Release types and mapping to product phases

4 Conventions on documents

4.1 Life Cycle Status

Other (informational)

TR_ConV_160

Valid live cycles of a released document are the following:

- Final = Valid document with in a release bundle
- Concluded = No further change is planned
- Obsolete = Not maintained. Is planned to be removed (status cancelled) with next release
- Canceled = Not released anymore in active release branch

4.2 Document types

Other (informational)

TR_ConV_156

C2C-CC Deliverables shall have one of the following Types:

- DocTyp = 2/3/4 letter abbreviation of:
 - EXP = Explanatory (white paper)
 - TR = Technical Report (position paper)
 - RS = Requirement Specification (TCs and BSP, PP,)
 - TS = Test Specification
 - PP = Protection Profile

4.3 File names

Other (informational)

TR_ConV_155

File names of C2C-CC deliverables shall follow the following schema:

- C2CCC_<DocType>_<ID>_<name>.pdf
 - <ID> = 4 digits (unordered number)
 - <Name> = document name, provided by Release Management and stored in the MasterDocumentList

5 Conventions on document content

5.1 Modal verbs terminology

Other (informational)

TR_ConV_152

In the C2C-CC requirement specifications verbal forms shall be used as defined in chapter '3.2 Verbal forms for the expression of provisions' of the 'ETSI Drafting Rules (EDR), 28 September 2018',

[https://portal.etsi.org/Portals/0/TBpages/edithelp/Docs/40_directives_apr_2019_part2%20\(EDR\).pdf](https://portal.etsi.org/Portals/0/TBpages/edithelp/Docs/40_directives_apr_2019_part2%20(EDR).pdf).

5.2 Requirement schema

Other (informational)

TR_ConV_154

C2C-CC requirements shall follow the following schema:

```
[<requirement id>]
<requirement text>
<trace to other requirements>
<trace to related tests>
```

Other (informational)

TR_ConV_158

C2C-CC requirement IDs shall follow the following schema:

```
<requirement id>    =    <DocType>_<DocAbbreviation>_<number>
<DocType>           =    see above
<DocAbbreviations> =    2-6 letter abbreviation of document name
                        (managed in the MasterDocumentList)
<number>            =    up to 5 digit and unique number within a document
```

Note: Requirements in Protection Profiles have in addition a "CC reference", which stands for Common Criteria reference and shall increase the usability of the C2C-CC documents for security experts.

5.3 Provisions from referenced documents

Other (informational)

TR_ConV_153

Unless otherwise specified in the present document, the normative requirements included in the referenced documents supporting the required functionality of the vehicle C-ITS station profile shall apply. The verbal forms for the definition of provisions of referenced documents are defined either inside the document, or generally by the SDO (standardization organization) or the organization providing them. For example, normative requirements in ETSI documents are indicated by the verbal form "shall".

In case of more than one option in the standard, this document specifies which one is the recommended choice to ensure interoperability and/or sufficient performance. The C2C-CC specifications supplement the standards in case where standards are open for interpretation or believed not to contain all necessary requirements to ensure interoperability and/or sufficient performance.

The C2C-CC specifications might also supplement standards in cases where it is believed that more stringent requirements than the minimum requirements in the standard shall be applied to ensure sufficient performance.

5.4 Requirements quality

Other (informational)

TR_ConV_424

All Requirements shall have the following properties:

- **redundancy:** Requirements shall not be repeated within one requirement or in other requirements
- **clearness:** All requirements shall allow one possibility of interpretation only. Only technical terms of the glossary may be used. Furthermore, it must be clear from the requirement, what object the statement is a requirement on. Examples:
 - The <...> module shall/should/may ...
 - The <...> module's environment shall ...
 - The <...> configuration shall...
 - The function <...> shall ...
 - The hardware shall ...
- **atomicity:** Each Requirement shall only contain one requirement. A Requirement is atomic if it cannot be split up in further requirements.
- **testability:** Requirements shall be testable by analysis, review or test.
- **traceability:** The source and status of a requirement shall be visible at all times.
- **formulation:** All requirements shall be formulated so that they can be interpreted without the surrounding context (for example: "the function Xyz..." instead of "this function...").