ETSI GS NFV-SOL 004 V4.3.1 (2022-07)

Network Functions Virtualisation (NFV) Release 4;
Protocols and Data Models;
VNF Package and PNFD Archive specification

Disclaimer

The present document has been produced and approved by the Network Functions Virtualisation (NFV) ETSI Industry Specification Group (ISG) and represents the views of those members who participated in this ISG.
It does not necessarily represent the views of the entire ETSI membership.

Reference

RGS/NFV-SOL004ed431

Keywords

data, NFV, protocol, virtualisation

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00   Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Important notice

The present document can be downloaded from:
http://www.etsi.org/standards-search

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the prevailing version of an ETSI deliverable is the one made publicly available in PDF format at www.etsi.org/deliver.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at
https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

If you find errors in the present document, please send your comment to one of the following services:
https://portal.etsi.org/People/CommiteeSupportStaff.aspx

If you find a security vulnerability in the present document, please report it through our
Coordinated Vulnerability Disclosure Program:
https://www.etsi.org/standards/coordinated-vulnerability-disclosure

Notice of disclaimer & limitation of liability

The information provided in the present deliverable is directed solely to professionals who have the appropriate degree of experience to understand and interpret its content in accordance with generally accepted engineering or other professional standard and applicable regulations.
No recommendation as to products and services or vendors is made or should be implied.
No representation or warranty is made that this deliverable is technically accurate or sufficient or conforms to any law and/or governmental rule and/or regulation and further, no representation or warranty is made of merchantability or fitness for any particular purpose or against infringement of intellectual property rights.
In no event shall ETSI be held liable for loss of profits or any other incidental or consequential damages.

Any software contained in this deliverable is provided "AS IS" with no warranties, express or implied, including but not limited to, the warranties of merchantability, fitness for a particular purpose and non-infringement of intellectual property rights and ETSI shall not be held liable in any event for any damages whatsoever (including, without limitation, damages for loss of profits, business interruption, loss of information, or any other pecuniary loss) arising out of or related to the use of or inability to use the software.

Copyright Notification

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Group Specification (GS) has been produced by ETSI Industry Specification Group (ISG) Network Functions Virtualisation (NFV).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

1 Scope

The present document specifies the structure and format of a VNF package file and its constituents, fulfilling the requirements specified in ETSI GS NFV-IFA 011 [1] for a VNF package.

The present document also specifies the structure and format of a PNFD archive file and its constituents, fulfilling the requirements specified in ETSI GS NFV-IFA 014 [i.9] for a PNFD archive.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at https://docbox.etsi.org/Reference/.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are necessary for the application of the present document.

[1] ETSI GS NFV-IFA 011: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; VNF Descriptor and Packaging Specification".

[2] OASIS TOSCA-Simple-Profile-YAML-v1.1-csprd01: "TOSCA Simple Profile in YAML Version 1.1".

NOTE: Available at http://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.1/csprd01/TOSCA-Simple-Profile-YAML-v1.1-csprd01.html.

[3] IETF RFC 3339: "Date and Time on the Internet: Timestamps".

[4] IANA register for Hash Function Textual Names.

NOTE: Available at https://www.iana.org/assignments/hash-function-text-names/hash-function-text-names.xhtml.

[5] IETF RFC 5652 (September 2009): "Cryptographic Message Syntax (CMS)".

[6] IETF RFC 7468: "Textual Encodings of PKIX, PKCS, and CMS Structures".

[7] Void.

[8] Recommendation ITU-T X.509: "Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks".

[9] Void.

[10] IETF RFC 2315: "PKCS #7: Cryptographic Message Syntax Version 1.5".

[11] OASIS TOSCA-Simple-Profile-yaml-v1.3: "TOSCA Simple Profile in YAML Version 1.3".

NOTE: Available at https://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.3/TOSCA-Simple-Profile-YAML-v1.3.html.

[12] ISO/IEC 21320-1: "Information technology -- Document Container File -- Part 1: Core".

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] Void.

[i.2] Void.

[i.3] ETSI GS NFV 003: "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV".

[i.4] ETSI GS NFV-SOL 001: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on TOSCA specification".

[i.5] ETSI NFV registry of non-MANO artifact sets.

NOTE: Available at http://register.etsi.org/NFV.

[i.6] ETSI GS NFV-SOL 006: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; NFV descriptors based on YANG specification".

[i.7] ETSI GS NFV-SOL 004 (V2.4.1): "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; VNF Package specification".

[i.8] ETSI GS NFV-SOL 004 (V2.5.1): "Network Functions Virtualisation (NFV) Release 2; Protocols and Data Models; VNF Package specification".

[i.9] ETSI GS NFV-IFA 014: "Network Functions Virtualisation (NFV) Release 3; Management and Orchestration; Network Service Templates Specification".

[i.10] ETSI GS NFV-SOL 005: "Network Functions Virtualisation (NFV) Release 3; Protocols and Data Models; RESTful protocols specification for the Os-Ma-nfvo Reference Point".

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in ETSI GS NFV 003 [i.3] and the following apply:

non-MANO artifact: artifact for use by functional blocks beyond NFV-MANO

non-MANO artifact set: set of related non-MANO artifacts which are intended to be used together

3.2 Symbols

Void.

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ASCII	American Standard Code for Information Interchange
CA	Certificate Authority
CMS	Cryptographic Message Syntax
CSAR	Cloud Service ARchive
IANA	Internet Assigned Number Association
MANO	Management and Orchestration
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
PKCS	Public Key Cryptographic Standard
PNF	Physical Network Function
PNFD	PNF Descriptor
TOSCA	Topology and Orchestration Specification for Cloud Applications
URI	Universal Resource Identifier
UTF	Unicode Transformation Format
VIM	Virtual Infrastructure Manager
VNF	Virtualised Network Function
VNFD	VNF Descriptor
YAML	YAML Ain't Markup Language
YANG	Yet Another Next Generation

4 VNF package

4.1 TOSCA YAML Cloud Service Archive (CSAR) overview

4.1.1 CSAR structure

TOSCA YAML CSAR file is an archive file using the ZIP file format whose structure complies with the TOSCA Simple Profile YAML v1.1 [2] or the TOSCA Simple Profile in YAML v1.3 [11]. According to the TOSCA Simple Profile YAML v1.1 [2], the CSAR file shall have one of the two following structures:

- CSAR containing a *TOSCA-Metadata* directory, which includes the TOSCA.meta metadata file providing an entry information for processing a CSAR file.

- CSAR without a *TOSCA-Metadata* directory and containing a single yaml file with a .yml or .yaml extension at the root of the archive. The yaml file is a TOSCA definition template that shall contain a metadata section with *template_name* and *template_version* keyname.

In addition, the CSAR file may optionally contain other directories with bespoke names and contents.

4.1.2 CSAR with TOSCA-Metadata directory

4.1.2.1 General

The TOSCA.meta metadata file includes *block_0* with the *Entry-Definitions* keyword pointing to a TOSCA definitions YAML file and optionally the Other-Definitions keyword as specified in TOSCA Simple Profile YAML v1.3 [11] pointing to other TOSCA definitions YAML files used as entries for parsing the contents of the overall CSAR archive.

Any TOSCA definitions files besides the one denoted by the *Entry-Definitions* and Other-Definitions keyword can be found by processing respective *imports* statements in the entry definitions files (or in recursively imported files).

Any additional artifacts files (e.g. scripts, binaries, configuration files) can be either declared explicitly through blocks in the TOSCA.meta file or pointed to by relative path names through artifact definitions in one of the TOSCA definitions files contained in the CSAR file as described in TOSCA Simple Profile YAML v1.1 [2].

Extension of the TOSCA.meta file is described in clause 4.1.2.2.

In order to indicate that the simplified structure (i.e. not all files need to be declared explicitly) of TOSCA.meta file allowed by TOSCA Simple profile YAML 1.1 [2] is used, the *CSAR-Version* keyword listed in block_0 of the meta-file denotes the version 1.1 as described in the below example.

EXAMPLE:

```
TOSCA-Meta-File-Version: 1.0
CSAR-Version: 1.1
Created-by: Onboarding portal
Entry-Definitions: Definitions/MainServiceTemplate.yaml
```

END OF EXAMPLE.

4.1.2.2 TOSCA.meta file extension

The TOSCA.meta file structure extension is used when files defined in clauses 4.3.2 to 4.3.6 of the present document are included in the VNF package and when using CSAR with TOSCA-Metadata directory, as described in clause 4.1.2.1.

NOTE: TOSCA Simple Profile YAML v1.1 [2] does not preclude the TOSCA.meta file block _0 to be extended with key value pairs.

4.1.2.3 TOSCA.meta file keynames extension

Table 4.1.2.3-1 specifies an extension of the list of recognized TOSCA.meta file keynames as specified in the present document for the TOSCA.meta file. The keynames represents the entries for artifacts defined in clauses 4.3.2 to 4.3.6 of the present document and shall be located in the block_0.

Table 4.1.2.3-1: List of TOSCA-meta file keynames extensions

Keyname	Required	Type	Description
ETSI-Entry-Manifest	yes	string	Location of the Manifest file as defined in clause 4.3.2
ETSI-Entry-Change-Log	yes	string	Location of the Change history file as defined in clause 4.3.3
ETSI-Entry-Tests	no	string	Location of the Testing files as defined in clause 4.3.4
ETSI-Entry-Licenses	yes	string	Location of the Licensing information as defined in clause 4.3.5
ETSI-Entry-Certificate	no	string	Location of the Certificate file as defined in clause 4.3.6

Use of the Entry-Manifest, Entry-Change-Log, Entry-Tests, Entry-Licenses and Entry-Certificate keynames defined in ETSI GS NFV-SOL 004 versions 2.4.1 [i.7] to 2.5.1 [i.8] of the present document is deprecated. These keynames are only provided for backward compatibility with legacy VNF Package consumers; VNF package providers are warned that support of these keynames can be removed in subsequent versions of the present document. The key with and without the ETSI-prefix should not be both present in the TOSCA.meta. If both are present they shall point to the same value.

EXAMPLE:

```
TOSCA-Meta-File-Version: 1.0
CSAR-Version: 1.1
Created-By: MyCompany
Entry-Definitions: MRF.yaml
ETSI-Entry-Manifest: MRF.mf
ETSI-Entry-Licenses: Files/Licenses
ETSI-Entry-Change-Log: Files/ChangeLog.txt
```

END OF EXAMPLE.

4.1.3 CSAR without TOSCA-Metadata directory

4.1.3.1 General

This CSAR structure is only applicable if a YANG-based VNFD as defined in ETSI GS NFV-SOL 006 [i.6] or a TOSCA-based VNFD with single deployment flavour design as defined in clause 6.11.3 in ETSI GS NFV-SOL 001 [i.4] is included in the VNF Package. The yaml file at the root of the archive is the *CSAR Entry-Definition* file. The CSAR-Version is defined by the *template_version* metadata as can be seen in the below example. The value of *template_version* shall be set to 1.1.

EXAMPLE:

```
tosca_definitions_version: tosca_simple_yaml_1_2
metadata:
   template_name: MainServiceTemplate
   template_author: Onboarding portal
   template_version: 1.1
```

END OF EXAMPLE.

4.1.3.2 TOSCA Entry definition file metadata extension for a YANG-based VNFD

Table 4.1.3.2-1 specifies an extension of the list of recognized metadata keynames as specified in TOSCA-Simple-Profile-YAML-v1.1 [2] for the main TOSCA Service Template.

Table 4.1.3.2-1: List of metadata keynames extensions

Keyname	Required	Type	Description
yang_definitions	no	string	Reference to a YANG definition file representing the VNFD within a VNF Package

If a YANG-based VNFD is included in the VNF Package, the main TOSCA definitions YAML file shall include a metadata section with an additional metadata entry, where the keyname is "yang_definitions" and the value is the path to the YANG file representing the VNFD within the VNF Package. No additional contents shall be included in the main TOSCA definitions YAML file.

NOTE: The above requirement ensures that there cannot be both a YANG-based and a TOSCA-based representation of a VNFD in the same package.

EXAMPLE:

```
tosca_definitions_version: tosca_simple_yaml_1_1
metadata:
template_name: MainServiceTemplate
template_author: Onboarding portal
template_version: 1.1
yang_definitions: Definitions/myvnfd.xml
```

END OF EXAMPLE.

4.1.4 Void

4.2 VNF package structure and format

The structure and format of a VNF package shall conform to the TOSCA Simple Profile YAML v1.1 Specification of the CSAR format [2]. The zip file format shall conform to Document Container Format File [12].

NOTE: This implies that the VNF package can be structured according to any of the two options described in clause 4.1.

The consumer of a VNF package complying with the present document shall be able to process a CSAR file structured according to any of the two options described in clause 4.1. If the CSAR file contains a TOSCA-Metadata directory and a single yaml file with a .yml or .yaml extension at the root of the archive, the TOSCA.meta file contained in the TOSCA-Metadata directory shall be used as an entry information for processing the CSAR file.

4.3 VNF package file contents

4.3.1 General

A VNF Package shall contain a main TOSCA definitions YAML file representing all or part of the VNFD, and additional files. It shall be structured according to one of the CSAR structure options described in clause 4.1.

NOTE 1: ETSI GS NFV-SOL 001 [i.4] specifies the structure and format of the VNFD based on TOSCA specifications.

NOTE 2: ETSI GS NFV-SOL 006 [i.6] specifies the structure and format of the VNFD based on YANG specifications.

Examples of VNF package options are described in annex A.

4.3.2 VNF package manifest file

A CSAR VNF package shall have a manifest file. In the case of a CSAR VNF package with a TOSCA-Metadata directory, the location, name, and extension of the manifest file shall be specified by means of the "ETSI-Entry-Manifest" keyname in the TOSCA.meta file. In the case of a CSAR VNF package without TOSCA-Metadata directory, the manifest file shall have an extension .mf, the same name as the main TOSCA definitions YAML file and be located at the root of the archive.

The manifest file shall start with the VNF package metadata in the form of a name-value pairs. Each pair shall appear on a different line. The "name" and the "value" shall be separated by a colon and, optionally, one or more blanks. The order of the name-value pairs is not significant.

The name shall be one of those specified in table 4.3.2-1 and the values shall comply with the provisions specified in table 4.3.2-1.

The "required" column in table 4.3.2-1 specifies constraints on the presence of each name in a manifest file. If the cell in the "required" column is set to "Yes", the corresponding name shall be included. If the cell in the "required" column is set to "No", the corresponding name may, but need not to, be included. A name shall not be included more than once.

Table 4.3.2-1: List of valid names and values for VNF package metadata

Name	Value	Required
vnfd_id	A sequence of UTF-8 characters. See note 1.	Yes
vnf_provider_id	A sequence of UTF-8 characters. See note 1.	Yes
vnf_product_name	A sequence of UTF-8 characters. See note 1.	Yes
vnf_release_date_time	A string formatted according to IETF RFC 3339 [3].	Yes
vnf_software_version	A string. See note 1.	Yes
vnf_package_version	A string. See note 2.	Yes
compatible_specification_versions	Indicates which versions of the present document the VNF package complies to, as known at package creation time. See note 3. The value shall be formatted as comma-separated list of strings. Each entry shall have the format <x>.<y>.<z> where <x>, <y> and <z> are decimal numbers representing the version of the present document. Whitespace between list entries shall be trimmed before validation.	Yes See note 4.

Name	Value	Required
vnfm_info	A comma-separated list of strings as defined in the VNFD. Whitespace between list entries shall be trimmed before validation.	Yes
NOTE 1: The value shall be identical to those specified in the VNFD. NOTE 2: The value shall be identical to the descriptor_version attribute specified in the VNFD. NOTE 3: As this list is determined at the time of package creation, it should not be inferred that a package is not compatible with future versions not present in this list. Whether the package will be compatible with such future versions depends on whether these future versions are backward compatible with the listed versions. NOTE 4: A package conformant to versions prior to 2.7.1 does not include this name. Therefore, if this field is missing, it shall be assumed that the package conforms to some previous version of the present document, i.e. a version prior to 2.7.1 and the package shall be considered valid.		

An example of valid manifest file metadata entries follows.

EXAMPLE 1:

```
metadata:
vnfd_id: 2116fd24-83f2-416b-bf3c-ca1964793aca
vnf_product_name: vMRF
vnf_product_name: Virtualized PowerMRF by MyCompany Inc.
vnf_provider_id: MyCompany
vnf_software_version: 1.0.0
vnf_package_version: 1.0
vnf_release_date_time: 2017-01-01T10:00:00+03:00
vnfm_info: etsivnfm:v2.3.1,0:myGreatVnfm-1
compatible_specification_versions: 2.7.1,3.1.1
```

END OF EXAMPLE 1.

The manifest file shall include a list of all files contained in or referenced from the VNF package with their location, expressed using a Source: location/name key-value pair. The manifest file itself may be included in the list.

Below is an example of valid manifest file entries for files contained in or referenced from the VNF package when authenticity and integrity of the VNF package is implemented according to option 1 as specified in clause 5.1.

EXAMPLE 2:

```
Source: MRF.yaml

Algorithm: SHA-256
Hash: 09e5a788acb180162c51679ae4c998039fa6644505db2415e35107d1ee213943

Source: scripts/install.sh
Algorithm: SHA-256
Hash: d0e7828293355a07c2dccaaa765c80b507e60e6167067c950dc2e6b0da0dbd8b

Source: https://www.vendor_org.com/MRF/v4.1/scripts/scale/scale.sh
Algorithm: SHA-256
Hash: 36f945953929812aca2701b114b068c71bd8c95ceb3609711428c26325649165
```

END OF EXAMPLE 2.

If the VNF package is built according to option 1 (clause 5.1), the manifest file shall contain digests of all individual files contained in or referenced from the package.

A consumer of the VNF package verifies the digests in the manifest file by computing the actual digests and comparing them with the digests listed in the manifest file.

The manifest file in option 1 is the key for decision regarding a VNF package integrity and validity in terms of its contained artifacts. The specification of the manifest file and specific algorithms used in digest creation and validation is described in the security related clause.

The details of specifying the local or externally located files and their security protection are described in clause 5.