|  |
| --- |
|  |
| **Title\*:** | SOL001ed451 correct ABNF Grammar  |
|  |  |
| from **Source**\*: | Huawei Technologies Co.,Ltd |
| Contact: | lishitao@huawei.com |
|  |  |
| input for **Committee**\***:** | NFV SOL |
|  |  |
| Contribution **For\*:** | Decision | **X** |  |
|  | Discussion |  |  |
|  | Information |  |  |
|  |  |
| Submission date**\***: | 2023-7-20 |
|  |  |
| Meeting & Allocation: | **NFVSOL****#253** |
| Relevant WI(s), or deliverable(s): |   |
|  |

**Decision requested:** Please approve

**ABSTRACT: This is related to issue#8137**

<http://oldforge.etsi.org/mantis/view.php?id=8137>

THE INCORRECT RULE SPECIFIED, THAT DOES NOT MATCH THE EXAMPLE:

>>> any\_etsi\_nfv\_compliant\_product = "etsivnfm" SEP "version"

THE LITERAL "version" SPECIFIED IS INCORRECT. IN FACT, IT'S NOT A LITERAL, RATHER IT'S NON-TERMINAL <version>

 THE CORRECT RULE WOULD BE:

>>> any\_etsi\_nfv\_compliant\_product = "etsivnfm" SEP version

(i.e. version WITHOUT THE QUOTES)

|  |
| --- |
| **CHANGE REQUEST**  |
|  |
| **SOL001** | **Version**  | **4.5.1** | **CR** |  | **rev**  |  |
|  |
| ***CR Title:*** | SOL001ed451 correct ABNF Grammar  |
|  |
| ***Source:*** | Huawei Technologies Co.,Ltd |
|  |
| ***Work Item Ref:*** | SOL001ed451 | ***Date:*** | 20/07/2023 |
|  |
| ***Category:*** | F | ***Release:*** | 4 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(corresponds to a correction in an earlier release)****B*** *(addition of feature)****C*** *(functional modification of feature)****D*** *(editorial modification)* |
|  |
| ***Reason for change:*** | This is related to issue#8137<http://oldforge.etsi.org/mantis/view.php?id=8137> |
|  |
| ***Summary of change:*** | THE CORRECT RULE WOULD BE:>>> any\_etsi\_nfv\_compliant\_product = "etsivnfm" SEP version(i.e. version WITHOUT THE QUOTES) |
|  |
| ***Clauses affected:*** | 6.8.1.2 |
|  |
| ***Other deliverables*** ***affected:*** |  |
|  |
| ***Other comments:*** | Change type: BWC |

Change 1

6.8.1.2 Properties

The properties of the VNF node type shall comply with the provisions set out in table 6.8.1.2-1.

**Table 6.8.1.2-1: Properties**

| **Name** | **Required** | **Type** | **Constraints** | **Description** |
| --- | --- | --- | --- | --- |
| descriptor\_id | yes | string |  | Identifier of this VNFD information element. This attribute shall be globally unique. See note 3.The VNFD Identifier shall be used as the unique identifier of the VNF Package that contains this VNFD. Any modification of the content of the VNFD or the VNF Package shall result in a new VNFD Identifier. |
| ext\_invariant\_id | no | string |  | Identifies the VNFD in a version independent manner. This property is invariant across versions of the VNFD that fulfil certain conditions related to the external connectivity and management of the VNF. See notes 3 and 6.When used in a VNF node template in an NSD it allows for VNF instances during NS LCM the use of a VNFD different from the one referenced by the descriptor\_id property, provided they have the same ext\_invariant\_id. See note 7.  |
| descriptor\_version | yes | string |  | Identifies the version of the VNFD. |
| provider | yes | string |  | Provider of the VNF and of the VNFD. |
| product\_name | yes | string |  | Name to identify the VNF Product. Invariant for the VNF Product lifetime. |
| software\_version | yes | string |  | Software version of the VNF. This is changed when there is any change to the software that is included in the VNF Package. |
| product\_info\_name | no | string |  | Human readable name for the VNF Product. Can change during the VNF Product lifetime. |
| product\_info\_description | no | string |  | Human readable description of the VNF Product. Can change during the VNF Product lifetime. |
| vnfm\_info | yes | list of string |  | Identifies VNFM(s) compatible with the VNF described in this version of the VNFD.To indicate that a VNF can be managed by any ETSI NFV compliant VNFM, the string value shall be the concatenation of the string "etsivnfm" and the minimum version of ETSI GS NFV‑SOL 002 [22] to be supported by this VNFM (e.g. etsivnfm:v2.3.1). If the VNF is compatible with multiple versions, multiple values may be included.See note 1.To indicate a specific VNFM product, the string value shall be the concatenation of the IANA enterprise number of the VNFM provider [5], followed by a product-specific string. |
| localization\_languages | no | list of string | Valid values: string values that comply with IETF RFC 5646 [13] | Information about localization languages of the VNF (includes e.g. strings in the VNFD). This allows to provide one or more localization languages to support selecting a specific localization language at VNF instantiation time. |
| default\_localization\_language | no | string | Valid values: string values that comply with IETF RFC 5646 [13] | Default localization language that is instantiated if no information about selected localization language is available.Shall be present if "localizationLanguage" is present and shall be absent otherwise. |
| configurable\_properties | no | tosca.datatypes.nfv.VnfConfigurableProperties |  | Describes the configurable properties of the VNF (e.g. related to auto scaling and auto healing). |
| modifiable\_attributes | no | tosca.datatypes.nfv.VnfInfoModifiableAttributes |  | Describes the modifiable attributes of the VNF. |
| lcm\_operations\_configuration | no | tosca.datatypes.nfv.VnfLcmOperationsConfiguration |  | Describes the configuration parameters for the VNF LCM operations. |
| monitoring\_parameters | no | map of tosca.datatypes.nfv.VnfMonitoringParameter |  | Describes monitoring parameters applicable to the VNF.See note 4 and note 5. |
| flavour\_id | yes | string |  | Identifier of this DF within the VNFD. |
| flavour\_description | yes | string |  | Human readable description of the DF. |
| vnf\_profile | no | tosca.datatypes.nfv.VnfProfile |  | Describes a profile for instantiating VNFs of a particular NS DF according to a specific VNFD and VNF DF.See note 2. |
| NOTE 1: When LCM scripts are used, the support of this minimum version might not be sufficient to ensure that the VNF can be managed by a VNFM. The support of the domain specific language(s) used by these LCM scripts is another criterion for determining the compatibility of the VNF with a VNFM.NOTE 2: This property is only used in an NSD service template when describing a VNF node template with the corresponding VnfProfile information.NOTE 3: The value of the descriptor\_id string shall comply with an UUID format as specified in section 3 of [9].NOTE 4: This property is only used in a VNFD service template when describing a VNF node template with the corresponding monitoring information.NOTE 5: This property shall not be present in a VNFD service template when all the virtualisation containers of the VNF are realized as OsContainers.NOTE 6: Different versions of a VNFD have different descriptor\_ids but can have the same ext\_invariant\_id. Different versions of the VNFD with the same ext\_invariant\_id shall have the same number and name of VNF deployment flavours, where each of them exposes:* same external connectivity, i.e. same number and name of the requirements for VirtualLinkable capability that represent external connection points
* same number and name of VNF instantiation levels

NOTE 6a: The content of each VNF instantiation level may change.* same VNF scaling aspects and same number of levels per aspect

NOTE 6b: The constituents of each scaling aspect and the deltas between levels may change.* same VNF indicators: same attribute names and possible values

NOTE 6c: This version of the present document does not support the indication of the possible values a VNF indicator can take. However, it supports Vnfindicator policies where conditions on specific values that a VNF indicator may take can be specified (see auto- scale and auto-heal policies in clause A.15.2). Conditions related to VNF indicator attributes in those policies shall be preserved unchanged across VNFDs with the same ext\_invariant\_id, since those values may be used for designing NsAutoScale policies in the NSD. This condition implies that VNFDs with the same ext\_invariant\_id preserve external invariancy. Therefore, fulfilling this condition allows to use a different version of a VNFD in an NS instance without modification of the NSD on which the NS instance is based. The use of a different version is ultimately under the control of the service provider and it should consider if the NSD fulfils the requirements of the VnfExtCps (e.g. bitrate, IP version, etc.).NOTE 7: When the VNF node is used in an NSD, this property may only be included if the VNF node type definition in the VNFD includes the property with a value. If the property is supported in the VNF node in the VNFD, it should be included in the VNF node in the NSD to avoid changes in the NSD caused by version changes in the VNFD. |

The syntax of the vnfm\_info string values shall comply with the following ABNF [6] snippet:

value = any\_etsi\_nfv\_compliant\_product| product\_specific

any\_etsi\_nfv\_compliant\_product = "etsivnfm" SEP version

version = "v" version\_identifier

version\_identifier = 1\*2DIGIT DOT 1\*2DIGIT DOT 1\*2DIGIT

; the version identifier is encoded as a sequence of items of 1 or 2 digits separated by dots representing the 3 fields (major, technical and editorial) of the version of an ETSI deliverable.

product\_specific = enterprise\_number SEP product\_specific\_string

enterprise\_number = 1\*DIGIT

product\_specific\_string = \*(ALPHA / DIGIT / "-" / ".")

SEP = ":"

DOT = "."

This implies that vnfm\_info string values shall also comply with the pattern defined by the following regular expression [15]: (^etsivnfm:v[0-9]?[0-9]\.[0-9]?[0-9]\.[0-9]?[0-9]$)|(^[0-9]+:[a-zA-Z0-9.-]+$).