**3GPP TSG-CT WG4 Meeting #101eC4-205249**

**E-Meeting, 03rd – 13th November 2020**

|  |
| --- |
| *CR-Form-v12.0* |
| **CHANGE REQUEST** |
|  |
|  | **29.501** | **CR** | **0094** | **rev** | **-** | **Current version:** | **16.5.0** |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network |  | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | API Versioning System based on Semantic Versioning |
|  |  |
| ***Source to WG:*** | Ericsson, Orange, Nokia, Nokia Shanghai Bell, Huawei |
| ***Source to TSG:*** | CT4 |
|  |  |
| ***Work item code:*** | SBIProtoc17 |  | ***Date:*** | 2020-09-23 |
|  |  |  |  |  |
| ***Category:*** | **F** |  | ***Release:*** | Rel-17 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)Rel-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | The current API versioning system is not compliant with Semantic Versioning Specification (https://semver.org), even though TS 29.501 claims to be following it.The non-compliance comes from the format used for API versions under-development ("alpha" versions) and, in general, any content added after the main 3 version fields "x.y.z" (e.g. operation-specific version info).According to SemVer, the main 3 version fields may be followed by a so-called "pre-release" version, that must be separated form the main 3 version fields by a hyphen "-" character.After that, any number of dot-separated labels containing alphanumeric character or hyphens, may be added.In summary, the alpha versions should be, for example: "1.0.0-alpha.1", instead of "1.0.0.alpha-1" as TS 29.501 currently requires.This non-compliance with SemVer has an effect on the usage of off-the-shelf software used to manage versions, where typical operations:- determine if API "version X" is *newer* than API "version Y", or- determine if API "version X" is *compatible* with API "version Y"cannot be done if the API versions deviate from standard SemVer specification.A consequence of the above is that the "operator-specific version info" should not use the same separator as the "pre-release version info", since this pre-release version info has an effect on the precedence rules between API versions; e.g. "1.2.0-alpha.1" < "1.2.0" with "<" meaning "older".It is proposed to use the "+" char as separator for operator-specific version info (defined by SemVer as "build metadata"), which does not have any effect on precedence rules. |
|  |  |
| ***Summary of change:*** | Define the correct syntax for API versions, following the rules defined by Semantic Versioning Specification.Define the hyphen "-" char as separator for the pre-release version info, and the plus "+" char as separator for the operator-specific version info (defined in SemVer as "build metadata"). |
|  |  |
| ***Consequences if not approved:*** | 3GPP API versioning system is not compliant with industry-standard Semantic Versioning. |
|  |  |
| ***Clauses affected:*** | 4.3.1.1, 4.3.1.2 |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** |  |

\* \* \* First Change \* \* \* \*

### 4.3.1 Structure of API version numbers

#### 4.3.1.1 API version number format

API version numbers shall consist of at least 3 fields, following a MAJOR.MINOR.PATCH pattern according to the Semantic Versioning Specification [17].

The 1st field (MAJOR), the 2nd field (MINOR), and the 3rd field (PATCH) shall contain unsigned integer numbers, and they shall not contain leading zeroes.

An additional field (called "pre-release version" in Semantic Versioning Specification [17]) is added to denote an OpenAPI version under development, i.e. prior to the freeze of the corresponding OpenAPI description for a given 3GPP Release. This additional field is appended after the 3 first version fields using the hyphen "-" character and shall have the format "alpha.*n*", where "*n*" is an unsigned integer number without leading zeroes.

After the freeze of a 3GPP Release, additional fields (called "build metadata" in Semantic Versioning Specification [17]), containing operator-specific version information, may be appended after the 3 first version fields using the plus sign "+" character and they shall consist of a list of dot-separated identifiers, where each identifier may contain only alphanumeric characters and hyphens ([0-9A-Za-z-]).

NOTE: Operator-specific version information are ignored when determining version precedence. Thus, two versions that differ only in the operator-specific version information, have the same precedence.

EXAMPLES:

"1.0.0-alpha.1"

"3.0.1+orange.2020-09"

#### 4.3.1.2 Rules for incrementing field values

The first version of a new API under development shall obtain the version number "1.0.0-alpha.1". At the first publication of the 3GPP Technical Specification defining the API after the OpenAPI freeze of the first 3GPP Release that contains the API, the version number of the API shall be set to "1.0.0".

When a new version of the 3GPP TS containing OpenAPI file(s) is published, the fields of the corresponding API version number(s) shall be incremented according to the following rules:

**1st Field (MAJOR):**

- This numerical field shall be incremented when:

a) there are one or more backward incompatible changes to the API after the OpenAPI freeze for a given 3GPP Release; and

b) there are the first backward incompatible change(s) to the existing API with respect to the latest version in the previous 3GPP Release while a 3GPP Release is under development (i.e. prior to the OpenAPI freeze for a given 3GPP Release).

EXAMPLE 1: Assuming that 3GPP Rel-16 under development contains API version "1.1.0-alpha.2", and a backward incompatible change with respect to the latest version in the previous 3GPP Release is applied to that API before the OpenAPI freeze, the new Rel-16 API version is "2.0.0-alpha.1".

NOTE 1: Subsequent changes in a given 3GPP Release under development do not lead to increment of the 1st field (MAJOR) and 2nd field (MINOR).

NOTE 2: Rules for determining backward incompatible changes are provided in Annex B.

NOTE 3: It is recommended to avoid backward incompatible change to the API after the OpenAPI freeze whenever possible, especially after OpenAPI freeze of a succeeding Release. It is preferable to introduce such changes only in the 3GPP Release under development.

- If a backward incompatible change needs to be applied to several 3GPP Releases the following applies:

a) If the 3GPP Releases contain different MAJOR versions of the same API, a new MAJOR API version shall be assigned to each 3GPP Release in the order of those 3GPP Releases in such a manner that the lowest of those 3GPP Releases shall obtain the first unassigned MAJOR version value.

EXAMPLE 2: Assuming that 3GPP Rel-15 contains API version "1.0.0", and Rel-16 contains API version "2.0.0", and that the same backward incompatible change is applied to that API in both Releases, the new Rel-15 API version is "3.0.0" and the new Rel-16 API version is "4.0.0".

b) If the 3GPP Releases contain the same MAJOR version but different MINOR versions of the same API, a single new MAJOR API version value shall be assigned for all those 3GPP Releases, unless other backward incompatible changes only applied to some of those Releases require the creation of separate MAJOR versions.

NOTE 4: For each such Release a new MINOR version is assigned.

EXAMPLE 3: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0", and Rel-17 contains API version "1.2.0", and that the same backward incompatible change is applied to that API in all 3GPP Releases, the new 3GPP Rel-15 and Rel-16 API version is "2.0.0" and the new 3GPP Rel-17 API version is "2.2.0".

c) If the 3GPP Releases contain the same API versions, a single new API version shall be assigned for all those 3GPP Releases, unless other changes only applied to some of those Releases require the creation of separate versions.

EXAMPLE 4: Assuming that 3GPP Rel-15 and 3GPP Rel-16 contain API version "1.0.0", and that only the same backward incompatible change is applied to that API in both 3GPP Releases, the new 3GPP Rel-15 and Rel-16 API version is "2.0.0".

EXAMPLE 5: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0", and that the same backward incompatible change is applied to that API in both Releases and an additional backward compatible change is applied in 3GPP Rel-16, the new 3GPP Rel-15 API version is "2.0.0", and the 3GPP Rel-16 API version is "2.1.0".

EXAMPLE 6: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0", and that the same backward incompatible change is applied to that API in both Releases and an additional backward incompatible change is applied in 3GPP Rel-16, the new 3GPP Rel-15 API version is "2.0.0", and the 3GPP Rel-16 API version is "3.0.0".

**2nd Field (MINOR):**

- This numerical field shall be incremented when:

a) there are the first one or more backward compatible changes not corresponding to changes to earlier 3GPP Releases (i.e. changes introduced by 3GPP CR with other categories than "mirror") to the same API in a given 3GPP Release without any prior backward incompatible changes in that Release. If the same 1st field (MAJOR) and the 2nd field (MINOR) are assigned to *n* previous 3GPP Releases, a MINOR version number shall be reserved for each intermediate 3GPP Release for possible subsequent changes in that Release and the MINOR version number shall be incremented by *n*; and

EXAMPLE 7: Assuming that 3GPP Rel-15 and Rel-16 contain API version "1.0.0" (because there were no changes to the API in Rel-16), and in Rel-17 the first backward compatible new feature is added before the OpenAPI freeze, the API version "1.2.0-alpha.1" is assigned to Rel-17.

b) there are one or more subsequent backward compatible additions of features not corresponding to changes to previous 3GPP Releases to the API in a frozen 3GPP Release before a higher MINOR number has been allocated for the same MAJOR version (for a subsequent Release).

- This field shall be reset to "0" if the 1st field (MAJOR) is changed, unless a backward incompatible change needs to be applied to several 3GPP Releases that already contain the same MAJOR but different MINOR API versions. In that case a single new major API version is assigned, and for each such 3GPP Release with an own MINOR version, a new MINOR version shall be assigned, starting with MINOR version "0" for the lowest such Release, and reserving a MINOR version number for each intermediate Release without an own MINOR version. (see Example 3)

NOTE 5: In most cases the MINOR version is incremented when new backward compatible features are added in a 3GPP Release. In rare cases, where only backward compatible changes not corresponding to changes to previous 3GPP Releases are applied to a 3GPP Release, the MINOR version is also incremented. It is recommended to avoid such changes in 3GPP Releases without added functionality whenever possible.

NOTE 6: Subsequent backward compatible changes in a given 3GPP Release before OpenAPI freeze do not lead to an increment of the 2nd field (MINOR).

NOTE 7: Changes corresponding to changes in previous 3GPP Releases do not lead to an increment of the 2nd field (MINOR).

NOTE 8: If two 3GPP Releases are under parallel development (because the work on Rel-*X+1* has commenced before the OpenAPI freeze of Rel-*X*), the corresponding APIs will obtain distinct values of the 1st field (MAJOR) or 2nd field (MINOR).

EXAMPLE 8: Assuming that an API was introduced with version "1.0.0" in Rel-15, and that the Rel-16 version is "1.1.0-alpha.5" because the OpenAPI is not yet frozen in Rel-16, and that a new backward compatible Rel-17 feature is added, the Rel-17 API version is "1.2.0-alpha.1".

**3rd Field (PATCH):**

- This numerical field shall be incremented:

a) if the changes are only one or more backward-compatible corrections (but no changes requiring an update of the 1st field (MAJOR) or of the 2nd field (MINOR) are made to the API after the OpenAPI freeze of a 3GPP Release; and

b) if one or more backward compatible additions of features, but no changes requiring an update of the 1st field (MAJOR) or of the 2nd field (MINOR), are made to the API after the OpenAPI freeze of a 3GPP Release and after the assignment of a MINOR version to a higher 3GPP Release.

- This field shall be reset to "0" if the 1st field (MAJOR) or 2nd field (MINOR) is changed.

NOTE 9: Before the OpenAPI freeze for a given 3GPP Release, the 3rd field will not be incremented.

NOTE 10: If the 1st field (MAJOR) and 2nd field (MINOR) were not incremented between 3GPP Releases (because there were no added features and no backward incompatible changes), and the same backward compatible changes are then applied to those 3GPP Releases, the API files in those 3GPP Releases are identical and will obtain the same API version number.

NOTE 11: In rare cases for which a new backward compatible functionality needs to be added in an older 3GPP Release after the OpenAPI freeze and work on that API already started in a later Release, the new functionality is exceptionally introduced as a PATCH correction and a new supported feature could be defined accordingly.

**Possible additional fields**:

- Pre-Release version field

- Before the OpenAPI freeze of a 3GPP Release, an additional field (separated from the 3 first version fields by a hyphen "-" character) shall be supplied as follows:

a) When the 1st or 2nd field is incremented before the OpenAPI freeze of a 3GPP Release, this field shall obtain the value "alpha.1".

b) The numerical value "*n*" within the field value "alpha.*n*" shall be incremented if one or more subsequent changes are made to the API under development.

- After the OpenAPI freeze of a 3GPP Release, this additional field (including the hyphen "-" character) containing pre-release version info is removed from the API version.

- Operator-specific version field(s)

- After the OpenAPI freeze of a 3GPP Release, additional fields may be supplied based on operator policy after the 3 first version fields and separated by a plus sign "+" character. The rules for setting, or incrementing, such fields are out of the scope of 3GPP, but they shall comply with the rules described in the Semantic Versioning Specification [17], and contain a list of dot-separated labels within the allowed character set ([0-9A-Za-z-]).

If no change is applied to an API in a new published TS version, the API version number shall not be incremented unless the additional field ("-alpha.*n*") needs to be removed at OpenAPI freeze. This also applies if the TS is published in a new 3GPP Release.

NOTE 12: OpenAPI files can contain references to other OpenAPI files. Changes to referenced parts of such other OpenAPI files need to be considered when determining if and how to update an API version.

NOTE 13: The API version number of those version fields managed by 3GPP is incremented using 3GPP change requests.

\* \* \* End of Changes \* \* \* \*