**3GPP TSG-RAN WG2 Meeting #119-e *R2-22xxxxx***

**Electronic, Aug 17 – 29, 2022**

**Agenda item:** 6.11.2.4

**Source:** Swift Navigation (Rapporteur)

**Title:** [AT119-e][416][POS] Rel-17 positioning integrity (Swift)

**Document for:**  Discussion, Agreement

# 1. Introduction

 **[AT119-e][416][POS] Rel-17 positioning integrity (Swift)**

      Scope: Evaluate the proposals in the following tdocs:

       R2-2207736

       R2-2208395

      Intended outcome: Agreed CRs for merge into LPP rapporteur CR; report in R2-2208793

      Deadline: Tuesday 2022-08-23 1200 UTC

Two CRs for LPP have been submitted on the topic of GNSS positioning Integrity (AI 6.11.2.4):

|  |  |  |
| --- | --- | --- |
| [R2-2207736](https://www.3gpp.org/ftp/TSG_RAN/WG2_RL2/TSGR2_119-e/Docs/R2-2207736.zip) | Corrections on the integrity of A-GNSS in TS 37.355 | CATT |
| [R2-2208395](https://www.3gpp.org/ftp/TSG_RAN/WG2_RL2/TSGR2_119-e/Docs/R2-2208395.zip) | Correction on the GNSS Orbit and Clock Integrity Bounds in TS 37.355 | Swift Navigation, ESA, Ericsson |

The purpose of this email discussion is to check the two CRs for acceptability and backward compatibility.

The deadline for comment is **Friday 19-Aug-2022 23:59 UTC** and the goal is to agree the CRs for final endorsement in the comebacks.

|  |  |
| --- | --- |
| Company | Contact: Name (E-mail) |
| Nokia | Mani Thyagarajan (Mani.Thyagarajan@nokia.com) |
| Swift Navigation | Grant Hausler (grant@swiftnav.com) |
| CATT | Jianxiang Li (lijianxiang@catt.cn) |

# 2. GNSS Integrity CRs

CATT proposes the following change which is non-backward compatible:

* The IE *horizontalProtectionLevel-r17* in *IntegrityInfo-r17* is changed as optional IE following the IE *horizontalAccuracy* in *QoS* which is optional*.*

Appendix A contains the proposed changes to LPP, further details are available in the full CR R2-2207736,

**Question 1: Do you agree with the text proposal in Appendix A (CR R2-2207736)?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes / No** | **Comments** |
| Nokia | No | LPP specification does not mention any dependencies between the LMF requested QoS for the position estimate and any of the integrity information reported by the UE to LMF. So, the horizontalAccuracy being optional is not a reason to make the horizontalProtectionLevel optional. Note also that the horizontal accuracy in QoS has an extension parameter with extended values and only one of the two must be included in QoS, At least for that reason the optionality of horizontalAccuracy IE is understandable. |
| Swift Navigation | Yes | Ok to set the *horizontalProtectionLevel-r17* as Optional |
| Qualcomm | No | A UE which can determine a VPL should also be able to determine the HPL. The LMF is free to ignore not needed values. The change is highly NBC. |
| CATT | Yes  (proponent) | When *horizontalAccuracy* which is optional in QoS isn’t provided to UE, the *horizontalProtectionLevel* reported to LMF doesn't make sense. This is a early release, NBC is still acceptable. |
| Huawei, Hisilicon | No | The whole IE *IntegrityInfo-r17* is optional. When it is present, at least “*horizontalProtectionLevel*” should be given, otherwise there is no need to provide the IE *IntegrityInfo-r17.* |
| ZTE | No | Agree with Nokia, HW and QC |
| vivo | No | Agree with Nokia that there is no strong relationship between the QoS and integrity. That is, the LCS client can request integrity only for one positioning session.  Besides, agree with QC that the LMF may ignore the HPL if the LMFonly wants the VPL. NBC change is not preferred for this corner case as the location estimate may include the horizontal estimate for most all cases. |
| Intel | No | Agree with Nokia, HW and QC. |
| Xiaomi | No |  |
| OPPO | No |  |
|  |  |  |

Swift Navigation, ESA and Ericsson propose the following changes which are non-backward compatible:

* The varOrbitError and varOrbitRateError fields are renamed and redefined as stdDevOrbitError and stdDevOrbitRateError, respectively.
* The range and resolution for meanOrbitError and stdDevOrbitError (under GNSS-SSR-OrbitCorrections) and meanClock, stdDevClock (under GNSS-SSR-ClockCorrections) have been updated to align with the meanIonosphere and stdDevIonosphere values (under GNSS-SSR-STEC-Correction).

Appendix A contains the proposed changes to LPP, further details are available in the full CR R2-2208395.

**Question 2: Do you agree with the text proposal in Appendix B (CR R2-2208395)?**

|  |  |  |
| --- | --- | --- |
| **Company** | **Yes / No** | **Comments** |
| Nokia | Yes | Please delete “No impact to ASN.1” under “Inter-operability” in the Summary of change field on the CR cover. |
| Swift Navigation | Yes |  |
| Qualcomm | Yes |  |
| CATT | Yes |  |
| Huawei, Hisilicon | Yes | “Standard Deviation” in the CR text should be “standard deviation” (with lower case in the first letter). |
| ZTE | Yes |  |
| vivo | Yes |  |
| Intel | Yes | The stage 2 also needs to be updated, i.e. . change “Variance Orbit Error  ” and “Variance Orbit Rate Error” to “'Standard Deviation Orbit Error” and “'Standard Deviation Orbit Rate Error”. I will capture the changes in offline discussion [408]. |
| Xiaomi | Yes |  |
| OPPO | Yes |  |
|  |  |  |

# 3. Conclusions and Proposals

The following proposals are made:

* TBD

# Appendix A – Proposed CR from CATT (R2-2207736)

#### – *CommonIEsProvideLocationInformation*

The *CommonIEsProvideLocationInformation* carries common IEs for a Provide Location Information LPP message Type.

-- ASN1START

CommonIEsProvideLocationInformation ::= SEQUENCE {

locationEstimate LocationCoordinates OPTIONAL,

velocityEstimate Velocity OPTIONAL,

locationError LocationError OPTIONAL,

...,

[[ earlyFixReport-r12 EarlyFixReport-r12 OPTIONAL

]],

[[ locationSource-r13 LocationSource-r13 OPTIONAL,

locationTimestamp-r13 UTCTime OPTIONAL

]],

[[

segmentationInfo-r14 SegmentationInfo-r14 OPTIONAL -- Cond Segmentation

]],

[[

integrityInfo-r17 IntegrityInfo-r17 OPTIONAL

]]

}

LocationCoordinates ::= CHOICE {

ellipsoidPoint Ellipsoid-Point,

ellipsoidPointWithUncertaintyCircle Ellipsoid-PointWithUncertaintyCircle,

ellipsoidPointWithUncertaintyEllipse EllipsoidPointWithUncertaintyEllipse,

polygon Polygon,

ellipsoidPointWithAltitude EllipsoidPointWithAltitude,

ellipsoidPointWithAltitudeAndUncertaintyEllipsoid

EllipsoidPointWithAltitudeAndUncertaintyEllipsoid,

ellipsoidArc EllipsoidArc,

...,

highAccuracyEllipsoidPointWithUncertaintyEllipse-v1510

HighAccuracyEllipsoidPointWithUncertaintyEllipse-r15,

highAccuracyEllipsoidPointWithAltitudeAndUncertaintyEllipsoid-v1510

HighAccuracyEllipsoidPointWithAltitudeAndUncertaintyEllipsoid-r15,

ha-EllipsoidPointWithScalableUncertaintyEllipse-v1680 HA-EllipsoidPointWithScalableUncertaintyEllipse-r16,

ha-EllipsoidPointWithAltitudeAndScalableUncertaintyEllipsoid-v1680

HA-EllipsoidPointWithAltitudeAndScalableUncertaintyEllipsoid-r16

}

Velocity ::= CHOICE {

horizontalVelocity HorizontalVelocity,

horizontalWithVerticalVelocity HorizontalWithVerticalVelocity,

horizontalVelocityWithUncertainty HorizontalVelocityWithUncertainty,

horizontalWithVerticalVelocityAndUncertainty

HorizontalWithVerticalVelocityAndUncertainty,

...

}

LocationError ::= SEQUENCE {

locationfailurecause LocationFailureCause,

...

}

LocationFailureCause ::= ENUMERATED {

undefined,

requestedMethodNotSupported,

positionMethodFailure,

periodicLocationMeasurementsNotAvailable,

...

}

EarlyFixReport-r12 ::= ENUMERATED {

noMoreMessages,

moreMessagesOnTheWay

}

LocationSource-r13 ::= BIT STRING { a-gnss (0),

wlan (1),

bt (2),

tbs (3),

sensor (4),

ha-gnss-v1510 (5),

motion-sensor-v1550 (6),

dl-tdoa-r16 (7),

dl-aod-r16 (8) } (SIZE(1..16))

IntegrityInfo-r17 ::= SEQUENCE {

horizontalProtectionLevel-r17 INTEGER (0..50000) OPTIONAL,

verticalProtectionLevel-r17 INTEGER (0..50000) OPTIONAL,

achievableTargetIntegrityRisk-r17 INTEGER (10..90) OPTIONAL,

...

}

-- ASN1STOP

| Conditional presence | Explanation |
| --- | --- |
| *Segmentation* | This field is optionally present, need OP, if *lpp-message-segmentation-req* has been received from the location server with bit 1 (*targetToServer*) set to value 1. The field shall be omitted if *lpp‑message‑segmentation-req* has not been received in this location session, or has been received with bit 1 (*targetToServer*) set to value 0. |

# Appendix B – Proposed CR from Swift Navigation, ESA, Ericsson (R2-2207736)

#### 6.5.2.2 GNSS Assistance Data Elements

**/\*\*Skip unrelated parts\*\*/**

#### *– GNSS-SSR-OrbitCorrections*

The IE *GNSS-SSR-OrbitCorrections* is used by the location server to provide radial, along-track and cross-track orbit corrections together with integrity information. The target device may use the *SSR-OrbitCorrectionList* to compute a satellite position correction to be combined with the satellite position calculated from broadcast ephemeris.

The parameters provided in IE *GNSS-SSR-OrbitCorrections –* except for *ORBIT-IntegrityParameters* and *SSR-IntegrityOrbitBounds –* are used as specified for SSR Clock Messages (e.g., message type 1057 and 1063) in [30] and apply to all GNSSs.

-- ASN1START

GNSS-SSR-OrbitCorrections-r15 ::= SEQUENCE {

epochTime-r15 GNSS-SystemTime,

ssrUpdateInterval-r15 INTEGER (0..15),

satelliteReferenceDatum-r15 ENUMERATED { itrf, regional, ... },

iod-ssr-r15 INTEGER (0..15),

ssr-OrbitCorrectionList-r15 SSR-OrbitCorrectionList-r15,

...,

[[

orbit-IntegrityParameters-r17 ORBIT-IntegrityParameters-r17 OPTIONAL -- Need ON

]]

}

SSR-OrbitCorrectionList-r15 ::= SEQUENCE (SIZE(1..64)) OF SSR-OrbitCorrectionSatelliteElement-r15

SSR-OrbitCorrectionSatelliteElement-r15 ::= SEQUENCE {

svID-r15 SV-ID,

iod-r15 BIT STRING (SIZE(11)),

delta-radial-r15 INTEGER (-2097152..2097151),

delta-AlongTrack-r15 INTEGER (-524288..524287),

delta-CrossTrack-r15 INTEGER (-524288..524287),

dot-delta-radial-r15 INTEGER (-1048576..1048575) OPTIONAL, -- Need ON

dot-delta-AlongTrack-r15 INTEGER (-262144..262143) OPTIONAL, -- Need ON

dot-delta-CrossTrack-r15 INTEGER (-262144..262143) OPTIONAL, -- Need ON

...,

[[

ssr-IntegrityOrbitBounds-r17 SSR-IntegrityOrbitBounds-r17 OPTIONAL -- Cond Integrity1

]]

}

ORBIT-IntegrityParameters-r17 ::= SEQUENCE {

probOnsetConstFault-r17 INTEGER (0..255),

meanConstFaultDuration-r17 INTEGER (1..3600),

probOnsetSatFault-r17 INTEGER (0..255),

meanSatFaultDuration-r17 INTEGER (1..3600),

orbitRangeErrorCorrelationTime-r17 INTEGER (0..255) OPTIONAL, -- Need ON

orbitRangeRateErrorCorrelationTime-r17 INTEGER (0..255) OPTIONAL, -- Cond Integrity2

...

}

SSR-IntegrityOrbitBounds-r17 ::= SEQUENCE {

meanOrbitError-r17 RAC-OrbitalErrorComponents-r17,

stdDevOrbitError-r17 RAC-OrbitalErrorComponents-r17,

meanOrbitRateError-r17 RAC-OrbitalErrorComponents-r17,

stdDevOrbitRateError-r17 RAC-OrbitalErrorComponents-r17,

...

}

RAC-OrbitalErrorComponents-r17 ::= SEQUENCE {

radial-r17 INTEGER (0..255),

alongTrack-r17 INTEGER (0..255),

crossTrack-r17 INTEGER (0..255)

}

-- ASN1STOP

| Conditional presence | Explanation |
| --- | --- |
| *Integrity1* | The field is mandatory present if *ORBIT-IntegrityParameters* is present*;* otherwise it is not present. |
| *Integrity2* | The field is mandatory present if *orbitRangeErrorCorrelationTime* is present*;* otherwise it is not present. |

| *GNSS-SSR-OrbitCorrections* field descriptions |
| --- |
| ***epochTime***  This field specifies the epoch time of the orbit corrections. The *gnss-TimeID* in *GNSS-SystemTime* shall be the same as the *GNSS-ID* in IE *GNSS-GenericAssistDataElement*. |
| ***ssrUpdateInterval***  This field specifies the SSR Update Interval. The SSR Update Intervals for all SSR parameters start at time 00:00:00 of the GPS time scale. A change of the SSR Update Interval during the transmission of SSR data should ensure consistent data for a target device. See table Value of *ssrUpdateInterval* to SSR Update Interval relation below. NOTE 1. |
| ***satelliteReferenceDatum***  This field specifies the satellite refence datum for the orbit corrections. |
| ***iod-ssr***  This field specifies the Issue of Data number for the SSR data. A change of *iod-ssr* is used to indicate a change in the SSR generating configuration. |
| ***svID***  This field specifies the satellite for which the orbit corrections are provided. |
| ***iod***  This field specifies the IOD value of the broadcast ephemeris for which the orbit corrections are valid (see IE *GNSS‑NavigationModel*). NOTE 2. |
| ***delta-radial***  This field specifies the radial orbit correction for broadcast ephemeris. NOTE 3.  Scale factor 0.1 mm; range ±209.7151 m. |
| ***delta-AlongTrack***  This field specifies the along-track orbit correction for broadcast ephemeris. NOTE 3.  Scale factor 0.4 mm; range ±209.7148 m. |
| ***delta-CrossTrack***  This field specifies the cross-track orbit correction for broadcast ephemeris. NOTE 3.  Scale factor 0.4 mm; range ±209.7148 m. |
| ***dot-delta-radial***  This field specifies the velocity of radial orbit correction for broadcast ephemeris. NOTE 3.  Scale factor 0.001 mm/s; range ±1.048575 m/s. |
| ***dot-delta-AlongTrack***  This field specifies the velocity of along-track orbit correction for broadcast ephemeris. NOTE 3.  Scale factor 0.004 mm/s; range ±1.048572 m/s. |
| ***dot-delta-CrossTrack***  This field specifies the velocity of cross-track orbit correction for broadcast ephemeris. NOTE 3.  Scale factor 0.004 mm/s; range ±1.048572 m/s. |
| ***probOnsetConstFault***  This field specifies the Probability of Onset of Constellation Fault per Time Unit where a constellation fault is at least two satellites being faulty simultaneously due to the same event.  This field specifies the onset probability that the residual range or range rate error exceeds a bound created using the minimum allowed inflation factor *Kmin*, and bounding parameters as *mean* + *Kmin* \* *stdDev* where *Kmin* = *normInv*(*irMaximum* / 2), with *irMaximum* as provided in IE *GNSS-Integrity-ServiceParameters*.  The probability is calculated by *P*=10-0.04*n* [hour-1] where *n* is the value of *probOnsetConstFault* and the range is 10-10.2 to 1 per hour. |
| ***meanConstFaultDuration***  This field specifies the Mean Constellation Fault Duration which is the mean duration between when a constellation fault occurs, and the user is alerted by the service through FFS (or the integrity violation is over).  Scale factor 1 s; range 1-3600 s. |
| ***probOnsetSatFault***  This field specifies the Probability of Onset of Satellite Fault per Time Unit which is the probability of occurrence of satellite error to exceed the residual error bound for more than the Time to Alert (TTA).  This field specifies the onset probability that the residual range or range rate error exceeds a bound created using the minimum allowed inflation factor *Kmin*, and bounding parameters as *mean* + *Kmin* \* *stdDev* where *Kmin* = *normInv*(*irMaximum* / 2), with *irMaximum* as provided in IE *GNSS-Integrity-ServiceParameters*.  The probability is calculated by *P*=10-0.04*n* [hour-1] where *n* is the value of *probOnsetSatFault* and the range is 10-10.2 to 1 per hour. |
| ***meanSatFaultDuration***  This field specifies the Mean Satellite Fault Duration which is the mean duration between when a satellite fault occurs, and the user is alerted by the service through FFS (or the integrity violation is over).  Scale factor 1 s; range 1-3,600 s. |
| ***orbitRangeErrorCorrelationTime***  This field specifies the Orbit Range Error Correlation Time which is the upper bound of the correlation time of the satellite residual range error due to orbit.  The time is calculated using:  Range is 1-28,200 s. |
| ***orbitRangeRateErrorCorrelationTime***  This field specifies the Orbit Range Rate Error Correlation Time which is the upper bound of the correlation time of the satellite residual range rate error due to orbit.  The time is calculated using:  Range is 1-28,200 s. |
| ***meanOrbitError***  This field specifies the Mean Orbit Error bound in satellite radial, along-track and cross-track coordinates, which are the mean values for a set of three overbounding models that bound the residual orbit error in satellite radial, along-track and cross-track directions.  Each mean is calculated using:  Range is 0-17.5 m. |
| ***stdDevOrbitError***  This field specifies the Standard Deviation Orbit Error bound in satellite radial, along-track and cross-track coordinates, which are the standard deviation values for a set of three overbounding models that bound the residual orbit error in satellite radial, along-track and cross-track directions.  Each standard deviation is calculated using:  Range is 0-17.5 m. |
| ***meanOrbitRateError***  This field specifies the Mean Orbit Rate Error in satellite radial, along-track and cross-track coordinates, which are the mean values for a set of three overbounding models that bound the residual satellite orbit rate error in satellite radial, along-track and cross-track directions.  Scale factor 0.001 m/s; range 0-0.255 m/s. |
| ***stdDevOrbitRateError***  This field specifies the Standard DeviationOrbit Rate Error in satellite radial, along-track and cross-track coordinates, which are the standard deviation values for a set of three overbounding models that bound the residual satellite orbit rate error in satellite radial, along-track and cross-track directions.  Scale factor 0.001 m/s; range 0-0.255 m/s. |

NOTE 1: The update intervals are aligned to the GPS time scale for all GNSSs in order to allow synchronous operation for multiple GNSS services. This means that the update intervals may not be aligned to the beginning of the day for another GNSS. Due to the leap seconds, this is generally the case for GLONASS.

NOTE 2: In the cases that *gnss-ID* indicates 'gps' or 'qzss', the *iod* refers to the NAV broadcast ephemeris (GPS L1 C/A or QZSS QZS-L1, respectively, in table GNSS to iod Bit String(11) relation in IE *GNSS‑NavigationModel).*

NOTE 3: The reference time *t0* is *epochTime* + ½ × *ssrUpdateInterval*. The reference time *t0* for *ssrUpdateInterval* '0' is *epochTime*.

Value of *ssrUpdateInterval* to SSR Update Interval relation

|  |  |
| --- | --- |
| Value of *ssrUpdateInterval* | SSR Update Interval |
| 0 | 1 second |
| 1 | 2 seconds |
| 2 | 5 seconds |
| 3 | 10 seconds |
| 4 | 15 seconds |
| 5 | 30 seconds |
| 6 | 60 seconds |
| 7 | 120 seconds |
| 8 | 240 seconds |
| 9 | 300 seconds |
| 10 | 600 seconds |
| 11 | 900 seconds |
| 12 | 1800 seconds |
| 13 | 3600 seconds |
| 14 | 7200 seconds |
| 15 | 10800 seconds |

#### *– GNSS-SSR-ClockCorrections*

The IE *GNSS-SSR-ClockCorrections* is used by the location server to provide clock correction parameters together with integrity information. The target device may use the *SSR-ClockCorrectionList* to compute a clock correction to be applied to the broadcast satellite clock parameters, identified by *iod* of corresponding *GNSS-SSR-OrbitCorrections*.

The parameters provided in IE *GNSS-SSR-ClockCorrections –* except for *CLOCK-IntegrityParameters* and *SSR-IntegrityClockBounds –* are used as specified for SSR Clock Messages (e.g., message type 1058 and 1064) in [30] and apply to all GNSSs.

-- ASN1START

GNSS-SSR-ClockCorrections-r15 ::= SEQUENCE {

epochTime-r15 GNSS-SystemTime,

ssrUpdateInterval-r15 INTEGER (0..15),

iod-ssr-r15 INTEGER (0..15),

ssr-ClockCorrectionList-r15 SSR-ClockCorrectionList-r15,

...,

[[

clock-IntegrityParameters-r17 CLOCK-IntegrityParameters-r17 OPTIONAL -- Need ON

]]

}

SSR-ClockCorrectionList-r15 ::= SEQUENCE (SIZE(1..64)) OF SSR-ClockCorrectionSatelliteElement-r15

SSR-ClockCorrectionSatelliteElement-r15 ::= SEQUENCE {

svID-r15 SV-ID,

delta-Clock-C0-r15 INTEGER (-2097152..2097151),

delta-Clock-C1-r15 INTEGER (-1048576..1048575) OPTIONAL, -- Need ON

delta-Clock-C2-r15 INTEGER (-67108864..67108863) OPTIONAL, -- Need ON

...,

[[

ssr-IntegrityClockBounds-r17 SSR-IntegrityClockBounds-r17 OPTIONAL -- Need ON

]]

}

CLOCK-IntegrityParameters-r17 ::= SEQUENCE {

clockRangeErrorCorrelationTime-r17 INTEGER (0..255),

clockRangeRateErrorCorrelationTime-r17 INTEGER (0..255),

...

}

SSR-IntegrityClockBounds-r17 ::= SEQUENCE {

meanClock-r17 INTEGER (0..255),

stdDevClock-r17 INTEGER (0..255),

meanClockRate-r17 INTEGER (0..255),

stdDevClockRate-r17 INTEGER (0..255),

...

}

-- ASN1STOP

| *GNSS-SSR-ClockCorrections* field descriptions |
| --- |
| ***epochTime***  This field specifies the epoch time of the clock corrections. The gnss-TimeID in *GNSS-SystemTime* shall be the same as the *GNSS-ID* in IE *GNSS-GenericAssistDataElement.* |
| ***ssrUpdateInterval***  This field specifies the SSR Update Interval. The SSR Update Intervals for all SSR parameters start at time 00:00:00 of the GPS time scale. A change of the SSR Update Interval during the transmission of SSR data should ensure consistent data for a target device. See table Value of *ssrUpdateInterval* to SSR Update Interval relation in IE *GNSS‑SSR‑OrbitCorrections*. |
| ***iod-ssr***  This field specifies the Issue of Data number for the SSR data. A change of iod-ssr is used to indicate a change in the SSR generating configuration. |
| ***svID***  This field specifies the satellite for which the clock corrections are provided. |
| ***delta-Clock-C0***  This field specifies the C0 polynomial coefficient for correction of broadcast satellite clock. NOTE 1.  Scale factor 0.1 mm; range ±209.7151 m. |
| ***delta-Clock-C1***  This field specifies the C1 polynomial coefficient for correction of broadcast satellite clock. NOTE 1.  Scale factor 0.001 mm/s; range ±1.048575 m/s. |
| ***delta-Clock-C2***  This field specifies the C2 polynomial coefficient for correction of broadcast satellite clock. NOTE 1.  Scale factor 0.00002 mm/s2; range ±1.34217726 m/s2. |
| ***clockRangeErrorCorrelationTime***  This field specifies the Clock Range Error Correlation Time which is the upper bound of the correlation time of the satellite residual range error due to clock.  The time is calculated using:  Range is 1-28,200 s. |
| ***clockRangeRateErrorCorrelationTime***  This field specifies the Clock Range Rate Error Correlation Time which is the upper bound of the correlation time of the satellite residual range rate error due to clock.  The time is calculated using:  Range is 1-28,200 s. |
| ***meanClock***  This field specifies the Mean Clock Error bound which is the mean value for an overbounding model that bounds the residual clock error.  The bound is *meanClock* + K \* *stdDevClock* and shall be so that the probability of it to be exceeded shall be lower than IRallocation for *irMinimum* < IRallocation < *irMaximum*, where K = normInv(IRallocation / 2) and *irMinimum*, *irMaximum* as provided in IE *GNSS-Integrity-ServiceParameters*.  This IRallocation is a fraction of the Target Integrity Risk that represents the integrity risk budget available.  The mean is calculated using:  Range is 0-17.5 m. |
| ***stdDevClock***  This field specifies the Standard Deviation Clock Error bound which is the standard deviation for an overbounding model that bounds the residual clock error.  The standard deviation is calculated using:  Range is 0-17.5 m. |
| ***meanClockRate***  This field specifies the Mean Clock Rate Error bound which is the mean value for an overbounding model that bounds the residual clock rate error.  The bound is *meanClockRate* + K \* *stdDevClockRate* and shall be so that the probability of it to be exceeded shall be lower than IRallocation for *irMinimum* < IRallocation < *irMaximum*, where K = normInv(IRallocation / 2) and *irMinimum*, *irMaximum* as provided in IE *GNSS-Integrity-ServiceParameters*.  This IRallocation is a fraction of the Target Integrity Risk that represents the integrity risk budget available.  Scale factor 0.001 m/s; range 0.000-0.255 m/s. |
| ***stdDevClockRate***  This field specifies the Standard Deviation Clock Rate Error bound which is the standard deviation for an overbounding model that bounds the residual clock rate error.  Scale factor 0.001 m/s; range 0.000-0.255 m/s. |

NOTE 1: The reference time *t0* is *epochTime* + ½ × *ssrUpdateInterval*. The reference time *t0* for *ssrUpdateInterval* '0' is *epochTime*.

**/\*\*Skip unrelated parts\*\*/**