GARMIS

gmv INNOVATING SOLUTIONS

CLIENT SOFTWARE USER MANUAL

16/03/2010

	Name (Company)	Signature	Date
Edited by	A. Rubio (GMV-SGI)		16/03/2010
Approved by:	J.A. Abánades (GMV-SGI)		16/03/2010
Authorized by:	A. Gavín (GMV AD)		16/03/2010

Document identification

Project	Company	WP	Nature	Chrono or deliverable number	Classification	Version Issue
EDAS	SGI	6160	O*	6162	U **	3.A

* C Contractual / D Deliverable / O Others
** U Unclassified / CUI Galileo Participants Use Only / CI Commercial in Confidence

Partner's reference (optional)
SGI-EDAS-CSUM-01 SGISA 20731/06 V9/10

Table of Contents

Document Contributors

Name	Company	Section
J. Ignacio Ormeño	GMV-SGI	
Ángel J. Gavín	GMV-AD	
Alberto Rubio	GMV-SGI	

Change Records

Version	Issue	Date	Description	Author (Company)
1	A	01/09/2006	First Version of this document	M. Alcón (GMV-SGI)
1	B	02/10/2006	Version for DSR-CO and ADR, including: • Final CS version installation and operation instructions • Application-layer description of the Client Software interface	J. Ignacio Ormeño (GMV-SGI)
1	C	01/08/2007	Version for ADR-CO and DDD, including: * Installation instructions for Windows	J. Ignacio Ormeño (GMV-SGI)
1	C	01/08/2007	Version for ADR-CO and DDD, including: • Clarification on EDAS RTCM encoding schema in section 5.1. (implements RID-ADR-003 and RID-ADR-005) • Clarification on messages 1006 and 1007 (implements RID-ADR-004)	J. Pereira (GMV-AD)
1	D	01/08/2007	Version for AR/SAR including: ▪ Clarification on APC messages ▪ Clarification on how to connect a Service Provider, HW prerequisites and comments.	J. Ignacio Ormeño (GMV-SGI)
2	A	24/11/2008	Document updated for EDAS v1.4 release. Changes originated by DRS in GMV-EDAS-NC-10:	J. Ignacio Ormeño (GMV-SGI)

Version	Issue	Date	Description	Author (Company)
			■ Section 2.1 updated as per item 1 & 2 in DRS (clarifications in EDAS connection requirements). ■ Section 2.1 updated as per item 3 in DRS (clarifications in Operating Systems and COTS required for EDAS CS installation) ■ Section 2.2 updated as per item 4 in DRS (typo fixed) ■ Section 2.2 updated as per item 5 & 6 in DRS (folder structure updated) ■ Section 3.2 updated as per item 7 in DRS (separation of cs.properties and log4j.properties files). ■ Section 3.2 updated as per item 8 in DRS (possibility of modification of log4j.properties file) ■ Section 3.3 updated as per item 9 in DRS (log file location configurability). ■ Section 4 removed as per item 10 in DRS. A basic test to check correct installation of the CS (and the connection) provided in a new section (3.4) instead. ■ Former section 5.1 (now 4.1) updated as per item 11 in DRS. ■ Former Table 5-1 and section 5.1 (now Table 4-1 and section 4.1, respectively) updated with remarks on ATC data availability, as per item 12 in DRS. ■ Former section 5.1 (now section 4.1) updated as per item 13 in DRS (possibility of having more than one RTCM message in single EDAS message). ■ Section 4.2.4.1 updated to warn about EDAS ATC availability (item 14 in DRS).	

<table>
<tr><th>Version</th><th>Issue</th><th>Date</th><th>Description</th><th>Author (Company)</th></tr>
<tr><td></td><td></td><td></td><td>Former section 5.2.1 moved to section 4.2. Subsections re-organised to meet with item 15 (in DRS) expectations (readability of the section).
EDAS message type 4 (APC) added to message list (item 16 in DRS).
On top of the changes above, the following modifications have been included:
Section 4.2.4.2 on RIMS APC data updated to reflect the changes in APC message format derived from GMV-EDAS-NC-10 (“Impossible to dispatch APC data in a single RTCM message”) and GMV-EDAS-NC-11 (“APC messages sent without RTCM header”) implementation. Section 4.2.6 (Data Fields) and 4.2.7 (Data Types) updated with new data types.
Footnote added to section 4.2.3 to clarify that ASN.1 “Navigation Correction Message” are in fact NLES Feedback messages.
ASN.1 message specification included in new section 5 (Annex A). A pointer to this section has been added to section 4.2.3.
Wording “Final Version” deleted from cover page of this document.</td><td></td></tr>
<tr><td>2</td><td>B</td><td>16/09/2009</td><td>Document updated for EDAS v1.5 release.
Changes originated by reported bugs:
Section 3.2 updated as per new configuration parameter (max_queue_size) in file cs.properties
Disclaimer added to Section 3.2 related to data unavailability
Added Section 6 (Annex 2), related to CRC32 Computation Method</td><td>Alberto Rubio
(GMV-SGI)</td></tr>
</table>

Version	Issue	Date	Description	Author (Company)
2	C	22/09/2009	Typo errors corrected. ■ 4.2.1: CRC sentence added.	J. Yarza (GMV-ADS)
2	D	06/10/2009	■ 3.2: New sentence added before disclaimer ■ 3.4: Explanation added for low message consumption case in this section. ■ 4.2.1: CRC sentence added. Annex D: Architecture of EDAS Client SW added.	J. Yarza (GMV-ADS)
3	A	16/03/2010	■ 4.2.3ASN.1 Frame modified ANNEX B:ASN.1 MESSAGE FIELDS added	J. Yarza (GMV-ADS)

1 INTRODUCTION

1.1 PURPOSE

This document will present the procedures for installing, configuring and using the Client Software.

1.2 REFERENCES

1.2.1 Reference Documents

Ref.	Document title	Document reference	Issue	Date
[RD.1]	Client Software User Manual	GARMIS-SGI-6160-D-6161-CUI-V2	1.A	16/03/2010
[RD.2]	EDAS Client Software Interface Control Document (EDAS CSW-ICD)	200194729P	1.A	11/08/2006
[RD.3]	EGNOS Interfaces Control Document for ATC Interface (ATC ICD)	EGN-ASPI-SYST-DRD 0112/0029	2.A	16/11/01
[RD.4]	RTCM recommended standards for differential GNSS Service Version 3.0. Developed by RTCM Special Committee NO. 104	RTCM Paper 30-2004/SC104-STD	3.0	10/02/04

Table 1-1: Reference Documents

2 INSTALLATION

2.1 INSTALLATION PREREQUISITES

The hardware requirements for Client Software are the following:

- 1 GHz 32-bit (x86) or 64-bit (x64) processor
- 1 GB of system memory
- 40 GB hard drive with at least 15 GB of available space
- A dedicated line from CS machine to EDAS, this line will have a minimum throughput or CIR equals to 600 Kbits/sec (SL0 messages) or 300 Kbits/sec (SL1 messages). A xDSL line could be used but it is not recommended for real time purposes.
- EDAS IP will have to be provided by EDAS Operator.

In order to install and use the Client Software, the following configuration items must be available:

a) For Unix/Linux systems:

- Any operating system compatible with Sun Java Development Kit 1.5.
- Java SDK 1.5.0_08-b03 or better.
- GNU Tar and Gzip utilities.

b) For Windows systems:

- Windows (XP, 2000, 2003) as operating system
- Java SDK 1.5.0_08-b03 or better
- Winzip, 7zip or similar tool

2.2 INSTALLATION PROCEDURE

To install the CS, copy the software package to desired installation directory. Once the installation package has been copied, access this directory and unzip the copied file

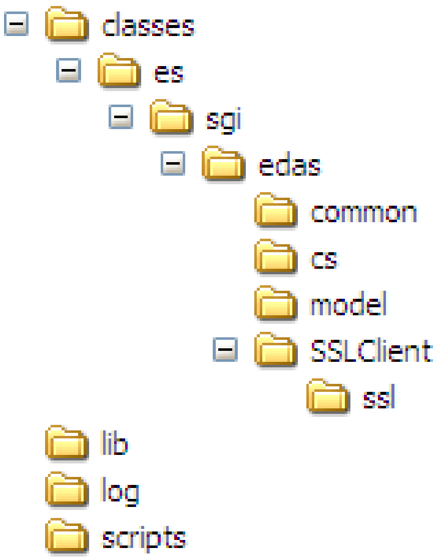
a) For Unix/Linux system this is achieved using the following commands:

```
$ cd INSTALLATION DIRECTORY

$ gunzip client-software.tar.gz

$ tar xvf client-software.tar
```

b) For Windows system this is achieved using the following steps:

```
1.- Open client-software.zip with winzip or similar archive utility

2.- Extract all contents into Installation-Directory
```

Once installed, the CS has the following file structure (in both systems Windows or Unix/Linux):



3 CLIENT SOFTWARE EXECUTION

3.1 CONFIGURING AND STARTING THE CS

After the installation, the user must be edit the configuration file in order to satisfy his needs (all parameters of this file will be explained below), once this file has been customised, the user must launch the Client Software through the following command:

a) Unix/Linux system:

```
$cd INSTALLATION DIRECTORY
$cd scripts
$./cs.sh start
```

b) Windows system:

```
c:\>cd INSTALLATION DIRECTORY
c:\INSTALLATION DIRECTORY>cd scripts
c:\INSTALLATION_DIRECTORY\scripts>cs.bat
```

3.2 PARAMETERS

Client Software has two configuration files:

- *cs.properties*: this file includes Client Software operational configuration parameters, such as:
    - EGNOS Data Server IP
    - EDS Client Control Module port
    - CS port to listen for SP connections
    - CS credentials to authenticate to EDS
    - Various message timing properties

```
#EDS-CCM address/listen port

eds ip=127.0.0.1

port tcp=8888


#CS-SP port

#The TCP Port to listen from Service Provider connections

sp port=3000


#CS UDP port

#The UDP Port to listen from the EDS

edas port=4000


#SSL over TCP,

#Valid values are true or false

SSL enabled=true


#Credentials to login into EDS

user=user

password=pass


#Max. queue size in Mbytes

#(in order to prevent Out of Memory Errors set this value between 5 and 10 Mb)


max queue size=3

[ALRM1]

#A connection is considered IDLE when the timeout is exceeded
```

```
timeout=10000

#String messages exchanged between EDS-CCM and CS
validation ok=validation ok
validation error=validation error
keep alive message=keepMeAlive
keep alive response=youAreAlive

##################################################
#####All the following parameters are in seconds #####
##################################################

#A connection is considered IDLE when the idle time is exceeded
idle time=10

#CS sends a keep alive message every keep alive time seconds
keep alive time=2

#CS monitors the connection to the EDS every monitoring time seconds
monitoring time=5

#When the CS loses connection with EDS, a login attempt is made every retry time seconds
retry_time=1
```

Table 3-1: cs.properties file content

For a detailed description of CS architecture go to **Error! Reference source not found.**.

DISCLAIMER:

> *Once reached the threshold value for the "max_queue_size" parameter, the Client Software does not guarantee data availability.*
>
> *This situation will arise if the transfer rate from the Client Software to the Service Provider is lower than the one from EDAS to the Client Software, whether due to the Service Provider is not able to process the packages or due to a connection with lower transfer rate.*

To check if the transfer of data is fast enough to the Service provider read 3.4.

- *log4j.properties*: basic logging parameters. The values in this file should be modified according to specs described in http://wiki.apache.org/logging-log4j/

```
#log4j.rootCategory=INFO, stdout
#log4j.appender.stdout=org.apache.log4j.ConsoleAppender
#log4j.appender.stdout.layout=org.apache.log4j.PatternLayout
#log4j.appender.stdout.layout.ConversionPattern=[%d{HH:mm:ss]     %p [%c] - %m%n
log4j.rootLogger=INFO, out
log4j.appender.out=org.apache.log4j.DailyRollingFileAppender
log4j.appender.out.File=../log/cs out.log
log4j.appender.out.DatePattern='.'dd-MM-yyyy
log4j.appender.out.layout=org.apache.log4j.PatternLayout
log4j.appender.out.layout.ConversionPattern= %d [%8c{1}]: %5p: %m%n
```

Table 3-2: log4j.properties file content

3.3 LOGS

The CS outputs all status information to the cs_out.log file, that is located in $INSTALLATION_DIRECTORY/log. This log file location is configurable modifying **log4j.appender.out.File** property included in log4j.properties file.

3.4 RUNNING CS FOR FIRST TIME

When running a CS for first time, operation can be checked analyzing log information.

If communication with EDAS is off, following lines will be written in log file:

```
2008-10-23 12:43:04,262 [ClientSoftware]:  INFO: Client Software Instance Started

2008-10-23 12:43:04,278 [  Client]:  INFO: SSL Activated

2008-10-23 12:43:04,356 [ClientSoftware]:  INFO: Listening at port 3000 for SP Connection

2008-10-23 12:43:04,356 [ClientSoftware]:  INFO: Listening at port 30000 for EDS Messages

2008-10-23 12:43:10,122 [  Client]: ERROR: Conection not ready yet. Retrying...

2008-10-23 12:43:17,541 [  Client]: ERROR: Conection not ready yet. Retrying...

2008-10-23 12:43:24,976 [  Client]: ERROR: Conection not ready yet. Retrying...

2008-10-23 12:43:32,395 [  Client]: ERROR: Conection not ready yet. Retrying...
```

Otherwise, in case communication with EDAS will be correct, following lines will be written in log files:

```
2008-01-29 00:00:27,890 [ClientSoftwareToCCMConnectorHandler]: INFO: [/192.168.26.100:8887] WRITE: DirectBuffer[pos=0 lim=12 cap=16: 6B 65 65 70 4D 65 41 6C 69 76 65 0A]

2008-01-29 00:00:27,890 [ClientSoftwareToCCMConnectorHandler]: INFO: [/192.168.26.100:8887] SENT: DirectBuffer[pos=0 lim=12 cap=16: 6B 65 65 70 4D 65 41 6C 69 76 65 0A]

2008-01-29 00:00:27,910 [Client]:  INFO: # messages from EDAS: 1518148, # messages to SP: 1518148

2008-01-29 00:00:28,062 [ClientSoftwareToCCMConnectorHandler]: INFO: [/192.168.26.100:8887] RECEIVED: DirectBuffer[pos=0 lim=12 cap=16: 79 6F 75 41 72 65 41 6C 69 76 65 0A]
```

```
2008-01-29 00:00:28,062 [ClientSoftwareToCCMConnectorHandler]: INFO: Keeping connection alive.
```

In this case the connection to EDAS is correct. There is a message (in green) that shows the messages that are received in EDSToCS handler and the messages that are dispatched to the SP (CSToSP handler). For a more detailed description of CS architecture go to **Error! Reference source not found.**. In case that the the value of packages received increases faster than the packages that are dispatched to the Service provider there is a problem of consumption of messages. The Service Provider is not fast enough to process the data. Once the threshold value is reached, the Client Software does not guarantee data availability. See DISCLAIMER on section 3.2

4 INTERFACE FROM CL. SOFTWARE TO SRV. PROVIDER

This section is aimed to provide the interface information that may be needed in order to help Service Providers to develop software able to receive the EDAS products delivered through the Client Software. The section will include a description of the Service Provider interface of the Client Software and also a description of different case studies, together with an analysis of the Client Software behaviour in any of those cases.

4.1 SERVICE LEVEL AND FORMAT DEFINITIONS

The following *Service Levels* and Formats are provided:

Service Level	**Data**	**Format**	**Remarks**
Service Level 0	INSPIRE/ATC	Raw format.	Please check availability of ATC data. Provision of ATC data in EDAS may be disabled.
	INSPIRE/DATA (a subset)	Encoded in ASN.1.	
Service Level 1	INSPIRE/ATC	Raw format.	Please check availability of ATC data. Provision of ATC data in EDAS may be disabled
	INSPIRE/DATA	Encoded as RTCM messages.	

Table 4-1. Service Levels and Formats Proposed.

The rationale for the selection is:

- *Service Level* 0: it is needed to either transmit data in raw format, or transmit them in a format that allows a lossless reconstruction after translation.
- *Service Level* 1: it is used to transmit data in an open standard. In addition, the *Service Provider* can split the incoming data stream into further groups (e. g., GPS, GLONASS, Augmentation, …).

Figure 5.1 below shows EDAS output format at a very high level. Each EDAS message is embedded in a frame which starts with the EDAS Control Header, which will be explained in section 5.2.3, followed by the platform message whose format depends on the Service Level and the kind of contained data.

As advanced in Table 5-1, INSPIRE/ATC Data are enclosed in the EDAS output frame with no format modification whatsoever, i.e., in the same format as they are received from INSPIRE/ATC. This is to say, INSPIRE/ATC can be provided in raw format for both Service Levels (check availability of ATC data with EDAS operator, provision of ATC data in EDAS may be disabled).

However, INSPIRE/DATA go inside the EDAS frame encoded as ASN.1 for Service Level 0, while they go encoded as RTCM for Level 1.

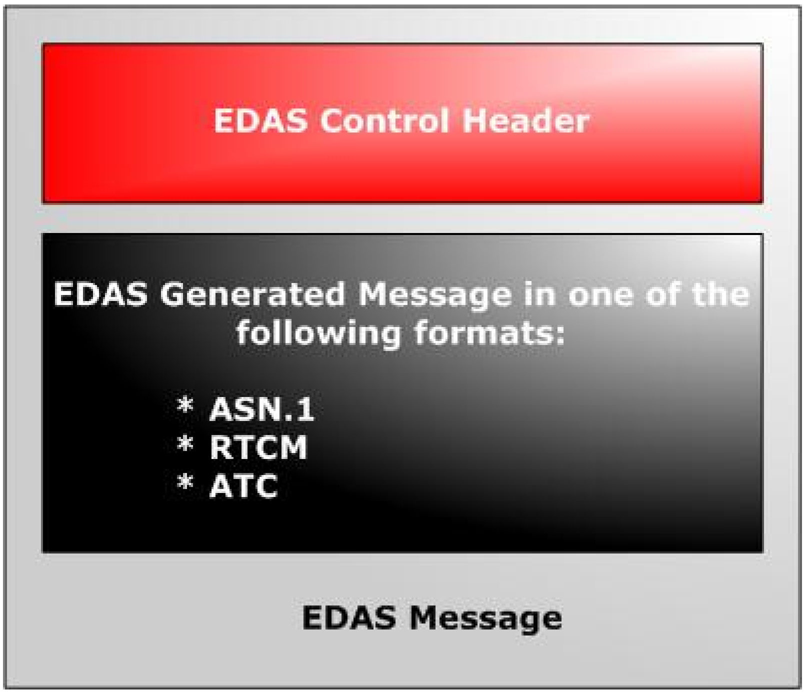


Figure 4-1. EDAS Output format.

In some cases, one EDAS message can encapsulate more than one RTCM message in Level 1, otherwise, EDAS only encapsulate one ASN.1 message into an EDAS message.

For some Level 1 messages a Message Number and a Message Structure have been defined since they are EDAS proprietary messages, and therefore they are not defined in RTCM v3. [RD.4]. For each of these cases the Message Number has been selected in the range of Proprietary Messages (4088 to 4095) specified by RTCM v3.0 [RD.4], in order to avoid overlapping with updates of the standard.

4.2 CONNECTION PROTOCOLS

In the Client Software to Service Provider connection, the Client Software will operate as server in the initial connection, and the Service Provider as client. However, after the initial connection, the data flow will be delivered from Client Software to Service Provider. The connection between CS and SP will be performed over a TCP connection.

The connection handling process can be summarised in the following steps:

1. When the Client Software is initiated, it starts listening for TCP connections on a configurable port.
2. The Service Provider software connects to the Client Software open port using a standard TCP *syn* message.
3. The Client Software replies with a TCP *ack* message.
4. The Client Software starts delivering the messages over the opened TCP channel.

In some situations, depending on the OS context and the installed java virtual machine version, a TCP frame may encapsulate several EDAS frames.

4.2.1 EDAS Control Header

The EGNOS Data Server embeds all platform messages in a new message containing an EDAS Control Header that allows the system to manage added value information, such as quota management or data integrity, without affecting the inserted EGNOS data.

This header contains the following information:

- **Computed message body CRC:** When the Client Service receives the message, it computes the message CRC and checks that the generated CRC and the CRC information in the EDAS Control Header are equal. The CRC is a 32-bits little-endian field.

  This CRC checks the integrity of the EDAS message. There is a possibility that the EGNOS broadcast GEO message (that contains a particular CRC) is wrong but EDAS message has been sent correctly. When this happens it could be observed that the CRC of EGNOS broadcast GEO message is not the expected one (because the message itself is corrupted and so, it should be discarded) but the global CRC of the

EDAS message is correct because it is included to enable the integrity check of the whole EDAS message

Message type: The type of message that the body contains. It may have the following values:

- o **0:** System Information Message
- o **1:** ATC format message
- o **2:** ASN.1 format message
- o **3:** RTCM format message
- o **4:** APC format message

- **Quota information:** Information relative to message quota status.
- **CRC CS:** When CS instances receive a message from the EDS they recalculate the CRC and compare it with the generated by the EDS. If CRCs are not equal, this bit is set to 1, otherwise it contains a 0.
- **CS Not Receiving Data (CSNRD):** When the CS instance does not receive any EDAS message from the EDS, it sends a System Information Message to the SP notifying this event.

The following table shows in detail the EDAS Control Header format:

<table>
<tr><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td colspan="32">CRC-32</td></tr>
<tr><td colspan="8">Byte 1</td><td colspan="8">Byte 2</td><td colspan="8">Byte 3</td><td colspan="8">Byte 4</td></tr>
</table>

<table>
<tr><th>32</th><th>33</th><th>34</th><th>35</th><th>36</th><th>37</th><th>38</th><th>39</th><th>40</th><th>41</th><th>42</th><th>43</th><th>44</th><th>45</th><th>46</th><th>47</th></tr>
<tr><td colspan="8">Message Type</td><td colspan="2">Quota</td><td>CRC CS</td><td>CSNRD</td><td colspan="4">Future Usage</td></tr>
<tr><td colspan="8">Byte 5</td><td colspan="8">Byte 6</td></tr>
</table>

48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
Future Usage															
Byte 7								Byte 8							

Table 4-2. EDAS Control Header definition.

For the quota field, four values can be contained:

Value	Meaning
0 (00)	No quota exceeded
1 (01)	More than 75 % of the quota consumed
2 (10)	More than 90 % of the quota consumed
3 (11)	Quota exceeded. (No message body is sent)

Table 4-3. EDAS Control Header Quota Field.

Note that a message with a value of 3 in the quota field (Quota exceeded) does not contain any message body inside (System Information Message). Furthermore, when the client software receives a quota exceeded message, it closes his connection with the Service Provider.

When CS detects an integrity failure in the communication between it and EDS, it delivers the message but activates a bit of integrity fail warning:

Value	Meaning
0	No CRC problem between EDS and CS
1	The CRC calculated in EDS and the one

Value	Meaning
	calculated in CS are not equal. Data integrity failure in communication between EDS an CS

Table 4-4. EDAS Message Header CRC CS Field.

When CS detects a lack of data from EDS, it sends a System Information Message with the CSNRD field activated:

Value	Meaning
0	CS is receiving data from EDS
1	Since 1 second CS is not receiving data from EDS

Table 4-5. EDAS Message Header CSNRD Field.

4.2.2 RTCM Frame

The basic frame consists of a RTCM v3.0 frame. The structure of the Frame Format is shown in [RD.4].

Preamble	Reserved	Message Length	Variable Length Data Message	CRC
1 byte	6 bits	10 bits	0 – 1023 bytes	24 bits

Table 4-6. Frame Structure.

- The *Preamble* byte, is a fixed 8-bit sequence 11010011.
- The *Reserved* field must be set to zero.
- The *Message Length*. This value specifies the number of bytes in the field *Variable Length Data Message* which immediately follow the bytes for the *Message Length*, excluding the CRC field.
- The *Variable Length Data Message*. Immediately following the *Message Length* field is the *Record Message*. The format of each *Record Message* depends on which type of record it is (identified, by the *Record ID*), and it is described in the *Presentation Layer* chapter.
- The *CRC*. 24 bits. Each message contains a checksum that is generated from all of the bits in the *Preamble, Reserved* field*, Message Length*, and *Variable Length Data Message*. The design for this sum is:
    - QualComm CRC algorithm CRC-24Q: Using generating polynomial = $x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^{7} + x^{6} + x^{5} + x^{4} + x^{3} + x^{1} + x^{0}$.

4.2.3 ASN.1 Frame

With the aim to avoid the use of tools with any kind of royalties to be a requirement for EDAS, the use an open source library for ASN.1 encoding has been decided. It is located in http://lionet.info/asn1c/. This is a free, open source compiler of ASN.1 specifications into C source code. It supports a range of ASN.1 syntaxes, uPer encoding was selected.

ASN.1 message length and ASN.1 message type (Receiver Measurements or Navigation Correction) become then necessary information in order to decode the message. Therefore it was decided to enclosed EDAS ASN.1 uPer encoded messages in to a frame containing a header with following specifications:

- **Location**

EDAS Control Header	ASN.1 Frame

- **Internal design (ASN.1 frame)**

DF: ASN.1 Message Type	DF: ASN.1 Message Length	ASN.1 encoded message

- **Data fields:**

DF Name	DF Range	Data Type	Data Field Notes
ASN.1 Message Type	{0,1}	Bit(1)	Defines the type of ASN.1 message. When the

			Bit is "0" the message is "Navigation Correction Message"[1]; when the Bit is "1" the message is "Receiver Measurements".
ASN.1 Message Length	0 - 32767	Uint15	Defines the size of the ASN.1 message in bytes. The size defined here discards the ASN.1 header size.

- **Data ordering:**
  - Header data ordering is big-endian bitwise and bytewise.
  - ASN.1 message data ordering is the one define by the ASN.1 standards.

ASN.1 message specifications can be found in Annex A and Annex B (page 46) of this document.

4.2.4 EDAS Level 0 messages

A detailed definition of the level 0 messages delivered through the Client Software can be found in [RD.2]. Apart from the messages defined in [RD.2], Level 0 include ATC messages and Station Description data.

4.2.4.1 Message 4091. ATC.

ATC information is also included in Service Level 0. The Message Type 4091 provides the ATC Data, as described in [RD.3] §3.2.3 and [RD.2] §4.3. ***Please check availability of ATC data. Provision of ATC data in EDAS may be disabled***.

Note this is a EDAS-defined message, since [RD.4] does not define any means to send this information.

Note that 8 different ATC messages are defined in [RD.2] §4.3.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 4091	DF002	Uint12	12
ATC Message Identifier	DF90	int8	8
ATC Message (one full message defined in [RD.2] §4.3).			

[1] NLES Feedback message

Table 4-7: Message 4091. ATC.

4.2.4.2 Message 4092. RIMS APC Data

The EDAS-specific Message Type 4092 provides a data record with position information on Antenna Phase Center for L1 and L2 for each RIMS channel currently described in a given configuration file. The configuration file is supposed to contain the required information for each RIMS channel currently used in EGNOS. Updates of the configuration file will be the responsibility of the operator.

The message heading is 20 bits long.

The Data Block length depends on the number of available RIMS channels described in the configuration file, i.e. on the number of included RIMS APC records (maximal 250).

IMPORTANT Note: the message is presented as a placeholder to convey information, that as of today it is not provided by INSPIRE.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 4092	DF002	Uint12	12
Number of total RTCM messages needed for the APC message	DF119	Uint4	4
Number of RTCM message in the APC RTCM messages sequence	DF120	Uint4	4
Number of included RIMS APC records	DF021	Unit8	8
RIMS Channel-Specific Portion, repeated for each RIMS Channel currently available in EGNOS.	–	See Table 5-3 below	218 bits for each RIMS

Table 4-8: Message 4092. RIMS APC Data. Heading and Data Block.

Table 5-3 describes the RIMS Channel-Specific Portion, repeated for each RIMS channel currently available in EGNOS

Data Field (DF)	DF Number	Data Type	Number of Bits
RIMS ID	DF022	Uint8	8
Position L1 X	DF023	Int38	38
Position L1 Y	DF024	Int38	38
Position L1 Z	DF025	Int38	38
Delta Position L2 X	DF026	Int32	32
Delta Position L2 Y	DF027	Int32	32
Delta Position L2 Z	DF028	Int32	32

Table 4-9: Message 4092. RIMS APC Data. Data Block.

Note: All integer fields are represented as **two's complement notation**. In two's complement notation, a positive number is represented by its ordinary binary representation, using enough bits that the high bit, the sign bit, is 0. The two's complement operation is the negation operation, so negative numbers are represented by the two's complement of the representation of the absolute value.

In finding the two's complement of a binary number, the bits are inverted, or "flipped", by using the bitwise NOT operation; the value of 1 is then added to the resulting value. Bit overflow is ignored, which is the normal case with zero.

For example, beginning with the signed 38-bit binary representation of the decimal value 5, using subscripts to indicate the base of a representation needed to interpret its value:

$$00\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0000\ 0101_2 = 5_{10}$$

The most significant bit is 0, so the pattern represents a non-negative (positive) value. To convert to −5 in two's-complement notation, the bits are inverted; 0 becomes 1, and 1 becomes 0:

$$11\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1010$$

At this point, the numeral is the **one's complement** of the decimal value 5. To obtain the **two's complement**, 1 is added to the result, giving:

$$11\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1111\ 1011_2 = -5_{10}$$

The result is a signed binary number representing the decimal value −5 in two's-complement form. The most significant bit is 1, so the value represented is negative.

4.2.5 EDAS Level 1 messages

This section describes the *Messages* for Level 1 Subscription Level. Each *Message* contains a specific set of *Data Fields* (described in next chapter 4.2.6), that can be repeated as needed (e. g., in order to provide information of several satellites).

The *Data Fields* are broadcast in the order listed.

Multi-byte values are expressed with the most significant byte transmitted first and the least significant byte transmitted last.

4.2.5.1 Message 1004. GPS Observations.

The Message Type 1004 supports dual-frequency GPS data, and includes several indicators measured by the reference station.

The message heading is 64 bits long.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 1004	DF002	Uint12	12
Reference Station ID	DF003	Uint12	12
GPS Epoch Time (TOW)	DF004	Uint30	30
Synchronous GNSS Message Flag.	DF005	Bit(1)	1
Number of GPS Satellite Signals Processed	DF006	Uint5	5
GPS Divergence-free Smoothing Indicator. It is always set to ‘0’.	DF007	Bit(1)	1
GPS Smoothing Interval. It is always set to ‘0’.	DF008	Bit(3)	3
GPS Satellite-Specific Portion, repeated for each GPS Satellite Processed	-	See Table 4-11 below	125 bits for each Satellite

Table 4-10: Message 1004. GPS Observations. Heading and Data Block.

Table 4-11 describes the GPS Satellite-Specific Portion, repeated for each GPS Satellite Processed.

Data Field (DF)	DF Number	Data Type	Number of Bits
GPS Satellite ID	DF009	Uint6	6
GPS L1 Code Indicator	DF010	Bit(1)	1
GPS L1 Pseudorange	DF011	Uint24	24
GPS L1 PhaseRange - L1 Pseudorange	DF012	Uint20	20
GPS L1 Lock Time Indicator	DF013	Uint7	7
GPS Integer L1 Pseudorange Modulus Ambiguity	DF014	Uint8	8
GPS L1 CNR	DF015	Uint8	8
GPS L2 Code Indicator	DF016	Bit(2)	2
GPS L2-L1 Pseudorange Difference	DF017	Uint14	14
GPS L2 PhaseRange – L1 Pseudorange	DF018	Uint20	20
GPS L2 Lock Time Indicator	DF019	Uint7	7
GPS L2 CNR	DF020	Uint8	8

Table 4-11: Message 1004. GPS Observations. Data Block.

4.2.5.2 Message 1010. GLONASS Observations.

The Message Type 1010 supports single-frequency GLONASS data, and includes several indicators measured by the reference station.

The message heading is 61 bits long.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 1010	DF002	Uint12	12
Reference Station ID	DF003	Uint12	12
GLONASS Epoch Time (t_k)	DF034	Uint27	27
Synchronous GNSS Message Flag.	DF005	Bit(1)	1
Number of GLONASS Satellite Signals Processed.	DF035	Uint5	5

Data Field (DF)	DF Number	Data Type	Number of Bits
GLONASS Divergence-free Smoothing Indicator. It is always set to '0'.	DF036	Bit(1)	1
GLONASS Smoothing Interval. It is always set to '0'.	DF037	Bit(3)	3
GLONASS Satellite-Specific Portion, repeated for each GLONASS Satellite Processed	-	See Table 4-13 below	79 bits for each Satellite

Table 4-12: Message 1010. GLONASS Observations. Heading and Data Block.

Table 4-13 describes the GLONASS Satellite-Specific Portion, repeated for each GLONASS Satellite Processed.

Data Field (DF)	DF Number	Data Type	Number of Bits
GLONASS Satellite ID (Satellite Slot Number)	DF038	Uint6	6
GLONASS L1 Code Indicator	DF039	Bit(1)	1
GLONASS Satellite Frequency Channel Number	DF040	Uint5	5
GLONASS L1 Pseudorange	DF041	Uint25	25
GLONASS L1 PhaseRange - L1 Pseudorange	DF042	Uint20	20
GLONASS L1 Lock Time Indicator	DF043	Uint7	7
GLONASS Integer L1 Pseudorange Modulus Ambiguity	DF044	Uint7	7
GLONASS L1 CNR	DF045	Uint8	8

Table 4-13: Message 1010. GLONASS Observations. Data Block.

4.2.5.3 Message 4089. GEO Observations.

The EDAS-specific Message Type 4089 supports single-frequency GEO data, and includes several indicators measured by the reference station.

The message heading is 64 bits long.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 4089	DF002	Uint12	12
Reference Station ID	DF003	Uint12	12
GPS Epoch Time (TOW)	DF004	Uint30	30
Synchronous GNSS Message Flag.	DF005	Bit(1)	1
Number of GEO Satellite Signals Processed	DF006	Uint5	5
GEO Divergence-free Smoothing Indicator. It is always set to '0'.	DF007	Bit(1)	1
GEO Smoothing Interval. It is always set to '0'.	DF008	Bit(3)	3
GEO Satellite-Specific Portion, repeated for each GEO Satellite Processed	-	See Table 4-15 below	64 bits for each Satellite

Table 4-14: Message 4089. GEO Observations. Heading and Data Block.

Table 4-13 describes the GEO Satellite-Specific Portion, repeated for each GEO Satellite Processed.

Data Field (DF)	DF Number	Data Type	Number of Bits
GEO Satellite ID	DF009	Uint6	6
GEO L1 Code Indicator	DF010	Bit(1)	1
GEO L1 Pseudorange	DF011	Uint24	24
GEO L1 PhaseRange - L1 Pseudorange	DF012	Uint20	20
GEO L1 Lock Time Indicator	DF013	Uint7	7
GEO Integer L1 Pseudorange Modulus Ambiguity	DF014	Uint8	8
GEO L1 CNR	DF015	Uint8	8

Table 4-15: Message 4089. GEO Observations. Data Block.

4.2.5.4 Message 4088. GPS/GLONASS/GEO Ephemeris.

The Message Type 4088 provides the Ephemeris of GPS, GLONASS and GEO satellites, as sent by each receiver.

Note this is a EDAS-defined message, since [RD.4] does not define ay means to send this information.

With the aim of preserving as possible the incoming INSPIRE information, the same data structure is used. The data structure is based on the GPS/GLONASS/GEO Signal In Space.

The Message contains the information provided by a receiver:

- The 50 bits Navigation information broadcast by each GPS satellite.
- The 50 bits Navigation information broadcast by each GLONASS satellite.
- The 250 bits Navigation/Corrections information broadcast by each GEO satellite.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 4088	DF002	Uint12	12
Reference Station ID	DF003	Uint12	12
GPS Epoch Time (TOW)	DF004	Uint30	30
Number of GPS Satellite Signals Processed	DF006	Uint5	5
GPS Satellite-Specific Portion, repeated for each GPS Satellite Processed	-	See Table 4-17 below	50 bits for each Satellite
Number of GLONASS Satellite Signals Processed	DF035	Uint5	5
GLONASS Satellite-Specific Portion, repeated for each GLONASS Satellite Processed	-	See below	50 bits for each Satellite
Number of GEO Satellite Signals Processed	DF006	Uint5	5
GEO Satellite-Specific Portion, repeated for each GEO Satellite Processed	-	See below	250 bits for each Satellite

Table 4-16: Message 4088. Ephemeris.

Table 4-17 describes the GPS Satellite-Specific Portion, repeated for each GPS Satellite Processed.

Data Field (DF)	DF Number	Data Type	Number of Bits
GPS Satellite ID	DF009	Uint6	6

Data Field (DF)	DF Number	Data Type	Number of Bits
GPS Navigation message	DF900	Bit(50)	50

Table 4-17: Message 4088. GPS Ephemeris. Data Block.

Table 4-18 describes the GLONASS Satellite-Specific Portion, repeated for each GLONASS Satellite Processed.

Data Field (DF)	DF Number	Data Type	Number of Bits
GLONASS Satellite ID (Satellite Slot Number)	DF038	Uint6	6
GLONASS Navigation message received each second from one GLONASS satellite.	DF901	Bit(50)	50

Table 4-18: Message 4088. GLONASS Ephemeris. Data Block.

Table 4-19 describes the GEO Satellite-Specific Portion, repeated for each GEO Satellite Processed.

Data Field (DF)	DF Number	Data Type	Number of Bits
GEO Satellite ID	DF009	Uint6	6
GEO Navigation message received each second from one GEO satellite.	DF902	Bit(250)	250

Table 4-19: Message 4088. GEO Ephemeris. Data Block.

4.2.5.5 Message 4090. NLES Cyclic Feedback.

The Message Type 4090 provides the NLES Cyclic Feedback.

Note this is a EDAS-defined message, [RD.4] does not define ay means to send this information.

Data Field (DF)	DF Number	Data Type	Number of Bits
Message Number 4090	DF002	Uint12	12
EGNOS Address (address of the Originating Element)	DF903	uint16	16
GPS Epoch Time (TOW)	DF004	Uint30	30

Data Field (DF)	DF Number	Data Type	Number of Bits
NLES Feed-back validity bits	DF904	bit(8)	8
GEO message selected for Uplink	DF902	Bit(250)	250
GEO Clock Drift	DF905	int32	32
Selected CPF (EGNOS Address)	DF903	uint16	16
GEO Uplinked previous cycle (N-1)	DF902	Bit(250)	250
Selected (N-1) CPF (EGNOS Address)	DF903	uint16	16
GEO Uplinked previous cycle (N-2)	DF902	Bit(250)	250
Selected (N-2) CPF (EGNOS Address)	DF903	uint16	16
GEO Uplinked previous cycle (N-3)	DF902	Bit(250)	250
Selected (N-3) CPF (EGNOS Address)	DF903	uint16	16

Table 4-20: Message 4090. NLES Cyclic Feedback.

4.2.5.6 Message 4091. ATC.

A description of Message 4091 can be found in §4.2.4.1

4.2.5.7 Message 4092. RIMS APC Data

INSPIRE.C Data. Heading and Data Block.el currently available in EGNOSA description of Message 4092 can be found in §4.2.4.2

4.2.6 Data Fields

This section describes the *Data Fields* used in the definition of the *Messages* in §4.2.4 and §4.2.5.

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
DF002	Message Number	0 - 4095		Uint12	
DF119	Number of total RTCM messages needed for the APC message	0 - 15		Uint4	
DF120	Number of RTCM	0 - 15		Uint4	

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
	message in the APC RTCM messages sequence				
DF003	Reference Station ID	0 - 4095		Uint12	Unique identifier. DF003 is generated from EGNOS Standard Header: Origin Address (which is 0 – 65535).
DF004	GPS Epoch Time (TOW)	0 – 604799999 ms	1 ms	Uint30	GPS Epoch Time is provided in milliseconds from the beginning of the GPS week, which begins at midnight GMT on Saturday night/Sunday morning, measured in GPS time (as opposed to UTC). DF004 is computed from EGNOS Standard Header: Time Stamp (after time shifting).
DF005	Synchronous GNSS Message Flag	–		Bit(1)	If the Synchronous GNSS Message Flag is set to '0', it means that no further GNSS observables referenced to the same Epoch Time will be transmitted. This enables the receiver to begin processing the data immediately after decoding the message. If it is set to '1', it means that the next message will contain observables of another GNSS source referenced to the same Epoch Time. In EDAS, DF005 is always set to '0', because the Service Provider is in charge of processing the different data groups.
DF006	Number of GPS Satellite Signals Processed	0 – 31		Uint5	The Number of GPS Satellites processed in the message. DF006 is computed from EGNOS message.
DF007	GPS Divergence-free Smoothing Indicator	–		Bit(1)	0= Divergence-free smoothing not used 1= Divergence-free smoothing

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
					used. DF007 is always set to '0'; i. e., Divergence-free smoothing not used
DF008	GPS Smoothing Interval	Refer to Table 4-22		Bit(3)	The GPS Smoothing Interval is the integration period over which reference station pseudorange code phase measurements are averaged using carrier phase information. Divergence-free smoothing may be continuous over the entire period the satellite is visible. DF008 will be always set to '0' in EDAS, i. e., No smoothing.
DF009	GPS/GEO Satellite ID	1 – 37 (GPS) 40 – 58 (GEO)		Uint6	A GPS Satellite ID number from 1 to 32 refers to PRN code of the GPS satellite. Satellite ID's higher than 32 are reserved for satellite signals from SBAS. Note that EGNOS allocates GPS PRN 1 to 37, although PRN 32 – 37 are not used by NAVSTAR GPS due to Gold code characteristics (e.g. C/A codes 34 and 37 are common). SBAS PRN codes cover the range 120 – 138. The satellite ID's reserved for SBAS satellites are 40-58, so that SBAS PRN codes are derived from the Satellite ID codes by adding 80. DF009 is taken from EGNOS "GPS PRN" field, filtering out GPS satellites with PRN > 32.
DF010	GPS L1 Code Indicator	–		Bit(1)	The GPS L1 Code Indicator identifies the code being tracked. "0" = C/A Code; "1" = P(Y) Code Direct DF010 is always set to '0' in EDAS.
DF011	GPS L1 Pseudorange	0-299792.46 m	0.02 m	Uint24	The GPS L1 Pseudorange field provides the raw L1 pseudorange measurement in meters, modulo one light-millisecond (299,792.458 meters). The measurement can be reconstructed by: (GPS L1 pseudorange measurement) = (GPS L1 pseudorange field) modulo

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
					(299,792.458 m) + integer GPS Integer L1 Pseudorange Modulus Ambiguity. If DF012 is set to 80000h, this field does not represent a valid L1 pseudorange, and is used only in the calculation of L2 measurements. DF011 is computed from EGNOS "GPS Primary Pseudorange".
DF012	GPS L1 PhaseRange - L1 Pseudorange	± 262.1435 m	0.0005 m	int20	The DF012 provides the information necessary to determine the L1 Phase Measurement. Note that the PhaseRange defined here has the same sign as Pseudorange. At start-up and after each cycle slip, the initial ambiguity is reset and chosen so that PhaseRange matches the L1 Pseudorange (i.e., within 1/2 L1 cycle). The Full GPS L1 PhaseRange is: (Full L1 PhaseRange) = (L1 pseudorange + DF012. Note: in case Pseudorange and PhaseRange diverge, the DF012 is adjusted (rolled over) by the equivalent of 1500 cycles. If DF012 is set to 80000h, indicates the L1 phase is invalid, and that DF011 field is used only in the calculation of L2 measurements. DF012 is computed from EGNOS "GPS L1 Accumulated Doppler" and "GPS Primary Pseudorange".
DF013	GPS L1 Lock Time Indicator	Refer to Table 4-23		Uint7	The DF013 provides a measure of the time that has elapsed during which the receiver has maintained continuous lock on that the satellite signal. If a cycle slip occurs during the previous measurement DF013 will be reset to zero. DF013 is computed from EGNOS "L1 Signal Quality", bits 4-5.
DF014	GPS Integer L1 Pseudorange Modulus Ambiguity	0 - 76447076.790 m	299792.458 m	Uint8	DF014 represents integer number of full pseudorange modulus divisions (299792.458 m) of the raw L1 pseudorange DF014 is computed from EGNOS

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
					“GPS Primary Pseudorange” field.
DF015	GPS L1 CNR	0-63.75 dB-Hz	0.25 dB-Hz	Uint8	DF015 provides the receiver estimate of the carrier-to-noise ratio of the satellite signal in dB-Hz. The value “0” means that the CNR is not computed. DF015 is computed from EGNOS “GPS L1 C/N0” field.
DF016	GPS L2 Code Indicator	–		Bit(2)	DF016 identifies which L2 code is being processed. “0” = C/A or L2C Code; “1” = P(Y) Code Direct; “2“ = P(Y) code cross-correlated “3“ = Correlated P/Y DF016 is always set to ‘0’ in EDAS.
DF017	GPS L2-L1 Pseudorange Difference	± 163.82 m	0.02m	int14	DF017 provides the information necessary to determine the L2 pseudorange. The GPS L2 Pseudorange is: (GPS L2 Pseudorange) = (L1 pseudorange + DF017. If DF017 is set to 2000h (-163.84m),means that there is no valid L2 code available, or that the value exceeds the allowed range. DF017 is taken from EGNOS “GPS L2-L1 Pseudorange”.
DF018	GPS L2 PhaseRange – L1 Pseudorange	± 262.1435 m	0.0005 m	int20	DF018 provides the information necessary to determine the L2 phase measurement. Note that the PhaseRange defined here has the same sign as Pseudorange. At start-up and after each cycle slip, the initial ambiguity is reset and chosen so that L2 PhaseRange matches the L1 Pseudorange (i.e., within 1/2 L2 cycle). The Full GPS L2 PhaseRange is: (Full L2 PhaseRange) = (L1 pseudorange + DF018. Note: in case Pseudorange and PhaseRange diverge, the DF018 is adjusted (rolled over) by the equivalent of 1500 cycles. If DF018 is set to 80000h, indicates the L2 phase is invalid. DF018 is computed from EGNOS “GPS L2 Accumulated Doppler”, “GPS L2-L1 Pseudorange” and “GPS Primary Pseudorange”.

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
DF019	GPS L2 Lock Time Indicator	Refer to Table 4-23		Uint7	The DF019 provides a measure of the time that has elapsed during which the receiver has maintained continuous lock on that the satellite signal. If a cycle slip occurs during the previous measurement DF019 will be reset to zero. DF019 is computed from EGNOS “L2 Signal Quality”, bits 4-5.
DF020	GPS L2 CNR	0-63.75 dB-Hz	0.25 dB-Hz	Uint8	DF020 provides the receiver estimate of the carrier-to-noise ratio of the satellite signal in dB-Hz. The value “0” means that the CNR is not computed. DF020 is computed from EGNOS “GPS L2 C/N0” field.
DF021	Number of included RIMS APC records.	1- 250	1	Uint8	For each RIMS channel currently used in EGNOS a record with APC position information is included.
DF022	RIMS ID	1- 250	1	Uint8	EGNOS Address identifying the RIMS. This value shall match to the value contained in the “Origin address” field of the EGNOS Standard Data Header for RIMS messages transmitted by INSPIRE as described in the INSPIRE ICD ([AD.1]). IDs 1 to 80 are used for RIMS A, IDs 81 to 160 are used for RIMS B, IDs 161 to 240 are used for RIMS C, IDs 241 to 250 are reserved.
DF023	Position L1 X	± 64000.00 m	0.001 m	Int38	X Coordinate for the position of the L1 center of phase of the RIMS channel expressed in Earth Centered Cartesian coordinates in the WGS84 system.
DF024	Position L1 Y	± 64000.00 m	0.001 m	Int38	Y Coordinate for the position of the L1 center of phase of the RIMS channel expressed in Earth Centered Cartesian coordinates in the WGS84 system.
DF025	Position L1 Z	± 64000.00 m	0.001 m	Int38	Z Coordinate for the position of the L1 center of phase of the RIMS channel expressed in Earth Centered Cartesian coordinates in the WGS84

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
					system.
DF026	Delta Position L2 X	± 1.000 m	0.001 m	Int32	X Coordinate for the difference vector from L1 antenna phase centre to the L2 antenna phase centre of omni-directional NLES antenna expressed in Earth Centered Cartesian coordinates in the WGS84 system.
DF027	Delta Position L2 Y	± 1.000 m	0.001 m	Int32	Y Coordinate for the difference vector from L1 antenna phase centre to the L2 antenna phase centre of omni-directional NLES antenna expressed in Earth Centered Cartesian coordinates in the WGS84 system.
DF028	Delta Position L2 Z	± 1.000 m	0.001 m	Int32	Z Coordinate for the difference vector from L1 antenna phase centre to the L2 antenna phase centre of omni-directional NLES antenna expressed in Earth Centered Cartesian coordinates in the WGS84 system.
DF034	GLONASS Epoch Time (t_k)	0 – 86400.999 ms	1 ms	Uint27	Rolls over at 86,400 seconds, except for the leap second second, where it rolls over at 86,401. DF034 is computed from EGNOS Standard Header: Time Stamp (after time shifting: tGLONASS = UTC + 03 hours 00 minutes).
DF035	Number of GLONASS Satellite Signals Processed	0 – 31		Uint5	The Number of GLONASS Satellites processed in the message. DF035 is computed from EGNOS message.
DF036	GLONASS Divergence-free Smoothing Indicator	–		Bit(1)	0= Divergence-free smoothing not used 1= Divergence-free smoothing used. DF036 is always set to ‘0’; i. e., Divergence-free smoothing not used

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
DF037	GLONASS Smoothing Interval	Refer to Table 4-22		Bit(3)	The GLONASS Smoothing Interval is the integration period over which reference station pseudorange code phase measurements are averaged using carrier phase information. Divergence-free smoothing may be continuous over the entire period the satellite is visible. DF037 will be always set to ‘0’ in EDAS, i. e., No smoothing.
DF038	GLONASS Satellite ID (Satellite Slot Number)	1 - 24		Uint6	A GLONASS Satellite ID number from 1 to 24 refers to the slot number of the GLONASS satellite.
DF039	GLONASS L1 Code Indicator	–		Bit(1)	The GLONASS L1 Code Indicator identifies the code being tracked. “0” = C/A Code; “1” = P(Y) Code Direct DF039 is always set to ‘0’ in EDAS.
DF040	GLONASS Satellite Frequency Channel Number	0 – 20		Uint5	Information is not provided in INSPIRE, it will be always set to 0.
DF041	GLONASS L1 Pseudorange	0-599,584.92 m	0.02 m	Uint25	The GLONASS L1 Pseudorange field provides the raw L1 pseudorange measurement in meters, modulo two light-millisecond (599,584.916 meters). The measurement can be reconstructed by: (GLONASS L1 pseudorange measurement) = (GLONASS L1 pseudorange field) modulo (599,584.916 m) + integer GLONASS Integer L1 Pseudorange Modulus Ambiguity. DF041 is computed from EGNOS “GLONASS Primary Pseudorange”.
DF042	GLONASS L1 PhaseRange - L1 Pseudorange	± 262.1435 m	0.0005 m	int20	The DF042 provides the information necessary to determine the L1 Phase Measurement. Note that the PhaseRange defined here has the same sign as Pseudorange. At start-up and after each cycle

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
					slip, the initial ambiguity is reset and chosen so that PhaseRange matches the L1 Pseudorange (i.e., within 1/2 L1 cycle). The Full GLONASS L1 PhaseRange is: (Full L1 PhaseRange) = (L1 pseudorange + DF042. Note: in case Pseudorange and PhaseRange diverge, the DF042 is adjusted (rolled over) by the equivalent of 1500 cycles. If DF042 is set to 80000h, indicates the L1 phase is invalid, and that DF011 field is used only in the calculation of L2 measurements. DF042 is computed from EGNOS "GLONASS L1 Accumulated Doppler" and "GLONASS Primary Pseudorange".
DF043	GLONASS L1 Lock Time Indicator	Refer to Table 4-23		Uint7	The DF043 provides a measure of the time that has elapsed during which the receiver has maintained continuous lock on that the satellite signal. If a cycle slip occurs during the previous measurement DF043 will be reset to zero. DF043 is computed from EGNOS "L1 Signal Quality", bits 4-5.
DF044	GLONASS Integer L1 Pseudorange Modulus Ambiguity	0 - 76447076. 790 m	599,584.916 m	Uint7	DF044 represents integer number of full pseudorange modulus divisions (599,584.916 m) of the raw L1 pseudorange DF044 is computed from EGNOS "GLONASS Primary Pseudorange" field.
DF045	GLONASS L1 CNR	0-63.75 dB-Hz	0.25 dB-Hz	Uint8	DF045 provides the receiver estimate of the carrier-to-noise ratio of the satellite signal in dB-Hz. The value "0" means that the CNR is not computed. DF045 is computed from EGNOS "GLONASS L1 C/N0" field.
DF900	GPS Navigation message			bit(50)	EDAS-specific. GPS Navigation message received each second from one GPS satellite. DF900 is taken from EGNOS "GPS 50 bits message" field.

DF#.	DF Name	DF Range	DF Resolution	Data Type	Data Field Notes
DF901	GLONASS Navigation message			bit(50)	EDAS-specific. GLONASS Navigation message received each second from one GLONASS satellite. DF901 is taken from EGNOS "GLONASS 50 bits message" field.
DF902	GEO Navigation message			bit(250)	EDAS-specific. GEO Navigation message received each second from one GEO satellite. DF902 is taken from EGNOS "GEO 250 bits message" field.
DF903	EGNOS Address	0 - 65535		uint16	EDAS-specific. DF903 is taken from EGNOS Address field.
DF904	NLES Feed-back validity bits			bit(8)	EDAS-specific. Validity associated with the present submessage. It indicates the presence of each of the 4 possible previous messages. bit 0: Message selected for Uplink valid. bit 1: GEO message Uplinked previous cycle (N-1) valid. bit 2: GEO message Uplinked cycle (N-2) valid. bit 3: GEO message Uplinked cycle (N-3) valid. bits 4 to 7: reserved. DF904 is taken from EGNOS "NLES Feed-back validity bits" field.
DF905	GEO Clock Drift	-8 E-6 to +8E-6	5E-11	int32	EDAS-specific. GEO Clock Drift specifies the time drift between the NLES Frequency Standard and the GEO payload clock. Note that MSB is always set to 0. DF905 is taken from EGNOS "GEO Clock Drift" field.

Table 4-21: Data Fields.

Indicator	Smoothing Interval
0	No smoothing
1	< 30 s
2	30-60 s
3	1-2 min

Indicator	Smoothing Interval
4	2-4 min
5	4-8 min
6	>8 min
7	Unlimited smoothing interval

Table 4-22: Carrier Smoothing Interval of Code Phase, Data Fields DF008 and DF037.

Indicator (i)	Minimum Lock Time (s)	Range of Indicated Lock Times
0 - 23	i	0 < lock time < 24
24 - 47	I*2 -24	24 <= lock time <72
48 – 71	I*4 – 120	72 <= lock time < 168
72 – 95	I*8 – 408	168 <= lock time < 360
96 – 119	I*16 – 1176	360 <= lock time < 744
120 – 126	I*32 – 3096	744 <= lock time < 937
127	----	lock time >= 937

Table 4-23: Lock Time Indicator, Data Fields DF013, DF019, DF043, DF049.

4.2.7 Data Types

The data types used are shown in Table 4-24

Data Type	Description	Range	Data Type Notes
Bit(n)	Bit field	0 or 1, each bit	Reserved bits are set to 0
Char8(n)	8 bit characters, ISO 8859-1 (not limited to ASCII)	character set	Reserved or unused characters are set to 0x00
int14	14 bit 2's complement integer	-8192 to +8191	
int20	20 bit 2's complement integer	-524288 to +524287	

Data Type	Description	Range	Data Type Notes
int32	32 bit 2's complement integer	-2147483648 to +2147483647	
int38	38 bit 2's complement integer	-137438953472 to +137438953471	
Unit4	4 bit unsigned integer	0 to 15	
uint5	5 bit unsigned integer	0 to 31	
Uint6	6 bit unsigned integer	0 to 63	
Uint7	7 bit unsigned integer	0 to 127	
Uint8	8 bit unsigned integer	0 to 255	
Uint12	12 bit unsigned integer	0 to 4095	
Uint16	16 bit unsigned integer	0 to 65535	
Uint20	20 bit unsigned integer	0 to 1048575	
Uint24	24 bit unsigned integer	0 to 16777215	
Uint25	25 bit unsigned integer	0 to 33554431	
Uint27	27 bit unsigned integer	0 to 134217727	
Uint30	30 bit unsigned integer	0 to 1073741823	

Table 4-24: Data Types.

4.3 CASE STUDIES

The current section is aimed to provide a description of the Client Software behaviour in different situations, including normal operation.

4.3.1 Case Study 1: Normal Operation

In a normal operation process, the following steps could be foreseen:

1. Execution of Client Software, following the operation described in §3.1. A TCP port will be opened and listen for connections.

2. Execution of Service Provider software, and initial TCP connection to the Client Software.
3. Client software TCP response, and data delivery startup.
4. Upon execution of the appropriate command, the Client Software can be stopped.

4.3.2 Case Study 2: Data Flow Interruption

One of the cases in which the Client Software would deliver additional information to the SP interface would be in the case of an interruption of the data flow coming from EDAS. In this case the Client Software would behave as follows:

1. While receiving the EDAS products, an interruption of data flow takes place.
2. The Client Software detects the interruption, and waits for a second to the data flow to be restored.
3. In case the data flow is not restored in a second, the Client Software sends an empty message to the SP with the CSNRD field of the EDAS Control Header (see §4.2.1) set to 1.

4.3.3 Case Study 3: Data CRC error

Another case in which the Client Software would deliver additional information to the SP interface would be in the case of an a message failing the CRC check. In this case the Client Software would behave as follows:

1. While receiving the EDAS products, a message fails the CRC check.
2. The Client Software sends an empty message to the SP with the CRC CS field of the EDAS Control Header (see §4.2.1) set to 1.

5 ANNEX A: ASN.1 SCHEMAS SPECIFICATION

5.1 MODULE EDAS-COMMON

```
EDAS-Common DEFINITIONS AUTOMATIC TAGS ::=
BEGIN


EGNOSProductHeader ::= SEQUENCE {
	originAddress			EGNOSaddress,
	messageTimeTag				SecondTTag}

EGNOSaddress ::= INTEGER (0 .. 65535)



--=====================================================================--
-- Min Value: 0x2^-16 = 0                                              --
-- Max Value: 2^48x2^-16 = 2^32 = 4,294967296e+9                       --
--=====================================================================--
SecondTTag ::=     REAL (WITH COMPONENTS {   mantissa (0..281474976710656),
                                 base (2),
                                 exponent (-16) })


--=====================================================================--
-- It shall be keep in mind that EGNOS flag bit equal to 0 (zero) could --
-- be translate by boolean value TRUE for a given EGNOS flag and FALSE  --
-- for another one.                                                     --
--=====================================================================--
Valid ::=    BOOLEAN


--=====================================================================--
-- SV PRN allocation:                                                  --
-- GPS PRN range : 1 -> 37                                             --
-- GLONASS PRN range : 38 -> 61                                        --
-- GEO PRN range : 120 -> 138                                          --
--                                                                     --
-- Warning:                                                            --
-- 211 is reserved for RIMS-A Signal generator for UTC-ENT computation. --
-- RIMS Channels will provide information on this satellite PRN as RIMS --
-- GEO Mesurements                                                     --
--=====================================================================--
SV-PRN ::= INTEGER (0 .. 211)


--=====================================================================--
-- GEO 250 bits navigation message as transmitted each second by GEO    --
-- The bits are in the same sequence in the BIT STRING than the sequence --
-- received in the signal in space by the RIMS.                         --
--=====================================================================--
GeoNavigationMessage ::= BIT STRING (SIZE (250))

END
```

5.2 MODULE EDAS-EGNOSRECEIVERMEASUREMENTS

```
EDAS-EGNOSReceiverMeasurements DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
       EGNOSProductHeader,
       Valid,
       SV-PRN,
       GeoNavigationMessage FROM EDAS-Common;


EGNOSReceiversMeasurements ::= SEQUENCE {
       productHeader                    EGNOSProductHeader,
       receiverMeasurements                    ReceiverMeasurements}


ReceiverMeasurements ::= SEQUENCE {
listOfGpsReceiverMeasurements           SEQUENCE (SIZE (0 .. 12)) OF
GPSReceiverMeasurements,
       listOfGlonassReceiverMeasurements       SEQUENCE (SIZE (0 .. 12)) OF
GLONASSReceiverMeasurements,
       listOfGeoReceiverMeasurements           SEQUENCE (SIZE (0 .. 4)) OF
GEOReceiverMeasurements,
       signalQualityOfTrackedSV         SEQUENCE (SIZE (0 .. 28)) OF
SVSignalQuality,
       delta1ppsMeasurementValidity            ReceiverDelta1ppsMeasurementValidity,
       delta1ppsMeasurement                    ReceiverDelta1ppsMeasurement}

--=====================================================================--
-- Validity field indicating the validity of the measurement:          --
--          RIMS Delta 1pps Measurement.                               --
-- It is valid if the value of this field is set to TRUE.              --
--=====================================================================--
ReceiverDelta1ppsMeasurementValidity ::= Valid

--=====================================================================--
-- Measurement of the difference between the 1pps of RIMS-PAR receiver  --
-- and 1pps of UTC(OP) clock and:                                      --
--          T_1PPS_CAL - T_1PPS_UTC(OP).                               --
--                                                                     --
-- This difference is used to compute the ENT-UTC offset within CPF.   --
-- Min Value: -200000x10^-11 = -2E-6 seconds                           --
-- Max Value:  200000x10^-11 =  2E-6 seconds                           --
--=====================================================================--
ReceiverDelta1ppsMeasurement ::= SecondPPS
SecondPPS ::=REAL (WITH COMPONENTS {    mantissa (-200000..200000),
                                        base (10),
                                        exponent (-11) })


GPSReceiverMeasurements ::= SEQUENCE {
       gpsPRN                           SV-PRN,
       gpsL1SignalStatus                ReceiverSignalStatus,
       gpsL2SignalStatus                ReceiverSignalStatus,
       gpsDataValidity                  RawMeasurementsValidity,
       gpsPrimaryPseudorange            PseudorangeAlternative,
       gpsL1AccumulatedDoppler          ReceiverAccumulatedDoppler,
       gpsL1CN0                         ReceiverChannelCN0,
       gpsL1CCC                         ReceiverCCC,
       gpsL2-L1Pseudorange              SVDeltaPseudoRange,
       gpsL2AccumulatedDoppler          ReceiverAccumulatedDoppler,
       gpsL2CN0                         ReceiverChannelCN0,
```

```
      gpsL2CCC                         ReceiverCCC,
      gpsRawData                       SVNavigationMessage}

--=====================================================================--
-- The value "other" of the Correlator spacing shall be used for DDC and--
-- MEDLL.                                                              --
-- Note that spare value is equivalent to undefined.                   --
--=====================================================================--
ReceiverSignalStatus ::= SEQUENCE {
      channelAssignmentMode            ENUMERATED {forced, auto},
      trackingState                    ENUMERATED {acquisition, re-acquisition,
codeLockedLoop, codeAndPhaseLockedLoops},
      satelliteAssignment              ENUMERATED {forced, auto},
      correlatorSpacing                ENUMERATED {standard, narrowCorrelator,
other, spare }}

RawMeasurementsValidity   ::= SEQUENCE {
      primaryPseudorangeValidity       Valid,
      l1DataValidity                   Valid,
      l2DataValidity                   Valid,
      navigationMessageValidity        Valid}

PseudorangeAlternative ::= CHOICE {
      l1Pseudorange                    SVBiasedPseudoRange,
      l2Pseudorange                    SVBiasedPseudoRange}

--=====================================================================--
-- Pseudo Range of a Navigation satellite (GPS, GEO or GLONASS).       --
-- Min Value: -549755813888x2^-7 = -4 294 967 296 meter                --
-- Max Value:  549755813887x2^-7 =  4 294 967 295,992 187 5 meter      --
--=====================================================================--
SVBiasedPseudoRange ::= PseudorangeMeter
PseudorangeMeter ::= REAL (WITH COMPONENTS {  mantissa (-
549755813888..549755813887),
                                              base (2),
                                              exponent (-7) })

--=====================================================================--
-- The accumulated Doppler is set to zero at signal acquisition and    --
-- counts continuously as long as stable signal tracking is performed. --
-- Unstable or scrambled tracking leads to cycle-slips, i.e. jumps in  --
-- counter values not accounted for by changes of geometry, clock drifts--
-- or changes in signal transmission delays. It has to be ensured that --
-- no induced cycle-slips occur at counter over- or underflow.         --
-- This information will roll-over every hour, about.                  --
-- Min Value: -2147483648x2^-9 = -4 194 304 cycles                     --
-- Max Value:  2147483647x2^-9 =  4 194 303,998 046 875 cycles         --
--=====================================================================--
ReceiverAccumulatedDoppler ::= Cycles
Cycles ::= REAL (WITH COMPONENTS {     mantissa (-2147483648..2147483647),
                                       base (2),
                                       exponent (-9) })

--=====================================================================--
-- Min Value:  0x2^-1 = 0 dBHz                                         --
-- Max Value:  120x2^-1 =  60 dBHz                                     --
--=====================================================================--
ReceiverChannelCN0 ::= DBHZ
DBHZ ::= REAL (WITH COMPONENTS { mantissa (0..120),
                                 base (2),
                                 exponent (-1) })

--=====================================================================--
-- Min Value: -128x2^-1 = -64   dBHz                                   --
```

```
-- Max Value:  127x2^-1 =  63.5 dBHz                                   --
--=====================================================================--
ReceiverCCC ::= CCCMeter
CCCMeter ::= REAL (WITH COMPONENTS {   mantissa (-128..127),
                                       base (2),
                                       exponent (-1) })

--=====================================================================--
-- L2 pseudorange - L1 pseudorange of a navigation satellite.          --
-- The coding capacity is -156 Meters to +355.992 Meters.              --
-- Min Value: -19968x2^-7 = -156 meter                                 --
-- Max Value:  45567x2^-7 =  355,9921875 meter                         --
--=====================================================================--
SVDeltaPseudoRange ::= DeltaPseudoRangeMeter
DeltaPseudoRangeMeter ::= REAL (WITH COMPONENTS {   mantissa (-19968..45567),
                                                    base (2),
                                                    exponent (-7) })

SVNavigationMessage ::= BIT STRING (SIZE (50))

GLONASSReceiverMeasurements ::= SEQUENCE {
      glonassSlotNumber                SV-PRN,
      glonassL1SignalStatus            ReceiverSignalStatus,
      glonassDataValidity              RawMeasurementsValidity,
      glonassPrimaryPseudorange        SVBiasedPseudoRange,
      glonassL1AccumulatedDoppler      ReceiverAccumulatedDoppler,
      glonassL1CN0                     ReceiverChannelCN0,
      glonassL1CCC                     ReceiverCCC,
      glonassRawData                   SVNavigationMessage}


GEOReceiverMeasurements ::= SEQUENCE {
      geoSlotNumber                    SV-PRN,
      geoL1SignalStatus                ReceiverSignalStatus,
      geoDataValidity                  RawMeasurementsValidity,
      geoPrimaryPseudorange            SVBiasedPseudoRange,
      geoL1AccumulatedDoppler          ReceiverAccumulatedDoppler,
      geoL1CN0                         ReceiverChannelCN0,
      geoL1CCC                         ReceiverCCC,
      geoRawData                       GeoNavigationMessage}


SVSignalQuality ::= SEQUENCE {
      svNumber                         SV-PRN,
      l1SignalQuality                  ReceiverSignalQuality,
      l2SignalQuality                  ReceiverSignalQuality}

--=====================================================================--
-- If the receiver detects something wrong but is not in a position to  --
-- isolate the origin of the failure or if failure is due to internal   --
-- failure (HW, calibration process unsuccessful), the "signal validity"--
-- flag must be set to unhealthy.                                       --
-- If jammer/multipath/cycle slip is detected, the " signal validity "  --
-- flag shall be set to unhealthy or invalid.                           --
--=====================================================================--
ReceiverSignalQuality ::= SEQUENCE {
      signalValidity                   ENUMERATED {valid, unhealthy, unvalid,
spare},
      jammer                           ENUMERATED {nothingDetected, detected},
      multipath                        ENUMERATED {nothingDetected, detected},
      cycleSlip                        ENUMERATED {nothingDetected, detected,
repaired, spare},
      interfrequencyBiasCompensation   ENUMERATED {activated, inactivated,
unsuccessful, spare}}
```

```
END
```

5.3 MODULE EDAS-EGNOSNAVIGATIONCORRECTION

```
EDAS-EGNOSNavigationCorrection DEFINITIONS AUTOMATIC TAGS ::=
BEGIN

IMPORTS
	EGNOSProductHeader,
	Valid,
	GeoNavigationMessage FROM EDAS-Common;

EGNOSNavigationCorrectionMessage ::= SEQUENCE {
	productHeader			EGNOSProductHeader,
	navigationCorrectionMessageData		NavigationCorrectionMessage}

NavigationCorrectionMessage ::= SEQUENCE {
	navigationCorrectionMessageValidity	CorrectionMessageValidity,
	geoMessageSelectedForUplink		GeoNavigationMessage,
	geoUplinkedPreviousCycleN-1		GeoNavigationMessage,
	geoUplinkedCycleN-2		GeoNavigationMessage,
	geoUplinkedCycleN-3		GeoNavigationMessage}



CorrectionMessageValidity ::= SEQUENCE {
	geoMessageSelectedForUplink-Status	Valid,
	geoUplinkedPreviousCycleN-1-Status	Valid,
	geoUplinkedCycleN-2-Status	Valid,
	geoUplinkedCycleN-3-Status	Valid}

END
```

6 ANNEX B:ASN.1 MESSAGE FIELDS

6.1 RIMS RAW MEASUREMENTS MESSAGE

	Field (section or data)	Description
1 section per message	Origin address	Address of the EGNOS Element.
1 section per message	Message Time tag	Time tag placed by originating element corresponding to the time of emission of this message toward INSPIRE.
Section repeated from 0 to 12 times according to the number of GPS SV tracked by the EGNOS Receiver	GPS PRN	PRN of the GPS satellite
	GPS L1 Signal Status	Status on GPS L1 channel and signal
	GPS L2 Signal Status	Status on GPS L2 channel and signal
	GPS data validity	Word of bits used to indicate validity of following fields in GPS raw data
	GPS Primary Pseudorange	Primary pseudorange measurement: L1 if L1 is valid L2 if L1 is not valid and L2 is valid
	GPS L1 Accumulated Doppler	GPS Carrier phase measurement on L1 frequency
	GPS L1 C/N0	C/N0 of the L1 receiver frequency.
	GPS L1 CCC	Carrier/Code_Coherency ratio corresponding to the GPS satellite reception by the Receiver on L1 frequency.
	GPS L2-L1 Pseudorange	L2-L1 value if both L2 and L1 measurements are valid
	GPS L2 Accumulated Doppler	GPS carrier phase measurement on L2 frequency
	GPS L2 C/N0	C/N0 of the L2 receiver signal
	GPS L2 CCC	Carrier/Code_Coherency ratio corresponding to the GPS satellite reception by the Receiver on L2 frequency.
	GPS 50 bits message	Navigation message received each second from one GPS satellite.
Section repeated from 0 to 12 times according to the number of GLONASS SV tracked by the EGNOS Receiver	GLONASS Slot number	Slot number of the GLONASS satellite
	GLONASS L1 Signal Status	Status on GLONASS L1 channel and signal
	GLONASS data validity	Word of bits used to indicate validity of following fields in GLONASS raw data
	GLONASS Primary Pseudorange	Primary pseudorange measurement: L1 if L1 is valid
	GLONASS L1 Accumulated Doppler	GLONASS Carrier phase measurement on L1 frequency
	GLONASS L1 C/N0	C/N0 of the L1 receiver frequency.
	GLONASS L1 CCC	Carrier/Code_Coherency ratio corresponding to the GLONASS satellite reception by the Receiver on L1 frequency.

	Field (section or data)	Description
	GLONASS 50bits message	Navigation message received each second from one satellite.
Section repeated from 0 to 4 times according to the number of GEO SV tracked by the EGNOS Receiver	GEO Slot number	Slot number of the GEO satellite
	GEO L1 Signal Status	Status on GEO L1 channel and signal
	GEO data validity	Word of bits used to indicate validity of following fields in GEO raw data
	GEO Primary Pseudorange	Primary pseudorange measurement: L1 if L1 is valid
	GEO L1 Accumulated Doppler	GEO Carrier phase measurement on L1 frequency
	GEO L1 C/N0	C/N0 of the L1 receiver frequency.
	GEO L1 CCC	Carrier/Code_Coherency ratio corresponding to the GEO satellite reception by the Receiver on L1 frequency.
	GEO 250bits message	Navigation message received each second from one satellite.
1 section per message	delta 1pps measurement validity	– Validity bit of the delta 1pps measurement for computing the difference between ENT and UTC. – Only the RIMS A channel located in Paris will fill this field by the appropriate value (VALID or NOT valid) depending of the current status of delta 1pps measurement All the other RIMS channels will fill this field with "NOT VALID" value
	delta 1pps measurement	delta 1pps measurement for computing the difference between ENT and UTC.
Section repeated from 0 to 28 times according to the number of GPS and GLONASS SV tracked by the EGNOS Receiver	SV PRN	PRN of the tracked satellite
	L1 Signal Quality	Indicator bits of SV L1 Signal Quality as observed at RIMS channel level.
	L2 Signal Quality	Indicator bits of SV L2 Signal Quality as observed at RIMS channel level.

6.2 NLES CYCLIC FEEDBACK MESSAGE

Those are the fields that are sent in SL0 ASN.1 format. A little description of the meaning is added for each field.

Field (section or data)	Description
Origin address	Address of the EGNOS Element.
Message Time tag	Time tag placed by originating element corresponding to the time of emission of this message toward INSPIRE.

Field (section or data)	Description
NLES Feed-back validity bits	Validity associated with the present submessage. It indicates the presence of each of the 4 possible previous messages.
GEO message selected for Uplink	Feed back of the GEO message selected for Uplink in this cycle (N). If the associated validity bit is not set within the "NLES Feed-back validity bits" field, this message is be fed-back as all binary zeros.
GEO Uplinked previous cycle (N-1)	Feed back of the GEO message Uplinked on the previous cycle (N-1). If the associated validity bit is not set, this message shall be fed-back as all binary zeros.
GEO Uplinked cycle (N-2)	Feed back of the GEO message Uplinked on cycle (N-2). If the associated validity bit is not set, this message shall be fed-back as all binary zeros.
GEO Uplinked cycle (N-3)	Feed back of the GEO message Uplinked on cycle (N-3). If the associated validity bit is not set, this message shall be fed-back as all binary zeros.

7 ANNEX C: CRC-32 COMPUTATION METHOD

7.1 DESCRIPTION

CRC-32 using the following polynom:

x**0+x**1+x**2+x**4+x**5+x**7+x**8+x**10+x**11+x**12+x**16+x**22+x**23

+x**26+x**32.

Software computation method for this polynom can be found in [RFC 1662].

The method is exposed in Section 6.2 only for informational purposes, as the CRC-32 check is done by the Client Software each time it receives a package, and consequently it must not be done by the service provider.

7.2 COMPUTATION METHOD (RFC 1662)

The following code provides a table lookup computation for calculating the 32-bit Frame Check Sequence as data arrives at the interface.

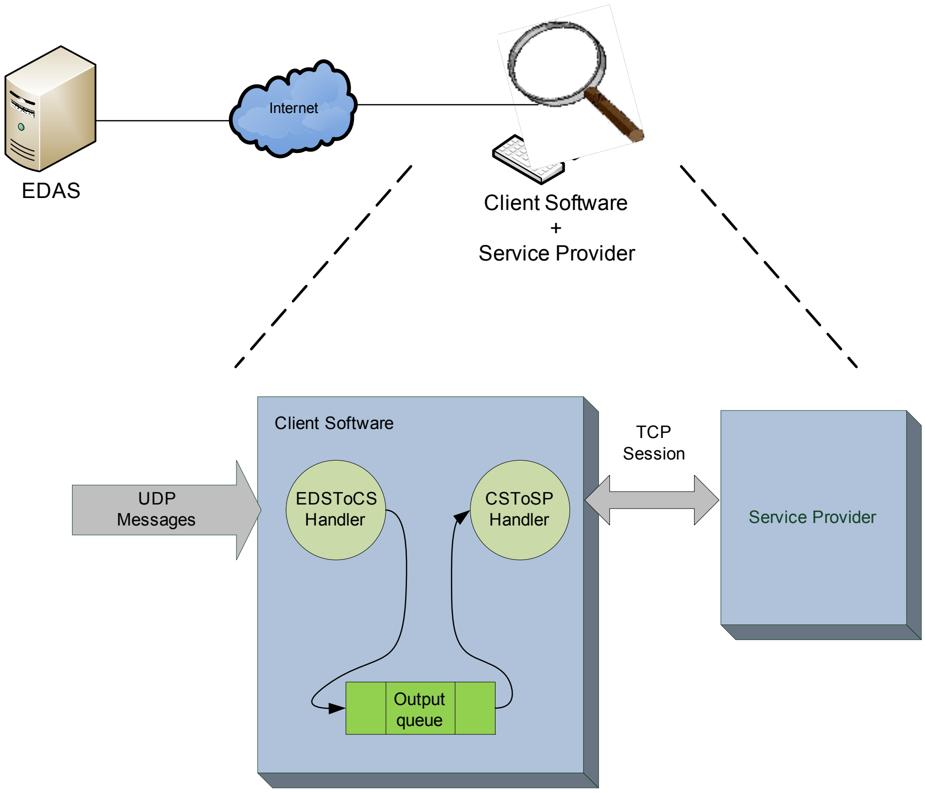
```
/*
 * The FCS-32 generator polynomial: x**0 + x**1 + x**2 + x**4 + x**5
 *                        + x**7 + x**8 + x**10 + x**11 + x**12 + x**16
 *                        + x**22 + x**23 + x**26 + x**32.
 */

/*
 * u32 represents an unsigned 32-bit number.  Adjust the typedef for
 * your hardware.
 */
typedef unsigned long u32;

static u32 fcstab_32[256] =
   {
   0x00000000, 0x77073096, 0xee0e612c, 0x990951ba,
   0x076dc419, 0x706af48f, 0xe963a535, 0x9e6495a3,
   0x0edb8832, 0x79dcb8a4, 0xe0d5e91e, 0x97d2d988,
   0x09b64c2b, 0x7eb17cbd, 0xe7b82d07, 0x90bf1d91,
   0x1db71064, 0x6ab020f2, 0xf3b97148, 0x84be41de,
   0x1adad47d, 0x6ddde4eb, 0xf4d4b551, 0x83d385c7,
   0x136c9856, 0x646ba8c0, 0xfd62f97a, 0x8a65c9ec,
   0x14015c4f, 0x63066cd9, 0xfa0f3d63, 0x8d080df5,
   0x3b6e20c8, 0x4c69105e, 0xd56041e4, 0xa2677172,
   0x3c03e4d1, 0x4b04d447, 0xd20d85fd, 0xa50ab56b,
   0x35b5a8fa, 0x42b2986c, 0xdbbbc9d6, 0xacbcf940,
   0x32d86ce3, 0x45df5c75, 0xdcd60dcf, 0xabd13d59,
   0x26d930ac, 0x51de003a, 0xc8d75180, 0xbfd06116,
   0x21b4f4b5, 0x56b3c423, 0xcfba9599, 0xb8bda50f,
   0x2802b89e, 0x5f058808, 0xc60cd9b2, 0xb10be924,
   0x2f6f7c87, 0x58684c11, 0xc1611dab, 0xb6662d3d,
   0x76dc4190, 0x01db7106, 0x98d220bc, 0xefd5102a,
```

```
    0x71b18589, 0x06b6b51f, 0x9fbfe4a5, 0xe8b8d433,
    0x7807c9a2, 0x0f00f934, 0x9609a88e, 0xe10e9818,
    0x7f6a0dbb, 0x086d3d2d, 0x91646c97, 0xe6635c01,
    0x6b6b51f4, 0x1c6c6162, 0x856530d8, 0xf262004e,
    0x6c0695ed, 0x1b01a57b, 0x8208f4c1, 0xf50fc457,
    0x65b0d9c6, 0x12b7e950, 0x8bbeb8ea, 0xfcb9887c,
    0x62dd1ddf, 0x15da2d49, 0x8cd37cf3, 0xfbd44c65,
    0x4db26158, 0x3ab551ce, 0xa3bc0074, 0xd4bb30e2,
    0x4adfa541, 0x3dd895d7, 0xa4d1c46d, 0xd3d6f4fb,
    0x4369e96a, 0x346ed9fc, 0xad678846, 0xda60b8d0,
    0x44042d73, 0x33031de5, 0xaa0a4c5f, 0xdd0d7cc9,
    0x5005713c, 0x270241aa, 0xbe0b1010, 0xc90c2086,
    0x5768b525, 0x206f85b3, 0xb966d409, 0xce61e49f,
    0x5edef90e, 0x29d9c998, 0xb0d09822, 0xc7d7a8b4,
    0x59b33d17, 0x2eb40d81, 0xb7bd5c3b, 0xc0ba6cad,
    0xedb88320, 0x9abfb3b6, 0x03b6e20c, 0x74b1d29a,
    0xead54739, 0x9dd277af, 0x04db2615, 0x73dc1683,
    0xe3630b12, 0x94643b84, 0x0d6d6a3e, 0x7a6a5aa8,
    0xe40ecf0b, 0x9309ff9d, 0x0a00ae27, 0x7d079eb1,
    0xf00f9344, 0x8708a3d2, 0x1e01f268, 0x6906c2fe,
    0xf762575d, 0x806567cb, 0x196c3671, 0x6e6b06e7,
    0xfed41b76, 0x89d32be0, 0x10da7a5a, 0x67dd4acc,
    0xf9b9df6f, 0x8ebeeff9, 0x17b7be43, 0x60b08ed5,
    0xd6d6a3e8, 0xa1d1937e, 0x38d8c2c4, 0x4fdff252,
    0xd1bb67f1, 0xa6bc5767, 0x3fb506dd, 0x48b2364b,
    0xd80d2bda, 0xaf0a1b4c, 0x36034af6, 0x41047a60,
    0xdf60efc3, 0xa867df55, 0x316e8eef, 0x4669be79,
    0xcb61b38c, 0xbc66831a, 0x256fd2a0, 0x5268e236,
    0xcc0c7795, 0xbb0b4703, 0x220216b9, 0x5505262f,
    0xc5ba3bbe, 0xb2bd0b28, 0x2bb45a92, 0x5cb36a04,
    0xc2d7ffa7, 0xb5d0cf31, 0x2cd99e8b, 0x5bdeae1d,
    0x9b64c2b0, 0xec63f226, 0x756aa39c, 0x026d930a,
    0x9c0906a9, 0xeb0e363f, 0x72076785, 0x05005713,
    0x95bf4a82, 0xe2b87a14, 0x7bb12bae, 0x0cb61b38,
    0x92d28e9b, 0xe5d5be0d, 0x7cdcefb7, 0x0bdbdf21,
    0x86d3d2d4, 0xf1d4e242, 0x68ddb3f8, 0x1fda836e,
    0x81be16cd, 0xf6b9265b, 0x6fb077e1, 0x18b74777,
    0x88085ae6, 0xff0f6a70, 0x66063bca, 0x11010b5c,
    0x8f659eff, 0xf862ae69, 0x616bffd3, 0x166ccf45,
    0xa00ae278, 0xd70dd2ee, 0x4e048354, 0x3903b3c2,
    0xa7672661, 0xd06016f7, 0x4969474d, 0x3e6e77db,
    0xaed16a4a, 0xd9d65adc, 0x40df0b66, 0x37d83bf0,
    0xa9bcae53, 0xdebb9ec5, 0x47b2cf7f, 0x30b5ffe9,
    0xbdbdf21c, 0xcabac28a, 0x53b39330, 0x24b4a3a6,
    0xbad03605, 0xcdd70693, 0x54de5729, 0x23d967bf,
    0xb3667a2e, 0xc4614ab8, 0x5d681b02, 0x2a6f2b94,
    0xb40bbe37, 0xc30c8ea1, 0x5a05df1b, 0x2d02ef8d
    };

#define PPPINITFCS32  0xffffffff   /* Initial FCS value */
#define PPPGOODFCS32  0xdebb20e3   /* Good final FCS value */
/*
 * Calculate a new FCS given the current FCS and the new data.
 */
u32 pppfcs32(fcs, cp, len)
    register u32 fcs;
    register unsigned char *cp;
    register int len;
    {
    ASSERT(sizeof (u32) == 4);
    ASSERT(((u32) -1) > 0);
    while (len--)
        fcs = (((fcs) >> 8) ^ fcstab_32[((fcs) ^ (*cp++)) & 0xff]);
```

```
    return (fcs);
    }

/*
 * How to use the fcs
 */
tryfcs32(cp, len)
    register unsigned char *cp;
    register int len;
{
    u32 trialfcs;

    /* add on output */
    trialfcs = pppfcs32( PPPINITFCS32, cp, len );
    trialfcs ^= 0xffffffff;             /* complement */
    cp[len] = (trialfcs & 0x00ff);      /* least significant byte first */
    cp[len+1] = ((trialfcs >>= 8) & 0x00ff);
    cp[len+2] = ((trialfcs >>= 8) & 0x00ff);
    cp[len+3] = ((trialfcs >> 8) & 0x00ff);

    /* check on input */
    trialfcs = pppfcs32( PPPINITFCS32, cp, len + 4 );
    if ( trialfcs == PPPGOODFCS32 )
        printf("Good FCS\n");
}
```

8 ANNEX D: ARCHITECTURE OF EDAS CLIENT SW

The figure below explains the internal architecture of the EDAS Client SW module. One handler is in charge of receiving data from EDAS (EDSToCS, through a UDP connection), whereas another one delivers them to the Service Provider application (CSToSP, through a TCP connection).



The Client SW was designed to ensure that all EDAS messages received are delivered to the user (i.e. the service provider application). The Client SW contains an internal queue to store the messages that will be sent to the Service Provider application (SP). Although data delivery to the SP is ensured, applications connected to EDAS have to be aware of this fact. Actually, if no application is connected to the output of the Client SW then the queue will get full quickly. Applications have to retrieve EDAS messages from the Client SW at an appropriate rate so as to avoid that the queue gets full. Once the queue fills up, EDAS messages received at the Client SW side are discarded.

Notice that the size of this queue is configurable. The configuration parameter max_queue_size contains the size (in Mb) of the queue in question. In order to prevent data gaps, it is recommended to set this value between 5 and 10 (Mb).

SP applications performing time-consuming transformations on EDAS data could minimize the impact of the issue described above by managing themselves the data received (e.g. they can implement different threads for receiving and processing data).

Annex 1
List of Acronyms

A	
A-GNSS	Assisted GNSS
APC	Antenna Phase Center
aPER	aligned PER (ASN.1 encoding)
ARP	Antenna Reference Point
ASN.1	Abstract Syntax Notation One
ATC	Air Traffic Control
B	
C	
CCF	Central Control Facility
CCM	Client Control Module
CMR	Compact Measurement Record
CPF	Central Processing Facility
CRC	Cyclic Redundancy Check
CNR	Carrier To Noise
CS	Client Software
D	
DAB	Digital Audio Broadcasting
DF	Data Field
DGNSS	Differential GNSS
E	
ECEF	Earth-Center-Earth-Coordinates
ECS	EGNOS Commercial Services
EDAS	EGNOS Data Access System
EDS	EGNOS Data Server
EGNOS	European Geostationary Navigation Overlay Service
EMS	EGNOS Message Server
ESA	European Space Agency
ESTB	EGNOS System Test-Bed
F	
FEE	Front-End Equipment
G	

GEO	Geostationary Earth Orbiter
GIVD	Grid Ionospheric Vertical Delay
GIVE	Grid Ionospheric Vertical Error
GJU	Galileo Joint Undertaking
GLONASS	GLObalnaya NAvigatsionnaya Sputnikovaya Sistema (Global Navigation Satellite System)
GNSS	Global Navigation Satellite System
GNU	GNU is Not Unix
GPS	Global Positioning System
H	
HMI	Hazardous Misleading Information
HTTP	HyperText Transfer Protocol
HW	Hardware
MMI	Man Machine Interface
I	
ICAO	International Civil Aviation Organization
ICD	Interface Control Document
IGS	International GNSS Service
IGS-RTWG	IGS Real Time Working Group (http://igscb.jpl.nasa.gov/projects/rtwg/index.html)
INSPIRE	INterface System for the Provision In Real-time of the Egnos product
IONEX	IONosphere map Exchange
IP	Internet Protocol
IPR	Intellectual Property Right
ITRF	International Terrestrial Reference Frame
J	
K	
KOM	Kick Off Meeting
L	
LSB	Least Significant Bit
M	
MCC	Master Control Centre
MSB	Most Significant Bit
MT	Message Type
MoM	Minutes of Meeting
MOPS	Minimum Operational Performance Standards
N	

NLES	Navigation Land Earth Station
NOF	Navigation Overlay Frame
NTRIP	Networked Transport of RTCM via Internet Protocol
O	
OSI	Open System Interconnection
P	
PER	Packed Encoding Rules (ASN.1 encoding)
PPP	Precise Point Positioning
ProDDAGE	Programme for Development and Demonstration of Applications for Galileo and EGNOS
PRN	Pseudo Random Noise
Q	
R	
RDS	Radio Data System
RIMS	Ranging and Integrity Monitoring Station
RINEX	Receiver Independent Exchange format
RTCA	Radio Technical Commission for Aeronautics
RTCM	Radio Technical Commission For Maritime Services
RTK	Real Time Kinematic
S	
SBAS	Satellite Based Augmentation System
SINEX	Solution INdependent EXchange Format
SIS	Signal In Space
SISNeT	Signal In Space through interNET
SOW	Statement Of Work
SW	Software
T	
TEC	Total Electron Content
TCP	Transmission Control Protocol
TOA	Time of Applicability
ToW	Time of Week
U	
UDP	User Datagram Protocol
UDRE	User Differential Range Error
V	

W	
WP	Work Package
WARTK	Wide Area Real Time Kinematics
WGS84	World Geodetic System 1984
X	
xPL	Vertical or Horizontal Protection Level
Y	
Z	