ACM Beleidsregel Handhaving Besluit Eindapparaten

Ziggo PSI/SI overzicht (voor lineaire digitale Radio & TV)

Versie 2.1 (2022-01)

Dit document is geschreven in de taal: Engels

(Blank page)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. REFERENCES

Normative reference is a term covering separate documents referenced within this document and means that, unless otherwise stated, the most recent versions of the separate documents should be referenced.

[1] ISO/IEC 13818-1:2018, "Generic coding of moving pictures and associated audio information – Part 1: Systems".

[2] ETSI EN 300 468 V1.11.1 (2010-04), "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".

[3] ETSI EN 300 472 V1.3.1 (2003-05), "Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bitstreams".

[4] ETSI EN 300 743 V1.3.1 (2006-11), "Digital Video Broadcasting (DVB); Subtitling systems".

[5] ETSI ETR 211 (1997-08), "Digital Video Broadcasting (DVB); Guidelines on implementation and usage of Service Information (SI)".

[6] ETSI TS 101 162 V1.7.1 (2014-02), "Digital Video Broadcasting (DVB); Allocation of identifiers and codes for Digital Video Broadcasting (DVB) systems".

[7] CI Plus Specification v1.3.1 (2011-09), "CI Plus Specification - Content Security Extensions to the Common Interface.".

[8] EICTA Ziggo V2.1, "EICTA Ziggo Digital Cable TV Receiver Specification V2.1 rev 1.4.3 (LGI phase 1 harmonised)".

[9] ISO 639-2:1998, "Codes for the representation of names of languages - Part 2: Alpha-3 code".

[10] ISO/IEC 8859-1:1998, "Information processing - 8-bit single-byte coded graphic character sets - Part 1: Latin alphabet No. 1".

[11] ETSI TS 102 006 V1.4.1 (2015-06), "Digital Video Broadcasting (DVB);. Specification for System Software Update in DVB Systems.".

[12] ISO/IEC 3166-1:2013, "Codes for the representation of names of countries and their subdivisions - Part 1: Country codes".

[13] LGI CTO, "UPC Genre definition".

[14] LGI CTO, "UPC CTO; D4A UCNP Age Ratings V1.3".

[15] ETSI TS 101 154 V1.9.1 (2009-09), "Digital Video Broadcasting (DVB); Specification for the use of Video and Audio Coding in Broadcasting Applications based on the MPEG-2 Transport Stream".

[16] ETSI EN 300 706 V1.2.1 (2003-04), "Enhanced Teletext Specification".

[17] ISO/IEC 6937:2001, "Information technology; Coded graphic character set for text communication - Latin alphabet".

[18] ISO/IEC 8859-9:1999, "Information technology; 8-bit single-byte coded graphic character sets - Part 9: Latin alphabet No. 5".

[20] OTV UCNP PSISI V19 (2011-10), "UPC-Corp Network Package; PSI/SI Specifications".

[21] ACM/17/019945 (2018-09), "Marktanalyse Wholesale Fixed Access".

[22] ETSI TS 102 809 V1.2.1 (2013-07), "Digital Video Broadcasting (DVB); Signalling and carriage of interactive applications and services in Hybrid broadcast/broadband environments".

2. SCOPE

This document describes the current (2021-08) MPEG PSI and DVB SI structure as used on the Ziggo cable network, further referred to as Ziggo network. Only the relevant tables and descriptors for a linear digital radio- and television services are documented. Proprietary descriptors for the 'interactive proposition' are not to be found in this document.

In time the MPEG PSI and DVB SI structure may change. In case relevant changes apply to the PSI/SI structure which are relevant to and/or applicable for the Manufacturer, an updated version of this document will be published.

The broadcast adheres to standards *ISO/IEC 13818-1* [1] and *ETSI EN 300 468* [2] with several Ziggo specific additions.

This document is organized as follows:

- Section 3 shows use cases detailing how PSI/SI information on the Ziggo network is to be used.
- Section 4 defines the PSI and SI tables as broadcasted by Ziggo, for both MPEG and DVB standard descriptors and tables and Ziggo specific additions and deviations from the standard.
- Section 5 explains the available descriptors for use by the Manufacturer in more detail, and shows where they are Ziggo proprietary.

It needs to be acknowledged that this documents specifically describes the PSI/SI available for the Manufacturer. Ziggo proprietary tables and private descriptors are either blacked-out or mentioned without syntax and descriptor formats.

In some cases the requirements are mandatory for passing the inter-operability tests and certification. The following phrases are used to identify what is mandatory or optional:

MUST	This word, or the terms "REQUIRED" or "SHALL", mean that the definition is an absolute requirement.
MUST NOT	This phrase, or the phrase "SHALL NOT", mean that the definition is an absolute prohibition of the specification.
SHOULD	This word, or the adjective "RECOMMENDED", mean that there may exist valid reasons in particular circumstances to ignore a particular item, but the full implications must be understood and carefully weighed before choosing a different course.
SHOULD NOT	This phrase, or the phrase "NOT RECOMMENDED" mean that there may exist valid reasons in particular circumstances when the particular behaviour is acceptable or even useful, but the full implications should be understood and the case carefully weighed before implementing any behaviour described with this label.
MAY	This word, or the adjective "OPTIONAL", mean that an item is truly optional.

When marked as Ziggo proprietary, the DVB-C receiver of the Manufacturer MUST ignore these tables and/or private descriptors. Other tables and descriptors will be marked “MUST” and “MAY”.

3. NETWORK USE CASES

This section contains use cases explaining how PSI/SI tables are used by DVB-C receivers on the Ziggo network.

3.1. NETWORK INSTALLATION

This use case describes how a DVB-C receiver can be installed on the Ziggo Network, and which PSI/SI tables are used in the process.

Different geographical regions on the Ziggo network may have only a subset of the Transport Streams available at the National Headend, as shown in figure 1. It is also possible that the Transport Streams are locally modulated with different parameters. Network information is stored in the service information (SI) in the Network Information Table (NIT) actual and others. To be able to use one headend across the whole network, a device is installed using the NIT other tables instead of the NIT actual table. One central headend creates a NIT other table for each and every region. While the NIT actual is not used by the DVB-C receivers, it is generated for DVB compliancy, and it describes the original network with all Transport Streams.

During installation the user inputs his geographical location in the form of a network ID. The network ID is used to find the NIT other table in the broadcast. This Network Information Table holds information on all Transport Streams available in the region (described in chapter 4.4). Each Transport Stream contains a reference to all services in the stream in the form of a Service Description Table (SDT). For each service that must be available to DVB-C receiver user the SDT contains a logical channel number descriptor (described in chapter 4.8).

NOTE: services may be without logical channel number descriptor and can be used by the DVB-C receiver itself instead of the end-user.

A service may be different per region because of advertising, regional news variations or because a channel is only broadcasted in one region. Next to this, each region within the network can have a different frequency grid on the local HFC network. This has an impact on the cable delivery system descriptor in the NIT other for that region, which specifies the Transport Stream modulation parameters (described in chapter 5.2.2).

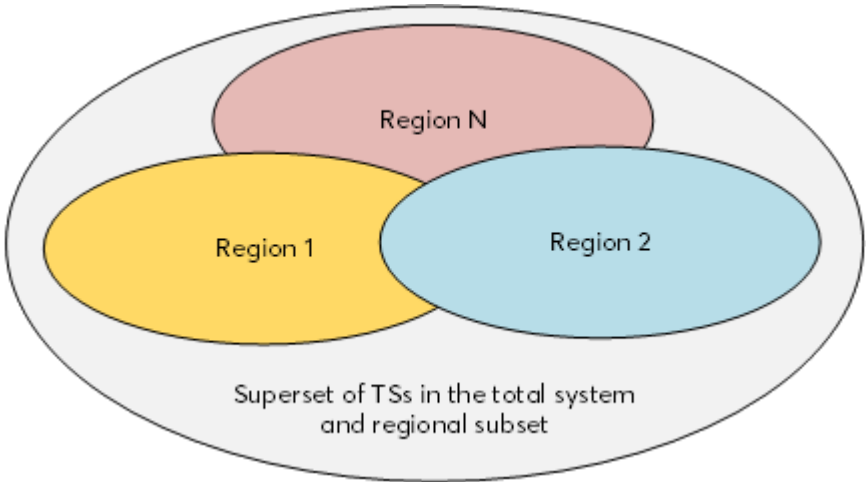

Figure 1: Graphical representation of Transport Streams in different regions

Use Case: Installation in a region

The Ziggo network is a combination of the former Ziggo network (before 2013) and the former UPC network. For backward compatibility reasons the DVB SI information has differences on the "homing" frequencies and some SI-tables. DVB-C receivers are installed on the Ziggo Network in the following way.

Network/tuning parameters		fZiggo values	fUPC values
Network ID (NID)		5555	5555
Home TS frequency	[kHz]	474000	474000
Symbol rate	[kS/s]	6875	6900
Modulation profile		QAM64	QAM64

Table 1: Network and tuning parameters for user installation

End-user determines the appropriate installation parameters, connects the DVB-C receiver to the Ziggo Network and powers it on.

1. User inputs a Network ID (see table 1),
2. User inputs a frequency of the Home TS (see table 1),
3. User inputs a symbol rate (see table 1),
4. User inputs a modulation profile (see table 1),
5. The DVB-C receiver tunes to the local Home Transport Stream and finds the NIT other table corresponding to the network ID,
6. The NIT other contains information on all Transport Streams available in the region (see cable delivery system descriptor in chapter 5.2.2), and these TS parameters are stored on the DVB-C receiver.

NOTE: It is strongly advised that the DVB-C receiver is autosensing on Symbol Rate and Modulation Profile for aiding the end-user in the installation process

3.2. ZIGGO SERVICE LIST CREATION

After the network installation, the service lists are created. These lists are defined using the applicable Network ID (NID) and Service Description Table (SDT). One list contains all TV services that the user is allowed to use in the municipality, which may be a subset of the real available services. The second list contains all radio services that the user may use. A third list is created with all services that have a logical channel number that is not valid for the city of installation. These services are called out-of-list services, and may for instance be downloadable applications that are accessible from a menu or other applications. Besides these service lists there may also be services that are not announced using a logical channel number at all but by a linkage descriptor (described in chapter 5.2.11). These will not be stored in the service lists.

The Ziggo Network is divided in different regions and per region a subset of Transport Streams is available. In addition to this configuration the Ziggo DVB-C receiver can build up a service list that depends on the location within a region. So even though the available Transport Streams in a region are the same for all cities in the same region, the service list can be customized per municipality for local community broadcasters. This includes the addition of the service to the service list, and also the logical channel number the service is stored with on the DVB-C receiver.

During the installation process, the user has given additional information on the location of the DVB-C receiver: the Network ID. The DVB-C receiver now knows two levels of geographical information: country and region (defined by NIT other). Geographical boundaries do not necessarily match the network topology (given by the NIT). Each TS has a Program Association Table, containing a list of program numbers, one program number per service. The Service Description Table defines for each service a list of service descriptors including the channel number descriptor (described in paragraph 5.2.7).

The service list is created by the DVB-C receiver by reading the channel number descriptor found in the NIT. The descriptor defines logical channel numbers for the service depending on the three geographical levels. If multiple channel numbers are defined in the descriptor, the DVB-C receiver chooses the last logical channel number structure that fits the device location up to the lowest geographical level.

Use Case: Service list creation
Ziggo DVB-C receivers create the service list in the following way.

1. During the network installation (previous use case) the DVB-C receiver discovers the NIT other table to use and the Program Association Table that contains references to all services (program_number).

2. The DVB-C receiver finds the service description for all relevant services in the Network Information Table. This includes the EACEM logical_channel_descriptor which may define a logical channel number.
3. If a logical_channel_descriptor is available with a channel number structure that is valid for the installation location, the service is added to the service list with the given logical channel number from the structure (described in chapter 5.2.7).
4. If no logical_channel_descriptor is available, or no data matching the location is given in the descriptor, the service is placed in a different list, and the service is dubbed out-of-list. This service will not be directly available to a user, but may be used in other ways (i.e. interactive applications, video streams used in applications).

4. TABLE DEFINITIONS

In this section all tables are defined, first for the MPEG program specific information and subsequently for the DVB service information. It also lists all descriptors used in the tables.

On the Ziggo Network the following actual and other tables are broadcasted, with Packet ID and Table ID as shown. Detailed descriptions of these tables can be found in the chapters.

Packet ID	Table ID	Table
0x00	0x00	PAT
0x01	0x01	CAT
(see note 1)	0x02	PMT
0x10	0x40	NIT actual
	0x41	NIT other
0x11	0x42	SDT actual
	0x46	SDT other
	0x4A	BAT
0x12	0x4E	EIT-P/F actual
	0x4F	EIT-P/F other
	0x50	EIT-S actual
	0x60	EIT-S other
0x14	0x07	TDT
	0x73	TOT

NOTE 1: Multiple PMT tables may be available and will be referenced in the PAT.

NOTE 2: The table above is not valid for video on demand Transport Streams. The VOD TSs do not contain CAT, NIT, SDT, EIT-P/F or EIT-S tables.

Table 2: Actual and other tables as broadcasted in the Ziggo network

4.1. PAT - PROGRAM ASSOCIATION TABLE

The PAT contains the Packet ID (PID) of the PMT for each program accessible on the current Transport Stream.

4.1.1. PAT table structure

The structure of the PAT is given in table 3.

PAT: Program Association Table	Nr. of bits	Identifier
program_association_section() {		
table_id *0x00*	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i=0; i < N; i++) {		
program_number	16	uimsbf
reserved	3	bslbf
if (program_number == '0') {		
network_PID *//not used*	13	uimsbf
}		
else {		
program_map_PID	13	uimsbf
}		
}		
CRC_32	32	rpchof
}		

Table 3: Structure of the Program Association Table

Semantics for the program association section:

transport_stream_id

This 16 bit field serves as a label to identify this Transport Stream from any other multiplex within a network.

program_number

This 16 bit field specifies the program to which the program_map_PID is applicable. This field shall not take any single value more than once within one version of the program association table. The program_number may be used as a designation for a broadcast channel, for example.

network_PID

This value is not used by Ziggo.

program_map_PID

This 13 bit field specifies the PID of the Transport Stream packets which shall contain the program_map_section applicable for the program as specified by the program_number. No program_number shall have more than one program_map_PID assignment.

4.2. CAT - CONDITIONAL ACCESS TABLE

The CAT is broadcasted on all 'broadcast' Transport Streams. No CAT is transmitted on VOD Transport Streams. The Ziggo Conditional Access (CA) Table provides the association between Nagra and Irdeto CA systems, their EMM streams and any special parameters associated with it.

4.2.1. CAT table structure

The structure of the CAT is given in table 4.

CAT: Conditional Access Table	Nr. of bits	Identifier
CA_section() {		
table_id *0x01*	8	uimsbf
section_syntax_indicator	1	bslbf
'0'	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
reserved	18	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

Table 4: Structure of the Conditional Access Table

4.2.2. CAT descriptors

Ziggo uses one descriptor in the CAT:

CA_descriptor (0x09)
The conditional access descriptor is used to specify where to find system wide conditional access management information such as EMMs.

4.3. PMT - PROGRAM MAP TABLE

The PMT lists all elementary streams of a program. One PMT is broadcasted per program.

4.3.1. PMT table structure

The structure of the PMT is given in table 5.

PMT: Program Map Table	Nr. of bits	Identifier
TS_program_map_section() {		
table_id *0x02*	8	uimsbf
section_syntax_indicator '1'	1	bslbf
'0'	1	bslbf
Reserved	2	bslbf
section_length	12	uimsbf
program_number	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved	3	bslbf
PCR_PID	13	uimsbf
reserved	4	bslbf
program_info_length	12	uimsbf
for (i=0; i<N; i++) {		
if(scrambled) {		
CA_descriptor()		
}		
}		
for (i=0; i<N; i++) {		
stream_type	8	uimsbf
reserved	3	bslbf
elementary_PID	13	uimsnf
reserved	4	bslbf
ES_info_length	12	uimsbf
for (j=0; j<N; j++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Table 5: Structure of the Program Map Table

Semantics for the program map section:

program_number

This 16 bit field specifies the program to which the program_map_PID is applicable. This program_number is the same as the service_id in the corresponding service_description_section, as shown in chapter 4.8.

program_info_length

This 12 bit field specifies the number of bytes used by the descriptors immediately following the program_info_length field. The first two bits shall be '00'

4.3.2. PMT descriptors

Ziggo uses the following descriptors in the PMT:

CA_descriptor (0x09)

The conditional access descriptor is used to specify the elementary stream specific information, such as ECM and EMM.

NOTE: It is generic that the CA rights are used at the service level only and not at the elementary stream level. This descriptor can appear in both PMT loops.

ISO_639_language_descriptor (0x0A)

The language descriptor specifies the language of the associated program element. This descriptor is defined in chapter 5.2.9.

maximum_bitrate_descriptor (0x0E)

The maximum bitrate descriptor indicates an upper bound of the bitrate, including transport overhead that will be encountered in this program element or program.

private_data_indicator_descriptor (0x0F) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

stream_identifier_descriptor (0x52)

The stream identifier descriptor enables specific streams to be associated with information in SI tables via the component_tag field. This field is for example present in the component descriptor and data broadcast descriptor.

teletext_descriptor (0x56)

The teletext descriptor identifies EBU Teletext data which is coded in accordance with *ETSI EN 300 472* [3]. This descriptor is defined in chapter 5.2.19.

subtitling_descriptor (0x59)

The subtitling descriptor identifies DVB subtitle data which is coded in accordance with *ETSI EN 300 743* [4]. This descriptor is defined in chapter 5.2.18.

data_broadcast_id_descriptor (0x66)

The data broadcast id descriptor identifies data broadcast services within the SI (SDT and/or EIT) in the DVB framework.

AC-3_descriptor (0x6A)

The AC-3 descriptor identifies streams which carry AC-3 audio. This descriptor is defined in chapter 5.2.1.

application_signaling_descriptor (0x6F)

The application signaling descriptor is defined for use in the elementary stream loop of the PMT where the stream_type of the elementary stream is 0x05. It identifies that the elementary stream carries an Application Information Table.

[private descriptor] (0x90) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xFD) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xFE) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

4.4. NIT - NETWORK INFORMATION TABLE

All NIT tables are broadcasted on all 'broadcast' Transport Streams of the Ziggo Network. The NIT to be used on an IRD is chosen on a regional basis, instead of using the NIT actual. The way NITs are populated and broadcasted is compliant to *ETR 211* [5]. Also see the Installation Use Case in chapter 3.1.

NOTE: It needs to be acknowledged that Ziggo replaces the NIT tables from the National Headend (PID 0x10) in every regional headend with localized versions of the NIT. This enables the use of a single NID for all end-users despite the differences in RF frequency allocation per region.

EXAMPLE: NID 5555 can be used by every end-user, independent of the geographical location within the Ziggo network. From the end-user perspective the installation is transparent, although the cable_delivery_descriptors of the NIT_other differ per region.

4.4.1. NIT table structure

The NIT structure is shown in table 6.

NIT: Network Information Table	Nr. of bits	Identifier
network_information_section() {		
table_id *0x41 and 0x40 // other/actual*	8	uimsbf
section_syntax_indicator '1'	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
network_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
network_descriptors_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for (i=0; i<N; i++) {		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for (j=0; j<N; j++) {		
cable_delivery_system_descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Table 6: Structure of the Network Information Table

Semantics for the network information section:

network_id

This 16 bit field serves as a label to identify the delivery system, about which the NIT informs. Ziggo DVB-C receivers select a NIT_other by network_id, based on geographical information. The NIT_actual always has the UPC original network id 1536.

transport_stream_id

This 16 bit field serves as a label for identification of this TS from any other multiplex within the delivery system.

original_network_id

This 16 bit field gives the label identifying the network_id of the originating delivery system.

4.4.2. NIT descriptors

The following descriptors may appear in the first loop:

network_name_descriptor (0x40)

The network name descriptor provides the network name in text form

linkage_descriptor (0x4A)

The linkage descriptor is used to locate some services which do not appear in any service list such as EPG, Portal applications, menu definition, etc.

private_data_specifier_descriptor (0x5F)

The private data specifier descriptor is inserted before any private descriptors in the NIT and is needed to introduce Ziggo private descriptors.

[private descriptor] (0x84) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x85) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x87) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x89) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x91) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x92) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x93) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x95) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x97) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x99) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x9A) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xD2) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

The following descriptors may appear in the 2nd loop, delivery_system_descriptor():

service_list_descriptor (0x41)

The service list descriptor is used to list the services and their types for each TS. The services are listed identified by service_id, which is the same as the program_number in the corresponding program_map_section (PMT). This descriptor is defined in chapter 5.2.16.

NOTE: This service list descriptor is ignored by Ziggo DVB-C receivers. Service types are derived from the service descriptor within the SDT. Descriptor is still carried in Ziggo networks for backward compatibility.

cable_delivery_system_descriptor (0x44)

The cable delivery system descriptor specifies the physical tuning information of the associated Transport Stream. This descriptor is defined in chapter 5.2.3.

EACEM_logical_channel_descriptor (0x83)

The EACEM logical channel descriptor provides a default channel number label for services. This descriptor is defined in chapter 5.2.7.

4.5. BAT – BOUQUET ASSOCIATION TABLE

The BAT provides information regarding bouquets. A bouquet is a collection of services, which may traverse the boundary of a network.

4.5.1. BAT table structure

The structure of the BAT is given in table 7.

BAT: Bouquet Association Table	Nr. of bits	Identifier
bouquet_association_section() {		
table_id *0x4A*	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
bouquet_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
reserved_future_use	4	bslbf
bouquet_descriptors_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
reserved_future_use	4	bslbf
transport_stream_loop_length	12	uimsbf
for (i=0; i<N; i++) {		
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
reserved_future_use	4	bslbf
transport_descriptors_length	12	uimsbf
for (j=0; j<N; j++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Table 7: Structure of the Bouquet Association Table

Semantics for the bouquet association section:

bouquet_id

This is a 16 bit field which serves as a label to identify the bouquet. Allocations of the value of this field are found in ETSI TS 101 162 [6]. The bouquet_ids used by Ziggo are listed in table 8.

Bouquet ID	Bouquet Name
0x0600	Ziggo Announcement Bouquet
0x3622	Master bouquet Irdeto
0x660A	HORIZON

Table 8: Ziggo bouquet_id values

transport_stream_id

This is a 16 bit field which serves as a label for identification of this TS from any other multiplex within the delivery system.

original_network_id

This is a 16 bit field identifying the network_id of the originating delivery system.

4.5.2. BAT descriptors

Bouquet_name_descriptor (0x47)

The bouquet name descriptor provides the bouquet name in text form.

linkage_descriptor (0x4A) Ziggo proprietary

All linkage descriptors within the BAT section MUST be ignored by the DVB-C receiver of the Manufacturer.

private_data_specifier_descriptor (0x5F)

The private data specifier descriptor is inserted before any private descriptors in the SDT and is needed to introduce Ziggo private descriptors.

[private descriptors] (0x80 to 0xFE) Ziggo proprietary

All private descriptors within the BAT section MUST be ignored by the DVB-C receiver of the Manufacturer.

4.6. TDT - TIME AND DATE TABLE

The TDT carries only the UTC-time and date information.

4.6.1. TDT table structure

The structure of the TDT is shown in table 9.

TDT: Time Date Table	Nr. of bits	Identifier
time_offset_section() {		
table_id *0x70*	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
UTC_time	40	bslbf
}		

Table 9: Structure of the Time Date Table

Semantics for the time offset section:

UTC_time

This 40 bit field contains the current time and date in UTC and MJD. This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 time digits in 4 bit BCD.

EXAMPLE: 93/10/13 12:45:00 is coded as "0xC079124500".

4.7. TOT - TIME OFFSET TABLE

The TOT carries the UTC-time and date information and local time offset.

4.7.1. TOT table structure

The structure of the TOT is shown in table 10.

TOT: Time Offset Table	Nr. of bits	Identifier
time_offset_section() {		
table_id *0x73*	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
UTC_time	40	bslbf
reserved	4	bslbf
descriptors_loop_length	12	uimsbf
for (i=0; i<N; i++) {		
descriptor()		
}		
CRC_32	32	rpchof
}		

Table 10: Structure of the Time Offset Table

Semantics for the time offset section:

UTC_time

This 40 bit field contains the current time and date in UTC and MJD. This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 time digits in 4 bit BCD.

EXAMPLE: 93/10/13 12:45:00 is coded as "0xC079124500".

descriptors_loop_length

This 12 bit field gives the total length in bytes of the following descriptors.

4.7.2. TOT descriptors

local_time_offset_descriptor (0x58)

The local time offset descriptor describes country specific dynamic changes of the local time offset relative to UTC (described in chapter 5.2.12).

4.8. SDT - SERVICE DESCRIPTION TABLE

The SDT is defined according to *ETSI EN 300 468* [2]. SDT actual and others are broadcasted according to *ETSI ETR 211* [5]. SDT actual and others are broadcasted on all Transport Streams except VOD Transport Streams.

4.8.1. SDT table structure

The SDT structure is shown in table 11.

SDT: Service Description Table	Nr. of bits	Identifier
service_description_section() {		
table_id *0x42*	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
transport_stream_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
original_network_id	16	uimsbf
reserved_future_use	8	bslbf
for (i=0; i<N; i++) {		
service_id	16	uimsbf
reserved_future_use	6	bslbf
EIT_schedule_flag	1	bslbf
EIT_present_following_flag	1	bslbf
running_status	3	uimsbf
free_CA_mode	1	bslbf
descriptors_loop_length	12	uimsbf
for (j=0; j<N; j++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Table 11: Structure of the Service Description Table

transport_stream_id

This 16 bit field serves as a label for identification of this TS from any other multiplex within the delivery system.

original_network_id

This 16 bit field gives the label identifying the network_id of the originating delivery system.

service_id

This 16 bit field serves as a label to identify this service from any other service within the TS. The service_id is the same as the program_number in the corresponding program_map_section (PMT).

running_status

This 3 bit field indicates the status of the service. Only used by Irdeto based Ziggo DVB-C receivers.

free_CA_mode

This 1 bit field when set to '0' indicates that all components are unscrambled. When set to '1' access to one or more streams may be controlled by a CAS. The free_CA_mode flag is ignored by Ziggo DVB-C receivers.

4.8.2. SDT descriptors

service_descriptor (0x48)

The service descriptor provides the names of the service provider and the service in text form together with the service_type. This descriptor is defined in chapter 5.2.15.

NOTE: Emphasis characters are ignored by Ziggo DVB-C receivers.

private_data_specifier_descriptor (0x5F)

The private data specifier descriptor is inserted before any private descriptors in the SDT and is needed to introduce Ziggo private descriptors.

channel_number_descriptor (0x81) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

NOTE: The Manufacturer MUST use the EACEM channel number descriptor located in the NIT.

[private descriptor] (0x83) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0x86) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

blackout_descriptor (0x8C)

This descriptor indicates if a channel shall be blacked-out or PIN protected at certain hours. This descriptor may be implemented when required by content owners.

[private descriptor] (0xA3) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xBD) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

CI_protection_descriptor (0xCE)

The common interface protection descriptor informs about the content protection of the current event for CI Plus CICAMs (described in chapter 5.2.4). This descriptor is used to enforce CI Plus CICAMs in CI plus Hosts. Currently implemented for all Ziggo services.

4.9. EIT - EVENT INFORMATION TABLE

The EIT is defined according to *ETSI EN 300 468* [2] and implemented according to *ETSI ETR 211* [5]. EIT-P/F (present/following) and EIT-S (schedule) for actual and other TS are broadcasted on all the TS of the Ziggo Network except VOD Transport Streams.

4.9.1. EIT table structure

The EIT structure is shown in table 12.

EIT: Event Information Table	Nr. of bits	Identifier
event_information_section() {		
table_id *0x4E, 0x4F 0x50, 0x6*0	8	uimsbf
section_syntax_indicator	1	bslbf
reserved_future_use	1	bslbf
reserved	2	bslbf
section_length	12	uimsbf
service_id	16	uimsbf
reserved	2	bslbf
version_number	5	uimsbf
current_next_indicator	1	bslbf
section_number	8	uimsbf
last_section_number	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
segment_last_section_number	8	uimsbf
last_table_id	8	uimsbf
for (i=0; i<N; i++) {		
event_id	16	uimsbf
start_time	40	bslbf
duration	24	uimsbf
running_status *// ignored by Ziggo*	3	uimsbf
free_CA_mode *// ignored by Ziggo*	1	bslbf
descriptors_loop_length	12	uimsbf
for (j=0; j<N; i++) {		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

Table 12: Structure of the Event Information Table

table_id

This field indicates the EIT type. Table_id values are represented in table 13.

Table_id	EIT type
0x4E	EIT-P/F actual
0x4F	EIT-P/F other
0x50	EIT-S actual first 4 days
0x51 to 0x5F	*EIT-S actual after 4th day - Not used*
0x60	EIT-S other first 4 days
0x61 to 0x6F	*EIT-S other after 4th day - Not used*

Table 13: Event information table - table_id values

service_id

This 16 bit field serves as a label to distinguish this service from any other service within the TS. The service_id is the same as the program_number in the corresponding program_map_section.

section_number

This 8 bit field indicates if the section describes the present event (value '0') or the following (value '1'). For EIT-S, the section numbering follows the requirements specified in *ETSI ETR 211* [5].

transport_stream_id
This 16 bit field serves as a label for identification of this TS from any other multiplex within the delivery system.

original_network_id
This 16 bit field gives the label identifying the network_id of the originating delivery system. For Ziggo the value is 1536

segment_last_section_number
This 8 bit field is used for EIT-S. It contains the section number of the last section that is part of the current segment. It is managed as specified in *ETSI ETR 211* [5].

event_id
This 16 bit field contains the identification number of the described event (uniquely allocated within a service definition).

start_time
This 40 bit field contains the start time of the event in Universal Time, Co-ordinated (UTC) and Modified Julian Date (MJD) (defined in annex C of *ETSI EN 300 468* [2]). This field is coded as 16 bits giving the 16 LSBs of MJD followed by 24 bits coded as 6 time digits in 4 bit Binary Coded Decimal (BCD). If the start time is undefined (e.g. for an event in a NVOD reference service) all bits of the field are set to "1".

EXAMPLE: 93/10/13 12:45:00 is coded as "0xC079124500".

Duration
This 24 bit field contains the duration of the event in hours, minutes, and seconds.
Format: 6 digits, 4 bit BCD = 24 bit.

EXAMPLE: 01:45:30 is coded as "0x014530".

running_status
Not used by Ziggo DVB-C receivers.

free_CA_mode
Not used by Ziggo DVB-C receivers.

4.9.2. EIT descriptors

The following descriptors may appear in the EIT:

short_event_descriptor (0x4D)

The short event descriptor provides the name of the event (and NOT a short description of the event).

extended_event_descriptor (0x4E)

The extended event descriptor provides a detailed text description of an event.

component_descriptor (0x50)

The component descriptor identifies the type of stream components the event contains. Ziggo DVB-C receivers use it only to identify a High Definition video component for an event. If not present, the event is NOT considered as an HD event on Ziggo DVB-C receivers.

content_descriptor (0x54)

The content descriptor is used to provide classification information for an event. This descriptor is defined in chapter 5.2.6.

parental_rating_descriptor (0x55)

The parental rating descriptor gives a rating based on age. This descriptor is defined in chapter 5.2.13.

private_data_specifier_descriptor (0x5F)

The private data specifier descriptor is inserted before any private descriptors in the EIT and is needed to introduce Ziggo private descriptors.

content_identifier_descriptor (0x76) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xA3) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xA4) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xA5) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xA6) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xA7) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

[private descriptor] (0xBB) Ziggo proprietary

This descriptor MUST be ignored by the DVB-C receiver of the Manufacturer.

5. DESCRIPTOR SPECIFICATIONS

5.1. DESCRIPTOR IDENTIFICATION AND LOCATION

Table 14 lists the descriptors used by Ziggo, giving the descriptor-tag values and their intended placement with the SI and PSI tables. Some other DVB/MPEG descriptors might not be listed here because unused or trivial use.

All NIT, SDT and EIT descriptors marked as “private” are always be preceded by a private data specifier descriptor (described in chapter 5.2.14). Private descriptors in the PMT are preceded by a private data indicator descriptor.

It must be acknowledged that differences in PSI/SI exist between the former UPC (fUPC) and the former Ziggo (fZiggo) networks. The difference is required for support of legacy DVB-C receivers in both footprints. A DVB-C receiver of a Manufacturer must support the applicable descriptors for both networks.

Descriptor	Tag value	Private descriptor	NIT	BAT	TOT	SDT	EIT	PMT	CAT	AIT
application_descriptor	0x00									✓
application_name_descriptor	0x01									✓
transport_protocol_descriptor	0x02									✓
video_stream_descriptor	0x02							✓		
audio_stream_descriptor	0x03							✓		
registration_descriptor	0x05							✓		
CA_descriptor	0x09							✓	✓	
ISO_639_language_descriptor	0x0A							✓		
maximum_bitrate_descriptor	0x0E							✓		
private_data_indicator_descriptor	0x0F							✓		
simple_application_location_descriptor	0x15									✓
network_name_descriptor	0x40		✓							
service_list_descriptor	0x41		✓							
cable_delivery_system_descriptor	0x44		✓							
bouquet_name_descriptor	0x47			✓						
service_descriptor	0x48					✓				
linkage_descriptor	0x4A		✓	✓						
short_event_descriptor	0x4D						✓			
extended_event_descriptor	0x4E						✓			
component_descriptor	0x50						✓			
stream_identifier_descriptor	0x52							✓		
content_descriptor	0x54						✓			
parental_rating_descriptor	0x55						✓			
teletext_descriptor	0x56							✓		
local_time_offset_descriptor	0x58				✓					
subtitling_descriptor	0x59							✓		
private_data_specifier_descriptor	0x5F		✓	✓		✓				
data_broadcast_id_descriptor	0x66							✓		
AC-3_descriptor	0x6A							✓		
application_signaling_descriptor	0x6F							✓		
content_identifier_descriptor	0x76						✓			
[private descriptor]	0x81	✓				✓				
EACEM_logical_channel_descriptor	0x83	✓	✓							
[private descriptor]	0x84	✓	✓							
[private descriptor]	0x85	✓	✓							
[private descriptor]	0x86	✓				✓				
[private descriptor]	0x87	✓	✓							
hd_simulcast_logical_channel_descriptor	0x88	✓	✓							
[private descriptor]	0x89	✓	✓							
[private descriptor]	0x8C	✓				✓				
[private descriptor]	0x90	✓						✓		
[private descriptor]	0x91	✓	✓							
[private descriptor]	0x92	✓	✓							
[private descriptor]	0x93	✓	✓							
[private descriptor]	0x95	✓	✓							
[private descriptor]	0x95	✓				✓				

Descriptor	Tag value	Private descriptor	NIT	BAT	TOT	SDT	EIT	PMT	CAT	AIT
[private descriptor]	0x97	✓	✓							
[private descriptor]	0x97	✓				✓				
[private descriptor]	0x99	✓	✓							
[private descriptor]	0x9A	✓	✓							
[private descriptor]	0x9B	✓		✓						
[private descriptor]	0xA3	✓					✓			
[private descriptor]	0xA3	✓				✓				
[private descriptor]	0xA4	✓					✓			
[private descriptor]	0xA5	✓				✓				
[private descriptor]	0xA6	✓					✓			
[private descriptor]	0xA7	✓					✓			
[private descriptor]	0xBD	✓				✓				
ci_protection_descriptor	0xCE	✓				✓				
[private descriptor]	0xD0	✓		✓						
[private descriptor]	0xD1	✓		✓						
[private descriptor]	0xD2	✓	✓							
[private descriptor]	0xD4	✓				✓				
[private descriptor]	0xD5	✓		✓						
[private descriptor]	0xD6	✓		✓						
[private descriptor]	0xD7	✓				✓				
[private descriptor]	0xD8	✓		✓						
[private descriptor]	0xFD	✓						✓		
[private descriptor]	0xFE	✓						✓		

Table 14: Location of descriptors

5.2. DESCRIPTOR CODING

This chapter describes the descriptors in more detail, including the descriptor coding and tag values used.

5.2.1. AC-3 descriptor (PMT)

The AC-3 descriptor (see table 15) is defined in *ETSI EN 300 468* [2], used to identify an AC-3 audio elementary stream. The descriptor tag provides a unique identification of the presence of the AC-3 elementary stream. Other optional fields in the descriptor are ignored by Ziggo DVB-C receivers. The descriptor has a minimum length of one byte, but may be longer depending upon the state of the flags and the additional info loop.

Syntax	Nr. of bits	Identifier
AC-3_descriptor() {		
descriptor_tag *0x6A*	8	uimsbf
descriptor_length	8	uimsbf
component_type_flag	1	bslbf
bsid_flag	1	bslbf
mainid_flag	1	bslbf
asvc_flag	1	bslbf
reserved_flags	4	bslbf
If (component_type_flag == 1) {		
component_type	8	uimsbf
}		
If (bsid_flag == 1) {		
Bsid	8	uimsbf
}		
If (mainid_flag==1) {		
Mainid	8	uimsbf
}		
If (asvc_flag==1) {		
Asvc	8	uimsbf
}		
for (i=0; i<N; i++) {		
Additional_info-byte	8	uimsbf
}		
}		

Table 15: Syntax for the AC-3 descriptor

5.2.2. CA Descriptor (PMT, CAT)

The conditional access descriptor (see table 16) is defined in *ISO/IEC 13818-1* [1], used to specify both system wide conditional access management information such as EMMs and elementary stream specific information such as ECMs.

If any elementary stream is scrambled, a CA descriptor must be present for the program containing that elementary stream. If any system wide conditional access management information exists within a Transport Stream, a CA descriptor shall be present in the conditional access table (CAT).

When the CA descriptor is found in the PMT, the CA_PID points to packets containing program related access control information, such as ECMs. Its presence for a program indicates applicability to the entire program. In the same case, its presence as extended ES information indicates applicability to the associated program element. Scrambling at ES level is currently not used.

When the CA descriptor is found in the CAT, the CA_PID points to packets containing system wide and/or access control management information, such as EMMs.

Syntax	Nr. of bits	Identifier
CA_descriptor() {		
descriptor_tag *0x09*	8	uimsbf
descriptor_length	8	uimsbf
CA_system_id	16	uimsbf
reserved	3	uimsbf
CA_PID	13	uimsbf
for (i=0; i<N; i++) {		
private_data_byte	8	uimsbf
}		
}		

Table 16: Syntax for the CA descriptor

CA_system_ID

This 16 bit field indicates the type of CA system applicable for either the associated ECM and/or EMM streams.

CA_PID

This 13 bit field indicates the PID of the Transport Stream packets which shall contain either ECM or EMM information for the CA system.

NOTE: Detailed information about the Conditional Access systems used by Ziggo is secret. It can be made available upon request and a mutually signed NDA.

5.2.3. Cable delivery system descriptor (NIT)

The cable delivery descriptor (see table 17) is defined in *ETSI EN 300 468* [2], used to describe the modulation parameters of an available Transport Stream. This descriptor appears in the second loop of the NIT.

Syntax	Nr. of bits	Identifier
Cable_delivery_system_descriptor() {		
descriptor_tag *0x44*	8	uimsbf
descriptor_length	8	uimsbf
frequency	32	bslbf
reserved_future_use	12	bslbf
FEC_outer	4	bslbf
modulation	8	bslbf
symbol-rate	28	bslbf
FEC_inner	4	bslbf
}		

Table 17: Syntax for the cable delivery system descriptor

Semantics for cable delivery system descriptor:

frequency

This 32 bit field contains the frequency in 4 bit BCD values specifying 8 decimals of the frequency value. The frequency is coded in MHz, where the decimal occurs after the fourth character (e.g. 0312.0000 MHz).

FEC_outer

This 4 bit field specifies the outer Forward Error Correction (FEC) scheme used according to table 18:

FEC_outer	Description
0x0	Not defined
0x1	No outer FEC coding
0x2	RS (204/188)
0x3 to 0xF	Reserved for future use

Table 18: Cable delivery system descriptor - FEC_Outer values

modulation

This 8 bit field specifies the modulation scheme used according to table 19:

Modulation	Description
0x00	Not defined
0x01	16-QAM not used
0x02	32-QAM not used
0x03	64-QAM
0x04	128-QAM not used
0x05	256-QAM
0x06 to 0xFF	Reserved for future use

Table 19: Cable delivery system descriptor - Modulation values

symbol_rate

This 28 bit field contains the symbol rate as 4 bit BCD values specifying 7 characters of the symbol rate in Megasymbols per second (MSym/s) where the decimal point occurs after the third character (e.g. 006.9000).

FEC_inner

This 4 bit field specifies the inner FEC scheme used according to table 20:

FEC_inner	Description
0x0	Not defined
0x1	1/2 conv. code rate
0x2	2/3 conv. code rate
0x3	3/4 conv. code rate
0x4	5/6 conv. code rate
0x5	7/8 conv. code rate
0x6	8/9 conv. code rate
0x7 to 0xE	Reserved for future use
0xF	No conv. coding

Table 20: Cable delivery system descriptor - FEC_inner values

5.2.4. CI protection descriptor (SDT)

The CI protection descriptor (see table 21) is defined in *CI Plus Specification v1.3.1* [7], used to indicate the CI operating mode required by a service. It shall appear in the service descriptor loop of the SDT-actual and shall be preceded by a private data specifier descriptor according to *ETSI EN 300 468* [2]

Syntax	Nr. of bits	Identifier
ci_protection_descriptor() {		
descriptor_tag *0xCE*	8	uimsbf
descriptor_length	8	uimsbf
free_ci_mode_flag	1	bslbf
match_brand_flag	1	bslbf
reserved_future_use *// all set to '1'*	6	bslbf
if (match_brand_flag == 1) {		
number_of_entries	8	uimsbf
for (i=0; i<N; i++) {		
cicam_brand_identifier	16	uimsbf
}		
}		
for (i=0; i<n; i++) {		
private_data_byte	8	uimsbf
}		
}		

Table 21: Syntax for the CI protection descriptor

Semantics for the component descriptor:

free_ci_mode_flag

This 1 bit field identifies the CI operating mode. When set to '0', indicates that all of the component streams of the service do not require CI Plus protection. When set to '1', indicates that all of the component streams of the service require CI Plus protection if they are not transmitted in the clear on the broadcast network.

match_brand_flag

This 1 bit field signifies that the descriptor includes a list of cicam_brand_identifiers. When set to '0', indicates that this service has no chosen CICAM brands. When set to '1' , indicates that this service has chosen to set CICAM brands. The match_brand_flag is only interpreted when the free_ci_mode_flag is set to '1'.

number_of_entries

This 8 bit field specifies the number of cicam_brand_identifiers that are contained in the brand identifier loop. When match_brand_flag field has been set to '1', the number_of_entries shall be ≠ 0.

cicam_brand_identifier

This 16 bit field identifies the CICAM brands that may be used with the service. When no CICAM brand identifiers are present, any CI Plus CICAM may be used with the Host. When one or more CICAM brand identifiers are specified, the Host shall only operate with a CI Plus CICAM device whose Device Certificate cicamBrandId matches the cicam_brand_identifier. If none of the cicam_brand_identifiers present are matched with the CICAM device certificate then the CICAM shall be shunned for this service. The cicam_brand_identifier value 0x0000 is reserved and shall not be used.

private_data_byte

This is included for future extensions to Host Service Shunning. For version 1.3.1 of the specification is undefined and if present shall be ignored.

5.2.5. Component descriptor (EIT)

The component descriptor (see table 22) is defined in *ETSI EN 300 468* [2], used to identify the type of component stream and may be used to provide a text description of the elementary stream. If broadcasted, it shall appear in the event loop of the EIT-P/F and EIT-S. Its syntax is defined as below. Ziggo only uses this descriptor to signal HD content for an event.

Syntax	Nr. of bits	Identifier
component_ descriptor() {		
descriptor_tag *0x50*	8	uimsbf
descriptor_length	8	uimsbf
stream_content_ext *// ignored by Ziggo*	4	uimsbf
stream_content	4	uimsbf
component_type	8	uimsbf
component_tag *// ignored by Ziggo*	8	uimsbf
ISO_639_language_code *// ignored by Ziggo*	24	bslbf
for (i=0;i<N;i++) {		
text_char *// ignored by Ziggo*	8	uimsbf
}		
}		

Table 22: Syntax for the component descriptor

Semantics for the component descriptor:

stream_content

This 4 bit field specifies the type of the video, audio, or EBU-data stream. The coding of this field is specified in table 23.

component_type

This 8 bit field specifies the type of the video, audio or EBU-data component. The coding of this field is specified in table 23.

Stream_ content	Stream_ content_ext	Component _type	Description	Detection of HD by VZ
0x0	0xF	0x00 to 0xFF	reserved for future use	No
0x1	0xF	0x00	reserved for future use	No
0x1	0xF	0x01	MPEG-2 video, 4:3 aspect ratio, 25 Hz	No
0x1	0xF	0x02	MPEG-2 video, 16:9 aspect ratio with pan vectors, 25 Hz	No
0x1	0xF	0x03	MPEG-2 video, 16:9 aspect ratio without pan vectors, 25 Hz	No
0x1	0xF	0x04	MPEG-2 video, > 16:9 aspect ratio, 25 Hz	No
0x1	0xF	0x05	MPEG-2 video, 4:3 aspect ratio, 30 Hz	No
0x1	0xF	0x06	MPEG-2 video, 16:9 aspect ratio with pan vectors, 30 Hz	No
0x1	0xF	0x07		No
0x1	0xF	0x08	MPEG-2 video, > 16:9 aspect ratio, 30 Hz	No
0x1	0xF	0x09	MPEG-2 high definition video, 4:3 aspect ratio, 25 Hz	Yes
0x1	0xF	0x0A	MPEG-2 high definition video, 16:9 aspect ratio with pan vectors, 25 Hz	Yes
0x1	0xF	0x0B	MPEG-2 high definition video, 16:9 aspect ratio without pan vectors, 25 Hz	Yes
0x1	0xF	0x0C	MPEG-2 high definition video, > 16:9 aspect ratio, 25 Hz	Yes
0x1	0xF	0x0D	MPEG-2 high definition video, 4:3 aspect ratio, 30 Hz	Yes
0x1	0xF	0x0E	MPEG-2 high definition video, 16:9 aspect ratio with pan vectors, 30 Hz	Yes
0x1	0xF	0x0F	MPEG-2 high definition video, 16:9 aspect ratio without pan vectors, 30 Hz	Yes
0x1	0xF	0x10	MPEG-2 high definition video, > 16:9 aspect ratio, 30 Hz	Yes
0x1	0xF	0x11 to 0xAF	reserved for future use	No
0x1	0xF	0xB0 to 0xFE	user defined	No
0x1	0xF	0xFF	reserved for future use	No
0x2	0xF	0x00	reserved for future use	No
0x2	0xF	0x01	MPEG-1 Layer 2 audio, single mono channel	No
0x2	0xF	0x02	MPEG-1 Layer 2 audio, dual mono channel	No
0x2	0xF	0x03	MPEG-1 Layer 2 audio, stereo (2 channel)	No
0x2	0xF	0x04	MPEG-1 Layer 2 audio, multi-lingual, multi-channel	No
0x2	0xF	0x05	MPEG-1 Layer 2 audio, surround sound	No
0x2	0xF	0x06 to 0x3F	reserved for future use	No
0x2	0xF	0x40	MPEG-1 Layer 2 audio description for the visually impaired	No
0x2	0xF	0x41	MPEG-1 Layer 2 audio for the hard of hearing	No
0x2	0xF	0x42	receiver-mixed supplementary audio as per annex G of TR 101 154	No
0x2	0xF	0x43 to 0xAF	reserved for future use	No
0x2	0xF	0xB0 to 0xFE	user-defined	No
0x2	0xF	0xFF	reserved for future use	No
0x3	0xF	0x00	reserved for future use	No
0x3	0xF	0x01	EBU Teletext subtitles	No
0x3	0xF	0x02	associated EBU Teletext	No
0x3	0xF	0x03	VBI data	No
0x3	0xF	0x04 to 0x0F	reserved for future use	No
0x3	0xF	0x10	DVB subtitles (normal) with no monitor aspect ratio criticality	No
0x3	0xF	0x11	DVB subtitles (normal) for display on 4:3 aspect ratio monitor	No
0x3	0xF	0x12	DVB subtitles (normal) for display on 16:9 aspect ratio monitor	No
0x3	0xF	0x13	DVB subtitles (normal) for display on 2.21:1 aspect ratio monitor	No
0x3	0xF	0x14	DVB subtitles (normal) for display on high definition monitor	No
0x3	0xF	0x15 to 0x1F	reserved for future use	No

Stream_ content	Stream_ content_ext	Component _type	Description	Detection of HD by VZ
0x3	*0xF*	0x20	DVB subtitles (for the hard of hearing) with no monitor aspect ratio criticality	No
0x3	*0xF*	0x21	DVB subtitles (for the hard of hearing) for display on 4:3 aspect ratio monitor	No
0x3	*0xF*	0x22	DVB subtitles (for the hard of hearing) for display on 16:9 aspect ratio monitor	No
0x3	0xF	0x23	DVB subtitles (for the hard of hearing) for display on 2.21:1 aspect ratio monitor	No
0x3	*0xF*	0x24	DVB subtitles (for the hard of hearing) for display on high definition monitor	No
0x3	0xF	0x25 to 0xAF	reserved for future use	No
0x3	*0xF*	0xB0 to 0xFE	user defined	No
0x3	0xF	0xFF	reserved for future use	No
0x4	*0xF*	0x00 to 0x7F	reserved for AC-3 audio modes	No
0x4	*0xF*	0x80 to 0xFF	reserved for enhanced AC-3 audio modes	No
0x5	0xF	0x00	reserved for future use	No
0x5	*0xF*	0x01	H264/AVC standard definition video, 4:3 aspect ratio, 25 Hz	No
0x5	0xF	0x02	reserved for future use	No
0x5	*0xF*	0x03	H264/AVC standard definition video, 16:9 aspect ratio, 25 Hz	No
0x5	*0xF*	0x04	H264/AVC standard definition video, > 16:9 aspect ratio, 25 Hz	No
0x5	0xF	0x05	H264/AVC standard definition video, 4:3 aspect ratio, 30 Hz	No
0x5	0xF	0x06	reserved for future use	No
0x5	0xF	0x07	H264/AVC standard definition video, 16:9 aspect ratio, 30 Hz	No
0x5	0xF	0x08	H264/AVC standard definition video, > 16:9 aspect ratio, 30 Hz	No
0x5	0xF	0x09 to 0x0A	reserved for future use	No
0x5	*0xF*	0x0B	H264/AVC high definition video, 16:9 aspect ratio, 25 Hz	Yes
0x5	*0xF*	0x0C	H264/AVC high definition video, > 16:9 aspect ratio, 25 Hz	Yes
0x5	0xF	0x0D to 0x0E	reserved for future use	No
0x5	0xF	0x0F	H264/AVC high definition video, 16:9 aspect ratio, 30 Hz	Yes
0x5	0xF	0x10	H264/AVC high definition video, > 16:9 aspect ratio, 30 Hz	Yes
0x5	0xF	0x11 to 0xAF	reserved for future use	No
0x5	*0xF*	0xB0 to 0xFE	user-defined	No
0x5	0xF	0xFF	reserved for future use	No
0x6	0xF	0x00	reserved for future use	No
0x6	*0xF*	0x01	HE-AAC audio, single mono channel	No
0x6	0xF	0x02	reserved for future use	No
0x6	*0xF*	0x03	HE-AAC audio, stereo	No
0x6	0xF	0x04	reserved for future use	No
0x6	*0xF*	0x05	HE-AAC audio, surround sound	No
0x6	0xF	0x06 to 0x3F	reserved for future use	No
0x6	*0xF*	0x40	HE-AAC audio description for the visually impaired	No
0x6	*0xF*	0x41	HE-AAC audio for the hard of hearing	No
0x6	*0xF*	0x42	receiver-mixed supplementary audio as per annex G of TR 101 154	No
0x6	0xF	0x43 to 0xAF	reserved for future use	No
0x6	*0xF*	0xB0 to 0xFE	user-defined	No
0x6	0xF	0xFF	reserved for future use	No
0x7	0xF	0x00 to 0x7F	reserved for DTS audio modes	No
0x7	0xF	0x80 to 0xFF	reserved for future use	No
0x8	0xF	0x00 to 0xFF	reserved for future use	No
0x9	0x0	0x00	HEVC Main Profile high definition video, 50 Hz	No

Stream_ content	Stream_ content_ext	Component _type	Description	Detection of HD by VZ
0x9	0x0	0x01	HEVC Main 10 Profile high definition video, 50 Hz	No
0x9	0x0	0x02	HEVC Main Profile high definition video, 60 Hz	No
0x9	0x0	0x03	HEVC Main 10 Profile high definition video, 60 Hz	No
0x9	0x0	0x04	HEVC ultra high definition video	No
0x9	0x0	0x05 to 0xFF	reserved for future use	No
0x9	0x1	0x00	AC-4 main audio, mono	No
0x9	0x1	0x01	AC-4 main audio, mono, dialogue enhancement enabled	No
0x9	0x1	0x02	AC-4 main audio, stereo	No
0x9	0x1	0x03	AC-4 main audio, stereo, dialogue enhancement enabled	No
0x9	0x1	0x04	AC-4 main audio, multichannel	No
0x9	0x1	0x05	AC-4 main audio, multichannel, dialogue enhancement enabled	No
0x9	0x1	0x06	AC-4 broadcast-mix audio description, mono, for the visually impaired	No
0x9	0x1	0x07	AC-4 broadcast-mix audio description, mono, for the visually impaired, dialogue enhancement enabled	No
0x9	0x1	0x08	AC-4 broadcast-mix audio description, stereo, for the visually impaired	No
0x9	0x1	0x09	AC-4 broadcast-mix audio description, stereo, for the visually impaired, dialogue enhancement enabled	No
0x9	0x1	0x0A	AC-4 broadcast-mix audio description, multichannel, for the visually impaired	No
0x9	0x1	0x0B	AC-4 broadcast-mix audio description, multichannel, for the visually impaired, dialogue enhancement enabled	No
0x9	0x1	0x0C	AC-4 receiver-mix audio description, mono, for the visually impaired	No
0x9	0x1	0x0D	AC-4 receiver-mix audio description, stereo, for the visually impaired	No
0x9	0x1	0x0E to 0xFF	reserved for future use	No
0x9	0x2 to 0xF	0x00 to 0xFF	reserved for future use	No
0xA	0x0 to 0xF	0x00 to 0xFF	reserved for future use	No
0xB	0x0 to 0xE	0x00 to 0xFF	reserved for future use	No
0xB	0xF	0x00	less than 16:9 aspect ratio	No
0xB	0xF	0x01	16:9 aspect ratio	No
0xB	0xF	0x02	greater than 16:9 aspect ratio	No
0xB	0xF	0x03	plano-stereoscopic top and bottom (TaB) frame-packing	No
0xB	0xF	0x04 to 0x0F		No
0xC to 0xF	NA	0x00 to 0xFF	user defined	No

Table 23: Component descriptor - stream_content and component_type values

5.2.6. Content descriptor (EIT)

The content descriptor (see table 24) is defined in *ETSI EN 300 468* [2], used to provide classification information for an event. If broadcasted, it shall appear in the event loop of the EIT-P/F and EIT-S.

Syntax	Nr. of bits	Identifier
content_descriptor() {		
descriptor_tag *0x54*	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
content_nibble_level_1	4	uimsbf
content_nibble_level_2	4	uimsbf
user_nibble	4	uimsbf
user_nibble	4	uimsbf
}		
}		

Table 24: Syntax for the Content descriptor

Semantics for the content descriptor:

content_nibble_level_1

This 4 bit field represents the first level of a content identifier:

Content_nibble_level_1	Description
0x0	Undefined
0x1	Movie/drama
0x2	News/current affairs
0x3	Show/game Show
0x4	Sports
0x5	Children's/Youth programmes
0x6	Music/Ballet/dance
0x7	Arts/Culture without music
0x8	Social/Political issues/Economics:
0x9	Education/Science/Factual topics:
0xA	Leisure hobbies:
0xB	Special characteristics:
0xC to 0x0E	Reserved
0xF	User defined

Table 25: Content descriptor - content_nibble_level_1 values

content_nibble_level_2

This 4 bit field represents the second level of a content identifier. It is used in combination with nibble 1 to display the event genre in the Channel Info Bar of a specific set DVB-C receivers in fUPC.

user_nibble

Ignored by the Ziggo DVB-C receivers.

5.2.7. EACEM logical channel descriptor (NIT)

The EACEM logical channel descriptor (see table 26) is defined in *EICTA C-book v2.1* [8], used to provide a default channel number label for services. The logical channel descriptor may be inserted more than once in the second descriptor loop of the NIT. Since a logical channel number may be duplicated with different service types, the receiver shall provide a means of selecting between different service types.

This private descriptor is preceded by a private data specifier descriptor (tag 0x5F) with PDS/PDI == 0x00000028 (EACEM)

Syntax	Nr. of bits	Identifier
EACEM_logical_channel_descriptor() {		
descriptor_tag *0x83*	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
service_id	16	uimsbf
visible_service_flag	1	bslbf
reserved_future_use	5	bslbf
logical_channel_number	10	uimsbf
}		
}		

Table 26: Syntax for the EACEM channel descriptor

Semantics for the content protection descriptor:

service_id

This is a 16 bit field which serves as a label to identify this service from any other service within the network. The service_id is the same as the program_number in the corresponding program_map_section. Services shall be included irrespective of their running status.

visible_service_flag

This 1 bit field when set to '1' indicates that the service is normally visible and selectable (subject to the service type being suitable, etc.) via the receiver service list. When set to '0' this indicates that the DVB-C receiver is not expected to offer the service to the user in navigation modes. The DVB-C receiver SHALL NOT provide a mechanism to access these invisible services.

logical_channel_number

This 10 bit field indicates the broadcaster preference for ordering services. Its use is defined in table 27:

logical_channel_ number	Description
0	Service not suitable for selection and/or showing by the end-user (see note)
1 to 999	Broadcast logical channel number
1000 to 1023	Reserved for DVB-C receiver manufacturer use
NOTE: For example, the value zero may be used for data services only intended for selection from interactive applications or for firmware download services, etc.	

Table 27: EACEM logical channel number descriptor - channel number values

5.2.8. Extended event descriptor (EIT)

The extended event descriptor (see table 28) is defined in *ETSI EN 300 468* [2], used to provide a detailed text description of an event, which may be used in addition to the short event descriptor (see table 46). If broadcasted, it shall appear in the event loop of the EIT-P/F and EIT-S. It can appear as many times as there are languages defined.

Syntax	Nr. of bits	Identifier
extended_event_descriptor() {		
descriptor_tag *0x4E*	8	uimsbf
descriptor_length	8	uimsbf
descriptor_number	4	uimsbf
last_descriptor_number	4	uimsbf
ISO_639_language_code	24	bslbf
length_of_items	8	uimsbf
for (i=0; i<N; i++) {		
item_description_length	8	uimsbf
for (j=0; j<N; j++) {		
item_description_char	8	Uimsbf
}		
item_length	8	uimsbf
for (j=0; j<N; j++) {		
item_char	8	uimsbf
}		
}		
text_length	8	uimsbf
for (i=0; i<N; i++) {		
text_char	8	uimsbf
}		
}		

Table 28: Syntax for the Extended event descriptor

Semantics for the extended event descriptor:

ISO_639_language_code

This 24 bit field contains the *ISO 639-2* [9] three character language code of the following text fields.

length_of_items

This 8 bit field specifies the length in bytes of the following items.

NOTE: The item loop is ignored by Ziggo DVB-C receivers.

Item_description_char

This is an 8 bit field. Ignored by Ziggo DVB-C receivers.

Item_char

This is an 8 bit field. Ignored by the Ziggo DVB-C receivers.

text_length

This 8 bit field specifies the length of the following text string. For Ziggo, this field has a maximum value of 240 in EIT-P/F and 100 in EIT-S. If text is too long, it is truncated by the EIT server before placement in the EIT. When text_length (contrary to what is specified here) is bigger than the expected maximum value, it will be truncated by Ziggo DVB-C receivers.

text_char

String of 8 bit characters specifying the non-itemized extended text. This text is used for display purposes on Ziggo DVB-C receivers. Text information is coded using the character sets and methods described in section annex A.

5.2.9. HD Simulcast Descriptor (NIT)

The HD simulcast descriptor (see table 29) is defined in *EICTA C-book v2.1* [8], used to provide a means to override the default (LCN) channel number label of services for an advanced receiver. The HD simulcast logical channel descriptor may be inserted in the second descriptor loop of the NIT and may appear more than once in this location. The constraints on uniqueness are the same as those for the logical channel descriptor.

This private descriptor is preceded by a private data specifier descriptor (tag 0x5F) with PDS/PDI == 0x00000028 (EACEM)

Syntax	Nr. of bits	Identifier
HD_simulcast_logical_channel_descriptor() {		
descriptor_tag *0x88*	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
service_id	16	uimsbf
visible_service_flag	1	bslbf
reserved_future_use	5	bslbf
logical_channel_number	10	uimsbf
}		
}		

Table 29: Syntax for the HD simulcast descriptor

Semantics for the extended event descriptor:

service_id

This 16 bit field identifies a service within a TS. The service_id is the same as the program_number in the corresponding program_map_section.

visible_service_flag

This 1 bit field when set to '1' indicates that the service is normally visible and selectable (subject to the service type being suitable etc.) via the receiver service list. When set to '0' this indicates that the receiver is not expected to offer the service to the user during navigation. The receiver shall NOT provide a mechanism to access these invisible services.

logical_channel_number

This 10 bit field indicates the broadcaster preference for the ordering of services. This descriptor shall only be interpreted by DVB-C receivers that are able to decode an advanced codec HD digital television service. The channel number label assignment defined by this descriptor overrides the channel number label assignment defined by the EACEM Logical Channel Descriptor that is located in the same network_id. The rules for the set of channel number labels used by this descriptor is the same as the rules for the set of channel number labels used by the EACEM Logical Channel Descriptor.

In the case where this descriptor assigns to a service (service A) a channel number label which is already assigned to another service (service B) (perhaps by the Logical Channel Descriptor), the DVB-C receiver shall treat the original service (service B) as having no assigned channel number label and assign one automatically in the normal manner.

This descriptor is intended, but not limited to be used for HD services broadcast in simulcast with the same service in SD so that the HD service appears at the primary channel number label on HD capable receivers while the SD service appears at that label for SD-only capable receivers.

5.2.10. ISO 639 language descriptor (PMT)

The ISO 639 language descriptor (see table 30) is defined in *ISO/IEC 13818-1* [1], used to specify the language of the associated program element.

Syntax	Nr. of bits	Identifier
ISO_639_language_descriptor() {		
Descriptor_tag *0x0A*	8	uimsbf
Descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
ISO_639_language_code	24	uimsbf
audio_type	8	uimsbf
}		
}		

Table 30: Syntax for the ISO 639 language descriptor

Semantics for the ISO 639 language descriptor:

ISO_639_language_code
This 24 bit field identifies the language or languages used by the associated program element.

The ISO_639_language_code contains a 3 character code as specified by *ISO 639-2* [9]. Each character is coded into 8 bits according to *ISO/IEC 8859-1* [10] and inserted in order into the 24 bit field.

In the case of multilingual audio streams the sequence of ISO_639_language_code fields shall reflect the content of the audio stream. For dual audio the sequence of codes identifies the language for each audio channel as follows: the first code signals the language of the left channel, channel 1, of the dual-mono stream. The second code signals the language of the right channel, channel 2.

NOTE: Both ISO 639-2/B and ISO 639-2/T may be used.

audio_type
This 8 bit field specifies the type of an audio stream. Ignored by Ziggo DVB-C receivers.

5.2.11. Linkage descriptor (NIT)

The linkage descriptor (see table 31) is defined in *ETSI EN 300 468* [2], used to locate the services which do not appear in any service list such as a system software update service, EPG and interactive applications.

Syntax	Nr. of bits	Identifier
linkage_descriptor() {		
descriptor_tag *0x4A*	8	uimsbf
descriptor_length	8	uimsbf
transport_stream_id	16	uimsbf
original_network_id	16	uimsbf
service_id	16	uimsbf
linkage_type	8	uimsbf
if (linkage_type==0x09) {		
system_software_update_link_structure ()		
} else if (linkage_type==0x81) {		
logical_service_name_length	8	uimsbf
for (i=0; i<N; i++) {		
logical_service_name_char	8	uimsbf
}		
location_length	8	uimsbf
for (i=0;i<N;i++) {		
location_char	8	uimsbf
}		
language_length	8	uimsbf
for (i=0; i<N; i++) {		
ISO_639_language_code	24	bslbf
service_title_length	8	uimsbf
for (j=0; j<N; j++) {		
service_title_char	8	uimsbf
}		
}		
}		
for (i=0; i<N; i++) {		
private_data_byte	8	uimsbf
}		
}		

Table 31: Syntax for the Linkage desciptor

Semantics for the Linkage descriptor:

transport_stream_id

This is a 16 bit field which identifies the Transport Stream containing the service indicated.

original_network_id

This is a 16 bit field giving the label identifying the network_id of the originating delivery system of the information service indicated.

service_id

This is a 16 bit field which uniquely identifies an information service within a TS. The service_id is the same as the program_number in the corresponding program_map_section.

linkage_type

This 8 bit field is used to define the type of Ziggo service, either a system software update location (0x09) or various (downloadable) interactive applications (0x81). Type 0x81 will signal the use of a private syntax for these services.

NOTE: Also a UPC private data specifier descriptor must be available before the linkage descriptor to introduce the private syntax.

logical_service_name_length

This 8 bit field specifies the length in bytes of the logical service name.

logical_service_name_char

String of 8 bit characters specifying the logical name of the service.

language_length

This 8 bit field specifies the number of languages, equal to the number of service names.

ISO_639_language_code

This 24 bit field contains the *ISO 639-2* [9] three character language code of the language of the following service name. Both ISO 639-2/B and ISO 639-2/T may be used.

service_title_length

This 8 bit field specifies the length in bytes of the following service title.

service_title_char

String of 8 bit characters specifying the title of the service for the given language. Text information is coded using the character sets and methods described in section annex A.

private_data_byte

This is an 8 bit field, the value of it is ignored by Ziggo DVB-C receivers.

system_software_update_link_structure()

The linkage descriptor with the linkage type of 0x09 (system software update service) conveys the location of the Transport Stream carrying a system software update service within a network. This descriptor is carried in the first loop of the NIT. Also see *ETSI TS 102 006* [11].

Syntax	Nr. of bits	Identifier
System_software_update_link_structure() {		
OUI_data_length	8	uimsbf
for (i=0; i<N; i++) {		
OUI	24	uimsbf
selector_length	8	uimsbf
for (j=0; j<N; j++) {		
selector_byte	8	uimsbf
}		
}		
}		

Table 32: Syntax for the System software update link structure

Semantics for the System software update link structure:

OUI_data_length

This 8 bit field specifies the total length in bytes of the following OUI loop.

OUI

This 24 bit field contains an IEEE OUI of the organization providing a system software update service.

selector_length

This 8 bit field specifies the total length in bytes of the following selector field.

selector_byte

This 8 bit field provides information additional to the OUI that can be used by a receiver to locate and identify the system software update service, e.g. model type or ranges. The syntax and semantics of the selector field are defined by Ziggo:

Syntax	Nr. of bits	Identifier
UPC_selector_structure() {		
platform_id	32	uimsbf
product_id	16	uimsbf
usage_id	8	uimsbf
version_number	16	uimsbf
packet_id	16	uimsbf
table_id	16	uimsbf
}		

Table 33: Syntax for the UPC selector structure

Semantics for the UPC selector structure:

platform_id

This 32 bit field identifies a decoder hardware model from a given manufacturer.

product_id

This 16 bit field identifies the customer product for this manufacturer.

usage_id

This 8 bit field allows distinguishing different usage of the same type of decoder. For example, some decoders may be reserved for beta testers and download a specific software version.

version_number

This 16 bit field indicates the version number of the software download.

packet_id

This 16 bit field indicates the PID used to broadcast the software update carousel.

table_id

This 16 bit field indicates the section's table ID used to broadcast the software update carousel.

5.2.12. Local time offset descriptor (TOT)

The local time offset descriptor (see table 35) is defined in *ETSI EN 400 468* [2], used to describe country specific dynamic changes of the local time offset relative to UTC. It is also used to determine the time information in general. This descriptor is carried in the TOT.

Syntax	Nr. of bits	Identifier
Local_time_offset_descriptor() {		
descriptor_tag *0x58*	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
country_code	24	bslbf
country_region	6	bslbf
reserved	1	bslbf
local_time_offset_polarity	1	bslbf
local_time_offset	16	bslbf
time_of_change	40	uimsbf
next_time_offset	16	bslbf
}		
}		

Table 34: Syntax for the Local time offset descriptor

Semantics for the Local time offset descriptor:

country_code

This 24 bit field identifies a country using the 3-character code as specified in *ISO 3166-1* [12]. Each character is coded into 8 bits according to *ISO/IEC 8859-1* [10] and inserted in order into the 24 bit field.

The default country code for Ziggo is NLD. Table 35 shows the country codes recognized by Ziggo DVB-C receivers:

Country_code	Country
NLD	The Netherlands
AUT	Austria
NOR	Norway
SWE	Sweden
FRA	France
HUN	Hungary
CZE	Czech Republic
SVK	Slovakia
SVN	Slovenia
ROU	Romania
IRL	Ireland
CHE	Switzerland
POL	Poland

Table 35: Local time offset descriptor - country_code values

local_time_offset_polarity

This 1 bit information indicates the polarity of the following local_offset_time. When set to '0' the polarity is positive and the local time is advanced to UTC (usually east direction from Greenwich). When set to '1' the polarity is negative and the local time is behind UTC.

local_time_offset

This 16 bit field contains the current offset time from UTC in the range between -12 hours and +13 hours at the area which is indicated by the combination of country_code and country_region_id. These 16 bits are coded as 4 digits in 4 bit BCD in the order – tens of hours, hour, tens of minutes, and minutes.

5.2.13. Parental Rating descriptor (EIT)

This parental rating descriptor (see table 36) is defined in *ETSI EN 400 468* [2], used to provide a rating based on age and allows for extensions based on other rating criteria. If broadcasted, it appears in the event loop of the EIT-P/F and EIT-S.

Syntax	Nr. of bits	Identifier
parental_rating_descriptor() {		
descriptor_tag *0x55*	8	uimsbf
descriptor_length	8	bslbf
for (i=0; i<N; i++) {		
country_code	24	bslbf
rating	8	uimsbf
}		
}		

Table 36: Syntax for the parental rating descriptor

Semantics for the parental rating descriptor:

country_code

This 24 bit field identifies a country using the 3-character code as specified in *ISO 3166-1* [12].

rating

This 8 bit field is coded as shown in table 37.

Rating	Description
0x00	Undefined
0x01 to 0x0F	Minimum age = rating + 3 years
0x10 to 0xFF	Ignored

Table 37: Parental rating descriptor - rating values

DVB rating	NLD
None	Blank
0x1	Blank
0x2	6
0x3	6
0x4	6
0x5	9
0x6	9
0x7	9
0x8	12
0x9	12
0xA	12
0xB	16
0xC	16
0xD	16
0xE	16
0xF	18

Table 38: overview of the displayed value in applications for a given DVB parental rating

For Ziggo the possible parental rating settings and defaults are specified, as shown in the following table.

Country code	Available setting	Default value
NLD	Off, 6, 9, 12, 16, 18	Off

Table 39: Parental rating descriptor - available settings for the Ziggo network

The following table shows for each user setting whether to block content or not, depending on the actual DVB parental rating of the content.

Country	Event parental rating value (DVB)																
NLD	NA	0x0	0x1	0x2	0x3	0x4	0x5	0x6	0x7	0x8	0x9	0xA	0xB	0xC	0xD	0xE	0xF
Off																	
-		L	L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
-			L	L	L	L	L	L	L	L	L	L	L	L	L	L	L
-				L	L	L	L	L	L	L	L	L	L	L	L	L	L
-					L	L	L	L	L	L	L	L	L	L	L	L	L
6						L	L	L	L	L	L	L	L	L	L	L	L
-							L	L	L	L	L	L	L	L	L	L	L
-								L	L	L	L	L	L	L	L	L	L
9									L	L	L	L	L	L	L	L	L
-										L	L	L	L	L	L	L	L
-											L	L	L	L	L	L	L
12												L	L	L	L	L	L
-													L	L	L	L	L
-														L	L	L	L
-															L	L	L
16 to 18																L	L
-																	L
NOTE: 'L' means content is locked and needs PIN code to unlock																	

Table 40: Parental rating descriptor - overview of content block for each rating value

These rules for displaying values, allowed settings and when to block content are kept in *LG CTO – UPC Genre definition* [13] and *LG CTO – D4A UCNP Age Ratings* [14]. The information listed above is an excerpt of the relevant information and made available for easy reference.

5.2.14. Private data specifier descriptor (NIT, SDT, EIT)

The private data specifier descriptor (see table 41) is defined in *ETSI EN 400 468* [2], used to announce a following private descriptor. Therefore, it is always present in a table before any descriptor with tag value above 0x80. It can be used in the first loop of the NIT, the service loop of the SDT, or the first loop of the NIT.

Syntax	Nr. of bits	Identifier
private_data_specifier_descriptor() {		
descriptor_tag *0x5F*	8	uimsbf
descriptor_length	8	uimsbf
private_data_specifier	32	uimsbf
}		

Table 41: Syntax for the private data specifier descriptor

Semantics for the private data specifier descriptor:

private_data_specifier
In case user-defined private descriptors are used, this private data specifier is inserted in descriptor loops containing Ziggo, Nagra, Irdeto or other private descriptors. These values are used:

Private Data specifier	Comment
0x00000016	Value obtained from DVB authority. Used for private descriptors related to the iTV platform of former Ziggo.
0x00000028	Value obtained from DVB authority. Used for private descriptors related to EACEM.
0x00000600	Value obtained from DVB authority. Used for private descriptors related to the D4A and CI+ platform of former UPC
0x00000601	Value obtained from DVB authority. Used for private descriptors related to Nagra conditional access.
0x00000602	Value obtained from DVB authority. Used for private descriptors related to the Ziggo Horizon platform
0x00362275	Value obtained from DVB authority. Used for private descriptors related to Irdeto conditional access.

Table 42: Private data specifier - private_data_specifier values

NOTE: Details on the Irdeto and Nagra descriptors are secret and proprietary to Ziggo and therefore not part of this document.

If this descriptor is located within a descriptor loop of any SI table then its specifier applies to all following descriptors in the particular descriptor loop until its end, or until another occurrence of a private_data_specifier_descriptor, whichever comes first.

The effect of a private_data_specifier_descriptor in a descriptor loop does not propagate to any following descriptor loop. It needs to acknowledge that the private_data_specifier has no influence on standard DVB descriptors, whatever their relative positions may be.

5.2.15. Service descriptor (SDT)

The service descriptor (see table 43) is defined in *ETSI EN 400 468* [2], used to indicate the service type. It appears in the service loop of the SDT.

Syntax	Nr. of bits	Identifier
service_descriptor() {		
descriptor_tag *0x48*	8	uimsbf
descriptor_length	8	uimsbf
service_type	8	uimsbf
service_provider_name_length	8	uimsbf
for (i=0; i<N; i++) {		
service_provider_name_char	8	uimsbf
}		
service_name_length	8	uimsbf
for (i=0; i<N; i++) {		
service_name_char	8	uimsbf
}		
}		

Table 43: Syntax for the service descriptor

Semantics for the service descriptor:

service_type

This 8 bit field identifies the type of the service. The following service types may be encountered on different services:

Service_type	Description
0x01	MPEG-2 SD digital television service
0x02	Digital radio sound service
0x0A	advanced codec digital radio sound service
0x0B	H.264/AVC mosaic service
0x0C	Data broadcast service
0x11	HD MPEG2 digital television service
0x16	SD H.264/AVC digital television service
0x19	HD H.264/AVC digital television service
0x1F	HEVC digital television service (see note 1)
0x20	HEVC UHD digital television service with HDR and/or a high framerate (see note 2)
0x80	user defined
NOTE 1:	This value should be used for backward compatible HLG10 HDR services, and/or backward compatible high frame rate (HFR) services which are decodable by HEVC_UHDTV_IRD as defined in *ETSI TS 101 154* [15].
NOTE 2:	The "HEVC digital television with HDR and/or a frame rate of 100 Hz or 120 Hz" service type shall only be used for all the possible types of an HEVC video services with HDR and/or a frame rate of 100 Hz or 120 Hz that are not decodable by a HEVC_UHDTV_IRD as defined in *ETSI TS 101 154* [15].

Table 44: Service descriptor - service_type values

The service type is used to select the list that service belongs to, i.e. television or radio list of channels. It needs to be acknowledged that the list in table 46 is just a subset of the most common values. The complete list can be found in *ETSI EN 300 468* [2].

NOTE 3: If a channel does not have a "logical channel number", Ziggo DVB-C receivers insert it in an out-of-list list where it is not visible to the user.

NOTE 4: If several digital television services have the same "channel number", the selection of the digital television service for this "channel number" would be performed as the best viewable for the user (in other words, a High Definition decoder will choose an High Definiton service over an Standard Definition service).

service_provider_name_length

This 8 bit field specifies the number of bytes used in describing the name of the service provider.

service_provider_name_char

This is an 8 bit field. A string of "char" specifies the service provider. This field is ignored on Ziggo DVB-C receivers for cable.

service_name_length

This 8 bit field specifies the length of the character string describing the name of the service.

service_name_char

String of 8 bit characters specifying the service name. This is used for display purposes on Ziggo DVB-C receivers. Text information is coded using the character sets and methods described in annex A.

5.2.16. Service list descriptor (NIT)

The service list descriptor (see table 45) is defined in *ETSI EN 400 468* [2], used to provide a means of listing the services by service_id and service type. If broadcasted, it appears in the first loop of the NIT.

Syntax	Nr. of bits	Identifier
service_list_descriptor() {		
descriptor_tag *0x41*	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
service_id	16	uimsbf
service_type	8	uimsbf
}		
}		

Table 45: Syntax for the service list descriptor

Semantics for the service list descriptor:

service_id

This 16 bit field uniquely identifies a service within a TS. The service_id is the same as the program_number in the corresponding program_map_section, except that in the case of service_type = 0x04, 0x18 or 0x1B (NVOD reference services) the service_id does not have a corresponding program_number.

service_type

This 8 bit field specifies the type of the service. The values are identical to the service_type field of the service_descriptor described in chapter 5.2.15.

NOTE: This service list descriptor is ignored by Ziggo DVB-C receivers. Service types are derived from the service descriptor within the SDT. Descriptor is still carried in Ziggo networks for backward compatibility.

5.2.17. Short event descriptor (EIT)

The short event descriptor (see table 46) is defined in *ETSI EN 400 468* [2], used to provide the name of the event and a short description of the event in text form. If broadcasted, it appears in the event loop of the EIT P/F and EIT-S. It can appear as many times as there are languages.

Syntax	Nr. of bits	Identifier
short_event_descriptor() {		
descriptor_tag *0x4D*	8	uimsbf
descriptor_length	8	uimsbf
ISO_639_language_code	24	bslbf
event_name_length	8	uimsbf
for (i=0; i<N; i++) {		
event_name_char	8	uimsbf
}		
text_length	8	uimsbf
for (i=0; i<N; i++) {		
text_char *// not used*	8	uimsbf
}		
}		

Table 46: Syntax for the short event descriptor

Semantics for the short event descriptor:

ISO_639_language_code

This 24 bit field contains the *ISO 639-2* [9] three characters language code of the event_name_char and text_char loops.

event_name_length

This 8 bit field specifies the length of the character string describing the name of the event.

event_name_char

String of 8 bit characters specifying the name of the event to be used for display purposes on Ziggo DVB-C receivers. Text information is coded using the character sets and methods described in Chapter 8. Existing Ziggo DVB-C receivers do not display more than 32 characters.

text_length

This 8 bit field specifies the length of the following string of characters describing the text.

text_char

String of 8 bit characters specifying the text description for the event. Text information is coded using the character sets and methods described in annex A.

NOTE: The text_length and text_char fields are not used by Ziggo DVB-C receivers.

5.2.18. Subtitling descriptor (PMT)

The subtitling descriptor (see table 47) is defined in *ETSI EN 400 468* [2], used to describe a DVB subtitling stream. If broadcasted, it appears in the PMT.

NOTE: the value of stream_type for any PID carrying DVB subtitle shall be 0x06 (this indicates a PES carrying private data).

Syntax	Nr. of bits	Identifier
subtitling_descriptor() {		
descriptor_tag *0x59*	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
ISO_639_language_code	24	bslbf
subtitling_type	8	bslbf
composition_page_id	16	bslbf
ancillary_page_id	16	bslbf
}		
}		

Table 47: Syntax for the subtitling descriptor

Semantics for the subtitling descriptor:

ISO_639_language_code

This 24 bit field contains the *ISO 639-2* [9] three character language code of the language of the subtitle. Both ISO 639-2/B and ISO 639-2/T may be used.

EXAMPLE: The Dutch language has 3-character code "dut", which is coded as: '0110 0100 0111 0101 0111 0100'.

NOTE: Ziggo also uses "888" as unofficial language code for identification of teletext based subtitling (teletekst ondertiteling pagina 888).

subtitling_type

This 8 bit field provides information on the content of the subtitle and the intended display.

The following subtitling types are supported:

Subtitling_type	Description
0x10	Normal
0x20	Hard of hearing
Other values	Reserved

Table 48: Subtitling descriptor - subtitling_type values

5.2.19. Teletext descriptor (PMT)

The teletext descriptor (see table 49) is defined in *ETSI EN 400 468* [2], used to identify streams which carry EBU Teletext data inside an elementary stream with stream_type == 0x06 describing teletext type and page information.

Syntax	Nr. of bits	Identifier
teletext_descriptor() {		
descriptor_tag 0x56	8	uimsbf
descriptor_length	8	uimsbf
for (i=0; i<N; i++) {		
ISO_639_language_code	24	bslbf
teletext_type	5	uimsbf
teletext_magazine_number	3	uimsbf
teletext_page_number	8	uimsbf
}		
}		

Table 49: Syntax for the Teletext descriptor

Semantics for the teletext descriptor:

ISO_639_language_code

This 24 bit field contains the *ISO 639-2* [9] three character language code of the Teletext. Both ISO 639-2/B and ISO 639-2/T may be used.

teletext_type

This 5 bit field indicates the type of Teletext page indicated. The DVB-C receiver uses values 0x02 and 0x05 to detect subtitle page numbers.

Teletext_type	Description
0x00	Future use
0x01	Full page Teletext
0x02	Teletext subtitle page
0x03 to 0x04	unused
0x05	Teletext subtitle page for hearing impaired people
0x06 to 0x1F	Future use

Table 50: Teletext descriptor - Teletext_type values

teletext_magazine_number

This 3 bit field identifies the magazine number as defined in *ETSI EN 300 706* [16].

NOTE: Range 1 to 8. A value of 0 is referred to as belonging to magazine 8.

teletext_page_number

This is an 8 bit field giving two 4 bit hex digits identifying the page number as defined in *ETSI EN 300 706* [16].

EXAMPLE: page 234 is identified by: Magazine number == 2, Teletext page number == 0x34 (and not 34, 0x22).

ANNEX A CHARACTER ENCODING

Ziggo uses multiple types of character encoding in general, but for all texts in the PSI/SI two character code tables are used. In this Chapter the signaling of these tables and the tables themselves are given.

A.1 CONTROL CODES

Text fields can optionally start with non-spacing, non-displayed data which specifies the alternative character table to be used for the remainder of the text item. The following table lists these codes.

First byte value	Character code table	Table description
0x00	ISO/IEC 6937 [17]	Code table 00 - Latin Alphabet
0x01 to 0x04	NOT used by Ziggo DVB-C receivers	
0x05	ISO/IEC 8859-9 [18]	Code table 05 - Latin/Turkish Alphabet
0x05 to 0x0B	NOT used by Ziggo DVB-C receivers	
0x0C to 0x0F	Reserved for future use by DVB	
0x10 to 0x15	NOT used by Ziggo DVB-C receivers	
0x16 to 0x1F	Reserved for future use by DVB	

Table 51: Character encoding - first byte values

Ziggo DVB-C receivers support Table 00 and 05 provided in chapter A.2. If the first byte value is within 0x20 to 0xFF, the default Latin alphabet table is used, Table 00.

A.2 CHARACTER CODE TABLES

Colour coding is used to indicate the general glyph category:

Colour	Description
light orange	letters of the Latin alphabet which are compatible with 7 bit US-ASCII encoding
light red	numbers of the Latin alphabet which are compatible with 7 bit US-ASCII encoding
light blue	blue marks, punctuation, symbols, and separators
light pink	non-spacing symbols (diacritical marks)
light green	region-specific alphabet symbols

Non-printing characters use the following mnemonics:

Mnemonic	Description
SPC	space
NBSP	no-break space
SHY	soft hyphen
LRM	left-to-right mark
RLM	right-to-left mark

Character code table 00 (Latin alphabet with Unicode equivalents) is shown in table 52 and character code table 05 (Latin/Turkish alphabet with Unicode equivalents) is shown in table 53.

First Nibble	Second Nibble -0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-																
1-																
2-	SPC 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3-	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4-	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5-	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6-	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7-	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	
8-																
9-																
A-	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	€ 20AC	¥ 00A5		§ 00A7	¤ 00A4	‘ 2018	“ 201C	« 00AB	← 2190	↑ 2191	→ 2192	↓ 2193
B-	° 00B0	± 00B1	² 00B2	³ 00B3	× 00D7	µ 00B5	¶ 00B6	· 00B7	÷ 00F7	’ 2019	” 201D	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C-		̀ 0300	́ 0301	̂ 0302	̃ 0303	̄ 0304	̆ 0306	̇ 0307	̈ 0308		̊ 030A	̛ 0327		̋ 030B	̧ 0328	̌ 030C
D-	— 2015	¹ 00B9	® 00AE	© 00A9	™ 2122	♪ 266A	¬ 00AC	¦ 00A6					⅛ 215B	⅜ 215C	⅝ 215D	⅞ 215E
E-	Ω 2126	Æ 00C6	Đ 0110	ª 00AA	Ħ 0126		Ĳ 0132	Ŀ 013F	Ł 0141	Ø 00D8	Œ 0152	º 00BA	Þ 00DE	Ŧ 0166	Ŋ 014A	ŉ 0149
F-	ĸ 0138	æ 00E6	đ 0111	ð 00F0	ħ 0127	ı 0131	ĳ 0133	ŀ 0140	ł 0142	ø 00F8	œ 0153	ß 00DF	þ 00FE	ŧ 0167	ŋ 014B	SHY 00AD
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F

Table 52: Character code table 00 - Latin alphabet with Unicode equivalents

NOTE: This table is a superset of ISO/IEC 6937 [17] with addition of the Euro symbol (U+20AC) in position 0xA4.

.

First Nibble	Second Nibble -0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F
0-																
1-																
2-	SPC 0020	! 0021	" 0022	# 0023	$ 0024	% 0025	& 0026	' 0027	(0028	) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
3-	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
4-	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
5-	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C	] 005D	^ 005E	_ 005F
6-	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
7-	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	| 007C	} 007D	~ 007E	
8-																
9-																
A-	NBSP 00A0	¡ 00A1	¢ 00A2	£ 00A3	¤ 00A4	¥ 00A5	¦ 00A6	§ 00A7	¨ 00A8	© 00A9	ª 00AA	« 00AB	¬ 00AC	SHY 00AD	® 00AE	¯ 00AF
B-	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C-	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Æ 00C6	Ç 00C7	È 00C8	É 00C9	Ê 00CA	Ë 00CB	Ì 00CC	Í 00CD	Î 00CE	Ï 00CF
D-	Ğ 011E	Ñ 00D1	Ò 00D2	Ó 00D3	Ô 00D4	Õ 00D5	Ö 00D6	× 00D7	Ø 00D8	Ù 00D9	Ú 00DA	Û 00DB	Ü 00DC	İ 0130	Ş 015E	ß 00DF
E-	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	æ 00E6	ç 00E7	è 00E8	é 00E9	ê 00EA	ë 00EB	ì 00EC	í 00ED	î 00EE	ï 00EF
F-	ğ 011F	ñ 00F1	ò 00F2	ó 00F3	ô 00F4	õ 00F5	ö 00F6	÷ 00F7	ø 00F8	ù 00F9	ú 00FA	û 00FB	ü 00FC	ı 0131	ş 015F	ÿ 00FF
	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-A	-B	-C	-D	-E	-F

Table 53: Character code table 05 - Latin/Turkish alphabet with Unicode equivalents

NOTE: This table is compatible with ISO/IEC 8859-9 [18].

ANNEX B PROPRIETARY LANGUAGE CODES

Ziggo uses some ISO 639 language codes for alternate use and/or backward compatibility. The Ziggo defined language codes are listed in table 54.

639-2	Language	Purpose	Comment
ang	English, Old	Audio	Legacy use. Placeholder for primary Dutch audiotrack.
dut (B) nld (T)	Dutch	Audio	In some cases the 639-2/T code “nld” is used. This will be replaced by "dut" in near future.
ina	Interlingua	Audio	Auxilary language
gos*	--	Audio	Descriptive audio for the vision impaired people
mis	--	Audio	Uncoded languages
mul	--	Audio	Multiple languages
888*	--	Subtitles	Signals "page 888" teletext subtitles
NOTE:	Language codes marked with an asterix (*) are Ziggo defined language codes.		

Table 54: Proprietary ISO 639-2 language codes used by Ziggo