



100/407/PAS

DRAFT PUBLICLY AVAILABLE SPECIFICATION

Project number IEC/PAS 62298	
IEC/TC or SC 100	Secretariat the Netherlands
Distributed on 2001- 08-10	Voting terminates on 2001-10-12
Also of interest to the following committees	Supersedes document

THIS DOCUMENT IS STILL UNDER STUDY AND SUBJECT TO CHANGE. IT SHOULD NOT BE USED FOR REFERENCE PURPOSES.

RECIPIENTS OF THIS DOCUMENT ARE INVITED TO SUBMIT, WITH THEIR COMMENTS, NOTIFICATION OF ANY RELEVANT PATENT RIGHTS OF WHICH THEY ARE AWARE AND TO PROVIDE SUPPORTING DOCUMENTATION.

Title

EACEM Technical Report - TeleWeb Application

Note:

The IEC has received from EACEM a copyright release notice allowing the IEC to process this document as a PAS

EACEM Technical Report

TR-045-r01

Title:
TeleWeb Application Profile 1 Reference Decoder

Date: 23 April 2001

Version History

Version	Date	Author	Description
1.0	21 Dec 2000	Alexander Wass	First released version.
1.1	31 Jan 2001	Jan van Lier	Add relation between logo and certification.
2.0	5 Feb 2001	Jo Vandale	Adapting to the new EACEM references codes.
2.1 / r00	15 Feb 2001	Jo Vandale	Adding the new EACEM template (Cover, Header, Footer, ...)
3.0 / r01	23 April 2001	Jo Vandale	EACEM Project Team 1.4 approved version.

Contents

1 Introduction..... 4

2 References..... 4

3 Profile 1 Reference Decoder 4

 3.1 TeleWeb Reference Receiver..... 4

 3.2 TeleWeb Reference Browser 4

 Software Design..... 5

 3.3 Hardware Requirements..... 5

4 Logo connected to Profile 1 Reference Decoder..... 5

5 Contacts..... 5

6 Schematic Overview 6

1 Introduction

TARA Systems developed on request of the TeleWeb Project a Reference Decoder (Receiver & Browser) for TeleWeb profile 1. With the PC based GDI Display Simulator TeleWeb content can be displayed. This Profile 1 Reference Decoder is the basis for further porting activities to specific embedded hardware platforms. Further information can be found in reference [4]

2 References

- [1] EACEM TR-046: "TeleWeb Application Part 1, General Description".
- [2] EACEM TR-047: "TeleWeb Application Part 2, Profile 1, Enhanced".
- [3] EACEM TR-048: "TeleWeb Application Part 3, Delivery Methods".
- [4] TARA Systems GMBH: "Available Documentation": Software Specifications for GDI, HTML-DOM Implementation, Receiver, Memory Management, HTML Viewer, Application. Content Generation Guidelines

3 Profile 1 Reference Decoder

The TeleWeb Profile 1 Reference Decoder consists of a Reference Receiver and a Reference Browser. This Reference Decoder is platform independent and is the basis for further porting activities to specific embedded hardware platforms. A PC based Graphical Device Interface Display Simulator is available for displaying TeleWeb content. A TeleWeb Logo as pictured in section 4 is connected to this Reference Decoder. More information is given in references [1], [2], [3] and [4]. Documentation can be achieved via the contact addresses in section 5. A schematic overview of the Profile 1 Reference Decoder and GDI Display Simulator is given in section 6.

3.1 TeleWeb Reference Receiver

- Data transmission using IDL format B.
- Based on DVB Data Carousels
- FEC and CRC
- Prioritised File Database
- Pre-Filtering of pages

3.2 TeleWeb Reference Browser

- Implements a browser based on the requirements as specified in [2]
- Easy Navigation with left, right, up, down, select, colour and number keys
- Bookmark Management
- History Support

Software Design

- Object Oriented ANSI C Design
- Platform independent
- Prepared for easy porting to embedded 16/32-bit platforms
- PC based development with slicer and display simulator
- Automatic test system

3.3 Hardware Requirements

- 16 or 32-bit controller
- 640x480 display for content
- min. 188 colours
- ~ 500 Kbytes for code (including GDI and fonts)
- >5 Mbytes RAM for database
- 1 Mbytes RAM workspace

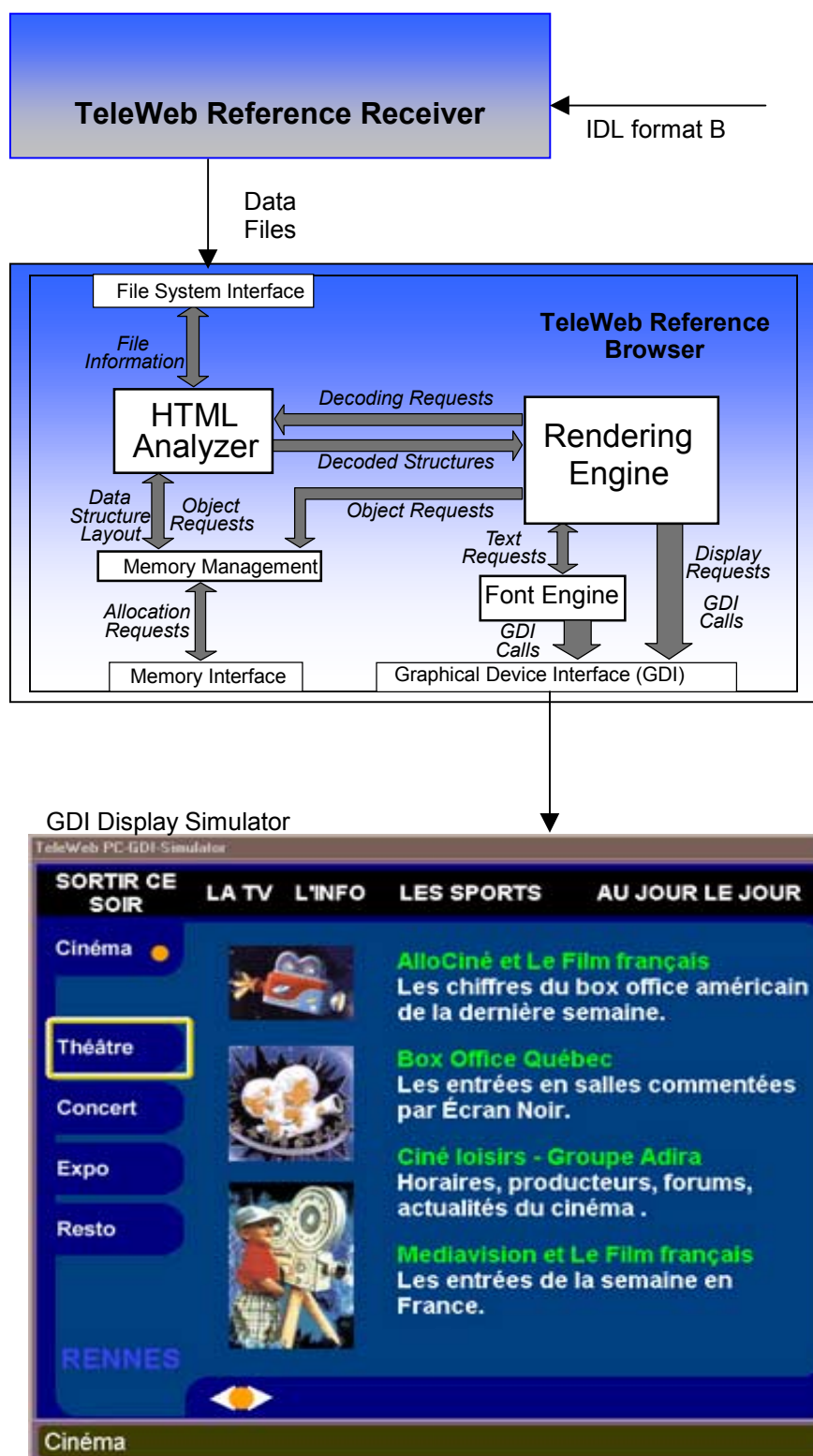
4 Logo connected to Profile 1 Reference Decoder



5 Contacts

<i>EACEM</i>	Brussels / Belgium	Internet address: http://www.eacem.be
<i>TARA Systems GmbH</i>	Munich / Germany	Internet address: http://www.tara-systems.de
<i>TeleWeb Project</i>		Internet address: http://www.superteletext.tv

6 Schematic Overview



EACEM Technical Report

TR-046-r02

Title:
TeleWeb Application Part 1, General Description

Proposed ETSI Title:
“TeleWeb Application Part 1, General Description”

Proposed ETSI keywords:
“TeleWeb, Superteletext, Browser, Enhanced, Internet, Interactive, Profile, Overview, Teletext, HTML”

Date: 18 May 2001

History

Ver.	Date	Author	Description
0.1	15 Feb 2000	David Tarrant	Initial split up from the original TeleWeb specification v1.6 in the TP1.4 meeting in Brussels
0.2	20 Juli 2000	Jo Vandale	Adaptations during the EACEM TP1.4 and TeleWeb meeting in Rennes
0.21	8 Aug 2000	Jo Vandale	Adaptations during the EACEM TP1.4 meeting in Brugge
0.22	31 Aug 2000	Jo Vandale	Minor adaptations during the TeleWeb meeting in Rousset
0.3	24 Oct 2000	Jo Vandale	Combining the requirement tables of profile 1.
0.4	2 Nov 2000	Jo Vandale	Adding the review remarks on version 0.3 by David Tarrant.
1.0	3 Nov 2000	Jo Vandale	Released version after review in the TeleWeb group.
1.9	4 Dec 2000	Jo Vandale	Rework after specification changes of the steering board.
2.0	11 Dec 2000	Jo Vandale	Released version after review in the Technical TeleWeb group.
2.1	31 Jan 2001	Jo Vandale	Released version after minor changes during the TM of January.
3.0	5 Feb 2001	Jo Vandale	Adapting to the new EACEM reference codes.
3.1 / r00	15 Feb 2001	Jo Vandale	Adding the new EACEM template (Cover, Header, Footer, ...)
4.0 / r01	23 April 2001	Jo Vandale	Adaptations resulting from the last Technical Meeting EACEM Project Team 1.4 approved.
4.1 / r02	5 May 2001	Jo Vandale	Making nexTView links optional as a result of the EACEM TC support group meeting.

Contents

1	Scope	12
2	References.....	12
3	Definitions and abbreviations	13
3.1	Definitions	13
3.2	Abbreviations.....	13
4	General description of TeleWeb.....	14
4.1	Aims.....	14
4.2	Overview.....	14
4.3	OSI seven-layer model.....	15
5	Documentation structure.....	15
6	Application Profiles.....	16
6.1	Profile 1 – Enhanced.....	16
6.2	Profile 2 - Interactive TeleWeb.....	17
6.3	Profile 3 – Internet TeleWeb	17
7	Display model.....	17
7.1	Display planes.....	18
7.1.1	Types.....	18
7.1.2	Overall sizes.....	18
7.2	Usable areas	18
7.2.1	Application Area.....	19
7.2.2	Content Area	19
7.2.3	User Interface Area	20
7.3	Display priority	20
8	Control model.....	20
8.1	User control device	20
8.2	Control functions	20
8.2.1	Selecting hypertext links	20
8.2.2	Primary link selection control	20

8.2.3 Mandatory functions 20

9 Referencing..... 21

10 General decoder architecture 21

(Blank Page)

Intellectual Property Rights

The application described here is intended to be an open standard, free of licensing and royalty restrictions for all parties.

Apart from the Unisys patent on the LZW compression scheme used in the GIF [4] graphics format and the licensing of the "Tiresias Screen" font from BITSTREAM [5], EACEM has not been informed of the existence of any Intellectual Property Rights (IPR) that could be, or could become essential to the present document. However, no investigation, including IPR searches, has been carried out. No guarantee can be given as to the existence of any IPRs that are, or may be, or may become, essential to the present document.

The licensing options to use the Tiresias font should be discussed with BITSTREAM [5] for each individual TeleWeb application. On a company level this licence requirement can be already covered due to other projects like DVB, MHP, ...

7 Scope

This document gives a general overview of the TeleWeb application that allows Web-style text and graphics to be broadcast to and displayed by suitable decoders.

TeleWeb services can be broadcast in a number of different ways, e.g. VBI, DVB, DAB, etc., and to a variety of decoder types, e.g. TVs, portable decoders, PCs, etc. These transmission protocols are described in separate documents.

8 References

- [4] CompuServe Incorporated: "Graphics Interchange Format, version GIF89a", July 1990.
- [5] BITSTREAM is the leading developer of font technology, digital fonts, and custom typeface designs for a wide variety of markets. Setting the standard of excellence in font technology, BITSTREAM holds key patents covering the creation of portable fonts for the Internet. www.bitstream.com
- [6] JPEG File Interchange Format Version 1.02 September 1, 1992
- [7] Digital Compression and Coding of Continuous Still Images
Part 1, Requirements and Guidelines
ISO/IEC JTC1 Draft International Standard 10918-1, Nov 1991
- [8] Digital Compression and Coding of Continuous Still Images
Part 2, Compliance Testing
ISO/IEC JTC1 Draft International Standard 10918-2, Dec 1991
- [9] ETSI, ETS 300 472: "Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bitstreams".
- [10] ETSI, EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting", V1.2.1 (1999-011)
- [11] ETSI, TR 101 202: "Implementation guidelines for data broadcasting", V1.1.1 (1999-02).
- [12] EACEM TR-045: "TeleWeb Application Profile 1 Reference Decoder".
- [13] ETSI, EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".
- [14] EACEM TR-037: Proposal for introducing a trigger mechanism into TV transmissions.
- [15] IETF RFC 1950 (1996): "ZLIB Compressed Data Format Specification version 3.3".

[16] ETSI TS 102 812: "Digital Video Broadcasting Multimedia Home Platform (MHP) Specification 1.1".

9 Definitions and abbreviations

9.1 Definitions

For the purposes of the present document, the following definitions apply:

Button: A part of the user interface that enables the viewer to select a page or trigger an event, etc. It may not necessarily exist as a physical button on a remote control handset.

Independent Data Line (IDL): A standalone Teletext packet containing both control and application data. It does not form part of a Teletext page. The packet address is either 30 or 31.

Semi-standby: A standby mode, currently known in the TV world, where the small-signal part of the set is powered to support acquisition and signal processing. The picture tube, audio power amplifiers and other large-signal parts are not powered. To the user the set appears to be switched off.

9.2 Abbreviations

For the purpose of the present document, the following abbreviations apply:

COP	Code Of Practice
DAB	Digital Audio Broadcasting
DARC	DAta Radio Channel
DSM-CC	Digital Storage Media Command and Control
DVB	Digital Video Broadcasting
EPG	Electronic Program Guide
ETSI	European Telecommunication Standard Institute
FM	Frequency Modulation
GIF	Graphics Interchange Format
HTML	Hyper Text Mark-up Language
IDL	Independent Data Line
IPR	Intellectual Property Rights
JPEG	Joint Picture Experts Group [6][7][8]
LZW	Lempel-Zif Welsh
MOT	Multimedia Object Transfer (DAB Protocol)
MPEG	Motion Picture Expert Group
OSI	Open System Interconnection
PPP	Point to Point Protocol
RGB	Red Green Blue
TCP/IP	Transmission Control Protocol / Internet Protocol
URL	Uniform Resource Locator
VBI	Vertical Blanking Interval
WAP	Wireless Application Protocol
WWW	World Wide Web
XHTML	Extensible Hyper Text Mark-up Language

10 General description of TeleWeb

1.1. Aims

The aim of TeleWeb is to deliver World Wide Web-style content to the living room TV to give the viewer an enhanced television experience without the inherent costs of connecting to the Internet via a modem and telephone line. A TeleWeb service broadcasts data files containing text and high-definition graphics to suitable decoders. The data transmitted can be closely linked to events within the accompanying TV programmes, or can be more general in nature to emulate a traditional, but higher definition, Teletext service. Hooks are left for future expansion and enhancement.

It is the intention that TV-based decoders can be implemented in a cost-effective manner without recourse to the technology normally associated with personal computers. In part, this is achieved by limiting the number of different types of multimedia data that can be used within a service. By careful design of the user-interface, decoder manufacturers will be able to offer easy to use equipment for accessing TeleWeb services without requiring the consumer to be computer literate. In addition, they will be able to customise their products to differentiate them from those of their competitors.

The encoding and transmission scheme is designed to be as efficient and robust as possible consistent with conveying potentially large data files via a unidirectional channel or bi-directional channel with a low or high data rate.

For example, TeleWeb data can be broadcast via Teletext packets using existing infrastructures. The TeleWeb data stream can be encoded into independent data packets that can be transmitted with minimal interference to existing Teletext services. In many instances it will be possible to recover otherwise wasted Teletext transmission capacity and the effect on existing services will be negligible.

It is possible to carry multiple services from different service providers on the same television channel. The fast transmission of services on analogue TV channels where there is no accompanying video component is also possible.

Three TeleWeb profiles are defined:

- Profile 1 - Enhanced Broadcast TeleWeb service without a return channel.
- Profile 2 - Interactive TeleWeb, extended from profile 1 with the addition of a return channel
- Profile 3 - Internet TeleWeb, extended from profile 2 offering full Internet access

Profiles 2 and 3 are designed to provide substantial backward compatibility with lower profile decoders.

10.1 Overview

A database of files is broadcast; some or all of which are captured and stored by a decoder. Certain files may be broadcast repetitively; others may be transmitted only once when they contain real-time updates or are linked to events in the accompanying TV programme. Each file has a set of attributes to define the file name, file type and other parameters as required. One of these is the theme or content description. This allows a decoder to be programmed to receive only specific information, or to exclude certain categories. This is useful where the volume of data transmitted exceeds the storage capacity available in the decoder.

On selecting a TeleWeb service, the viewer is first presented with the home page of the service. Navigation to other pages is via embedded links. It is possible that there may be more than one TeleWeb service on a given channel. Therefore each service includes additional information to allow a menu of available services to be presented to the user. The user interface is at the discretion of the decoder manufacturer, as is the provision of "bookmark" and "history" browser features.

1 **12 Application Profiles**

2 **12.1 Profile 1 – Enhanced**

3 This sub-clause lists the features of a TeleWeb Profile I service and a corresponding decoder. Each feature forms part of a
4 minimum specification for a service and/or decoder, unless marked as optional.

- 5 • Fixed content width and height (640 x 480 pixels).
- 6 • One proportional font, 5 sizes, 4 styles. The styles 'normal' is mandatory, the styles 'bold', 'italics' and 'bold & italics'
7 are optional.
- 8 • The “Tiresias Screen” font is used for the proportional font.
- 9 • One fixed font, 5 sizes, 4 styles. The styles 'normal' is mandatory, the styles 'bold', 'italics' and 'bold & italics' are
10 optional.
- 11 • Defined fixed font character cells (Height x Width in pixels).
- 12 • Support for all East and West European Latin-1 based languages.
- 13 • Minimum colour resolution of 12 bit (RGB = 444), 24 bit recommended (RGB 888).
- 14 • A minimum of 188 colours are defined in a colour palette (same as used by MHP [16]).
- 15 • Bit-mapped graphics using the GIF and JPEG formats [4][6][7][8].
- 16 • Dithering to achieve best colour matching (optional).
- 17 • Support for full and partial transparency.
- 18 • Animation using animated GIF and marquee
- 19 • Accentuation using blink (optional).
- 20 • Display of text and graphics over video.
- 21 • The content is authored using a TeleWeb HTML.
- 22 • Additional navigation possibilities via coloured (Fastext, TOP, ...) and numbered keys.
- 23 • Home page indication.
- 24 • Service identification.
- 25 • Content identification via themes.
- 26 • Age rating system based on the DVB-SI [13].
- 27 • Indication of a priority level (filtering possibilities in case of decoder memory restrictions).
- 28 • Control of primary link selection in the decoder.
- 29 • Decoder acquisition in semi-standby mode (Optional).

- 5 Mbytes of database broadcast capacity. If the broadcaster applies compression to the HTML pages and uses the most appropriate encoding for images, more pages can be broadcasted. This broadcast capacity is divided over 2 services, a 'Short TeleWeb Service' (100Kbytes) and a 'Full TeleWeb Service' (4.9Mbytes).
- Decoder must support one 'Preferred' Full TeleWeb Service.
- Decoder must support the Short TeleWeb Service on the currently tuned channel.
- Cross-linking between several services is supported through absolute TeleWeb URL's (Optional).
- Supporting multiple Full TeleWeb Services is optional.
- Decoder support for internal links within the same TeleWeb service.
- Decoders supporting Teletext should also support links from TeleWeb to Teletext, for all other decoders this is optional.
- Support of nexTView links from TeleWeb to nexTView (optional).
- Support of file compression based on ZLIB [15].
- Support of the trigger specification [14]
Restriction on the number of trigger streams and used transport layers will be defined in a COP.
- Decoder support for the processing of programme related information to build a TeleWeb EPG application (Optional).
- Support for group and individual decoder addressing.

12.2 Profile 2 - Interactive TeleWeb

Decoder profile as for profile 1 adding a back channel for registration, voting, etc.

- Support of the Document Object Model and ECMAScript.
- Support of HTML Frames.
- Encryption possibilities
- HTML Forms

12.3 Profile 3 – Internet TeleWeb

As profile 2 and offering full Internet access including limited browser plug-in capabilities.

13 Display model

This section defines the features of a TeleWeb display. They are applicable to both editing stations and decoders. The information is presented as a set of concepts to convey the general requirements. No particular method of realisation is implied.

The behaviour of a Profile 1 display should conform to the TeleWeb reference decoder as described in [12]. The latter should be used as a guideline to ensure conformity of all decoders in the market.

1 13.1 Display planes

2 13.1.1 Types

3 The conceptual TeleWeb "Display Model" consists of up to five aligned planes, Figure 3.

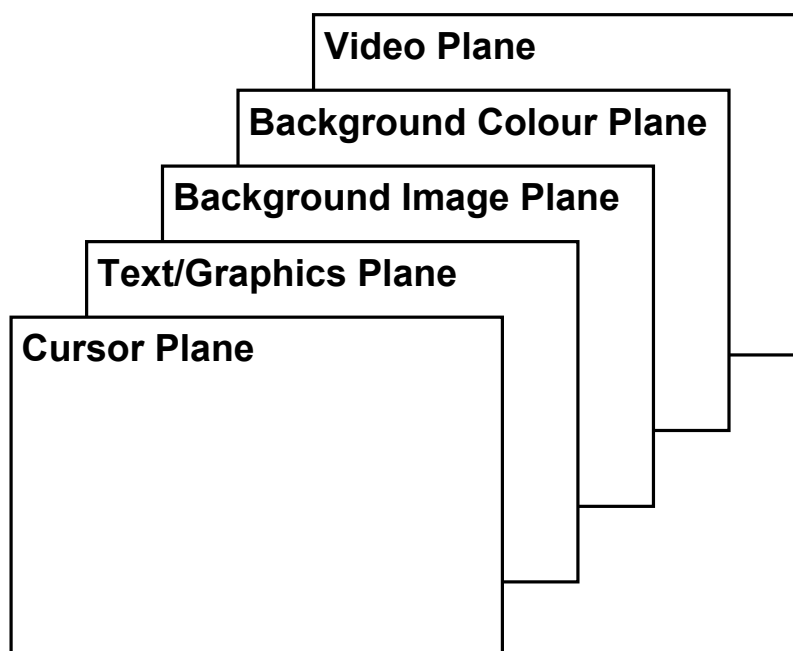
4 The **Cursor Plane** does not form part of this specification. It is an option for the decoder manufacturer and might be
5 included as part of the user interface. If present it is assumed to have the highest display priority.

6 The **Text/Graphics Plane** is used to display all visible foreground elements of a TeleWeb service. This includes the text
7 and image data defined and invoked within the Body section of an HTML file.

8 The **Background Image Plane** displays an image behind the foreground elements.

9 The **Background Colour Plane** displays a single uniform colour.

10 The **Video Plane** contains the video of any accompanying TV signal. This plane has the lowest display priority.



11
12 **Figure 3: Display planes and their priority order**

13 13.1.2 Overall sizes

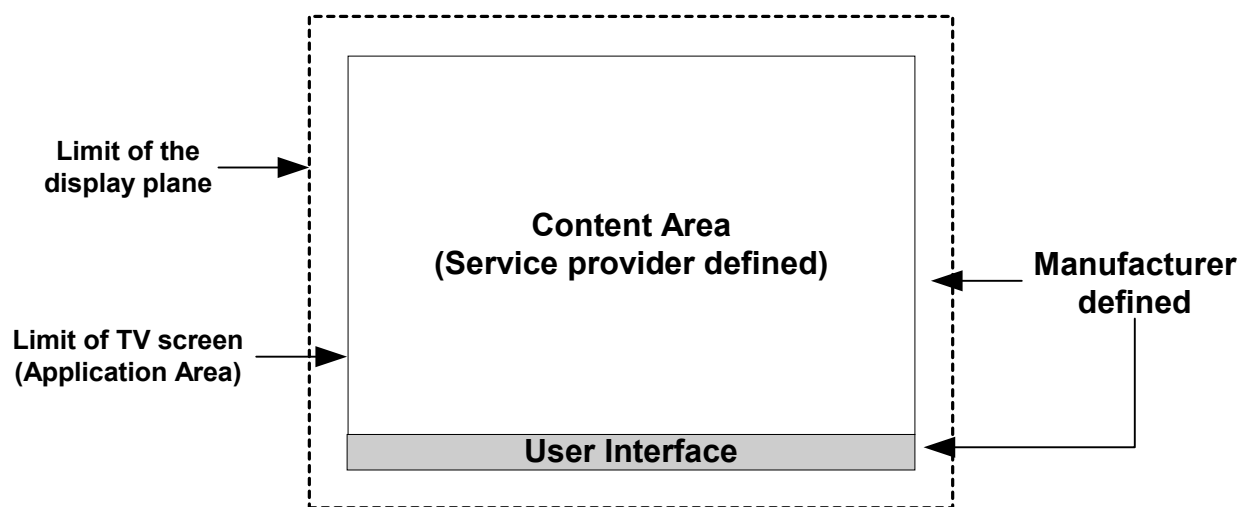
14 All planes are considered to be the same physical size. The Text/Graphics and Background Image planes are modelled as
15 720 x 576 pixels (Horizontal x Vertical) for reasons of compatibility with digital TV. The Background Colour plane can be
16 treated in the same way.

17 The Video Plane is capable of displaying the TV picture in full, regardless of the screen aspect ratio of the equipment.

18 13.2 Usable areas

19 The display area contains the TeleWeb content and the elements of the user interface. It is divided into Application, Content
20 and User Interface Areas.

1 An example of how the screen might be configured is shown in Figure 4. The design and contents of the user interface areas
 2 marked as “Manufacturer defined” are not specified by the current document. They can differ depending on the aspect ratio
 3 supported by the screen.



4
 5 **Figure 4: Screen layout example.**

6 **13.2.1 Application Area**

7 For information purposes only: The overscan found on most consumer TV receiving equipment has to be taken into
 8 account. Typically this is 5% at each border. This reduces the effective area of the Text/Graphics and Background planes to
 9 648 x 518 pixels. This "safe" area is referred to in this document as the **Application Area**. The Application Area will
 10 normally be available unless the equipment is configured in a non-standard way, perhaps when the underlying and visible
 11 4:3 format video has been manipulated to fill a 16:9 display.

12 **13.2.2 Content Area**

13 The **Content Area** is used for the display of the TeleWeb service. Its size is fixed at 640 x 480 pixels regardless of the
 14 screen aspect ratio of the equipment. Content should be authored to take account of these limits if uniformity of display
 15 across all decoders is considered to be essential. A decoder shall ensure that the Content Area is always visible in full within
 16 the Application Area unless the equipment is configured in a non-standard way. Its exact position relative to the boundaries
 17 of the Application Area is a manufacturer's option.

18 If a decoder is required to display content which exceeds the available space then the decoder may discard part of that
 19 content and should initially display the upper left part.

20 A background image that is smaller than the background image plane should be tiled to fill the available area.

21 The Content Area is under the control of the TeleWeb service provider, although a decoder manufacturer may choose to
 22 superimpose a cursor, or similar, within this area as part of the user interface. The contents of the Text/Graphics plane are
 23 determined by the data in the text and image files. This shall be regarded as "foreground" information for the purposes of
 24 determining display priorities. The background of an HTML table cell is also regarded as "foreground" information. The
 25 "background" to the Content Area can consist of elements from the Background Image, Background Colour and Video
 26 planes in any combination. The contents of the Background Image and Background Colour planes are determined by
 27 HTML attributes. Video is displayed if the transparent colour is selected on both planes. The priority mechanism is
 28 described in section 13.3.

29 Decoders with a screen aspect ratio of 16:9 can choose to render the TeleWeb content so that its 4:3 aspect ratio is
 30 maintained. Thus a circle in an image should still appear as a circle.

1 Alternatively, a decoder may choose to expand the TeleWeb content to fill the available space, thus distorting the page to
2 some extent.

3 **13.2.3 User Interface Area**

4 In principle, the entire display area outside of the Content Area is under the control of the decoder manufacturer and can
5 accommodate elements of the user interface. However, to ensure visibility, the manufacturer is likely to restrict the
6 foreground elements of the user interface to the area lying within the Application Area.

7 The service provider is able to indicate when the data is best displayed without any user interface text or graphics. If this
8 condition is indicated (Suppress User Interface flag set), the screen areas under the control of the decoder manufacturer
9 should be set to display video. However, it shall be permitted to display user-interface information temporarily, for
10 example, in response to a command from the user or to indicate navigation options.

11 **13.3 Display priority**

12 The display priority order of the planes shown in Figure 3 is in the order "Cursor" down to "Video". This order is important
13 as the use of the 100% transparent colour feature at a given position makes visible the plane immediately below.

14 In general, 100% transparency should be assumed when there is no explicit definition for the colour of a particular pixel
15 within the Content Area. For example, in the absence of a background image file but in the presence of <body
16 bgcolor=blue>, the background of the content area should be coloured blue overall.

17 It is possible to set an intermediate level of transparency, see TeleWeb Application Part 2. This semi-transparency only
18 takes effect when the lower plane is the video plane. If an intermediate level is selected elsewhere, the response of a decoder
19 is not defined.

20 **14 Control model**

21 **14.1 User control device**

22 The physical appearance of the control device and the method of interaction is not covered by this specification.

23 **14.2 Control functions**

24 **14.2.1 Selecting hypertext links**

25 The apparatus through which the user controls the decoder shall provide a means for choosing hypertext links displayed
26 within the Content Area. Conceptually, this may involve moving a cursor between the links and providing a method of
27 selection but other methods are not precluded (e.g jumping highlights).

28 **1.1.1. Primary link selection control**

29 To provide better support for interactive applications an URL can hold extra selection information in the URL's fragment
30 defining the primary selected anchor in the page referenced by the URL.

31 **14.2.2 Mandatory functions**

32 Dedicated **buttons** (or their equivalent) numbered 0 to 9 shall be provided on the user control device, as shall four other
33 **buttons** (or their equivalent) coloured red, green, yellow and blue. The coloured **buttons** shall be grouped in that order
34 from left to right or top to bottom.

1 The function of all these **buttons** shall be under the control of the service provider while a TeleWeb page is displayed in the
2 Content Area.

3 Conceptually, hypertext links can be mapped to any of these **buttons**.

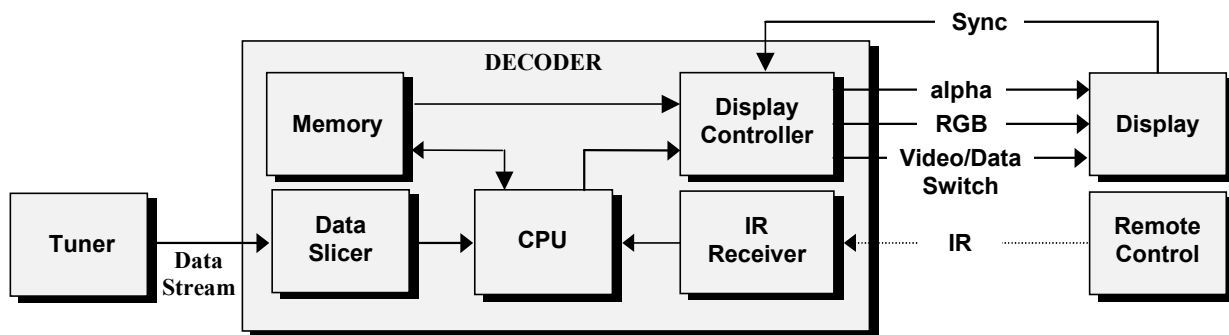
4 **15 Referencing**

5 Referencing to other services, like Teletext, nexTVView, Internet services, ... will be provided using the standard TeleWeb
6 URL mechanism defined in TeleWeb Application Part 2.

7 These references provide a way to reuse data which is already being transmitted (or is otherwise available) within the
8 context of the TeleWeb application.

9 **16 General decoder architecture**

10 The following drawing is an example of a TV-based Enhanced (P1) TeleWeb decoder architecture.



11
12 **Figure 5: Block diagram of a TV-based Enhanced TeleWeb Decoder**

1
2
3
4
5
6
7
8
9
10
11

EACEM Technical Report

TR-047-r01

Title:
TeleWeb Application Part 2 Profile 1 Enhanced

Proposed ETSI Title:
“TeleWeb Application Part 2 Profile 1 Enhanced”

Proposed ETSI keywords:
“TeleWeb, Enhanced, Superteletext, Teletext, HTML, Browser, Profile”

Date: 23 April 2001

History

Document history		
V0.0 - V0.4	Jan. – June 1998	First drafts. Very incomplete.
V1.0	June 1998	First full draft containing HTML description and display model.
V1.1	October 1998	Updated following Brussels meeting, 13/10/98
V1.2	January 1999	Updated following Hamburg (25/11/98) and Fellbach (15/12/98) meetings
V1.3	March 1999	Updated following Southampton meeting (11/2/99)
V1.4	July 1999	Updated following Paris (23/3/99) and Fellbach (20/5/99) meetings, and subsequent teleconferences up to 9/7/99.
V1.4.1	July 1999	Updated following Rousset meeting (19/7/99)
V1.5	December 1999	Updated following Redhill (30/9/99) and Rennes meetings (26/10/99) and interim telephone conferences
V1.6	February 2000	Updated following Grenoble meeting (25/1/00). Profile information added
V1.7	July 2000	First splitted Version Part II
V1.8	September and October 2000	Updated to reflect DTD and Style Sheet decisions
V1.9	October 2000	Merge new HTML part and latest changes of TeleWeb group
V2.0	November 2000	Updated following the Munich Meeting (2,3/11/2000)
V2.1	December 5, 2000	Updated following the Stuttgart Meeting (4,5/11/2000)
V2.2	8 Dec 2000	Updated according to the remarks of Manfred Schmidt
V3.0	5 Feb 2001	Adding the changes approved during the Nurnberg Technical Meeting (Jan 2001) Adapting to the new EACEM reference codes.
V3.1 / r00	15 Feb 2001	Adding the new EACEM template (Cover, Header, Footer, ...)
V4.0 / r01	23 April 2001	Adaptations resulting from the Technical Meeting EACEM Project Team Approved version.

Contents

1	Scope	31
2	References.....	31
2.1	TeleWeb.....	31
2.2	Character encoding	31
2.3	HTML.....	31
2.4	Image formats	32
2.5	Analogue video broadcasting.....	32
2.6	Digital video broadcasting	32
2.7	Miscellaneous	32
3	Definitions and abbreviations	33
3.1	Definitions	33
3.2	Abbreviations.....	33
4	Display.....	33
4.1	Colour representation.....	34
4.1.1	General requirements	34
4.1.2	Colour resolution	34
4.1.3	Gamma.....	34
4.1.4	Transparency.....	34
4.1.5	Colour palette.....	34
4.2	Text representation	35
4.2.1	Required sizes	35
4.2.2	Bold and Italic styles.....	35
4.2.3	Character repertoire	36
4.2.4	Default attributes.....	36
4.3	Text placement.....	37
4.3.1	Text width.....	37
4.3.2	Text-width calculation	37

4.3.3	Line breaks.....	37
4.3.4	Vertical line spacing	37
4.4	Image representation.....	37
4.4.1	Format.....	37
4.4.2	Animation	37
4.4.3	Use on the background image plane	37
4.5	4:3 and 16:9 aspect ratio displays	38
5	URLs.....	38
5.1	File naming	38
5.2	TeleWeb file reference scheme.....	39
5.2.1	Absolute URLs.....	39
5.2.2	Relative URLs.....	40
5.2.3	URL Query.....	40
5.2.4	Examples.....	41
5.3	Teletext page reference scheme	41
5.4	NexTVView reference scheme	42
5.5	Special function URLs	43
6	Text and hypertext	43
6.1	TeleWeb HTML file format.....	44
6.1.1	Tags and attributes	44
6.1.2	Text.....	44
6.1.3	File structure	45
6.2	Syntax of TeleWeb HTML tags.....	45
6.2.1	Guidelines for content providers: XHTML conformance.	45
6.2.2	Characters	46
6.2.3	Whitespace.....	46
6.2.4	Short tag.....	46
6.2.5	Attributes	46
6.3	Document structure elements	47
6.3.1	<!DOCTYPE> document type definition tag.....	47

6.3.2	<HTML> hypertext mark-up language tag	47
6.3.3	<HEAD> document header tag.....	48
6.3.4	<BODY> document body tag	48
6.3.5	<!-- --> Comment tag.....	50
6.4	Header section tags	50
6.4.1	<LINK> link structure tag.....	50
6.4.2	<TITLE> title tag.....	51
6.4.3	<BASE> tag.....	51
6.5	Paragraph formatting	51
6.5.1	<P> paragraph tag.....	51
6.5.2	 line break tag.....	52
6.5.3	<Hn> Heading tags, <H1> through <H6>	52
6.5.4	<DIV> division tag	53
6.5.5	<CENTER> centre text tag.....	53
6.5.6	<ADDRESS> address tag	53
6.5.7	<BLOCKQUOTE> Quoted passage tag.....	53
6.5.8	<HR> horizontal rule tag	54
6.5.9	<PRE> preformatted text tag	54
6.6	Character formatting	55
6.6.1	 font tag	55
6.6.2	<BIG> increment font size tag.....	55
6.6.3	<SMALL> decrement font size tag.....	55
6.6.4	<SUB> subscript tag	56
6.6.5	<SUP> superscript tag	56
6.6.6	<BLINK> flashing text tag	56
6.6.7	<STRIKE> strike through tag	56
6.6.8	<U> underline tag	56
6.6.9	 bold tag	56
6.6.10	 strong tag	57
6.6.11	<I> italics tag	57

6.6.12	<CITE> citation tag	57
6.6.13	<DFN> definition tag.....	57
6.6.14	 emphasis tag	57
6.6.15	<TT> Teletype, or monospaced, font tag.....	57
6.6.16	<CODE> program code tag	57
6.6.17	<KBD> keyboard input tag.....	57
6.6.18	<SAMP> sample tag	57
6.6.19	<VAR> variable tag.....	58
6.6.20	<BASEFONT> tag	58
6.7	Hypertext links.....	58
6.7.1	<A> anchor tag	58
6.7.2	<MAP> map tag.....	60
6.7.3	<AREA> area tag.....	60
6.8	Lists.....	63
6.8.1	 unordered list tag.....	63
6.8.2	<DIR> directory list tag.....	64
6.8.3	<MENU> menu list tag.....	64
6.8.4	 ordered list tag.....	64
6.8.5	 list item tag	65
6.8.6	<DL> definition list tag.....	65
6.8.7	<DT> term name tag	66
6.8.8	<DD> term definition tag.....	66
6.9	Tables.....	66
6.9.1	<TABLE> table tag	66
6.9.2	<CAPTION> table caption tag	68
6.9.3	<TR> table row tag	69
6.9.4	<TD> table data tag	70
6.9.5	<TH> table heading tag	72
6.10	Images.....	72
6.10.1	 image tag.....	72

6.11	Ticker Text	75
6.11.1	<MARQUEE> tag	75
6.12	Exceptional ignored tags	77
7	TeleWeb Default Style	77
8	Image files	77
8.1	GIF	77
8.2	JPEG	77
9	Content labelling	78
9.1	Predefined themes and identifier coding	78
9.2	Parental Ratings	78
10	Special data	78
10.1	Service Identification graphic	78
10.2	Home Page	78
10.3	Default Page	79
10.4	Profile Upgrade Page	79
10.5	ZLIB dictionary files	79
11	Service-related attributes	79
11.1	Name	80
11.2	Information	80
11.3	Language	80
11.4	Transmission Schedule	80
12	File-related attributes	81
12.1	General file attributes	81
12.1.1	Type	81
12.1.2	Name	82
12.1.3	CRC	82
12.1.4	Copyright	82
12.1.5	Encryption/Conditional Access	82
12.1.6	Version	82
12.1.7	Compression	82

12.2	File content attributes	83
12.2.1	Information	83
12.2.2	Theme	83
12.2.3	Parental Rating.....	83
12.2.4	Character Set (Encoding).....	83
12.2.5	Language.....	83
12.2.6	Suppress User Interface	84
12.2.7	Profile	84
12.3	Time and date related attributes.....	84
12.3.1	Creation Time	84
12.3.2	Start Validity.....	84
12.3.3	Expire Time	84
12.3.4	Repetition Distance.....	84
12.4	Storage related attributes.....	85
12.4.1	Priority	85
12.5	Special function attributes.....	85
12.5.1	Service Ident	85
12.5.2	Home Page.....	85
12.5.3	User Group ID	85
13	Short and Full TeleWeb Service	85
14	Individual Addressing - Group Addressing.....	86

(Blank Page)

Intellectual Property Rights

The application described here is intended to be an open standard, free of licensing and royalty restrictions for all parties.

Apart from the Unisys patent on the LZW compression scheme used in the GIF graphics format and the licensing of the "Tiresias Screen" font from BITSTREAM [46], EACEM has not been informed of the existence of any Intellectual Property Rights (IPR) that could be, or could become essential to the present specification. However, no investigation, including IPR searches, has been carried out. No guarantee can be given as to the existence of any IPRs that are, or may be, or may become, essential to the present specification.

The licensing options to use the Tiresias font should be discussed with BITSTREAM [46] for each individual TeleWeb application. On a company level this licence requirement can be already covered due to other projects like DVB, MHP, ...

17 Scope

This document specifies the TeleWeb Profile 1 application that allows Web-style text and graphics to be displayed on suitable decoders. A "TeleWeb" service comprises multimedia data files whose format and attributes are defined by this specification. This specification focuses on the presentation layer especially the implementation of TeleWeb HTML. It further defines the graphical requirements like colours and fonts and the used content formats. For information regarding general information and the transport layer refer to [17] and [18].

18 References

18.1 TeleWeb

[17] EACEM TR-046: "TeleWeb Application Part 1 General Description"

[18] EACEM TR-048: "TeleWeb Application Part 3 Delivery Methods"

18.2 Character encoding

[19] IETF RFC 2152 (1994): "UTF-7: A Mail-Safe Transformation Format of Unicode".

[20] IETF RFC 2279 (1998): "UTF-8, a transformation format of ISO 10646".

[21] "Information Processing – 8 bit single byte coded character sets – Part 1: Latin alphabet No 1", ".
ISO 8859-1-1:1987

18.3 HTML

[22] W3C, REC-htm32: "HTML 3.2 Reference Specification", 1997.

[23] W3C, RFC-html40: "HTML 4.0 Reference Specification", 1999.

[24] W3C: "XHTML 1.0: The Extensible HyperText Markup Language", 1999.

[25] Cascading Style Sheets, level 2 CSS2 Specification W3C Recommendation 12-May-1998
<http://www.w3.org/TR/REC-CSS2/>

1 **18.4 Image formats**

- 2 [26] CompuServe Incorporated: "Graphics Interchange Format, version GIF89a", July 1990.
- 3 [27] <http://members.aol.com/royalef/gifabout.htm>: "Royal Frazier's GIF animation on the WWW".
- 4 [28] JPEG File Interchange Format Version 1.02 September 1, 1992
- 5 [29] Digital Compression and Coding of Continuous Still Images
6 Part 1, Requirements and Guidelines
7 ISO/IEC JTC1 Draft International Standard 10918-1, Nov 1991
- 8 [30] Digital Compression and Coding of Continuous Still Images
9 Part 2, Compliance Testing
10 ISO/IEC JTC1 Draft International Standard 10918-2, Dec 1991

11 **18.5 Analogue video broadcasting**

- 12 [31] ETSI, ETS 300 706: "Enhanced Teletext Specification".
- 13 [32] ETSI, TS 101 231: "Television systems; Register of Country and Network Identification (CNI) and of
14 Video Programming System (VPS) codes".
- 15 [33] ETSI, ETS 300 231: "Specification of the domestic video Programme Delivery Control system (PDC)"
- 16 [34] ETSI, ETS 300 707: "Electronic Programme Guide
17 Protocol for a TV Guide using electronic data transmission, May 1997"
- 18 [35] EACEM TR-037: "Proposal for introducing a trigger mechanism into TV transmissions".

19 **18.6 Digital video broadcasting**

- 20 [36] ETSI, EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB
21 systems".
- 22 [37] ISO/IEC 13818-1: "Information technology - Generic coding of moving pictures and associated audio
23 information - Part 1: Systems - International Standard (IS)".

24 **18.7 Miscellaneous**

- 25 [38] ISO 639-2: "Codes for the representation of names and languages - Part 2: Alpha-3 code3".
- 26 [39] IETF RFC 1950 (1996): "ZLIB Compressed Data Format Specification version 3.3".
- 27 [40] IETF RFC 2396 (1998): "Uniform Resource Identifiers (URI): Generic Syntax".
- 28 [41] IETF RFC 2045 (1996): "Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet
29 Message Bodies".
- 30 [42] IETF RFC 2046 (1996): "Multipurpose Internet Mail Extension (MIME) Part Two: Media Types".
- 31 [43] IETF RFC 1738 (1994): "Uniform Resource Locators".
- 32 [44] IETF RFC 1808 (1995): "Relative Uniform Resource Locators".
- 33 [45] The design of a Typeface for subtitling on Digital Television Gill, J.H. Silver and C.Sharville, BBC,
34 January 1998

1 [46] BITSTREAM is the leading developer of font technology, digital fonts, and custom typeface designs for a
2 wide variety of markets. Setting the standard of excellence in font technology, BITSTREAM holds key
3 patents covering the creation of portable fonts for the Internet. www.bitstream.com

4 19 Definitions and abbreviations

5 19.1 Definitions

6 For the purposes of the present specification, the following definitions apply:

7 **button:** A part of the user interface that enables the viewer to select a page or trigger an event, etc. It may not necessarily
8 exist as a physical button on a remote control handset.

9 **CDATA:** Character data in an HTML document. Character entities and HTML mark-up is not recognised.

10 **Conditional Access (CA):** A mechanism by which user access to service components can be restricted.

11 **PCDATA:** Parsed character data in an HTML document. Character entities (numeric and named entities) as well as HTML
12 mark-up is recognised in the data.

13 **signed_integer:** A positive or negative integer value, in decimal notation. The first digit is preceded by a mandatory plus (+)
14 or minus (-) symbol. There shall be no white space between the symbol and the first digit.

15 **text_string:** A sequence of displayable Latin-1 characters.

16 **unsigned_integer:** An integer value, in decimal notation, without a preceding plus (+) or minus (-) symbol.

17 19.2 Abbreviations

18 For the purpose of the present specification, the following abbreviations apply:

19	ASCII	American Standard Code for Information Interchange
20	CA	Conditional Access
21	CDATA	Character Data
22	CRC	Cyclic Redundancy Check
23	DTD	Document Type Definition
24	DVB	Digital Video Broadcasting
25	ETS	European Telecommunication Standard
26	GIF	Graphics Interchange Format
27	HTML	Hyper Text Mark-up Language
28	JPEG	Joint Picture Experts Group
29	LSB	Least significant bit
30	LZW	Lempel-Zif-Welsh
31	MJD	Modified Julian Date
32	MSB	Most significant bit
33	PCDATA	Parsed Character Data between tags
34	RFC	Internet Request for Comments
35	URL	Uniform Resource Locator
36	UTC	Universal Time Co-ordinated
37	WWW	World Wide Web

38 20 Display

39 This section defines the minimum requirements of a TeleWeb display. They are applicable to both editing stations and
40 decoders.

1 **20.1 Colour representation**

2 **20.1.1 General requirements**

3 A decoder shall be able to display each pixel on the Text/Graphics, Background Image and Background Colour planes in a
4 different colour from a palette of at least 188 colours. Full transparency and one level semi-transparency to video shall also be
5 supported.

6 A decoder may be implemented with either a true-colour store or an indexed colour system. To accommodate the latter type
7 of decoder and to define a palette for GIF images that do not specify a palette themselves, a single 188 colour palette is
8 defined in section 20.1.5. As a minimum, the Text/Graphics, Background Image and Background Colour planes shall be able
9 to support colours subjectively equivalent to these colours. Content can be authored using other colours but authors should be
10 aware that some decoders may not be able to reproduce them exactly and may map such a colour to the closest match in the
11 pre-defined colour palette.

12 For authoring purposes the Text/Graphics, Background Image and Background Colour planes shall support at least 188
13 colours on the screen at any one time. In practical terms, a decoder is likely to be able to display at least 256 colours, giving
14 the equipment manufacturers at least 68 colours for their user interface.

15 To get an optimal display, the content should be authored using the TeleWeb default colour palette. The response of a
16 decoder is not defined by this specification if the colours used are not all taken from the pre-defined colour palette. Under
17 these circumstances colour dithering or matching techniques may need to be applied and the response of decoders may differ.

18 **20.1.2 Colour resolution**

19 Each colour shall be defined by Red, Green and Blue (RGB) components or by a colour constant, see section 22.2.5.1.
20 The authored content shall define colours as 24 bit values, i.e. 8 bits for each component in the order R, G, B.
21 A decoder is required to have a colour resolution of at least 4 bits per component (12 bits minimum overall).

22 **20.1.3 Gamma**

23 Decoders shall assume that all RGB values defined and invoked by authored content have been gamma pre-corrected for the
24 eye.

25 **20.1.4 Transparency**

26 Decoders are required to implement a minimum of three levels of transparency – opaque, semi transparency and completely
27 transparent.

28 The colour palette is optimised for 30% semi transparency. Where the decoder cannot implement the value of 30% semi-
29 transparency it shall replace it with the nearest value of semi-transparency it can implement. If the encoded value of
30 transparency is in the range 10%-90% it shall not be approximated as either 0% or 100% transparency. So, 9% may be
31 approximated as 0% but 10% shall be represented with a value in the range 10% to 90% such as 30%. Similarly, 91% may be
32 approximated as 100%.

33 **20.1.5 Colour palette**

34 The predefined set of 188 colours is shown in Table 1 and in Annex B. The colours chosen have a perceptually uniform
35 distribution over the colour space. A service provider may choose to author content using only this set of colours. A decoder
36 shall adopt this set of colours when rendering a GIF image if colour palette data is omitted from the file.

37 As a minimum, a decoder shall support these colours up to its resolution limit. It is then the responsibility of the decoder to
38 map the colours defined within the authored content to their closest approximation within the predefined set.

Table 1 Default colour palette

Transparency level	Total number of entries	Quantisation levels for Red	Quantisation levels for Green	Quantisation levels for Blue
0% (fully opaque)	135	5 levels: 0, 63, 127, 191, 255	9 levels: 0, 31, 63, 95, 127, 159, 191, 223, 255	3 levels: 0, 127, 255
0% (fully opaque)	4	4 additional grey scale colours: R = G = B = 42, 85, 170, 212		
30%	48	4 levels 0, 85, 170, 255	6 levels: 0, 51, 102, 153, 204, 255	2 levels: 0, 255
100% (fully transparent)	1	-	-	-
decoder definable	68			

Note: Levels are quoted as decimal values in the range 0 to 255.

To allow these colours to be used to render a GIF image from which the colour palette information has been omitted, it is necessary to assign a unique value to each colour as defined in Annex B.

20.2 Text representation

Text is rendered using one proportionally spaced font and one monospaced font, each in five sizes and in two styles - Plain and Bold. Italics and Bold Italics styles are optional. The proportional font is Tiresias [28]. The non-proportional font is not defined by this specification. Instead, and to ensure compatibility between equipment, the height and width of a character cell is defined for all sizes.

A service provider who wishes to ensure consistent displays must author content using the same metrics as implemented in decoders. Content shall be authored with kerning disabled.

20.2.1 Required sizes

Table 2 shows the supported font sizes for the proportional and monospaced font and the assignment to the size attribute of the HTML font tag. The font and character metrics including the width (advance) of all character cells for the proportional and the monospaced font for the different font sizes are specified in Annex F.

Table 2: Font sizes

Size [pixels]	HTML Font tag
22	SIZE = 1 or 2
24	SIZE = 3
27	SIZE = 4
31	SIZE = 5
36	SIZE = 6 or 7

20.2.2 Bold and Italic styles

The Tiresias font does not support the Bold style. A Bold version can be emulated by writing a second instance of the required character with an offset by one pixel to the right of the normal position. The Bold style should be restricted to small parts of the text e.g. headings. It should not be used as the standard for the whole text. The emulated Bold style may not be

1 acceptable for certain characters especially for the small font sizes (e.g. the double quote character or the characters with
2 umlaut may be critical). In this case the Bold style should be avoided for these characters.

3 For the proportional font the width (advance) of a character cell containing a Bold character may be increased by a maximum
4 of one pixel per character, as shown in Figure 1. Content shall be authored with this extra pixel space in mind. For the
5 monospaced font the width of all characters of a given font size is the same for all styles.

6 The implementation of the optional Italic and Bold Italic style is left open. However, the corresponding character cell width
7 shall always be the same as for the Plain resp. Bold style.

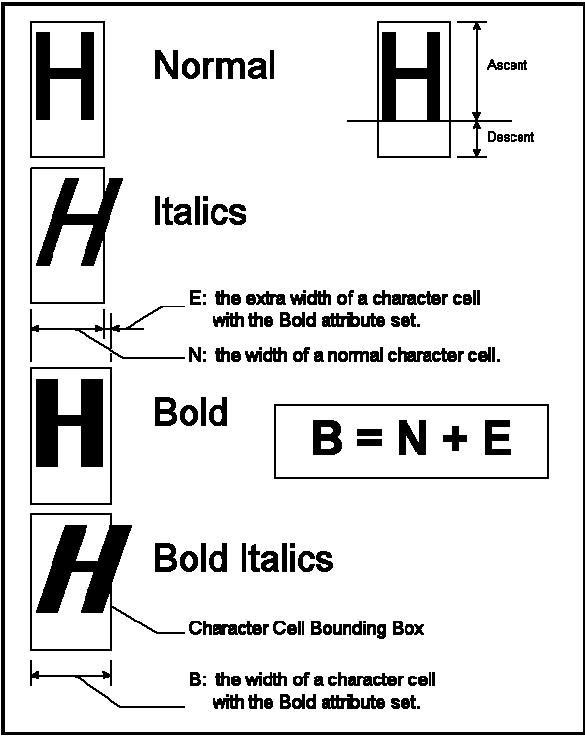


Figure 6: Character metrics

20.2.3 Character repertoire

Decoders shall support the full ISO Latin-1 character set [21] and the EURO sign character (Unicode 0x20AC). The EURO sign can be used by the name character entity (&euro) or the numerical representations (#&X20AC or #&8364). If the character encoding of the HTML file is different from ISO Latin-1 or the character is not defined in ISO Latin-1 and is not the EURO sign the decoder must display a replacement character represented by an empty square with the size of a capital 'V'.

Note: The characters 0x00 to 0x1F and 0x7F to 0x9F are not defined by ISO Latin-1 thus a replacement character will be shown instead.

20.2.4 Default attributes

In the absence of any font, size, foreground colour or background information within the authored content, a decoder is free to set its own default conditions. It is recommended that the decoder follow the default style defined in Annex E.

1 **20.3 Text placement**

2 **20.3.1 Text width**

3 To ensure that text will flow identically on different decoders and authoring equipment regardless of the quality of the
4 character rendering, simple algorithms are defined to determine vertical placement and when to wrap lines of text. The flow is
5 considered identical if lines and words break at the same character position. The calculations allow content creators to
6 provide sufficient space for their strings at authoring time. They make no demands of the particular rendering system
7 employed. The characters can be bit-map or vector fonts, aliased or anti-aliased, etc. The calculations shall be applied in both
8 authoring equipment and decoders.

9 **20.3.2 Text-width calculation**

10 The width calculation is defined in terms of the bounding rectangle of each character as defined within the font metrics shown
11 in Figure 6. Any extra width due to bolding shall be taken into account for the proportional font. The width of a text string is
12 the sum of the character widths. This calculation is carried out at pixel-resolution.

13 **20.3.3 Line breaks**

14 Once the text-width calculation defined in 20.3.1 produces a result that exceeds the space available, the first word after white
15 space (but not a non-breaking space (0xA0)), a hyphen (0x2D) or a soft hyphen (0xAD) that will not fit completely on the
16 line shall be rendered on the line below. This implies the equipment does not have to know or apply word hyphenation rules.

17 **20.3.4 Vertical line spacing**

18 The baseline of a font is determined by its ascent and descent metrics, see Figure 1. All characters are vertically aligned to
19 their baseline. The vertical space required for a line is the sum of the largest ascent and the largest decent in that line plus the
20 gap (external leading). The gap (external leading) between lines is set to -1 for all fonts and all sizes.

21 **20.4 Image representation**

22 **20.4.1 Format**

23 A decoder is required to implement the full GIF specification [4], apart from the "plain text" extension. A transmitted file may
24 include "plain text". If the GIF file does not define a colour palette the TeleWeb default colour palette will be used. In
25 addition the sequential baseline JPEG image format is supported [6][7][8].

26 **20.4.2 Animation**

27 Animation and looping of GIF images are supported as described in [4] and [27] respectively.

28 Note: There is no minimum time specified for the display of one frame when animating or looping. This will depend upon the
29 processing power in the decoder and the complexity and size of the images. Different decoders may show different response
30 speeds. The size of an animated GIF image may be restricted by a code of practice.

31 **20.4.3 Use on the background image plane**

32 An image that is smaller than the background image plane should be tiled to fill the available area.

33 A decoder is not required to reproduce animation on the background image plane. If multiple frames are present in the
34 designated file, the first frame should be displayed continuously.

20.5 4:3 and 16:9 aspect ratio displays

Content should be authored for a 4:3 display. Decoders with a screen aspect ratio of 16:9 can choose to render the TeleWeb content so that its 4:3 aspect ratio is maintained. Thus a circle in an image should still appear as a circle.

Alternatively, a decoder may choose to expand the TeleWeb content to fill the available space, thus distorting the page to some extent.

21 URLs

A uniform resource locator (URL) is a compact textual notation for accessing information within the TeleWeb system. The URLs within the TeleWeb system shall conform, except where otherwise noted, to the syntax for Internet URLs as specified in [40][43] and [44]. Note that the URL character set is limited to the lower-half of the ISO-Latin-1 character set [21]. To ensure compatibility with other browsers, URLs must encode character codes in the upper half.

A URL consists of a scheme and a scheme specific part. All characters in a URL belong to the ISO Latin 1 character set [21].

scheme : scheme-specific-part

In a complete URL the scheme is followed by a colon (':') and the string for the scheme specific part. Profile 1 decoders shall ignore schemes not listed in this section when such schemes appear in TeleWeb documents.

The scheme name is a string of characters. The only characters permitted in a scheme name are the lowercase characters 'a' through 'z', the digits '0' through '9', and the characters plus, hyphen, and full-stop, i.e. '+', '-', '.'. For robustness, software that interprets URLs (e.g. the TeleWeb Decoder) must map uppercase alphabetic characters to their corresponding lowercase characters.

The scheme-specific-part is a string of safe characters. Safe characters have a printable graphic representations in ISO Latin-1 [21]. Furthermore the set of safe characters is restricted to respect the reserved use of some punctuation characters. The alphabetic characters are at positions 41 hex through 5A hex, 61 hex through 7A hex, C0 hex through D6 hex, D8 hex through F6 hex, and F8 through FF hex. Thus only alphanumeric characters, characters in the string "\$-_.+!*'(),", and scheme-specific key characters from the string ";/?:@=&" may be used unencoded within a URL. Any data character, on the other hand, may be encoded. Encoded characters must be recognised and decoded by software that interprets URLs. A data character is one not used as punctuation by a scheme.

Data character encoding is a mapping of an 8-bit value to a 3-character sequence. The percent character, '%', indicates a start of the code. The next two characters are the hexadecimal value of the encoded character. For example, a space is encoded as %20 since a space has the code 32 decimal, which is 20 hexadecimal. A percent data character is encoded as %25 since the percent sign is at position 37 decimal (or 25 hex). An incomplete encoding sequence, e.g. %F=, has no specified meaning.

21.1 File naming

A Name attribute (section 28.1.2) shall be associated with each file. This attribute, a text string, must include both name and type (extension) elements, e.g. "file1.html", where "file1" is the name element and "html" is the type element. This name should be used to reference the corresponding file in any hypertext links. The file type element for HTML (text) files shall be either "htm" or "html". Table 3: File naming specifies naming conventions for the file types processed by TeleWeb.

Table 3: File naming

Content type	Filename extension	MIME type
HTML	.htm or .html	text/html
GIF	.gif	image/gif

JPEG	.jpg or .jpeg	image/jpeg
ZLIB Dictionary	.dic	application/x-zlib-dic

1

2 A Type attribute (section 28.1.1) shall be included with the file and shall indicate the type of the file e.g. for HTML
3 "text/html" . If a Type attribute is omitted or missing, a decoder should attempt to determine the file type by inspecting the
4 Name attribute. In the event of a conflict, the Type attribute shall have priority.

5 File names are not case sensitive.

6 21.2 TeleWeb file reference scheme

7 TeleWeb file reference URLs may be absolute or relative. An absolute URL specifies the service name of the content
8 provider, while a relative URL cannot. An absolute URL must specify a complete path to a resource, while a relative URL
9 can specify a partial path.

10 21.2.1 Absolute URLs

11 An absolute TeleWeb URL has the form:

12 *tw : // service-name / path-name [?query] [#fragment]*

13 where all uppercase characters are considered equivalent to their lowercase forms, and where portions within brackets may be
14 omitted.

15 The components are specified below:

16 **tw** : is the scheme name, which identifies the scheme specific part as a TeleWeb URL.

17 *// service-name* : identifies the name of the service. See Section 27.1.

18 */ path-name* : The path to the information, i.e. loosely the name of the file. Slashes within the path-name identify
19 hierarchical sections of the name, but do not require a specific data structure within TeleWeb
20 components, though an analogy to a file system is possible. The slash (/) is thus a reserved
21 character, and a data slash must be encoded as %2F if needed. Furthermore, the hierarchical section
22 names . and .. ("dot" and "dot-dot") are reserved, and data with these names may be specified with
23 %2E and %2E%2E if needed.

24 The dot-separator preceding the file type element is clearly data, since it is not itself a hierarchical
25 element. Therefore, a dot-separator does not have to be encoded, though it may, since it is
26 considered data when specifying a URL.

27 Once the hierarchical sections have been identified, all . sections are discarded, then all section-..
28 section pairs are discarded. Finally all remaining sections are recombined, separated by slashes to
29 give the resource name (file name). See also 21.2.4

30 *#fragment* : The fragment indicates the anchor within the accessed document to be focused when displaying the
31 linked document. The anchor in the linked document is defined by the corresponding name (for <A>
32 elements) or id attribute (for image maps). The indicated anchor shall be appropriately highlighted.

33
34 If the fragment is empty or invalid (not present in the target document, for example), the first
35 available anchor in the upper left corner shall be highlighted.

36 *?query* : The query defines additional parameters for the URL, see section 21.2.3.

21.2.2 Relative URLs

A relative TeleWeb file reference URL has the form:

[/] path-name [?query] [#fragment]

Where the meaning of the components is as was specified in the section on absolute URLs (21.2.1).

If the relative URL begins with a slash, then the path name is taken as absolute within the service used in the context where the link is found. If the relative URL does not begin with a slash then the service and the path is inherited from the link's context. If it is not empty, the URL's path name then replaces the last hierarchical section in the inherited path name. The resulting path name is evaluated as in the section on absolute URLs.

Anchor names and query parts are never inherited.

21.2.3 URL Query

The query part of a TeleWeb URL specifies additional parameters for the URL. Parameters are either keywords or keyword value pairs. Parameters are separated by semicolons. Keywords are separated from values by equal-signs. Thus ';' and '=' are reserved in the query part. The order and presence of keyword value pairs is arbitrary. The keyword value pairs that must be recognised are defined in the following sections. Any two parameters in a query must have different types.

21.2.3.1 The EPG parameters

These parameters can be used to realise a TeleWeb EPG.

Table 4: EPG Parameters

Parameter	Syntax	Format	Example
Network Operator	<i>cni=CNI_code</i>	4-digit hexadecimal value. The country and network identification (CNI) code for the broadcaster of the enclosed programme. If a three character VPS code has to be used, it shall be preceded with 0.	<i>cni=C380</i>
PIL	<i>pil=PIL_code</i>	5-digit hexadecimal value (20 bits). The PDC PIL for the programme. If the network conforms to VPS then this field contains the VPS label for the programme. If neither VPS nor PDC is supported, this field shall contain timer control codes as specified in ETS 300 231 [32].	<i>pil=FFFFE</i>
Start Time	<i>sta=start_time</i>	Defines the start date/time of the enclosed programme. The <i>time</i> format conforms to the ISO-8601 standard, except that it is assumed to be UTC. A recommended usage is the form <i>yyyymmddThhmmss</i> , where the capital letter "T" separates the date from the time. It is possible to shorten the time string by reducing the resolution. For example <i>yyyymmddThhmm</i> (no seconds specified) is valid.	<i>sta=20001103T2030</i>
Duration	<i>dur=time</i>	Defines the duration in number of minutes of the enclosed programme.	<i>dur=90</i>

1 If only the CNI parameter is defined the decoder should go to the channel broadcasting the CNI and stay there even if the
2 TeleWeb application is closed.

3 **21.2.3.2 The Profile parameter**

4 This parameter insures the upgrade compatibility to future higher profile services.

5 **Table 5: The Profile parameter**

Parameter	Syntax	Format	Example
Profile	<i>pro=profile</i>	The profile parameter indicates the profile of the referenced page. This parameter should not be defined if the link is referencing a profile 1 page. For a Profile 1 decoder, the presence of the profile parameter indicates that the link is referencing a higher profile page. As a result the decoder displays the profile upgrade page 26.4. The format of the parameter value will be defined in the specification documents for future profiles. Be aware that this parameter is only needed when the profile domain border is crossed from a lower to a higher profile.	<i>pro=...</i>

6
7 Note: The parameter value of the example can be any text excluding “;”.

8 **21.2.4 Examples**

9 This section is merely informative and does not specify required service names, path names or query parts.

```
10 <BASE                                HREF="tw://base/today/news/local">  
11 <A                                NAME=top></A>  
12  
13 a                                lot                                of                                text                                ...  
14  
15 <A NAME=ex-1 HREF="tw://sport/football.htm">Latest Football News</A>  
16 <A NAME=ex-2 HREF="sport/football.htm"> Today's Football Programme</A>  
17 <A NAME=ex-3a HREF="/tomorrow/programme.htm#football"> Tomorrow's Football Programme  
18 </A>  
19 <A NAME=ex-3b HREF=".../tomorrow/programme.htm#football"> Tomorrow's Soccer  
20 Programme </A>  
21 <A NAME=ex-4 HREF="#top"> Return to the top of the document </A>
```

22 Note that the document context is set by the BASE element to tw://base/today/news/local. Anchor EX-1 is an
23 absolute reference to the service "sport", page "football.htm". Anchor EX-2 is a relative reference equivalent to
24 tw://base/today/news/sport/football.htm. Anchors EX-3a and EX-3b are both equivalent to
25 tw://base/tomorrow/programme.htm#football, the football anchor on the programme.htm page. Anchor EX-4 is a
26 reference to the TOP anchor in the current document.

27 **21.3 Teletext page reference scheme**

28 A page in a Teletext service can be referenced by a hyperlink using the URL syntax:

```
29 ttx : // cni / page_number [ / page_subcode ]
```

1 where the sequence is not case sensitive and:

2 **ttx** : Identifies the scheme specific part as a Teletext page URL.

3 *// cni* : The country and network identification (CNI) code for the broadcaster as defined in [32]. If a three
4 character VPS CNI code has to be used, it shall be preceded with 0. The value 0000 shall be
5 interpreted as the TV channel of the TeleWeb service the URL is used in.

6 */ page_number* : A three hexadecimal character value in the range 100 to 8FF representing the magazine, page tens
7 and page units values respectively of a Teletext page as defined in [31].

8 */ page_subcode* : A four hexadecimal character value in the range 0000 to 3F7F representing the S4, S3, S2 and S1
9 values respectively of a Teletext page subcode as defined in [31]. The value 3F7F shall indicate that
10 no particular subcode value is being defined. S2 has a valid range of 0 to 7, and S4 a valid range of 0
11 to 3. The inclusion of a page_subcode element is optional.

12 On processing a link to a Teletext page, a decoder should display the most recent version of the page available. Handling of
13 rolling sequences of pages is at the discretion of the decoder manufacturer.

14

15 **21.4 NexTView reference scheme**

16 A nexTView Listing can be invoked by a hyperlink using the URL syntax:

17 **nextview** : *// cni ? filter-criteria*

18 Where the sequence is not case sensitive and:

19 **nextview** : Identifies the following sequence as a nexTView URL.

20 *// cni* : The country and network identification (CNI) code for the broadcaster as defined in [32]. If a three
21 character VPS CNI code has to be used, it shall be preceded by 0. The value 0000 shall be
22 interpreted as the TV channel of the TeleWeb service the URL is used in.
23
24 The CNI code uniquely identifies the nexTView service because there may be at most one nexTView
25 service per channel.

26 *? filter-criteria* : A list of keyword-value pairs. The pairs are separated by semicolons. A keyword is separated from
27 a value by an equals-sign. The order and presence of keyword-value pairs is arbitrary. The
28 keyword-value pairs that must be recognised are listed in Table 6. The interpretation of keyword
29 value pairs is specified in [34].

30 The decoder shall generate a listing according to the defined filter criteria. If no filter criteria are
31 defined it is up to the decoder where to enter the nexTView Application.

32 TeleWeb decoders are not required to support nexTView. In this case nexTView links are ignored by the decoder.

33

34 **Table 6: Filter Attributes for nexTView References**

Item	Syntax	Format	Example
Relative date	<i>rd=date_offset</i>	decimal value from 0 to 255 0 = today 1 = tomorrow :	<i>rd=0</i>

Item	Syntax	Format	Example
First programme	<i>fp=prog_offset</i>	decimal value from 0 to 255 0 = this programme 1 = next programme :	fp=0
last program	<i>lp=prog_offset</i>	decimal value from 0 to 255 0 = this programme 1 = next programme :	lp=1
Network operator	<i>nw=netwop_no</i>	decimal value from 0 to 79	nw=0
Theme	<i>th=theme</i>	hexadecimal value from 0 to FF conforming [34]	th=1F
Sorting criterion	<i>sc=sortcrit</i>	hexadecimal value between 0 and FF	sc=2A
Editorial rating	<i>er=editorial_rating</i>	decimal value from 0 to 15 conforming [34]	er=1
Parental rating	<i>pr=parental_rating</i>	decimal value from 0 to 15 conforming [34]	pr=4
start time	<i>sta=time_code</i>	4-digit hexadecimal value	sta=1230
stop time	<i>sto=time_code</i>	4-digit hexadecimal value	sto=1300
Features	<i>ff=feature_flags</i>	6-digit hexadecimal value with bits specified by [34]	ff=001001
Language	<i>la=language</i>	3-character language code defined in [38]	la=eng
subtitle language	<i>sl=subtitle_language</i>	3-character language code defined in [38]	sl=ger

Note: Each single type of filter must not be defined more than once in a query. If more than one different filter type are defined an AND operation is applied to the filters. An OR operation on filters is not supported. (for more information see [18]).

21.5 Special function URLs

To accommodate embedded browser control, three special function URLs are defined.

Table 7: Special function URLs

URL	Description
function:forward	Displays the next page in the history.
function:back	Displays the previous page in the history.
function:home	Displays the home page.

If one of the pages is not available no display action is executed.

22 Text and hypertext

The text content of a TeleWeb service is based on HTML [22][23]. The HTML tags and their attributes that are supported are described in this section.

1 Other HTML tags and attributes that are not listed in this section shall be ignored by decoders designed to this specification.
2 These are summarised in Annex A, but the DTD in Annex D is definitive. A decoder should only ignore the unsupported tags
3 and attributes and should still execute any valid text, tags or attributes enclosed by such tags. Unsupported tags and attributes
4 may appear in the transmitted files.

5 To assist in achieving uniformity of display across all decoders, the tag definitions in this section include mandatory display-
6 related aspects that do not appear in the HTML specification [22][23]. Additional rendering information can be found in the
7 TeleWeb default style sheet defined in Annex E.

8 **22.1 TeleWeb HTML file format**

9 The processing of a TeleWeb HTML file shall be limited to the constructs in the TeleWeb DTD, including character entity
10 references (both named and numeric character specification strings). Excluded from interpretation are SGML (meta-)
11 constructs such as Markup Declarations (e.g. `<! ... >`), Processing Instructions (e.g. `<? ... >`, including XML declarations and
12 tags) and Marked Sections (e.g. `<[CDATA[...]]>`). These excluded elements may be ignored or treated as data. Also
13 excluded are Function Character References (e.g. `&#SPACE`, `&#RS`, `&#RE`, etc.) and the encoding of functional characters as
14 character entity references. All character entity references should be interpreted as denoting data characters.

15 **22.1.1 Tags and attributes**

16 A TeleWeb HTML tag has the general form: `<tag_name attribute_1 attribute_2>` where *tag_name* identifies the tag,
17 and *attribute_1*, *attribute_2* ... are any number of modifying attributes, including zero. The angled brackets, `<` `>`, are
18 mandatory. In the tag definition in section 0, items that are optional, e.g. text, attributes or closing tags, are shown in
19 { braces }.

20 For some tags there is a complementary closing tag of the form: `</tag_name>`. This type of tag is referred to as a container.
21 In general, any tags occurring within the container have no effect outside of it. Tags without complementary closing tags are
22 called 'empty tags'.

23 Text strings entered as values for tag attributes, e.g. `NAME = text_string`, must be enclosed in quotation marks if the string
24 contains whitespace or special characters, e.g. `NAME = "text string"`.

25 Tags not specified in the TeleWeb DTD should be ignored.

26 Tags not specified within a container should be ignored within the container. Tags allowed in a container's enclosing context
27 may be assumed to close the container.

28 **22.1.2 Text**

29 Any text outside of a tag but within the body section (see section 22.3.4) is to be regarded as text to be displayed in the
30 Content Area. The appearance of this text will depend upon any preceding tags.

31 **22.1.2.1 Spaces and tabs**

32 Except within preformatted elements (`PRE`), decoders shall treat all sequences of spaces (0x20), tabs (0x09), newlines (0x0A)
33 and carriage returns (0x0D) within CDATA and PCDATA as a single space character.

34 Within attribute values, leading and trailing spaces shall be ignored.

35 Within preformatted elements, a tab shall be replaced by from 1 to 8 space characters such that the last space inserted shall be
36 the *n*th character on the line, where *n* is an integral multiple of 8.

37 Within preformatted elements, all space characters shall be treated as data (i.e. each shall appear in the output, there shall be
38 no merging). A newline, or a carriage-return, or a carriage-return and newline pair (whichever is the longest match) shall
39 signal a line feed / newline position.

22.1.3 File structure

Each TeleWeb HTML file shall contain a maximum of one document. The minimum document is an empty file.

The generic contents of an HTML file for TeleWeb applications are shown in Figure 7.

```
<!DOCTYPE HTML-TW PUBLIC "Profile 1.0">
<HTML>
  <HEAD>
    :
    :
    :
  </HEAD>
  <BODY>
    :
    :
    :
  </BODY>
</HTML>
```

(Header section)

(Body section)

Figure 7: Generic contents of an HTML file for TeleWeb use

The `<!DOCTYPE>` tag identifies the version of TeleWeb HTML in use. The correct syntax for services designed to this specification is given in section 22.3.1. The use of this tag is not mandatory for services designed to this specification.

The `<HTML>` and `</HTML>` tags enclose the header and the body sections. The use of these tags is not mandatory. A decoder should interpret the end of the file as equivalent to an `</HTML>` tag.

The `<HEAD>` and `</HEAD>` tags enclose the **header section**. The use of these tags is not mandatory. The header section contains data applicable to the entire document. This can include the title of the document. A decoder shall ignore any text immediately within the header section. To clarify: text immediately within the header section (i.e. PCDATA) occurs at the top level, i.e. it is not enclosed within a header element, e.g. TITLE. Thus, title text is not ignored, but any text characters immediately preceding the title start tag, or immediately following the title end tag, shall be ignored.

The `<BODY>` and `</BODY>` tags enclose the **body section**. The use of these tags is not mandatory. The body section contains the text and hypertext links to be displayed in the Content Area. In the absence of a BODY tag, the first occurrence of a tag defined in the DTD as being within the body section shall be interpreted by a decoder as the start of the body section.

22.2 Syntax of TeleWeb HTML tags

TeleWeb HTML tags and attributes described in this section are mandatory supported by every TeleWeb browser although the use of these tags and attributes may be optional in a TeleWeb HTML document.

22.2.1 Guidelines for content providers: XHTML conformance.

Where appropriate the syntax restrictions in the XHTML specification [24] should be applied.

Documents should be well-formed. The nesting should be correct.

Element and attribute names should be in lowercase characters. However, conforming browsers must accept upper- and lower-case element and attribute names equivalently.

For non-empty elements, end tags should be included. However, conforming browsers must treat omissible end tags correctly.

Attribute values should always be quoted. However, conforming browsers must treat unquoted numeric and alphanumeric values correctly.

- 1 Attribute names should always be present. However, conforming browsers must correctly accept unnamed attribute values
2 where the value is identical to the name and can have no other value, e.g. compact.
- 3 Empty tags should end with />. However, conforming browsers should not require this behaviour, nor should they be broken
4 by it.
- 5 Whitespace at the beginning or end of an attribute value should not be considered significant; furthermore a sequence of
6 whitespace characters within an attribute value should be considered equivalent to a single space character.
- 7 No CDATA sections should be put in documents, as they are not supported.
- 8 The ID attribute should be preferred to the deprecated NAME attribute for fragment identification. But for backward
9 compatibility, use NAME with an identical value in elements that declare a name attribute.

10 **22.2.2 Characters**

- 11 All tags shall only use characters from the ASCII character set.
- 12 Character entity references are always treated as data. This requirement is at variance with SGML usage, but is enforced to
13 simplify implementation and match the behaviour of existing Internet browsers.
- 14 Upper or lowercase characters can be used, and in any combination.
- 15 Keywords within tags (e.g. style, align) shall be written in full and only using the specified spellings.
16 Note: American spellings are used for certain keywords, e.g. color, center.

17 **22.2.3 Whitespace**

- 18 Whitespace is any combination of blank (0x20), tab (0x09), carriage-return (0x0D) and newline (0x0A) characters.
- 19 The tag (or element) name shall follow the tag open symbol ('<') immediately, without intervening whitespace. Attribute
20 names shall be preceded by whitespace and may be followed by whitespace. Attribute values may be preceded by
21 whitespace, and may be followed by whitespace.
- 22 Thus, for example, <P ALIGN = LEFT > is equivalent to <P ALIGN=LEFT>, but < P> is not allowed.

23 **22.2.4 Short tag**

- 24 The SHORTTAG feature is not supported. The only form of attribute minimisation supported is where the only legitimate
25 attribute value is identical to the attribute name, e.g. COMPACT = COMPACT may be written as just COMPACT. Examples of
26 unsupported notation include <html<head><title/hello/<body></></html>.

27 **22.2.5 Attributes**

- 28 All attributes are optional unless otherwise stated. There shall be a maximum of one attribute of each permitted type per tag.
29 Attributes can be listed in any order within a tag.
- 30 Unsupported attributes may appear in the tags, but these attributes should be ignored. In the case of an unknown attribute
31 value, the attribute should be ignored.

1 **22.2.5.1 Colour attributes**

2 Colours can be denoted in two ways:

- 3 1. by "#RRGGBB" where RR, GG and BB are case-insensitive hexadecimal representations of values in the range 00 to FF
4 that define the amplitude of the red, green and blue components respectively. Example: <BODY BGCOLOR = "#1E47DA">
- 5 2. by *standard_colour* where *standard_colour* is one of the named colours defined below. The colour names are not case
6 sensitive. Example: <BODY BGCOLOR = "Yellow">

7
8 **Named colours:**

9

10	BLACK	= "#000000"	GRAY	= "#808080"	SILVER	= "#C0C0C0"	WHITE	= "#FFFFFF"
11	MAROON	= "#800000"	RED	= "#FF0000"	PURPLE	= "#800080"	FUCHSIA	= "#FF00FF"
12	GREEN	= "#008000"	LIME	= "#00FF00"	OLIVE	= "#808000"	YELLOW	= "#FFFF00"
13	NAVY	= "#000080"	BLUE	= "#0000FF"	TEAL	= "#008080"	AQUA	= "#00FFFF"

14
15

16 Note: The standard HTML values are not supported by the TeleWeb default colour palette. Decoders that use only that
17 palette should substitute the hexadecimal value 7F for the 80 RR GG BB values given above.

18
19 Note: For robustness, the colour name GREY should be treated as equivalent to the standard colour GRAY.

20
21 Note: The predefined colour SILVER is not supported by the TeleWeb default colour palette. Decoders that use only that
22 palette should use the colour #D4D4D4 (colour index 138, see Annex B) instead.

23
24

25 **22.3 Document structure elements**

26 **22.3.1 <!DOCTYPE> document type definition tag**

27 **Function:** Defines the version of HTML used to author the document.

28 **Format:** <!DOCTYPE {version number}>

29 **Attributes:** None

30 **Use:** For files compatible with this specification, the tag shall be entered as:

31 <!DOCTYPE HTML-TW PUBLIC "Profile 1"> with whitespace between each string.

32 If present it shall be the first line of a file. It shall occur no more than once in a file. If this tag is present the browser should
33 discard every tag not supported by TeleWeb Profile 1.

34 **22.3.2 <HTML> hypertext mark-up language tag**

35 **Function:** Encloses an HTML document.

36 **Format:** <HTML> ... </HTML> (The start and end tags are optional.)

37 **Attributes:** None

38 **Use:** The inclusion of this tag is optional. No particular response is expected of a decoder if it is present. It shall occur no
39 more than once in a file. Text following the end tag shall be ignored.

1 **22.3.3 <HEAD> document header tag**

2 **Function:** Encloses the header section of the document.

3 **Format:** <HEAD> ... </HEAD> (The start and end tags are optional.)

4 **Attributes:** None

5 **Use:** The inclusion of this tag is optional. No particular response is expected of a decoder if it is present. It shall occur no
6 more than once in a file. A decoder shall ignore any text immediately within the header section. Text contained within the
7 title tag is not ignored, since the element is expected and contains the text itself.

8 **22.3.4 <BODY> document body tag**

9 **Function:** Encloses the body section of the document. Text within the body section shall be displayed in the Content Area.

10 **Format:** <BODY {attribute_1} {attribute_2} ... > ... </BODY> (start and end tag are optional.)

11 **Attributes:** ALINK, BACKGROUND, BGCOLOR, BOTTOMMARGIN, LEFTMARGIN, LINK, RIGHTMARGIN,
12 TEXT, TOPMARGIN, VLINK, TRANSPARENCY

13 **Use:** The inclusion of this tag is optional. No particular response is expected of a decoder if it is present. It shall occur no
14 more than once in a file. The contents of the body section can consist of an unordered collection of any tag defined starting
15 with sections 22.5 until the end of [section 22](#).

16 **ALINK attribute**

17 **Function:** Defines the foreground colour for a hypertext link at the moment it is activated while waiting for the new page to
18 be displayed.

19 **Format:** ALINK = *colour* The format of *colour* is defined in section 22.2.5.1.

20 **Use:** A decoder is allowed to treat the ALINK colour as a recommendation and can indicate that a link has been activated in
21 any way it chooses.

22 **BACKGROUND attribute**

23 **Function:** Defines the URL of an image file to be displayed on the Background Image plane.

24 **Format:** BACKGROUND = TeleWeb *url* The format of *url* is defined in section 21.2.

25 **Use:** In absence of this attribute, the background colour will be displayed (see BGCOLOR) in co-ordination with the
26 transparency value. Animation of background images is not supported.

27 **BGCOLOR attribute**

28 **Function:** Defines the colour to be displayed on the Background Colour plane.

29 **Format:** BGCOLOR = *colour* The format of *colour* is defined in section 22.2.5.1.

30 **Use:** The stated colour shall be used to fill completely that part of the Background Colour plane visible within the Content
31 Area. If this attribute is not defined, then it is up to the decoder to choose the background colour.

32 **BOTTOMMARGIN attribute**

33 **Function:** This attribute specifies the bottom margin for the entire body of the page in pixels and overrides the default
34 margin. When set to 0, the bottom margin is the bottom edge of the window the content is displayed in.

- 1 **Format:** BOTTOMMARGIN = *unsigned_integer*
- 2 **LEFTMARGIN attribute**
- 3 **Function:** This attribute specifies the left margin for the page in pixels, overriding the default margin. When set to 0, the left
4 margin is the left edge of the window the content is displayed in.
- 5 **Format:** LEFTMARGIN = *unsigned_integer*
- 6 **LINK attribute**
- 7 **Function:** Defines the foreground colour for a hypertext link that has not been visited and is not activated.
- 8 **Format:** LINK = *colour* The format of *colour* is defined in section 22.2.5.1.
- 9 **Use:** If this attribute is not defined, then it is up to the decoder to choose the colour.
- 10 **RIGHTMARGIN attribute**
- 11 **Function:** This attribute specifies the right margin for the page in pixels, overriding the default margin. When set to 0, the
12 right margin is the right edge of the window the content is displayed in.
- 13 **Format:** RIGHTMARGIN = *unsigned_integer*
- 14 **TEXT attribute**
- 15 **Function:** Defines the default foreground colour for the text of the document.
- 16 **Format:** TEXT = *colour* The format of *colour* is defined in section 22.2.5.1.
- 17 **Use:** If this attribute is not defined, then it is up to the decoder to choose the colour.
- 18 **TOPMARGIN attribute**
- 19 **Function:** This attribute specifies the top margin for the page in pixels, overriding the default margin. When set to 0, the top
20 margin is the top edge of the window the content is displayed in.
- 21 **Format:** TOPMARGIN = *unsigned_integer*
- 22 **TRANSPARENCY attribute**
- 23 **Function:** Defines the opacity of the background colour to underlying video.
- 24 **Format:** TRANSPARENCY = *unsigned_integer*
- 25 **Use:** If this attribute is not defined, then total opacity is assumed.
- 26 **VLINK attribute**
- 27 **Function:** Defines the preferred foreground colour for a hypertext link that has been visited.
- 28 **Format:** VLINK = *colour* The format of *colour* is defined in section 22.2.5.1.
- 29 **Use:** In case this attribute is not defined, it is up to the decoder to choose the colour. To increase manufacturers control over
30 look-and-feel, this value is only a recommendation.

1

2

3

4

5
6
7
8

9

10
11

12

13

14

15

16
17

18

19

20

21
22
23

24

25

26

27
28

29

30
31

32

33
34

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31

TITLE attribute

Function: Provides an advisory title for the linked resource.

Formats: TITLE = text *string*

The TITLE attribute may be ignored by a decoder.

22.4.2 <TITLE> title tag

Function: Encloses the title of the document.

Format: <TITLE> *title text* </TITLE>

Attributes: None

Use: The inclusion of this tag is required. No particular response is expected of a decoder if it is present. It shall occur no more than once in a file. The content of the TITLE element is PCDATA. This means that no further markup is allowed. However, character entities (= named characters) are supported. If a decoder chooses to present the title, it shall be displayed outside of the Content Area.

22.4.3 <BASE> tag

Function: Defines a base URL for resolving relative URLs. When present it shall appear before any HEAD section. The Start tag is **required** and the End tag is **forbidden**.

Format: <BASE HREF = *url* >

Attributes: Mandatory HREF element

Use: The inclusion of this tag is optional. It provides the base URL for dereferencing relative URLs.

HREF attribute

Function: Defines the address of the base.

Formats: HREF = *url*, see section 21.

22.5 Paragraph formatting

22.5.1 <P> paragraph tag

Function: The enclosed text is classified as a paragraph. Extra space is inserted at the end and at the start of a paragraph (see Section 23).

Format: <P {attribute} > </P> (The end tag is optional.)

Attribute: ALIGN

ALIGN attribute

Function: Defines the horizontal alignment of the paragraph relative to the current margins of the document. Alignment is only valid within the paragraph in which it has been defined.

1 **Format:** ALIGN = *position*

2 **position:** LEFT : The paragraph shall be rendered flush left;
3 RIGHT : The paragraph shall be rendered flush right;
4 JUSTIFY : The paragraph shall be rendered flush left and flush right by adjusting the width of the white space
5 between the words;
6 CENTER : The paragraph shall be centred.

7 **Use:** In the absence of an ALIGN attribute, the default alignment is LEFT.

8 **22.5.2
 line break tag**

9 **Function:** Forces a line break. The CLEAR attribute allows text to be positioned relative to images.

10 **Format:** <BR {attribute}>

11 **Attribute:** CLEAR

12 **Use:** The text immediately after the tag shall start on the following line. Existing alignment conditions are preserved. In the
13 absence of a CLEAR attribute, CLEAR = NONE shall be assumed.

14 **CLEAR attribute**

15 **Function:** Defines how the text should be positioned relative to images.

16 **Format:** CLEAR = *position*

17 **position:** LEFT : Clears text that flows around left-aligned images to the next full left margin
18 RIGHT : Clears text that flows around right-aligned images to the next full right margin
19 ALL : Clears text until it can reach both full margins
20 NONE : Introduces a "carriage return" and nothing more

21

22 **22.5.3 <Hn> Heading tags, <H1> through <H6>**

23 **Function:** These tags implement six levels of document headings; <H1> is the most prominent one, and <H6> is the least
24 prominent.

25 **Format:** <Hn {attribute} > ... </Hn> where n is an integer in the range 1 to 6 inclusive. The </Hn> closing tag is mandatory.

26 **Attributes:** ALIGN

27 **Use:** In the absence of the ALIGN attribute, the text shall be positioned LEFT. This can be overridden by an enclosed
28 <DIV> or <CENTER> tag. Nesting of heading tags is not permitted.

29 **ALIGN attribute**

30 **Function:** Defines the horizontal alignment of the enclosed text relative to the current margins. Alignment is only valid for
31 the heading in which it has been defined.

32 **Format:** ALIGN = *position*

33 **position:** LEFT : The heading shall be rendered flush left;
34 RIGHT : The heading shall be rendered flush right;
35 CENTER : The heading shall be centred.

JUSTIFY : The heading shall be rendered flush left and flush right by adjusting the width of the white space between the words;

22.5.4 <DIV> division tag

Function: Allows a document to be structured as a hierarchy of divisions.

Format: <DIV {attribute_1}> </DIV>

Attributes: ALIGN

ALIGN attribute

Function: Defines the horizontal alignment of the enclosed text relative to the current margins. Alignment is only valid for the division in which it has been defined.

Format: ALIGN = *position*

position: LEFT : The heading shall be rendered flush left;

RIGHT : The heading shall be rendered flush right;

CENTER : The heading shall be centred.

JUSTIFY : The division shall be rendered flush left and flush right by adjusting the width of the white space between the words;

22.5.5 <CENTER> centre text tag

Function: The enclosed text is displayed centred. (Equivalent to the <DIV ALIGN = CENTER> tag.)

Format: <CENTER> </CENTER>

Attributes: None

Use: The enclosed text is positioned centred relative to the current left and right margins.

22.5.6 <ADDRESS> address tag

Function: The enclosed text is (usually) a name, address and other contact information.

Format: <ADDRESS> ... </ADDRESS>

Attributes: None

Use: The enclosed text shall be treated as a paragraph.

22.5.7 <BLOCKQUOTE> Quoted passage tag

Function: The enclosed text is (usually) a quoted passage. It is presented as a paragraph, indented from both margins, and aligned flush left with a ragged right margin.

Format: <BLOCKQUOTE> </BLOCKQUOTE>

Attributes: None

Use: The enclosed text shall be treated as a paragraph.

1 **22.5.8 <HR> horizontal rule tag**

2 **Function:** Inserts a horizontal rule (line).

3 **Format:** <HR {attribute_1} {attribute_2} ... >

4 **Attributes:** ALIGN, COLOR, NOSHADE, SIZE, WIDTH

5 **ALIGN attribute**

6 **Function:** Defines the horizontal alignment of the enclosed rule relative to the current margins. Alignment is only valid for
7 the <HR> tag in which it has been defined.

8 **Format:** ALIGN = *position*

9 **position:** LEFT : The rule shall be rendered flush left;
10 RIGHT : The rule shall be rendered flush right;
11 CENTER : The rule shall be centred.

12 **COLOR attribute**

13 **Function:** Defines the foreground colour of the enclosed rule.

14 **Format:** COLOR = *colour*

15 **Colour:** The format of *colour* is defined in section 22.2.5.1.

16 **NOSHADE attribute**

17 **Function:** Forces the rule to be displayed as a solid bar, without shading.

18 **Format:** NOSHADE

19 **SIZE attribute**

20 **Function:** Defines the thickness (height) of the rule in pixels.

21 **Format:** SIZE = *unsigned_integer*

22 **WIDTH attribute**

23 **Function:** Defines the length of the rule, either as a number of pixels or as a percentage of the width between the current
24 margins.

25 **Formats:** WIDTH = *unsigned_integer* : number of pixels
26 WIDTH = *unsigned_integer*% : percentage of the width of the Content Area

27 **Use:** The default length is the width between the current left and right margins.

28 **22.5.9 <PRE> preformatted text tag**

29 **Function:** Renders the enclosed text in a monospaced font and preserves layout defined by whitespace and line break
30 characters.

31 **Format:** <PRE> ... </PRE>

32 **Attribute:** none

1 **Use:** The end tag is mandatory. Any "auto word wrap" function should be disabled. A horizontal tab character (0x09) should
2 be interpreted as the smallest non-zero number of spaces that will move the insert position along the line to the next column
3 position that is a multiple of eight.

4 **22.6 Character formatting**

5 **22.6.1 font tag**

6 **Function:** Defines the size and/or foreground colour of the enclosed text.

7 **Format:** ...

8 **Attributes:** COLOR, SIZE

9 **Note:** For robustness, it is desirable that decoders treat FONT as a block level element (see the DTD). Doing so will reflect
10 common usage and allow font changes to be applied to a document globally.

11 **COLOR attribute**

12 **Function:** Defines the foreground colour of the enclosed text.

13 **Format:** COLOR = *colour*

14 *colour:* The format of *colour* is defined in section 22.2.5.1.

15 **SIZE attribute**

16 **Function:** Sets the size of the enclosed text, either absolutely or relative to the size of the base font.

17 **Formats:** SIZE = *unsigned_integer* : Absolute font size. Valid range is 1 to 7, where 1 is the smallest size;
18 SIZE = *signed_integer* : A font size relative to the base font. The resulting size will be limited by the
19 range of 1 to 7.

20
21 **Examples:** To select font size 4:
22 To select a font two sizes smaller than the base font:

23 **22.6.2 <BIG> increment font size tag**

24 **Function:** The enclosed text is to be displayed one size bigger than the current font.

25 **Format:** <BIG> </BIG>

26 **Attributes:** None

27 **Use:** The font size is increased by one from that in effect at the position the <BIG> tag is encountered. If the current font size
28 is already the maximum available then there shall be no change of size. Following the </BIG> tag the font size reverts to the
29 previous value.

30 **22.6.3 <SMALL> decrement font size tag**

31 **Function:** The enclosed text is to be displayed one size smaller than the current font.

32 **Format:** <SMALL> </SMALL>

33 **Attributes:** None

1 **Use:** The font size is decreased by one from that in effect at the position the <SMALL> tag is encountered. If the current font
2 size is already the minimum available then there shall be no change of size. Following the </SMALL> tag the font size reverts
3 to the previous value.

4 **22.6.4 <SUB> subscript tag**

5 **Function:** The enclosed text is to be displayed as subscript text.

6 **Format:** _{....}

7 **Use:** The font size of the subscript text shall be one size smaller than the current font, if possible. The text shall be positioned
8 below the normal character baseline. Following the </SUB> tag the font size reverts to the original value. Nested <SUB> and
9 <SUP> elements should be avoided.

10 **22.6.5 <SUP> superscript tag**

11 **Function:** The enclosed text is to be displayed as superscript text.

12 **Format:** ^{....}

13 **Use:** The font size of the subscript text shall be one sizes smaller than the current font, if possible. The text shall be
14 positioned above the normal character baseline. Following the </SUP> tag the font size reverts to the original value. Nested
15 <SUP> and <SUB> elements should be avoided.

16 **22.6.6 <BLINK> flashing text tag**

17 **Function:** This tag defines flashing text.

18 **Format:** <BLINK> </BLINK>

19 **Use:** Only one phase of flashing text is supported. In the on state, the text shall be displayed in the current foreground colour.
20 In the off state, the colour of the foreground text shall be forced to be transparent so that the underlying background plane is
21 made visible. The on-off mark-space ratio is at the discretion of the decoder manufacturer. The total area of flashing text in
22 one page may be restricted by a code of practice.

23 **22.6.7 <STRIKE> strike through tag**

24 **Function:** The enclosed text is to be displayed as if struck out, e.g. ~~strike out~~.

25 **Format:** <STRIKE> </STRIKE>

26 **22.6.8 <U> underline tag**

27 **Function:** The enclosed text is to be displayed underlined, e.g. underline. The underline shall be in the same colour as the
28 text, and all characters, including whitespace, shall be underlined.

29 **Format:** <U> </U>

30 **22.6.9 bold tag**

31 **Function:** The enclosed text is to be displayed in bold face. See Section 23.

32 **Format:**

1 **22.6.10 strong tag**

2 **Function:** The enclosed text is to be displayed with strong emphasis. See Section 23.

3 **Format:**

4 **22.6.11 <I> italics tag**

5 **Function:** The enclosed text is to be displayed in italics. See Section 23.

6 **Format:** <I> </I>

7 **22.6.12 <CITE> citation tag**

8 **Function:** The enclosed text is to be displayed as is appropriate for a citation. See Section 23.

9 **Format:** <CITE> </CITE>

10 **22.6.13 <DFN> definition tag**

11 **Function:** The enclosed text is to be displayed as is appropriate for a definition. See Section 23.

12 **Format:** <DFN> </DFN>

13 **22.6.14 emphasis tag**

14 **Function:** The enclosed text is to be displayed with emphasis. See Section 23.

15 **Format:**

16 **22.6.15 <TT> Teletype, or monospaced, font tag**

17 **Function:** The enclosed text is to be displayed in a fixed-width font. See Section 23.

18 **Format:** <TT> </TT>

19 **22.6.16 <CODE> program code tag**

20 **Function:** The enclosed text is to be displayed as is appropriate for a sample of computer programme code. See Section 23.

21 **Format:** <CODE> </CODE>

22 **22.6.17 <KBD> keyboard input tag**

23 **Function:** The enclosed text is to be displayed as is appropriate for a sample of keyboard input. See Section 23.

24 **Format:** <KBD> </KBD>

25 **22.6.18 <SAMP> sample tag**

26 **Function:** The enclosed text is to be displayed as is appropriate for a sample of computer output. See Section 23.

1 **Format:** <SAMP> </SAMP>

2 **22.6.19 <VAR> variable tag**

3 **Function:** The enclosed text is to be displayed as is appropriate for a variable. See Section 23.

4 **Format:** <VAR> </VAR>

5 **22.6.20 <BASEFONT> tag**

6 **Function:** Used to set the base font size and colour.

7 **Format:** <BASEFONT>

8 **Attributes:** COLOR, SIZE

9 **Use:** There is no end tag. The base font size is determined by the SIZE attribute. It applies to normal and preformatted text
10 but not to headings, except where these are modified using the FONT element with a relative font size.

11 **COLOR attribute**

12 **Function:** This attribute sets the text colour.

13 **Format:** COLOR = *colour* The format of *colour* is defined in section 6.2.5.1.

14 **SIZE attribute**

15 **Function:** This attribute specifies the font size as either a numeric or relative value. Numeric values range from 1 to 7 with 1
16 being the smallest and 3 the default.

17 **Formats:** SIZE = *unsigned_integer*: Absolute font size. Valid range is 1 to 7, where 1 is the smallest size;
18 SIZE = *signed_integer*: A font size relative to the current base font. The resulting size will be
19 limited by the range of 1 to 7.

20 **22.7 Hypertext links**

21 See also the <LINK> tag, section 22.4.1.

22 **22.7.1 <A> anchor tag**

23 **Function:** Defines hypertext links to text or image files. The link can be external or internal to the document. Also used to
24 define named locations within documents for use as targets for hypertext links.

25 **Format:** <A {attribute_1} {attribute_2} .. > {text}

26 **Attributes:** HREF, NAME, REL, ACCESSKEY, TITLE, REV, ID

27 **Use:** Anchor tags cannot be nested. The foreground colour used to display the text enclosed by the tag is defined by the
28 ALINK, LINK and VLINK attributes of a <BODY> tag, depending on whether the link has been selected, not visited, or
29 visited respectively. In the absence of an appropriate attribute in the <BODY> tag, the decoder manufacturer can decide the
30 appearance of the link information but the use of the default style presented in annex E is encouraged.

31 **Examples:** Page 2 The text *Page 2* is a hyperlink to the file *page_2.htm*.

1 Section 3 The text *Section 3* is a defined location (anchor) within the
 2 document and can be referenced by both internal and external
 3 hyperlinks.

4 **HREF attribute**

5 **Function:** Defines the address of the link.

6 **Formats:** HREF = *url*, see section 21.

7 **Use:** When the hyperlink is selected, the anchor position is made visible within the Content Area.

8 **NAME attribute**

9 **Function:** Defines a unique name for an anchor.

10 **Formats:** NAME = *text_string*

11 **Use:** When a link is selected whose URL defines a fragment the anchor of the referenced file with a matching name attribute
 12 shall be visible. If the specified anchor also defines an HREF attribute the anchor shall be selected by the browser after the
 13 page is displayed. This feature enables full control over the link that should be selected first when coming from another page.

14 **REL attribute**

15 **Function:** Defines the relationship of an anchor or hyperlink to the current or target document. See Section 22.4.1.

16 **Formats:** REL = *link type*

17 **Use:** No usage behaviour is currently specified.

18 **ACCESSKEY attribute**

19 **Function:** Allows a link to be mapped to a particular **button** (or equivalent) on the user's control device.

20	Format: ACCESSKEY = R	Link	is	mapped	to	the	Red	key
21	ACCESSKEY = G	Link	is	mapped	to	the	Green	key
22	ACCESSKEY = Y	Link	is	mapped	to	the	Yellow	key
23	ACCESSKEY = B	Link	is	mapped	to	the	Blue	key
24	ACCESSKEY = 0..9	Link is mapped to a numeric key (0 - 9)						

25 **Use:** Specifies the hot-key that when pressed has the effect of selecting the containing anchor tag and following its hypertext
 26 link.

27 **TITLE attribute**

28 **Function:** Provides an advisory title for the linked resource.

29 **Format:** TITLE= *text_string*

30 **Use:** Can be used to supply descriptive text for the function of the anchor. This value may be ignored.

31 **REV attribute**

32 **Function:** Specifies an inverted REL link relationship. See REL.

33 **Format:** REV = *link type*

- 1 **ID attribute**
- 2 **Function:** Specifies a unique identifier for the anchor, similar to the NAME attribute.
- 3 **Format:** ID = name
- 4 **Use:** Used to specify cursor positioning information. The default value of the ID attribute is the value of the NAME
- 5 attribute.
- 6 **22.7.2 <MAP> map tag**
- 7 **Function:** Provides a mechanism for client-side image maps.
- 8 **Format:** <MAP NAME = *text_string* > </MAP>
- 9 **Attribute:** NAME
- 10 **Use:** The inclusion of a NAME attribute is mandatory. The enclosed elements must include at least one <AREA> tag. The
- 11 map should be in the same document as the image tag which references it.
- 12 **NAME attribute**
- 13 **Function:** Defines the name of the client-side image.
- 14 **Format:** NAME = *text_string*
- 15 **Use:** The name maps to a USEMAP attribute in an tag. A *text_string* is used to reference a location, it is case
- 16 insensitive in this context.
- 17 **22.7.3 <AREA> area tag**
- 18 **Function:** Specifies a hot spot within an image and binds it to a URL. The End tag is **forbidden**.
- 19 **Format:** <AREA {attribute_1} {attribute_2} ... >
- 20 **Attribute:** ALT, SHAPE, COORDS, NOHREF, HREF, ACCESSKEY, ID
- 21 **Use:** The NOHREF attribute takes precedence over the HREF attribute. RECT is the default SHAPE value. If the COORDS
- 22 value is missing, incomplete or invalid, then the AREA element is ignored.
- 23 **ALT attribute**
- 24 **Function:** Defines alternative text for the image map.
- 25 **Format:** ALT = *text_string*
- 26 **Use:** Can be displayed within the user interface area of the screen when the cursor is over the hotspot. The alt text may also
- 27 be displayed when the image is not available.
- 28 **SHAPE attribute**
- 29 **Function:** Used in conjunction with the COORDS attribute to define a hotspot region on the image.
- 30 **Format:** SHAPE = DEFAULT | RECT | CIRCLE | POLY

1 **Use:** In Profile 1 decoders, only rectangular shapes must be supported. Decoders may convert other shapes to the smallest
2 enclosing rectangle ("bounding box"). For robustness the values RECTANGLE (for RECT) CIRC (for CIRCLE) and
3 POLYGON (for POLY) should also be supported.

4 **COORDS attribute**

5 **Function:** This attribute contains a set of values specifying the co-ordinates of the hot-spot region. The number and meaning
6 of the values depend upon the value specified for the SHAPE attribute.

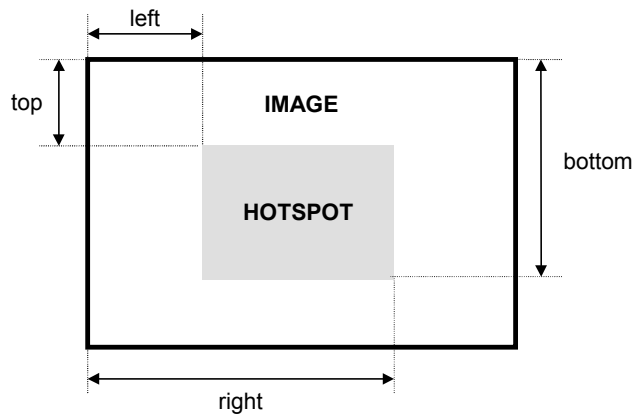
7 **Format:** COORDS = "*left, top, right, bottom*" Used when SHAPE = RECT

8 **left:** horizontal measurement from the top left corner of the image to the top left corner of the hotspot:
9 *unsigned_integer* : measurement in pixels;
10 *unsigned_integer%* : measurement as a percentage of the width of the image.

11 **top:** vertical measurement from the top left corner of the image to the top left corner of the hotspot:
12 *unsigned_integer* : measurement in pixels;
13 *unsigned_integer%* : measurement as a percentage of the height of the image.

14 **right:** horizontal measurement from the top left corner of the image to the top right corner of the hotspot:
15 *unsigned_integer* : measurement in pixels;
16 *unsigned_integer%* : measurement as a percentage of the width of the image.

17 **bottom:** vertical measurement from the top left corner of the image to the bottom right corner of the hotspot:
18 *unsigned_integer* : measurement in pixels;
19 *unsigned_integer%* : measurement as a percentage of the height of the image.



20
21 **Figure 8: COORDS attribute (SHAPE=RECT)**

22 **Format:** COORDS = "*centre-x, centre-y, radius*". Used when SHAPE = CIRCLE

23 **centre-x:** horizontal measurement from the top left corner of the image to the horizontal centre co-ordinate of the
24 circle-hotspot:
25 *unsigned_integer* : measurement in pixels;
26 *unsigned_integer%* : measurement as a percentage of the width of the image.

27 **centre-y:** vertical measurement from the top left corner of the image to the vertical centre co-ordinate of the
28 circle-hotspot:
29 *unsigned_integer* : measurement in pixels;
30 *unsigned_integer%* : measurement as a percentage of the height of the image.

31 **radius:** radius of the circle-hotspot. Note. When the radius value is a percentage value, browsers should
32 calculate the final radius value based on the associated object's width and height. The radius should be the
33 smaller value of the two.
34 *unsigned_integer* : measurement in pixels;

unsigned_integer% : measurement as a percentage of the width or height of the image.

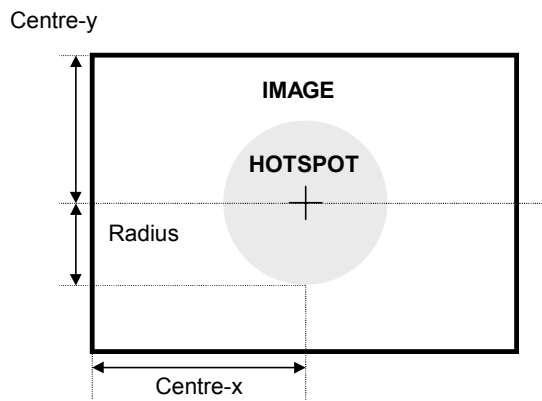


Figure 9: COORDS attribute (SHAPE=CIRCLE)

Format: COORDS = "x₁, y₁, x₂, y₂, ..., x_n, y_n"

Used when SHAPE = POLY. The co-ordinate pairs specify the vertices of the polygon.

x_i: horizontal measurement from the top left corner of the image to the vertex i of the polygon:

unsigned_integer : measurement in pixels;

unsigned_integer% : measurement as a percentage of the width of the image.

y_i: vertical measurement from the top left corner of the image to the vertex i of the polygon:

unsigned_integer : measurement in pixels;

unsigned_integer% : measurement as a percentage of the height of the image.

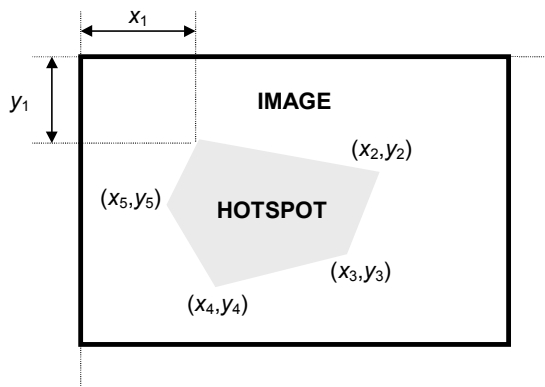


Figure 10: COORDS attribute (SHAPE=POLY)

href attribute

Function: Defines the destination URL for the hotspot.

Formats: See section 21.

nohref attribute

Function: Allows a region to be defined within the image but without a URL so that it does not act as a hotspot.

1 **Format:** NOHREF

2 **ACCESSKEY attribute**

3 **Function:** Allows a link to be mapped to a particular **button** (or equivalent) on the user's control device.

4	Format:	ACCESSKEY = R	Link	is	mapped	to	the	Red	key
5		ACCESSKEY = G	Link	is	mapped	to	the	Green	key
6		ACCESSKEY = Y	Link	is	mapped	to	the	Yellow	key
7		ACCESSKEY = B	Link	is	mapped	to	the	Blue	key
8		ACCESSKEY = 0..9	Link is mapped to a numeric key (0 - 9)						

9 **ID attribute**

10 **Function:** Specifies a unique identifier for the AREA element.

11 **Format:** ID = *text_string*

12 **Use:** Used to specify cursor positioning information.

13 **22.8 Lists**

14 Three kinds of lists are supported: ordered, unordered and definition. The nesting of lists is permitted. Nested lists may be of
15 the same type or of a mix of types.

16 **22.8.1 unordered list tag**

17 **Function:** Displays text as a bulleted list. Each list item is preceded by a bullet. The bullets are either discs, squares or circles
18 depending on the level. See section 23.

19 **Format:** <UL {attribute} > {list elements}

20 **Attributes:** TYPE

21 **Use:** Items to appear in the list are preceded by tags, see section 22.8.5. All list items shall be indented relative to the
22 current left margin. A bullet shall be displayed within the indent region before each list item. In the absence of a TYPE
23 attribute, the bullet style depends on the nesting level of this unordered list. See Section 23. A tag closes an open
24 tag.

25	Example:	HTML		DISPLAY	
26					
27		 Item 1	●	Item	1
28		 Item 2	●	Item	2
29		 Item 3	●	Item	3
30					

31 **TYPE attribute**

32 **Function:** Defines the type of bullet to be used in an unordered list.

33 **Formats:** TYPE = *style_information*

34	style_information:	DISC	Example:	●
35		SQUARE	Example:	■
36		CIRCLE	Example:	○

1 **Use:** The actual appearance of a bullet is not defined by this specification, but the bullet shall fit within the space occupied by
2 a capital 'V' in the current font, style and size.

3 **22.8.2 <DIR> directory list tag**

4 To be interpreted in the same way as the unordered list tag , section 22.8.1.

5 **22.8.3 <MENU> menu list tag**

6 To be interpreted in the same way as the unordered list tag , section 22.8.1.

7 **22.8.4 ordered list tag**

8 **Function:** Displays text as an ordered list. Each list item is preceded by a sequence number or character. A variety of
9 sequencing styles are possible. See section 23.

10 **Format:** <OL {attribute_1} {attribute_2} ... > {list elements}

11 **Attributes:** START, TYPE

12 **Use:** Items to appear in the list are preceded by tags. All list items shall be indented relative to the current left margin.
13 The sequence symbol shall be displayed right aligned just in front of the list item. In the absence of a TYPE attribute,
14 TYPE = 1 shall be assumed (i.e. Arabic numbering). In the absence of a START attribute, START = 1 shall be assumed. The
15 sequence symbol is increased by one for successive list items unless an tag contains a VALUE attribute. See Section 23.
16 A tag closes an open tag.

17	Example:	HTML		DISPLAY	
18		<OL	TYPE	=	A>
19		 Item 1	A.	Item	1
20		 Item 2	B.	Item	2
21		 Item 3	C.	Item	3
22					

23 **TYPE attribute**

24 **Function:** Defines the sequencing scheme.

25 **Formats:** TYPE = *style_information*

26	style_information:	1	:	Arabic numbers	1,	2	,3,	...
27		a	:	Lower alpha	a,	b,	c,	...
28		A	:	Upper alpha	A,	B,	C,	...
29		i	:	Lower roman	i,	ii,	iii,	...
30		I	:	Upper roman	I, II, III, ...			

31 **Use:** The initial value is specified by a START attribute. If absent, the first value shown above should be used as a default.

32 **START attribute**

33 **Function:** Defines the initial value for the sequencing scheme.

34 **Formats:** START = *unsigned_integer*

35 **Use:** The integer value defines a positive offset from the start of the sequence, where START = 1 implies the first symbol in
36 the sequence.

1 **Example:** <OL TYPE = A START = 3> indicates the first symbol in the list shall be preceded with the symbol C.

2 **22.8.5 list item tag**

3 **Function:** Used to indicate individual items or entries within a list.

4 **Format:** <LI {attribute} > {text} (The closing tag is optional.)

5 **Attribute:** VALUE, TYPE

6 **Use:** A bullet point or sequence symbol (depending on the type of list) shall be inserted before the text. The text is indented.

7 See Section 23. An open tag is closed by another tag or a closing tag of the current list type (, , etc.).

8 **VALUE attribute**

9 **Function:** Overrides the automatic sequencing of an ordered list.

10 **Formats:** VALUE = *unsigned_integer*

11 **Use:** Only valid when the tag is used within an ordered list. The current list item adopts the new value and subsequent

12 items are preceded with a symbol incremented from this value.

13 **TYPE attribute**

14 **Function:** Defines the sequencing scheme. See Section 22.8.4

15 **Formats:** TYPE = *style_information*

16	style_information:	1	Arabic numbers	1,	2	,3,	...
17		a	Lower alpha	a,	b,	c,	...
18		A	Upper alpha	A,	B,	C,	...
19		i	Lower roman	i,	ii,	iii,	...
20		I	Upper roman	I,	II,	III,	...
21		DISC	Example:				●
22		SQUARE	Example:				■
23		CIRCLE	Example:				○
24							

25 **22.8.6 <DL> definition list tag**

26 **Function:** Defines a definition or glossary list in which the description terms (DTs) are displayed beginning in the first

27 column and the description definitions (DDs) are displayed in a second column farther to the right.

28 **Format:** <DL> </DL>

29 **Attribute:** none

30 **Use:** Used in conjunction with the <DT> and <DD> tags. Formatting indicates the two roles. See Section 23.

31	Example:	HTML	DISPLAY
32		<DL>	
33		<DT>Point 1<DD>Approve agenda	Point 1
34		<DT>Point 2<DD>Previous minutes	Approve agenda
35		:	Point 2
36		</DL>	Previous minutes

1 **22.8.7 <DT> term name tag**

2 **Function:** This tag denotes the term portion of an item within a definition list (<DL>).

3 **Format:** <DT> </DT> (The end tag is optional.)

4 **Attributes:** None

5 **Use:** The text is not displayed indented. It is displayed on a separate line from any following <DD> tags. See Section 23.

6 This tag shall only be used within a definition list. An open <DT> tag is also closed by another <DT>, <DD> tag or a </DL>

7 tag. The closing </DT> tag is optional.

8 **22.8.8 <DD> term definition tag**

9 **Function:** This tag denotes the definition portion of an item within a definition list (<DL>).

10 **Format:** <DD> </DD> (The end tag is optional.)

11 **Attributes:** None

12 **Use:** The text is displayed indented and on a separate line from the text of the <DT> element. See Section 23. This tag shall

13 only be used within a definition list. An open <DD> tag is also closed by another <DD> tag, a <DT> or </DT> tag, or a

14 </DL> tag. The closing </DD> tag is optional.

15 **22.9 Tables**

16 The display parameters of a table, e.g. width, cell spacing, etc., are specified by the <TABLE> tag. The <CAPTION> tag

17 defines text to appear above or below the actual table. Each table row is contained within a <TR> tag. Normal table cells are

18 defined within <TD> tags, and header cells within <TH>tags. Functionally, the <TD> and <TH> tags are similar, but not

19 identical. They allow the two types of cell to be rendered in different styles.

20 Example HTML code for a table:

```
21      <TABLE>
22      <CAPTION> Caption Text</CAPTION>
23      <TR><TH> Heading for column 1<TH> Heading for column 2
24      <TR><TD> Cell 1, column 1<TD> Cell 1, column 2
25          :
26          :
27      </TABLE>
```

28 **22.9.1 <TABLE> table tag**

29 **Function:** Defines a table.

30 **Format:** <TABLE> {attribute_1} {attribute_2} ... > </TABLE>

31 **Attributes:** ALIGN, BACKGROUND, BGCOLOR, BORDER, CELLPADDING, CELLSPACING, HEIGHT,

32 TRANSPARENCY, WIDTH

33 **Use:** The following defaults shall be applied in the absence of the attribute:

34 ALIGN = LEFT

35 BORDER = 0

36 CELLPADDING = 1

37 CELLSPACING = 2

38 BGCOLOR = WHITE

1 TRANSPARENCY = 100 if no BGCOLOR defined, otherwise 0

2 In the absence of a WIDTH attribute, the width of the table shall be determined by the contents of the table. The </TABLE>

3 tag closes open <CAPTION>, <TD>, <TH> and <TR> tags.

4 **ALIGN attribute**

5 **Function:** Defines the horizontal alignment of the enclosed table relative to the left and right boundaries of the Content Area.

6 Alignment is only valid for the table in which it has been defined.

7 **Format:** ALIGN = *position*

8 *position:* LEFT : The table shall be positioned flush left, and text flows to the right around the table;

9 RIGHT : The table shall be positioned flush right, and text flows to the left around the table;

10

11 CENTER : The table shall be centred.

12 **BACKGROUND attribute**

13 **Function:** This attribute specifies the URL of a background image for the table. This image is tiled if it is smaller than the

14 table dimensions and clipped on the right and bottom side if it is larger than the table.

15 **Format:** BACKGROUND = *TeleWeb URL* (see section 5.2).

16 **BORDER attribute**

17 **Function:** Defines the width (in pixels) of an outer border to be drawn around the table.

18 **Format:** BORDER = *unsigned_integer*

19 BORDER

20 **Use:** If BORDER appears alone (without a value or equals sign), then BORDER = 1 shall be assumed. A value of 0 shall

21 indicate that the border is to be suppressed.

22 **CELLPADDING attribute**

23 **Function:** Defines the padding (in pixels) between the border around each cell and the contents of the cell.

24 **Format:** CELLPADDING = *unsigned_integer*

25 **CELLSPACING attribute**

26 **Function:** Defines the spacing (in pixels) between adjacent cells, and between the outer cells and the boundary of the table.

27 **Format:** CELLSPACING = *unsigned_integer*

28 **WIDTH attribute**

29 **Function:** Defines the width of the table, either as a number of pixels or as a percentage of the width between the current

30 margins.

31 **Formats:** WIDTH = *unsigned_integer* : number of pixels

32 WIDTH = *unsigned_integer*% : percentage of the width of the Content Area

33 **HEIGHT attribute**

34 **Function:** Defines the height of the table as a number of pixels.

- 1 **22.9.3 <TR> table row tag**
- 2 **Function:** Encloses the elements of one row of a table.
- 3 **Format:** <TR {attribute_1} {attribute_2}> </TR> (The end tag is optional)
- 4 **Attributes:** ALIGN, VALIGN, TRANSPARENCY, BGCOLOR
- 5 **Use:** This tag shall only be used within a table definition. An open <TR> tag is closed by another <TR> tag, a </TABLE> tag
- 6 or a </TR> tag. The following defaults shall be applied in the absence of an attribute:
- 7 ALIGN = CENTER
- 8 VALIGN = MIDDLE
- 9 TRANSPARENCY = 100 if no BGCOLOR defined, otherwise 0
- 10 BGCOLOR inherited from enclosing TABLE
- 11 **ALIGN attribute**
- 12 **Function:** Defines the horizontal alignment of the contents of every cell within the enclosed row relative to the left and right
- 13 boundaries of the cell. Alignment is only valid for the row in which it has been defined.
- 14 **Format:** ALIGN = *position*
- 15 ***position:*** LEFT : The content of all cells in the row shall be rendered flush left;
- 16 RIGHT : The content of all cells in the row shall be rendered flush right;
- 17 JUSTIFY : The content of all cells in the row shall be rendered flush left and flush right by adjusting the width of
- 18 the white space between the words;
- 19 CENTER : The content of all cells in the row shall be centred.
- 20 **VALIGN attribute**
- 21 **Function:** Defines the vertical alignment of the contents of every cell within the enclosed row relative to the top and bottom
- 22 boundaries of the current row. Alignment is only valid for the row in which it has been defined.
- 23 **Format:** VALIGN = *v_position*
- 24 ***v_position:*** TOP : A cell's contents shall be positioned at the top of the cell;
- 25 MIDDLE : A cell's contents shall be positioned in the middle of the cell;
- 26 BOTTOM : A cell's contents shall be positioned at the bottom of the cell.
- 27 **Use:** For robustness an attribute value of CENTER should be interpreted as MIDDLE.
- 28 **TRANSPARENCY attribute**
- 29 **Function:** Defines the transparency level for all table cells in the row unless overwritten by a TH or TD TRANSPARENCY
- 30 attribute.
- 31 **Format:** TRANSPARENCY = *unsigned_integer* : Valid range = 0 (fully opaque) to 100 (fully
- 32 transparent)
- 33 **Use:** Depending on the transparency level, the cells in the row are alpha blended with an underlying video.
- 34 **BGCOLOR attribute**
- 35 **Function:** Defines the background colour of all cells in the row, unless overridden by a TH or TD BGCOLOR attribute.
- 36 **Format:** BGCOLOR = *colour* The format of *colour* is defined in section 22.2.5.1.

1 **Use:** The stated colour shall be used to fill completely that part of cells within the current row for which no text or images are
2 defined.

3 **22.9.4 <TD> table data tag**

4 **Function:** Encloses the content for one normal data cell within the current row of the current table.

5 **Format:** <TD {attribute_1} {attribute_2} ... > </TD> (The end tag is optional.)

6 **Attributes:** ALIGN, BACKGROUND, BGCOLOR, COLSPAN, ROWSPAN, HEIGHT, NOWRAP, VALIGN, WIDTH,
7 TRANSPARENCY

8 **Use:** This tag shall only be used within a <TR> element. An open tag is also closed by another <TD> tag, a <TH>, <TR>,
9 </TR>, <CAPTION> or </TABLE> tag. The following defaults shall be applied in the absence of the attribute:

10 ALIGN = LEFT
11 BGCOLOR inherited from enclosing TR
12 COLSPAN = 1
13 ROWSPAN = 1
14 TRANSPARENCY defaults to 0 if a background colour is specified, and defaults to 100 otherwise.
15 VALIGN = MIDDLE
16

17 The ALIGN = LEFT default is overridden by an ALIGN attribute in the <TR> tag.

18 **ALIGN attribute**

19 **Function:** Defines the horizontal alignment of the contents of the cell within the boundaries of the cell.

20 **Format:** ALIGN = *h_position*

21 ***h_position:*** LEFT : The cell's contents shall be positioned flush left;
22 RIGHT : The cell's contents shall be positioned flush right;
23 JUSTIFY : The cell's contents shall be positioned flush left and flush right by adjusting the width of
24 the white space between the words;
25 CENTER : The cell's contents shall be centred.

26

27 **Use:** Overrides any ALIGN attribute within the current <TR> tag. Alignment is only valid for the cell in which it has been
28 defined.

29 **BACKGROUND attribute**

30 **Function:** This attribute specifies the URL of a background image for the table cell. This image is tiled if it is smaller than
31 the cell dimensions and clipped on the right and bottom side if it is larger than the cell.

32 **Formats:** BACKGROUND = *TeleWeb URL* (see section 5.2).

33 **BGCOLOR attribute**

34 **Function:** Defines the background colour of the cell.

35 **Format:** BGCOLOR = *colour* The format of *colour* is defined in section 22.2.5.1.

36 **Use:** The stated colour shall be used to fill completely that part of the Text/Graphics plane within the current cell for which
37 no text or images are defined. The value of the TRANSPARENCY attribute shall determine the opacity of the background
38 colour to the video signal. The effect is only valid for the cell in which it has been defined.

1 **COLSPAN attribute**

2 **Function:** Defines the number of columns spanned by the cell.

3 **Format:** COLSPAN = *unsigned_integer*

4 **Use:** This attribute is only valid for the cell in which it has been defined.

5 **ROWSPAN attribute**

6 **Function:** Defines the number of rows spanned by the cell.

7 **Format:** ROWSPAN = *unsigned_integer*

8 **Use:** This attribute is only valid for the cell in which it has been defined.

9 **HEIGHT attribute**

10 **Function:** Defines the height of the table cell in pixels.

11 **Formats:** HEIGHT = *unsigned_integer* : number of pixels

12 **NOWRAP attribute**

13 **Function:** No automatic line-breaks are inserted. Table cell is automatically extended to fit the text-content. If this attribute is

14 present, the WIDTH attribute is disregarded if it is smaller than the content width.

15 **Format:** NOWRAP

16 **VALIGN attribute**

17 **Function:** This attribute specifies the vertical position of data within a cell.

18 **Format:** VALIGN = TOP | MIDDLE | BOTTOM

19 TOP : Cell data is flush with the top of the cell.

20 MIDDLE : Cell data is centred vertically within the cell. This is the default value.

21 BOTTOM : Cell data is flush with the bottom of the cell.

22

23 **Use:** For robustness an attribute value of CENTER should be interpreted as MIDDLE.

24 **WIDTH attribute**

25 **Function:** Specifies the intended width of the table cell, either as an absolute number of pixels or as a percentage of the width

26 of the Content Area.

27 **Formats:** WIDTH = *unsigned_integer* : number of pixels.

28 WIDTH = *unsigned_integer*% : percentage of the width of the current margins.

29 **TRANSPARENCY attribute**

30 **Function:** Defines the transparency level of the cell's background colour.

31 **Format:** TRANSPARENCY = *unsigned_integer* : Valid range = 0 (fully opaque) to 100 (fully

32 transparent)

1 **Use:** Depending on the transparency level, the data cell is alpha-blended with an underlying video. If there is no video
2 available, this attribute has no effect.

3 **22.9.5 <TH> table heading tag**

4 **Function:** Enclosed the text for one header cell within the current row of the current table.

5 **Format:** <TH {attribute_1} {attribute_2} ... > </TH> (The end tag is optional.)

6 **Attributes:** ALIGN, BACKGROUND, BGCOLOR, COLSPAN, ROWSPAN, HEIGHT, NOWRAP, VALIGN, WIDTH and
7 TRANSPARENCY as defined in section 22.9.4.

8 **Use:** This tag shall only be used within a <TR> tag. An open tag is also closed by another <TH> tag, a <TD>, <CAPTION>,
9 <TR>, </TR> or </TABLE> tag. The following defaults shall be applied in the absence of the attribute:

10 ALIGN = CENTER but only if there is no ALIGN attribute specified in the <TR> tag
11 BGCOLOR inherited from enclosing TR
12 COLSPAN = 1
13 ROWSPAN = 1
14 TRANSPARENCY defaults to 0 if a background colour is specified, and defaults to 100 otherwise.
15 VALIGN = MIDDLE
16

17 **22.10 Images**

18 **22.10.1 image tag**

19 **Function:** Used to insert a graphics file.

20 **Format:**

21 **Attributes:** ALIGN, ALT, BORDER, HEIGHT, HSPACE, LOWSRC, SRC, USEMAP, VSPACE, WIDTH,
22 TRANSPARENCY

23 **Use:** Together, the HEIGHT and WIDTH attributes allow an area of the Content Area to be reserved for the image if the
24 image file is not immediately available. The following defaults shall be applied in the absence of the attribute:

25 ALIGN = BOTTOM
26 BORDER = 0
27 HEIGHT = height defined within the image file
28 WIDTH = width defined within the image file
29 HSPACE = 0
30 VSPACE = 0
31 TRANSPARENCY = 0

32 **Note:** A decoder is not required to scale an image to fit within the area defined by the HEIGHT and WIDTH attributes. If the
33 HEIGHT and WIDTH attributes do not match the actual parameters carried within the file, a decoder shall reserve an
34 area for the image of the size specified by the HEIGHT and WIDTH attributes. If the image size is not identical to the
35 specified area, the behaviour is not specified.

36 **ALIGN attribute**

37 **Function:** Specifies how the image is positioned relative to the current text line.

38 **Format:** ALIGN = *position*

1 **position:** ABSBOTTOM: Aligns the bottom of the image with the bottom of the current line.
2 ABSMIDDLE: Aligns the middle of the image with the middle of the current line
3 BASELINE: Aligns the bottom of the image with the baseline of the current line (same as BOTTOM).
4 BOTTOM: Aligns the bottom of the image with the baseline.
5 CENTER: Aligns the text baseline with the middle of the image (same as MIDDLE).
6 LEFT: Floats the image to the current left margin, temporarily changing this margin so that subsequent
7 text flows along the image's right-hand side.
8 MIDDLE: Aligns the middle of the image with the baseline for the current text line.
9 RIGHT: Floats the image to the current right margin, temporarily changing this margin, so that subsequent
10 text flows along the image's left-hand side.
11 TEXTTOP: Aligns the top of the image with the top of the text on that line.
12 TOP: Aligns the top of the image with the top of the tallest object on that line.
13
14 Note: For ALIGN = LEFT (or ALIGN = RIGHT), the rendering will depend on whether there is any left (right)
15 aligned text or images that appear earlier than the current image in the document. Such text, but not images,
16 generally forces left (right) aligned images to wrap to a new line, with the subsequent text continuing on the former
17 line.

18 **ALT attribute**

19 **Function:** Provides a text description of the image.

20 **Format:** ALT = *text_string*

21 **Use:** A decoder may choose to display this text while waiting for the image file to arrive.

22 **BORDER attribute**

23 **Function:** Defines the width (in pixels) of a border to be drawn around the image

24 **Format:** BORDER = *unsigned_integer*
25 BORDER

26 **Use:** If BORDER appears alone (without a value or equals sign), then BORDER = 1 shall be assumed. A value of 0 shall
27 indicate that the border is to be suppressed.

28 **HEIGHT attribute**

29 **Function:** Specifies the intended height of the image in pixels.

30 **Formats:** HEIGHT = *unsigned_integer* : number of pixels

31 **Use:** When the actual image height is not equal to the HEIGHT attribute, decoder behaviour is unspecified.

32 **HSPACE attribute**

33 **Function:** Specifies the width (in pixels) of space to be inserted immediately to the left and to the right of the image.

34 **Format:** HSPACE = *unsigned_integer*

35 **Use:** The background signal shall be visible in this space.

36 **LOWSRC attribute**

37 **Function:** Specifies an URL for a placeholder image.

1 **Format:** LOWSRC = *url*

2 *url*: As defined in section 5.2

3 **Use:** Use this URL when the image defined by the SRC attribute is still not available. As soon as the image defined by the
4 SRC attribute is available the image defined by this attribute should be replaced by the image defined by the SRC attribute.
5 When the image defined by the SRC attribute is already available this attribute should be ignored.

6 This attribute can be used to define a placeholder to be used until the image is available (received). This placeholder can be
7 an image of lower resolution and therefore transmitted more often. The same placeholder can be used for different images
8 (e.g. a news icon for all images used in news pages). The images defined by the LOWSRC attribute and the SRC attribute
9 must have the same size.

10 Decoder support for this attribute is optional.

11 **SRC attribute**

12 **Function:** Specifies the URL of the image.

13 **Format:** SRC = *url*

14 *url*: As defined in section 21.2.

15 **Use:** The use of this attribute is mandatory.

16 **USEMAP attribute**

17 **Function:** Identifies a <MAP> tag that defines a number of hotspots within the image.

18 **Format:** USEMAP = *url*

19 *url*: As defined in section 21.2. This *url* should only refer to a map defined in the same document.

20 **Example:**
21 <MAP NAME = "map_1"> ...

22

23 **VSPACE attribute**

24 **Function:** Specifies the height (in pixels) of space to be inserted immediately above and below the image.

25 **Format:** VSPACE = *unsigned_integer*

26 **Use:** The background signal shall be visible in this space.

27 **WIDTH attribute**

28 **Function:** Specifies the intended width of the image.

29 **Formats:** WIDTH = *unsigned_integer* : number of pixels

30 **Use:** When the actual image width is not equal to the WIDTH attribute, decoder behaviour is unspecified.

31 **TRANSPARENCY attribute**

32 **Function:** Defines the transparency level of the image.

1 **Format:** TRANSPARENCY = *unsigned_integer* : Valid range = 0 (fully opaque) to 100 (fully
2 transparent)

3 **Use:** Depending on the transparency level, the image is alpha-blended with an underlying video. If there is no video available,
4 this attribute has no effect.

5 **22.11 Ticker Text**

6 Ticker text is supported by the MARQUEE tag.

7 **22.11.1 <MARQUEE> tag**

8 **Function:** This Element defines a scrolling, sliding or bouncing text marquee.

9 **Format:** <MARQUEE {attribute_1} {attribute_2} ... > </MARQUEE>

10 **Attributes:** BEHAVIOR, BGCOLOR, DIRECTION, HEIGHT, HSPACE, LOOP, SCROLLAMOUNT, SCROLLDELAY,
11 TRANSPARENCY, VSPACE, WIDTH

12 **Use:** The following defaults shall be applied in the absence of the attribute:

13 BEHAVIOR = SCROLL
14 BGCOLOR = WHITE
15 DIRECTION = LEFT
16 HEIGHT = *content height*, if DIRECTION is LEFT or RIGHT, 200 if direction is UP or DOWN
17 HSPACE = 0
18 LOOP = INFINITE if BEHAVIOR is SCROLL or ALTERNATE, 1 if BEHAVIOR = SLIDE
19 SCROLLAMOUNT = 10
20 SCROLLDELAY = 100
21 TRANSPARENCY = 100 if no BGCOLOR specified, otherwise 0
22 VSPACE = 0
23 WIDTH = 100%

24 A decoder is only required to scroll the text contained within the tag. Scrolling of images is at the discretion of the
25 manufacturer.

26 **BEHAVIOR attribute**

27 **Function:** Defines the motion of the text.

28 **Format:** BEHAVIOR = *behaviour*

29 ***behaviour:*** ALTERNATE : The text bounces back and forth between the left and right margins of the message
30 window.
31 SCROLL : The text scrolls in the direction defined by the DIRECTION attribute and disappears
32 completely before starting again.
33 SLIDE : The text scrolls in the direction defined by the DIRECTION attribute and stops as soon as
34 the other margin is reached.

35 **BGCOLOR attribute**

36 **Function:** Defines the background colour of the message window.

37 **Format:** BGCOLOR = *colour* The format of *colour* is defined in section 22.2.5.1.

38 **Use:** The stated colour shall be used to fill completely that part of the Text/Graphics plane enclosed by the current
39 MARQUEE tag for which no content is defined.

1 **DIRECTION attribute**

2 **Function:** Defines the direction of scrolling, sliding or alternation.

3 **Format:** DIRECTION = *direction*

4 **direction:** LEFT : Starts from the right margin and moves towards the left

5 RIGHT : Starts from the left margin and moves towards the right

6 UP : Starts from the bottom margin and moves towards the top

7 DOWN : Starts from the **top** margin and moves towards the bottom

8

9 **HEIGHT attribute**

10 **Function:** Specifies the height of the message window in pixels, either as an absolute number of pixels or as a percentage of

11 the height of the Content Area.

12 default: up/down 200 pixels.

13 **Formats:** HEIGHT = *unsigned_integer* : number of pixels

14 HEIGHT = *unsigned_integer*% : percentage of height of Content Area

15 **HSPACE attribute**

16 **Function:** Specifies the width (in pixels) of background to be visible immediately left and right of the message window.

17 **Format:** HSPACE = *unsigned_integer*

18 **LOOP attribute**

19 **Function:** Defines the number of times the marquee shall be repeated.

20 **Format:** LOOP = *unsigned_integer*

21 LOOP = INFINITE or -1

22 **SCROLLAMOUNT attribute**

23 **Function:** Defines the amount in pixels between two successive displays of the scrolling text in the marquee.

24 **Format:** SCROLLAMOUNT = *unsigned_integer*

25 **SCROLLDELAY attribute**

26 **Function:** Defines the delay, in milliseconds, between two successive draws.

27 **Format:** SCROLLDELAY = *unsigned_integer*

28 **TRANSPARENCY attribute**

29 **Function:** Defines the transparency of the background.

30 **Format:** TRANSPARENCY = *unsigned_integer* : Valid range = 0 (fully opaque) to 100 (fully

31 transparent)

32 **Use:** Depending on the transparency level, the marquee is alpha-blended with an underlying video.

1 **VSPACE attribute**

2 **Function:** Specifies the height (in pixels) of background to be visible immediately above and below the message window.

3 **Format:** VSPACE = *unsigned_integer*

4 **WIDTH attribute**

5 **Function:** Specifies the width of the message window in pixels, either as an absolute number of pixels or as a percentage of
6 the width of the Content Area.

7 **Formats:** WIDTH = *unsigned_integer* : number of pixels
8 WIDTH = *unsigned_integer*% : percentage of width of Content Area

9

10 **22.12 Exceptional ignored tags**

11 All contents within FORM, APPLET, SCRIPT and STYLE elements are ignored.

12

13 **23 TeleWeb Default Style**

14 The TeleWeb default style is defined in Annex E of this document. Although the TeleWeb profile 1 decoder does not support
15 cascading style sheets, the CSS2 format, defined in [25], is used as a syntax to describe the behaviour of the browser.

16

17 **24 Image files**

18 **24.1 GIF**

19 A decoder is required to implement the full GIF specification [4], apart from the "plain text" extension. Therefore a
20 transmitted file shall not include "plain text".

21 Animation and looping of GIF images is described in [4] and [27] respectively.

22 Note: There is no minimum time specified for the display of one frame when animating or looping. This will depend upon the
23 processing power in the decoder and the complexity and size of the images. Different decoders may show different response
24 speeds.

25 A decoder is not required to reproduce animation on the Background Image plane. If multiple frames are present in the
26 designated file, the first frame should be displayed continuously.

27 If the GIF file does not define a colour palette the TeleWeb default colour palette will be used.

28 The total size of animated GIFs inside the page or content area may be restricted by a code of practice

29 **24.2 JPEG**

30 The sequential baseline JPEG image format [7][8] is supported. The progressive, hierarchical and lossless JPEG formats are
31 not supported. The overall size of the JPEG images within the content area may be restricted by a code of practice.

32 For the transmission of JPEG streams the JFIF format defined in [6] is used.

1 **25 Content labelling**

2 To enable a decoder to offer additional facilities for content selection or filtering, it is necessary to have a method of
3 assigning labels to individual files. For example, if the editorial theme of each page is indicated and the local memory in the
4 decoder is insufficient to store the complete database, the equipment could be programmed by the viewer to accept only those
5 pages that match the viewer's interests. Alternatively, it might be required to rate the content in some way so that a parent can
6 prevent a child from seeing particular content that the parent considers to be unsuitable.

7 **25.1 Predefined themes and identifier coding**

8 A common coding scheme is used for predefined themes. Their 16-bit identifier values are shown in Annex C . The 5 MSBs
9 are used to divide the table into 32 sections. In general, each section covers a certain top-level topic or subject area, with up
10 to 2048 entries. In each section, the identifier value with the 11 LSBs set to 0x001 is allocated to identify the main index page
11 for the implicit topic.

12 A theme is assigned to a file via a Theme attribute, section 28.2.2. The Identifier Value for the required theme is specified.

13 **25.2 Parental Ratings**

14 Ratings can be used to quantify content so that a parent can prevent a child from seeing particular content that the parent
15 considers to be unsuitable. The rating is defined as a recommended minimum viewing age.

16 A rating is assigned to a file via a Parental Rating attribute, section 28.2.3

18 **26 Special data**

19 **26.1 Service Identification graphic**

20 In an environment where multiple TeleWeb services are available, the decoder may wish to present the viewer with a top-
21 level menu of the different services available. The format of the menu and selection methods used shall be determined by the
22 decoder manufacturer. However, each TeleWeb service needs to provide suitable data in a standardised form to allow an
23 entry to be included in the menu. An image file can be used for this purpose. This file is optional.

24 In order to ensure that a reasonable number of choices can be placed on the screen the graphic must not extend 160 Pixels
25 horizontal x 80 Pixels vertical. The graphic should only contain the logo for the Service. The actual selection list should be
26 based on the mandatory "Information attribute" of the service. The transmission and display of the identification logo is
27 optional.

28 Note: To reduce the transmission overhead the broadcasters should try to reduce the size of the identification graphic. He
29 can do that by reducing the physical dimension of the graphic, reducing the detail level used in the graphics and by using
30 JPEG or GIF compression whichever gives the better compression.

31 Identification of the file containing the service identification graphic is by the setting of the Service Ident attribute (section
32 28.5.1). This attribute shall be set on only one file per service.

33 **26.2 Home Page**

34 Typically, the "Home Page" of a TeleWeb service will be presented to the viewer when he starts a TeleWeb viewing session.
35 The service provider may choose to design this page as an entry point into the service.

36 It is mandatory to have a "Home Page" in every TeleWeb service.

1 There shall be only one instance of a "Home Page" within a TeleWeb service. It is identified through the setting of the "Home
2 Page" attribute for the appropriate file (section 28.5.2). The file shall be an HTML file only. If the "Home Page" references a
3 number of image files, the Home Page attribute shall not be set for the image files.

4 **26.3 Default Page**

5 The decoder will display a default page whenever an unresolved link is selected. The mandatory filename for this default
6 page is "default.htm" or "default.html". If the page is not provided by the broadcaster or not received yet, the decoder must
7 generate the page. In the latter case the decoder manufacturer is in full control of the content of the page, however the content
8 should indicate to the user that the requested page is for the moment not available.

9 With the special function URL function:back (see section 21.5) a link should be defined in the page to enable the user to go
10 back to the previously displayed page.

11 **26.4 Profile Upgrade Page**

12 A Profile 1 decoder will display a default page whenever an unresolved higher profile link is selected. The mandatory
13 filename for this default page is "upgrade.htm" or "upgrade.html". If the page is not provided by the broadcaster or not
14 received yet, the decoder must generate the page. In the latter case the decoder manufacturer is in full control of the content
15 of the page, however the content should indicate to the user that his decoder is not capable of processing the page referenced
16 by this link.

17 With the special function URL function:back (see section 21.5) a link should be defined in the page to enable the user to go
18 back to the previously displayed page.

19

20 **1.1 ZLIB dictionary files**

21 For better compression of files the zlib decompression supports the use of dictionaries. ZLIB dictionaries are transmitted as
22 files within the TeleWeb service. The name of a file containing a dictionary shall be zlibHHHHHHHH.dic where
23 HHHHHHHH is the hexadecimal representation of the Adler32 checksum of the dictionary (see section 11.1.6).
24 Dictionary files must be transmitted uncompressed.

25 **27 Service-related attributes**

26 The service attributes define attributes for the whole service. To assure a fast detection and identification of a TeleWeb
27 service these attributes should be transmitted at least every minute.

28 The service-related attributes are summarised in the table below. The "Mandatory" column indicates which attributes are
29 mandatory, and which attributes are optional.

30 **Table 8: List of service-related attributes**

Attribute	Function	Mandatory
Name	The name of the service. Used in absolute URLs.	yes
Information	A textual description of the service. Used for the service selection.	yes
Language	The principle language used for this service	yes
Transmission Schedule	List of maximum cycle times	no

31 Note: A decoder will process service-related attributes defined within this section. A decoder shall ignore service-related
32 attributes not listed in the table above.

1 **27.1 Name**

2 The name of a TeleWeb service is transmitted via the Name attribute. The name is used as identification of the service in the
3 TeleWeb URL (see section 5.2.1). The Service name is not intended to be presented to the user. For a service selection menu
4 the data in the Information attribute should be used (see section 11.1).

5 **27.2 Information**

6 The Service information attribute contains the textual identification for the Service. This information should be used when a
7 service selection menu is presented to the user. Such a service selection list can also show the (optional) service identification
8 graphics. The maximum length for the information text is restricted to 32 characters.

9 **27.3 Language**

10 The Language attribute defines the default language for the TeleWeb service. The definition of this service attribute is
11 mandatory.

12 **27.4 Transmission Schedule**

13 With this optional service attribute the cycle times of the whole service can be defined. The cycle times may vary during the
14 day. For example at night there may be a higher transmission rate of the data. The following data is transmitted:

15	Time 1	Cycle	Time	1
16	Time 2	Cycle	Time	2
17	:			
18	Time n	Cycle Time 2		

19 The following list gives an example for the Transmission Schedule attribute:

20	00:00	20min
21	03:00	10min
22	06:00	20min
23	12:00	30min
24	20:00	40min

25 With this data a TeleWeb decoder can choose the time with the highest transmission rate to receive the data. If only one entry
26 is given the maximum transmission rate should be constant for the whole day.

28 File-related attributes

Sections 22 and 24 define the contents of the text and graphics files forming the TeleWeb service. Attributes for each file such as name, file type, creation date, etc., also need to be provided to enable a decoder to process the files correctly and manage the local database.

The attributes are assumed to exist outside of the files to which they apply. The method of linking attributes to files is a function of the transport protocol. Under some circumstances it may be possible to apply an attribute to multiple files in the interests of transmission efficiency.

The TeleWeb specific file-related attributes applicable to data and graphics files are summarised in Table 9. The "No. Per File" column indicates which attributes are mandatory, and which attributes can be included more than once per file.

Table 9: List of file-related attributes

Attribute	Function	No. Per File
Type	The type of data in the file	0 or 1
Name	The name of the file	1
CRC	Cyclic Redundancy Code (CRC) checksum	0 or 1
Copyright	Indicates that the contents are copyright protected and must not be used outside of their host TeleWeb service	0 or 1 Note 1
Encryption/ Conditional Access	Indicates that the contents of the file have been encrypted	0 or 1 Note 1
Compression	Indicates that the contents of the file have been compressed	0 or 1 Note 1
Information	A textual description of the file's contents	0 or 1
Parental Rating	Age rating of the content according to [36]	0 or 1
Theme	An indication of the contents of the file in a "computer-readable" format to allow intelligent and personalised search engines to be implemented	Several (including 0)
Language	The principle language used to author the text	0 or 1
Character Set (Encoding)	The character encoding used to author the content	0 or 1
Suppress User Interface	Indicates that the author intends the page is displayed with video in the background and without the display components of the user interface	0 or 1 Note 1
Creation Time	The authoring time and date of the file	0 or 1
Start Validity	The time from which the file may be used (displayed) by the decoder	0 or 1
Expire Time	The time after which the file is no longer valid and can be deleted by the decoder	0 or 1
Repetition Distance	The maximum time between transmissions of the file	0 or 1
Priority	Indicates the files considered to be more important to help the decoder with its "housekeeping" functions.	1
Service Indent	Indicates if the file contains data relating to the identity of the TeleWeb service	0 or 1 Note 1
Home Page	Indicates if the file contains data relating to the mandatory "Home Page" of the TeleWeb service	0 or 1 Note 1
User Group ID	Allows files to be distributed to particular decoders in order to support closed user group or conditional access services	0 or 1
Profile	Defines the TeleWeb profiles the page is intended for	0 or 1
Version	Defines the version of the file	1

Note 1: These attributes are mandatory for the files supporting or requiring these particular functions.

Note 2: A decoder will process file-related attributes defined within this section. A decoder shall ignore file-related attributes not listed in the table above.

28.1 General file attributes

28.1.1 Type

The Type attribute indicates the nature of the data in the file, e.g. HTML, GIF, etc. This attribute should conform to the file extension. The type is specified using the Media Type descriptions [41] and [42].

The inclusion of this attribute is optional. If it is omitted or missing, a decoder should attempt to determine the file type by inspecting the Name attribute. In the event of a conflict, the Type attribute shall have priority.

- 1 **28.1.2 Name**
- 2 The Name attribute defines the "name" of the file as a text string. This name is used as the reference (URL) to the file. For
3 text and image files it is recommended to included a file type extension, e.g. "file1.htm".
- 4 The string shall be coded using the ISO Latin-1 character set [21]. A maximum length limit is not placed on the string but it
5 may be limited in practice by the transport protocol. A decoder is not expected to display this string on the screen.
- 6 The inclusion of this attribute is mandatory for every file.
- 7 It is not intention to display the name of a file to the user
- 8 **28.1.3 CRC**
- 9 The CRC attribute carries a 32-bit checksum for the file. This is calculated according to annex B of [36].
- 10 The inclusion of this attribute is optional.
- 11 **28.1.4 Copyright**
- 12 The Copyright attribute is a single-bit Boolean entity indicating if the file is copyright protected. If set the entire contents of
13 the file are subject to copyright and no part may be used in combination with any other application, nor by any other
14 application, nor shall it be copied to a composite database. It shall be used exclusively within the TeleWeb service to which it
15 belongs.
- 16 The presence of this attribute is mandatory if the data is to be protected in this way. In the absence of this attribute a decoder
17 may assume that the contents of the file are not copyright protected.
- 18 **28.1.5 Encryption/Conditional Access**
- 19 The presence of an Encryption attribute informs the decoder that the contents of the file have been encrypted and an
20 appropriate key will be needed to unscramble the data prior to use. Profile 1 decoders will not support encryption. Thus files
21 containing an encryption attribute shall be ignored by a Profile 1 decoder.
- 22 **28.1.6 Version**
- 23 A version number is transmitted with each file. Whenever the content of a file is changed this version number is changed also.
- 24 When a new version of a TeleWeb HTML file is received and this file is currently displayed then the display should be
25 automatically updated with the new content. When a new version of an image that is displayed is received and this image is
26 referenced by an tag then the new version of the image should replace the old one on the screen.
- 27 **28.1.7 Compression**
- 28 The Compression attribute identifies the compression method in use, and, if relevant, the original size of the uncompressed
29 file in bytes. For systems designed to this edition, the compression algorithm shall be "zlib" [15].
- 30 The zlib decompression shall support dictionaries as defined in [15]. The dictionary to be used to decompress a file shall be
31 transmitted as a file within the TeleWeb service. A zlib compressed file that needs a dictionary to be decompressed defines
32 the Adler32 checksum of that dictionary. Let HHHHHHHH be the hexadecimal representation of that Adler32 checksum,
33 then the content of the file zlibHHHHHHHH.dic shall be used as the dictionary. The dictionary file must be transmitted
34 uncompressed and the size must not exceed 32Kbytes. For different zlib compressed files different dictionaries may be used
35 as long as their Adler32 checksums are different.
- 36 Note: When a file is to be subject to both compression and encryption, the encoding process should perform compression first
37 and then encryption.

1 **28.2 File content attributes**

2 **28.2.1 Information**

3 The Information attribute contains a text string describing the contents of the file. Text information is coded using the ISO
4 Latin-1 character set [21]. A maximum length for the string is not specified but may be limited by the transmission method.

5 The inclusion of this attribute is optional.

6 **28.2.2 Theme**

7 The Theme attribute is describing the contents of the file. A theme is selected by a unique 16-bit value which is defined in a
8 list of predefined themes, see Annex C .

9 The inclusion of this attribute is optional. A maximum of 8 themes shall be applied to one file.

10 NOTE: Themes might be used within the decoder to implement customisable search engines and to filter the incoming data to
11 restrict the local database to information that is of particular interest to the viewer, especially when the local storage capacity
12 is insufficient to hold the entire database.

13 **28.2.3 Parental Rating**

14 This 8-bit field is coded according to Table 10, giving the recommended minimum age in years of the end user.

15

16

Table 10: Rating scheme

Rating	Description
0x00	Undefined
0x01 – 0x0F	minimum age = rating + 3 years
0x10 – 0xFF	Reserved

17

18 Example: 0x04 implies that end users should be at least 7 years old.

19 **28.2.4 Character Set (Encoding)**

20 This attribute specifies the character set used to author the content of the file. It is defined by a string of 7-bit ASCII coded
21 characters. For profile 1 decoder only ISO-8859-1 (Latin-1) [21] is supported which is also the default character set when this
22 attribute is not defined.

23 **28.2.5 Language**

24 The Language attribute defines the principle language used to author the text in HTML files and image files.

25 The language is identified by a three-byte code as specified in [38].

26 The inclusion of this attribute is optional. In its absence, the language specified for the service shall be adopted as the default
27 (see section 27.3).

1 **28.2.6 Suppress User Interface**

2 The Suppress User Interface attribute is a Boolean entity indicating that screen areas under control of the decoder
3 manufacturer should be set to display video.

4 **28.2.7 Profile**

5 The profile attribute indicates the profiles of the decoder the page is intended for. For each profile an information flag is
6 reserved. If this attribute is available the decoder must ignore pages which do not contain a flag for its profile. Pages without
7 this attribute are applicable for all profiles.

8 **28.3 Time and date related attributes**

9 Unless otherwise stated, all times are absolute and are specified using Universal Co-ordinated Time (UTC) to a resolution of
10 one second. Individual components shall be used to represent hours, minutes and seconds, and shall each be coded as 8-bit
11 values. Dates are also absolute and are defined using the Modified Julian Date (MJD) form as a 17-bit value. (The latest date
12 that can be specified is 17 November 2172). The MJD value increments daily at 00:00 UTC.

13 **28.3.1 Creation Time**

14 The Creation Time attribute defines the authoring time and date of the file, or the time/date at which it was last updated. The
15 time and date shall be absolute.

16 The inclusion of this attribute is optional.

17 **28.3.2 Start Validity**

18 The Start Validity attribute indicates the earliest time at which the file may be used or displayed. The time shall be specified
19 as an absolute time and date.

20 The inclusion of this attribute is optional. In its absence, a decoder may use the file as soon as it is acquired.

21 **28.3.3 Expire Time**

22 The Expire Time attribute indicates the latest time at which the file may be used or displayed. Once the Expire Time has been
23 exceeded, a decoder may delete the file from its memory. The time shall be specified as an absolute time and date.

24 If this optional attribute is omitted, the file is valid for an undefined period of time at the discretion of the decoder.

25 **28.3.4 Repetition Distance**

26 To support advanced caching of files in the decoder, the Repetition Distance attribute indicates a guaranteed maximum time
27 until the next transmission of the file. The decoder may use this information to indicate the worst case waiting time if the
28 requested data has not been pre-captured.

29 The time is specified as a 16-bit value in units of one second. A value of 0xFFFF shall indicate that the time is greater than 18
30 hours, 12 minutes and 14 seconds.

31 The inclusion of this attribute is optional.

1 **28.4 Storage related attributes**

2 **28.4.1 Priority**

3 The Priority attribute indicates the relevance of the contents of the associated file with regard to the overall service. It
4 recommends a storage priority, i.e. in case of a "memory full" state only files having a high priority should be stored.

5 Priorities are rated on a scale of 0 to 255, where 0 implies highest priority and 255 lowest priority.

6 The inclusion of this attribute is mandatory.

7 **28.5 Special function attributes**

8 **28.5.1 Service Ident**

9 The Service Ident attribute is a single-bit Boolean entity that enables the file containing data relating to the identity of the
10 TeleWeb service to be detected. This allows a decoder to construct a top level menu of TeleWeb services that are available
11 from different service providers.

12 The inclusion of this attribute is mandatory for the file that contains this information. There shall be only one instance of this
13 attribute per service.

14 **28.5.2 Home Page**

15 The Home Page attribute is a single-bit Boolean entity that enables the file containing data relating to the mandatory "Home
16 Page" of the TeleWeb service. This allows the decoder to display the service provider's introductory page when the user
17 selects this TeleWeb service.

18 The inclusion of this attribute is mandatory for the file containing the "Home Page" data. There shall be only one file with
19 this attribute per TeleWeb service.

20 **28.5.3 User Group ID**

21 The User Group ID attribute allows files to be distributed to particular decoders in order to support closed user group or
22 conditional access services. Each ID comprises a string of 7-bit ASCII coded characters with a maximum length of 32
23 characters.

24 The inclusion of this attribute is optional.

25 **29 Short and Full TeleWeb Service**

26 There are two types of TeleWeb services defined by this specification, a Short TeleWeb service and a Full TeleWeb service.

27 A Short service is a small TeleWeb service. The broadcast size for a Short service shall not exceed 100 Kbytes and all data in
28 the Short service shall normally be transmitted within 1 minute. The decoder may clear the stored database for the Short
29 service on each channel change releasing resources to acquire the Short service from the new channel. The Short service
30 contains programme related information that is often updated. A trigger application, defined in [35], shall only reference
31 pages from the Short service.

32 The second type of service is a Full TeleWeb service. The broadcast size for a Full service shall not exceed 4.9 Mbytes. The
33 cycle time for a Full service may be much larger than the cycle time of a Short service. Therefore, a 'Preferred' Full service
34 may be selected by the user through an installation menu. In this case the data for the Full service will be available
35 independent of the selected TV channel. Future TeleWeb decoder may, if storage becomes cheaper, store the full service of
36 all service providers.

1 The Short and Full TeleWeb service are completely separated. Therefore, it is possible to receive a Short and a Full service
2 from different service providers. It is however possible to define cross-links from one service to the other by using absolute
3 TeleWeb URL's. This is not recommended as it will only work if the stored Full and current Short TeleWeb service are from
4 the same service provider. Due to the minimum requirements of Profile1 a low-end decoder will not support cross-links.

5 A TeleWeb profile 1 decoder must support at least one Short TeleWeb service and one Full TeleWeb service providing
6 storage for the specified broadcast sizes (4.9 + 0.1 Mbytes). The Short and Full TeleWeb service do define their own entry
7 points via their mandatory home pages. The user must be able to select the service he or she likes to access. It is the
8 responsibility of the decoder manufacturer to provide a proper user interface.

9 **30 Individual Addressing - Group Addressing**

10 Individual addressing and group addressing is realised by means of the user group id. This is a simple string which is assigned
11 to files as an file attribute. It can be the name of a group or simply an e-mail address. However there is no security provided
12 with this mechanism. The user group id is not decoder specific and can be changed by the user.

13 If a file carries a user group id and the decoder does not keep the same id, it shall ignore the data.

14

Annex A HTML compatibility (informative)

In general, the HTML tags presented in section 22 have been chosen for TeleWeb use as they are widely supported by existing browsers and they are likely to be present in future versions of HTML. This appendix lists a) the tags present in HTML version 3.2 [23] that are not supported by the TeleWeb application, b) attributes that are not supported by this specification but are part of version 3.2 [23], c) browser specific tags that are supported by TeleWeb, d) extensions to HTML 3.2 attributes and attribute values.

A.1 HTML V3.2 tags not supported by TeleWeb

Table 11 lists the HTML V3.2 tags that are not supported by TeleWeb.

Table 11: HTML V3.2 tag not supported

Tag	Function
<APPLET>	Java applet
<FORM>	Fill-in form
<INPUT>	Input text field, radio buttons, etc. in <FORM> elements
<ISINDEX>	Request for a single line text input field
<OPTION>	Defines menu item within a <SELECT> element
<PARAM>	Used within <APPLET> elements
<SCRIPT>	Reserved for further use with scripting languages
<SELECT>	Menu within <FORM> elements
<STYLE>	Reserved for future use with style sheets
<TEXTAREA> >	Defines multi-line text fields in <FORM> elements

A.2 HTML V3.2 attributes not supported by TeleWeb

Table 12 lists the attributes that are not supported by TeleWeb although the HTML V3.2 tag itself, and possibly other attributes, are supported. The unsupported tags may be in use on the Internet or supported by existing browsers.

Table 12: HTML V3.2 attributes not supported

Tag	Tag Name	Attributes not supported
<DL>	Definition list	COMPACT
	Ordered list	COMPACT
	Unordered list	COMPACT
<MENU> >	Menu list	COMPACT
<DIR>	Directory list	COMPACT
<PRE>	Preformatted Text	WIDTH

1 A.3 Browser specific tags supported by TeleWeb

2 Table 13 lists the browser specific tags supported by TeleWeb.

3 **Table 13: Browser specific tags supported**

Tag	Function
<BLINK>	Enables flashing text
<MARQUEE>	Scrolling text

4 A.4 Extensions to HTML 3.2 attributes and attribute values

5 Table 14 lists the attributes and attribute values supported by the TeleWeb Browser that are not defined in HTML 3.2. Most
6 of the extensions are defined in HTML 4.0.

7 **Table 14: TeleWeb specific tags supported**

Tag	New attribute	Function
<A> anchor	ACCESSKEY	Maps hyperlink to a particular button or key on the user's control device
<AREA>	ACCESSKEY	Maps hyperlink to a particular button or key on the user's control device
<BASEFONT>	COLOR	Text colour of the base font
>		
<BODY>	TRANSPARENCY	Alpha component (opacity) of background colour
	BOTTOMMARGIN	Bottom margin of the page in pixels
	LEFTMARGIN	Left margin of the page in pixels
	RIGHTMARGIN	Right margin of the page in pixels
	TOPMARGIN	Top margin of the page in pixels
<CAPTION>	ALIGN = LEFT, RIGHT, CENTER	vertical or horizontal position of the caption
	VALIGN	vertical position of the caption
<HR>	COLOR	Specifies rendering colour
<P>	ALIGN=JUSTIFY	Layout option
<TABLE>	BACKGROUND	URL of the table background image
<TD>, <TH>	ALIGN=JUSTIFY	Horizontal layout information
	BGCOLOR	Specifies background colour
	TRANSPARENCY	Alpha component (opacity) of background to video
	BACKGROUND	URL of the table cell background image
<TR>	BGCOLOR	Specifies background colour
	TRANSPARENCY	Alpha component (opacity) of background to video
	ALIGN=JUSTIFY	Layout option
	ALIGN= ABSBOTTOM ABSMIDDLE BASELINE CENTER TEXTTOP	Image alignment option
	LOWSRC	Definition of alternative image

Annex B Default colour palette specification

Table 15 Default colour palette specification

Transparency Level	Default colour palette entry definition		
	Format : $clut[n] = [R, G, B]$, where n denotes the clut entry, and R, G and B denote the values of the Red, Green and Blue components associated to clut entry		
0 % (fully opaque)	$clut[0] = [0, 0, 0]$ $clut[1] = [0, 0, 127]$ $clut[2] = [0, 0, 255]$ $clut[3] = [0, 31, 0]$ $clut[4] = [0, 31, 127]$ $clut[5] = [0, 31, 255]$ $clut[6] = [0, 63, 0]$ $clut[7] = [0, 63, 127]$ $clut[8] = [0, 63, 255]$ $clut[9] = [0, 95, 0]$ $clut[10] = [0, 95, 127]$ $clut[11] = [0, 95, 255]$ $clut[12] = [0, 127, 0]$ $clut[13] = [0, 127, 127]$ $clut[14] = [0, 127, 255]$ $clut[15] = [0, 159, 0]$ $clut[16] = [0, 159, 127]$ $clut[17] = [0, 159, 255]$ $clut[18] = [0, 191, 0]$ $clut[19] = [0, 191, 127]$ $clut[20] = [0, 191, 255]$ $clut[21] = [0, 223, 0]$ $clut[22] = [0, 223, 127]$ $clut[23] = [0, 223, 255]$ $clut[24] = [0, 255, 0]$ $clut[25] = [0, 255, 127]$ $clut[26] = [0, 255, 255]$ $clut[27] = [63, 0, 0]$ $clut[28] = [63, 0, 127]$ $clut[29] = [63, 0, 255]$ $clut[30] = [63, 31, 0]$ $clut[31] = [63, 31, 127]$ $clut[32] = [63, 31, 255]$ $clut[33] = [63, 63, 0]$ $clut[34] = [63, 63, 127]$ $clut[35] = [63, 63, 255]$ $clut[36] = [63, 95, 0]$ $clut[37] = [63, 95, 127]$ $clut[38] = [63, 95, 255]$ $clut[39] = [63, 127, 0]$ $clut[40] = [63, 127, 127]$ $clut[41] = [63, 127, 255]$ $clut[42] = [63, 159, 0]$ $clut[43] = [63, 159, 127]$ $clut[44] = [63, 159, 255]$	$clut[45] = [63, 191, 0]$ $clut[46] = [63, 191, 127]$ $clut[47] = [63, 191, 255]$ $clut[48] = [63, 223, 0]$ $clut[49] = [63, 223, 127]$ $clut[50] = [63, 223, 255]$ $clut[51] = [63, 255, 0]$ $clut[52] = [63, 255, 127]$ $clut[53] = [63, 255, 255]$ $clut[54] = [127, 0, 0]$ $clut[55] = [127, 0, 127]$ $clut[56] = [127, 0, 255]$ $clut[57] = [127, 31, 0]$ $clut[58] = [127, 31, 127]$ $clut[59] = [127, 31, 255]$ $clut[60] = [127, 63, 0]$ $clut[61] = [127, 63, 127]$ $clut[62] = [127, 63, 255]$ $clut[63] = [127, 95, 0]$ $clut[64] = [127, 95, 127]$ $clut[65] = [127, 95, 255]$ $clut[66] = [127, 127, 0]$ $clut[67] = [127, 127, 127]$ $clut[68] = [127, 127, 255]$ $clut[69] = [127, 159, 0]$ $clut[70] = [127, 159, 127]$ $clut[71] = [127, 159, 255]$ $clut[72] = [127, 191, 0]$ $clut[73] = [127, 191, 127]$ $clut[74] = [127, 191, 255]$ $clut[75] = [127, 223, 0]$ $clut[76] = [127, 223, 127]$ $clut[77] = [127, 223, 255]$ $clut[78] = [127, 255, 0]$ $clut[79] = [127, 255, 127]$ $clut[80] = [127, 255, 255]$ $clut[81] = [191, 0, 0]$ $clut[82] = [191, 0, 127]$ $clut[83] = [191, 0, 255]$ $clut[84] = [191, 31, 0]$ $clut[85] = [191, 31, 127]$ $clut[86] = [191, 31, 255]$ $clut[87] = [191, 63, 0]$ $clut[88] = [191, 63, 127]$ $clut[89] = [191, 63, 255]$	$clut[90] = [191, 95, 0]$ $clut[91] = [191, 95, 127]$ $clut[92] = [191, 95, 255]$ $clut[93] = [191, 127, 0]$ $clut[94] = [191, 127, 127]$ $clut[95] = [191, 127, 255]$ $clut[96] = [191, 159, 0]$ $clut[97] = [191, 159, 127]$ $clut[98] = [191, 159, 255]$ $clut[99] = [191, 191, 0]$ $clut[100] = [191, 191, 127]$ $clut[101] = [191, 191, 255]$ $clut[102] = [191, 223, 0]$ $clut[103] = [191, 223, 127]$ $clut[104] = [191, 223, 255]$ $clut[105] = [191, 255, 0]$ $clut[106] = [191, 255, 127]$ $clut[107] = [191, 255, 255]$ $clut[108] = [255, 0, 0]$ $clut[109] = [255, 0, 127]$ $clut[110] = [255, 0, 255]$ $clut[111] = [255, 31, 0]$ $clut[112] = [255, 31, 127]$ $clut[113] = [255, 31, 255]$ $clut[114] = [255, 63, 0]$ $clut[115] = [255, 63, 127]$ $clut[116] = [255, 63, 255]$ $clut[117] = [255, 95, 0]$ $clut[118] = [255, 95, 127]$ $clut[119] = [255, 95, 255]$ $clut[120] = [255, 127, 0]$ $clut[121] = [255, 127, 127]$ $clut[122] = [255, 127, 255]$ $clut[123] = [255, 159, 0]$ $clut[124] = [255, 159, 127]$ $clut[125] = [255, 159, 255]$ $clut[126] = [255, 191, 0]$ $clut[127] = [255, 191, 127]$ $clut[128] = [255, 191, 255]$ $clut[129] = [255, 223, 0]$ $clut[130] = [255, 223, 127]$ $clut[131] = [255, 223, 255]$ $clut[132] = [255, 255, 0]$ $clut[133] = [255, 255, 127]$ $clut[134] = [255, 255, 255]$
0 % (fully opaque)	$clut[135] = [42, 42, 42]$ $clut[136] = [85, 85, 85]$	$clut[137] = [170, 170, 170]$	$clut[138] = [212, 212, 212]$

<i>Transparency Level</i>	<i>Default colour palette entry definition</i> <i>Format : clut[n] = [R, G, B], where</i> <i>n denotes the clut entry, and</i> <i>R, G and B denote the values of the Red, Green and Blue components associated to clut entry</i>		
30 %	clut[139] = [0, 0, 0] clut[140] = [0, 0,255] clut[141] = [0, 51, 0] clut[142] = [0, 51,255] clut[143] = [0,102, 0] clut[144] = [0,102,255] clut[145] = [0,153, 0] clut[146] = [0,153,255] clut[147] = [0,204, 0] clut[148] = [0,204,255] clut[149] = [0,255, 0] clut[150] = [0,255,255] clut[151] = [85, 0, 0] clut[152] = [85, 0,255] clut[153] = [85, 51, 0] clut[154] = [85, 51,255]	clut[155] = [85,102, 0] clut[156] = [85,102,255] clut[157] = [85,153, 0] clut[158] = [85,153,255] clut[159] = [85,204, 0] clut[160] = [85,204,255] clut[161] = [85,255, 0] clut[162] = [85,255,255] clut[163] = [170, 0, 0] clut[164] = [170, 0,255] clut[165] = [170, 51, 0] clut[166] = [170, 51,255] clut[167] = [170,102, 0] clut[168] = [170,102,255] clut[169] = [170,153, 0] clut[170] = [170,153,255]	clut[171] = [170,204, 0] clut[172] = [170,204,255] clut[173] = [170,255, 0] clut[174] = [170,255,255] clut[175] = [255, 0, 0] clut[176] = [255, 0,255] clut[177] = [255, 51, 0] clut[178] = [255, 51,255] clut[179] = [255,102, 0] clut[180] = [255,102,255] clut[181] = [255,153, 0] clut[182] = [255,153,255] clut[183] = [255,204, 0] clut[184] = [255,204,255] clut[185] = [255,255, 0] clut[186] = [255,255,255]
100 % (fully transparent)	clut[187] = [x, x, x], where x indicates "don't care".		
Privately definable	clut[188] : reserved for private use : clut[255] : reserved for private use		

Annex C Table of Predefined Themes

Implicit Subject Area	Theme	Identifier Value
None	Reserved	0x0000
	Index	0x0001
(5 MSBs = 0x00)		
	Reserved for future use	0x0002 - 0x7FFF
News	News (top level)	0x0800
	Index	0x0801
(5 MSBs = 0x01)	General	0x0802
	National	0x0803
	International	0x0804
	Short stories	0x0805
	Public affairs	0x0806
	Domestic affairs	0x0807
	Legal and social affairs	0x0808
	Cultural affairs	0x0809
	Educational affairs	0x080A
	Communication affairs	0x080B
	International relations	0x080C
	Defence	0x080D
	Housing	0x080E
	Environment	0x080F
	Health	0x0810
	Science	0x0811
	Technology	0x0812
	Reserved for future use	0x0813 - 0x0FFF
Politics	Politics (top level)	0x1000
	Index	0x1001
(5 MSBs = 0x02)	News general	0x1002
	Special	0x1003
	National	0x1004
	International	0x1005
	Short stories	0x1006
	Reserved for future use	0x1007 - 0x17FF
Finance	Finance (top level)	0x1800
	Index	0x1801
(5 MSBs = 0x03)	News general	0x1802
	Special	0x1803
	National	0x1804
	International	0x1805
	Short stories	0x1806
	Stocks and shares	0x1807
	Exchange rates	0x1808
	Reserved for future use	0x1809 - 0x1FFF
Weather	Weather (top level)	0x2000
	Index	0x2001
(5 MSBs = 0x04)	Conditions	0x2002

Implicit Subject Area	Theme	Identifier Value
	Forecast	0x2003
	Warnings	0x2004
	Reserved for future use	0x2005 - 0x27FF

1

Implicit Subject Area	Theme	Identifier Value
Sport	Sport (top level)	0x2800
	Index	0x2801
	(5 MSBs = 0x05) News general	0x2802
	Special	0x2803
	National	0x2804
	International	0x2805
	Short stories	0x2806
	Results	0x2807
	Tables	0x2808
	Football	0x2809
	Handball	0x280A
	Basketball	0x280B
	Hockey	0x280C
	Ice hockey	0x280D
	Golf	0x280E
	Horse racing	0x280F
	Volleyball	0x2810
	Tennis	0x2811
	NBA	0x2812
	NFL	0x2813
	NHL	0x2814
	Formula 1	0x2815
	Motor sports	0x2816
	Winter sports	0x2817
	Water sports	0x2818
	Cycling	0x2819
	Boxing	0x281A
	Athletics	0x281B
	Equestrian	0x281C
	Martial arts	0x281D
	Local	0x281E
	Reserved for future use	0x281F - 0x2FFF
TV and Radio	TV and Radio (top level)	0x3000
	Index	0x3001
(5 MSBs = 0x06)	Current programme	0x3002
	"Now and Next"	0x3003
	TV listings	0x3006
	Radio listings	0x3007

Implicit Subject Area	Theme	Identifier Value
TV and Radio / Movie	Movie (general)	0x3010
	Detective/thriller	0x3011
	Adventure/western/war	0x3012
	Science fiction/fantasy/horror	0x3013
	Comedy	0x3014
	Soap/melodrama/folklore	0x3015
	Romance	0x3016
	Serious/classical/religious/historical drama	0x3017
	Adult movie	0x3018
	Reserved for future use	0x3019- 0x301F
TV and Radio / News	News/current affairs (general)	0x3020
	News/weather report	0x3021
	News magazine	0x3022
	Documentary	0x3023
	Discussion/interview/debate	0x3024
	Social/political issues/economics (general)	0x3025
	Magazines/reports/documentary	0x3026
	Economics/social advisory	0x3027
	Remarkable people	0x3028
	Reserved for future use	0x3029- 0x302F
TV and Radio / Show	Show/game show (general)	0x3030
	Game show/quiz/contest	0x3031
	Variety show	0x3032
	Talk show	0x3033
	Leisure hobbies (general)	0x3034
	Tourism/travel	0x3035
	Handicraft	0x3036
	Motoring	0x3037
	Fitness & health	0x3038
	Cooking	0x3039
	Advertisement/shopping	0x303A
	Reserved for future use	0x303B- 0x303E
	Alarm/emergency identification	0x303F
TV and Radio / Sports	Sports (general)	0x3040
	Special events (Olympic Games, World Cup etc.)	0x3041
	Sports magazines	0x3042
	Football/soccer	0x3043
	Tennis/squash	0x3044
	Team sports (excluding football)	0x3045
	Athletics	0x3046
	Motor sport	0x3047
	Water sport	0x3048
	Winter sports	0x3049
	Equestrian	0x304A
	Martial sports	0x304B
	Local sports	0x304C
	Reserved for future use	0x304D- 0x304F
TV and Radio / Children	Children's/youth programmes (general)	0x3050
	Pre-school children's programmes	0x3051
	Entertainment programmes for 6 to 14	0x3052
	Entertainment programmes for 10 to 16	0x3053
	Informational/educational/school programmes	0x3054

Implicit Subject Area	Theme	Identifier Value
	Cartoons/puppets	0x3055
	Education/science/factual topics (general)	0x3056
	Nature/animals/environment	0x3057
	Technology/natural sciences	0x3058
	Medicine/physiology/psychology	0x3059
	Foreign countries/expeditions	0x305A
	Social/spiritual sciences	0x305B
	Further education	0x305C
	Languages	0x305D
	Reserved for future use	0x305E- 0x305F
TV and Radio / Music	Music/ballet/dance (general)	0x3060
	rock/pop	0x3061
	Serious music/classical music	0x3062
	folk/traditional music	0x3063
	Jazz	0x3064
	Musical/opera	0x3065
	Ballet	0x3066
	Reserved for future use	0x3067- 0x306F
TV and Radio / Arts	arts/culture (without music, general)	0x3070
	Performing arts	0x3071
	fine arts	0x3072
	Religion	0x3073
	Popular culture/traditional arts	0x3074
	Literature	0x3075
	film/cinema	0x3076
	Experimental film/video	0x3077
	Broadcasting/press	0x3078
	new media	0x3079
	arts/culture magazines	0x307A
	Fashion	0x307B
	Reserved for future use	0x307C - 0x37FF

Implicit Subject Area	Theme	Identifier Value
Lifestyle	Lifestyle (top level)	0x3800
	Index	0x3801
(5 MSBs = 0x07)	Tips / trends	0x3802
	Finance / law	0x3803
	Computer – software	0x3804
	Computer – hardware	0x3805
	Computer - PC Games	0x3806
	Pets	0x3807
	Cars	0x3808
	Children	0x3809
	Health	0x380A
	Recipes	0x380B
	Communications	0x380C
	Satellite	0x380D
	Multimedia	0x380E
	Internet	0x380F
	Games	0x3810
	Dating	0x3811
	Horoscope	0x3812
	Betting	0x3813
	Contacts	0x3814
	Lottery	0x3815
	Reserved for future use	0x3816 - 0x3FFF

Implicit Subject Area	Theme	Identifier Value
Entertainment	Entertainment (top level)	0x4000
	Index	0x4001
(5 MSBs = 0x08)	General	0x4002
	Music - news general	0x4003
	Music – charts	0x4004
	Music – tips	0x4005
	Music - tours	0x4006
	Music - latest release	0x4007
	Cinema - news general	0x4008
	Cinema – charts	0x4009
	Cinema - tips	0x400A
	Cinema - tours	0x400B
	Cinema - latest release	0x400C
	Video - news general	0x400D
	Video – charts	0x400E
	Video – tips	0x400F
	Video - tours	0x4010
	Video - latest release	0x4011
	Reserved for future use	0x4012 - 0x47FF
Travel	Travel (top level)	0x4800
	Index	0x4801
(5 MSBs = 0x09)	Special	0x4802
	National	0x4803
	International	0x4804
	Short stories	0x4805
	Roads	0x4806
	Railways	0x4807
	Airlines	0x4808
	Airports	0x4809
	Ferries	0x480A
	Holidays	0x480B
	Reserved for future use	0x480C - 0x4FFF
Children	Children (top level)	0x5000
	Index	0x5001
(5 MSBs = 0x0A)		
	Reserved for future use	0x5002 - 0x57FF
"Topic 11"	"Topic 11" (top level)	0x5800
(Service provider defined)	Index	0x5801
(5 MSBs = 0x0B)	Reserved for future use	0x5802 - 0x5FFF
"Topic 12"	"Topic 12" (top level)	0x6000
(Service provider defined)	Index	0x6001
(5 MSBs = 0x0C)	Reserved for future use	0x6002 - 0x67FF

Implicit Subject Area	Theme	Identifier Value
"Topic 31"	"Topic 31" (top level)	0xF700
(Service provider defined)	Index	0xF701
(5 MSBs = 0x1F)	Reserved for future use	0xF702 - 0xFFFF

1

Annex D TeleWeb DTD, Document Type Definition

```
<!SGML "ISO 8879:1986"

-- SGML Declaration for TeleWeb Profile 1 Documents --
-- Revision: 1.00 --
-- Date: 2001-04-23 --

CHARSET
  BASESET "ISO 646:1983//CHARSET
    International Reference Version
    (IRV)//ESC 2/5 4/0"
  DESCSET 0 9 UNUSED
          9 2 9
          11 2 UNUSED
          13 1 13
          14 18 UNUSED
          32 95 32
          127 1 UNUSED
  BASESET "ISO Registration Number 100//CHARSET
    ECMA-94 Right Part of
    Latin Alphabet Nr. 1//ESC 2/13 4/1"
  DESCSET 128 32 UNUSED
          160 96 32
          55296 2048 UNUSED -- Unicode Surrogate Area --
          57344 7936 UNUSED -- Unicode Private Use Area --
          65280 240 32
          65520 14 UNUSED -- Unicode Special --
          65534 1 UNUSED -- Unicode Byte-Order Marker --
          65535 1 UNUSED -- Unicode Not-A-Character --

CAPACITY SGMLREF
          TOTALCAP 150000
          GRPCAP 150000
          ENTCAP 150000

SCOPE DOCUMENT
SYNTAX
  SHUNCHAR CONTROLS
    0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
    16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 127
  BASESET "ISO 646:1983//CHARSET
    International Reference Version
    (IRV)//ESC 2/5 4/0"
  DESCSET 0 128 0

FUNCTION
  RE 13
  RS 10
  SPACE 32
  TAB SEPCHAR 9

NAMING LCNMSTRT ""
       UCNMSTRT ""
       LCNMCHAR ".-:_"
       UCNMCHAR ".-:_"
       NAMECASE GENERAL YES
              ENTITY NO
DELIM GENERAL SGMLREF
      HCRO "&#38;#x" -- 4.0 extension --
      SHORTREF SGMLREF
NAMES SGMLREF
```

```

1      QUANTITY SGMLREF
2          ATTCNT      60
3          ATTSPLEN    65535
4          LITLEN      65535
5          NAMELEN     65535
6          PILEN       65535
7          TAGLVL      100
8          TAGLEN      65535
9          GRPGTCNT    150
10         GRPCNT      64
11
12     FEATURES
13         MINIMIZE
14             DATATAG    NO
15             OMITTAG    YES
16             RANK       NO
17             SHORTTAG   NO
18         LINK
19             SIMPLE     NO
20             IMPLICIT   NO
21             EXPLICIT   NO
22         OTHER
23             CONCUR    NO
24             SUBDOC     NO
25             FORMAL     YES
26         APPINFO       NONE
27     >
28
29     <!ENTITY % TeleWeb.version "Profile 1">
30
31     <!--===== Parameter Entities =====>
32
33     <!ENTITY % head.misc "SCRIPT | STYLE | LINK" >
34     <!ENTITY % heading "H1 | H2 | H3 | H4 | H5 | H6" >
35     <!ENTITY % list "UL | OL | DIR | MENU" >
36     <!ENTITY % preformatted "PRE" >
37
38     <!--===== Types =====>
39
40     <!ENTITY % Border "NUMBER" -- pixels, but BORDER means BORDER=1 -->
41     <!ENTITY % Color "CDATA" -- color specification: #RRGGBB -->
42     <!ENTITY % FontSize "CDATA" -- [+]nn, e.g. size="+1", size=4 -->
43     <!ENTITY % Length "CDATA" -- n for pixels or n% for percent -->
44     <!ENTITY % LiStyle "CDATA" -- constraint: "(%ULStyle|%OLStyle)" -->
45     <!ENTITY % LinkType "CDATA" -- see LINK element comment -->
46     <!ENTITY % OlStyle "CDATA" -- constrained to [laAiI] -->
47     <!ENTITY % Pixels "NUMBER" -- integer, length in pixels -->
48     <!ENTITY % URL "CDATA" -- uniform resource locator -->
49     <!ENTITY % UlStyle "(disc | square | circle)" -- bullet styles -->
50     <!ENTITY % AccessKey "CDATA" -- [0123456789RGYB]i -->
51     <!ENTITY % Transparency "NUMBER" -- 0 .. 100% -->
52
53
54     <!--===== Character Mnemonic Entities =====>
55
56     <!-- Portions (C) International Organization for Standardization 1986-->
57     <!-- Permission to copy in any form is granted for use with -->
58     <!-- conforming SGML systems and applications as defined in -->
59     <!-- ISO 8879, provided this notice is included in all copies. -->
60
61     <!ENTITY nbsp CDATA "&#160;" -- no-break space -->
62     <!ENTITY iexcl CDATA "&#161;" -- inverted exclamation mark -->
63     <!ENTITY cent CDATA "&#162;" -- cent sign -->
64     <!ENTITY pound CDATA "&#163;" -- pound sterling sign -->
65     <!ENTITY curren CDATA "&#164;" -- general currency sign -->
66     <!ENTITY yen CDATA "&#165;" -- yen sign = yuan sign -->
67     <!ENTITY brvbar CDATA "&#166;" -- broken (vertical) bar -->

```

1	<!ENTITY sect	CDATA	"§"	-- section sign	-->
2	<!ENTITY uml	CDATA	"¨"	-- umlaut (dieresis)	-->
3	<!ENTITY copy	CDATA	"©"	-- copyright sign	-->
4	<!ENTITY ordf	CDATA	"ª"	-- ordinal indicator, feminine	-->
5	<!ENTITY laquo	CDATA	"«"	-- angle quotation mark, left	-->
6	<!ENTITY not	CDATA	"¬"	-- not sign	-->
7	<!ENTITY shy	CDATA	"­"	-- soft hyphen	-->
8	<!ENTITY reg	CDATA	"®"	-- registered sign	-->
9	<!ENTITY macr	CDATA	"¯"	-- macron	-->
10	<!ENTITY deg	CDATA	"°"	-- degree sign	-->
11	<!ENTITY plusmn	CDATA	"±"	-- plus-or-minus sign	-->
12	<!ENTITY sup2	CDATA	"²"	-- superscript two	-->
13	<!ENTITY sup3	CDATA	"³"	-- superscript three	-->
14	<!ENTITY acute	CDATA	"´"	-- acute accent	-->
15	<!ENTITY micro	CDATA	"µ"	-- micro sign	-->
16	<!ENTITY para	CDATA	"¶"	-- pilcrow (paragraph sign)	-->
17	<!ENTITY middot	CDATA	"·"	-- middle dot	-->
18	<!ENTITY cedil	CDATA	"¸"	-- cedilla	-->
19	<!ENTITY sup1	CDATA	"¹"	-- superscript one	-->
20	<!ENTITY ordm	CDATA	"º"	-- ordinal indicator, masculine	-->
21	<!ENTITY raquo	CDATA	"»"	-- angle quotation mark, right	-->
22	<!ENTITY frac14	CDATA	"¼"	-- fraction one-quarter	-->
23	<!ENTITY frac12	CDATA	"½"	-- fraction one-half	-->
24	<!ENTITY frac34	CDATA	"¾"	-- fraction three-quarters	-->
25	<!ENTITY iquest	CDATA	"¿"	-- inverted question mark	-->
26	<!ENTITY Agrave	CDATA	"À"	-- capital A, grave accent	-->
27	<!ENTITY Aacute	CDATA	"Á"	-- capital A, acute accent	-->
28	<!ENTITY Acirc	CDATA	"Â"	-- capital A, circumflex accent	-->
29	<!ENTITY Atilde	CDATA	"Ã"	-- capital A, tilde	-->
30	<!ENTITY Auml	CDATA	"Ä"	-- capital A, dieresis or umlaut mark	-->
31	<!ENTITY Aring	CDATA	"Å"	-- capital A, ring	-->
32	<!ENTITY Aelig	CDATA	"Æ"	-- capital AE diphthong (ligature)	-->
33	<!ENTITY Ccedil	CDATA	"Ç"	-- capital C, cedilla	-->
34	<!ENTITY Egrave	CDATA	"È"	-- capital E, grave accent	-->
35	<!ENTITY Eacute	CDATA	"É"	-- capital E, acute accent	-->
36	<!ENTITY Ecirc	CDATA	"Ê"	-- capital E, circumflex accent	-->
37	<!ENTITY Euml	CDATA	"Ë"	-- capital E, dieresis or umlaut mark	-->
38	<!ENTITY Igrave	CDATA	"Ì"	-- capital I, grave accent	-->
39	<!ENTITY Iacute	CDATA	"Í"	-- capital I, acute accent	-->
40	<!ENTITY Icirc	CDATA	"Î"	-- capital I, circumflex accent	-->
41	<!ENTITY Iuml	CDATA	"Ï"	-- capital I, dieresis or umlaut mark	-->
42	<!ENTITY ETH	CDATA	"Ð"	-- capital Eth, Icelandic	-->
43	<!ENTITY Ntilde	CDATA	"Ñ"	-- capital N, tilde	-->
44	<!ENTITY Ograve	CDATA	"Ò"	-- capital O, grave accent	-->
45	<!ENTITY Oacute	CDATA	"Ó"	-- capital O, acute accent	-->
46	<!ENTITY Ocirc	CDATA	"Ô"	-- capital O, circumflex accent	-->
47	<!ENTITY Otilde	CDATA	"Õ"	-- capital O, tilde	-->
48	<!ENTITY Ouml	CDATA	"Ö"	-- capital O, dieresis or umlaut mark	-->
49	<!ENTITY times	CDATA	"×"	-- multiply sign	-->
50	<!ENTITY Oslash	CDATA	"Ø"	-- capital O, slash	-->
51	<!ENTITY Ugrave	CDATA	"Ù"	-- capital U, grave accent	-->
52	<!ENTITY Uacute	CDATA	"Ú"	-- capital U, acute accent	-->
53	<!ENTITY Ucirc	CDATA	"Û"	-- capital U, circumflex accent	-->
54	<!ENTITY Uuml	CDATA	"Ü"	-- capital U, dieresis or umlaut mark	-->
55	<!ENTITY Yacute	CDATA	"Ý"	-- capital Y, acute accent	-->
56	<!ENTITY THORN	CDATA	"Þ"	-- capital THORN, Icelandic	-->
57	<!ENTITY szlig	CDATA	"ß"	-- small sharp s, German sz ligature	-->
58	<!ENTITY agrave	CDATA	"à"	-- small a, grave accent	-->
59	<!ENTITY aacute	CDATA	"á"	-- small a, acute accent	-->
60	<!ENTITY acirc	CDATA	"â"	-- small a, circumflex accent	-->
61	<!ENTITY atilde	CDATA	"ã"	-- small a, tilde	-->
62	<!ENTITY auml	CDATA	"ä"	-- small a, dieresis or umlaut mark	-->
63	<!ENTITY aring	CDATA	"å"	-- small a, ring	-->
64	<!ENTITY aelig	CDATA	"æ"	-- small ae diphthong (ligature)	-->
65	<!ENTITY ccedil	CDATA	"ç"	-- small c, cedilla	-->
66	<!ENTITY egrave	CDATA	"è"	-- small e, grave accent	-->
67	<!ENTITY eacute	CDATA	"é"	-- small e, acute accent	-->

```

1 <!ENTITY ecirc CDATA "&#234;" -- small e, circumflex accent -->
2 <!ENTITY euml CDATA "&#235;" -- small e, dieresis or umlaut mark -->
3 <!ENTITY igrave CDATA "&#236;" -- small i, grave accent -->
4 <!ENTITY iacute CDATA "&#237;" -- small i, acute accent -->
5 <!ENTITY icirc CDATA "&#238;" -- small i, circumflex accent -->
6 <!ENTITY iuml CDATA "&#239;" -- small i, dieresis or umlaut mark -->
7 <!ENTITY eth CDATA "&#240;" -- small eth, Icelandic -->
8 <!ENTITY ntilde CDATA "&#241;" -- small n, tilde -->
9 <!ENTITY ograve CDATA "&#242;" -- small o, grave accent -->
10 <!ENTITY oacute CDATA "&#243;" -- small o, acute accent -->
11 <!ENTITY ocirc CDATA "&#244;" -- small o, circumflex accent -->
12 <!ENTITY otilde CDATA "&#245;" -- small o, tilde -->
13 <!ENTITY ouml CDATA "&#246;" -- small o, dieresis or umlaut mark -->
14 <!ENTITY divide CDATA "&#247;" -- divide sign -->
15 <!ENTITY oslash CDATA "&#248;" -- small o, slash -->
16 <!ENTITY ugrave CDATA "&#249;" -- small u, grave accent -->
17 <!ENTITY uacute CDATA "&#250;" -- small u, acute accent -->
18 <!ENTITY ucirc CDATA "&#251;" -- small u, circumflex accent -->
19 <!ENTITY uuml CDATA "&#252;" -- small u, dieresis or umlaut mark -->
20 <!ENTITY yacute CDATA "&#253;" -- small y, acute accent -->
21 <!ENTITY thorn CDATA "&#254;" -- small thorn, Icelandic -->
22 <!ENTITY yuml CDATA "&#255;" -- small y, dieresis or umlaut mark -->
23
24 <!--===== Currency Symbols =====>
25
26 <!ENTITY euro CDATA "&#8364;" -- euro sign -->
27
28 <!--===== Entities for Special Symbols =====>
29
30 <!ENTITY quot CDATA "&#34;" -- quotation mark -->
31 <!ENTITY QUOT CDATA "&#34;" -- quotation mark -->
32 <!ENTITY amp CDATA "&#38;" -- ampersand -->
33 <!ENTITY AMP CDATA "&#38;" -- ampersand -->
34 <!ENTITY lt CDATA "&#60;" -- less than -->
35 <!ENTITY LT CDATA "&#60;" -- less than -->
36 <!ENTITY gt CDATA "&#62;" -- greater than -->
37 <!ENTITY GT CDATA "&#62;" -- greater than -->
38
39
40 <!--===== Text Markup =====>
41
42 <!ENTITY % font
43 "TT | I | B | U | STRIKE | BIG | SMALL | SUB | SUP | BLINK"
44 >
45 <!ENTITY % phrase
46 "EM | STRONG | DFN | CODE | SAMP | KBD | VAR | CITE"
47 >
48 <!ENTITY % special
49 "A | IMG | APPLET | FONT | BASEFONT | BR | SCRIPT | MAP"
50 >
51 <!ENTITY % text
52 "#PCDATA | %font | %phrase | %special"
53 >
54
55 <!--===== Entities for Widely-Known Color Names =====>
56
57 <!-- There are also 16 widely known color names: -->
58 <!-- -->
59 <!-- aqua, black, blue, fuchsia, gray, green, lime, maroon, -->
60 <!-- navy, olive, purple, red, silver, teal, white, and yellow -->
61 <!-- -->
62 <!-- These colors were originally picked as being the standard -->
63 <!-- 16 colors supported with the Windows VGA palette. -->
64
65 <!ENTITY % vga_color
66 "aqua | black | blue | fuchsia | gray | green | lime |
67 maroon | navy | olive | purple | red | silver | teal |

```

```

1      white | yellow"
2      >
3
4      <!ENTITY % stdcolor "(&#x000000; | grey)" >
5
6      <!ENTITY % color "%stdcolor" >
7
8
9      <!ELEMENT (%font|&#x000000;) - - (&#x000000;)* >
10
11      <!-- NB: FONT contents promoted -->
12      <!ELEMENT FONT - - (&#x000000;)* -- local change to font -->
13      <!ATTLIST FONT
14          size          %fontsize          #IMPLIED
15          color         %Color             #IMPLIED
16      >
17
18      <!ELEMENT BASEFONT - O EMPTY -- base font size (1 to 7) -->
19      <!ATTLIST BASEFONT
20          size          %fontsize          #IMPLIED -- base font size --
21          color         %color            #IMPLIED -- base font color --
22      >
23
24      <!ELEMENT BR - O EMPTY -- forced line break -->
25      <!ATTLIST BR
26          clear        (left|all|right|none) none -- control of text flow --
27      >
28
29      <!--===== HTML content models =====>
30
31      <!-- HTML has three basic content models: -->
32      <!-- -->
33      <!-- %text character level elements and text strings -->
34      <!-- %flow block-like elements, e.g. paragraphs and lists -->
35      <!-- %body.content as %flow plus headers H1-H6 and ADDRESS -->
36
37      <!ENTITY % block
38          "P | &#x000000; | &#x000000; | DL | DIV | CENTER |
39          BLOCKQUOTE | FORM | ISINDEX | HR | TABLE | MARQUEE"
40      >
41
42      <!ENTITY % flow "(&#x000000; | %block)*" -- used for DD and LI -->
43
44      <!--===== Document Body =====>
45
46      <!ENTITY % body.content "(&#x000000; | %text | %block | ADDRESS)*" >
47
48      <!ELEMENT BODY O O %body.content>
49      <!ATTLIST BODY
50          background    %URL #IMPLIED -- tile for document background --
51          bgcolor       %Color #IMPLIED -- background color --
52          text          %Color #IMPLIED -- text color --
53          link          %Color #IMPLIED -- link color --
54          vlink         %Color #IMPLIED -- visited link color --
55          alink         %Color #IMPLIED -- active link color --
56          transparency %Transparency #IMPLIED
57          leftmargin    %pixels #IMPLIED -- left margin for whole page --
58          rightmargin   %pixels #IMPLIED -- right margin for whole page --
59          topmargin     %pixels #IMPLIED -- top margin for whole page --
60          bottommargin  %pixels #IMPLIED -- bottom margin for whole page --
61      >
62
63      <!ENTITY % address.content "((&#x000000;) | P)*" >
64
65      <!ELEMENT ADDRESS - - %address.content >
66
67      <!ENTITY % txtset "(left | center | right | justify)" >

```



```

1
2 <!ELEMENT DIV - - %body.content >
3 <!ATTLIST DIV
4     align    %txtset #IMPLIED          -- alignment of following text    --
5     >
6
7 <!ELEMENT CENTER - - %body.content -- short for DIV ALIGN=CENTER -->
8
9 <!--===== The Anchor Element =====>
10
11 <!ELEMENT A - - (%text)* -(A) >
12 <!ATTLIST A
13     name      CDATA      #IMPLIED      -- named link end          --
14     ....id..   CDATA      #IMPLIED      -- named link end          --
15     href       %URL       #IMPLIED      -- URL for linked resource  --
16     rel        %linktype  #IMPLIED      -- forward link type       --
17     rev        %linktype  #IMPLIED      -- reverse link type       --
18     title      CDATA      #IMPLIED      -- advisory title string   --
19     ....target (_SELF|_TOP) _SELF
20     accesskey  %AccessKey #IMPLIED      -- function key navigation  --
21     >
22
23 <!--===== Client-Side Image Maps =====>
24 <!-- These must be placed in the same document. -->
25
26 <!ENTITY % Shape      "(rect | circle | poly)" >
27 <!ENTITY % Coords     "CDATA"          -- comma separated list of numbers -->
28
29 <!ELEMENT MAP - - (AREA)* >
30 <!ATTLIST MAP
31     name      CDATA      #IMPLIED
32     >
33
34 <!ELEMENT AREA - O EMPTY >
35 <!ATTLIST AREA
36     shape      %shape     rect
37     ....id..   CDATA      #IMPLIED      -- named link end          --
38     coords     %coords    #IMPLIED      -- defines coordinates for shape --
39     href       %URL       #IMPLIED      -- this region acts as link  --
40     nohref     (nohref)   #IMPLIED      -- this region has no action --
41     alt        CDATA      #REQUIRED      -- for non-graphical user agents --
42     accesskey  %AccessKey #IMPLIED      -- function key navigation  --
43     >
44
45 <!--===== The Link Element =====>
46
47 <!ELEMENT LINK - O EMPTY >
48 <!ATTLIST LINK
49     href       %URL       #IMPLIED      -- URL for linked resource  --
50     rel        %linktype  #IMPLIED      -- forward link type       --
51     rev        %linktype  #IMPLIED      -- reverse link type       --
52     title      CDATA      #IMPLIED      -- advisory title string   --
53     >
54
55 <!--===== Images =====>
56
57 <!-- Suggested widths are used for negotiating image size -->
58 <!-- with the module responsible for painting the image. -->
59 <!-- ALIGN=LEFT or RIGHT cause image to float to margin -->
60 <!-- and for subsequent text to wrap around image. -->
61
62 <!ENTITY % IAlign
63     "( absbottom | absmiddle | baseline | bottom |
64     center | middle | right | top)" >
65
66
67 <!ELEMENT IMG - O EMPTY          -- Embedded image          -->

```

```

1  <!--ATTLIST IMG
2      src          %URL      #REQUIRED    -- URL of image to embed      --
3      lowsrc       %URL      #IMPLIED     -- ULR of image placeholder   --
4      alt          CDATA     #IMPLIED     -- for display in place of image --
5      align        %IAAlign  #IMPLIED     -- vertical or horiz. alignment --
6      height       %Pixels   #IMPLIED     -- suggested height in pixels  --
7      width        %Pixels   #IMPLIED     -- suggested width in pixels   --
8      border       %Border   #IMPLIED     -- suggested link border width --
9      hspace       %Pixels   #IMPLIED     -- suggested horizontal gutter --
10     vspace       %Pixels   #IMPLIED     -- suggested vertical gutter   --
11     usemap       %URL      #IMPLIED     -- use client-side image map   --
12     ismap        (ismap)   #IMPLIED     -- use server image map        --
13     transparency %Transparency #IMPLIED
14 >
15 <!-- USEMAP points to a MAP element which may be in this document -->
16 <!-- or, though not widely supported, in an external document. -->
17
18 <!--===== Java APPLET tag =====>
19 <!-- This tag is not supported by TeleWeb browsers, but proper -->
20 <!-- parsing can only be achieved by recognising the expected -->
21 <!-- syntax. -->
22
23 <!--ELEMENT APPLET - - (PARAM | %text)* >
24
25 <!--ELEMENT PARAM - O EMPTY >
26
27 <!--===== Horizontal Rule =====>
28
29 <!--ELEMENT HR - O EMPTY >
30 <!--ATTLIST HR
31     align        (left|right|center) #IMPLIED
32     noshade       (noshade) #IMPLIED
33     size         %Pixels #IMPLIED
34     width        %Length #IMPLIED
35     color        %Color #IMPLIED
36 >
37
38 <!--===== Paragraph =====>
39
40 <!--ELEMENT P - O (%text)* >
41 <!--ATTLIST P
42     align        %txtset #IMPLIED
43 >
44
45 <!--===== Headings =====>
46 <!-- There are six levels of headers from H1 (the most important) -->
47 <!-- to H6 (the least important). -->
48
49 <!--ELEMENT ( %heading ) - - (%text;)* >
50 <!--ATTLIST ( %heading )
51     align        %txtset #IMPLIED
52 >
53
54 <!--===== Preformatted Text =====>
55
56 <!-- excludes images and changes in font size -->
57 <!--ENTITY % pre.exclusion "IMG | BIG | SMALL | SUB | SUP | FONT" >
58
59 <!--ELEMENT PRE - - (%text)* -(%pre.exclusion) >
60 <!--ATTLIST PRE
61     width        NUMBER #IMPLIED
62 >
63
64 <!--===== Block-like Quotes =====>
65
66 <!--ELEMENT BLOCKQUOTE - - %body.content >
67

```

```

1  <!--===== Lists =====-->
2  <!-- HTML 3.2 lets you control the sequence number for ordered -->
3  <!-- lists. You can set the sequence number with the START and -->
4  <!-- VALUE attributes. The TYPE attribute may be used to specify -->
5  <!-- the rendering of ordered and unordered lists. -->
6
7  <!-- Definition lists - DT for term, DD for its definition -->
8  <!ELEMENT DL - - (DT|DD)+ >
9  <!ATTLIST DL
10     compact      (compact) #IMPLIED -- more compact style --
11     >
12
13  <!ELEMENT DT - O (%text)* >
14  <!ELEMENT DD - O %flow; >
15
16  <!-- Ordered lists OL, and unordered lists UL -->
17  <!ELEMENT (OL|UL) - - (LI)+ >
18
19  <!-- Numbering style -->
20  <!-- 1 Arabic numbers 1, 2, 3, ... -->
21  <!-- a lower alpha a, b, c, ... -->
22  <!-- A upper alpha A, B, C, ... -->
23  <!-- i lower Roman i, ii, iii, ... -->
24  <!-- I upper Roman I, II, III, ... -->
25  <!-- -->
26  <!-- The style is applied to the sequence number which by default -->
27  <!-- is reset to 1 for the first list item in an ordered list. -->
28  <!-- -->
29  <!-- This can't be expressed directly in SGML due to case folding. -->
30
31  <!ATTLIST OL -- Ordered Lists --
32     type      %OLStyle      #IMPLIED -- numbering style --
33     start     NUMBER        #IMPLIED -- starting sequence number --
34     compact   (compact)     #IMPLIED -- reduced interitem spacing --
35     >
36
37  <!ATTLIST UL -- Unordered Lists --
38     type      %ULStyle      #IMPLIED -- bullet style --
39     compact   (compact)     #IMPLIED -- reduced interitem spacing --
40     >
41
42  <!ELEMENT (DIR|MENU) - - (LI)+ -(%block) >
43  <!ATTLIST DIR
44     compact   (compact)     #IMPLIED
45     >
46  <!ATTLIST MENU
47     compact   (compact)     #IMPLIED
48     >
49
50
51  <!-- The type attribute can be used to change the bullet style -->
52  <!-- in unordered lists and the numbering style in ordered lists -->
53
54  <!ELEMENT LI - O %flow -- list item -->
55  <!ATTLIST LI
56     type      %LISyle      #IMPLIED -- list item style --
57     value     NUMBER        #IMPLIED -- reset sequence number --
58     >
59
60  <!--===== Forms =====-->
61  <!-- This tag is not supported by TeleWeb Profile 1 browsers -->
62  <!-- The whole content of form elements must be ignored. -->
63
64  <!ELEMENT FORM - - %body.content -(FORM) >
65
66  <!--===== Tables =====-->
67  <!-- Widely deployed subset of the full table standard, see RFC 1942 -->

```

```

1  <!-- e.g., at http://www.ics.uci.edu/pub/ietf/html/rfc1942.txt      -->
2
3  <!-- horizontal placement of table relative to window              -->
4  <!ENTITY % Where          "(left | center | right)" >
5
6  <!-- horizontal alignment attributes for cell contents              -->
7  <!ENTITY % cell.halign    "align %txtset #IMPLIED" >
8
9  <!-- vertical alignment attributes for cell contents                -->
10 <!ENTITY % cell.valign     "valign (top | middle | bottom)  #IMPLIED" >
11
12 <!ELEMENT TABLE          - - (CAPTION?, TR+) >
13 <!ELEMENT TR              - O (TH|TD)* >
14 <!ELEMENT (TH|TD)        - O %body.content >
15
16 <!ATTLIST TABLE          -- Table Element                        --
17   align                   %Where  #IMPLIED    -- table position in window --
18   width                   %Length #IMPLIED    -- table width in window   --
19   height                  %Pixels #IMPLIED    -- table height in pixels  --
20   border                  %Border #IMPLIED    -- frame width around table --
21   cellspacing             %Pixels #IMPLIED    -- spacing between cells   --
22   cellpadding            %Pixels #IMPLIED    -- spacing within cells    --
23   bgcolor                 %Color  #IMPLIED    -- background color        --
24   background              %URL    #IMPLIED    -- background image        --
25   transparency            %Transparency #IMPLIED
26 >
27
28 <!ELEMENT CAPTION - - (%text;)*          -- table or figure caption -->
29 <!ATTLIST CAPTION
30   align (top|bottom|left|center|right)  #IMPLIED
31   valign (top|bottom)                   #IMPLIED
32 >
33
34 <!ATTLIST TR              -- table row                            --
35   %cell.halign            -- horizontal alignment in cells      --
36   %cell.valign            -- vertical alignment in cells        --
37   bgcolor                 %Color  #IMPLIED    -- background color        --
38   transparency            %Transparency #IMPLIED
39 >
40
41 <!ATTLIST (TH|TD)         -- header or data cell                --
42   nowrap                  (nowrap) #IMPLIED    -- suppress word wrap      --
43   rowspan                 NUMBER    1          -- number of rows spanned  --
44   colspan                 NUMBER    1          -- number of cols spanned  --
45   %cell.halign            -- horizontal alignment in cell       --
46   %cell.valign            -- vertical alignment in cell         --
47   width                   %Pixels #IMPLIED    -- suggested width for cell --
48   height                  %Pixels #IMPLIED    -- suggested height for cell --
49   bgcolor                 %Color  #IMPLIED    -- background color        --
50   background              %URL    #IMPLIED    -- background image        --
51   transparency            %Transparency #IMPLIED
52 >
53
54 <!--===== Document Head =====>
55
56 <!-- %head.misc defined earlier on as "SCRIPT|STYLE|LINK"          -->
57
58 <!ENTITY % head.content    "TITLE & ISINDEX? & BASE?" >
59
60 <!ELEMENT HEAD O O  (%head.content) +(%head.misc) >
61
62 <!-- The TITLE element is not part of the flow of text.  It should -->
63 <!-- be displayed, for example, as the page header or window title. -->
64
65 <!ELEMENT TITLE - - (#PCDATA)* -(%head.misc) >
66
67 <!ELEMENT ISINDEX - O EMPTY >

```

```

1
2 <!-- The BASE element gives an absolute URL for dereferencing -->
3 <!-- relative URLs. -->
4 <!-- -->
5 <!-- In the absence of a BASE element the document URL should be -->
6 <!-- used. Note that this is not necessarily the same as the URL -->
7 <!-- used to request the document, as the base URL may be -->
8 <!-- overridden by an HTTP header accompanying the document. -->
9
10 <!ELEMENT BASE - O EMPTY >
11 <!ATTLIST BASE
12     href      %URL      #REQUIRED
13     >
14
15 <!-- SCRIPT/STYLE are place holders for transition to Profile 2/3 -->
16
17 <!ELEMENT STYLE - - CDATA -- placeholder for style info -->
18 <!ELEMENT SCRIPT - - CDATA -- placeholder for script statements -->
19
20 <!--===== Document Structure =====>
21
22 <!ENTITY % attr.version "VERSION CDATA #FIXED '%TeleWeb.version;'" >
23
24 <!ENTITY % html.content "HEAD, BODY" >
25
26 <!ELEMENT HTML O O (%html.content) >
27
28 <!--===== TeleWeb Extensions =====>
29
30 <!ELEMENT MARQUEE - - (#PCDATA | FONT | %font)* >
31
32 <!-- Note: a marquee cannot contain anchors, images, scripts, maps -->
33 <!-- or other inappropriate elements. It can only contain text and -->
34 <!-- font requests. -->
35
36 <!ATTLIST MARQUEE
37     behavior      (alternate | scroll | slide)      scroll
38     bgcolor        %Color      #IMPLIED
39     direction      (left | right | up | down)      #IMPLIED
40     height         %Length     #IMPLIED
41     hspace         %Pixels     0
42     loop           NUMBER      1
43     scrollamount    %Pixels     3
44     scrolldelay     NUMBER      16 -- ms --
45     transparency   %Transparency #IMPLIED
46     vspace         %Pixels     0
47     width          %Length     #IMPLIED
48     >
49
50 <!-- The End -->
51

```

Annex E TeleWeb Default CSS2 style sheet

TeleWeb does not support Style sheets. The style sheet is used here as syntax to describe the default behaviour.

The following style sheet describes the default style sheet for TeleWeb . Browsers which do not support the italic font style shall use the plain style instead.

```
A:link      IMG
            { border-width:  0px;
              border-style:  solid;
              solid border-color:  blue }

A:visited  IMG
            { border-width:  0px;
              border-style:  solid;
              border-color:  blue }

A:active   IMG
            { border-width:  0px;
              border-style:  solid;
              border-color:  blue }

ADDRESS    { font-style:    italic }

B          { font-weight:   bolder }

BIG        { font-size:     larger }

BLOCKQUOTE { margin-left:   35px;
              margin-right:  35px }

BODY       { margin-left:    9px;
              margin-right:   9px;
              margin-top:     15px;
              margin-bottom:  15px;
              font-family:    proportional;
              font-size:      small;
              font-style:     normal;
              font-weight:    normal;
              background:     #D4D4D4;
              color:          black }

CAPTION    { margin-bottom:  10px }

CENTER     { text-align:    center }

CITE       { font-style:    italic }

CODE       { font-family:   monospace }

DD         { margin-left:    50px }

DFN        { font-family:   italic }

DIV        { }

DIR        { list-style:    disc }

DT         { }

EM         { font-style:    italic }

H1         { margin-top:     15px;
              margin-bottom:  15px;
              font-size:      x-large;
              font-weight:    bold }

H2         { margin-top:     15px;
```

```

1      margin-bottom: 15px;
2      font-size:      large;
3      font-weight:    bold }
4
5  H3      { margin-top:    15px;
6           margin-bottom: 15px;
7           font-size:     medium;
8           font-weight:   bold }
9
10 H4      { margin-top:    15px;
11           margin-bottom: 15px;
12           font-size:     small;
13           font-weight:   bold }
14
15 H5      { margin-top:    15px;
16           margin-bottom: 15px;
17           font-size:     x-small;
18           font-weight:   bold }
19
20 H6      { margin-top:    15px;
21           margin-bottom: 15px;
22           font-size:     xx-small;
23           font-weight:   bold }
24
25 HR      { margin-top:    6px;
26           margin-bottom: 6px;
27           text-align:    center;
28           border-top-color: #555555;
29           border-right-color: #AAAAAA;
30           border-bottom-color: #AAAAAA;
31           border-left-color: #555555 }
32
33 I        { font-style:    italic }
34
35 KBD      { font-family:   monospace }
36
37 LI       { margin-left:   35px }
38
39 MENU     { }
40
41 OL       { list-style:    decimal;
42           margin-top:     0px }
43
44 P        { margin-top:    18px;
45           margin-bottom: 18px }
46
47 PRE      { margin-top:    18px;
48           margin-bottom: 18px;
49           font-family:   monospace;
50           white-space:   pre }
51
52 SAMP     { font-family:   monospace }
53
54 SMALL    { font-size:     smaller }
55
56 STRIKE   { text-decoration: line-through }
57
58 STRONG   { font-weight:   bolder }
59
60 SUB      { vertical-align: sub }
61
62 SUP      { vertical-align: super }
63
64 TD       { vertical-align: middle;
65           text-align:    left }
66
67 TH       { vertical-align: middle;
68           font-weight:   bolder;
69           text-align:    center }
70
71 TT       { font-family:   monospace }

```

```
1
2  U          { text-decoration: underline }
3
4  UL         { list-style:    disc  }
5
6  UL UL      { list-style:    circle }
7
8  UL UL UL   { list-style:    square }
9
10 VAR        { font-style:    italic }
11
```


Annex F Font Metrics

The following tables define the metrics of all fonts. There is one proportional and one monospaced font each in 5 different sizes. For the proportional font the advances of all glyphs are listed. For the monospaced font the advances of all glyphs are equal. If bold or italics styles are supported the same metrics as for the plain styles shall be used.

Generic Name = sans-serif
Type = Proportional
HTML Size = 2 pixels
Size = 22 pixels
Ascent = 17 pixels
Descent = 5 pixels
Text Lines = 22.9
Line Gap = -1

Table of Glyph Advances:

32	' '	6	33	'!'	6	34	'"'	8	35	'#'	16	36	'\$'	11	37	'%'	14
38	'&'	12	39	'''	5	40	'('	8	41	')'	8	42	'*'	7	43	'+'	13
44	','	5	45	'-'	7	46	'.'	5	47	'/'	7	48	'0'	11	49	'1'	11
50	'2'	11	51	'3'	11	52	'4'	11	53	'5'	11	54	'6'	11	55	'7'	11
56	'8'	11	57	'9'	11	58	':'	5	59	';'	5	60	'<'	13	61	'='	13
62	'>'	13	63	'?'	9	64	'@'	15	65	'A'	11	66	'B'	11	67	'C'	9
68	'D'	12	69	'E'	10	70	'F'	9	71	'G'	11	72	'H'	12	73	'I'	5
74	'J'	5	75	'K'	10	76	'L'	9	77	'M'	15	78	'N'	13	79	'O'	12
80	'P'	11	81	'Q'	12	82	'R'	11	83	'S'	10	84	'T'	9	85	'U'	12
86	'V'	11	87	'W'	15	88	'X'	10	89	'Y'	10	90	'Z'	9	91	'['	8
92	'\'	7	93	']'	8	94	'^'	11	95	'_'	8	96	'`'	10	97	'a'	9
98	'b'	10	99	'c'	8	100	'd'	10	101	'e'	10	102	'f'	6	103	'g'	10
104	'h'	10	105	'i'	5	106	'j'	5	107	'k'	9	108	'l'	5	109	'm'	16
110	'n'	10	111	'o'	10	112	'p'	10	113	'q'	10	114	'r'	7	115	's'	9
116	't'	7	117	'u'	10	118	'v'	9	119	'w'	13	120	'x'	9	121	'y'	9
122	'z'	8	123	'{'	6	124	' '	4	125	'}'	6	126	'~'	13			
160	' '̂'	11	161	' '̇'	6	162	' '̈'	11	163	' '̉'	11	164	' '̊'	11	165	' '̋'	11
166	' '̌'	11	167	' '̍'	11	168	' '̎'	10	169	' '̏'	15	170	' '̐'	11	171	' '̑'	9
172	' '̒'	13	173	' '̓'	7	174	' '̔'	15	175	' '̕'	11	176	' '̖'	8	177	' '̗'	11
178	' '̙'	11	179	' '̚'	11	180	' '̛'	10	181	' '̜'	11	182	' '̝'	12	183	' '̞'	5
184	' '̠'	11	185	' '̡'	11	186	' '̢'	11	187	' '̣'	9	188	' '̤'	17	189	' '̥'	17
190	' '̦'	17	191	' '̧'	9	192	' '̨'	11	193	' '̩'	11	194	' '̪'	11	195	' '̫'	11
196	' '̬'	11	197	' '̭'	11	198	' '̮'	16	199	' '̯'	9	200	' '̰'	10	201	' '̱'	10
202	' '̲'	10	203	' '̳'	10	204	' '̴'	5	205	' '̵'	5	206	' '̶'	5	207	' '̷'	5
208	' '̹'	11	209	' '̺'	13	210	' '̻'	12	211	' '̼'	12	212	' '̽'	12	213	' '̾'	12
214	' '̿'	12	215	' '̀'	13	216	' '́'	12	217	' '̂'	12	218	' '̃'	12	219	' '̄'	12
220	' '̆'	12	221	' '̇'	10	222	' '̈'	11	223	' '̉'	11	224	' '̊'	9	225	' '̋'	9
226	' '̌'	9	227	' '̍'	9	228	' '̎'	9	229	' '̏'	9	230	' '̐'	15	231	' '̑'	8
232	' '̒'	10	233	' '̓'	10	234	' '̔'	10	235	' '̕'	10	236	' '̖'	5	237	' '̗'	5
238	' '̙'	5	239	' '̚'	5	240	' '̛'	11	241	' '̜'	10	242	' '̝'	10	243	' '̞'	10
244	' '̠'	10	245	' '̡'	10	246	' '̢'	10	247	' '̣'	13	248	' '̤'	10	249	' '̥'	10
250	' '̦'	10	251	' '̧'	10	252	' '̨'	10	253	' '̩'	9	254	' '̪'	11	255	' '̫'	9

```

1 -----
2 Generic Name = sans-serif
3 Type        = Proportional
4 HTML Size   = 3
5 Size        = 24 pixels
6 Ascent      = 19 pixels
7 Descent     = 5 pixels
8 Text Lines  = 20.9
9 Line Gap    = -1
10
11 Table of Glyph Advances:
12
13 32 ' ' 6    33 '!' 6    34 '"' 9    35 '#' 17    36 '$' 12    37 '%' 15
14 38 '&' 14   39 ''' 5    40 '(' 8    41 ')' 8    42 '*' 7    43 '+' 15
15 44 ',' 5    45 '-' 8    46 '.' 6    47 '/' 8    48 '0' 12   49 '1' 12
16 50 '2' 12   51 '3' 12   52 '4' 12   53 '5' 12   54 '6' 12   55 '7' 12
17 56 '8' 12   57 '9' 12   58 ':' 6    59 ';' 5    60 '<' 15   61 '=' 15
18 62 '>' 15   63 '?' 10   64 '@' 17   65 'A' 12   66 'B' 12   67 'C' 10
19 68 'D' 13   69 'E' 11   70 'F' 10   71 'G' 12   72 'H' 13   73 'I' 6
20 74 'J' 6    75 'K' 12   76 'L' 9    77 'M' 17   78 'N' 14   79 'O' 14
21 80 'P' 12   81 'Q' 14   82 'R' 12   83 'S' 11   84 'T' 10   85 'U' 13
22 86 'V' 12   87 'W' 17   88 'X' 12   89 'Y' 11   90 'Z' 10   91 '[' 8
23 92 '\' 8    93 ']' 8    94 '^' 13   95 '_' 9    96 '`' 11   97 'a' 10
24 98 'b' 11   99 'c' 9    100 'd' 11  101 'e' 11  102 'f' 7    103 'g' 11
25 104 'h' 11  105 'i' 5   106 'j' 5   107 'k' 10  108 'l' 6    109 'm' 17
26 110 'n' 11  111 'o' 11  112 'p' 11  113 'q' 11  114 'r' 7    115 's' 9
27 116 't' 7   117 'u' 11  118 'v' 10  119 'w' 14  120 'x' 10   121 'y' 10
28 122 'z' 9   123 '{' 7   124 '|' 5   125 '}' 7   126 '~' 15
29
30 160 ' ' 12   161 'ı' 6    162 'ç' 12   163 '£' 12   164 '¤' 13   165 '¥' 12
31 166 'ı' 13   167 '§' 13   168 '¨' 11   169 '©' 17   170 'ª' 13   171 '«' 10
32 172 '¬' 15   173 '¯' 8    174 '®' 17   175 '™' 13   176 '°' 9    177 '±' 13
33 178 '²' 13   179 '³' 13   180 '´' 11   181 'µ' 13   182 '¶' 13   183 '·' 6
34 184 '½' 13   185 '¹' 13   186 'º' 13   187 '»' 10   188 '¼' 19   189 '½' 19
35 190 '¾' 19   191 '¿' 10   192 'À' 12   193 'Á' 12   194 'Â' 12   195 'Ã' 12
36 196 'Ä' 12   197 'Å' 12   198 'Æ' 18   199 'Ç' 10   200 'È' 11   201 'É' 11
37 202 'Ê' 11   203 'Ë' 11   204 'Ì' 6    205 'Í' 6    206 'Î' 6    207 'Ï' 6
38 208 'Ð' 13   209 'Ñ' 14   210 'Ò' 14   211 'Ó' 14   212 'Ô' 14   213 'Õ' 14
39 214 'Ö' 14   215 '×' 15   216 'Ø' 14   217 'Ù' 13   218 'Ú' 13   219 'Û' 13
40 220 'Ü' 13   221 'Ý' 11   222 'Þ' 13   223 'ß' 12   224 'à' 10   225 'á' 10
41 226 'â' 10   227 'ã' 10   228 'ä' 10   229 'å' 10   230 'æ' 16   231 'ç' 9
42 232 'è' 11   233 'é' 11   234 'ê' 11   235 'ë' 11   236 'ì' 5    237 'í' 5
43 238 'î' 5    239 'ï' 5    240 'ð' 13   241 'ñ' 11   242 'ò' 11   243 'ó' 11
44 244 'ô' 11   245 'õ' 11   246 'ö' 11   247 '÷' 15   248 'ø' 11   249 'ù' 11
45 250 'ú' 11   251 'û' 11   252 'ü' 11   253 'ý' 10   254 'þ' 13   255 'ÿ' 10
46
47

```

```

1 -----
2 Generic Name = sans-serif
3 Type        = Proportional
4 HTML Size   = 4
5 Size        = 27 pixels
6 Ascent      = 21 pixels
7 Descent     = 6 pixels
8 Text Lines  = 18.5
9 Line Gap    = -1
10
11 Table of Glyph Advances:
12
13 32 ' ' 7 33 '!' 7 34 '"' 9 35 '#' 19 36 '$' 14 37 '%' 17
14 38 '&' 15 39 ''' 6 40 '(' 9 41 ')' 9 42 '*' 8 43 '+' 16
15 44 ',' 6 45 '-' 9 46 '.' 6 47 '/' 8 48 '0' 14 49 '1' 14
16 50 '2' 14 51 '3' 14 52 '4' 14 53 '5' 14 54 '6' 14 55 '7' 14
17 56 '8' 14 57 '9' 14 58 ':' 6 59 ';' 6 60 '<' 16 61 '=' 16
18 62 '>' 16 63 '?' 10 64 '@' 19 65 'A' 13 66 'B' 13 67 'C' 11
19 68 'D' 14 69 'E' 12 70 'F' 11 71 'G' 14 72 'H' 14 73 'I' 7
20 74 'J' 7 75 'K' 13 76 'L' 10 77 'M' 18 78 'N' 15 79 'O' 15
21 80 'P' 13 81 'Q' 15 82 'R' 13 83 'S' 12 84 'T' 11 85 'U' 14
22 86 'V' 13 87 'W' 19 88 'X' 13 89 'Y' 12 90 'Z' 11 91 '[' 9
23 92 '\' 8 93 ']' 9 94 '^' 14 95 '_' 10 96 '`' 12 97 'a' 11
24 98 'b' 12 99 'c' 10 100 'd' 12 101 'e' 12 102 'f' 7 103 'g' 12
25 104 'h' 12 105 'i' 6 106 'j' 6 107 'k' 11 108 'l' 6 109 'm' 19
26 110 'n' 12 111 'o' 12 112 'p' 12 113 'q' 12 114 'r' 8 115 's' 10
27 116 't' 8 117 'u' 12 118 'v' 11 119 'w' 15 120 'x' 11 121 'y' 11
28 122 'z' 10 123 '{' 8 124 '|' 5 125 '}' 8 126 '~' 16
29
30 160 ' ' 14 161 'ı' 7 162 'ç' 14 163 '£' 14 164 '¤' 14 165 '¥' 14
31 166 'ı' 14 167 '§' 14 168 '¨' 12 169 '©' 18 170 'ª' 14 171 '«' 11
32 172 '¬' 16 173 '¯' 9 174 '®' 18 175 '™' 14 176 '°' 9 177 '±' 14
33 178 '²' 14 179 '³' 14 180 '´' 12 181 'µ' 14 182 '¶' 15 183 '·' 6
34 184 '¸' 14 185 '¹' 14 186 'º' 14 187 '»' 11 188 '¼' 21 189 '½' 21
35 190 '¾' 21 191 '¿' 10 192 'À' 13 193 'Á' 13 194 'Â' 13 195 'Ã' 13
36 196 'Ä' 13 197 'Å' 13 198 'Æ' 19 199 'Ç' 11 200 'È' 12 201 'É' 12
37 202 'Ê' 12 203 'Ë' 12 204 'Ì' 7 205 'Í' 7 206 'Î' 7 207 'Ï' 7
38 208 'Ð' 14 209 'Ñ' 15 210 'Ò' 15 211 'Ó' 15 212 'Ô' 15 213 'Õ' 15
39 214 'Ö' 15 215 '×' 16 216 'Ø' 15 217 'Ù' 14 218 'Ú' 14 219 'Û' 14
40 220 'Ü' 14 221 'Ý' 12 222 'Þ' 14 223 'ß' 13 224 'à' 11 225 'á' 11
41 226 'â' 11 227 'ã' 11 228 'ä' 11 229 'å' 11 230 'æ' 18 231 'ç' 10
42 232 'è' 12 233 'é' 12 234 'ê' 12 235 'ë' 12 236 'ì' 6 237 'í' 6
43 238 'î' 6 239 'ï' 6 240 'ð' 14 241 'ñ' 12 242 'ò' 12 243 'ó' 12
44 244 'ô' 12 245 'õ' 12 246 'ö' 12 247 '÷' 16 248 'ø' 12 249 'ù' 12
45 250 'ú' 12 251 'û' 12 252 'ü' 12 253 'ý' 11 254 'þ' 14 255 'ÿ' 11
46
47

```

```

1 -----
2 Generic Name = sans-serif
3 Type        = Proportional
4 HTML Size   = 5
5 Size        = 31 pixels
6 Ascent      = 24 pixels
7 Descent     = 7 pixels
8 Text Lines  = 16.0
9 Line Gap    = -1
10
11 Table of Glyph Advances:
12
13 32 ' ' 8    33 '!' 8    34 '"' 11    35 '#' 22    36 '$' 16    37 '%' 20
14 38 '&' 18   39 ''' 7    40 '(' 11    41 ')' 11    42 '*' 10    43 '+' 19
15 44 ',' 7    45 '-' 10   46 '.' 7    47 '/' 10    48 '0' 16    49 '1' 16
16 50 '2' 16   51 '3' 16   52 '4' 16   53 '5' 16   54 '6' 16   55 '7' 16
17 56 '8' 16   57 '9' 16   58 ':' 7    59 ';' 7    60 '<' 19    61 '=' 19
18 62 '>' 19   63 '?' 12   64 '@' 22   65 'A' 16   66 'B' 16   67 'C' 13
19 68 'D' 16   69 'E' 14   70 'F' 13   71 'G' 16   72 'H' 17   73 'I' 8
20 74 'J' 8    75 'K' 15   76 'L' 12   77 'M' 21   78 'N' 18   79 'O' 18
21 80 'P' 15   81 'Q' 18   82 'R' 16   83 'S' 14   84 'T' 13   85 'U' 17
22 86 'V' 16   87 'W' 22   88 'X' 15   89 'Y' 15   90 'Z' 13   91 '[' 11
23 92 '\' 10   93 ']' 11   94 '^' 16   95 '_' 12   96 '`' 14   97 'a' 13
24 98 'b' 14   99 'c' 11  100 'd' 14  101 'e' 14  102 'f' 9    103 'g' 14
25 104 'h' 14  105 'i' 7   106 'j' 7   107 'k' 13  108 'l' 8    109 'm' 22
26 110 'n' 14  111 'o' 14  112 'p' 14  113 'q' 14  114 'r' 9    115 's' 12
27 116 't' 9   117 'u' 14  118 'v' 13  119 'w' 18  120 'x' 13   121 'y' 13
28 122 'z' 11  123 '{' 9   124 '|' 6   125 '}' 9    126 '~' 19
29
30 160 ' ' 16   161 'ı' 8    162 'ç' 16   163 '£' 16   164 '¤' 16   165 '¥' 16
31 166 'ı' 16   167 '§' 16   168 '¨' 14   169 '©' 22   170 'ª' 16   171 '«' 13
32 172 '¬' 19   173 '¯' 10   174 '®' 22   175 '™' 16   176 '°' 11   177 '±' 16
33 178 '²' 16   179 '³' 16   180 '´' 14   181 'µ' 16   182 '¶' 17   183 '·' 7
34 184 '¸' 16   185 '¹' 16   186 'º' 16   187 '»' 13   188 '¼' 24   189 '½' 24
35 190 '¾' 24   191 '¿' 12   192 'À' 16   193 'Á' 16   194 'Â' 16   195 'Ã' 16
36 196 'Ä' 16   197 'Å' 16   198 'Æ' 23   199 'Ç' 13   200 'È' 14   201 'É' 14
37 202 'Ê' 14   203 'Ë' 14   204 'Ì' 8    205 'Í' 8    206 'Î' 8    207 'Ï' 8
38 208 'Ð' 16   209 'Ñ' 18   210 'Ò' 18   211 'Ó' 18   212 'Ô' 18   213 'Õ' 18
39 214 'Ö' 18   215 '×' 19   216 'Ø' 18   217 'Ù' 17   218 'Ú' 17   219 'Û' 17
40 220 'Ü' 17   221 'Ý' 15   222 'Þ' 16   223 'ß' 16   224 'à' 13   225 'á' 13
41 226 'â' 13   227 'ã' 13   228 'ä' 13   229 'å' 13   230 'æ' 21   231 'ç' 11
42 232 'è' 14   233 'é' 14   234 'ê' 14   235 'ë' 14   236 'ì' 7    237 'í' 7
43 238 'î' 7    239 'ï' 7    240 'ð' 16   241 'ñ' 14   242 'ò' 14   243 'ó' 14
44 244 'ô' 14   245 'õ' 14   246 'ö' 14   247 '÷' 19   248 'ø' 14   249 'ù' 14
45 250 'ú' 14   251 'û' 14   252 'ü' 14   253 'ý' 13   254 'þ' 16   255 'ÿ' 13
46
47

```

```

1  -----
2  Generic Name = sans-serif
3  Type        = Proportional
4  HTML Size   = 6, 7
5  Size        = 36 pixels
6  Ascent      = 28 pixels
7  Descent     = 8 pixels
8  Text Lines  = 13.7
9  Line Gap    = -1
10
11 Table of Glyph Advances:
12
13   32 ' ' 9    33 '!' 9    34 '"' 13    35 '#' 26    36 '$' 18    37 '%' 23
14   38 '&' 20   39 ''' 8    40 '(' 12    41 ')' 12    42 '*' 11    43 '+' 22
15   44 ',' 8    45 '-' 12   46 '.' 8     47 '/' 11    48 '0' 18    49 '1' 18
16   50 '2' 18   51 '3' 18   52 '4' 18    53 '5' 18    54 '6' 18    55 '7' 18
17   56 '8' 18   57 '9' 18   58 ':' 8     59 ';' 8     60 '<' 22    61 '=' 22
18   62 '>' 22   63 '?' 14   64 '@' 25    65 'A' 18    66 'B' 18    67 'C' 15
19   68 'D' 19   69 'E' 16   70 'F' 15    71 'G' 18    72 'H' 19    73 'I' 9
20   74 'J' 9    75 'K' 17   76 'L' 14    77 'M' 25    78 'N' 21    79 'O' 20
21   80 'P' 17   81 'Q' 20   82 'R' 18    83 'S' 17    84 'T' 15    85 'U' 19
22   86 'V' 18   87 'W' 25   88 'X' 17    89 'Y' 17    90 'Z' 15    91 '[' 13
23   92 '\' 11   93 ']' 13   94 '^' 19    95 '_' 13    96 '`' 16    97 'a' 15
24   98 'b' 16   99 'c' 13  100 'd' 16   101 'e' 16   102 'f' 10   103 'g' 16
25  104 'h' 17  105 'i' 8   106 'j' 8    107 'k' 15   108 'l' 9    109 'm' 25
26  110 'n' 17  111 'o' 16   112 'p' 16   113 'q' 16   114 'r' 11   115 's' 14
27  116 't' 11  117 'u' 17   118 'v' 14   119 'w' 21   120 'x' 15   121 'y' 14
28  122 'z' 13  123 '{' 10   124 '|' 7    125 '}' 10   126 '~' 22
29
30  160 ' ' 18   161 '¡' 9    162 '¢' 18   163 '£' 18   164 '¤' 19   165 '¥' 18
31  166 '¦' 19   167 '§' 19   168 '¨' 16   169 '©' 25   170 'ª' 19   171 '«' 14
32  172 '¬' 22   173 '­' 12   174 '®' 25   175 '¯' 19   176 '°' 13   177 '±' 19
33  178 '²' 19   179 '³' 19   180 '´' 16   181 'µ' 19   182 '¶' 20   183 '·' 8
34  184 '¸' 19   185 '¹' 19   186 'º' 19   187 '»' 14   188 '¼' 28   189 '½' 28
35  190 '¾' 28   191 '¿' 14   192 'À' 18   193 'Á' 18   194 'Â' 18   195 'Ã' 18
36  196 'Ä' 18   197 'Å' 18   198 'Æ' 26   199 'Ç' 15   200 'È' 16   201 'É' 16
37  202 'Ê' 16   203 'Ë' 16   204 'Ì' 9    205 'Í' 9    206 'Î' 9    207 'Ï' 9
38  208 'Ð' 19   209 'Ñ' 21   210 'Ò' 20   211 'Ó' 20   212 'Ô' 20   213 'Õ' 20
39  214 'Ö' 20   215 '×' 22   216 'Ø' 20   217 'Ù' 19   218 'Ú' 19   219 'Û' 19
40  220 'Ü' 19   221 'Ý' 17   222 'Þ' 19   223 'ß' 18   224 'à' 15   225 'á' 15
41  226 'â' 15   227 'ã' 15   228 'ä' 15   229 'å' 15   230 'æ' 24   231 'ç' 13
42  232 'è' 16   233 'é' 16   234 'ê' 16   235 'ë' 16   236 'ì' 8    237 'í' 8
43  238 'î' 8    239 'ï' 8    240 'ð' 19   241 'ñ' 17   242 'ò' 16   243 'ó' 16
44  244 'ô' 16   245 'õ' 16   246 'ö' 16   247 '÷' 22   248 'ø' 16   249 'ù' 17
45  250 'ú' 17   251 'û' 17   252 'ü' 17   253 'ý' 14   254 'þ' 19   255 'ÿ' 14
46
47

```

```

1  -----
2  Generic Name = monospace
3  Type        = Monospaced
4  HTML Size   = 1, 2
5  Size        = 22 pixels
6  Ascent      = 16 pixels
7  Descent     = 6 pixels
8  Text Lines  = 22.8
9  Line Gap    = -1
10 Advance     = 9 pixels (for all glyphs)
11
12
13  -----
14 Generic Name = monospace
15 Type        = Monospaced
16 HTML Size   = 3
17 Size        = 24 pixels
18 Ascent      = 18 pixels
19 Descent     = 6 pixels
20 Text Lines  = 20.9
21 Line Gap    = -1
22 Advance     = 10 pixels (for all glyphs)
23
24
25  -----
26 Generic Name = monospace
27 Type        = Monospaced
28 HTML Size   = 4
29 Size        = 27 pixels
30 Ascent      = 20 pixels
31 Descent     = 7 pixels
32 Text Lines  = 18.5
33 Line Gap    = -1
34 Advance     = 11 pixels (for all glyphs)
35
36
37  -----
38 Generic Name = monospace
39 Type        = Monospaced
40 HTML Size   = 5
41 Size        = 31 pixels
42 Ascent      = 23 pixels
43 Descent     = 8 pixels
44 Text Lines  = 16.0
45 Line Gap    = -1
46 Advance     = 13 pixels (for all glyphs)
47
48
49  -----
50 Generic Name = monospace
51 Type        = Monospaced
52 HTML Size   = 6
53 Size        = 36 pixels
54 Ascent      = 27 pixels
55 Descent     = 9 pixels
56 Text Lines  = 13.7
57 Line Gap    = -1
58 Advance     = 15 pixels (for all glyphs)
59
60
61
62
63
64

```

EACEM Technical Report

TR-048-r01

Title:

TeleWeb Application Part 3, Delivery Methods

Proposed ETSI Title:

“TeleWeb Application Part 3, Delivery Methods”

Proposed ETSI keywords:

“TeleWeb, Delivery, Teletext, VBI, Superteletext, Data Carousel, Transport”

Date: 23 April 2001

Document History

Version	Date	Author	Description
0.1	July 2000	David Tarrant	TeleWeb specification split into 3 parts. This document becomes part III.
0.2	August 2000	Vincent Tauzia	Updated following Philips Bruges meeting (10/8/00)
0.3	August 2000	Vincent Tauzia	Updated following ST Rousset meeting (31/08/00) and comments from D. Tarrant
1.0	Nov. 2000	Vincent Tauzia	Updated after D. Tarrant comments and inputs from the TeleWeb technical group meeting held in Munich.
2.0	5 Feb 2001	Jo Vandale	Adapting to the new EACEM references
2.1 / r00	15 Feb 2001	Jo Vandale	Adding the new EACEM template (Cover, Header, Footer, ...)
3.0 / r01	23 April 2001	Jo Vandale	Adaptations resulting from the last Technical Meeting. EACEM Project Team 1.4 Approved version.

Contents

1	Scope	122
2	References.....	122
2.1	TeleWeb.....	122
2.2	Character encoding	122
2.3	Analogue video broadcasting.....	122
2.4	Digital video broadcasting	122
2.5	Miscellaneous	123
3	Definitions and abbreviations	123
3.1	Definitions	123
3.2	Abbreviations.....	123
4	Delivery Profiles.....	124
4.1	TeleWeb delivered via Teletext packets	124
4.2	TeleWeb delivered via other methods	125
5	Transport layer protocols	125
5.1	DSM-CC data carousel	125
5.1.1	Overview.....	125
5.1.2	DSM-CC messages	127
5.1.2.1	DownloadInfoIndication message.....	127
5.1.2.2	DownloadServerInitiate message.....	130
5.1.2.3	DownloadDataBlock message.....	133
5.1.3	Handling updates	134
5.1.4	Descriptors	134
5.1.4.1	Type descriptor	135
5.1.4.2	Name descriptor	136
5.1.4.3	Info descriptor.....	136
5.1.4.4	CRC32 descriptor	136
5.1.4.5	Compressed module descriptor	137
5.1.4.6	Priority descriptor	137

5.1.4.7	Feature flags descriptor	138
5.1.4.8	Encryption/Conditional Access descriptor	138
5.1.4.9	Parental rating descriptor	139
5.1.4.10	Themes descriptor	139
5.1.4.11	Language descriptor	139
5.1.4.12	Character set descriptor	140
5.1.4.13	Creation Time descriptor	140
5.1.4.14	Start Validity descriptor	140
5.1.4.15	Expire Time descriptor	141
5.1.4.16	Repetition Distance descriptor	141
5.1.4.17	User Group ID descriptor	141
5.1.4.18	Profile Descriptor	142
5.1.4.19	Transmission Schedule descriptor	142
6	TeleWeb DSM-CC transmission via Teletext	143
6.1	Transmission of Data Carousels	143
6.2	Transmission protocol	144
6.2.1	CNI Code	144
6.2.2	Message checksum	144
6.2.3	Delimiting between messages	144
6.2.4	Byte order	145
6.3	Teletext packet format	145
6.3.1	Data Channel and Designation Code	145
6.3.2	Application Identifier	145
6.3.3	Format Type	145
6.3.4	Continuity Index	145
6.3.5	User Data	145
6.3.6	Forward Error Correction	146

Intellectual Property Rights

EACEM has not been informed of the existence of any Intellectual Property Rights (IPR) that could be, or could become essential to the present document. However, no investigation, including IPR searches, has been carried out. No guarantee can be given as to the existence of any IPRs that are, or may be, or may become, essential to the present document.

31 Scope

This document specifies methods for broadcasting TeleWeb services. A TeleWeb service comprises multimedia data files.

The present version covers transmission via Teletext data packets.

They are used to deliver a service to suitably equipped TV receivers and multimedia PCs with video capture cards. Such packets could be part of an analogue television signal, using either a few lines in the vertical blanking interval or all lines of the field, or part of an MPEG data stream. Prior to transmission, the files are packaged for delivery within a DSM-CC style data carousel.

32 References

32.1 TeleWeb

[47] EACEM TR-046: "TeleWeb Application Part 1 General Description".

[48] EACEM TR-047: "TeleWeb Application Part 2 Profile 1 Enhanced".

32.2 Character encoding

[49] IETF RFC 2152 (1994): "UTF-7: A Mail-Safe Transformation Format of Unicode".

[50] ISO 8859: "Information processing - 8-bit single-byte coded graphic character sets, Latin alphabets".

32.3 Analogue video broadcasting

[51] ETSI, ETS 300 706: "Enhanced Teletext Specification".

[52] ETSI, ETS 300 708: "Television Systems; Data Transmission within Teletext".

32.4 Digital video broadcasting

[53] ISO/IEC 13818-6: "Information technology - Generic coding of moving pictures and associated audio information - Part 6: Extension for Digital Storage Media Command and Control (DSM-CC) - International Standard (IS)".

[54] ETSI, ETS 300 472: "Digital Video Broadcasting (DVB); Specification for conveying ITU-R System B Teletext in DVB bit streams".

[55] ETSI, EN 301 192: "Digital Video Broadcasting (DVB); DVB specification for data broadcasting", V1.2.1 (1999-01).

[56] ETSI, TR 101 202: "Implementation guidelines for data broadcasting", V1.1.1 (1999-02).

1 [57] ISO/IEC 13818-1: "Information technology - Generic coding of moving pictures and associated audio
2 information - Part 1: Systems - International Standard (IS)".

3 32.5 Miscellaneous

4 [58] ISO 639-2: "Codes for the representation of names and languages - Part 2: Alpha-3 code3".

5 [59] IETF RFC 1950 (1996): "ZLIB Compressed Data Format Specification version 3.3".

6 [60] IETF RFC 2045 (1996): "Multipurpose Internet Mail Extension (MIME) Part One: Format of Internet
7 Message Bodies".

8 [61] IETF RFC 2046 (1996): "Multipurpose Internet Mail Extension (MIME) Part Two: Media Types".

9 [62] EACEM TR-054: "Register of Application Codes for Teletext based systems".

10 33 Definitions and abbreviations

11 33.1 Definitions

12 For the purposes of the present document, the following terms and definitions in singular or plural form do apply:

13 **bit ordering:** In all schematics the numeric values are ordered with the most significant bit at the left hand side and the least
14 significant bit at the right hand side.

15 **Conditional Access (CA):** A mechanism by which user access to service components can be restricted.

16 **Independent Data Line (IDL):** A standalone teletext packet containing both control and application data. It does not form
17 part of a teletext page. The packet address is either 30 or 31.

18 **Module:** When broadcast within a DSM-CC data carousel, the contents of a file and its attributes (e.g. file type, creation date,
19 etc) are transmitted separately. The file itself is carried by a number of DDB messages and its attributes appear as descriptors
20 within its module loop within a DII control message. For the purposes of this specification, a module is defined to include
21 both the DDBs message and the associated descriptors in the DII message.

22 **Signed integer:** A positive or negative integer value, in decimal notation. The first digit is preceded by a mandatory plus (+)
23 or minus (-) symbol. There shall be no white space between the symbol and the first digit.

24 **Text string:** A sequence of displayable Latin-1 characters.

25 **Unsigned integer:** An integer value, in decimal notation, without a preceding plus (+) or minus (-) symbol.

26 33.2 Abbreviations

27 For the purpose of the present document, the following abbreviations apply:

28	ASCII	American Standard Code for Information Interchange
29	CA	Conditional Access
30	CLUT	Colour Look Up Table
31	CRC	Cyclic Redundancy Check
32	DAB	Digital Audio Broadcasting
33	DDB	DownloadDataBlock message
34	DII	DownloadInfoIndication message
35	DOS	Disk Operating System
36	DSI	DownloadServerInitiate message
37	DSM-CC	Digital Storage Media Command and Control
38	DVB	Digital Video Broadcasting

1	ETS	European Telecommunication Standard
2	FEC	Forward Error Correction
3	HTML	Hyper Text Mark-up Language
4	JPEG	Joint Picture Experts Group
5	IDL	Independent Data Line
6	LSB	Least Significant Bit
7	MJD	Modified Julian Date
8	MSB	Most Significant Bit
9	PNG	Portable Network Graphics
10	RFC	Internet Requests for Comments
11	URL	Uniform Resource Locator
12	UTC	Universal Time Co-ordinated
13	VBI	Vertical Blanking Interval
14	WWW	World Wide Web

15 **34 Delivery Profiles**

16 There are several ways in which a TeleWeb application can be delivered, see figure 1 in [47]. For the purposes of this
17 document, a Delivery Profile specifies layers 1 to 4 of the OSI seven-layer model.

18 **34.1 TeleWeb delivered via Teletext packets**

19 Figure 11 shows layers 1 to 4 of the OSI seven-layer model for delivering a TeleWeb service via Teletext packets. The
20 application files are formed into a DSM-CC data carousel at the transport layer, as defined in section
21 35.1. The components of the carousel are then encapsulated in independent Teletext data packets, see
22 section 36.3. These can be transmitted as part of an analogue TV signal [52] or within an MPEG-2 data
23 stream [6] JPEG File Interchange Format Version 1.02 September 1, 1992

24 [7] Digital Compression and Coding of Continuous Still Images
25 Part 1, Requirements and Guidelines
26 ISO/IEC JTC1 Draft International Standard 10918-1, Nov 1991

27 [8] Digital Compression and Coding of Continuous Still Images
28 Part 2, Compliance Testing
29 ISO/IEC JTC1 Draft International Standard 10918-2, Dec 1991

1 [9].

Layer
Generic content
TeleWeb specific

Layer Transport	4:	Arranging the data in a suitable way for transport	DSM-CC data carousel: Blocks and modules Descriptors Groups and supergroups Delimiting between messages Forward error correction
Layer Network	3:	Logical functions related to the multiplexing and demultiplexing of data packets belonging to different communications flows: Data channel addressing Data packet sequencing	Format B independent data line as defined in [52] ("Packet 30/31")
Layer Link	2:	Logical functions related to the data transmission: Byte synchronisation Error control (framing, mis-direction and false detection) Data formatting	Normal teletext packet format as defined in [51]
Layer Physical	1:	Electrical transmission of the data signal	Normal teletext parameters as defined in [51]

2 **Figure 11: Delivery method for TeleWeb using Teletext packets**

3 **34.2 TeleWeb delivered via other methods**

4 This document will be amended when necessary with other transport methods.

5 **35 Transport layer protocols**

6 This section defines protocols for implementing the transport layer.

7 **35.1 DSM-CC data carousel**

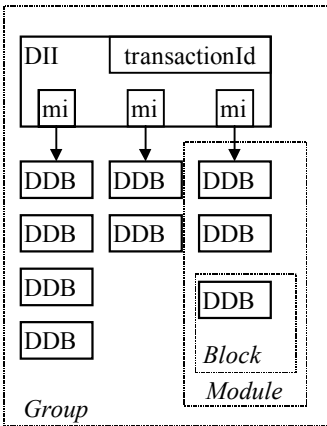
8 **35.1.1 Overview**

9 The files of a TeleWeb service are organised in DSM-CC data carousels according to the general principles defined in [53]
10 and adapted for DVB applications as described in [10] and [11]. The DSM-CC data carousel specification embodies the
11 cyclic transmission of data to receivers. The data transmitted within a carousel is first organised into "modules", which are
12 then sub-divided into "blocks". All the blocks of all modules within the data carousel are of the same size, except for the last
13 block of each module which may be of a smaller size. Each individual file in a TeleWeb service is treated as a module.
14 Modules can be clustered together to form a "group". Likewise, groups can be clustered to form "supergroups".

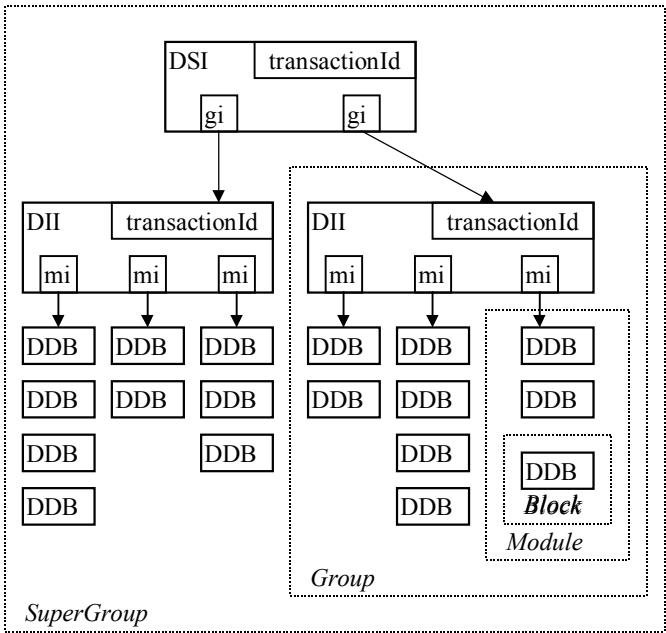
15 The data carousel specification defined here uses three messages from the full data carousel specification defined in [53]. The
16 data blocks are carried in DownloadDataBlock (DDB) messages, while control over the modules is provided by
17 DownloadInfoIndication (DII) and DownloadServerInitiate (DSI) messages. Other DSM-CC messages listed in [53] are not
18 used in the TeleWeb application and should be ignored by receivers designed to this edition. All messages begin with the

- 1 generic DSM-CC Message Header.¹ This header contains information about the type and size of the message. The maximum
- 2 size of any DII, DSI or DDB message shall be 4 084 bytes (including the message header).
- 3 A receiver shall ignore message fields marked “reserved” (ISO designated) or “reserved for future use” (ETSI designated),
- 4 noting that such fields may be variable in size.
- 5 A data carousel can have one or two layers of control information as shown in Figure 12. The service provider is free to
- 6 choose the most appropriate type. A receiver shall be able to work with both types.

One-layer data carousel



Two-layer data carousel



DSI: DownloadServerInitiate message
DII: DownloadInfoIndication message
DDB: DownloadDataBlock message
gi: Group Information bytes
mi: Module Information bytes

Figure 12: Structure of one-layer and two-layer data carousels

- 9 A one-layer carousel defines a single group. The top-level control message is a DII message. This describes all the modules in
- 10 the carousel. The module description includes a descriptor loop that carries the attributes (e.g. type, theme) of each module.
- 11 A two-layer carousel comprises a number of one-layer carousels. The top-level control message is a DownloadServerInitiate
- 12 message (DSI), section 35.1.2.2. This describes the different groups in the supergroup. A descriptor loop allows attributes to
- 13 be assigned to each group individually. Any module-specific attributes defined at the group level are inherited by all modules
- 14 within that group. Each group is described by a DII message as for a one-layer carousel.
- 15 A TeleWeb service may be implemented across a number of carousels. The need for multiple carousels arises because of the
- 16 significant overhead associated with updating the contents of control messages when transmitting real-time information. This
- 17 can be minimised by limiting the amount of data in a carousel that contains information that changes very frequently. The
- 18 maximum number of carousels may be limited by the transmission method.

¹ Ref. [53], section 2.

35.1.2 DSM-CC messages

35.1.2.1 DownloadInfoIndication message

A DownloadInfoIndication message contains the description of the modules within a group as well as some general parameters of the data carousel such as block size. Each module within the group can be described by a number of descriptors. These carry the TeleWeb-specific attributes.

The syntax of a DII message is shown in Table 1. The final column indicates if a field has a fixed or variable value in the TeleWeb application.

Table 1: Syntax of the DownloadInfoIndication message

Syntax	No. of Bytes	TeleWeb Use
DownloadInfoIndication() {		
protocolDiscriminator	1	Fixed at 0x11
dsmccType	1	Fixed at 0x03
messageId	2	Fixed at 0x1002
transactionId	4	Variable
reserved	1	Fixed at 0xFF
adaptationLength	1	Variable
messageLength	2	Variable
dsmccAdaptationHeader()	adaptationLength	Reserved for future use
downloadId	4	Variable
blockSize	2	Variable
windowSize	1	Fixed at 0x00
ackPeriod	1	Fixed at 0x00
tCDownloadWindow	4	Fixed at 0x00 throughout
tCDownloadScenario	4	Fixed at 0x00 throughout
compatibilityDescriptorLength	2	Variable
compatibilityDescriptor()	compatibilityDescriptorLength	Reserved for future use
numberOfModules	2	Variable
for(l=0; l< numberOfModules; l++) {		
moduleId	2	Variable
moduleSize	4	Variable
moduleVersion	1	Variable
moduleInfoLength	1	Variable
for(l=0; l< moduleInfoLength;		
l++) {		
moduleInfoByte	1	Variable
}		
}		
privateDataLength	2	Variable
for(l=0; l< privateDataLength; l++) {		
privateDataByte	1	reserved for future use
}		
}		

The **protocolDiscriminator** field is used to indicate that the message is a DSM-CC message within a particular environment. This field has the fixed value of 0x11.

1 The **dsmtccType** field is used to indicate the type of DSM-CC message.² The “User-to-Network” download messages of
2 which data carousels are a part have been allocated the value 0x03.

3 The **messageId** field is fixed at 0x1002 to identify the message as a DownloadInfoIndication message.³

4 The **transactionId** field provides both a unique identification of a control message and version information. The version
5 component is changed whenever any field of the message is modified. Reference [53] defines the transactionId as consisting
6 of a 2-bit transactionId_originator field (2 MSBs) and a 30-bit transaction_number field.⁴ The TeleWeb application adopts
7 the DVB interpretation [11] and divides the field into the four sub-fields shown in Figure 13.

8 In the case of a two-layer carousel, each DII message is referenced from within the group loop of the DSI message. The
9 transactionId field in a DII message and the corresponding groupId field in the DSI message are coded identically.

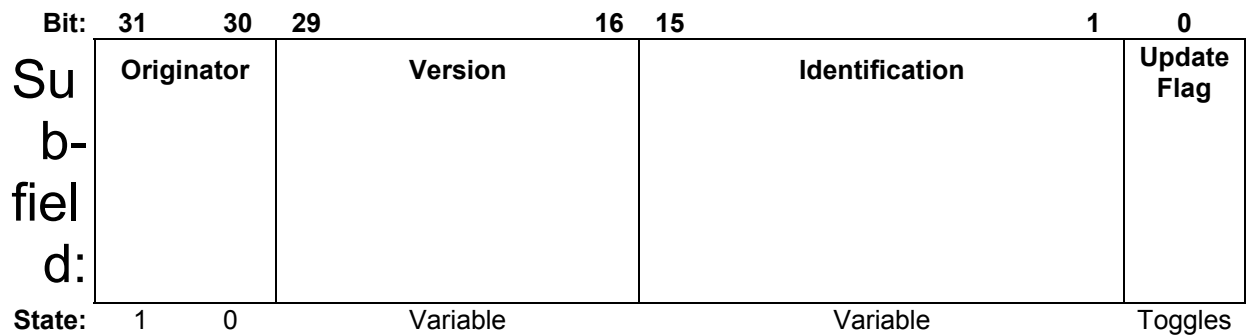


Figure 13: Format of transactionId field

11 The **Update Flag** (bit 0) shall be toggled every time the message is updated.

12 The **Identification** sub-field (bits 1 to 15) shall be set to 0x0000 for the top-level control message in the data carousel. All
13 other control messages shall have one or more non-zero bits, [10] This coding scheme enables a receiver to determine
14 whether a carousel is one-or two-layered having acquired any DII message:

15 For the one-layer carousel:
16 - The Identification field in the DII message will be 0x0000.

17 For the two-layer carousel:
18 - The Identification field in a DII message will be in the range 0x0001 to 0x7FFF.
19 - The Identification field in the DSI message will be 0x0000.

20

21 NOTE: If multiple carousels are used, a receiver is likely to detect DII messages with Identification values from 0x0000 to
22 0x7FFF.

23 The **Version** sub-field (bits 16 to 29) conveys a version number for the message. This value shall be incremented modulo
24 0x4000 every time the control message is updated.

25 NOTE: A receiver should not expect to see linearly incrementing version numbers. There may be missing values. For
26 example, the encoder might have prepared a new DII message (and incremented the version number) following a change, but
27 before it can be transmitted another change is detected, causing the version number to be incremented yet again before
28 transmission. The updating actions required on a change of data are described in section 35.1.3.

² Ref. [53], section 2, table 2-2.

³ Ref. [53], section 7.3, table 7-4.

⁴ Ref. [53], section 2, figure 2-1.

1 In order to be strictly DSM-CC compliant, the value used in **Originator** sub-field (bits 31 to 32) shall be fixed at 0x02 to
2 indicate that the transactionId has been assigned by the network.⁵

3 The **reserved** field shall be set to 0xFF as required by [53].⁶

4 The **adaptationLength** field indicates the total length in bytes of the DSM-CC Adaptation Header
5 (**dsmccAdaptationHeader**).⁷

6 NOTE: The Adaptation Header is not used to carry TeleWeb-specific information. Receiver designers should note that data
7 carousels may use the Adaptation Header for other purposes. Receivers shall be able to accept an adaptationLength of any
8 value and shall ignore any data in the dsmccAdaptationHeader field.

9 The **messageLength** field is used to indicate the total length in bytes of the remainder of the message following this field.
10 This value includes any Adaptation Headers and Compatibility Descriptors that may be present.

11 The **downloadId** field is used as an identifier for the messages of a particular TeleWeb carousel. All DII and DDB messages
12 within the same carousel shall use the same value in their downloadId fields.

13 The **blockSize** field defines the number of blockDataBytes carried in the DDB messages described by this DII.

14 NOTE: The last block of each module may contain fewer blockDataBytes than the value indicated by blockSize.

15 The **windowSize**, **ackPeriod**, **tCDownloadWindow** fields are not used by data carousels and all bytes shall be fixed at 0x00.

16 The **tCDownloadScenario** field indicates a time-period (in microseconds) for the entire download scenario. If the value is
17 unknown or is considered to be unimportant, a value of 0xFFFFFFFF shall be inserted in this field.

18 The **compatibilityDescriptorLength** field indicates the total length in bytes of the following DSM-CC Compatibility
19 Descriptor.⁸

20 NOTE 1: According to [53], the compatibilityDescriptorLength is the first field in the DSM-CC compatibilityDescriptor. It is
21 shown here as a separate field to aid understanding.

22 NOTE 2: The compatibilityDescriptor is not used to carry TeleWeb-specific information and the
23 compatibilityDescriptorLength field shall be set to 0x0000. Receiver designers should note that future transmissions may use
24 the compatibilityDescriptor. Receivers shall be able to accept a compatibilityDescriptorLength of any value and shall ignore
25 any data in the compatibilityDescriptor field.

26 The **numberOfModules** field indicates the number of modules described in the loop following the field. In a one-layer
27 carousel, this loop will describe all the modules associated with the carousel.

28 The **moduleId** field is an identifier for the module that is described by the following moduleSize, moduleVersion, and
29 moduleInfoByte fields. The value must be unique within the scope of the message sharing the same downloadId value,
30 i.e. within the same data carousel. Values in the range 0x0000 to 0xFFEF shall be used.

31 NOTE: ModuleId values in the range 0xFFFF0 to 0xFFFF are reserved for DAVIC compliant applications⁹ and shall not be
32 used in TeleWeb applications.

⁵ Ref. [53], section 2, table 2-3.

⁶ Ref. [53], section 2.

⁷ Ref. [53], section 2.1.

⁸Ref. [53], section 6.1.

⁹ Ref. [10], section 8.1.3.

- 1 The **moduleSize** field defines the number of blockDataBytes that make up the described module. This equates to the size of
2 the TeleWeb file carried by this module.
- 3 The **moduleVersion** field provides a version number for the described module. The value inserted here shall match the
4 current value of the moduleVersion field carried in the DDB messages of the described module. The moduleVersion field is
5 used for the version attribute defined in [48].
- 6 The **moduleInfoLength** field defines the number of moduleInfoBytes that follow.

7 **The moduleInfoBytes describe the module. These bytes**
8 **shall contain the descriptors defined in section 35.1.4.**

9 **The privateDataLength field defines the length in bytes of**
10 **the following privateDataByte field. The function of the**
11 **data in the privateDataByte field is specific to the**
12 **TeleWeb application and is reserved for future use.**

13 **35.1.2.2 DownloadServerInitiate message**

- 14 The DownloadServerInitiate message is used to build a supergroup within a two-layer data carousel.
- 15 The syntax of a DSI message is shown in Table 2. The final column indicates if a field has a fixed or variable value in the
16 TeleWeb application. The private data fields of the generic DSI message [53] are used here to carry information relating to
17 each group according to the interpretation in [10] and [11]. To allow future enhancements, provision is made to carry
18 additional private data.
- 19 Data fields not described in this section have the same function and coding as their equivalents in the DII message, section
20 35.1.2.1.

Table 2: Syntax of the DownloadServerInitiate message

Syntax	No. of Bytes	TeleWeb Use
DownloadServerInitiate() {		
protocolDiscriminator	1	Fixed at 0x11
dsmccType	1	Fixed at 0x03
messageId	2	Fixed at 0x1006
transactionId	4	Variable
reserved	1	Fixed at 0xFF
adaptationLength	1	Variable
messageLength	2	Variable
dsmccAdaptationHeader()	adaptationLength	Reserved for future use
serverId	20	Fixed at 0xFF throughout
compatibilityDescriptorLength	2	Variable
compatibilityDescriptor()	compatibilityDescriptorLength	
privateDataLength	2	Variable
GroupInfoIndication() {		
NumberOfGroups	2	Variable
for(l=0; l< NumberOfGroups;		
l++) {		
GroupId	4	Variable
GroupSize	4	Variable
GroupCompatibilityDescriptorLength	2	Variable
GroupCompatibility()	GroupCompatibilityDescriptorLength	Reserved for future use
GroupInfoLength	2	Variable
for(j=0; j<		
GroupInfoLength; j++) {		
groupInfoByte	1	Variable
}		
}		
futureUseLength	2	Variable
ServiceInfoLength	2	Variable
for(k=0; k<ServiceInfoLength; k++) {		
ServiceInfoByte	1	Variable
}		
for(k=0; k<futureUseLength-		
ServiceInfoLength;		
k++) {		
futureUseByte	1	Variable
}		
}		

2 NOTE: The fields with a shaded background replace the privateDataBytes of the generic DSM-CC specification for a DSI
3 message.

4 The **messageId** field is fixed at 0x1006 to identify the message as a DSI message.¹⁰

¹⁰ Ref. [53], section 7.3, table 7-4.

1 The **transactionId** field has the same four sub-fields as its equivalent in the DII message. By definition, the **Identification**
2 sub-field shall always be set to 0x0000.

3 The **serverId** field is unused and shall be set to 0xFF throughout.¹¹

4 The **privateDataLength** field defines the length in bytes of the remaining part of the message. This consists of group-related
5 data (GroupInfoIndication structure) and bytes reserved for future use (futureUseLength and futureUseByte).

6 The **GroupInfoIndication** structure follows the DVB assignment of the private data bytes of the generic DSM-CC DSI
7 message, [10] and [11]. The bytes are used to convey information about the structure of each group.

8 The **NumberOfGroups** field indicates the number of groups described in the following loop.

9 The **GroupId** field enables a particular group to be identified. The field shall contain the same value as is used in the
10 transactionId field of the DII control message that describes the group. This value will change when an update is made to the
11 DII control message, as described in section 35.1.3. All GroupIds should be unique within the service.

12 The **GroupSize** field indicates the total number of blockDataBytes that make up the modules in the group.

13 The **GroupCompatibilityDescriptorLength** field indicates the total length in bytes of the following Group Compatibility
14 structure.

15 The **GroupCompatibility** structure has the same format as the DSM-CC compatibilityDescriptor structure.¹²

16 NOTE 1: According to [53], the compatibilityDescriptorLength is the first field in the DSM-CC compatibilityDescriptor. The
17 GroupCompatibilityDescriptorLength is shown here as a separate field outside of the GroupCompatibility structure to aid
18 understanding.

19 NOTE 2: The Group Compatibility structure is not used in this edition to carry TeleWeb-specific information. Transmissions
20 designed to this edition shall set the GroupCompatibilityDescriptorLength field to 0x0000. To ensure future compatibility,
21 receivers designed to this edition shall be able to accept a GroupCompatibilityDescriptorLength of any value and shall ignore
22 any data in the GroupCompatibility structure.

23 The **GroupInfoLength** field defines the number of GroupInfoBytes that follow.

24 The **GroupInfoBytes** convey a list of descriptors defining attributes and characteristics of a group. The coding is given in
25 section 35.1.4.

26 The **futureUseLength** field defines the length in bytes of the following service info and the bytes reserved for future use.

27 The **futureUseBytes** are reserved for future enhancements.

28 The **ServiceInfoLength** field defines the number of ServiceInfoBytes that follow.

29 The **ServiceInfoByte** convey a list of descriptors defining attributes and characteristics of the service. The coding is given in
30 section 35.1.4.

31

¹¹ Ref. [10], section 8.1.2.

¹² Ref. [53], section 6.1.

35.1.2.3 DownloadDataBlock message

In a data carousel, the DownloadDataBlock messages contain the blocks of the fragmented modules (i.e. the files of the TeleWeb service). A DDB message contains a single data block of a module. The syntax of the message is shown in Table 3. The final column indicates if a field has a fixed or variable value in the TeleWeb application.

Data fields not described in this section have the same function and coding as their equivalents in the DII message, section 35.1.2.1. The first part of the message, the header, is similar to that of DII and DSI messages. The main difference concerns the transactionId field which here becomes the downloadId field.

Table 3: Syntax of the DownloadDataBlock message

Syntax	No. of Bytes	TeleWeb Use
DownloadDataMessage() {		
protocolDiscriminator	1	Fixed at 0x11
dsmccType	1	Fixed at 0x03
messageId	2	Fixed at 0x1003
downloadId	4	Variable
reserved_1	1	Fixed at 0xFF
adaptationLength	1	Variable
messageLength	2	Variable
dsmccAdaptationHeader()	adaptationLength	Reserved for future use
moduleId	2	Variable
moduleVersion	1	Variable
reserved_2	1	Fixed at 0xFF
blockNumber	2	Variable
for(l=0; l< blockSize;		
l++) {		
blockDataByte	1	Variable
}		
}		

The **messageId** field is fixed at 0x1003 to identify the message as a DownloadDataBlock message.¹³

The **downloadId** field identifies the particular carousel to which the block belongs. It maps to the downloadId field in the parent DII message.

The **reserved_1** and **reserved_2** fields shall be set to 0xFF as required by [53].¹⁴

The **moduleId** field identifies the unique module to which this block belongs.

The **moduleVersion** field identifies the version of the module to which this block belongs. When a module is updated, the current value shall be incremented modulo 0x100.

The **blockNumber** field identifies the position of the block within the module. Block number 0 shall be the first block of a module.

The **blockDataBytes** convey the data of the block. The blockSize variable is conveyed in the parent DII message. However, the last block of a module may be smaller in size. A receiver can calculate the blockNumber value and the size of the last block of a module from the moduleSize and blockSize information carried in the parent DII message.

¹³ Ref. [53], section 7.3, table 7-4.

¹⁴ Ref. [53], section 2.

1 **35.1.3 Handling updates**

2 The transactionId field located in both DII and DSI messages provides unique identification and versioning information. The
3 coding of the sub-fields is described in section 35.1.2.1.

4 As the transactionId functions as a versioning mechanism, ANY change to ANY file, attribute, module, moduleId or
5 descriptor in the data carousel shall cause the version sub-field of the transactionId in the top-level control message to be
6 incremented and the Update Flag sub-field to be toggled. The change propagates up through the structure of the carousel as
7 follows:

8 1. Any change to the contents of a file, but not the attributes of that file¹⁵, shall cause the moduleVersion field to be
9 incremented in each DDB message required to transmit the module containing the file. This change must be reflected in
10 the moduleVersion field in the module loop of the DII message that includes this module.

11 2. A field in a DII message can change either as a result of an update to the contents of a module (in which case its
12 moduleVersion field will have altered according to 1) or because of a change in the descriptors of a module (leading to
13 modifications to its moduleInfoBytes). When a field in a DII message changes, the version sub-field of its transactionId
14 must be incremented to indicate a new version of the message, and the Update Flag must be toggled.

15 3. In the case of a two-layer carousel, the change to a DII message must be reflected in the corresponding groupId field in
16 the group loop of the DSI message.

17 4. In the case of a two-layer carousel, since a field in the DSI message has changed the version sub-field of its transactionId
18 must also be incremented, and the Update Flag must be toggled.

19 Thus by inspection of the transactionId in the top-level control message, a receiver can detect any changes to the carousel.

20 **35.1.4 Descriptors**

21 The file attributes presented in [48] appears in the data carousel as descriptors. Specific descriptors can be inserted for
22 individual modules (i.e. TeleWeb files). These are carried in the moduleInfoBytes of the DII message containing the module.
23 The version attribute defined in [48] is not realised with a descriptor. The moduleVersion in the DII message is taken as the
24 value for the version attribute.

25 Certain module-related descriptors can also be applied at the group level and affect all modules within that group. Group
26 specific descriptors are carried in the GroupInfoBytes of the DSI message. In general, it is not possible to “turn-off” the effect
27 of a DII-wide descriptor defined at the DSI level with a module-specific descriptor at the DII level.

28 Table 4 lists the descriptors, their tag values and where they may be applied within the carousel. The tag value identifies
29 individual descriptors. Certain descriptors (tag values 0x00 to 0x09) are identical to those defined for DVB data broadcasting
30 in [10]. The remainder are TeleWeb specific and are allocated tag values from 0x80 as required of private descriptors in [10].

¹⁵ It is allowed to change the descriptors without changing the module's moduleVersion value. The transactionId value is, of course, changed as a result. When a revised DII arrives, the receiver can detect whether the changes affect modules or descriptors by comparing the current moduleVersion values with those received last time. If a particular moduleVersion value has not altered, all the descriptors for that module have to be processed in order to establish the changes. If the value has altered, both the contents of the module and its descriptors have to be processed.

1

Table 4: Descriptors, tag values and allowed locations

Descriptor	Tag value	Module Level	Group Level	Service Level
Type	0x01	✓	✓	
Name	0x02	✓		✓
Info	0x03	✓	✓	✓
CRC32	0x05	✓		
Compressed module	0x09	✓		
Priority	0x80	✓	✓	
Feature flags	0x81	✓	✓	
Encryption/Conditional Access	0x82	✓	✓	
Parental rating	0x83	✓	✓	
Theme	0x84	✓	✓	
Language	0x85	✓	✓	✓
Character set	0x86	✓	✓	
Creation Time	0x87	✓	✓	
Start Validity	0x88	✓	✓	
Expire Time	0x89	✓	✓	
Repetition Distance	0x8A	✓	✓	
User ID	0x8B	✓	✓	
Profile	0x8C	✓	✓	
Transmission Schedule	0x8D			✓

2 There shall be a maximum of one descriptor of each applicable type for each module or group.

3 To ensure future compatibility, receivers designed to this specification should ignore entire descriptors and additional bytes
4 appearing in current descriptors whose functions are not defined by this edition.

5 The info, priority, parental rating, theme, language, creation time, start validity, expire time, repetition distance and the User
6 ID descriptor may be updated at any time without changing the version of the module. The CRC32, compressed module,
7 feature flags, encryption/conditional access, and character set descriptor are only allowed to be updated/changed when the
8 version of the module also changes. The type, name and profile descriptor must not be changed at all even when the version
9 of the file is changed.

10 35.1.4.1 Type descriptor

11 The type_descriptor contains the Type attribute defined in [48] for the data in a module or a group.

12

Table 5: Syntax of type_descriptor

Syntax	No. of Bytes	Value
type_descriptor() {		
descriptor_tag	1	0x01
descriptor_length	1	Variable
for (l=0; l < descriptor_length; l++) {		
text_char	1	Text string, e.g. "text/html"
}		
}		

13 The **descriptor_tag** field is common to all descriptors. It identifies the descriptor according to Table 4.

14 The **descriptor_length** field is common to all descriptors. It specifies the number of bytes in the descriptor immediately
15 following this field.

16 The **text_char** field contains a text string specifying the type of module or group. It follows the Media Type specifications in
17 [41] and [42]. Supported types are defined in [48].

35.1.4.2 Name descriptor

The name_descriptor contains the Name attribute defined in [48] to be associated with the data in a module or service.

Table 6: Syntax of name_descriptor

Syntax	No. of Bytes	Value
name_descriptor() {		
descriptor_tag	1	0x02
descriptor_length	1	Variable
for (l=0; l< descriptor_length; l++) {		
text_char	1	Name of the module, e.g. "index.html"
}		
}		

The **text_char** field specifies the name of the module as a string of text characters. Text information is coded using the ISO Latin-1 character set.

35.1.4.3 Info descriptor

The info_descriptor contains the info attribute defined in [48] for a module, group or service.

Table 7: Syntax of info_descriptor

Syntax	No. of Bytes	Value
info_descriptor() {		
descriptor_tag	1	0x03
descriptor_length	1	Variable
ISO_639_language_code	3	Language code
for (l=0; l< descriptor_length- 3; l++) {		
text_char	1	Description of the module or group
}		
}		

The **ISO_639_language_code** field identifies the language of the following text field. The ISO_639_language_code contains a 3 character code as specified in [58]. Both ISO 639.2/B and ISO 639.2/T formats may be used. Each character is coded into 8-bits according to [50] and inserted in order into the 24-bit field. Only lower-case letters shall be used, e.g. for German the code 'deu' (ISO 639-2/T) or 'ger' (ISO 639-2/B) shall be used.

Text_char specifies a text description of the module or group as a string of text characters. Text information is coded using the ISO Latin-1 character set. This string represent the info attribute specified in [48].

35.1.4.4 CRC32 descriptor

The CRC32_descriptor carries the CRC attribute defined in [48] for a complete module.

Table 8: Syntax of CRC32_descriptor

Syntax	No. of Bytes	Value
CRC32_descriptor() {		
descriptor_tag	1	0x05
descriptor_length	1	0x04
CRC_32	4	CRC value
}		

The **CRC_32** field contains the CRC calculated over this module. This is calculated according to Annex B of [57].

35.1.4.5 Compressed module descriptor

The presence of the `compressed_module_descriptor` indicates that the data in the module has the “zlib” structure as defined in [15]. Table 9 shows the syntax of the `compressed_module_descriptor`.

Table 9: Syntax of compressed_module_descriptor

Syntax	No. of Bytes	Value
<code>compressed_module_descriptor () {</code>		
descriptor_tag	1	0x09
descriptor_length	1	0x05
compression_method	1	Variable
original_size	4	Variable
<code>}</code>		

The **compression_method** field identifies the compression method used. This identification follows the definition of zlib structure of RFC 1950 [15].

The **original_size** field indicates the size in bytes of the module prior to compression.

35.1.4.6 Priority descriptor

The `priority_descriptor` carries the Priority attribute described in [48].

Table 10: Syntax of priority_descriptor

Syntax	No. of Bytes	Value
<code>priority_descriptor() {</code>		
descriptor_tag	1	0x80
descriptor_length	1	0x01
priority	1	Variable
<code>}</code>		

The format of the **priority** byte is shown below:

B7	B6	B5	B4	B3	B2	B1	B0
Priority_7	Priority_6	Priority_5	Priority_4	Priority_3	Priority_2	Priority_1	Priority_0

Priority_7 to **Priority_0** shall be interpreted as an 8-bit unsigned integer (LSB = bit 0) representing the Priority attribute.

35.1.4.7 Feature flags descriptor

The feature_flags_descriptor carries the Copyright, Suppress User Interface, Home Page and Service Ident attributes described in [48].

Table 11: Syntax of feature_flags_descriptor

Syntax	No. of Bytes	Value
feature_flags_descriptor() {		
descriptor_tag	1	0x81
descriptor_leng	1	0x01
feature_flags	1	Variable
}		

The format of the **feature_flags** byte is shown below:

B7	B6	B5	B4	B3	B2	B1	B0
Reserved	Reserved	Reserved	Reserved	Service_Ident	Home_Page	Suppress_User_Interface	Copyright

Copyright is an active high flag representing the Copyright attribute specified in [48].

Suppress_User_Interface is an active high flag representing the Suppress_User_Interface attribute specified in [48].

Home_Page is an active high flag representing the Home Page attribute specified in [48].

Service_Ident is an active high flag representing the Service Ident attribute specified in [48].

The remaining bits are reserved for future use and should be ignored by a decoder designed to this edition.

Note: If this descriptor is used at the group level, only the Copyright and Suppress_User_Interface flags have any significance. The Home_Page flag and the Service_Ident flag should be set to 0. If there is a conflict between a flag defined at the group level and a flag defined at the module level, the module level setting shall have priority.

35.1.4.8 Encryption/Conditional Access descriptor

The encryption_ca_descriptor carries the encryption/conditional access attribute described in [48].

Table 12: Syntax of encryption_ca_descriptor

Syntax	No. of Bytes	Value
encryption_ca_descriptor () {		
descriptor_tag	1	0x82
descriptor_length	1	Variable
for (l=0; l< descriptor_length ; l++) {		
descriptor_byte	1	Variable
}		
}		

Note: The payload of the encryption descriptor is not defined. However, Profile 1 decoders must ignore pages with this attribute/descriptor. Therefore, this container is defined in this specification so that profile 1 decoders can react on the presence of such a descriptor in future.

35.1.4.9 Parental rating descriptor

The rating_descriptor allows a parental rating to be associated with a file.

Table 13: Syntax of rating_descriptor

Syntax	No. of Bytes	Value
rating_descriptor() {		
descriptor_tag	1	0x83
descriptor_length	1	0x01
rating_value	1	Variable
}		
}		

The **rating_value** field identifies a parental rating as defined in [48].

35.1.4.10 Themes descriptor

The themes_descriptor carries the Theme attributes defined in [48] to be associated with a module or a group.

Table 14: Syntax of themes_descriptor

Syntax	No. of Bytes	Value
themes_descriptor () {		
descriptor_tag	1	0x84
descriptor_length	1	Variable, even
for (j=0; j< descriptor_length / 2; j++) {		
theme	2	Theme identifier
}		
}		

The **theme** field identifies a theme as defined in [48].

35.1.4.11 Language descriptor

The language_descriptor carries the Language attribute defined in [48] for a module, group or service.

Table 15: Syntax of language_descriptor

Syntax	No. of Bytes	Value
language_descriptor() {		
descriptor_tag	1	0x85
descriptor_length	1	0x03
ISO_639_language_code	3	Language code
}		

The **ISO_639_language_code** – as defined in 35.1.4.3.

35.1.4.12 Character set descriptor

Table 16: Syntax of character_set_descriptor

Syntax	No. of Bytes	Value
character_set_descriptor () {		
descriptor_tag	1	0x86
descriptor_length	1	Variable
for (l=0; l<descriptor_length; l++) {		
text_char	1	Text string, e.g. 'iso-8859-1'
}		
}		

The **text_char** field specifies the character set (or character encoding) used for the module as a string of text characters. Text information is coded using the ISO Latin-1 character set.

35.1.4.13 Creation Time descriptor

The creation_time_descriptor carries the Creation Time attribute for a module or a group as specified in [48].

Table 17: Syntax of creation_time_descriptor

Syntax	No. of Bytes	Value
creation_time_descriptor () {		
descriptor_tag	1	0x87
descriptor_length	1	0x05
MJD_offset	2	Variable
UTC_hours	1	Variable
UTC_minutes	1	Variable
UTC_seconds	1	Variable
}		

The **MJD_offset** field specifies the offset (in days) from the reference date of 14 June 1993 (MJD = 0xC000). Thus the actual date according to the Modified Julian coding strategy is given by 0xC000 + MJD_offset.

NOTE: The absolute MJD value increments daily at 00:00 UTC (Co-ordinated Universal Time). The latest date supported by this coding scheme is 17 November 2172.

The **UTC_hours** field specifies the hours component of a “time” referenced to UTC. The valid range is 0 to 23 (decimal) inclusive.

The **UTC_minutes** field specifies the minutes component of a “time” referenced to UTC. The valid range is 0 to 59 (decimal) inclusive.

The **UTC_seconds** field specifies the seconds component of a “time” referenced to UTC. The valid range is 0 to 59 (decimal) inclusive.

35.1.4.14 Start Validity descriptor

The start_validity_descriptor carries the Start Validity attribute for a module or a group as specified in [48].

Table 18: Syntax of start_validity_descriptor

Syntax	No. of Bytes	Value
start_validity_descriptor () {		
descriptor_tag	1	0x88
descriptor_length	1	0x05
MJD_offset	2	Variable
UTC_hours	1	Variable
UTC_minutes	1	Variable
UTC_seconds	1	Variable
}		

Apart from the descriptor tag, the coding is identical to the creation_time_descriptor defined in 35.1.4.13.

35.1.4.15 Expire Time descriptor

The expire_time_descriptor carries the Expire Time attribute for a module or a group as specified in [48].

Table 19: Syntax of expire_time_descriptor

Syntax	No. of Bytes	Value
expire_time_descriptor () {		
descriptor_tag	1	0x89
descriptor_length	1	0x05
MJD_offset	2	Variable
UTC_hours	1	Variable
UTC_minutes	1	Variable
UTC_seconds	1	Variable
}		

Apart from the descriptor tag, the coding is identical to the creation_time_descriptor defined in 35.1.4.13.

35.1.4.16 Repetition Distance descriptor

The repetition_distance_descriptor contains the Repetition Distance for a module or a group as specified in [48].

Table 20: Syntax of repetition_distance_descriptor

Syntax	No. of Bytes	Value
repetition_distance_descriptor () {		
descriptor_tag	1	0x8A
descriptor_length	1	0x02
repetition_distance	2	Variable
}		

The **repetition_distance** field defines the time in seconds. A value of 0xFFFF shall indicate that the time is greater than 0xFFFE seconds (18 hours, 12 minutes, 14 seconds).

35.1.4.17 User Group ID descriptor

The user_group_id_descriptor contains the User Group Id attribute defined in [48] to be associated with the data in a module.

Table 21: Syntax of user_group_id_descriptor

Syntax	No. of Bytes	Value
user_group_id_descriptor() {		
descriptor_tag	1	0x8B
descriptor_length	1	Variable
for (l=0; l< descriptor_length;		
l++) {		
text_char	1	User Group ID
}		
}		

The **text_char** field specifies the name of the module as a string of text characters. Text information is coded using the ISO Latin-1 character set.

35.1.4.18 Profile Descriptor

The profile_descriptor carries the Profile attribute defined in [48] for a module or group.

Table 22: Syntax of profile_descriptor

Syntax	No. of Bytes	Value
profile_descriptor () {		
descriptor_tag	1	0x8C
descriptor_length	1	0x01
profile_flags	1	Variable
}		

The **profile_flags** field defines the TeleWeb profile the module is intended to be displayed on.

The format of the **profile_flags** byte is shown below:

B7	B6	B5	B4	B3	B2	B1	B0
Reserved	Reserved	Reserved	Reserved	Reserved	Profile_3	Profile_2	Profile_1

Profile_1 is an active high flag representing the Profile 1 attribute specified in [48].

Profile_2 is an active high flag representing the Profile 2 attribute specified in [48].

Profile_3 is an active high flag representing the Profile 3 attribute specified in [48].

35.1.4.19 Transmission Schedule descriptor

The transmission_schedule_descriptor contains the Transmission Schedule attribute defined in [2] to be associated with the TeleWeb service.

Table 23: Syntax of transmission_schedule_descriptor

Syntax	No. of Bytes	Value
Transmission_schedule_descriptor() {		
descriptor_tag	1	0x02
descriptor_length	1	Variable
no_of_entries	1	Variable
for (i=0; i< no_of_entries; i++) {		
UTC_hours	1	hour 0..23
UTC_minutes	1	minute 0..59
repetition_distance	1	repetition distance in minutes
}		
}		

The **UTC_hours** field specifies the hours component of a “time” referenced to UTC. The valid range is 0 to 23 (decimal) inclusive.

The **UTC_minutes** field specifies the minutes component of a “time” referenced to UTC. The valid range is 0 to 59 (decimal) inclusive.

The **repetition_distance** field defines the time in minutes. A value of 0xFF shall indicate that the time is greater than 0xFE seconds (4 hours, 14 minutes).

36 TeleWeb DSM-CC transmission via Teletext

This section describes how Teletext data packets are used to transmit a TeleWeb service once it has been encoded into a data carousel as described in section 34.1.

36.1 Transmission of Data Carousels

A TeleWeb service shall always consist of eight data carousels: one two-layer carousel and seven one-layer carousels. The limitation of one two-layer carousel arises because there is no downloadId field in a DSI message to allow differentiation. The top level control message of each carousel shall be transmitted continuously. If a one-layer carousel is empty of data, the numberOfModules in its DII messages shall be set to 0. If the two-layer carousel is empty of data, the NumberOfGroups field in its DSI message shall be set to 0. The minimum repetition rate of empty carousels shall be defined in a code of practice.

The data carousel to which a particular DDB or DII message belongs shall be indicated through the coding of its downloadId field. The 29 MSBs shall all be set to 0x0. The 3 LSBs shall be set to 0x0 for the DDB and DII messages belonging to a two-layer carousel. A unique value in the range 0x1 to 0x7 shall be allocated to each one-layer carousel and the 3 LSBs of the downloadId field shall be set to this value.

The top-level control messages of each of the eight possible carousels shall be transmitted continuously. If a one-layer carousel is not in use, the numberOfModules field in its DII messages shall be set to 0. If the two-layer carousel is not in use, the NumberOfGroups field in its DSI messages shall be set to 0. The minimum repetition rate of empty carousels shall be defined in a code of practice

36.2 Transmission protocol

The transmission of each individual DSM-CC message described in section 35.1.1 is preceded by a delimiting character and followed by a 16-bit CNI code and a 16-bit checksum as shown in Figure 14. Successive DSM-CC messages are concatenated to form a serial data stream.

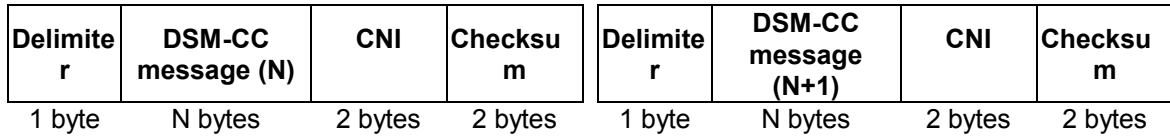


Figure 14: Message transmission sequence

36.2.1 CNI Code

The 2 Byte CNI code (MSB first) of the TV channel transmitting the TeleWeb service should be transmitted after each DSM-CC message and before the Checksum (CRC). In this case a TV channel change (especially if the data is received from an external satellite receiver) can be properly detected.

Note: Otherwise there is always the possibility that the messages of different services are mixed together which results in corrupted data.

The checksum should contain both the DSM-CC message and the CNI code.

36.2.2 Message checksum

Prior to transmission, the CCITT CRC algorithm as defined in Annex A is applied to a complete DSM-CC message and the 16-bit CNI code. The start value loaded into the shift register is 0xFFFF. If the message, the CNI code and the appended CRC value are received correctly, a receiver will calculate a value of 0x0000. The CRC value is appended after the CNI code with most significant byte occurring first. The addition of this CRC does not alter any size or CRC parameter integral to the message.

36.2.3 Delimiting between messages

A framing protocol is inserted prior to the start of each message in order to allow a receiver to identify the start and finish of the component messages of the data carousel. This protocol requires the special use of two code values, 0xC0 and 0xDB.

The code 0xC0 is used as a delimiter between messages and shall not appear in the data stream anywhere else. It shall be inserted immediately before the start of any DSM-CC message. If a data byte within a message, the CNI code or the checksum (calculated as described in section 36.2.2) has the code value 0xC0, it shall be replaced by the two byte sequence 0xDB, 0xDC.

If a data byte within a message, the CNI code or the checksum has the code value 0xDB, it shall be replaced by the two-byte sequence 0xDB, 0xDD.

Any number of 0xC0 bytes may be inserted between messages, allowing the byte to be used as a “time filler” if required. As a minimum, a single instance shall be inserted after the CRC for the previous message and before the first byte of the next message.

NOTE: Conceptually, the substitution of bytes in the data stream with values 0xC0 and 0xDB during the encoding process is carried out once the data stream has been assembled. Thus the substitution process does not alter the value of any message size, length or checksum values, etc. Accordingly, a receiver should substitute the sequences 0xDB, 0xDC and 0xDB, 0xDD with the appropriate single byte values before checking message sizes or checksums.

1 **36.2.4 Byte order**

2 The fields of the DSM-CC messages are transmitted in the order they occur in the message and descriptor syntax tables
3 shown in sections 35.1.2.1 to 35.1.2.3. Multi-byte values are encoded most significant byte first, i.e. Big Endian. The same
4 applies to the CNI code and the checksum.

5 **36.3 Teletext packet format**

6 The TeleWeb data stream formed according to section 36.2 shall be transmitted using IDL format B packets. These packets
7 are fully specified in subclause 6.8.1 of ETS 300 708 [52].

8 Restriction in the number of packets transmitted per field can be specified in the Code of Practice.

9 **36.3.1 Data Channel and Designation Code**

10 The four message bits of the Hamming 8/4 encoded Designation Code field shall be set to 1111 to specify an IDL. The values
11 permitted are specified in subclause 6.4.2 of ETS 300 708 [52].

12 **36.3.2 Application Identifier**

13 The four message bits of the Application Identifier byte shall be set to 0000 to indicate a TeleWeb service, as shown in Table
14 24. The message bits are numbered 1 to 4 for compatibility with [52].

15 NOTE: The Application Identifier value 0000 is allocated to TeleWeb in [62].

16 **Table 24: Allocation of the message bits in the Application Identifier byte**

B4	B3	B2	B1	Application
0	0	0	0	TeleWeb

17 **36.3.3 Format Type**

18 When the Application Identifier byte is set to the value defined in section 36.3.2, the coding of the Format Type byte is
19 according to Table 25. The message bits are numbered 1 to 4 for compatibility with [52].

20 **Table 25: Allocation of the message bits in the Format Type byte**

B4	B3	B2	B1	Application
0	0	0	1	reserved for future use
0	1	0	1	TeleWeb application, Short TeleWeb Service
1	0	0	1	TeleWeb application, Full TeleWeb Service
1	1	0	1	reserved for future use

21 B1 = 1 and B2 = 0 define the packet type to be IDL-format B.

22 **36.3.4 Continuity Index**

23 As defined in subclause 6.8.1.3 of ETS 300 708 [52].

24 **36.3.5 User Data**

25 The User Data bytes carry the data stream as defined in section 36.1.

1 **36.3.6 Forward Error Correction**

2 As defined in subclause 6.8.2 of ETS 300 708 [52].

3

Annex G CCITT CRC-16 (informative)

The following C program can be used to calculate the CCITT CRC-16. This code is for information only and is not optimised for efficiency. Any algorithm delivering the same result may be used. At the beginning the variable 'crc' is set to 0xFFFF. For each bit the data starting with the most significant bit of the first byte and ending with the least significant bit of the last byte the algorithm is executed with 'ser-data' set to the value of the bit. After the algorithm is called for the last bit the variable 'crc' holds the result of the CRC calculation.

For encoding, the most significant byte of this value is appended to the data followed by the least significant byte.

For decoding, the additional two CRC bytes are also fed to the CRC algorithm. In this case the result in the 'crc' variable should be 0 for the correct data.

```
// Update the CRC for transmitted and received data using
// the CCITT 16-bit algorithm ( $X^{16} + X^{12} + X^5 + 1$ )
unsigned char ser_data;
static unsigned int crc
crc = (unsigned char)(crc>>8)|(crc<<8);
crc ^= ser_data;
crc ^= (unsigned char)(crc & 0xff)>>4;
crc ^= crc <<12;
crc ^= (crc & 0xff) << 5;
```