

# ETSI TS 103 256 V1.1.1 (2014-10)



**Speech and multimedia Transmission Quality (STQ);  
Reference webpage for subjective testing**

---

Reference

DTS/STQ-226

---

Keywords

internet, quality

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2014.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM®** and the GSM logo are Trade Marks registered and owned by the GSM Association.

---

# Contents

Intellectual Property Rights .....	4
Foreword.....	4
Modal verbs terminology.....	4
Introduction .....	4
1 Scope .....	5
2 References .....	5
2.1 Normative references .....	5
2.2 Informative references.....	5
3 Definitions and abbreviations.....	6
3.1 Definitions .....	6
3.2 Abbreviations .....	6
4 General aspects of a reference web page.....	6
4.1 Web QoE reference page composition .....	6
4.2 Summary of typical parameters for a Web QoE reference page .....	7
4.3 Composition rules for a Web QoE reference page .....	8
<b>Annex A (informative): Configuration of the Web QoE page.....</b>	<b>9</b>
A.1 News content acquisition.....	9
A.2 Harvesting script requirements.....	9
A.3 RSS Feed harvesting script configuration .....	10
A.4 Database setup.....	11
<b>Annex B (informative): Installation of the Web QoE page.....</b>	<b>12</b>
B.1 Installation.....	12
B.2 Configuration .....	12
History .....	14

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://ipr.etsi.org>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Speech and multimedia Transmission Quality (STQ).

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

## Introduction

As interactive web services such as web browsing are gaining considerable importance in mobile broadband networks, it is also crucial to assess its performance in terms of QoS and QoE. While the interaction patterns of web browsing are simple (request - response), proper performance testing of this service in mobile communications networks is not. There is a complex interplay between web content, Internet and fixed network infrastructure, air interface, end-user devices including actual web browser applications and the end user's perception.

With regards to technical performance testing for web browsing, ETSI has taken efforts by approving the COPERNICUS web page and its successor KEPLER (<http://portal.etsi.org/stq/WebReferencePage.asp>) as reference web pages for technically instrumented QoS parameter tests in live networks. However, due to the fact that their content is not interesting for users to interact with, they cannot be effectively used in subjective tests for web browsing QoE. Therefore, in the present document content for subjective web-browsing QoE is developed.

---

# 1 Scope

The present document describes the structure of a news web page with an underlying database of news articles, which can be used for subjective QoE testing of web browsing. In addition, it also describes how the database and the web page are set up on a server and how content of different web pages can be harvested, in order to build up the database of news articles.

Different to the approach taken for deriving the parameters for the Kepler and Copernicus pages described in ETSI TR 102 505 [i.3], this page is not an "average" of several different pages, but is targeted towards the following focuses:

- 1) provisioning of news content that can be used throughout several consecutive web browsing sessions without repetition of already seen or read news content; and
- 2) create a page that adheres with W3C<sup>®</sup> guidelines for proper web design in terms of appearance and technical implementation in order to minimize the influence of badly designed or programmed web pages.

The page described within the present document is only a snapshot of a typical news page at the moment of creation, however it provides content that can be used by different QoE labs for testing and therefore allows comparisons of the results between different labs.

---

## 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, ETSI cannot guarantee their long term validity.

### 2.1 Normative references

The following referenced documents are necessary for the application of the present document.

Not applicable.

### 2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] Skadberg04 Y. X. Skadberg and J. R. Kimmel: "Visitors' flow experience while browsing a Web site: its measurement, contributing factors and consequences". Computers in Human Behavior, vol. 20, pp. 403-422, 2004.
- [i.2] MoellerRaake14 Quality of Experience: "Advanced Concepts, Applications and Methods". Springer, Cham, 2014.

NOTE 1: Running example of the test web page: <http://kepler.dataworkers.eu>.

NOTE 2: Download of current version: <https://ace.ftw.at/downloads/web-qoe-testing-content/web-content-v2.0>.

- [i.3] ETSI TR 102 505: "Speech and multimedia Transmission Quality (STQ); Development of a Reference Web page".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

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

**Quality of Experience (QoE):** degree of delight or annoyance of a person whose experiencing involves an application, service, or system

NOTE: It results from the person's evaluation of the fulfilment of his or her expectations and needs with respect to the utility and/or enjoyment in the light of the person's context, personality and current state (MoellerRaake14 [i.2]).

### 3.2 Abbreviations

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

CSS3	Third revision of the Cascading Style Sheets standard
DB	Data Base
HTML	Hyper Text Markup Language
HTML5	Fifth revision of the HTML standard
JS	Javascript
JSON	JavaScript Object Notation
PHP	Hypertext Preprocessor
QoE	Quality of Experience
RSS	Really Simple Syndication
URL	Uniform Resource Locator
W3C®	World Wide Web Consortium

---

## 4 General aspects of a reference web page

Quality of web experience represents an important topic for service or content providers. Measurement of such services at the end-user side is not a trivial task and various factors are influencing the overall QoE perception. In order to suppress the influence of the content a standardized web page is necessary. The most important requirement for such a webpage is the ability to establish a certain browsing mode for the test users and create a flow experience that is typical in web browsing (as shown by Skadberg04 [i.1]). Therefore the page should be interactive and has to provide sufficient content such that the test participant can browse through it over several consecutive test conditions without getting bored.

The aim of the Web QoE test content described in the present document here is to ensure above addressed requirements. Additionally the language of the presented news content should be selectable, such that the page can be used across different countries with identical technical properties (picture count, page size, etc.). This is achieved by the attached content harvesting script that downloads recent news from an RSS feed that can be specified in the configuration file.

The page can be stored on a local server within the test infrastructure, thereby eliminating imponderabilities from different interconnecting networks.

### 4.1 Web QoE reference page composition

After successful recording of the articles from the RSS feeds, all content of the web page is stored locally in the database, which allows perfect control over page loading times. Interactive content such as the news articles is updated on a daily basis, with the possibility to update the database also manually from the administration control panel. Images, and video files are stored in public folders, and all text information is stored in a local SQLite database, hence any calls to other servers, or network elements are eliminated which allows to out rule variances of the connection to the server via the Internet.

As the content of the news page is harvested via the attached script from the RSS feed provided in the configuration file, the page can be easily filled with content of any language. The internal structure of the page is designed to have minimal design and language impact on the overall perception, hence the page depends only on the RSS feed content and original posts.

The internal structure of the Web QoE reference page is based on the newest HTML5 and CSS3 standards (see note 1). Although these are not yet fully standardized, currently they represent the most spread technologies among modern web pages and applications. Overall page design and source code follows the W3C® standards.

NOTE 1: See <http://www.w3.org/standards/>.

The Web QoE reference page uses several open-source technologies, which are nowadays widely used around the Internet. Design of the web page utilizes Bootstrap CSS (Cascading Style Sheets) framework in 3.0 revision (see note 2). This framework allows rapid development of new page elements and ensures that the page remains appealing to the user, and the overall perception is not distracted by the badly programmed CSS code. This approach also allows to present the page on different devices and maintain usability of the application and readability of content. Responsive features of the Bootstrap CSS framework allow to control which parts or elements of the pages are displayed on which device. Hence it is possible to decrease the number of visible elements on the mobile devices, or to show advanced elements (e.g. a video player) on the devices with larger screens. Moreover, the Bootstrap CSS framework is maintained by large online community of developers, which ensures low number of issues and guarantees high cross-browser compatibility.

NOTE 2: See <http://getbootstrap.com/css/>.

Other technologies used for the Web QoE reference page originate from the best practices for web development as published and maintained by Html 5 Boilerplate (see note 3). This contains also the Apache configuration file for proper compression on the server side and optimal caching instructions, which are sent from the server to the client. These settings represent the optimal setup for current web applications and ensure minimal loading times and minimal amount of sent data.

NOTE 3: See <http://www.html5boilerplate.com>.

## 4.2 Summary of typical parameters for a Web QoE reference page

Table 1 shows the parameters of the existing Kepler and Copernicus pages versus the proposed Web QoE reference page as determined by ETSI TR 102 505 [i.3] document (defining development of a reference Web page).

**Table 1**

	<b>Total size, bytes</b>	<b>% non-lossless compressible</b>	<b>Object count</b>	<b>Composition</b>
Kepler	807 076 (Windows® size)	60	75	125 000 bytes in "flash video" surrogate, 3 jpgs of ~45 000 bytes each
Copernicus	200 000	57	38	Largest object: ~36 000 bytes, type jpg
Subjective Web QoE reference page	690 000 (customizable)	50 (customizable)	23 (customizable)	Video, content images, text of articles, CSS files, JS files

## 4.3 Composition rules for a Web QoE reference page

Design and composition of the page was developed by taking into account the recommendations for the development of reference web pages from W3C<sup>®</sup> and general concepts of various existing news pages on the Internet (Spiegel.de<sup>®</sup>, Guardian.com<sup>®</sup>). Therefore, the following features are essential.

- Responsive design, such that the web page is usable via PCs, Tablets and smartphones. Moreover, the site is presented in the optimal, device-dependent design, such that the usability and the overall content experience are ensured.
- Configurable number of articles on one page, hence configurable amount of data which have to be loaded.
- Option to play video content on the page.
- Configurable news ticker, top articles position and visibility and main content area with downloaded news from a RSS feed.

### EXAMPLE:

- 3 "Top stories" with 3 medium sized images (non-compressible content), header and teaser text (loss-less compressible elements).
- 5 "Recent news" with 5 small sized images (non-compressible content), header and teaser text (loss-less compressible elements).
- Video player - Largest single file non-compressible content. Configurable to load and play video immediately or after hitting play button. Hence the large amount of data could be loaded at the beginning, or only after user action.
- 5 main articles with medium sized images (non-compressible content), header, teaser text and content excerpt (loss-less compressible elements).



---

## Annex A (informative): Configuration of the Web QoE page

To fill the news page with realistic content, the installer package includes the PHP script which harvests a defined RSS feed, stores the text in a SQLite-DB and saves referenced/contained images to local storage. The RSS feed contains the news content itself and not only a link to the original page. Therefore the content itself represents a real news the user is acquainted with, and moreover, the connection to 3rd party servers is not affecting the loading behaviour of the page.

---

### A.1 News content acquisition

To get the harvesting script running, a proper RSS feed with certain properties needs to be selected, and the configuration file `app/config/config.neon` needs to be adapted accordingly. We prepared two 2 RSS feeds as an example that meets the required criteria:

For English content, The Guardian® News RSS feed can be used. To be found on <http://www.theguardian.com/uk>.

For the German content, the Spiegel.de® RSS feed can be used. To be found on <http://www.spiegel.de/>.

General criteria for the RSS feed are these:

- Title of the news.
- Title Image with sufficient width. Width more than 400 px is recommended.
- Text teaser, which sums up what the respective news is about.
- Text of the news itself.

---

### A.2 Harvesting script requirements

The RSS Feed needs to fulfil several aspects, so the RSS harvesting script on your server can accept it. The output RSS feed have to include these fields:

- *'title'* - **REQUIRED** - Title of article to be displayed on a new page.
- *'link'* - **REQUIRED** - Original link to the actual article on a 3rd party server. This link is used for downloading the content and further processing within the harvesting script.
- *'pub\_date'* - **REQUIRED** - Original publication date of an article.
- *'category'* - **RECOMMENDED** - Category of an article. This category is also used for creation of a category menu.
- *'description'* - Complete article body with all images, teaser text and body text.
- *'guide'* - This field is usually present in every RSS feed, and does not need to be changed. Content of *'guide'* is not used in presentation of a web page.

Here is an example of a PHP object retrieved from a Yahoo Pipe RSS JSON string:

```
SimpleXMLElement(6) {
  title => "Grangemouth petrochemical plant to close - live" (47)
  link =>
"http://feeds.theguardian.com/c/34708/f/663879/s/32cbd936/sc/2/l/0L0Stheguardian0N0Cbusiness0C20A130
Coct0C230Cchinese0Ebanking0Efears0Emarkets0Eminutes ... " (176)
  description => "<p>Up to 800 jobs at risk as Ineos announces closure of part of its Grangemouth
plant, after a bitter dispute with management</p><div class='author'>< ... " (1394)
  category => 'business'
  pubDate => "Wed, 23 Oct 2013 12:13:49 GMT" (29)
  guide => "http://www.theguardian.com/business/2013/oct/23/chinese-banking-fears-markets-minutes-
greece-live" (97)
}
```

---

## A.3 RSS Feed harvesting script configuration

The PHP script URL for manual recording of the new articles to localhost server is located in `example.com/feed/`. For using the CRON process please rather use the direct address: `example.com/feed/recordConsole`.

This harvesting script is responsible for finding the images within the HTML file loaded from the RSS link feed. It records and stores only articles with image widths larger than 300 pixels. Lower resolution images will not be stored.

Before running the script, it is necessary to configure several variables in the `app/config/config.neon` configuration file:

```
common:
  parameters:
    ...
    rssFeeds:
      url : "http://feeds.theguardian.com/theguardian/world/rss"
      image: '#main-content-picture > img'
      teaser: '#stand-first'
      body: '#article-body-blocks'
    ...
```

The above code serves as an example how to set up TheGuardian.com® news RSS. For the parts of the page, which we want to use we have to define its unique CSS Selector, which will be used for searching of the content in the downloaded HTML file.

- *url*: The actual URL of the RSS stream which we want to use.
- *image*: Title image, with sufficient width. The title image can be addressed with `#main-content-picture > img` CSS Selector.
- *teaser*: Teaser text - usually under the title image.
- *body*: Column with the actual news.

For the sake of a complete example, we provide also the configuration for the Spiegel.de® RSS stream:

```
common:
  parameters:
    ...
    rssFeeds:
      url : "http://www.spiegel.de/schlagzeilen/index.rss"
      image: '.js-module-box-image img'
      teaser: '.article-intro'
      body: '.article-section'
    ...
```

---

## A.4 Database setup

The PHP script stores all extracted data to the `SQLITE` database located in the application root folder. The name of the database file, which is to be used for storing the content can be configured like this:

```
common:
  parameters:
    ...
    database:
      file: 'kepler_news.db'
    ...
```

If the database is not yet located in the root folder, you can go ahead and create it by running `kepler_news_db.sh` script, provided with the installer package.

Initial database with example data can be also found in the root directory of this application. The file `kepler_news_init.db` contains all necessary data to run this web page. Before using it, do not forget to change the database file name in to `app/config/config.neon` configuration file, as shown above.

---

## Annex B (informative): Installation of the Web QoE page

PHP version 5.2.0 or higher to run Nette Framework is required. Also, the hosting server has to comply with the required configuration of PHP. A requirements checker is also included in this package, and can be found in *checker/checker.php*. To check your server compatibility just run the script from your browser by typing url to *checker.php* script (in example *http://example.com/checker/checker.php*). Now you will find if your server meets minimal requirements.

Green message with "Congratulations!" is indicated that one can use Nette right away!

---

### B.1 Installation

The installation is straightforward: just download and unpack the .zip file into a directory accessible from a web server. After the database has been successfully initialized from the provided script, and the application structure has been copied to a web server, you need to configure the application.

---

### B.2 Configuration

The news page module needs to be configured in the *app/config/config.neon* file. Some preview of the config is displayed below:

```
common:
  parameters:
    page:
      pageName: Newspaper.at
      articlePerPage: 5

  database:
    file: 'kepler_news.db'

  images:
    # Folder with saved images from RSS feeder, relative to WWW folder
    # folder: images/startup-pics/article_images/
    folder: images/article_images/
    gallery: images/galleries

  video:
    video1:
      # Type the title and the relative src url (without suffix)
      title: "Spiegel TV: Sintel is presenting a new cartoon movie. Trailer available"
      src: flowplayer/videos/sintel_trailer-720p
    video2:
      title: "Spiegel TV: Video title 2"
      src: flowplayer/videos/sintel_trailer-720p
    video3:
      title: "Spiegel TV: Video title 3"
      src: flowplayer/videos/sintel_trailer-720p

  login:
    user: etsi
    password: testcontent12
    required: false

  rssFeeds:
    url : "http://feeds.theguardian.com/theguardian/world/rss"
    image: '#main-content-picture > img'
    teaser: '#stand-first'
    body: '#article-body-blocks'
```

Description, how to write and edit a config file in "neon" syntax can be found here.

In this file you can configure database connection, and also image folders relative to the www folder - as they have been defined in former section about RSS feeder here.

The configuration file is very straight-forward, the following variables can be set:

- *page* - Basic variables like name of the page visible in the header of the page. Also number of articles visible on the homepage and the category pages can be set here.
- *images* - Here you need to set image folders - folder variable holds the images - downloaded for the corresponding articles.

*video* - Holds the information about videos in the video player. Since the player is relying on HTML5 technology, every video needs to be recorded with both .mp4 and .webm suffixes.

---

## History

<b>Document history</b>		
V1.1.1	October 2014	Publication