

Document-Info

Document	Data Application NewsService Journaline® – Overview (Journaline_Overview_1.3_20040318.doc)
Author	Alexander Zink
Service Definition:	Thomas Frühwald, Thomas Jaumann, Gerd Kilian, Olaf Korte, Markus Prosch, Michael Reichenbächer, Alexander Zink
Date	2004-03-18
Version	1.3
Status	Fraunhofer IIS Standard

Content

1. ABSTRACT	3
2. FUNCTIONALITY	4
3. OBJECTS – MESSAGES AND MENUS	5
3.1 Types of Objects	5
3.1.1 Menu Objects – Menus	5
3.1.2 Message Objects	6
3.2 Object Hierarchy	8
4. OBJECTS – FORMAT SPECIFICATION	9
4.1 General Object Description	9
5. RECEIVER BEHAVIOR	10
5.1 Initial Object	10
5.2 Object History	10
5.3 Favorites Functionality	10
5.4 Content Update While Object Is Displayed	11
5.5 Reception Status Indication	11
5.6 Object Caching Strategies	11
6. DATA APPLICATION DEFINITION	12
6.1 Transport in DAB – Digital Audio Broadcasting	12
6.1.1 Data transport	12
6.1.2 User Application Signaling	13
6.2 Transport in DRM – Digital Radio Mondiale	14
7. REFERENCES	15

1. Abstract

The purpose of this document is to describe and define the XML based information service 'NewsService Journaline®'.

'NewsService Journaline®' is a textual data application optimized for digital broadcast systems like DRM (Digital Radio Mondiale, see [drm]) and DAB (Digital Audio Broadcasting, see [dab]).

This document demonstrates the overall functionality of the data application 'NewsService Journaline®', specifies the broadcast system signalization and describes the expected behavior for a 'NewsService Journaline®' compliant receiver.

2. Functionality

Through 'NewsService Journaline[®]', the radio user can easily and instantly access information according to her or his current type of interest. The information is provided in simple textual form with the future option for richer graphical representation including multimedia elements like images or video sequences.

The information is hierarchically organized in the form of Menus. Every Menu contains a list of Sub-Menus and/or Messages (e.g. news of a certain topic). Messages are the leaves of the hierarchical structure and contain one actual information unit (one news message) each.

Messages and (Sub-)Menus will be called '**Objects**' ('**Message Objects**' and '**Menu Objects**') in the remaining document.

'NewsService Journaline[®]' is based on public standards:

- Objects are generated and fed into the 'NewsService Journaline[®]' enabled transmission system as XML files [xml10].
- 'NewsService Journaline[®]' capable receivers can easily restore an XML structure of each Object for further processing [xml10].

'NewsService Journaline[®]' is highly optimized for the use in digital broadcast systems:

- Objects are broadcast using data carousels.
- Objects are sorted into priority classes to meet individual transmission repetition requirements.
- To handle potential receiver rendering limitations, an Object's content is formatted as pure textual information.
- To prevent unnecessary data overhead (and thereby minimize transmission time), Objects are re-formatted for transmission into NML – the XML structured binary 'NewsService Markup Language'.

3. Objects – Messages and Menus

3.1 Types of Objects

'NewsService Journaline®' currently specifies three types of Objects:

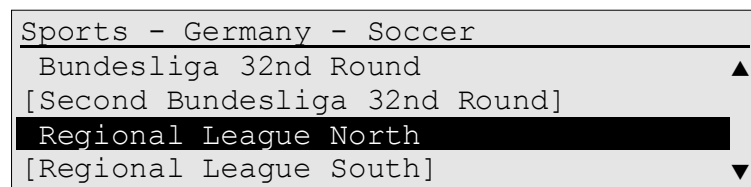
3.1.1 Menu Objects – Menus

Menus contain a title and a list of links as the body section. If the user scrolls to and activates one of the provided labels, the linked Object shall be displayed (and thereby replaces the current Object on the screen).

Every link contains a reference to a foreign Object and a label, which is shown as one line on the screen. If the title or a label exceeds the length of one line on the screen, the content might be truncated. Any font may be used for rendering this type of Object.

Content Update (see also chapter 5.4):

In case of an information update while the Menu Object is displayed on the screen, the user's original scrolling situation shall be restored (thus the same body section line index shall appear at the same screen location, provided that the new Menu Object contains enough lines of information).



Inverted line: current user selection (cursor indicator)

Lines with squared brackets: linked Object (Message/Sub-Menu) was not yet received

Up/down arrows on the right: more menu items above/below

3.1.2 Message Objects

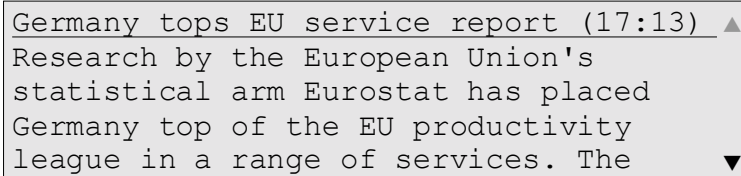
3.1.2.1 Plain Text Messages

Plain Text Messages contain a title and body text.

For presentation to the user, the title and the following body text is rendered on the screen with automatic line wrapping. The user scrolls vertically through the text. The Message title is attached to the text (and therefore probably scrolls out of view). Any font may be used for rendering this type of Object.

Content Update (see also chapter 5.4):

In case of an information update while a Message is displayed on screen, the new Message is displayed scrolled back to the top.



Germany tops EU service report (17:13) ▲
Research by the European Union's
statistical arm Eurostat has placed
Germany top of the EU productivity
league in a range of services. The ▼

Light up arrow on the top right:

head of message is shown, no more text above (title would scroll out of view)

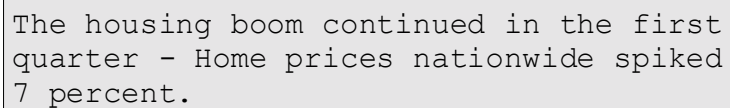
3.1.2.2 Title-Only Messages

Title-Only Messages contain only a title and no body text.

For presentation to the user, the title text is rendered on the screen with automatic line wrapping. The user scrolls vertically through the text. Any font may be used for rendering this type of Object.

Content Update (see also chapter 5.4):

In case of an information update while a Message is displayed on screen, the new Message replaces the existing message immediately (without explicit user interaction).



The housing boom continued in the first
quarter - Home prices nationwide spiked
7 percent.

Might be displayed centered both horizontally and vertically;
Automatic updates without user interaction

3.1.2.3 List Messages

A List Message contains a title and a list of text lines.

For presentation to the user, the title should remain on the screen even if the user scrolls through the lines of information in the body section. A non-proportional font is recommended for rendering this type of Object.

If the title or any text line exceeds the length of one line on the screen, the list item might be truncated.

Content Update (see also chapter 5.4):

In case of an information update while the Message is displayed on the screen, the user's original scrolling situation shall be restored (thus the same body section line index shall appear at the same screen location, provided that the new Messages contains enough lines of information).

Soccer - Bundesliga 32nd Round (16:15)		
TSV 1860 - Cottbus	3:0	▲
Dortmund - Nürnberg	4:1	
Hertha - Bayern	3:6	
Stuttgart - Bremen	0:1	▼

On automatic content update: scroll position within list would be retained

3.2 Object Hierarchy

Every Menu contains a list of references to other Menu Objects or Message Objects. Messages are the 'leaves' of the tree-structure created by Menus.

The Main Menu Object is initially shown when the user tunes to the 'NewsService Journaline®' for the first time without a specific Object ID requested. It has the fixed Object ID of value '0x0000' (see below for explanation).

Note that the Main Object may also be a Message Object e.g. in case of a service-out-of-order message.

A **maximum of 20 hierarchy levels** is set (including the Main Menu and the final Message Object).

Every Object must be referenced from at least one Menu Object. It may be referenced from more than one Menu Object.

The Main Menu/Message Object (Object ID 0x0000) is the only Object which does not need to (but may) be referenced from a Menu Object.

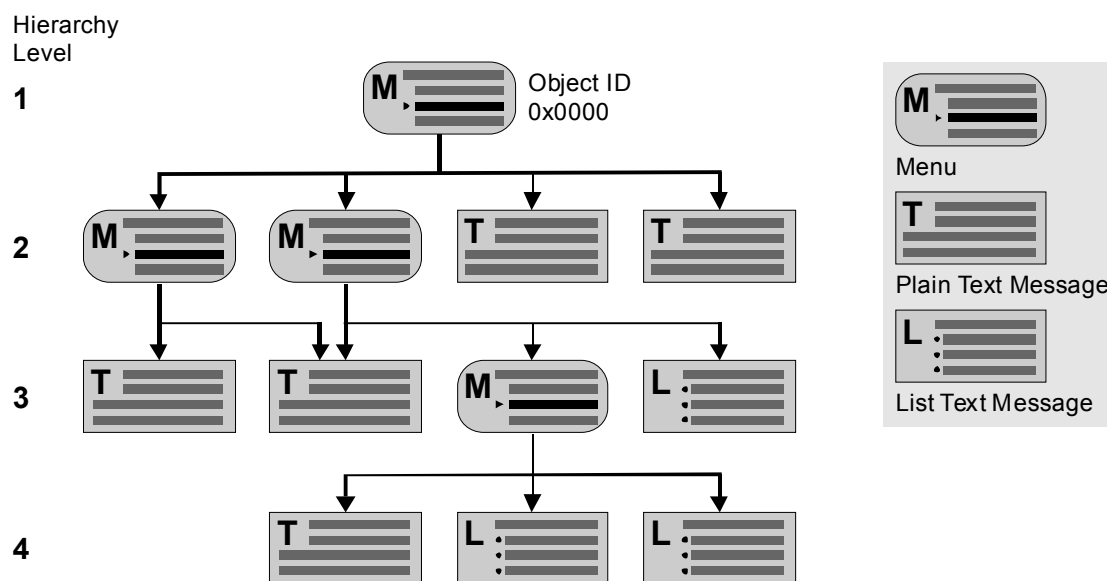


Figure 1: Hierarchical Object ordering example

4. Objects – Format Specification

4.1 General Object Description

An Object (Message or Menu) comprises an Object Header (in a binary format) and a Content Section (comprising a title section and an optional body section).

Note: In case of a very short message text the body section of a Plain Text Message Object may be absent (so that the Plain Text Message Object's Content Section only consists of a title section while the body section is omitted).

An Object is self-descriptive. No external information (like filename or type specification) is required to reference/access an Object and render it on the screen.

An **Object is limited to 2044 bytes** (the Object's Header plus Content Section length). This value refers to the **uncompressed version** of the Object's Content Section. Objects **must not be compressed** if the total Object size of the compressed object exceeds the size of the uncompressed Object!

The title and body section of an Object may contain escape sequences to implement extended functionality.

5. Receiver Behavior

Some general rules apply to 'NewsService Journaline®' compliant receivers.

5.1 Initial Object

If the user tunes into the 'NewsService Journaline®', the first Object to be displayed is the Object with Object ID 0x0000 (Menu or Message).

5.2 Object History

The receiver should always keep a record of all Menu Object IDs ('path' of Object IDs) from the currently displayed Menu/Message Object back to the Main Menu Object (Object ID 0x0000). In this way it can provide a 'back to higher menu level' functionality. To prevent circular references, the receiver should check whether a new Object ID is already contained in the current 'path' of Object IDs. In this case it might remove all items from the 'path' from this Object ID on.

5.3 Favorites Functionality

The idea of a 'Favorites' functionality is to enable the user to store certain Menu or Message Objects for quick and direct future access.

If an Object Header's 'Static Flag' is enabled, the Object ID is considered static for a certain Menu or Message Object (e.g. the Menu Object covering 'Weather forecast Bamberg' may always be broadcast by a certain NewsService as Object ID 0x0020). Therefore it is sensible to allow the user to add such an Object to a 'Favorites' list (if such a functionality is intended to be offered by a certain receiver).

Objects with the 'Static Flag' disabled are supposed to carry random content (e.g. messages belonging to different topics with every Object update). If a receiver offers a 'Favorites' functionality, the user should be prevented from adding such an Object to her or his list of 'Favorites'.

A receiver providing 'Favorites' functionality should store the full 'path' of Object IDs from the Main Menu/Message Object (ID 0x0000) through all Menu Objects to the final Menu/Message Object to be stored to the Favorites. Otherwise the receiver will not be able to provide a 'return to higher menu level' functionality to the user the next time the user recalls this 'Favorite' Object (although the Object itself as well [as all its Sub-Objects in case of a Menu Object] would still be fully usable).

5.4 Content Update While Object Is Displayed

If an Object with a certain Object ID is currently displayed (presented to the user) when an updated version of this same Object (identical Object ID but different Revision Index) is received, it should at least be signalled to the user that an updated version of this Object was received (e.g. by showing a flashing 'Update' icon on the screen).

In case of 'Menu Objects' or 'List Message Objects' it is recommended to immediately update the screen with the new Object's content while maintaining the user's current relative position within the list (list item index) at the same position on screen. Instead of the list item index, a receiver might also evaluate the list item text or (in case of a Menu Object) the Object ID of the referenced Object.

In case of 'Title-Only Message Objects' it is recommended to immediately update the screen with the new Object's content.

5.5 Reception Status Indication

If a Menu Object contains references to Object IDs which were not yet received, the menu item should be presented to the user nevertheless. Still it should be clearly indicated to the user which menu items are instantly available (the referenced Objects were already received) and which are not (e.g. by surrounding the menu item title with squared brackets).

5.6 Object Caching Strategies

If a receiver has limited caching capacity and runs out of memory, it is recommended to first remove Objects from memory which are not accessible from the current 'Menu Object path' (from any Menu Object between the Main Menu/Message Object and the currently displayed Object). If still more memory needs to be freed, the oldest Objects from the furthest Menu Object should be removed first while deleting Message Objects before Menu Objects and Objects with the 'Static Flag' disabled before those with this flag set. The Objects which can be accessed by the user through a 'Favorites' feature (if applicable) or which are directly accessible from the current Menu Object should only be removed from memory as a last option.

If a receiver has extremely limited caching memory capacity, it may behave in a TV's 'Video Text Mode' such that only the currently displayed Object (along with the 'Object ID path' belonging to the currently displayed Object) is kept in memory. If the user requests another Object, she/he has to wait until this requested Object is received the next time.

6. Data Application Definition

6.1 Transport in DAB – Digital Audio Broadcasting

'NewsService Journaline®' can be broadcast via DAB – Digital Audio Broadcasting (see [dab]) as a new user application.

6.1.1 Data transport

Every single 'NewsService Journaline®' Object is transported as one 'MSC data group' (see [dab] chapter 5.3.3 'Packet mode – data group level').

A MSC data group contains the following items:

- MSC data group header (2 or 4 bytes)
- Session header (optional, 3+n bytes)
- MSC data group field, which carries one 'NewsService Journaline®' Object (m bytes, max. 2044 bytes)
- MSC data group CRC (2 bytes),
which is mandatory for the 'NewsService Journaline®'

The following restrictions apply to 'NewsService Journaline®' Version 0x00:

The MSC data group header field shall have the following layout:

- Extension flag = 0
(if set to 1, the receiver must support Conditional Access to decode the information; if a receiver does not support Conditional Access, it must discard this MSC data group)
- CRC flag = 1
- Segment flag = 0 (value depends on Extension flag)
- User access flag = 0 (value depends on Extension flag)
- Data group type = 0000_b ('General data')
- Continuity index:
increments continuously for every Object, but may safely be ignored
- Repetition index:
will usually carry the value 0000_b, but may safely be ignored
- Extension field:
not present if Extension flag is set to 0 (CA is not used for 'NewsService Journaline®')

The Session header is not present (if Segment flag and User access flag set to 0).

The MSC data group CRC field is present (CRC flag shall be set to 1). See [dab] 'Annex E (normative): Calculation of the CRC word'.

6.1.2 User Application Signaling

The following information shall be used to signal the DAB data application 'NewsService Journaline®' (see [dab] chapter 8.1.20 'User application information').

The preliminary 11-bit '**User Application Type**' ID for the 'NewsService Journaline®' shall be **0x44A** (from the range of proprietary user applications).

This value equals 100 0100 1010_b or (100 0000 0000_b [proprietary user application] + 'J' [as ASCII code]).

The '**User Application data**' field shall have the following structure:

- **Version** – 1 byte unsigned integer; current value: 0x00:
Indicates the current version of the 'NewsService Journaline®' in use.

Note:

All future versions will be downward compatible such that a version-0x00 receiver still can decode all version-0x01, 0x02 etc. data, although it may not be able to decode or use some additional information.

Additional information may be added in a fully downward compatible way in two places:

- in the overall Service Signaling ('DAB User application information') by extending the length of the 'User Application data' section (while keeping all the existing bytes and their definition intact as specified in any previous version of the 'NewsService Journaline®')
- in the Object Header Section of every single Object by extending the length of the 'Extended Header' field (while keeping all the existing bytes and their definition intact as specified in any previous version of the 'NewsService Journaline®')

Note that a new 'User Application Type' ID must be assigned to introduce changes to the current specification of the 'NewsService Journaline®' which are incompatible with previous revisions of the 'NewsService Journaline®' specification.

- **Length of Extended Header** (number of 'Extended Header' bytes in the Header Section of every Object) – 1 byte unsigned integer; current value: 0x00:
The length of the Extended Header field is equal for every Object. For future (downward compatible) extensions of the 'NewsService Journaline®' this field may carry configuration information per Object. A 'NewsService Journaline®' receiver not able to interpret the information in an Object's Extended Header field may safely ignore it. However every receiver must at least be able to skip the Extended Header field by interpreting the 'Length of Extended Header' value given in the 'User Application data' field of the DAB 'User application information' signaling.

6.2 Transport in DRM – Digital Radio Mondiale

‘NewsService Journaline®’ can be broadcast via DRM – Digital Radio Mondiale (see [drm]) as a data service component.

The data application ‘NewsService Journaline®’ is signalled in DRM’s SDC data entity 5 (‘Application Information Data Entity’), see [drm] chapter 6.4.3.6.

Every single Object is transported in Packet Mode as one DRM Data Unit carrying a ‘MSC data group’. See chapter 6.1.1 ‘Data transport’ for details.

The data application is signalled as belonging to the ‘DAB domain’ using the Service Signaling (‘User Application Type’ ID and ‘User Application data’) as specified in chapter 6.1.2 ‘Document-Info’.

7. References

- [dab] "Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to mobile, portable and fixed receivers", ETSI EN 301 401 V1.3.2, 2000-09,
<http://www.etsi.org>
- [drm] "Digital Radio Mondiale (DRM); System Specification", ETSI ES 201 980 V.1.2.1 (2002-07, Final Draft),
<http://www.etsi.org>
- [xml10] "Extensible Markup Language (XML) 1.0 (Second Edition)",
W3C Recommendation, 2000-10-06,
<http://www.w3.org/TR/2000/REC-xml-20001006>
- [rfc1951] "DEFLATE Compressed Data Format Specification version 1.3";
RFC1951, 1996-05,
e.g. from <http://www.rfc-editor.org>

*** End of Document ***