XCMD=

<rds> <item

The

X-Command

for RDS Encoders

<artist>

**A communication protocol extension
for exposing tagged information
in FM radio broadcast**

<news>

Revision 2020-10-01

</item> </rds

Ing. Jan Kolar, PIRA Digital s.r.o.

Pira.cz

Table of Contents

Introduction

Benefit of the X-Command is well illustrated on following situations. A typical broadcast automation system is taking internal text fields and loading the RDS encoder by a text describing current program being broadcast.

1. Conventional RDS Encoder ∀

```
TEXT=James Arthur - Say You Won't Let Go↵
```


Figure 1 – Final result – standard RDS encoder

2. RDS Encoder with X-Command Support ∀

```
XCMD=<rds><item><dest>3</dest><text><artist>James Arthur</artist> -
<title>Say You Won't Let Go</title></text></item></rds>↵
```


Figure 2 – Final result – RDS encoder with X-Command support

Motivation for the X-Command Implementation

The X-Command for RDS encoders has been originally developed by Pira.cz to create an extended interface for forwarding tagged text information provided by FM broadcast automation systems. Because the X-Command was positively accepted by the radio community, several extensions were subsequently added.
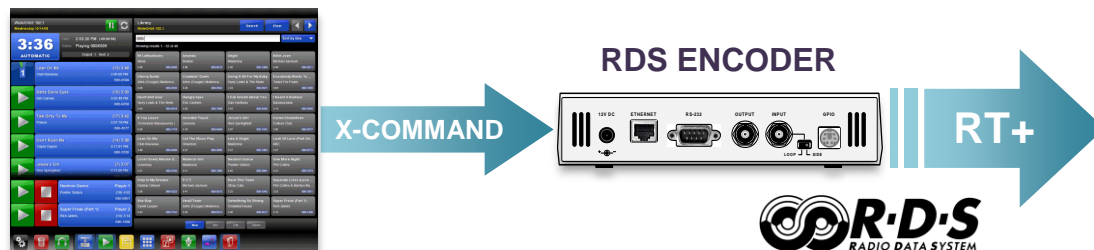

Figure 3 – The X-Command tagging application example

Radiotext Plus (RT+) is a technology based on Radio Data System (RDS) standard that tags text messages so specific content type can be retrieved from them. The RT+ enhances possibilities of existing Radiotext service as it helps the receivers to recognize and classify what kind of information is currently put in the Radiotext. The purpose of RT+ is to enable analogue FM RDS radios to display metadata such as Artist and Title details for songs, scrolling news headlines, information about the radio station, etc. The investment in RT+ is future proof – RT+ equivalents exist in all digital radio and mobile audio technologies: DMB/DAB/DAB+, DRM+, HD Radio, MP3 etc.

The RT+ specification was published before 2007. After a period of 10 years we can make following summarization:

- RT+ is supported by many RDS encoders. Although the level of support significantly varies between products of different brands, radio stations can select a suitable solution from the market offer.
- RT+ is well supported by current radio receivers, especially receivers integrated in mobile equipment and in car radios, which nowadays represent typical equipment for FM radio listening
- Almost all FM radio stations use some kind of broadcast automation software which holds appropriate text information describing current program, incl. separate artist and title fields, commercials, news, program guide etc.
- Despite all this, expansion of the RT+ is still unsatisfactory in 2017 (globally estimated at less than 10 % of radio stations using RDS).

The reason is primarily not inside the products themselves but in the manner *how the products communicate* between themselves. Loading the RDS encoders with RT+ metadata has not yet been covered by any suitable standard which could help mass deployment of the RT+. Let's look at some methods which are currently in use:

- **RTP= command or equivalent**

  This method provides the tagging data in two steps. First, the plain text is sent to

the RDS encoder. Next, it is necessary to specify positions in the text where to find the tags:

```
TEXT=James Arthur - Say You Won't Let Go↵
RTP=04,00,11,01,15,19↵
```

This solution is still well suitable for connection between control software and RDS encoder of the same brand. The RTP= command was however unsuccessful in general use. Support of the RTP= command across different broadcast automation systems is insufficient. Providing the information in two steps may disallow smooth visual transition between the tracks. Last but not least, it makes only a small sense to annoy software developers with RDS and RT+ coding as this should be entirely a job for the RDS encoder.

- **UECP free format or ODA groups**

  This solution is sometimes used by regional or national stations for RT+, it is however obsolete due to its complexity, requirement of access to low layers of RDS processing, dependence on the RDS encoder's setup and possible conflicts with other RDS services. The UECP does not provide any widely applicable method how to send metadata.

The X-Command brings a solution for all the disadvantages listed above. It unambiguously tells the RDS encoder *what* do to, without solving *how* to do that. One text, one line, metadata included.

> *"The X-Command potentially brings the same revolution to the RDS like the RT+ service as it finally allows the RT+ to become widely used."*

Even through the broadcast automation system does not handle the RDS RT+ feature, the X-Command allows forwarding the metadata to the RDS encoder without losing any information. Just send the X-Command and forget. The RDS encoder does the entire job – parsing of the command, loading the text, RT+ coding and optimization and sending everything to your listeners.

The X-Command Characteristics

- Based on simplified markup language
- Single line format
- XML and RT+ compatible
- Already supported by many broadcast automation systems
- Performs better than RTP= command used by conventional RDS encoders
- The use is completely free of charges and restrictions
- No need to know anything about RDS RT+ coding
- Low consumption of system resources
- Reduced protocol overhead (usually less than 50 % of total capacity)
- Simply readable and writable by human
- Open to future extensions

The X-Command Support

The X-Command is currently supported by following RDS encoders and devices from Pira.cz:

	P132 RDS Encoder
	P232(U) RDS Encoder
	P332 RDS Encoder
	P164 RDS Encoder
	P232 Microcontroller

The X-Command is naturally supported by all broadcast automation systems which enable the user to define output data template. See the section 'Recommended Steps for Broadcasters' for more details.

Frequently Asked Questions

Is the X-Command compatible with CENELEC RDS standards?

The X-Command only extends standard set of commands accepted by the RDS encoder on its communication port. Therefore, no compatibility issue can occur from the perspective of RDS standards. For tagged text, the X-Command leads to standard RT+ service (see the figure 3).

The X-Command Specification

The X-Command Coding Rules

Communication platforms

The list of communication platforms suitable for the X-Command is including but not limited to: serial RS-232 channel, TCP connection, reading file via HTTP protocol etc.

The X-Command prefix

The X-Command entry normally starts with `XCMD=` prefix placed before the content. The prefix may be omitted if misinterpretation cannot occur. That case must be explicitly specified in the RDS encoder's documentation.

Total length limit

The X-Command content length including all necessary elements and tags must not exceed 255 bytes. Since the Radiotext length is limited to 64 characters by the RDS standard (128 characters in possible future enhancement), the overall length limit is still bountiful but allows seamless implementation also on the smallest platforms and encapsulating into other communication protocols.

Escape sequences and character replacing

The tags (incl. angle brackets) accepted by the X-Command do not occur in the text under consideration. Moreover, escaped text is potentially not available from the source. Thus, no escaping is required in general. The X-Command parser in the RDS encoder shall ignore an occurrence of single angle bracket in the text. If angle brackets form a fake (unknown) tag, that tag will completely disappear from the resulting text. The X-Command processor shall replace < and > entities with correct characters < and >.

The X-Command processor replaces characters with ASCII code 31 or lower with a space character and a sequence of space characters with a single space.

Character encoding

The X-Command expects UTF-8 encoding. Conversion to EBU Latin CP is made inside the RDS encoder. Systems using 8-bit ANSI encoding must perform either conversion to UTF-8 or removal of all characters with ASCII code 128 or higher.

An exception exists for fixed text messages stored in the RDS encoder's non-volatile memory. These text messages (in X-Command format) are encoded according to the RDS encoder's default (usually EBU Latin CP).

Compatibility with XML

The X-Command content can be generated using the same tools and rules valid for the XML (Extended Markup Language). The X-Command prohibits use of attributes inside the tags. The XML declaration (prolog) is accepted but ignored.

Case sensitivity

The X-Command is case insensitive when parsing the elements.
For example, tags <title>, <TITLE> or <Title> have the same functionality.

Terminating character

The X-Command entry must be terminated by CR character (0x0D) in case of serial or
TCP connection. The terminating character is not required for file input type.

Root element

The X-Command content is delimited by element **<rds>…</rds>**. Anything outside
this root element is ignored by the X-Command parser (but it's still counted in the total
length limit). If the root element is not found or incomplete, no operation is performed.

Child elements

Several child elements are defined. The child element represents a specific operation
to be performed:

Child element	Operation
<item>…**</item>**	Loads a text (for example 'now-playing' information) into a destination (for example Radiotext and RT+).
<tmr>…**</tmr>**	Enables a timed loop of fixed text messages which were previously stored in the RDS encoder's non-volatile memory. The text messages are stored in the X-Command format.
<msg>…**</msg>**	Processes a fixed text message which was previously stored in the RDS encoder's non-volatile memory. The text message is stored in the X-Command format.
<exe>…**</exe>**	Executes an ASCII command supported by the RDS encoder.
<hd>…**</hd>**	This optional header defines site address and encoder address to which the X-Command is being sent. It can be used together with any of the elements listed above.

If not stated otherwise, only one child element is supported inside the root element. If
no supported child element is found, no operation is performed.

The <item> element

The "item" typically represents information related to current program being broadcast. Only one item is accepted inside the root element.

The element format

The **item** element always forms a single line (without CR/LF characters). Default **item** element format is as follows:

<rds><item><dest> *destination_code* **</dest><text>** *tagged_text* **</text> </item></rds>**

With optional timeout parameter the format is as follows:

<rds><item><dest> *destination_code* **</dest><text>** *tagged_text* **</text> <tmo>** *timeout* **</tmo></item></rds>**

List of destination codes

The destination code effectively determines how the text information will appear on the receiver.

Code	Destination
0	If the destination is not specified or its code is set to 0, a default destination will apply. In the Pira.cz products the default destination is Radiotext (RT1) without RT+.
1	Radiotext
3	Radiotext incl. RT+
4	Dynamic PS
5	Radiotext and Dynamic PS
7	Radiotext incl. RT+ and Dynamic PS
32 to 255	(The RDS encoder's manufacturer may define proprietary codes)

Other codes are reserved and should not be used.

Length limits

The Radiotext lenght is limited by RDS standard to 64 characters maximum. The same limit applies to RT+ (sum of all tags). The Dynamic PS text length is usually limited to 128 characters maximum. Overflowing characters are truncated by the RDS encoder.

List of tags supported in the text

The tags in the text are optional. Following table summarizes the tags supported and their RT+ equivalent:

Priority	Tag	RT+ equivalent
1	`artist`	Item.Artist
2	`title`	Item.Title
3	`album`	Item.Album
4	`comment`	Item.Comment
5	`genre`	Item.Genre
6	`news`	Info.News
7	`sport`	Info.Sport
8	`time`	Info.Date_Time
9	`weather`	Info.Weather
10	`traffic`	Info.Traffic
11	`ad`	Info.Advertisement
12	`url`	Info.Url
13	`info`	Info.Other
14	`short`	Stationname.Short
15	`long`	Stationname.Long
16	`now`	Programme.Now
17	`next`	Programme.Next
18	`host`	Programme.Host
19	`page`	Programme.Homepage
20	`phone`	Phone.Studio
21	`sms`	SMS.Studio
22	`email`	Email.Studio
23	`subchn`	Programme.Subchannel

All tags must be written in the format `<tag>`...`</tag>`.

Limit for number of tags inside the text

The RT+ specification does not explicitly state a limit for total number of tags inside the text. The RT+ coding rules however do not allow to mark more than two tags at one time. Numerous experiments have shown that sending more than two tags sequentially for the same Radiotext gives unpredictable results on various receivers. Additionally, due to the limit of 64 characters in total, the information frequently got truncated.
The X-Command accepts unlimited number of tags in the text. However only two tags at maximum are put into the RT+ service, the decision is made on the basis of the tag priority given above.

Timeout parameter

The timeout parameter is optional and can apply only to Radiotext or RT+ destination. Once included in the item element, it specifies a time for which the text is valid. The countdown begins when the RDS encoder receives the command. After the internal counter reaches zero, the text is cleared and default Radiotext is showed instead. Several input formats are accepted for the timeout, as given in the table:

Format	Description	Examples
h:m:s	Hours, minutes and seconds separated by colons.	\<tmo\>1:06:00\</tmo\>
m:s	Minutes and seconds separated by colon.	\<tmo\>5:26\</tmo\> \<tmo\>120:00\</tmo\>
s	Seconds.	\<tmo\>324\</tmo\>

Maximum number of seconds is 65535. Maximum number of minutes is 1092. Maximum number of hours is 18. Maximum total time must not exceed 18:12:15.

To avoid unwanted transitional effects, the timeout should not be used for tracks playing immediately behind or the timeout should be a bit longer in such a case. To disable the timeout for the current text, set the timeout to 0 or simply do not include the timeout at all inside the item element. A default text timeout will then apply (see the RDS encoder's documentation for details).

Text content comparison

To allow repeated receiving of the same item without unwanted effects (e.g. extra toggle of RT type), the RDS encoder shall not toggle RT type and RT+ item bit if the received item text represents the same Radiotext which is already being broadcast. In other words, the RDS encoder toggles the RT type and RT+ item bit only if the Radiotext changes.

The <tmr> element

Enables a timed loop of fixed text messages which were previously stored in the RDS encoder's non-volatile memory. The text messages are in the X-Command format. See the RDS encoder's documentation for details about how to store a list of text messages with the X-Command destination.

The element format

The <tmr> element format is as follows:

<rds><tmr>*timer_value*</tmr></rds>

The element parameters

Parameter	Description	Example
timer_value	Delay between the text messages in format m:s	<tmr>4:00</tmr>

The Timer is disabled when another application or component updates the Radiotext. Various automation effects may be created by placing the Timer into the RDS encoder's default text X-Command message or internal scheduler. First message in the list is loaded immediately upon launching the <tmr> element.

Note: The Seconds portion of the timer value is ignored in current implementation but must be entered to allow compatibility with future extensions.

The <msg> element

Processes a fixed text message which was previously stored in the RDS encoder's non-volatile memory. The text message is in the X-Command format.
See the RDS encoder's documentation for details about how to store a text message in X-Command format.

The element format

The <msg> element format is as follows:

<rds><msg>*message_index*</msg></rds>

The element parameters

Parameter	Description	Example
message_index	Message index.	<msg>12</msg>

Various automation effects may be created by placing the <msg> element into the RDS encoder's internal scheduler.

The <exe> element

Executes an ASCII command from the repertoire supported by the RDS encoder.

The element format

The <exe> element format is as follows:

<rds><exe>*ASCII_command*</exe></rds>

The element parameters

Parameter	Description	Example
ASCII_command	Any ASCII command accepted by the RDS encoder.	<exe>TA=1</exe>

The <exe> element is useful in two situations:
- when need to apply the ASCII command only on specific encoder in the network (see related <hd> element).
  For example: `XCMD=<rds><hd>123,5</hd><exe>TA=1</exe></rds>`
- when need to enter a text in UTF-8 encoding (for example PS).

The <hd> element

Adds an optional header to the command. This header defines site number and encoder number to which the command is being sent. Other encoders shall completely ignore the command, i.e. no response is returned and no action is taken.

The element format

The <hd> element format is as follows:

<rds><hd>*site_address,encoder_address*</hd>…</rds>

The element parameters

Parameter	Description	Example
site_address	Defines the site or group of sites to which the command is being sent: 0: all sites 1 to 1023: specific site or group of sites	<hd>123,5</hd>
encoder_address	Defines to which encoder(s) at a particular site the command is being sent: 0: all encoders at the site 1 to 63: specific encoder	

For a message to be acceptable to a particular encoder both the site address and the encoder address must be contained within the respective address lists of the encoder. These lists should be common with the UECP address lists so address lists set up for the UECP will apply to the X-Command as well.

Encapsulation in the UECP protocol

In some specific applications the X-Command may need to be encapsulated in the UECP protocol rather than to be sent directly. Typically such encapsulation may be required in existing UECP-based networks or if CRC protection of the datagram is necessary (serial channel over satellite).

For the intended purpose it is possible to use UECP Message Element Code 0x2D (Manufacturer / Transmission operator specific command). Following rules apply:

- The X-Command is supposed to be manufacturer independent so the manufacturer designation bytes are set to zero (0x00).
- The encapsulated X-Command starts with <rds> tag and ends with </rds> tag.
- Maximum length of the X-Command is 251 bytes (UECP restriction).
- Any text is UTF-8 encoded.
  Take care that national characters occupy more than 1 byte.
- The XCMD= prefix is omitted.
- The terminating character (CR) is omitted.
- Other message element is not allowed within the same UECP frame.

Final UECP message must be encoded (addressing, byte-stuffing, CRC etc.) according to the SPB 490 specification in [1].

Example

Let's assume following X-Command:

```
<rds><item><dest>1</dest><text>Hello World</text></item></rds>
```

Resulting UECP message in hexadecimal representation would be:

```
FE 00 00 00 42 2D 40 00 00 3C 72 64 73 3E 3C 69 74 65 6D 3E 3C 64 65
73 74 3E 31 3C 2F 64 65 73 74 3E 3C 74 65 78 74 3E 48 65 6C 6C 6F 20
57 6F 72 6C 64 3C 2F 74 65 78 74 3E 3C 2F 69 74 65 6D 3E 3C 2F 72 64
73 3E 8F 28 FF
```

---

[1] SPB 490 UECP Version 7.05. Geneva, RDS Forum, 2010.

The X-Command Examples

```
XCMD=<rds><item><text>This is a minimum format for the X-Command
item</text></item></rds>↵
```

Radiotext	This is a minimum format for the X-Command item
RT+	*Cleared*
Dynamic PS	*not affected*

```
XCMD=<rds><item><dest>7</dest><text>Now Playing:
<artist>Prodigy</artist> - <title>Full Throttle</title> (<album>Music
for the Jilted Generation</album>)</text></item></rds>↵
```

Radiotext	Now Playing: Prodigy - Full Throttle (Music for the Jilted Gener
RT+	🗣 Prodigy ♫ Full Throttle
Dynamic PS	Now Playing: Prodigy - Full Throttle (Music for the Jilted Generation)

```
XCMD=<rds><item><dest>3</dest><text>Now Playing: <artist>Julia Michaels
</artist> - <title>Issues</title></text><tmo>2:56</tmo></item></rds>↵
```

Radiotext	Now Playing: Julia Michaels - Issues
RT+	🗣 Julia Michaels ♫ Issues
Dynamic PS	*not affected*
Timeout	Sets RT and RT+ to default after 2:56.

```
XCMD=<rds><item><dest>3</dest><text><long>Radio National</long> - call
us: <phone>236-689-1122</phone></text></item></rds>↵
```

Radiotext	Radio National - call us: 236-689-1122
RT+	🔊 Radio National ☎ 236-689-1122
Dynamic PS	*not affected*

```
XCMD=<rds><item><dest>7</dest><text>Visit the website:
<url>http://myradio.com/</url></text></item></rds>↵
```

Radiotext	Visit the website: http://myradio.com/
RT+	ⓘ http://myradio.com/
Dynamic PS	Visit the website: http://myradio.com/

Conclusion
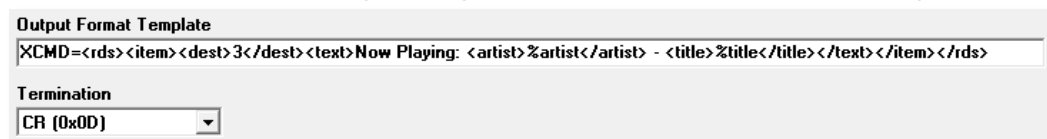
Recommended Steps for Broadcasters

- Contact your software vendor with questions related to the X-Command support in your broadcast automation system.

- **If your RDS encoder supports X-Command and your broadcast automation software meets the requirements stated below, you may put the RT+ on-air immediately.**

- Consider suppressing or entirely leaving dynamic PS usage. The dynamic PS has never been supported by RDS standard, therefore its behavior on different receivers is unpredictable and it's often confusing for the listener. The RT+ service is a good substitute which works well on many receivers currently available on the market.

Recommended Steps for Broadcast Software Developers

- The broadcast automation software should be able to establish a connection to the RDS encoder by either a serial COM port or via TCP/IP (the RDS encoder acts as a server).

- The software shall allow the serial port baudrate to be set at least in range of 1200 to 38400 bps (no parity, 8 data bits). The software shall implement a reconnection algorithm for the TCP/IP – check the connection status and reconnect if necessary before sending next data.

- Implement a direct support for the X-Command at least for the <artist> and <title> text elements, in the format defined in this document, incl. user selectable destination.
  - or -
  Allow the user to define output template for the data, like on this example:

- Keep the text actual by sending it once when the playing song changes or on similar event.

- Include the information in your documents and user guide for quick access.

X-Command History

Date	Event
25th of April 2017	Introduction of the X-Command, experimental implementation in the P232 RDS Encoder (fw version 2.1c)
24th of May 2017	Implementation of the X-Command in entire P132 RDS encoders family (P132, P232, P332, fw version 2.1d)
17th of Sept. 2018	Optional timeout tag added for Radiotext
6th of October 2018	<tmr> and <msg> tags added for sending various RT+ content in offline mode or as a default text (P132, P232, P332 fw version 2.1e)
17th of March 2019	<hd> tag added for selection of RDS encoder with specific address. <exe> tag added for direct execution of any ASCII command. (P132, P232, P332 fw version 2.1f)
30th of March 2019	UECP encapsulation added. (P132, P232, P332 fw version 2.1g) <tmr> and <tmo> elements time format specified.
1st of October 2020	The P164 RDS Encoder added to the list of supported devices.

Support & Newsgroup

Pira.cz Technical Forum: http://pira.cz/forum/index.php?board=8.0

Logo