ETSI TS 103 850 V1.1.1 (2022-10)

Reconfigurable Radio Systems (RRS);
Definition of Radio Application Package

Reference
DTS/RRS-0133

Keywords
radio, RAP, RRS

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 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Contents

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are 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 (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, 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.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Reconfigurable Radio Systems (RRS).

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**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.

1 Scope

This Technical Specification details the format of a Radio Application Package (RAP).

2 References

2.1 Normative 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 https://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.

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

[1] ETSI TS 103 436 (V1.2.1): "Reconfigurable Radio Systems (RRS); Security requirements for reconfigurable radios".

2.2 Informative 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.

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

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] ETSI EN 303 641 (V1.1.2): "Reconfigurable Radio Systems (RRS); Radio Equipment (RE) reconfiguration requirements".

[i.2] ETSI EN 303 648 (V1.1.2): "Reconfigurable Radio Systems (RRS); Radio Equipment (RE) reconfiguration architecture".

[i.3] ETSI EN 303 681-1 (V1.1.2): "Reconfigurable Radio Systems (RRS); Radio Equipment (RE) information models and protocols for generalized software reconfiguration architecture; Part 1: generalized Multiradio Interface (gMURI)".

[i.4] ETSI EN 303 681-2 (V1.1.2): "Reconfigurable Radio Systems (RRS); Radio Equipment (RE) information models and protocols for generalized software reconfiguration architecture; Part 2: generalized Reconfigurable Radio Frequency Interface (gRRFI)".

[i.5] ETSI EN 303 681-3 (V1.1.2): "Reconfigurable Radio Systems (RRS); Radio Equipment (RE) information models and protocols for generalized software reconfiguration architecture; Part 3: generalized Unified Radio Application Interface (gURAI)".

[i.6] ETSI EN 303 681-4 (V1.1.2): "Reconfigurable Radio Systems (RRS); Radio Equipment (RE) information models and protocols for generalized software reconfiguration architecture; Part 4: generalized Radio Programming Interface (gRPI)".

[i.7] ETSI EN 302 969 (V1.3.1): "Reconfigurable Radio Systems (RRS); Radio Reconfiguration related Requirements for Mobile Devices".

[i.8] ETSI EN 303 095 (V1.3.1): "Reconfigurable Radio Systems (RRS); Radio reconfiguration related architecture for Mobile Devices (MD)".

[i.9] ETSI EN 303 146-1 (V1.3.1): "Reconfigurable Radio Systems (RRS); Mobile Device (MD) information models and protocols; Part 1: Multiradio Interface (MURI)".

[i.10] ETSI EN 303 146-2 (V1.2.1): "Reconfigurable Radio Systems (RRS); Mobile Device (MD) information models and protocols; Part 2: Reconfigurable Radio Frequency Interface (RRFI)".

[i.11] ETSI EN 303 146-3 (V1.3.1): "Reconfigurable Radio Systems (RRS); Mobile Device (MD) information models and protocols; Part 3: Unified Radio Application Interface (URAI)".

[i.12] ETSI EN 303 146-4 (V1.1.2): "Reconfigurable Radio Systems (RRS); Mobile Device (MD) information models and protocols; Part 4: Radio Programming Interface (RPI)".

3 Definition of terms, symbols and abbreviations

3.1 Terms

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

administrative RAP metadata: information to help manage a resource, like resource type, permissions, and when and how it was created

descriptive RAP metadata: descriptive information about a resource, such as a target reconfiguration platform, a compiler, etc. used for discovery and identification and including elements such as title, abstract, author and keywords

NOTE: A Radio Application Package (RAP) contains metadata itself.

legal RAP metadata: information about the creator, copyright holder, and public licensing, if provided

metadata: data about the data, which can be structural or descriptive

NOTE: In the present document Metadata related to Descriptive RAP metadata, Structural RAP metadata, Administrative RAP metadata, Legal RAP metadata or a combination of some or all.

Radio Application (RA): software which enforces the generation of the transmit RF signals or the decoding of the receive RF signals

NOTE 1: The Software is executed on a particular radio platform or an RVM as part of the radio platform.

NOTE 2: RAs might have different forms of representation. They are represented as:

- Source codes including Radio Library calls of Radio Library native implementation and Radio HAL calls.
- IRs including Radio Library calls of Radio Library native implementation and radio HAL calls.
- Executable codes for a particular radio platform.

radio computer: part of Radio Equipment working under ROS control and on which RAs are executed

NOTE 1: A radio computer typically includes programmable processors, hardware accelerators, peripherals, software, etc. RF part is considered to be part of peripherals.

NOTE 2: The Radio Platform is the hardware part of the radio computer.

radio reconfiguration: reconfiguration of parameters related to air interface

Radio Virtual Machine (RVM): abstract machine which supports reactive and concurrent executions

NOTE: A RVM may be implemented as a controlled execution environment which allows the selection of a trade-off between flexibility of base band code development and required (re-)certification efforts.

reconfigurable equipment: part of a reconfigurable radio system

NOTE: The Reconfigurable Equipment is capable of being dynamically reconfigured to adapt to a wide range of communications conditions. Such reconfiguration may include the band of operation, the radio access technology, the associated networks and the services accessed. The reconfiguration may occur after initial sale deployment and operation.

Reconfigurable Radio System (RRS): radio systems encompassing Software Defined and/or Cognitive Radio Systems

structural RAP metadata: metadata about Radio Application Package structure, indicating how compound objects are put together, for example, where to find information fields related to Code, Security, etc. and describing the types, versions, relationships and other characteristics of digital materials

user: user of the Reconfigurable Radio System or the Reconfigurable Equipment

3.2 Symbols

Void.

3.3 Abbreviations

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

BE	Back End
HAL	Hardware Abstraction Layer
HW	HardWare
IR	Intermediate Representation
MDRC	Mobile Device Reconfiguration Class
OS	Operating System
RA	Radio Application
RAP	Radio Application Package
RC	Radio Controller
RE	Reconfigurable Equipment
RERC	Radio Equipment Reconfiguration Class
RF	Radio Frequency
ROS	Radio Operating System
RRS	Reconfigurable Radio System
RVM	Radio Virtual Machine
SFB	Standard Functional Block
UDFB	User Defined Functional Block
URA	Unified Radio Application

4 Usage of Radio Application Packages

4.1 Context

A Radio Application Package (RAP) is being used in order to provide new Radio Applications to a target radio equipment. As detailed in ETSI EN 303 648 [i.2] for radio equipment in general and in ETSI EN 303 095 [i.8] for the specific case of mobile devices, the RAP is used for distribution and installation of RA codes on the target reconfigurable REs. During the design time, the RA codes provider will generate a Radio Application Package (RAP) that includes metadata and RA codes. Note that the Radio Computer (RC) codes are part of the RA codes. In case that RC codes are executed in the non-real-time environment, they are compiled to be executed in a given Computational Resources before they are included in the RAP.

During the installation time, the RAP will be downloaded from a RadioApp Store and installed in the reconfigurable RE. The RA codes, including RC codes, and metadata included in the RAP are installed in the reconfigurable RE. Note that the RC codes are installed in the Computational Resources for operations that do not have to be executed in real time processing such as context information processing, while the Functional Block (SFBs & UDFBs) codes shall be installed in the radio computers to be processed in real-time.

The full chain is illustrated in Figure 1 for the case of general radio equipment as defined in ETSI EN 303 648 [i.2]:

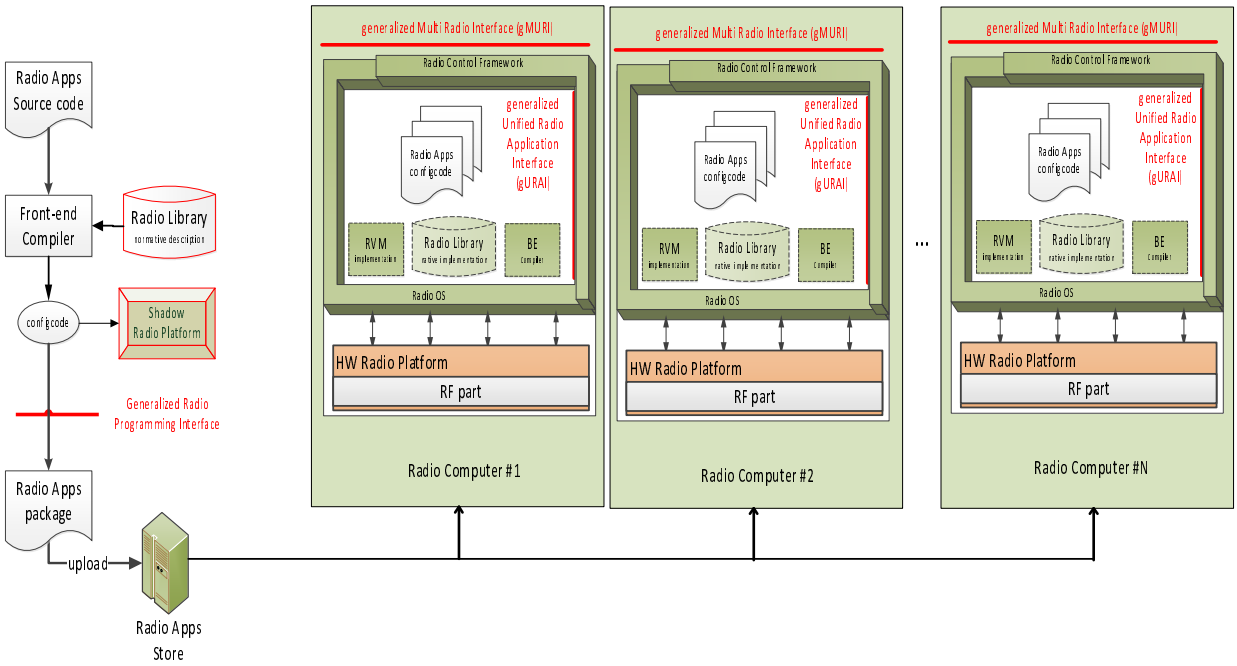

Figure 1: System architecture for radio computers where Radio Library and Back End (BE) compiler are included within the radio computers

The present document will define the structure of a RAP that shall be used for the software reconfiguration framework defined in [i.1] to [i.12].

4.2 Reconfiguration Classes

The information to be provided in a Radio Application Package (RAP) is dependent on the level of reprogrammability of the concerned platform. In the present document, it is thus referred to the Mobile Device Reconfiguration Classes (MDRC) as defined in ETSI EN 302 969 [i.7] for reconfigurable Mobile Devices and Radio Equipment Reconfiguration Classes (RERC) as defined in ETSI EN 303 641 [i.1] for any other reconfigurable equipment. A related summary is provided in Figures 2 and 3.

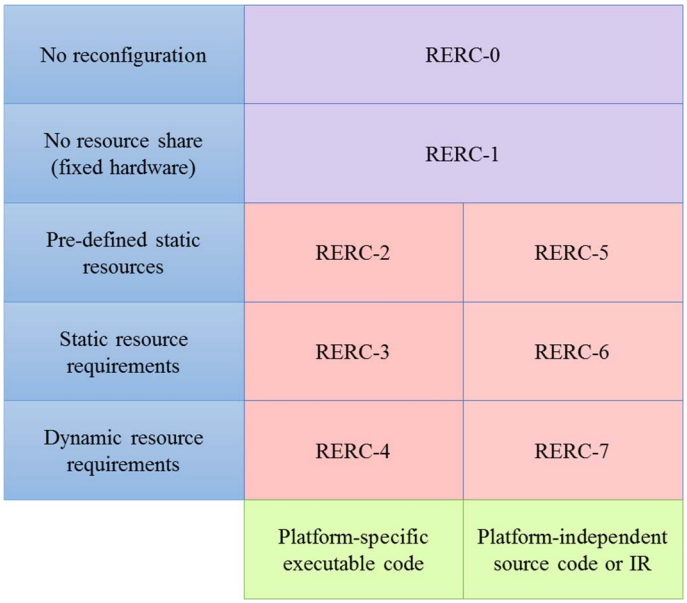

Figure 2: Definition of Radio Equipment Reconfiguration Classes (RERCs) according to reconfiguration capabilities [i.1]

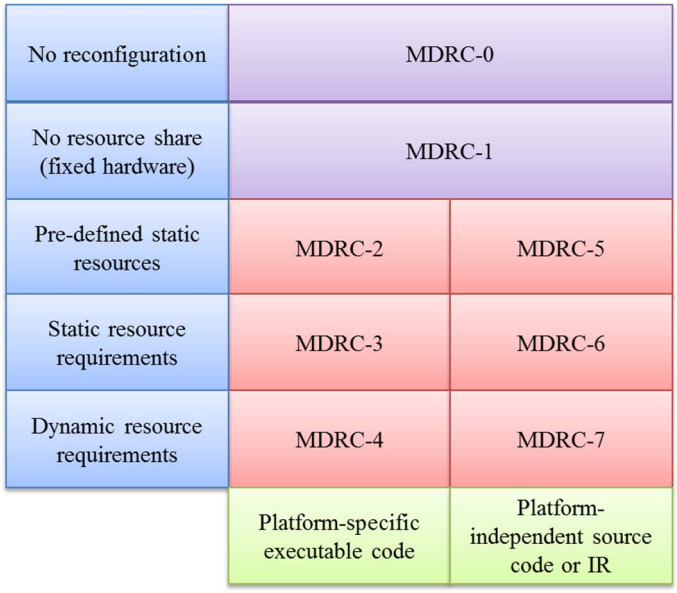

Figure 3: Definition of Mobile Device Reconfiguration Classes (MDRCs) according to reconfiguration capabilities [i.7]

5 Definition of Radio Application Packages

5.1 Content

In the present clause, the format of the RAP container is described on a high level. In order to keep the approach flexible, each of the information elements will be combined with a length indication (number of octets) such that manufacturers can adapt the size of any information element as required. The high-level RAP container format shall be defined as indicated in Figure 4.

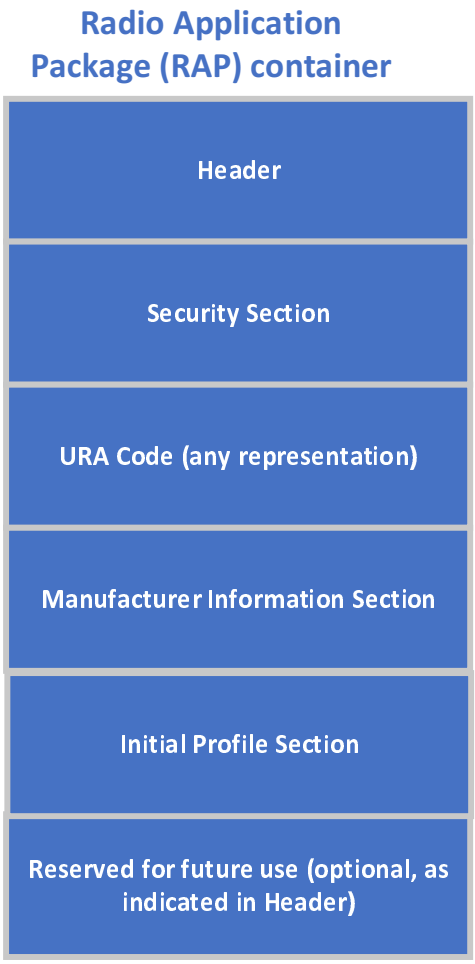

Figure 4: High level Radio Application Package container format

5.2 Tree Structure

5.2.1 Introduction

The RAP is a complex hierarchical bit field which is represented as a graph with specific properties. The RAP bit field is a sequence of bits. Therefore, the graph properties shall insure the only way for the RAP writing operation when the RAP is created as well as the RAP reading operation during which all RAP elements are decoded.

The RAP graph consists of nodes and edges between them. Each node of the RAP graph represents some bit field. Two nodes are connected by the edge if and only if one of them is a part of another one assuming that the one bit field is a part of another bit field. The RAP graph is a tree. It means that there is the only path between any pair of graph nodes. All the RAP graph elements are ordered in the following way:

- From the top to the bottom, the RAP graph is structured by layers. Nodes from the upper layer are connected with nodes from the adjusted lower layer. There are no edges between nodes from the same layer. If there is an edge between the upper layer node and the lower layer node it means that the bit field represented by the lower layer node is a part of the bit field represented by the upper layer node.
- From the left-hand to the right-hand, nodes from the same layer have fixed layer positions. It defines a position of the RAP element in the RAP bit field sequence: the left-hand elements are placed before the right-hand RAP element.

Below the RAP graph nodes are denoted by circles and edges by lines. Some of the RAP elements can be optional. In such case corresponding nodes can be omitted in the RAP graph structure and related edges are drawn as dotted lines. They also will be labelled by the letter "o" ("optional") or by a particular condition defining the related option. All other edges are drawn by the solid lines.

5.2.2 Top Level

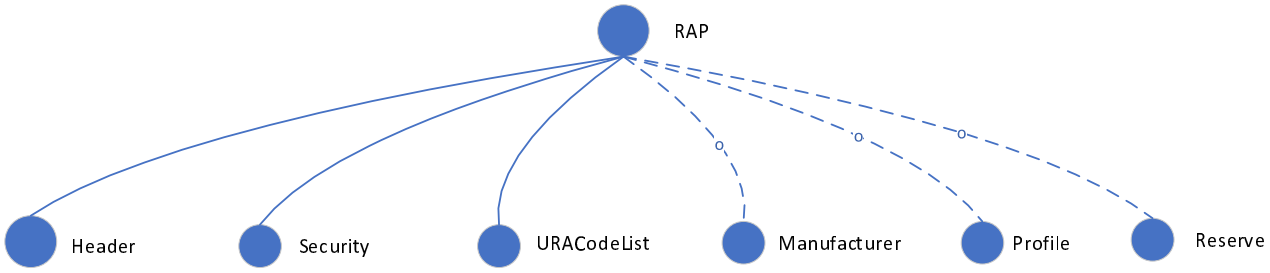

Figure 5: Top Level Tree Structure

The highest layer of the RAP graph consists of the only one the RAP node which represents the whole RAP bit field as it is pointed out in Figure 5 The lower layer consists of the following nodes representing particular RAP sections from Figure 4.

Table 1: Top Level Tree Structure Bit Fields

N	Node name	Node type	Bit field
1	Header	mandatory	The RAP header
2	Security	mandatory	The Security section
3	URACodeList	mandatory	URA codes (any representation). The URA code list consisting of code sections for a few Radio Applications
4	Manufacturer	optional	The Manufacturer Information section
5	Profile	optional	The Initial Profile section
6	Reserve	optional	Reserved for the future use

5.2.3 Tree - Header Section

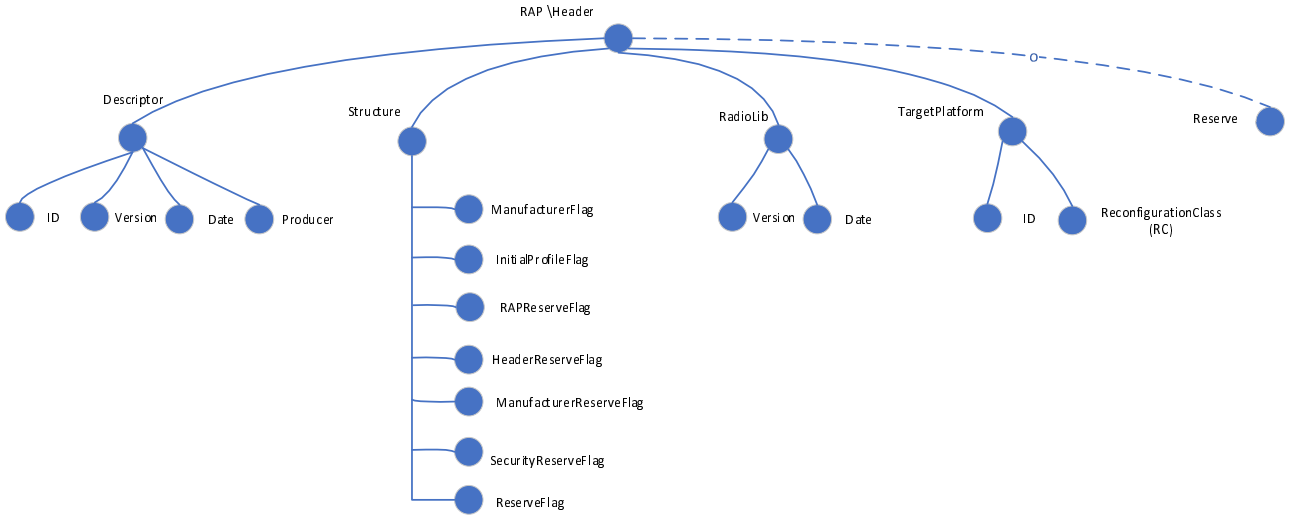

Figure 6: Tree - Header Section

The RAP Header subtree is depicted in Figure 6 Elements of the RAP Header are described in Table 2.

Table 2: Header Section Bit Fields

N	Node name	Node type	Bit field
1	Descriptor	mandatory	The RAP header descriptor
2	Descriptor\ID	mandatory	The RAP ID
3	Descriptor\Version	mandatory	The RAP version
4	Descriptor\Date	mandatory	The RAP day issue
5	Descriptor\Producer	mandatory	The RAP producer
6	Structure	mandatory	The bit field for the RAP structure description
7	Structure\ManufacturerFlag	mandatory	ManufacturerFlag = 1 than Manufacturer is a part of the RAP and otherwise if ManufacturerFlag = 0
8	Structure\InitialProfileFlag	mandatory	InitialProfileFlag = 1 than Profile is a part of the RAP and otherwise if InitialProfileFlag = 0
9	Structure\RAPReserveFlag	mandatory	RAPReserveFlag = 1 than Reserve is a part of the RAP and otherwise if RAPReserveFlag = 0
10	Structure\HeaderReserveFlag	mandatory	HeaderReserveFlag = 1 than Header\Reserve is a part of the RAP and otherwise if HeaderReserveFlag = 0
11	Structure\ManufacturerReserveFlag	mandatory	ManufacturerReserveFlag = 1 than Manufacturer\Reserve is a part of the RAP and otherwise if ManufacturerReserveFlag = 0
12	Structure\SecurityReserveFlag	mandatory	SecurityReserveFlag = 1 than Security\Reserve is a part of the RAP and otherwise if SecurityReserveFlag = 0
13	Structure\ReserveFlag	mandatory	ReserveFlag = 1 than Reserve is a part of the RAP and otherwise if ReserveFlag = 0
14	RadioLib	mandatory	The Radio Lib description
15	RadioLib\Version	mandatory	The Radio Lib version
16	RadioLib\Date	mandatory	The Radio Lib day issue
17	TargetPlatform	mandatory	The target platform description
18	TargetPlatform\ID	mandatory	The target platform ID
19	TargetPlatform\ReconfigurationClass	mandatory	The target platform Reconfiguration Class (see Figures 2 and 3)
20	Reserve	mandatory	The RAP reserved bit field

5.2.4 Tree - Security Section

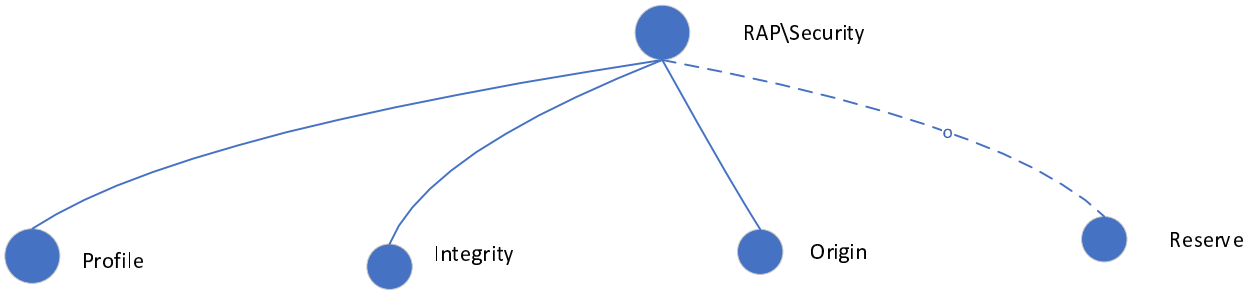

Figure 7: Tree - Security Section

The Security section subtree is depicted in Figure 7. Elements of the Security section are described in Table 3.

Table 3: Security Section Bit Fields

N	Node name	Node type	Bit field
1	Profile	mandatory	Description of security profile. Indication of the security features implemented for the specific case (taken out of the identified list of options available in the RRS security study ETSI TS 103 436 [1])
2	Integrity	mandatory	Parameterization for Integrity Protection: Hash Algorithm, Hash size, etc.
3	Origin	mandatory	Parameterization for Proof of Origin: Digital signature algorithm, digital signature size, etc.
4	Reserve	optional	Reserved for the future use

5.2.5 Tree - Unified Radio Application (URA) Code Section

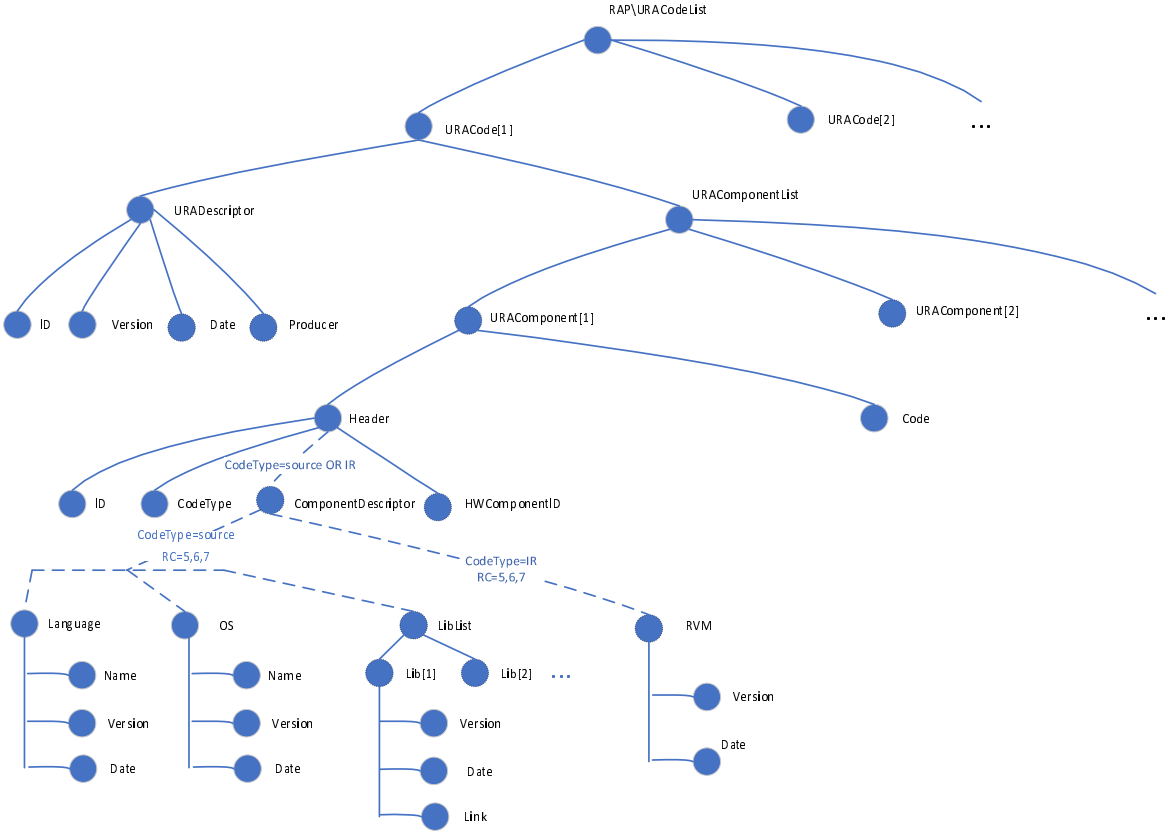

Figure 8: Tree - Unified Radio Application (URA) Code Section

The URA Code section subtree is depicted in Figure 8. Elements of the URA Code section are described in Table 4.

Table 4: Code Section Bit Fields

N	Node name	Node type	Bit field
1	URACode[k]	mandatory	The URA code section for Radio Application k. The RAP can provide program code for a few Radio Applications in one package
2	URACode[k]\URADescriptor	mandatory	The descriptor of URA code section
3	URACode[k]\URADescriptor\ID	mandatory	The URA k ID
4	URACode[k]\URADescriptor\Version	mandatory	The URA k version
5	URACode[k]\URADescriptor\Date	mandatory	The URA k day issue
6	URACode[k]\URADescriptor\Producer	mandatory	The URA k producer
7	URACode[k]\URAComponentList	mandatory	The URA component list for particular Radio Application
8	URACode[k]\URAComponentList\ URAComponent[n]	mandatory	Description of a particular URA component n for a particular Radio Application k
9	URACode[k]\URAComponentList\ URAComponent[n]\Header	mandatory	The header of a particular URA component
10	URACode[k]\URAComponentList\ URAComponent[n]\Code	mandatory	Code section of a particular URA component
11	URACode[k]\URAComponentList\ URAComponent[n]\Header\ID	mandatory	Code component ID
12	URACode[k]\URAComponentList\ URAComponent[n]\Header\CodeType	mandatory	Code type of the code component. CodeType = "source" for source code, CodeType = IR for intermedia representation (config code), CodeType = "bin" for executable machine code
13	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor	optional	CodeType = "source" or "IR". URA component code description

N	Node name	Node type	Bit field
14	URACode[k]\URAComponentList\ URAComponent[n]\Header\ HWComponentID	mandatory	HW component ID on which the component code is mapped
15	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\Language	optional	CodeType = "source" & RC ∈ {5, 6, 7}, language specification for source code of the URA component
16	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\OS	optional	CodeType = "source" & RC ∈ {5, 6, 7}, Operating System specification for source code of the URA component
17	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\LibList	optional	CodeType = "source" & RC ∈ {5, 6, 7}, specification of libraries used for source code compilation of the corresponding URA component
18	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\Language\ID	mandatory	The language ID for specification of the URA component
19	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ Language\Version	mandatory	The language version for specification of the URA component
20	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ Language\Date	mandatory	The language date issue for specification of the URA component
21	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\OS\ID	mandatory	The OS ID for specification of the URA component
22	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ OS\Version	mandatory	The OS version for specification of the URA component
23	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ OS\Date	mandatory	The OS date issue for specification of the URA component
24	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\LibList\Lib[m]	optional	CodeType = "source" & RC ∈ {5, 6, 7}, specification of libraries used for source code compilation of the corresponding URA component
25	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ LibList\Lib[m]\ID	mandatory	The library m ID used for compilation of the URA component
26	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ LibList\Lib[m]\Version	mandatory	The library m version for compilation of the URA component
27	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ LibList\Lib[m]\Date	mandatory	The library m date issue for compilation of the URA component
28	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ LibList\Lib[m]\Link	mandatory	The link to the location of the library m
29	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\RVM	optional	CodeType = "IR" & RC ∈ {5, 6, 7}, RVM specification for execution of IR of the URA component
30	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ RVM\Version	mandatory	The RVM version for execution of IR of the URA component
31	URACode[k]\URAComponentList\ URAComponent[n]\Header\ ComponentDescriptor\ RVM\Date	mandatory	The RVM date issue for execution of IR of the URA component

5.2.6 Tree - Manufacturer Information Section

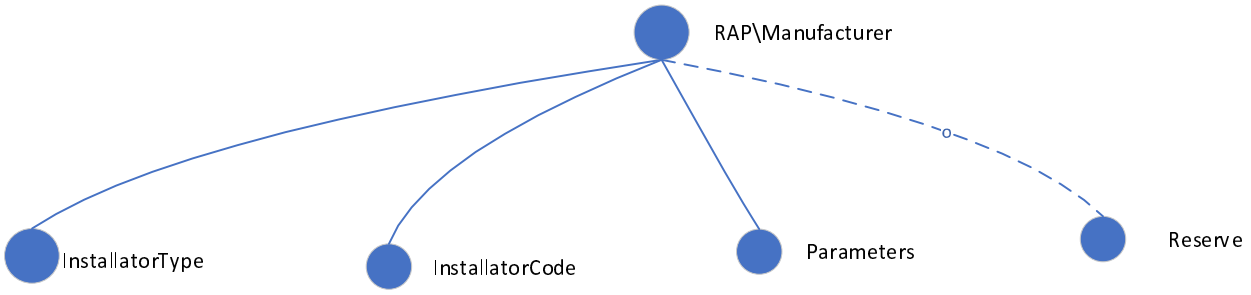

Figure 9: Tree - Manufacturer Information Section

The Manufacturer Information section subtree is depicted in Figure 9. Elements of the Manufacturer Information section are described in Table 5.

Table 5: Manufacturer Information Section Bit Fields

N	Node name	Node type	Bit field
1	InstallatorType	mandatory	E.g. cmake, make, etc. Some script, some specific installator for complex code (installator does everything), etc.
2	InstallatorCode	mandatory	Executable code for the Installator
3	Parameters	mandatory	Installation parameters
4	Reserve	optional	Reserved for the future use

5.2.7 Tree - Initial Profile Section

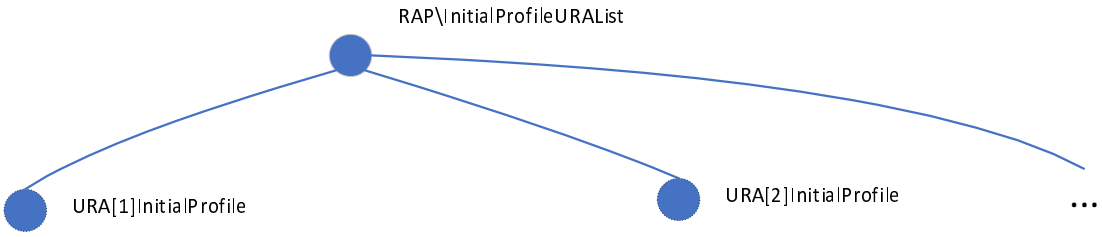

Figure 10: Tree - Initial Profile Section

The Initial Profile section subtree is depicted in Figure 10. It represents a list of Initial Profiles for each URA from the RAP. Each URA Initial Profile is a bit field which internal structure is recognized by a particular URA.

5.3 Internal Structure

5.3.1 Introduction and conventions

Within the RAP Top Level Tree structure introduced in clause 5.2.1, the structures of the various information fields are further detailed in the present clause. The definition of the information elements follows in clause 5.2.2 to 5.2.7.

NOTE: In the graphical representations in this clause, the structures of the package are presented vertically for better readability.

The RAP and all internal elements of the RAP are represented by the bit fields which shall have one of the following types:

- Static bit field. It has the predefined size and consists of only data. The size of the RAP bit field Size(BitField) is defined as the number of its bits. The static fields will be labelled by the symbol "s". The format of the static bit field is the following: *{Data}*.

- Dynamic bit field. It has variable data size and consist of two parts where the one part is the explicit data size value and the second part is data field itself. The size of the data size subfield is static. The format of the dynamic bit field is the following: *{Size | Data}*.
- List bit field. A list of any type bit fields with implicit size. Each bit field from the list is followed by the padding bit which point out on the next bit field: if PaddingBit = 1 than the next bit field is an element of the list and otherwise if PaddingBit = 0. Any dynamic, static or list bit field can be an element of the list bit field. The format of the list bit field is the following: *{PaddingBit | Data|…}*.
- Complex bit field. It is any combination of any type bit fields.

Bit fields will use the following graphical notation for mandatory and optional fields depicted in Figure 11.

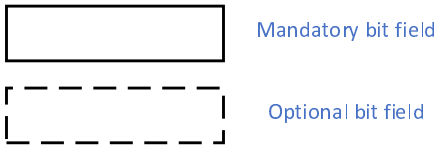

Figure 11: Mandatory and Optional bit fields

The RAP bit field is a complex bit field and its structure is depicted in Figure 12. It is represented by rectangles which follow each other and create the sequence of rectangles. Each rectangle exposes the sequence of bit consisting of related RAP bit filed. The leftmost bit in the RAP field is the least significant bit. It has 0 number and the rightmost bit in the RAP bit filed is the most significant bit with the number Size(BitField) -1. Despite the different size of the RAP fields for clarity they will be represented by rectangles of the same size. The size of the complex field is calculated as the sum of sizes of all bit fields included in the complex field.

The RAP includes mandatory fields such as RAP Header, Security Section, URA Code Section and optional fields:

- Manufacturer Info Section, if Manufacturer Info Flag = 1.
- Initial Profile, if Initial Profile Flag = 1.
- Reserve, if Reserve Flag = 1.

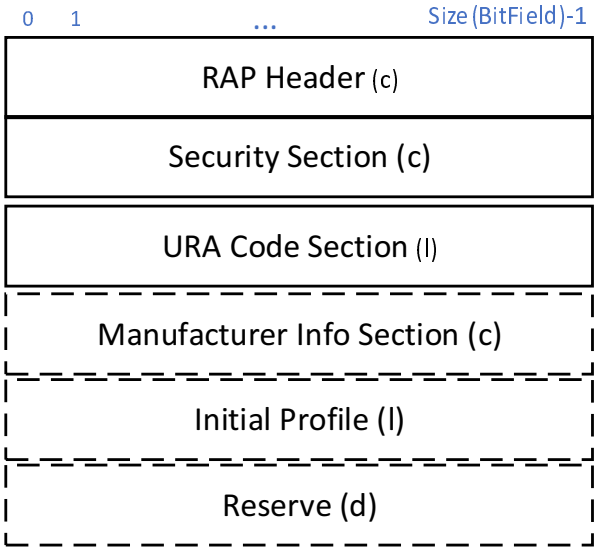

Figure 12: The RAP Bit Field

5.3.2 Radio Application Package (RAP) Header

The Radio Application Package Header consists of a sequence of static fields which size is defined in Table 6, clause 5.4. The format the sequence is presented in Figure 13 and the structure is described by the graph in Figure 12. Only the Reserve field is the dynamic field. It is optional field and it takes in place when Header Reserve Flag = 1. In the opposite case it will be omitted. Other bit fields are mandatory fields: RAP Descriptor, RAP Structure Descriptor, Radio Lib and Target Platform Descriptor. The RAP Descriptor consists of the following static fields which size is pointed out in Table 6 (rows #23 - 26), clause 5.4: RAP ID, RAP Version, RAP Date, RAP Producer ID. The size of the RAP Structure Descriptor is depicted in Table 6 (row #8), clause 5.4. It consists of multiple 1-bit flags which define the structure of the RAP by describing conditions for optional RAP bit fields: ManufacturerInfoFlag, InitialProfileFlag, RAPReserveFlag, HeaderReserveFlag, ManufacturerReserve, SecurityReserve, ReserveFlag. The Radio Lib Descriptor consists of the static fields the Radio Lib Version and Radio Lib Date which size is shown in Table 6 (rows #34, 35), clause 5.4. The Target Platform Descriptor consists of the following static fields: the Target Platform ID and the MDRC/RERC. Their size is pointed out in Table 6 (rows #36, 37), clause 5.4.

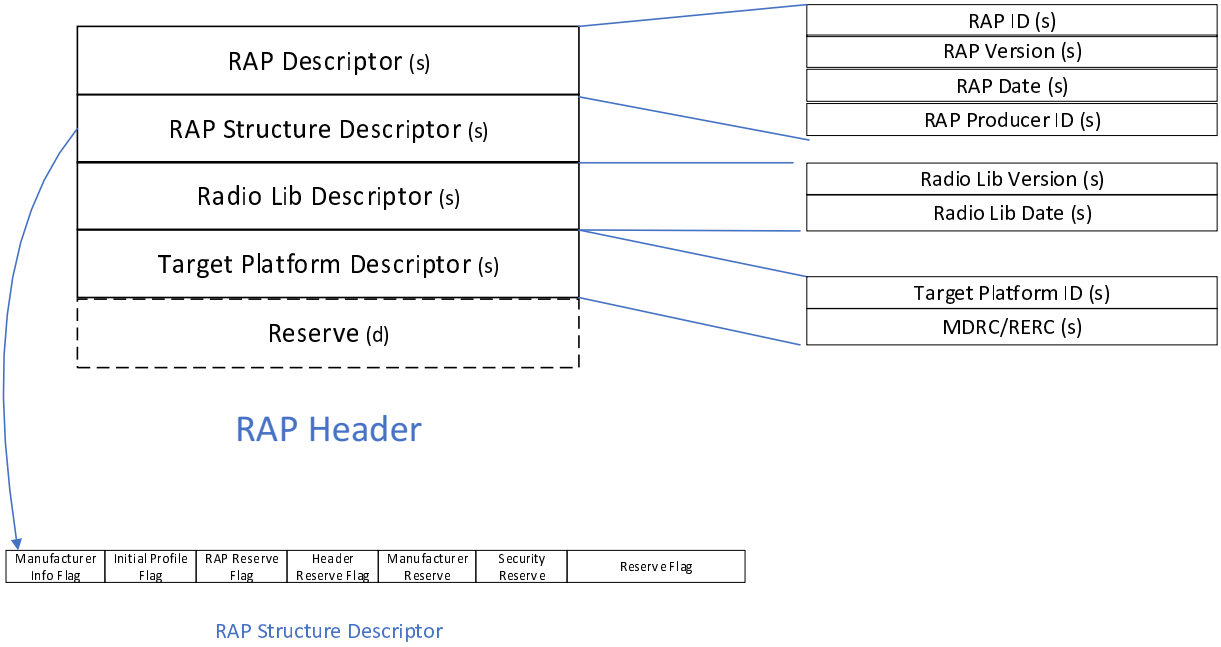

Figure 13: Radio Application Package (RAP) Header Structure

5.3.3 Security Section

Following a risk analysis, ETSI TS 103 436 [1] introduces security requirements for reconfigurable radio systems. The Security Section includes the following information elements:

i) a Security Profile (d) which details the structure of the security section and includes security information available in the reserve part of the security section (i.e. security information beyond the mandatory elements);

ii) the mandatory information on Integrity Protection in accordance to clause 7.2 of ETSI TS 103 436 [1];

iii) the mandatory information on Proof of Origin in accordance to clause 6.5 of ETSI TS 103 436 [1];

iv) the Reserve part which includes any additional security features as defined in ETSI TS 103 436 [1] and as listed in the Security Profile part.

Note that ETSI TS 103 436 [1] is limited to a single code point in its current release. Introducing multiple code points will require a corresponding extension.

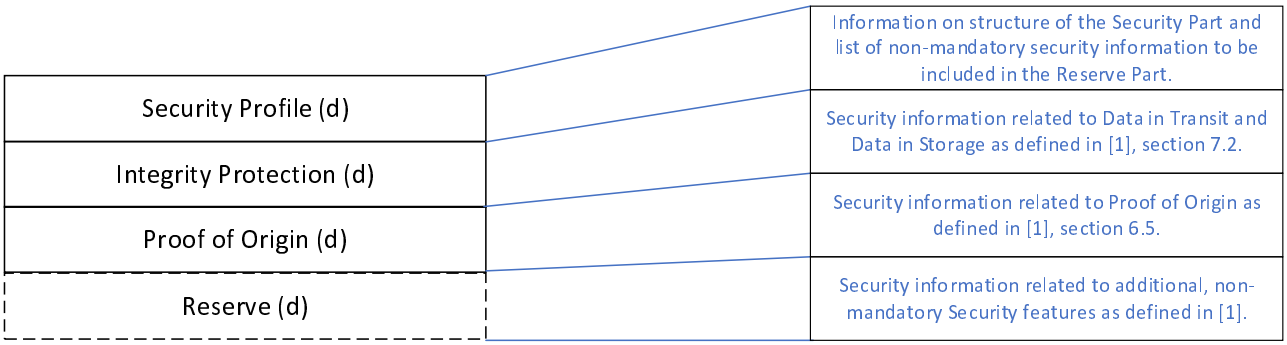

Figure 14: Security Section

5.3.4 Unified Radio Application (URA) Code Section

The Unified Radio Application Package Code Section consists of the list of particular URA subsection which format is depicted in Figure 14 and the structure is explained in clause 5.2.2.

Each URA[k] subsection, for some k = 0, 1,…, consists of the static field URA Descriptor and the list of URA components. The URA Descriptor consists of the following static fields: URA ID, URA Version, URA Date, URA Producer ID. The size of these fields is pointed out in Table 6 (rows #41, 42, 43, 44), clause 5.4.

The URA component list provides information about particular URA[k] components which are the subject for reconfiguration. Each component from the URA componet list is described by the URA Component Header and the URA Component Code. The last one is a dynamic bit field and contains the software code of particular component. The URA Component Header provides all needed information to interprete correctly the software component code. It contains the following static bit fields: URA Component ID, URA Component Code Type and HW Component ID. The size of these bit field is pointed out in Table 6 (rows #48, 49, 51), clause 5.4. The URA Component Descriptor is an optional part of the URA Component Header. If the CodeType = IR and MDRC ∈ {5, 6, 7} than the URA Component Descriptor consists of only the RVM Descriptor. If the Code Type = «source», i.e. the component code is a source code, and MDRC ∈ {5, 6, 7 than the URA Component Descriptor consists of the Language Descriptor, OS Descriptor and the list of the Lib Descriptors.

The RVM Descriptor is a static bit field consisting of the static fields the RVM Version and the RVM Date. Their sizes are pointed out in Table 6 (rows # 67, 68), clause 5.4.

The Language Descriptor consists of three static bit fields: the Language Name, the Language Version and the Language Date. Their sizes are pointed out in Table 6 (rows #56, 57, 58), clause 5.4.

The OS Descriptor also consists of three static bit fields: the OS Name, the OS Version and the OS Date. Their sizes are pointed out in Table 6 (rows #59, 60, 61), clause 5.4.

The Lib Descriptors forms the List of Lib Descriptors. Each Lib Descriptor consists of three static fields Lib Name, the Lib Version and the Lib Date which size is presented in Table 6 (rows #63, 64, 65), clause 5.4. It also includes the Lib Link which is the dynamic field.

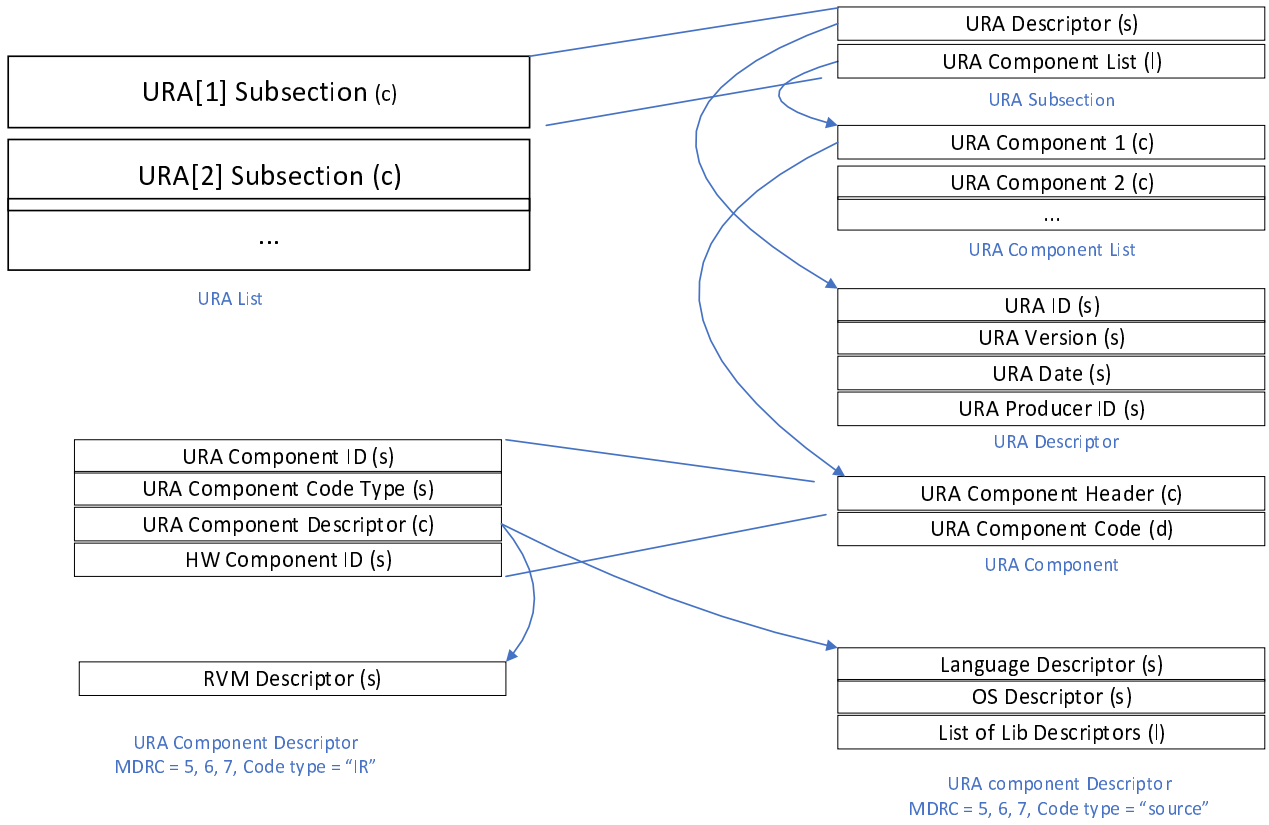

Figure 15: Unified Radio Application (URA) Code Section

Figure 16: Radio Application Package (RAP) Descriptors

5.3.5 Manufacturer Information Section

The Manufacturer Information Section is the optional bit field and it takes place when the ManufacturerInfoFlag = 1. It consists of three mandatory bit fields: InstallatorType, InstallatorCode and InstallationParameters. The InstallatorType is the static field which size is pointed out in Table 6 (row #17), clause 5.4. The InstallatorCode and the InstallationParameters are the dynamic bit fields. The structure and size of these fields are defined by a particular manufacturer. The Reserve field is the optional field which take place if ManufacturerReserveFlag = 1. It is the dynamic bit field. The value of this flag as well as the structure and size of the Reserve field are defined by a manufacturer.

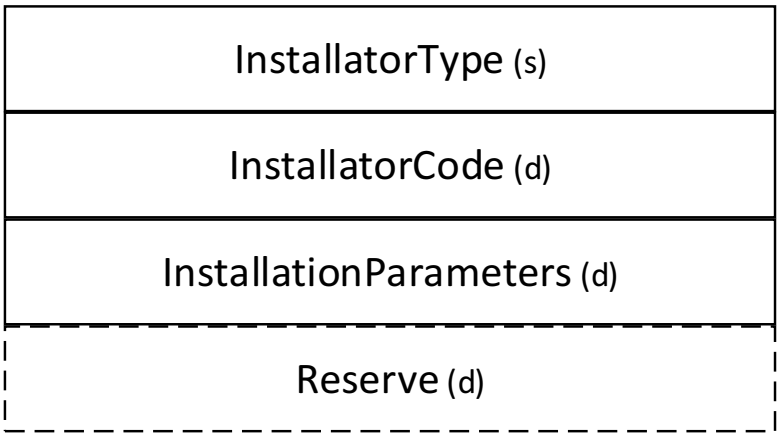

Figure 17: Manufacturer Information Section

5.3.6 Initial Profile Section

The Initial Profile Section is the optional field which take place when InitialProfileFlag = 1. It represents the list of initial profiles of particular URAs: URA[k] Initial Profile, for k = 1, 2,… . Each URA[k] Initial Profile, for some k, is the dynamic field which structure and size are defined by a particular manufacturer.

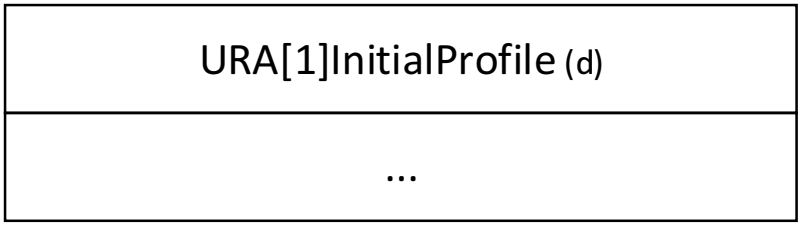

Figure 18: Initial Profile Section

5.4 Format

The present clause introduces the detailed information contained in the high level Radio Application Packet structure as defined above.

Table 6 summarizes information about bit fields of the RAP. The column "Path\Node Name" navigates the way from the root node which is "RAP" to the target node with corresponding "Node Name". This node represents a bit field characterized in the next columns:

- Column "Type" provides information about the type of the bit fields which might be static (s), dynamic (d), list (l) or complex (c).
- Column "Opt" indicates optional property of the bit field which might be mandatory (m) or optional(o).
- Column "Size" provides information about the size of the bit field if they are static and information about the size of the subfield "Data Size" in case of dynamic bit fields. In case that no value is provided, it is a manufacturer choice to determine the appropriate size.

Table 6: RAP Fields

N	Path\Node Name	RAP Element	Type	Opt	Size (bits)
0	RAP	RAP	c	M	
1	RAP\header	RAP Header	c	M	
2	RAP\code	URA Code Section	c	M	
3	RAP\security	Security Section	c	M	
4	RAP\manufacturer	Manufacturer Infor Section	c	O	
5	RAP\initial_profile	Initial Profile	c	O	

N	Path\Node Name	RAP Element	Type	Opt	Size (bits)
6	RAP\reserve	Reserve	d	O	16
7	RAP\header\descriptor	RAP Descriptor	s	M	128
8	RAP\header\structure	RAP Structure Descriptor	s	M	8
9	RAP\header\RadioLib	Radio Lib Descriptor	s	M	56
10	RAP\header\TargetPlatform	Target Platform Descriptor	s	M	16
11	RAP\header\reserve	RAP Header Reserve	d	O	8
12	RAP\code\URACodeList	URA Code List	l	M	
13	RAP\security\profile	Security Profile	d	M	8
14	RAP\security\integrity	Integrity Protection	d	M	8
15	RAP\security\origin	Proof of Origin	d	M	8
16	RAP\security\reserve	Security Reserve	d	O	8
17	RAP\manufacturer\InstallatorType	Installator Type	s	M	8
18	RAP\manufacturer\InstallatorCode	Installator Code	d	M	64
19	RAP\manufacturer\parameters	Installation Parameters	d	M	16
20	RAP\manufacturer\reserve	Manufacturer Info Reserve	d	O	16
21	RAP\initial_profile\URAList	URA Initial Profile List	l	M	
22	RAP\initial_profile\URAList\URA[1]InitialProfile	Initial profile of URA[1]	d	M	
23	RAP\header\descriptor\id	RAP ID	s	M	64
24	RAP\header\descriptor\version	RAP Version	s	M	32
25	RAP\header\descriptor\date	RAP Date	s	M	24
26	RAP\header\descriptor\producer	RAP Producer ID	s	M	8
27	RAP\header\structure\ManufacturerFlag	Manufacturer Info Flag	s	M	1
28	RAP\header\structure\InitialProfileFlag	Initial Profile Flag	s	M	1
29	RAP\header\structure\RAPReserveFlag	RAP Reserve Flag	s	M	1
30	RAP\header\structure\HeaderReservFlag	RAP Header Reserve Flag	s	M	1
31	RAP\header\structure\ManufacturerReserve	Manufacturer Info Reserve Flag	s	M	1
32	RAP\header\structure\SecurityReserve	Security Reserve Flag	s	M	1
33	RAP\header\structure\ReserveFlags	Reserve Flags	s	M	2
34	RAP\header\RadioLib\Version	Radio Lib Version	s	M	32
35	RAP\header\RadioLib\Date	Radio Lib Date	s	M	24
36	RAP\header\TargetPlatform\ID	Target Platform ID	s	M	8
37	RAP\header\TargetPlatform\ReconfigurationCode	MDRC/RERC	s	M	8
38	RAP\code\URACodeList\URACode[1]	URA[1] Subsection	c	M	
39	RAP\code\URACodeList\URACode[1]\URADescriptor	URA Descriptor	s	M	128
40	RAP\code\URACodeList\URACode[1]\URAComponentList	URA Component List	l	M	
41	RAP\code\ URACodeList\URACode[1]\ URADescriptor\ID	URA ID	s	M	64
42	RAP\code\ URACodeList\URACode[1]\ URADescriptor\Version	URA Version	s	M	32
43	RAP\code\ URACodeList\URACode[1]\ URADescriptor\Date	URA Date	s	M	24
44	RAP\code\ URACodeList\URACode[1]\ URADescriptor\producer	URA Producer ID	s	M	8
45	RAP\code\URACodeList\ URACode[1]\ URAComponentList \URAComponent[1]	URA Component 1	c	M	
46	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ Header	URA Component Header	c	M	
47	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ Code	URA Component Code	d	M	64
48	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ID	URA Component ID	s	M	64

N	Path\Node Name	RAP Element	Type	Opt	Size (bits)
49	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \CodeType	URA Component Code Type	s	M	8
50	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor	URA Component Descriptor	c	M	
51	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \HWComponentID	HW Component ID	s	M	8
52	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\Language	URA Component Language Descriptor	s	M	64
53	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\OS	URA Component OS Descriptor	s	M	64
54	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\LibList	List of Lib Descriptors	l	M	
55	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\RVM	URA Component RVM Descriptor	s	M	56
56	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\Language\Name	URA Component Language Name	s	m	8
57	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\Language\Version	URA Component Language version	s	M	32
58	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\Language\Date	URA Component Language Date	s	M	24
59	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\OS\Name	URA Component OS Name	s	M	8
60	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\OS\Version	URA Component OS version	s	M	32
61	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\OS\Date	URA Component OS Date	s	M	24
62	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\LibList\Lib[1]	Lib Descriptor	c	M	
63	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\LibList\Lib[1]\Name	URA Component Lib Name	s	M	64
64	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\LibList\Lib[1]\Version	URA Component Lib version	s	M	32
65	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\LibList\Lib[1]\Date	URA Component Lib Date	s	M	24
66	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\LibList\Lib[1]\Link	URA Component Lib Link	d	M	16
67	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\RVM\Version	URA Component RVM Version	s	M	32
68	RAP\code\URACodeList\ URACode[1]\URAComponentList\URAComponent[1]\ header \ComponentDescriptor\RVM\Date	URA Component RVM Date	s	m	24

History

Document history		
V1.1.1	October 2022	Publication