THIS DOCUMENT IS DEPRECATED AND SHOULD NOT BE USED FOR NEW PRODUCTION SYSTEMS

REFER TO ETSI TS 102 818 (latest version)

SEE https://radiodns.org/technical

RadioEPG Technical Specification

REPG0 V1.0.0 (2 12-04)

The RadioDNS Project

http://radiodns.org/

feedback@radiodns.org

Important notice

If you find errors in the present document, please send your comment to the document editors
feedback@radiodns.org

Copyright Notification

Contents

Intellectual Property Rights

The contributors to the present document have declared that they have no IPRs in respect of their contributions. However no investigation, including IPR searches, has been carried out, so no guarantee can be given as to the existence of other IPRs which are, or may be, or may become, essential to the present document.

Foreword

Modern connected radio devices have the ability to use IP and broadcast technologies together to provide a compelling user experience that combines the best of both.

The RadioEPG application provides a methodology for clients receiving a service to acquire additional metadata, including service and schedule information.

It enables devices to be more service-oriented, rather than platform-oriented, and provides details of the different ways in which a service may be received, allowing the client to choose the most appropriate. This enables a device to implement *Service Following*, where it will select between receiving the service over Internet Streaming or Broadcast as the conditions change – an implementation of a **Hybrid Radio**.

This specification has been based upon the existing DAB EPG XML specification [4]. This is in order to ease adoption for device manufacturers and service providers who may already have some familiarity with it.

Scope

The present document defines the protocol for RadioEPG to allow implementation from both a service provider and client perspective.

References

The following documents contain provisions, which through reference in this text constitute provisions of the present document.

- References are either specific (identified by date of publication and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] RFC 2782 (2000): 'A DNS RR for specifying the location of services (DNS SRV)'

[2] RDNS01: 'RadioDNS Technical Specification'

[3] RFC 2616 (1999): 'Hypertext Transfer Protocol – HTTP/1.1'

[4] ETSI TS 102 818: 'Digital Audio Broadcasting (DAB); Digital Radio Mondial (DRM); XML Specification for Electronic Programme Guide (EPG)'

[5] IEC 62106 (2009): 'Specification of the Radio Data System (RDS) for VHF/FM sound broadcasting in the frequency range from 87,5 MHz to 108,0 MHz'

[6] ISO 3166-1, 'Codes for the representation of names of countries and their subdivisions – Part 1: Country codes'

[7] NRSC-5-B:2008, 'In-band/on-channel Digital Radio Broadcasting Standard'

[8] RFC 2046 (1996): 'Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types'

[9] RFC 1738 (1994): 'Uniform Resource Locators (URL)'

Definitions and abbreviations

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

RadioDNS Centralised lookup for radio services, allowing the resolution of broadcast parameters to an authoritative FQDN as detailed in the RadioDNS specification [2].

service A service such as a talk or music radio station

service provider The organisation providing a service

client A device or application for receiving and listening to a service, or an implementation of a directory service, unless explicitly differentiated

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

AMSS	Amplitude Modulation Signalling System
CNAME	DNS Canonical Name record
DAB	Digital Audio Broadcasting
DNS	Domain Name System
DRM	Digital Radio Mondiale
EPG	Electronic Programme Guide
FCC	Federal Communications Commission (of the United States of America)
FQDN	Fully Qualified Domain Name
IP	Internet Protocol
NS	Name Server
RBDS	Radio Broadcast Data System
RDS	Radio Data System
SRV	DNS Service record
VHF/FM	Very High Frequency/Frequency Modulation

1 Application Discovery

A device must be capable of resolving the authoritative FQDN for a service via the methodology defined in the RadioDNS specification [2].

Application lookup may then be performed against this FQDN by means of a SRV Record request for the RadioEPG application by using the service name: `radioepg`

If at least one SRV record is successfully resolved, this service supports the RadioEPG application, accessed on the host and port indicated in the relevant SRV record. For example, for a query made to:

```
_radioepg._tcp.rdns.musicradio.com
```

Using the `nslookup` tool, this would yield the following SRV record:

```
service = 0 100 80 epg.musicradio.com.
```

This indicates that the RadioEPG application can be accessed on the FQDN `epg.musicradio.com`, port `80`.

Note that more than one SRV record may be returned with different values. This can be used for loadbalancing purposes by providing different FQDNs/Ports with different priorities/weightings. See the SRV record specification [1] for a more detailed explanation on handling SRV resolution results.

2 Document retrieval

The host and port returned with the SRV record(s) refers to a HTTP server containing XML documents. The following sections of the present document detail these documents and how they are retrieved and parsed.

Because the resource is returned via HTTP, the HTTP specification [3] should be correctly and fully implemented. Attention is particularly drawn to the status codes section, which may be used to indicate problems and failures during attempts to retrieve documents.

In particular, the following behaviours must be adhered to:

A device **MUST** correctly follow any HTTP redirects that are returned when retrieving a document.

A device **MUST** respect any indicated document expiry in the HTTP response.

It is **RECOMMENDED** that devices cache retrieved documents, as per the HTTP specification.

It must also be noted that any filenames given in this document should be treated as *case sensitive* in order to support different web servers.

2.1 Document Compression

Should a client request indicate that they accept compressed documents using the `Accept-Encoding` header, the service provider may respond with a compressed document using one of the indicated encodings.

The service provider **MUST** include the compression method of the response within the `Content-Encoding` header in the HTTP response, as per standard HTTP behaviour.

If requesting any compression, the client **MUST** then inspect the response to determine if, and how, any returned documents have been compressed and expand accordingly.

3 XSI document

This document is intended to detail service information for a service provider, in much the same way as provided by an SI file in the DAB EPG XML specification [4], with a restructuring of the document to make it suitable for signalling non-DAB bearers.

It holds a definition of services provided by the service provider, including any relevant metadata and bearer details, such as:

- Names (in different lengths and languages)
- Descriptions (in different lengths and languages)
- Logos
- Genres
- Keywords
- Bearers the service can be received over

In contrast with the SI file format, the XSI document allows the definition of **all** bearers the service is available on, over all platforms and in all locations.

This may provide enough information to allow a device to perform S*ervice Following* – i.e. enabling a switch between bearers (both broadcast and streaming) as suits the quality of service provided. More details on how this may be implemented are given in Section 5, Broadcast/IP Service Following.

3.1 Location

The XSI document can be acquired in one of three ways:

- Performing a RadioDNS lookup for the RadioEPG application
- Link to URL of the XSI document within the HTML pages of the service website
- Document placed in a defined location on the service website

The first method is applicable to devices able to receive broadcast parameters, and using these to form a RadioDNS query to locate the RadioEPG application and associated documents.

The second and third methods are applicable to devices, or the directory service provider supporting them, capable of locating, parsing and storing metadata from RadioEPG documents by *web crawling* across service provider websites.

3.1.1 Performing a RadioDNS lookup

The URL to obtain the XSI document is constructed as follows:

`http://<host>:<port>/radiodns/epg/XSI.xml`

Where `host` and `port` are populated by the host and port values obtained from the SRV record lookup for the RadioEPG application.

3.1.2 Link to URL of the XSI document within the Service website

The URL of the XSI document is contained within the (X)HTML header section `<link>` tag of any HTML pages likely to be discovered by a Directory Service Provider.

For example, for a service with a website at `http://www.capitalfm.com/london`, the header section may contain:

```
<head>
    <link rel="radioepg"
          href="http://epg.musicradio.com/radiodns/epg/XSI.xml">
</head>
```

Similarly, for a service with a website at `http://www.classicfm.com`, the header section may contain the same link as above, representing the XSI document for that service provider across all the service website it manages.

3.1.3 Using placement in a defined location on the Service Website

The URL to obtain the XSI document is constructed as follows:

`http://<host>:<port>/radiodns/epg/XSI.xml`

Where `host` and `port` are populated by the host and port of the service website.

For example, for a service with a website on port 80 at `http://www.capitalfm.com/london`, the URL will be:

`http://www.capitalfm.com/radiodns/epg/XSI.xml`

3.2 Implementation

For service providers implementing the RadioEPG application:

- You **MUST** support implementation of Section 3.1.1, Performing a RadioDNS lookup
- You are **STRONGLY RECOMMENDED** to support implementation of document discovery on the standard HTTP port 80, due to the possibility that traffic on a non-standard port may be rejected by firewall/proxy configurations.
- You are **STRONGLY RECOMMENDED** to support implementation of Section 3.1.2, Link to URL of the XSI document within the Service website.
- You are **STRONGLY RECOMMENDED** to support implementation of Section 3.1.3, Using placement in a defined location on the Service Website.

For device manufacturers and developers implementing a RadioEPG client:

- You **MUST** implement at least one method for discovering XSI documents.
- It is **RECOMMENDED** to discover the XSI document immediately upon starting reception of a service.
- It is **RECOMMENDED** to cache any discovered documents as per standard HTTP behaviour.

For directory service providers implementing the ingestion of RadioEPG documents:

- You **MUST** implement at least one method for discovering XSI documents.
- It is **STRONGLY RECOMMENDED** to discover XSI documents by using Section 3.1.2, Link to URL of the

XSI document within the Service website or Section 3.1.3, Using placement in a defined location on the Service Website.

3.3 Contents

Many elements in the XSI document are taken from the DAB EPG XML specification [4], and will be referred to as such in the text and XSD definitions. The following sections detail the points where the specifications diverge and other important notes.

An example of an XSI is shown in Section 8.1, Example XSI Document.

The XSI document has its own schema, but the DAB EPG XML schema is also referenced in order to utilise common datatypes. The XSI namespace and schema are given in Section 8.1, Example XSI Document.

3.3.1 serviceInformation

This is the root element of an XSI file and can contain the following elements:

- `services`
- `groups`

Its attributes are detailed below:

Attribute	Description	Type	Status
`creationTime`	**Document creation datetime** Service broadcast RDS PI code.	*epg:timePointType*	
`originator`	**Originator** Describes the originator of the schedule	*epg:originatorType*	
`serviceProvider`	**Service Provider** Describes the service provider	*epg:serviceProviderType*	
`xml:lang`	**Document Language** Defines the base language of the document	*xml:lang*	*mandatory*

3.3.2 services

Contains zero or more `service` elements

3.3.3 service

Describes metadata and available bearers for a service.

Can contain the following elements:

- `serviceID`
- `radiodns`

It also uses the following elements from the DAB EPG XML Specification:

- Service Name Group (`epg:shortName, epg:mediumName, epg:longName`)
- Media Description (`mediaDescription`)

- genre
- keywords
- link

3.3.4 radiodns

This element is used to signal RadioDNS lookup parameters for the service, and can be used in the discovery of additional RadioDNS applications as per the RadioDNS specification [2], Section 6.

Parameters	Description	Value	Status
fqdn	**RadioDNS Authoritative FQDN** The Authoritative FQDN used in the discovery of RadioDNS applications when no broadcast parameters are available.	*Valid domain*	*mandatory*
serviceIdentifier	**RadioDNS Service Identifier** The Service Identifier used in the discovery of RadioDNS applications when no broadcast parameters are available.	*Maximum 16 lower case characters in the range [a-z][0-9]*	*mandatory*

3.3.5 serviceID

Describes an individual bearer upon which this service is carried. Its attributes are detailed below:

Attribute	Description	Type	Status
id	**Bearer URI** A descriptor of the bearer details. This is platform specific and detailed in Section 7, Bearer URI Construction.	*URI*	*Mandatory*
cost	**Bearer Cost** An indication of a relative 'cost' of acquiring the service from the service provider and may by the device as a means of selecting an appropriate bearer to use. This is detailed in Section 5, Broadcast/IP Service Following.	*Non-negative non-zero integer*	*Mandatory*
mime	**MIME Type** The MIME Type [8] of the audio carried by the bearer.	*Valid MIME type*	*Dependant on the bearer*
bitrate	**Audio Bitrate** Bitrate of the audio carried by the bearer, in kilobits per second (kbps).	*integer*	*Dependant on the bearer*
offset	**Audio Offset** An indication of the offset given to the audio on this bearer by the service provider, in milliseconds relative to other bearers in the same	*Non-negative integer*	*Optional, defaults to zero*

	document.		

Each bearer may have additional requirements, as listed below:

3.3.5.1 VHF/FM

Each VHF/FM bearer is **RECOMMENDED** to be given in **BOTH** forms, as specified in Section 7.1, VHF/FM. That is, one bearer URI constructed using the GCC, and another bearer URI constructed using the country code.

For example, a single FM bearer may be expressed using the following two URIs:

```
fm:gb.c479.09580
fm:ce1.c479.09580
```

3.3.5.2 DAB/DAB+ Digital Radio

The `mime` attribute is **MANDATORY** for this bearer, and must be defined as `audio/mpeg` for DAB and `audio/aacp` for DAB+ services.

3.3.5.3 Digital Radio Mondiale (DRM)/AM Signalling System (AMSS)

The `mime` attribute is **MANDATORY** for DRM, and should be defined as the MIME type of the encoded audio stream.

3.3.5.4 iBiquity Digital Corporation's HD Radio™ (HD Radio™)

No additional requirements.

3.3.5.5 HTTP-based

The `mime` attribute is **MANDATORY** and indicates the MIME type of the audio codec carried in the HTTP response.

The `bitrate` attribute is **RECOMMENDED** and should indicate the bitrate of the audio stream.

3.3.6 memberOf

This can be used to group services together for the purposes of similar service selection or presentation to a user. A service may be a member of zero or more groups, using zero or more `memberOf` elements.

The group the service belongs to is indicated with the `id` attribute in the following manner:

Parameters	Description	Value	Status
`id`	**Group Identifier** An identifying string for the group.	*string*	*mandatory*

Other members of the group should share this identifier to be placed in the same group, and the group should be defined with the exact identifying string as stated within the group definition within the relevant `group` element (see Section 3.3.8, group).

3.3.7 groups

Contains zero or more `group` elements.

3.3.8 group

This can be used to hold the details of a group, which can be used to group services together for the purposes of similar service selection or presentation to a user. The group identifier is contained within the `id` attribute in the following manner:

Parameters	Description	Value	Status
`id`	**Group Identifier** An identifying string for the group.	*string*	*mandatory*

It can also contain the following descriptive elements from the DAB EPG XML Specification:

- Service Name Group (`epg:shortName, epg:mediumName, epg:longName`)
- Media Description (`mediaDescription`)
- `genre`
- `keywords`
- `link`

These are all specific to the group they are specified within.

A service wishing to be a member of the group should use this group identifier in their `memberOf` element (see Section 3.3.6, memberOf).

A group may contain zero or more services.

3.4 Service Filtering

A device implementing direct ingestion of XSI files from a service provider, rather than through a proprietary directory service provider interface, may wish to reduce the amount of data returned in the response. This may be a consequence of hardware resource limitations, e.g. processing power or available memory, or a desire to minimise bandwidth consumption.

A device or other client, requesting an XSI, may optionally provide information to the service provider to indicate particular bearers in which they are interested in receiving further service information.

The service provider may optionally respond with a response containing a subset of their complete service list, more applicable to the bearers specified in the request. This may be only the service(s) using these bearers, but the client must not make this assumption and should still parse and perform any bearer matching to identify particular services (see Section 5.4, Bearer Matching).

3.4.1 Client Request

The client request takes any of the forms given in Section 3.1, Location, with a querystring appended in the following format:

```
<url>[?[bearer=<bearer>][[&bearer=<bearer>]...]]
```

Where `url` is the original base URL. The querystring is populated as follows:

Parameters	Description	Value	Status
`bearer`	**Bearer URI** Bearers(s) the client wishes to request from the service provider, expressed in the format as specified in Section 7, Bearer URI Construction. This parameter may be repeated in the querystring to request multiple bearers.	*URI*	*optional*

All query parameter values must be properly URL-encoded [9]. For example, a request for the following bearer URIs:

```
fm:ce1.c479.09580
dab:ce1.c185.c479.0
http://vis.media-ice.musicradio.com/Capital
```

To the XSI URL:

```
http://epg.musicradio.com/radiodns/epg/XSI.xml
```

Should result in a final URL of:

```
http://epg.musicradio.com/radiodns/epg/XSI.xml?bearer=fm%3Ace1.c479.09580&b
earer=dab%3Ace1.c185.c479.0&bearer=http%3A%2F%2Fvis.media-
ice.musicradio.com%2FCapital
```

3.4.2 Service Provider Response

The service provider may use the requested bearer(s) and return a more appropriate response, as determined by the service provider.

The exact implementation is beyond the scope of this document, but a response may include services using the requested bearers, services related to the requested bearers, or indeed no services at all.

3.4.3 Client Ingestion

A client **MUST NOT** assume that the original request will be precisely fulfilled by the service provider. Any matching of services should still be done by searching for services using the relevant bearer(s), as would happen when ingesting a more complete XSI (see Section 5.4, Bearer Matching).

Should the client deem the response as not having satisfied the original request, i.e. the requested bearers not matching any services, it may request the URL again without any querystring parameters, i.e. to retrieve the more complete XSI document.

4 PI document

This document contains programme and schedule information for a service and is an extension of the PI file in the DAB EPG XML specification [4].

New elements are added in the RadioEPG namespace to provide additional functionality, and the filename conventions and methods of discovery differ to that of the core DAB EPG XML specification. It is important to note that, because of this and the fact that the DAB EPG XML PI XSD declaration does not explicitly accommodate elements foreign to its own namespace, a strict validation of this augmented PI file against the PI XSD will fail.

4.1 Location

The document is acquired using a URL constructed using the format:

`http://<host>:<port>/radiodns/epg/<broadcast parameters>/<date>_PI.xml`

Where `host` and `port` are populated by the host and port values obtained from the SRV record lookup for the RadioEPG application.

PI documents are stored as a file per day's schedule contained within. The `date` value represents the day you wish to obtain the schedule for. It is populated in the format `YYYYMMDD`, for example Sunday, 9th October, 2011 would be represented as `20111009`.

The broadcast parameters are based on the bearer of the service being consumed and specified in the following subsections, specific to each bearer.

4.1.1 7.1.1 VHF/FM

The broadcast parameters value for a VHF/FM service PI request URI is constructed as follows:

`fm/(<gcc>|<country>)/<pi>/<frequency>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`gcc`	**Global Country Code** The Country Code (first nibble of the broadcast RDS PI code) concatenated with the broadcast RDS [7] ECC.	*3-char hexadecimal*	*mutually exclusive*
`country`	**ISO 3166 two-letter country code** In the event that a service broadcast ECC is unavailable, an ISO 2-letter country code [6] must be provided.	*2-char string*	
`pi`	**Programme Identification (PI)** Service broadcast RDS PI code.	*4-char hexadecimal*	*mandatory*
`frequency`	**Frequency** Frequency on which the service broadcast is received, formatted to 5 characters in units of 100KHz. Frequencies below 100Mhz must be supplied with a leading zero, for example 95.8MHz would be represented as 09580,	*5-char string*	*mandatory*

	104.9MHz as 10490.		

A service provider is **RECOMMENDED** to accept URL requests formatted for both `gcc` and `country` values to handle a situation where the device may not acquire the RDS ECC.

For a detailed explanation on these parameters and their values, please refer to the VHF/FM section of RDNS01 [2].

4.1.2 7.1.2 DAB/DAB+ Digital Radio

The broadcast parameters value for a DAB/DAB+ Digital Radio service PI request URI is constructed as follows:

`dab/<gcc>/<eid>/<sid>/<scids>/[(<appty-uatype>|<pa>)]`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`gcc`	**Global Country Code** For services with a 16-bit Service Identifier (SId), this is the first nibble of the SId followed by the Extended Country Code (ECC). For services with a 32-bit SId, this is the third nibble of the SId followed by the first two nibbles of the SId.	*3-char hexadecimal*	*mandatory*
`eid`	**Ensemble Identifier (EId)** Service broadcast multiplex ensemble ID code.	*4-char hexadecimal*	*mandatory*
`sid`	**Service Identifier (SId)** Service broadcast identifier.	*4 or 8-char hexadecimal*	*mandatory*
`scids`	**Service Component Identifier within the Service (SCIdS)** Service broadcast component identifier within the service.	*1 or 3-char hexadecimal*	*mandatory*

If the audio service is delivered as data via X-PAD, the following additional parameter is mandatory:

Parameters	Description	Value	Status
`appty-uatype`	**X-PAD Application Type (AppTy) and User Application type (UAtype)** The X-PAD Application Type number and User Application Type, concatenated with a hyphen (only for applications broadcast in X-PAD). Where Application Types are allocated in pairs, the lower value (indicating the start of the application data group) must be used.	*2-char hexadecimal, hyphen, 3-char hexadecimal*	*mandatory, when referring to an X-PAD component, otherwise omitted*

If the service is delivered as data in an independent Service Component, the following additional parameter is mandatory:

Parameters	Description	Value	Status
`pa`	**Packet Address** Packet address of the data service delivering the audio service.	*integer, between 1 and 1023*	*mandatory, when referring to a data service*

			component, otherwise omitted

For a detailed explanation on these parameters and their values, please refer to the DAB/DAB+ section of RDNS01 [2].

4.1.3 Digital Radio Mondiale (DRM)/AM Signalling System (AMSS)

The broadcast parameters value for a DRM/AMSS service PI request URI is constructed as follows:

`(drm|amss)/<sid>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`sid`	**Service Identifier (SId)** Service broadcast identifier.	*6-char hexadecimal*	mandatory

For a detailed explanation on these parameters and their values, please refer to the DRM/AMSS section of RDNS01 [2].

4.1.4 iBiquity Digital Corporation's HD Radio™ (HD Radio™)

The broadcast parameters value for a HD Radio™ [7] service PI request URI is constructed as follows:

`hd/<cc>/<tx>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`cc`	**Country Code** Service broadcast country code	*3-char hexadecimal*	*mandatory*
`tx`	**Transmitter Identifier** Service broadcast identifier	*5-char hexadecimal*	*mandatory*

For a detailed explanation on these parameters and their values, please refer to the HD Radio™ section of RDNS01 [2].

4.1.5 IP-delivered audio service

The broadcast parameters value for a request URI when receiving IP-delivered audio is constructed as follows:

`id/<fqdn>/<sid>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`fqdn`	**Authoritative FQDN** This **MUST** match the `fqdn` parameter signalled for the IP	*Valid domain name*	*mandatory*

	stream as per the RadioDNS Specification [2], and the `fqdn` attribute on the `radiodns` element of the service in the relevant XSI document (see Section 3.3.4, radiodns).		
`sid`	**Service Identifier** This **MUST** match the `ServiceIdentifer` parameter signalled for the IP stream as per the RadioDNS Specification [2], and the `serviceIdentifier` attribute on the `radiodns` element of the service in the relevant XSI document.	*Maximum 16 lower case characters in the range [a-z][0-9]*	*mandatory*

4.2 Contents

All standard elements within the DAB EPG XML specification PI file can be used, in the method that specification states. The RadioEPG specification adds to this with extra elements in the RadioEPG namespace, detailed in the following sections.

4.2.1 bearer

This element exists under the core DAB EPG XML `location` element of each programme, and details the non-DAB bearers that this programme is available on, in the same way that the core DAB EPG XML `bearer` element does for DAB bearers.

Its attributes are formatted in the same way as for the `serviceID` element in the XSI, as detailed in Section 3.3.5, serviceID.

Defining any `bearer` elements within the `location` element of a `programme` element will override a global declaration of bearers in the XSI for the service the programme is carried on, over the duration of the programme. This applies across both common DAB EPG XML bearers and RadioEPG bearers.

Declaring no bearers at this point will mean that any bearers for the programme will inherit from the global declaration of bearers in the XSI for the service the programme is carried on.

5 Broadcast/IP Service Following

Intelligent switching between broadcast and streaming can be used to provide the device with a common experience between different bearers of the same service, appropriate to the situation. It can also optimise the costs to both device and service provider, associated with the different bearers, e.g. by using broadcast instead of IP streaming.

It is important to note that Service Following using RadioEPG is not intended to be a replacement for similar bearer-specific functionality, for instance the AF/Service Following techniques on broadcast platforms. It is primarily intended for a device to follow between IP-delivered and broadcast bearers.

Service following can be defined globally in the XSI document, or on a per-programme basis in the relevant PI document if available.

5.1 Initial Bearer Selection

It is up to the device to select an initial bearer, but it is **RECOMMENDED** that this be a function of device preference, possible device user preference and indicated bearer cost.

A device should determine the relative preference between certain bearers based on device functionality (e.g. what bearers the device is able to use, available codecs).

A device may wish to expose a degree of choice of bearer to the user and allow them to indicate a preference to a particular bearer. This may also be an indirect consequence of a user action – for example, if a user deactivates Wi-Fi functionality on a mobile/cellular network phone, the device may decide to use FM instead of IP streaming over mobile data.

A *cost* is indicated against each bearer for a service, as determined by the service provider and indicates an order of preference *in respect to the service provider*. This is a relative non-negative non-zero integer, which may be used to select the *most preferred* bearer from the bearer list. The bearer with the lower cost value should be preferred when performing a comparison.

A device should start from the most preferred bearer and work down the list until it is deemed a successful reception has been made. A device should apply its own rules to determine what constitutes a successful reception, such as whether the broadcast signal quality is sufficiently strong, or whether an IP connection can be made and the available bandwidth is sufficient.

5.2 Bearer Switching Behaviour

Service following information provided in the XSI document enables a device to consider a transition to IP streaming of the current service when all possible service following possibilities in the broadcast domain for the current service have been exhausted. It also allows a device receiving a service through IP streaming to consider switching to the same service on broadcast.

In all cases, the provided bearer cost should be considered in the decision to switch between broadcast and IP, and when deciding which of either broadcast or IP to switch to if multiple equitable options are available.

Information provided in the XSI shall not be used in preference to information provided by the current broadcast platform, such as AF information in RDS-FM and Service Following information in DAB. Where the broadcast platform allows signalling of similar services, such as Soft Links in DAB, the device may decide whether to offer the user a switch to the same service on IP streaming, or one of the alternative similar services specified in the broadcast domain.

For example, consider a service being received on DAB. DAB Service Following provides alternative locations for the current service on other ensembles and on FM radio, but the device finds that none are of an acceptable signal quality.

The device inspects the RadioEPG Service Following information, finds an appropriate IP streaming bearer for the current service, and switches to that. The device continues to monitor the broadcast signals available to it, and finds at a later time that the same service is now available with equitable signal qualities on both DAB and FM.

The service provider has specified a lower cost for the DAB bearer, so the device switches from IP Streaming to DAB. Devices should implement appropriate strategies for managing the frequency and duration of switches between IP and broadcast.

The value for an *offset* of a particular bearer, in milliseconds relative to other bearers in the same document, may allow the device to implement functionality to attempt co-timing when switching bearers, or to assist a decision as to the most appropriate bearer. It should be noted that any offset is an indicative, rather than precise value.

5.3 Implementation

The following matrix gives the conditions under which Service Following to another bearer may or may not be implemented by a device:

XSI Available	**PI Available**		**Device Behaviour**
Bearers defined for this service		**Bearers defined in current programme**	
N	N	-	Service following **not** allowed
Y	N	-	Service following allowed to bearers defined for this service, within the XSI
Y	Y	N	Service following allowed to bearers defined for this service, within the XSI
Y	Y	Y	Service following allowed to bearers defined for the current programme, within the `location` element of the PI
N	Y	Y	Service following allowed to bearers defined for the current programme, within the `location` element of the PI
N	Y	N	Service following **not** allowed

Available and *Unavailable* refer to whether a document (XSI or PI) can or cannot be retrieved using HTTP as per Section 2, Document retrieval.

Bearer present refers to a bearer being within the document, either within the `service` element of the XSI file, or the `location` element of the current programme within the PI file. This signals that the bearer is allowed for that service/programme.

Bearer missing refers to a bearer not being within the document, either within the `service` element of the XSI file, or the `location` element of the current programme within the PI file. This signals that the bearer is not allowed for that service/programme.

A Service Provider may wish to signal different bearer availability on a per-programme for a variety of reasons, e.g. to enforce licensing restrictions.

A Service Provider wishing to allow Service Following **MUST** provide bearer information in the XSI document, and **MAY** provide a PI document with programme-specific bearer information.

5.4 Bearer Matching

A device may ingest an XSI document for a variety of reasons. For example, in order to determine which service is currently being received and its associated metadata, or to find other bearers the service can be received on.

The implementation of this should be through *Bearer Matching*, i.e. constructing the URI of the currently received bearer using the methods described in Section 7, Bearer URI Construction. This URI can then be matched against bearers within the XSI document to find the relevant service(s). Note that more than one service may be matched, and a device may take additional steps to match the current service.

It is **RECOMMENDED** that a device also check the current bearer parameters, if available, against the additional bearer parameters listed in the XSI (e.g. bitrate, MIME type).

6 Preference of RadioEPG over existing EPG delivery methodologies

It is acknowledged that DAB and DRM have existing methods in place for the delivery of EPG data over the broadcast platform. This specification is not designed to replace these. However, it may only be possible for some service providers to offer EPG data over IP through a system such as RadioEPG. The aim of this specification is to provide a common method for multiple audio delivery protocols for radio.

On certain services it is possible that EPG data may be available both as part of a broadcast and also via RadioEPG over IP. In this case, it is up to device implementation which data is used.

It is **RECOMMENDED** that service providers ensure any broadcast EPG data is similarly available over IP using RadioEPG.

7 Bearer URI Construction

A bearer may be expressed in a URI format, with the format specific to the bearer platform. The following sections detail the schemes as defined within this version of the RadioEPG specification.

7.1 VHF/FM

The bearer string for a VHF/FM service is constructed as follows:

`fm:(<gcc>|<country>).<pi>.<freq>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`gcc`	**Global Country Code** The Country Code (first nibble of the broadcast RDS PI code) concatenated with the broadcast RDS [7] ECC.	*3-char hexadecimal*	*mutually exclusive*
`country`	**ISO 3166 two-letter country code** In the event that a service broadcast ECC is unavailable, an ISO 2-letter country code [6] must be provided.	*2-char string*	
`pi`	**Programme Identification (PI)** Service broadcast RDS PI code.	*4-char hexadecimal*	*mandatory*
`frequency`	**Frequency** Frequency on which the service broadcast is received, formatted to 5 characters in units of 100KHz. Frequencies below 100MHz must be supplied with a leading zero, for example 95.8MHz would be represented as 09580, 104.9MHz as 10490.	*5-char string*	*mandatory*

For a detailed explanation on these parameters and their values, please refer to the VHF/FM section of RDNS01 [2].

7.2 DAB/DAB+ Digital Radio

The bearer string for a DAB/DAB+ Digital Radio service is constructed as follows:

`dab:<gcc>.<eid>.<sid>.<scids>[.(<appty-uatype>|<pa>)]`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`gcc`	**Global Country Code** For services with a 16-bit Service Identifier (SId), this is the first nibble of the SId followed by the Extended Country Code (ECC). For services with a 32-bit SId, this is the third nibble of the SId followed by the first two nibbles of the SId.	*3-char hexadecimal*	*mandatory*
`eid`	**Ensemble Identifier (EId)**	*4-char*	*mandatory*

	Service broadcast multiplex ensemble ID code.	*hexadecimal*	
`sid`	**Service Identifier (SId)** Service broadcast identifier.	*4 or 8-char hexadecimal*	*mandatory*
`scids`	**Service Component Identifier within the Service (SCIdS)** Service broadcast component identifier within the service.	*1 or 3-char hexadecimal*	*mandatory*

If an audio service is delivered as data via X-PAD, the following additional parameter is mandatory:

Parameters	Description	Value	Status
`appty-uatype`	**X-PAD Application Type (AppTy) and User Application type (UAtype)** The X-PAD Application Type number and User Application Type, concatenated with a hyphen (only for applications broadcast in X-PAD). Where Application Types are allocated in pairs, the lower value (indicating the start of the application data group) must be used.	*2-char hexadecimal, hyphen, 3-char hexadecimal*	*mandatory, when referring to an X-PAD component, otherwise omitted*

If an audio service is delivered as data in an independent Service Component, the following additional parameter is mandatory:

Parameters	Description	Value	Status
`pa`	**Packet Address** Packet address of the data service delivering the audio service.	*integer, between 1 and 1023*	*mandatory, when referring to a data service component, otherwise omitted*

For a detailed explanation on these parameters and their values, please refer to the DAB/DAB+ section of RDNS01 [2].

7.3 Digital Radio Mondiale (DRM)/AM Signalling System (AMSS)

The bearer string for a DRM/AMSS service is constructed as follows:

`(drm|amss):<sid>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`sid`	**Service Identifier (SId)** Service broadcast identifier.	*6-char hexadecimal*	mandatory

For a detailed explanation on these parameters and their values, please refer to the DRM/AMSS section of RDNS01 [2].

7.4 iBiquity Digital Corporation's HD Radio™ (HD Radio™)

The bearer string for a HD Radio™ [7] service is constructed as follows:

`hd:<cc>.<tx>.<frequency>`

The parameters are populated with the following values:

Parameters	Description	Value	Status
`cc`	**Country Code** Service broadcast country code	*3-char hexadecimal*	*mandatory*
`tx`	**Transmitter Identifier** Service broadcast identifier	*5-char hexadecimal*	*mandatory*
`frequency`	**Frequency** Frequency on which the service broadcast is received, formatted to 5 characters in units of 100KHz. Frequencies below 100MHz must be supplied with a leading zero, for example 95.9MHz would be represented as 09590, 104.9MHz as 10490.	*5-char string*	

For a detailed explanation on these parameters and their values, please refer to the HD Radio™ section of RDNS01 [2].

7.5 HTTP-based

The bearer string takes the form of a valid URL as defined in the HTTP specification [3]. For example:

`http://media-ice.musicradio.com/Capital`

7.6 Other Bearers

While other bearers exist over which audio can be carried (e.g. DVB, IPTV, etc.) this version of the RadioEPG specification does not define the format by which they may be expressed.

8 Example RadioEPG Documents

8.1 Example XSI Document

```
<?xml version="1.0" encoding="UTF-8"?>
<serviceInformation xmlns="http://schemas.radiodns.org/epg/10"
xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://schemas.radiodns.org/epg/10 http://schemas.radiodns.org/epg/10
/radioepg_xsi_10.xsd"
creationTime="2011-07-01T00:00:00" originator="Global Radio" serviceProvider="Global Radio"
xml:lang="en">
  <services>
    <service>
      <serviceID id="dab:ce1.c185.c479.0" mime="audio/aacp" offset="3000" cost="10"/>
      <serviceID id="fm:ce1.c479.09580" cost="30"/>
      <serviceID id="http://media-ice.musicradio.com/Capital" offset="4000" mime="audio/aacp"
bitrate="48" cost="40"/>
      <serviceID id="http://media-ice.musicradio.com/CapitalMP3Low" offset="4000" mime="audio/mpeg"
bitrate="48" cost="40"/>
      <epg:shortName xml:lang="en">Capital</epg:shortName>
      <epg:mediumName xml:lang="en">95.8 Capital FM</epg:mediumName>
      <epg:longName xml:lang="en">95.8 Capital FM</epg:longName>
      <mediaDescription>
        <epg:shortDescription xml:lang="en">London's No.1 Hit Music Station</epg:shortDescription>
      </mediaDescription>
      <mediaDescription>
        <epg:multimedia url="http://www.capitalfm.com/capital_logo.png" type="logo_colour_square"/>
      </mediaDescription>
      <keywords xml:lang="en">London, music, pop, rock, dance, urban</keywords>
      <link url="http://www.capitalfm.com/london" mimeValue="text/html" xml:lang="en"/>
      <memberOf id="CapitalNetwork"/>
      <radiodns fqdn="rdns.musicradio.com" serviceIdentifier="caplon"/>
    </service>
  </services>
  <groups>
    <group id="CapitalNetwork">
      <epg:shortName xml:lang="en">Capital</epg:shortName>
      <epg:mediumName xml:lang="en">Capital Network</epg:mediumName>
      <epg:longName xml:lang="en">The Capital Network</epg:longName>
      <mediaDescription>
        <epg:shortDescription xml:lang="en">The Capital Network</epg:shortDescription>
      </mediaDescription>
      <mediaDescription>
        <epg:multimedia url="http://www.capitalfm.com/capital_network.png" type="logo_colour_square"/>
      </mediaDescription>
      <keywords xml:lang="en">music, pop, rock, dance, urban</keywords>
      <link url="http://www.capitalfm.com/london" mimeValue="text/html" xml:lang="en"/>
    </group>
  </groups>
</serviceInformation>
```

8.2 Example PI Document

```
<?xml version="1.0" encoding="UTF-8"?>
<epg xmlns="http://www.worlddab.org/schemas/epgSchedule/14"
     xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:repg="http://schemas.radiodns.org/epg/10"
```

```
     system="DAB"
     xsi:schemaLocation="http://www.worlddab.org/schemas/epgSchedule/14
http://www.worlddab.org/schemas/epgSchedule/14/epgSchedule_14.xsd"
xml:lang="en">
  <schedule version="1" creationTime="2009-10-05T00:00:00+01:00" originator="Global Radio">
    <scope startTime="2009-10-05T06:00:00+01:00" stopTime="2009-10-05T13:00:00+01:00">
      <serviceScope id="ce1.c185.c586.0"/>
    </scope>
    <programme shortId="1190223" id="crid://thisisglobal.com/1190223" recommendation="yes">
      <epg:shortName>B'fast</epg:shortName>
      <epg:mediumName>Breakfast</epg:mediumName>
      <epg:longName>Capital Breakfast</epg:longName>
      <epg:location>
        <epg:time time="2009-10-05T06:00:00+01:00" duration="PT4H0M0S"
                  actualTime="2009-10-05T06:00:00+01:00" actualDuration="PT4H0M0S"/>
        <epg:bearer id="ce1.c185.c586.0"/>
        <repg:bearer id="dab:ce1.c185.c586.0" mime="audio/mpeg" bitrate="128" cost="10"/>
        <repg:bearer id="fm:ce1.c586.09580" cost="30"/>
        <repg:bearer id="http://media-ice.musicradio.com/Capital" offset="4000" mime="audio/aacp"
                     bitrate="56" cost="40"/>
        <repg:bearer id="http://vis.media-ice.musicradio.com/CapitalMP3Low" mime="audio/mpeg"
                     bitrate="56" cost="40"/>
      </epg:location>
      <epg:mediaDescription>
        <epg:shortDescription>Forget the coffee, Capital gives you the perfect morning pick-me-up with
a blend of the latest hits, travel news and incomparable morning banter.</epg:shortDescription>
      </epg:mediaDescription>
      <epg:genre href="urn:tva:metadata:cs:ContentCS:2002:3.6.8">
        <epg:name><![CDATA[ Electronic/Club/Urban/Dance]]></epg:name>
      </epg:genre>
      <epg:genre href="urn:tva:metadata:cs:IntentionCS:2002:1.1">
        <epg:name><![CDATA[ ENTERTAINMENT]]></epg:name>
      </epg:genre>
      <epg:memberOf shortId="1000" id="crid://thisisglobal.com/Capital/breakfastGroup"/>
      <epg:link url="mailto:capital.breakfast@capitalfm.com" description="Email:"/>
      <epg:link url="http://www.capitalfm.com/on-air/breakfast-show/" description="Web:"/>
      <epg:programmeEvent shortId="11902231" id="crid://thisisglobal.com/1190223/1"
                          recommendation="yes">
        <epg:shortName xml:lang="en">Pun</epg:shortName>
        <epg:mediumName xml:lang="en">No.1 Pun</epg:mediumName>
        <epg:longName xml:lang="en">London's No. 1 Pun</epg:longName>
        <epg:location>
          <epg:relativeTime time="PT3H10M" duration="PT25M"/>
        </epg:location>
        <epg:mediaDescription>
          <epg:shortDescription xml:lang="en">
            Can you come up with London's No.1 Pun for our story of the day?
          </epg:shortDescription>
        </epg:mediaDescription>
      </epg:programmeEvent>
    </programme>
  </schedule>
</epg>
```

9 XSD Definitions

9.1 XSI XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns="http://schemas.radiodns.org/epg/10"
           targetNamespace="http://schemas.radiodns.org/epg/10"
           xmlns:epg="http://www.worlddab.org/schemas/epgDataTypes/14"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           elementFormDefault="qualified"
           attributeFormDefault="unqualified">

  <xs:import namespace="http://www.w3.org/XML/1998/namespace"
             schemaLocation="http://www.w3.org/2001/xml.xsd" />
  <xs:import namespace="http://www.worlddab.org/schemas/epgDataTypes/14"

schemaLocation="http://www.worlddab.org/schemas/epgDataTypes/14/epgDataTypes_14.xsd" />

  <xs:element name="serviceInformation">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="services" type="servicesType" minOccurs="0" maxOccurs="1" />
    <xs:element name="groups" type="groupsType" minOccurs="0" maxOccurs="1" />
   </xs:sequence>
   <xs:attribute name="creationTime" type="epg:timePointType" />
   <xs:attribute name="originator" type="epg:originatorType" />
   <xs:attribute name="serviceProvider" type="epg:serviceProviderType" />
   <xs:attribute ref="xml:lang" use="required" />
  </xs:complexType>
 </xs:element>

 <xs:complexType name="groupsType">
  <xs:sequence>
   <xs:element name="group" type="groupType" maxOccurs="unbounded" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="servicesType">
  <xs:sequence>
   <xs:element name="service" type="serviceType" maxOccurs="unbounded" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="serviceType">
  <xs:sequence>
   <xs:element name="serviceID" type="serviceIDType" minOccurs="0" maxOccurs="unbounded" />
   <xs:group ref="epg:serviceNameGroup" maxOccurs="unbounded" />
   <xs:element name="mediaDescription" type="epg:mediaDescriptionType"
               minOccurs="0" maxOccurs="unbounded" />
   <xs:element name="genre" type="epg:genreType" minOccurs="0" maxOccurs="unbounded" />
   <xs:element name="keywords" type="epg:keywordsType" minOccurs="0"
               maxOccurs="unbounded" />
   <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded" />
```

```
   <xs:element name="memberOf" type="memberOfType" minOccurs="0" maxOccurs="unbounded"/>
   <xs:element name="radiodns" type="radiodnsType" minOccurs="0" maxOccurs="1" />
  </xs:sequence>
 </xs:complexType>

 <xs:complexType name="memberOfType">
  <xs:attribute name="id" type="xs:string" use="required" />
 </xs:complexType>

 <xs:complexType name="radiodnsType">
  <xs:attribute name="fqdn" type="xs:string" use="required" />
  <xs:attribute name="serviceIdentifier" type="xs:string" use="required" />
 </xs:complexType>

 <xs:complexType name="serviceIDType">
  <xs:attribute name="id" type="xs:anyURI" />
  <xs:attribute name="mime" type="xs:string" />
  <xs:attribute name="bitrate" type="xs:nonNegativeInteger" />
  <xs:attribute name="cost" type="xs:nonNegativeInteger" />
  <xs:attribute name="offset" type="xs:nonNegativeInteger" default="0"/>
 </xs:complexType>

 <xs:complexType name="groupType">
  <xs:sequence>
   <xs:group ref="epg:serviceNameGroup" maxOccurs="unbounded" />
         <xs:element name="mediaDescription" type="epg:mediaDescriptionType"
                     minOccurs="0" maxOccurs="unbounded" />
         <xs:element name="genre" type="epg:genreType"
                     minOccurs="0" maxOccurs="unbounded" />
         <xs:element name="keywords" type="epg:keywordsType"
                     minOccurs="0" maxOccurs="unbounded" />
         <xs:element name="link" type="epg:linkType" minOccurs="0" maxOccurs="unbounded" />
  </xs:sequence>
  <xs:attribute name="id" type="xs:string" />
 </xs:complexType>

</xs:schema>
```

9.2 PI XSD Extensions

As the PI XML file has been augmented with the RadioEPG-specific `bearer` element, and the original DAB EPG XML XSD file does not explicitly allow foreign elements, it may no longer be deemed valid by a validating parser.

However, for reference purposes, the following element is defined:

```
<xs:element name="bearer" type="serviceIDType"/>
```

10 History

Document history		
V0.6.1	June, 2009	First Working Draft
V1.0.0	April, 2012	Publication