



# Service Identification [v2.2]

for Broadcasters & Aggregators

## Executive Summary

The IMDA publishes their Service Identification for Broadcasters & Aggregators in an attempt to standardise the industry of Internet Radio, in the first instance, and establish a working framework for future media services.

The Service Identification describes a way for a broadcaster, or media organisation, to expose their data to a hardware or software solution run by a third party. The data from the media organisation contains details of itself, its brands and its brands' transport methods.

Not only is this technical information but also editorial information allowing the solution to correctly categorise and expose brands to the audience, and to allow a solution to evaluate appropriateness of brands to criteria supplied where the solution includes some search functionality or automated browsing features.

In order for an organisation to make available this data to others the IMDA recommends the organisation to follow the Service Identification specifications included in this document so that common understanding, ease of repurposing and avoidance of publishing data in different formats for different solutions is achieved. This provides benefits for both the media organisation and the solution provider.

## Table of Contents

1. Introduction
  - 1.1. Licence
  - 1.2. How this document is structured
  - 1.3. General advice
    - 1.3.1. Why UTF-8?
    - 1.3.2. Are there XML schemas available?
    - 1.3.3. Is there a Data Relationship Diagram available?
    - 1.3.4. Why split data into multiple XML documents?
    - 1.3.5. Are there any requirements for the web server that the files reside on?
    - 1.3.6. How should a solution determine if a file has been updated?
  - 1.4. How to provide feedback and input to specifications
2. Service Identification
  - 2.1. imda node
    - 2.1.1. imda@modified
    - 2.1.2. imda@owner
    - 2.1.3. imda@imdaversion

- 2.2. imda/terms
  - 2.2.1. imda/terms/general
  - 2.2.2. imda/terms/exceptions
  - 2.2.3. imda/terms/privacy
  - 2.2.4. imda/terms/privacyexceptions
  - 2.2.5. imda/terms/safecontent
- 2.3. imda/organisation
- 2.4. imda/brands
- 2.5. imda/transport
- 2.6. imda/manifests
- 2.7. Protecting your metadata
- 3. Organisation Identification
  - 3.1. organisation node
    - 3.1.1. organisation@modified
    - 3.1.2. organisation@owner
  - 3.2. organisation/id
    - 3.2.1. organisation/id@crd
  - 3.3. organisation/title
  - 3.4. organisation/description
  - 3.5. organisation/links
  - 3.6. organisation/tags
  - 3.7. organisation/genres
  - 3.8. organisation/logos
  - 3.9. organisation/headquarters
  - 3.10. organisation/locations
    - 3.10.1. .../locations/location@id
    - 3.10.2. .../locations/location types
- 4. Brands Identification
  - 4.1. brands node
    - 4.1.1. brands@modified
    - 4.1.2. brands@owner
    - 4.1.3. brands@orgcrd
  - 4.2. brand
    - 4.2.1. brand@crd
    - 4.2.2. brand@id
    - 4.2.3. brand@type
  - 4.3. brand/title
  - 4.4. brand/description
  - 4.5. brand/links
  - 4.6. brand/tags
  - 4.7. brand/genres
  - 4.8. brand/intendedaudience
  - 4.9. brand/safe
  - 4.10. brand/logos
  - 4.11. brand/spoken
  - 4.12. brand/location
  - 4.13. brand/editorialrelevance

## 5. Transports Identification

- 5.1. transports node
  - 5.1.1. transports@modified
  - 5.1.2. transports@owner
  - 5.1.3. transports@orgerid
- 5.2. transports/brand
  - 5.2.1. transports/brand@id
  - 5.2.2. transports/brand@type
- 5.3. transports/brand/transport
  - 5.3.1. transport@type : ip\_direct or broadcast
  - 5.3.2. transport@type : ip\_indirect
  - 5.3.3. transport@type : the future
- 5.4. transport/brand/radiodns
- 5.5. Additional features
  - 5.5.1. RadioDNS
  - 5.5.2. Your feature here!

## 6. Manifests Identification

- 6.1. manifest@type
- 6.2. manifest/format
- 6.3. manifest[@type='epg']/format[@name='dabepg']
  - 6.3.1. basetimezone
  - 6.3.2. si
  - 6.3.3. pi
  - 6.3.4. gi
  - 6.3.5. mapping
  - 6.3.6. urlpattern macros
- 6.4. manifest[@type='epg']/format[@name='dabepg\_drm']
- 6.5. manifest[@type='epg']/format[@name='xmltv']
  - 6.5.1. basetimezone
  - 6.5.2. mapping
- 6.6. manifest[@type='epg']/format[@name='webepg']
  - 6.6.1. basetimezone
  - 6.6.2. specification
  - 6.6.3. mapping
- 6.7. manifest[@type='podcast']/format[@name='rss20']
  - 6.7.1. basetimezone
  - 6.7.2. mapping
- 6.8. manifest[@type='podcast']/format[@name='atom10']
  - 6.8.1. basetimezone
  - 6.8.2. mapping
- 6.9. Your favourite manifest type or EPG format here!

## 7. Shared Nodes

- 7.1. [organisation|brands/brand]/title
- 7.2. [organisation|brands/brand]/description
- 7.3. [organisation|brands/brand]/links
  - 7.3.1. .../links/link@devicetype
  - 7.3.2. .../links/link@mimetype
- 7.4. [organisation|brands/brand]/tags

- 7.4.1. .../tags/tag
- 7.4.2. .../tags/tag@weight
- 7.5. [organisation|brands/brand]/genres
  - 7.5.1. .../genres/genre
- 7.6. [organisation|brands/brand]/logos
  - 7.6.1. .../logos/logo@type
  - 7.6.2. .../logos/logo@url
  - 7.6.3. .../logos/logo@background

## Appendices

- A. XML Schemas
- B. Example documents
- C. Frequently Asked Questions

# 1 | Introduction

Before we go further – it is important to understand some basic terminology we use as this is often a point of confusion:

### *solution* ~

we believe that a device exists, in the majority, connected to a server run by the manufacturer of the device or the aggregator that controls the device – often it is easier to refer to this as a 'solution' which is the combination of the device and the server that powers it. Consider the 'solution' a black-box which you cannot, and do not need to, see inside.

### *media organisation* ~

we identify the broadcaster, publisher or individual that has media services as a 'media organisation' for simplicity. (e.g. BBC, Soma FM, Clear Channel, CNN, etc)

### *brand* ~

in order to identify a service (e.g. BBC Radio 4, Groove Salad, ITV1, etc) we refer to an individual service as a 'brand' which a media organisation has one or more of.

### *transport* ~

each brand may have multiple ways of distributing the content via Internet Protocol - we identify these as a 'transport'. (e.g. via Windows Media streaming, Shoutcast streaming, Flash streaming, etc)

These documents describe a way for a media organisation to expose their data to a solution (or solutions). This data regards details of itself, its brands and its brands' transport methods.

Not only is this technical information but also editorial information allowing the solution to correctly categorise and expose brands to the audience, and to allow a solution to

evaluate appropriateness of brands to criteria supplied where the solution includes some search functionality or automated browsing features.

In order for an organisation to make available this data to others the IMDA recommends the organisation to follow the specification included in this document so that common understanding, ease of repurposing and avoidance of publishing data in different formats for different solutions is achieved. This provides benefits for both the media organisation and the solution provider.

The IMDA specification defines a number of XML documents and the Schema for these documents that a media organisation can use to expose their data.

## 1.1 | Licence

The IMDA is publishing this specification under the GNU General Public License version 3 [<http://www.gnu.org/licenses/>] and the GNU Free Documentation License version 1.3 [<http://www.gnu.org/copyleft/fdl.html>]. We feel that it is important that access to this is not limited, and allows any media organisation to make use of it. We have made every effort not to include references to other types of licensed specifications – where this simply hasn't been possible we clearly sign post areas which you need to pay special attention to.

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

```
"IMDA Service Identification for Broadcasters & Aggregators"

- GNU General Public License version 3 -

Copyright (C) 2010 Internet Media Device Alliance

This program is free software: you can redistribute it and/or modify it
under the terms of the GNU General Public License as published by the
Free Software Foundation, either version 3 of the License, or (at your
option) any later version.

This program is distributed in the hope that it will be useful, but
WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General
Public License for more details.

You should have received a copy of the GNU General Public License along
with this program. If not, see http://www.gnu.org/licenses/ [http://bit.ly/gnulicenses]. For questions please contact the Internet Media Device
Alliance (IMDA) Program Manager at +1 631 261 5536 or send an email to
info@imdalliance.org

- GNU Free Documentation License version 1.3 -

Copyright (C) 2010 Internet Media Device Alliance

Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
```

Texts.

A copy of the full license is available at the following location:  
<http://www.gnu.org/copyleft/fdl.html> [<http://www.gnu.org/copyleft/fdl.html>]

## 1.2 | How this document is structured

Each section will give you an overview of the specification by using a simple block diagram, then the section will go into detail of the data types and values required to correctly build your XML document(s).

When defining these data types and values we have provided examples where necessary to identify best practice. If there is information specific to a media organisation or the solution provider then we will show this as follows:



*This 'call out' highlights specific information that the media organisation or broadcaster may find useful.*



*This 'call out' highlights specific information that the solution provider or aggregator may find useful.*

With our specifications there is much which is optional. We will attempt to identify which areas *must* be completed by the visual 'black triangle' in our box diagrams (see below for an example) which we use to provide an initial overview of a section and by highlighting a summary of what is required on each section (in a box to the right of a section's title). This is the '**minimum metadata set**'. We will also use the 'call outs' above to advise you about areas which may not be required but to advise you of reasons why you may wish to complete the information for best practice. There are many example XML documents used throughout this specification, and we have grouped the main example documents together in Appendix B – we have also made available there copies of the main documents cut down to the minimum they need, but urge you to read the specification as we give guidance on why you should use optional nodes and attributes to give more information which may result in your brands being easier to find.



**Fig 1:** How to identify minimum data areas that must be completed at the very least.

We also use a shorthand to refer to a particular node or attribute and its parentage. This 'shorthand' is actually XPath – a syntax for defining parts of an XML document. Although a good introduction to XPath is available [<http://bit.ly/xpath>] from W3Schools, a quick check here may help. In this very simple example we have an XML fragment that is:

```

1 | <name>
2 |   <firstname nickname="bobby">robert</firstname>
3 |   <lastname>ewing</lastname>
4 | </name>

```

Then if we wanted to identify the 'nickname' attribute we would write:

```
name/firstname@nickname
```

Which simply means that `firstname` is a child of `name` – the order reads left to right and this identifies the parentage of children, we separate nodes by a '/' (slash). The `@nickname` is an attribute and we identify attributes with an '@' (at symbol). This allows you to see what area of an XML document we are referring to.

We have tried to provide examples of xml which can be understood by most, to have some shared understanding, so we have used some of the BBC's brands and details – though the BBC is a founding member of the IMDA, it should be noted that the examples contained here are fictitious and are for explanation only. The BBC's real IMDA Service Identification is very different.

## 1.3 | General advice

### 1.3.1 | Why UTF-8?

The IMDA Service Identification define a set of files that are all XML. All your xml should open with the XML declaration which identifies the character set we will be using: UTF-8. Reasons for picking UTF-8 are covered in excellent article from IBM [<http://bit.ly/whyutf8>].

### 1.3.2 | Are there XML schemas available?

Yes, a set of XML schemas are available on our website – these can be used to validate your content. All full list of all schemas is available in the Appendices (see Appendix "XML Schemas")

### 1.3.3 | Is there a Data Relationship Diagram available?

In a future version of these specifications we will include a data relationship diagram (probably in the Appendix). We have a working diagram used to help us understand the relations between data entities (objects) described here. We are currently considering including this as some may find it useful where it gives an overview of relations and can be helpful in understanding the structure of the XML file. It may also prove useful as a guideline when implementing a database (e.g. for the aggregator). If you are interested, please provide feedback to us.

### 1.3.4 | Why split data into multiple XML documents?

As you will discover, our specification identifies multiple XML files that should be created. Why? Surely we could have put everything into one XML document?

There are few reasons for splitting things up – it's mainly about making things more manageable and also thinking to how others will pick up this data in an Internet Protocol world. It's also about the future.

Although we have provided the initial split up we are allowing XML Inclusions [<http://bit.ly/xinclude>] (XInclude) in the sub-documents, but not in the root document.

Here's a few reasons why we have split things up:

- By grouping data into separate XML documents, where similar data may come from different areas in an media organisation's – we've tried to group data together so that they could be managed by different teams. Indeed – you can make use of XInclude in the sub-documents should there be further need to split things up.
- Not everything updates as regularly as everything else – so information about the media organisation may not change very much for long periods, whereas services of the media organisation may change occasionally. In the future we will include being able to expose Electronic Programme Guides and On-Demand services – this changes far more frequently. Combine this with the 'weight' (file size) of the documents – heavy documents that frequently change need to be accessed by solutions very often to keep things up to date. Keeping things grouped into as small a file size as possible, without having any extraneous information, can be desirable.
- Changing specifications – because the IMDA Service Identification will be a live, ever changing, set of specifications we need to consider how changes will be rolled out and the scope of the impact on existing organisations using them. By also keeping stuff in separate documents we should be able to minimise the impact, the dependencies.



*Important: as an aggregator you must make sure you check for a resolve XInclude directives as per the XML Inclusions standard from the W3C [<http://bit.ly/xinclude>] as an organisation may have used these in sub-documents (Organisation, Brands, Transports...)*



*Why don't we allow XInclude within the root document? You might wonder why the root document contains references to external sub-documents (organisation, brands, transports, ...) rather than allowing XInclude's to insert the content of those into the root document. There was much discussion about this but, in the end, one of the reasons why we decided to restrict the use of XInclude to sub-documents was because you would require one schema document to validate the entire document (where XInclude results in including everything into the root node somehow). This would make the schema potentially unwieldy, not to mention requiring solutions to read the root document and pull in all the XInclude external references every time. It just wasn't a good idea.*

### 1.3.5 | Are there any requirements for the web server that the files reside on?

It is important that an organisation comply with the following requirements, this is mainly to help the solutions that may be requesting data:

- Web Server should comply to HTTP/1.1 – this is a recommendation to allow organisations to use the new cache features that 1.1 brought, including headers like `If-modified-since` which could allow an aggregator/solution to be more efficient.
- Atomic updates! As an organisation you should avoid putting partial files onto your server, even if you have a backend system that creates any of the documents you should create them elsewhere and then overwrite the old files with them – this is to lessen the likelihood that a solution would request data and receive unfinished content.
- Don't leave old files lying around on your web server – because our spec allows you to put sub-documents into any location you please it is important that, should you move any of those documents, you delete the old file. A request for the old document location should result in an HTTP 404 error being returned – if a solution receives a 404, then they should restart the parsing of an organisation's IMDA Service Identification documents at the root document - `imda.xml`

### 1.3.6 | How should a solution determine if a file has been updated?

We cannot prescribe the method a solution would use, however here are some recommendations:

- an organisation's web server may return the modified date of the file on the file system – it would be sensible to make sure the modified date doesn't keep changing when the file itself doesn't, this may be because you have some script running on

your web server.

- an organisation can suggest the modified date also in the document itself (this is an attribute in the root node of each document, as you will see). This modified date should only be updated if you update the file. Also – the modified date of a sub-document shouldn't be 'bubbled' upwards to the root document, the modified date attribute is about the document it is defined in and not anything else.

---

## 1.4 | How to provide feedback and input to specifications

We are keen to hear your feedback on these specifications. Whether it's a specific problem or an improvement idea please – [get in touch with us by email](#). There are also specific areas in the specification where we maintain a list of possible values for particular attributes – we point these out in the documentation and feedback via our email is requested if the values don't fit your needs.

You should be aware that this specification may change over time as we add new services, so we have made sure the version of this specification is available at the top of the document. We have made every effort to make sure we don't make significant structural changes to existing specifications – however, we cannot guarantee that future changes won't require structural changes. In the near future we will add a notification system that you can subscribe to in order to make sure you are kept up to date with any changes.

Another resource is our Frequently Asked Questions regarding this specification document – you can find these in the [Appendix: Frequently Asked Questions](#). You may want to check this before getting in touch.

Our website is: [www.imdalliance.org](http://www.imdalliance.org) [<http://bit.ly/imda>]

Email your feedback regarding the specifications: [metadata@imdalliance.org](mailto:metadata@imdalliance.org)

## 2 | Service Identification

In order for media organisations to expose the availability of their IP services we ask the media organisation to publish several XML files publicly available on their web servers. The first of which allows easy discovery of services and is called `imda.xml` and an example of where this would be located is:

```
http://www.organisation.com/imda.xml
```

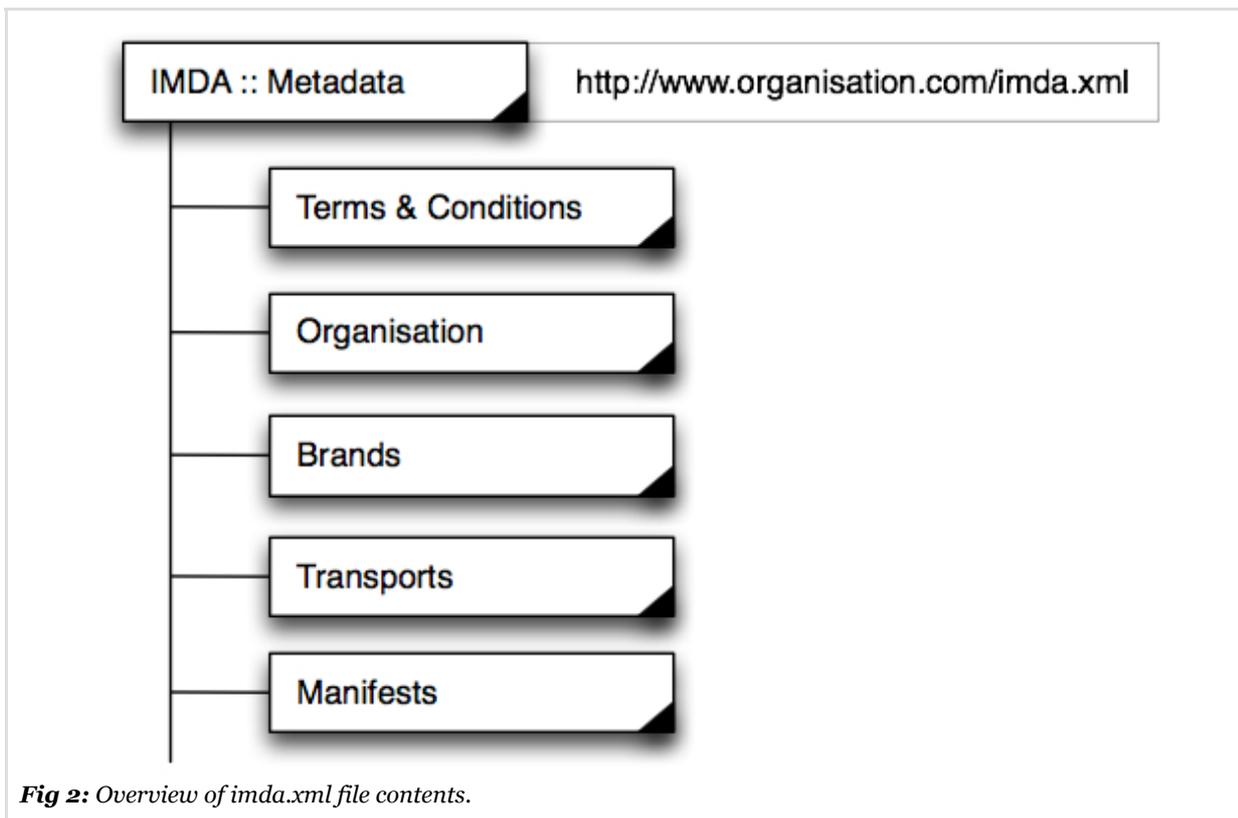
The `imda.xml` file is an organisation's entry point to all service related metadata that follows the IMDA specifications. The file is relatively short and merely acts like a sitemap of service.



*Having your metadata organized in this standardised way is really useful. However being able to provide metadata yields to the next question: How do I provide my XML-files or even just the change of one certain file to the solution provider? As, from a media organisations point of view, there are a lot of solution providers to address... and from a solution providers point of view there are even more media organisations.*

*That's why the IMDA is currently also working on an optional service which works alongside this open specification, which will allow a simple and easy-to-use way of providing notifications around changes to your metadata via a central entity for the benefit of both organisations and solution providers alike. We call it the CDS – the Central Discovery Service. And it will be coming soon.*

Let's take a look at the main elements of this file:



**Fig 2:** Overview of imda.xml file contents.

Reading from the top of this tree, we see that it contains information about:

*Terms & Conditions ~*

Presents the terms and conditions of use of this data. The IMDA has suggested a default wording here to allow most systems to immediately make use of your data. However if you specify any exceptions to those general terms it may delay or stop your listing in particular solutions.

This also includes the IMDA approved 'safe content' definition - referred to by 'brands'. It also includes the ability to specify a Privacy Policy.

*Organisation ~*

Contains information that describes the broadcaster or media organisation. Includes: title, description, logos, location, genres and other classification information.

*Brands ~*

Contains information that describes the brand/services/channels offered by the organisation : e.g. BBC Radio 4, Groove Salad, etc. Also contains information on logical groupings of brands.

*Transports ~*

Contains information that describes how to playback audio/video of a particular brand. e.g. the Windows Media simulcast stream details for the brand BBC Radio 4. In the future this may contain information about other transport types not just IP services.

*Manifests ~*

Contains information around schedule and programme data for brands. e.g. the schedule of programmes for BBC Radio 3, the podcast RSS for NPR podcasts.

As a quick example of how this could appear in the XML file, using the BBC as the media organisation:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <imda modified="2010-10-02T150530Z" owner="imda.services@bbc.co.uk" imdaversion="1.0.0">
3    <terms>
4      <general url="http://www.imdalliance.org/metadata/termsV1.txt [http://www.imdalliar
5      <privacy url="http://www.imdalliance.org/metadata/privacyV1.txt [http://www.imdall:
6      <safecontent url="http://www.imdalliance.org/metadata/safeV1.txt [http://www.imdal
7    </terms>
8    <organisation url="http://imda.bbc.co.uk/imda_org.xml [http://imda.bbc.co.uk/imda_org.xml
9    <brands url="http://imda.bbc.co.uk/imda_brands.xml [http://imda.bbc.co.uk/imda_brands.xml
10   <transports url="http://imda.bbc.co.uk/imda_transports.xml [http://imda.bbc.co.uk/imda_
11   <manifests url="http://imda.bbc.co.uk/radio/imda/imda_manifests.xml [http://imda.bbc.co
12 </imda>
    
```

Hopefully, from this simple example, there should be much that makes sense. You should see that it is fairly straightforward. We'll now look at the nodes and attributes in detail – our root node is `imda`.

## 2.1 | imda node

» node is required

The `imda` node encloses our service metadata but itself contains information in attributes about the last modified date and an email contact address for the owner of the data. Inside `imda` we will keep references to various files/services – these are currently information regarding the media organisation, the brands of the organisation and the transports for the brands.

### 2.1.1 | imda@modified

» attribute is required

This attribute contains a time stamp of when you last updated the file. The data type used here follows ISO 8601 [http://bit.ly/isodate] recommendations, and we use 'Z' to denote the timezone 'Zulu Time' (UTC). Do not use a local timezone here for simplicity of comparison. Example – 27th of October 2009, at 15 hrs 5 minutes and 30 seconds Zulu would be:

```
2009-10-27T150530Z
```

### 2.1.2 | imda@owner

» attribute is required

Define an email address that can be used to contact the department/person looking after these files – this is to allow solution providers to contact them should there be a problem. We recommend you to use an alias here to allow you to route the email to the correct department/person rather than a real name here for privacy reasons. From the example:

```
imda.services@bbc.co.uk
```

### 2.1.3 | imda@imdaversion

» attribute is required

In order to assist solutions to make sure they are checking your document, based on compatibility with a particular version of these IMDA Service Identification specifications, an organisation should specify here the version number of these specs that they have used in **this and all sub-documents**. This is to make sure that, as an organisation providing the data, you can be sure that interpretation of the data you provide is based on the version of the specifications you used to create them. At the moment, this is for future use – should new versions be provided then solutions know what to expect: when checking for compatibility only the major and minor version should match, changes only made to the revision should have no influence on the implementation.

The version of this specification document is visible on the first page, at the top and is also printed at the foot of the document. The current version of this document is:

2.2

Our version number consists of three parts:

<major>.<minor>.<revision>

Rules for updating the major, minor and revision numbers:

- **major** – incremented by one if the specification is not upward compatible with the previous specification. This means that your implementation needs to be modified in order to be compliant with the specification. Incrementing the major version will reset the minor and revision version to zero. Examples of changes that will cause an update of the major version number:
  - If the hierarchy of existing nodes and/or attributes changes. This will be visible in a change of the XPath of an element.
  - Addition of new nodes and/or attributes which are required "to make it work" (e.g. because it was an omission in a previous specification)
- **minor** – incremented by one if the specification is upward compatible with the previous specification but your implementation needs to be modified in order to be compliant with the specification. Incrementing the minor version will reset the revision version to zero. The major version number will not be affected. Examples of changes that will cause an update of the minor version number:
  - addition of nodes and/or attributes which are not required "to make it work" but offer new or extended (optional) functionality which was not present in the previous specification.
  - Changes in the semantics of existing nodes and/or attributes which may include:
    - changes in the interpretation of attribute values
    - changes in the name (not hierarchy) of a node and/or attribute
- **revision** – incremented by one if the specification is upward compatible with the previous specification. Your implementation does not have to be modified in order

to be compliant with the specification. Incrementing the revision version will not affect the major and minor version. Examples of changes that will cause an update of the revision number:

- changes in text which have no influence on the functionality. This will mainly be applicable to non-functional requirements such as legal issues.

## 2.2 | imda/terms

» node is required

The terms node contains definitions used by the organisation to set out the terms of use for their content (metadata, logos, media... ) and allow an organisation to make a declaration about the suitability of their content. This is to allow any solution provider to find this out easily. The general terms and conditions are defined in the `terms/general` node and if you have any specific exceptions to these, we recommend you list these in `terms/exception` node.

### 2.2.1 | imda/terms/general

» node is required, @url is required

The general terms and conditions of use for the organisation's content is presented here. This should cover any content available in this or other linked resources. This is a required node.

The general node contains a `url` attribute which should be an absolute URL to a plain text file that contains your full terms and conditions. The IMDA provides a default set of terms and conditions which is available here, and you can use as the value of the `url` attribute:

```
http://www.imdalliance.org/metadata/docs/termsV1.txt
```



*Best practice guidance: you can use terms and conditions provided by your organisation – however you should really consider following the default terms and conditions provided by the IMDA. These have been created to allow a solution provider to see quickly that your terms and conditions are a previously agreed set from the IMDA. If you provide your own terms and conditions then a solution provider will have to review your content manually – this may delay or disqualify you entirely from that solution provider. You should seriously consider being able to accept the default IMDA terms and conditions.*



*Because the default terms and conditions the IMDA provides is stored on our own servers at the URL above, for a solution it is simple to check to see if the organisation is using them – just look for the URL above.*

### 2.2.2 | imda/terms/exceptions

A textual description of any restrictions or exceptions you may have to your general terms and conditions defined in `terms/general`. This is an optional node.

If you have no exceptions, do not include an `exceptions` node.

If you do, then the `exceptions` node contains a `url` attribute which should be an absolute URL to a plain text file that contains details of these exceptions.



*Best practice guidance: if you have used the default IMDA terms and conditions in `terms/general` then you should have no exceptions – and therefore no `terms/exceptions` node. If you do put exceptions in here then you should understand this will require review of your terms by the solution provider – this may delay or disqualify you entirely from that solution provider.*



*If `terms/general` is the default IMDA terms and conditions and `terms/exceptions` is empty then you can assume that the content of these files is within the default terms and conditions provided by the IMDA – in so far as the IMDA doesn't check all media organisations for compliance with these and cannot be held responsible for any misuse or liability resulting. The responsibility of checking this is between the media organisation and the solution provider.*

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

### 2.2.3 | `imda/terms/privacy`

» node is required, `@url` is required

This node presents the privacy policy that a media organisation needs any third-party using their content to meet or exceed. This should cover any content available in this or other linked resources. This is a required node.

The privacy node contains a `url` attribute which should be an absolute URL to a plain text file that contains your full privacy policy. The IMDA provides a default privacy policy which is available here, and you can use as the value of the `url` attribute:

```
http://www.imdalliance.org/metadata/docs/privacyV1.txt
```



*Best practice guidance: you can use a privacy policy provided by your organisation – however you should really consider following the default one provided by the IMDA. These have been created to allow a solution provider to see quickly that they can meet a known privacy policy level agreed previously with the IMDA. If you provide your own privacy policy then a solution provider will have to review your content manually – this may delay or disqualify you entirely from that solution provider. You should seriously consider being able to accept the default Privacy Policy provided by the IMDA.*



*Because the default Privacy Policy the IMDA provides is stored on our own servers*



at the URL above, for a solution it is simple to check to see if the organisation is using them – just look for the URL above.

## 2.2.4 | imda/terms/privacyexceptions

» node is optional, @url only required if node is provided

A textual description of any restrictions or exceptions you may have to your privacy policy defined in `terms/privacy`. This is an optional node.

If you have no exceptions, do not include a `privacyexceptions` node.

If you do, then the `privacyexceptions` node contains a `url` attribute which should be an absolute URL to a plain text file that contains details of these exceptions.



*Best practice guidance: if you accept the default Privacy Policy provided by the IMDA in `terms/privacy` then you should have no exceptions – and therefore no `terms/privacyexceptions` node. If you do put exceptions in here then you should understand this will require review of your terms by the solution provider – this may delay or disqualify you entirely from that solution provider.*



*If `terms/privacy` is the default Privacy Policy provided by the IMDA and `terms/privacyexceptions` is empty then you can assume that the content of these files is within the default Privacy Policy provided by the IMDA – in so far as the IMDA doesn't check all media organisations for compliance with these and cannot be held responsible for any misuse or liability resulting. The responsibility of checking this is between the media organisation and the solution provider.*

## 2.2.5 | imda/terms/safecontent

» node is required, @url is required

Definition of what 'safe content' is in terms of the media organisation providing the enclosed data and services. This is used in other linked resources to identify brands/services as 'safe content'. This is a required node.

The `safecontent` node contains a `url` attribute which should be an absolute URL to a plain text file that contains your definition of safe content. The IMDA provides a default definition of safe content which is available here, and you can use this as the value for the `url` attribute:

<http://www.imdalliance.org/metadata/docs/safeV1.txt>



*Best practice guidance: you can use the definition of 'safe content' provided by your organisation – however you should really consider following the default definition provided by the IMDA. This has been created to allow a solution provider to see quickly that your definition of 'safe content' is a previously agreed definition from*



*the IMDA. If you provide your own definition, then a solution provider will have to review your content manually – this may delay or disqualify you entirely from that solution provider. You should seriously consider being able to accept the default IMDA definition.*



*Because the default 'safe content' definition that the IMDA provides is stored on our own servers at the URL above, for a solution it is simple to check to see if the organisation is using them – just look for the URL above.*

## 2.3 | imda/organisation

» node is required, @url is required,  
@protect is optional

This node contains information about the location of the XML file that describes the media organisation in greater detail. The @url attribute of this node contains an absolute URL to the location of your organisation xml file. The location may use http or https, and may be protected with HTTP basic authentication – please see 'Protecting your metadata' below. If your url describes a location protected in this manner then you should add the protect attribute with the value "true" to flag to the solution that they will need to provide authentication. The authentication information may be negotiated with the owner of the metadata by contacting the email address described in the imda@owner attribute.

## 2.4 | imda/brands

» node is required, @url is required,  
@protect is optional

This node contains information about the location of the XML file that describes the brands of the media organisation. The url attribute of this node contains an absolute URL to the location of your brands xml file. The location may use http or https, and may be protected with HTTP basic authentication – please see 'Protecting your metadata' below. If your url describes a location protected in this manner then you should add the protect attribute with the value "true" to flag to the solution that they will need to provide authentication. The authentication information may be negotiated with the owner of the metadata by contacting the email address described in the imda@owner attribute.

## 2.5 | imda/transports

» node is required, @url is required,  
@protect is optional

This node contains information about the location of the XML file that describes the transport methods for the brands. The @url attribute of this node contains an absolute URL to the location of your transports xml file. The location may use http or https, and may be protected with HTTP basic

authentication – please see 'Protecting your metadata' below. If your `url` describes a location protected in this manner then you should add the `protect` attribute with the value `"true"` to flag to the solution that they will need to provide authentication. The authentication information may be negotiated with the owner of the metadata by contacting the email address described in the `imda@owner` attribute.

## 2.6 | imda/manifests

» node is optional, if node is used then `@url` is required, `@protect` is optional

This node contains information about the location of the XML file that describes manifests of data regarding particular brands – the data held might be schedule information regarding programming related to a particular brand, or the information required to find editorial details of podcast programming for a particular brand. The `@url` attribute of this node contains an absolute URL to the location of your `manifests.xml` file. The location may use `http` or `https`, and may be protected with HTTP basic authentication – please see 'Protecting your metadata' below. If your `url` describes a location protected in this manner then you should add the `protect` attribute with the value `"true"` to flag to the solution that they will need to provide authentication. The authentication information may be negotiated with the owner of the metadata by contacting the email address described in the `imda@owner` attribute.



*Why 'manifests'? It might sound like an unusual word choice but we wanted something that signified 'a list of things'. Manifests was suggested earlier on, with a number of other options in different languages - but we came back to 'manifests'. Consider the dictionary definition:*

*manifest [noun] ~ a document giving comprehensive details of a ship and its cargo and other contents, passengers, and crew for the use of customs officers. [verb, transitive] to record in such a manifest - e.g. every passenger is manifested at the point of departure.*



*In future releases of this specification the manifests document will be used for 'on-demand' services related to brands.*

## 2.7 | Protecting your metadata

The IMDA strongly recommends that you make all XML documents publicly available – allowing anyone to request the information. However, we understand that some organisations may require to protect some of their data for their own reasons – we ask that organisations always make the root `imda` document publicly available but that they can protect sub-documents using HTTP basic authentication, as noted in the above

details.

If an organisation decides to protect a sub-document then they can use HTTPS to protect the data transferred and use HTTP basic authentication to allow or deny access to the location of the document. If you do this, please make sure that your `imda@owner` email address is valid and allows solutions to contact you about gaining access. The authentication information may be negotiated with the owner of the metadata by contacting the email address described in the `imda@owner` attribute.



*Be aware that a solution may or may not automatically list the organisation's services should you protect your data in this way as the solution will need to contact you and seek agreement on use. This allowance has been provided for organisations who wish to control who has access to their metadata.*

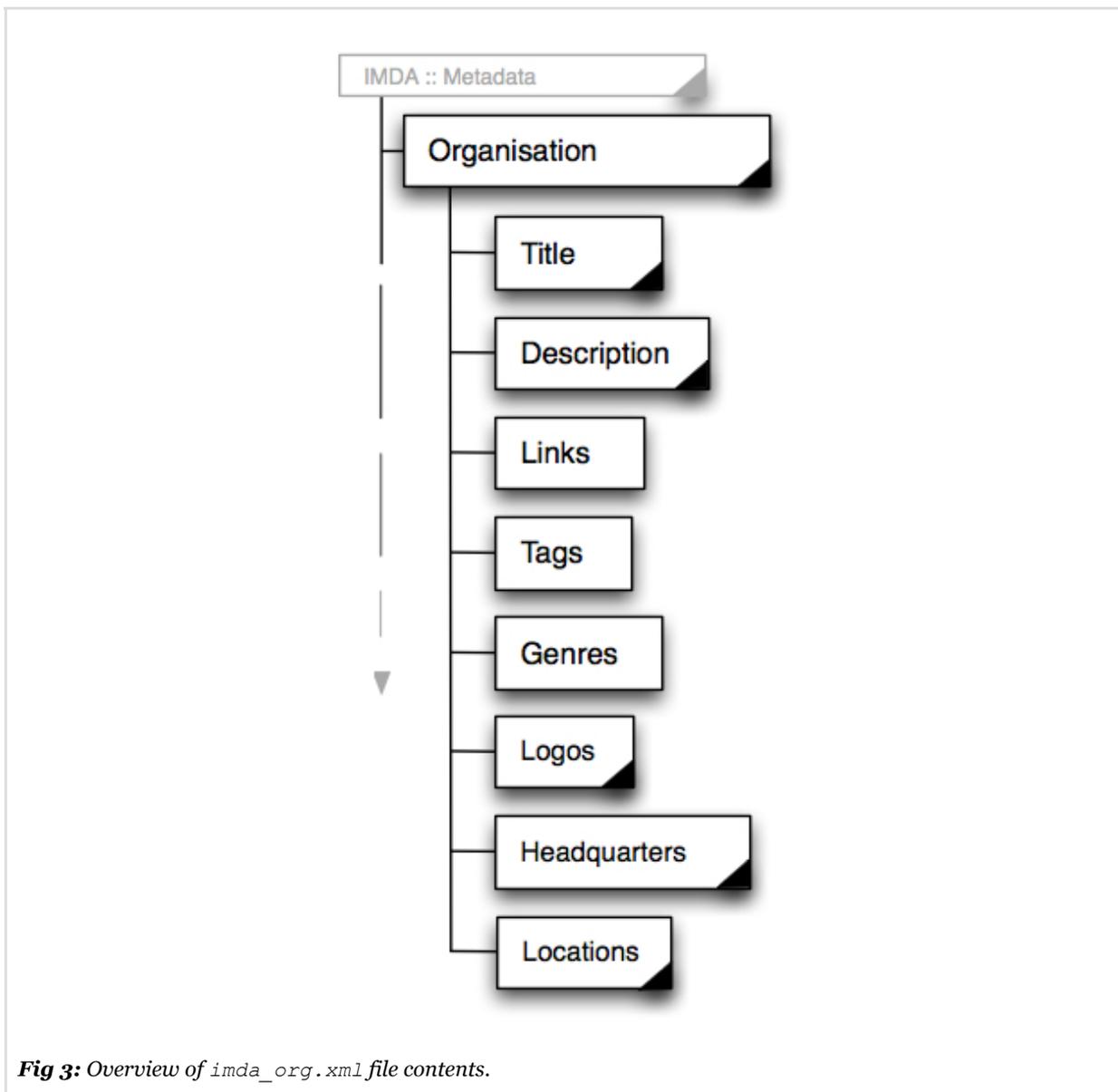
## 3 | Organisation Identification

In order to provide information to a solution about the organisation's identification we provide information in a file referenced from the `imda.xml` root file in `imda/organisation@url`. As we said, we don't mind what this file is called or where it is located – as long as it's publicly available. As an example – our organisation identification file could be here:

```
http://imda.organisation.com/imda_org.xml
```

The contents of this file allow an organisation to describe both data that is used visually by a solution to identify the organisation to a consumer, as well as data that is used purely to create links to show ownership of the services identified in other IMDA specifications.

Let's take a look at the main elements of this file:



**Fig 3:** Overview of *imda\_org.xml* file contents.

Reading from the top of this tree, we see that it contains information about:

*Title ~*

Title of organisation - in multiple lengths (short, medium, long and extended) and with multiple language support.

*Description ~*

Description of organisation - in multiple lengths (short, long and extended) and with multiple language support.

*Links ~*

Allows organisation to associate urls for further information about themselves. e.g. organisation's home page on the web, or their mobile service.

*Tags ~*

Allows organisation to provide free-form tagging and key wording. Each tag may signal its weighting to other tags provided, and future extensions may allow categorisation.

*Genres ~*

Categorise the organisation by Genre (if appropriate).

*Logos ~*

Publish logos that solutions can use to represent this organisation.

*Headquarters ~*

The location of the organisation's head office of a location id (see Locations)

*Locations ~*

Allows an organisation to define various geographical data to be used in this or other files: latitude/longitude points, lat/long polygons or countries.

Here is a quick example of how this could appear in the XML file, again we're using the BBC as the media organisation in this example:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <organisation modified="2009-10-27T150530Z" owner="imda.services@bbc.co.uk">
3   <id crid="//bbc.co.uk/"></id>
4   <title>
5     <short lang="en-GB">BBC</short>
6     <medium lang="en-GB">BBC</medium>
7     <long lang="en-GB">British Broadcasting Corporation</long>
8   </title>
9   <description>
10    <short lang="en-GB">Nulla aliquet mauris nec elit tempor ultrices. Sed faucibus,
11    <long lang="en-GB">Sed faucibus, nunc id dapibus lacinia, velit nisi hendrerit d
12    <extended lang="en-GB">Aliquam et lorem lectus, id aliquet nisi. Mauris non male:
13  </description>
14  <links>
15    <link devicetype="screen" mimetype="text/html" url="http://www.bbc.co.uk/ [http:/
16    <link devicetype="handheld" mimetype="text/html" url="http://www.bbc.co.uk/mobil
17  </links>
18  <tags>
19    <tag weight="10">radio</tag>
20    <tag weight="9">television</tag>
21    <tag weight="5">beeb</tag>
22    <tag>big</tag>
23    <tag>british</tag>
24    <tag>castle</tag>
25  </tags>
26  <genres>
27    <genre vocabulary="tva1.5.1">3.1.1.8.1</genre>
28    <genre vocabulary="tva1.5.1">3.1.1.10.2</genre>
29    <genre vocabulary="tva1.5.1">3.1.1.10.3</genre>
30    <genre vocabulary="tva1.5.1">3.1.1.10.4</genre>
31    <genre vocabulary="tva1.5.1">3.1.1.10.5</genre>
32    <genre vocabulary="tva1.5.1">3.1.1.10.6</genre>
33  </genres>
34  <logos>
35    <logo type="imda_micro_1:1" url="http://imda.bbc.co.uk/imda/logos/org_logomicro1
36    <logo type="imda_mobile_1:1" url="http://imda.bbc.co.uk/imda/logos/org_logomobil
37    <logo type="imda_mobile_3:2" url="http://imda.bbc.co.uk/imda/logos/org_logomobil
38    <logo type="imda_mobile_4:3" url="http://imda.bbc.co.uk/imda/logos/org_logomobil
39    <logo type="imda_tv_4:3" url="http://imda.bbc.co.uk/imda/logos/org_logotv4-3.png
40    <logo type="imda_tv_16:9" url="http://imda.bbc.co.uk/imda/logos/org_logotv16-9.pi
41    <logo type="imda_svg_1:1" url="http://imda.bbc.co.uk/imda/logos/org_logosvg1-1.s
42    <logo type="imda_svg_3:2" url="http://imda.bbc.co.uk/imda/logos/org_logosvg3-2.s
43    <logo type="imda_svg_4:3" url="http://imda.bbc.co.uk/imda/logos/org_logosvg4-3.s
44    <logo type="imda_svg_16:9" url="http://imda.bbc.co.uk/imda/logos/org_logosvg16-9
45  </logos>
46  <headquarters locationid="londonbh"></headquarters>
47  <locations>
48    <location id="ukcountries">
49      <country>GBR</country>
50      <country>IMN</country>
51      <country>GGY</country>

```

Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

```

52     <country>JEY</country>
53 </location>
54 <location id="global">
55     <country>XXX</country>
56 </location>
57 <location id="nonukcountries">
58     <ref id="ukcountries" exclude="true"></ref>
59 </location>
60 <location id="londonbh">
61     <point>51.518329 -0.143638</point>
62 </location>
63 <location id="randomarea">
64     <poly>45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45</poly>
65 </location>
66 <location id="w9">
67     <postal country="GBR">w9</postal>
68 </location>
69 <location id="westlondon">
70     <ref id="w9"></ref>
71     <ref id="w2" exclude="true"></ref>
72     <ref id="randomarea"></ref>
73 </location>
74 </locations>
75 </organisation>

```

Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

In the above example we've tried to keep things simple, to show some of the ways of describing the data. However – the details begin with our root node for this document: `organisation`.

## 3.1 | organisation node

» node is required

The `organisation` node encloses all our metadata that describes our media organisation. This root node contains information in attributes about the last modified date and an email contact address for the owner of the data.

### 3.1.1 | organisation@modified

» attribute is required

This attribute contains a time stamp of when you last updated the file. The data type used here follows ISO 8601 [<http://bit.ly/isodate>] recommendations, and we use 'Z' to denote the timezone 'Zulu Time' (UTC). Do not use a local timezone here for simplicity of comparison. Example – 27th of October 2009, at 15 hrs 5 minutes and 30 seconds Zulu would be:

```
2009-10-27T150530Z
```

### 3.1.2 | organisation@owner

» attribute is required

Define an email address that can be used to contact the department/person looking after these files – this is to allow solution providers to contact them should there be a problem. We recommend you to use an alias here to allow you to route the email to the correct department/person rather than a real name here for privacy reasons. From the

example:

```
imda.services@bbc.co.uk
```

## 3.2 | organisation/id

» node is required

We want to be able to refer to this organisation by a globally unique identifier. This allows solutions to link other services with this organisation that are defined in other XML documents.

### 3.2.1 | organisation/id@crid

» attribute is required

Following the advice from RFC 4078 [<http://bit.ly/rfc4078>] – the value here should be in the form of a Content Reference Identifier (CRID). For examples of producing CRIDs - please see Section 4 of RFC 4078 [<http://bit.ly/rfc4078s4>]. In summary the CRID is based on your main domain name, and there is a method of encoding non-URI compliant character sets into UTF-8 compliant serialisation.

## 3.3 | organisation/title

» node is required

This node allows us to identify how an audience member would see the organisation's name, and it allows us to do this in a set of common sizes that are currently used by devices. It also provides the ability to have alternative language support.

`title` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/title".

## 3.4 | organisation/description

» node is required

This node allows us to provide a description of the organisation that would be visible to an audience member, and it allows us to do this in a set of common sizes that are currently used by devices. It also provides the ability to have alternative language support.

`description` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/description".

## 3.5 | organisation/links

This node allows an organisation to provide links to their own internet accessible services – for example a link to their home page on the web, or perhaps their mobile website.

`links` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/links".

---

### 3.6 | organisation/tags

» node is optional

The ability to provide free-text 'tags' allows the organisation to provide some useful keywords that describe the organisation. `Tags` is an optional node.

`tags` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/tags".

---

### 3.7 | organisation/genres

» node is optional

Categorise your organisation by providing `genres` that describe the overall service of the organisation. (Note that this is different from categorising the services that are provided by your organisation – this can be done in the `brands` document). `Genres` is an optional node.

`genres` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/genres".

---

### 3.8 | organisation/logos

» node is required

This node allows an organisation to provide various logo assets – graphic imagery. So that a solution can provide the correct branding. Since there are a potential infinite number of possible solutions that could use different sizes and formats, we specify here a range of assets that you should provide based on common solutions currently in the industry.

`logos` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/logos".

---

### 3.9 | organisation/headquarters

» node is required

Identify where your organisation resides geographically. The text node of

headquarters should contain an id of a location (see below - 'organisation/locations) of 'point' type only.



*Tip: for larger organisations specify the head office or headquarters.*

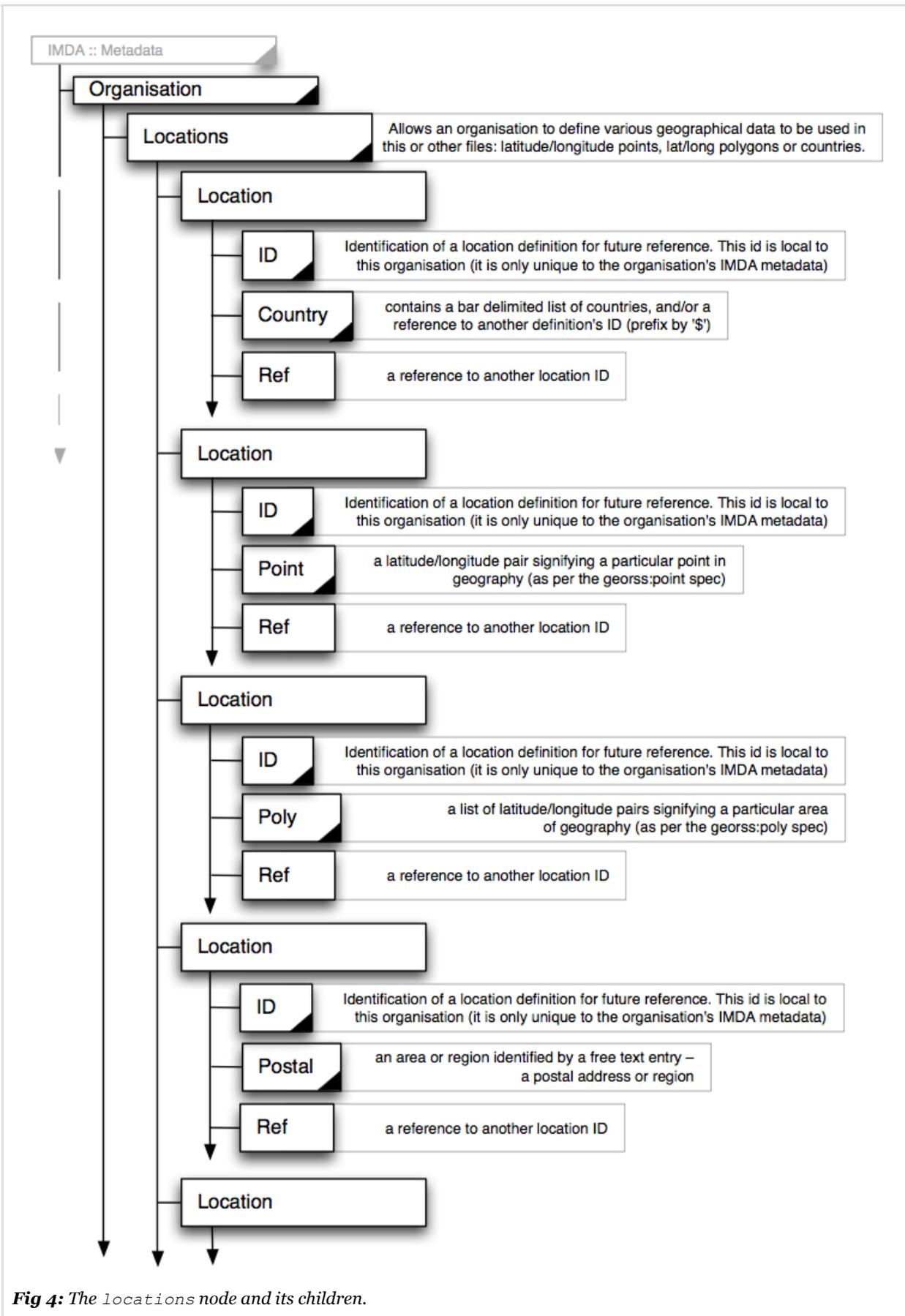


*Why a location id? As you'll see, locations allow you to specify geographical locations and areas once and reuse them again and again in other xml nodes... like this one. This location may be useful to other specifications. If we use the locations specification it means you can specify the location in any of the ways that the locations specification allows...*

## 3.10 | organisation/locations

» node is required, with 1 or more children

As an organisation you may have one or more territories or geographical regions that you target or allow your content to play in. The `locations` allows you to create simple or complex groupings of geographical data which you can refer to in other appropriate xml nodes (like `organisation/headquarters` above). Let's take a quick overview of `locations`:



**Fig 4:** The locations node and its children.

As you can see, the locations node can be made up of a number of location

children, we've shown 4 of the standard `location` children types (or 'location types'). There is a 5th which isn't shown – one that only contains `ref` nodes – this simply allows you to group other locations together. You can have multiple location nodes, of same or different types depending on your needs.

Each location type can contain one or more of the 'country' or 'point' or 'poly' or 'postal' or 'ref' nodes - but, as shown, each location type can only contain the nodes shown... in other words, you can't mix 'country' and 'point' in the same location node. In order to do the latter – you can use that 5th type which only has references.

You clearly need to define at least one `location` as `organisation/headquarters` requires it. However - the minimum number of locations depends on how many brands and transports that you provide. Rather than trying to complete this data early on when implementing this specification, you will need to add locations as you complete Brands and Transports. However, it's worth making a quick note of what will be expected there.

In Brands you will use `location` ids to signal a particular location that a brand is associated with – e.g. BBC Radio 4 is located in London, UK. Also in Brands you will signal the brand's editorial relevance in terms of a `location` – e.g. BBC Radio 4 is globally relevant, but a tightly focused community radio service may only be relevant to a geographic area specified by a polygon.

For Transports, where you are actually specifying the physical streaming information, you will specify two things for every media streaming format and method: any physical restriction you place on a stream, and if the stream itself has a different editorial relevance than its parent brand (thus allowing you to have multiple streams with, for example, rights blanking identified as editorial relevance). More explanation about this in Transports section.

`Locations` allows you, as an organisation, to suggest the appropriate geographical reference for something that can be picked up by a solution/aggregator and used to categorise or search.

The powerful feature of `locations` is that 5th type – which can simply have references to other `location` nodes inside – this is powerful because it allows you to build areas which can then be grouped together. For example - I may use a `country` type of `location` to specify a few countries and give it one ID, then I could specify another `location` that was the negative of the previous `location` ID... meaning that I want all the countries not specified to be this location. Another example for this could be: allowing you to specify multiple `location`'s based on postal zones using the `postal` type and then grouping them together so that you can refer to them using one `location` ID.



**WARNING:** *There is a danger here that you could create an infinite loop – one location referring to another which refers to that location... and so on. So you will*



*need to be careful to check that you haven't inadvertently created this. It is not possible to model this in XML Schema languages, and so we can't automatically validate without further processing. So locations are both powerful, and potentially complex depending on the number of brands and transports you have. For small organisations with basic rights restrictions or global relevance – this should be relatively straightforward. If not, then keep your wits about you!*

Here's a quick example of how you might flex your locations:

```

1  <locations>
2    <location id="ukcountries">
3      <country>GBR</country>
4      <country>IMN</country>
5      <country>GGY</country>
6      <country>JEY</country>
7    </location>
8    <location id="global">
9      <country>XXX</country>
10   </location>
11   <location id="nonukcountries">
12     <ref id="ukcountries" exclude="true" ></ref>
13   </location>
14   <location id="londonbh">
15     <point>51.518329 -0.143638</point>
16   </location>
17   <location id="randomarea">
18     <poly>45.256 -110.45 46.46 -109.48 43.84 -109.86 45.256 -110.45</poly>
19   </location>
20   <location id="w9">
21     <postal country="GBR">w9</postal>
22   </location>
23   <location id="westlondon">
24     <ref id="w9" ></ref>
25     <ref id="w2" exclude="true" ></ref>
26     <ref id="randomarea" ></ref>
27   </location>
28 </locations>

```

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

### 3.10.1 | .../locations/location@id

» attribute is required

The `id` attribute allows you to supply a short name to refer to this location. In other locations we can use this id by prefixing referring to it in the `ref` nodes. Your location ID should be a string of lower case alphanumeric characters with no spaces or punctuation but could include underscore – the following regular expression specifies how an ID can be formed (this is in our XML Schema for validation but is reprinted here for those not using validation):

```
^[a-z0-9_]+$
```

### 3.10.2 | .../locations/location types

Each location node contains an `id` attribute plus one or more of the following sets of nodes:

<b>Type</b>	<b>Possible child nodes of location</b>	<b>Description</b>
country	country, ref	<p>The <code>country</code> node allows you to specify geographical areas using country codes. Country codes specified from 'ISO 3166-1 alpha-3' specification. You can specify groups of countries by having more than one <code>country</code> node.</p> <p>The <code>country</code> node can contain an <code>exclude</code> attribute (with value 'true') to show that this country is to be excluded from the grouping.</p> <p>There is a special country code – 'XXX' – which simply means 'global', this is allowable in the ISO specification we use.</p> <p>For information on <code>ref</code> nodes see the reference location type below.</p>
point	point, ref	<p>The <code>point</code> node should contain a <code>georss:point</code> [<a href="http://bit.ly/georsspoint">http://bit.ly/georsspoint</a>] type - a point specified by latitude and longitude.</p> <p>For information on <code>ref</code> nodes see the reference location type below.</p>
poly	poly, ref	<p>This <code>poly</code> node should contain a <code>georss:poly</code> [<a href="http://bit.ly/georsspoly">http://bit.ly/georsspoly</a>] type - a polygon described using latitude and longitude points.</p> <p>For information on <code>ref</code> nodes see the reference location type below.</p>
circle	circle, ref	<p>This <code>circle</code> node should contain a <code>georss:circle</code> [<a href="http://georss.org/simple#Circle">http://georss.org/simple#Circle</a>] type - a circle is defined by signalling the centre point of the circle (in terms of latitude and longitude) plus the radius of the circle in metres.</p> <p>e.g. "42.943 -71.032 500" - would define a circle whose centre point is located at 42.943 (lat) and -71.032 (long), with a radius of 500 metres.</p> <p>For information on <code>ref</code> nodes see the reference location type below.</p>
postal	postal, ref	<p>This <code>postal</code> node consists of a country attribute and a text node.</p> <p>The attribute should contain a single country code (following 'country' specification above). The text node contains a string that can be interpreted as a postal location inside that country.</p> <p>A solution can look at the text node and interpret it based on the postal addressing standards of the country specified in the attribute.</p>

<b>Type</b>	<b>Possible child nodes of location</b>	<b>Description</b>
reference	ref	<p>Therefore, for example, specifying the country to be 'USA' and the string says "479" then you may interpret this as the common Address format in the United States and calculate it as a partial US Zip Code (in this example Tippecanoe County, Indiana. Another example: country is "GBR" and string says "W9" then we could determine this is common address format to the UK and the string is a partial UK Postcode.</p> <p>For information on <code>ref</code> nodes see the reference location type below.</p> <p>The <code>ref</code> node consists of a text node that contains an id of another location node, this allows you to group location nodes together.</p> <p>For example - you may have a few postal locations, which you can group together using another location so that they can be referred to using that location id.</p>

**Table 1:** The different location types and how to use them.

 *Is geo:rss the best thing to use? We answer some questions in our FAQ section about geographical reference systems. We invite feedback should you find that you are unable to represent locations that you need.*

## 4 | Brands Identification

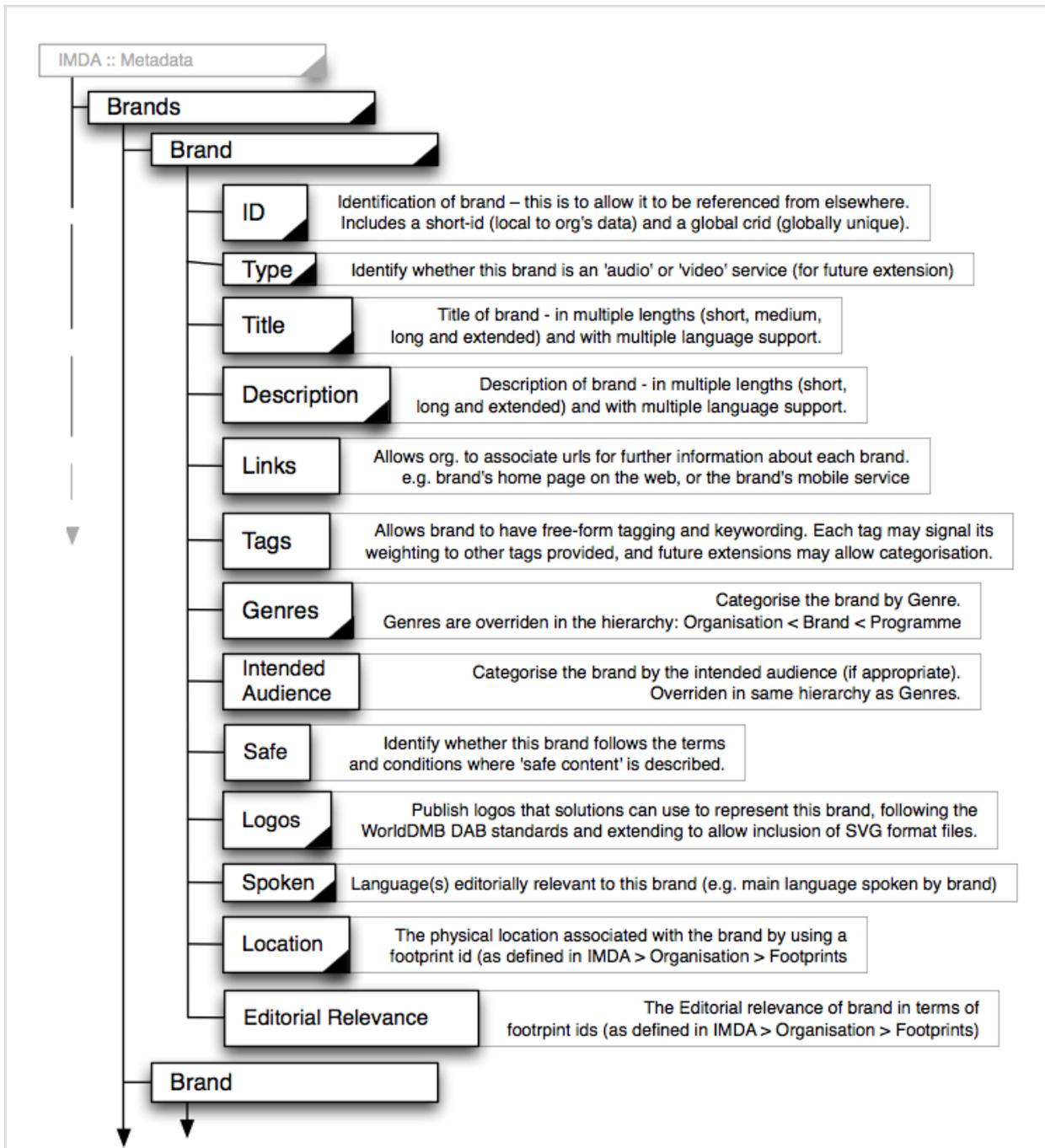
An organisation now needs to present the brands that it has, these are abstracted away from the actual technical information on the streams, and other services, which is held within the transports document. This abstraction lets us keep the editorial information away from the transports and allow us to link services to the brand(s) as needed.

The identification of your brands is made available in a file referenced from the `imda.xml` root file in `imda/brands@url`. As we said, we don't mind what this file is called or where it is located – as long as it's publicly available. As an example – our brands identification file could be here:

```
http://imda.org.uk/imda_brands.xml
```

The contents of this file allow an organisation to describe the data that is used visually by a solution to identify the organisation's brand(s) to a consumer, as well as data that is used purely to create links to show ownership of the services identified in other IMDA specifications.

Let's take a look at the main elements of this file:



**Fig 5:** Overview of *imda\_org.xml* file contents.

Reading from the top of this tree, we see that it contains information about:

- ID* ~  
Identification of brand – this is to allow it to be referenced from elsewhere. This includes a short-id (local to org's data) and a global crid (globally unique).
- Type* ~  
Identify whether this brand is an 'audio' or 'video' service (for future extension)
- Title* ~  
Title of brand - in multiple lengths (short, medium, long and extended) and with multiple language support.

*Description ~*

Description of brand - in multiple lengths (short, long and extended) and with multiple language support.

*Links ~*

Allows org. to associate urls for further information about each brand. e.g. brand's home page on the web, or the brand's mobile service.

*Tags ~*

Allows brand to have free-form tagging and key wording. Each tag may signal its weighting to other tags provided, and future extensions may allow categorisation.

*Genres ~*

Categorise the brand by Genre.

*Intended Audience ~*

Categorise the brand by the intended audience (if appropriate).

*Safe ~*

Identify whether this brand follows the terms and conditions where 'safe content' is described.

*Logos ~*

Publish logos that solutions can use to represent this organisation.

*Spoken ~*

Language(s) editorially relevant to this brand (e.g. main language spoken by brand)

*Location ~*

The physical location associated with the brand by using a location id (as defined in the organisation identification document – organisation/locations )

*Editorial Relevance ~*

The Editorial relevance of brand in terms of location ids (as defined in the organisation identification document – organisation/locations )

Multiple brands are represented by having multiple `brand` nodes. Here is a quick example of how all of this could appear in the XML file, again we're using the BBC as the media organisation in this example:

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <brands orgcrid="crid://bbc.co.uk/" modified="2010-01-25T133530Z" owner="imda.services@
3   <brand id="r1" crid="crid://bbc.co.uk/a/radio1" type="audio">
4     <title>
5       <short lang="en-GB">BBC R1</short>
6       <medium lang="en-GB">BBC Radio 1</medium>
7     </title>
8     <description>
9       <short lang="en-GB">Pellentesque ut massa in est viverra vehicula nec vitae
10      <long lang="en-GB">Nullam quis ligula in libero tempor venenatis et vitae ei
11      <extended lang="en-GB">Etiam fermentum libero at arcu aliquet dictum. Donec
12     </description>
13     <links>
14       <link devicetype="web" mimetype="text/html" url="http://www.bbc.co.uk/radio:
15       <link devicetype="handheld" mimetype="text/html" url="http://www.bbc.co.uk/
16     </links>
17     <tags>
18       <tag>radio</tag>

```

```

19     <tag>radio1</tag>
20     <tag>national</tag>
21     <tag>music</tag>
22 </tags>
23 <genres>
24   <genre vocabulary="tva1.5.1">3.1.1.10.5</genre>
25   <genre vocabulary="tva1.5.1">3.1.1.10.5</genre>
26   <genre vocabulary="tva1.5.1">3.1.4.1</genre>
27   <genre vocabulary="tva1.5.1">3.1.10.4</genre>
28   <genre vocabulary="tva1.5.1">3.6.4</genre>
29 </genres>
30 <intendedaudience>
31   <audience>4.2.1.3</audience>
32   <audience>4.2.2.1</audience>
33   <audience>4.2.2.2</audience>
34   <audience>4.5.3</audience>
35   <audience>4.7.7</audience>
36   <audience>4.7.3</audience>
37 </intendedaudience>
38 <safe>true</safe>
39 <logos>
40   <logo type="imda_micro_1:1" url="http://imda.bbc.co.uk/imda/logos/radio1_lo
41   <logo type="imda_mobile_1:1" url="http://imda.bbc.co.uk/imda/logos/radio1_l
42   <logo type="imda_mobile_3:2" url="http://imda.bbc.co.uk/imda/logos/radio1_l
43   <logo type="imda_mobile_4:3" url="http://imda.bbc.co.uk/imda/logos/radio1_l
44   <logo type="imda_tv_4:3" url="http://imda.bbc.co.uk/imda/logos/radio1_logot
45   <logo type="imda_tv_16:9" url="http://imda.bbc.co.uk/imda/logos/radio1_logo
46   <logo type="imda_svg_1:1" url="http://imda.bbc.co.uk/imda/logos/radio1_logo
47   <logo type="imda_svg_3:2" url="http://imda.bbc.co.uk/imda/logos/radio1_logo
48   <logo type="imda_svg_4:3" url="http://imda.bbc.co.uk/imda/logos/radio1_logo
49   <logo type="imda_svg_16:9" url="http://imda.bbc.co.uk/imda/logos/radio1_log
50 </logos>
51 <spoken lang="en-GB"></spoken>
52 <location refid="londonbh"></location>
53 <editorialrelevance refid="global"></editorialrelevance>
54 </brand>
55 <brand id="r2" crid="crid://bbc.co.uk/a/radio2" type="audio">
56   <title>
57     <short lang="en-GB">BBC R2</short>
58     <medium lang="en-GB">BBC Radio 2</medium>
59   </title>
60   <description>
61     <short lang="en-GB">Pellentesque ut massa in est viverra vehicula nec vitae
62     <long lang="en-GB">Nullam quis ligula in libero tempor venenatis et vitae ei
63     <extended lang="en-GB">Etiam fermentum libero at arcu aliquet dictum. Donec
64   </description>
65   <links>
66     <link devicetype="web" mimetype="text/html" url="http://www.bbc.co.uk/radio:
67     <link devicetype="handheld" mimetype="text/html" url="http://www.bbc.co.uk/i
68   </links>
69   <tags>
70     <tag>radio</tag>
71     <tag>radio2</tag>
72     <tag>national</tag>
73     <tag>music</tag>
74   </tags>
75   <genres>
76     <genre vocabulary="tva1.5.1">3.1.1.10.5</genre>
77     <genre vocabulary="tva1.5.1">3.1.4.1</genre>
78     <genre vocabulary="tva1.5.1">3.1.10.4</genre>
79     <genre vocabulary="tva1.5.1">3.6.4</genre>
80   </genres>
81   <intendedaudience>
82     <audience>4.2.2.3</audience>
83     <audience>4.2.2.4</audience>
84     <audience>4.2.2.5</audience>
85     <audience>4.2.2.6</audience>
86     <audience>4.2.2.7</audience>
87   </intendedaudience>
88   <safe>true</safe>
89   <logos>
90     <logo type="imda_micro_1:1" url="http://imda.bbc.co.uk/imda/logos/radio2_lo
91     <logo type="imda_mobile_1:1" url="http://imda.bbc.co.uk/imda/logos/radio2_l

```

Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

```

92     <logo type="imda_mobile_3:2" url="http://imda.bbc.co.uk/imda/logos/radio2_l
93     <logo type="imda_mobile_4:3" url="http://imda.bbc.co.uk/imda/logos/radio2_l
94     <logo type="imda_tv_4:3" url="http://imda.bbc.co.uk/imda/logos/radio2_logot
95     <logo type="imda_tv_16:9" url="http://imda.bbc.co.uk/imda/logos/radio2_logor
96     <logo type="imda_svg_1:1" url="http://imda.bbc.co.uk/imda/logos/radio2_logor
97     <logo type="imda_svg_3:2" url="http://imda.bbc.co.uk/imda/logos/radio2_logor
98     <logo type="imda_svg_4:3" url="http://imda.bbc.co.uk/imda/logos/radio2_logor
99     <logo type="imda_svg_16:9" url="http://imda.bbc.co.uk/imda/logos/radio2_logor
100    </logos>
101    <spoken lang="en-GB"></spoken>
102    <location refid="londonbh"></location>
103    <editorialrelevance refid="global"></editorialrelevance>
104  </brand>
105 </brands>

```

In the above example we've tried to keep things simple, to show some of the ways of describing the data. However – the details begin with our root node for this document: brands.

## 4.1 | brands node

» node is required

The `brands` node encloses all our metadata that describes our brands. This root node contains information in attributes about the last modified date, an email contact address for the owner of the data and a reference to the organisation that owns the brand(s).

### 4.1.1 | brands@modified

» attribute is required

This attribute contains a time stamp of when you last updated the file. The data type used here follows ISO 8601 [<http://bit.ly/isodate>] recommendations, and we use 'Z' to denote the timezone 'Zulu Time' (UTC). Do not use a local timezone here for simplicity of comparison. Example – 27th of October 2009, at 15 hrs 5 minutes and 30 seconds Zulu would be:

```
2009-10-27T150530Z
```

### 4.1.2 | brands@owner

» attribute is required

Define an email address that can be used to contact the department/person looking after these files – this is to allow solution providers to contact them should there be a problem. We recommend you to use an alias here to allow you to route the email to the correct department/person rather than a real name here for privacy reasons. From the example:

```
imda.services@bbc.co.uk
```

### 4.1.3 | brands@orgcrd

» attribute is required

To show the ownership of these brands, we refer to the `organisation/id@crd` value

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

that we specified in the organisation identification file. This is mainly here for convenience, as it allows a solution provider to pick the file up and immediately the data up appropriately where they already have the rest of your IMDA metadata documents. In our ongoing example, the BBC is the organisation, so we put the crid defined in the BBC's organisation identification document here:

```
crid://bbc.co.uk/
```

## 4.2 | brand

» node is required, 1 or more

There can be one or many of these nodes, each identifying one particular brand from the organisation.

We need to be able to refer to it so we will provide identification in the form of a crid and a short-id. We also need to identify the type of service – is it audio-only or audio+video?

### 4.2.1 | brand@crid

» attribute is required

We want to be able to refer to this brand by a globally unique identifier – this allows solutions to link other services with the brands that are defined in other XML documents and future services.

Following the advice from RFC 4078 [<http://bit.ly/rfc4078>] – the value here should be in the form of a Content Reference Identifier (CRID). For examples of producing CRIDs - please see Section 4 of RFC 4078 [<http://bit.ly/rfc4078s4>]. In summary the CRID is based on your main domain name, and there is a method of encoding non-URI compliant character sets into UTF-8 compliant serialisation.

### 4.2.2 | brand@id

» attribute is required

This is a short-id and an organisation makes no guarantee that it is globally unique. The uniqueness should be local to the organisation. This can be a short-hand, if you prefer, used in the IMDA metadata documents where repeating the full crid isn't necessary as the link is implied by the structure and hierarchy of the metadata documents.

Your short-id should be a string of alphanumeric lower case characters which contain no other punctuation or other symbols apart from '\_' (underscore). The string should follow the following regular expression:

```
^[a-z0-9_]+$
```

### 4.2.3 | brand@type

To identify the type of a brand - whether it is an audio-only or an audio and video service. In this version of the Service Identification document the only valid value for the attribute is 'audio'.

This is merely here to allow future expansion of the specification to other types of services.

In the future the attribute may contain 'audio' or 'video'. For example - a 'radio' brand would be audio-only service, and a TV brand would be video. Currently this specification is predominately for audio-only brands – so it is currently unnecessary to represent brands which are video services – this is for future use.

---

### 4.3 | brand/title

» node is required

This node allows us to identify how an audience member would see the brand's title (or name), and it allows us to do this in a set of common sizes that are currently used by devices. It also provides the ability to have alternative language support.

`title` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/title".

---

### 4.4 | brand/description

» node is required

This node allows us to provide a description of the brand that would be visible to an audience member, and it allows us to do this in a set of common sizes that are currently used by devices. It also provides the ability to have alternative language support.

`description` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/description".

---

### 4.5 | brand/links

» node is optional

This node allows an organisation to provide links relevant to the brand about links to internet accessible services – for example a link to the home page of the brand on the web, or perhaps its mobile website.

`links` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/links".

## 4.6 | brand/tags

» node is optional

The ability to provide free-text 'tags' allows the organisation to provide some useful keywords that describe the brand. `Tags` is an optional node.

`tags` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/tags".

## 4.7 | brand/genres

» node is required

Categorise your brand by providing `genres` that describe it. (Note that this is different from categorising the organisation itself – this can be done in the organisation document). `Genres` is an optional node.

`genres` is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/genres".

## 4.8 | brand/intendedaudience

» node is optional

To allow a solution to understand the type of audience that would perhaps find your brand more suitable then you can optionally categorise by identifying the 'intended audience' of your brand in the `intendedaudience` node.

If you choose to categorise your brand(s) in this way, we have decided to adopt TV-Anytime's Intended Audience specification, specifically version 1.5.1 of their specification.

TV-Anytime is an "association of organizations which seeks to develop specifications to enable audio-visual and other services based on mass-market high volume digital storage in consumer platforms." ~ TV-Anytime website [<http://bit.ly/tvanytime>]

The published document for version 1.5.1 of their specification is referred to as: "ETSI TS 102 822-3-1 V1.5.1 (2009-05)" and the full xml document with the options is available in Appendix A.7 (on page 113). This specification is available from the ETSI website [<http://bit.ly/aty1GW>].

You can have one or more `audience` child nodes of `intendedaudience` and each child should be considered by the solution as additive. Example: specifying several specific age

groups means that the brand is intended by the organisation of each of those groups.

The IMDA has republished the TV-Anytime xml document here for simplicity:  
tva\_IntendedAudienceCS\_v151.xml

Use the Term@TermID reference in the text node of our audience node.

An example of how an intendedaudience node may appear:

```

1 <intendedaudience>
2   <audience>4.2.1.3</audience>
3   <audience>4.2.2.1</audience>
4   <audience>4.2.2.2</audience>
5   <audience>4.5.3</audience>
6   <audience>4.7.7</audience>
7   <audience>4.7.3</audience>
8 </intendedaudience>
```

## 4.9 | brand/safe

» node is required

A simple identification of whether a brand is of 'safe content' as described in your root document. The text node of safe should contain a boolean string - either 'true' or 'false'.



*Use lower case – no spaces or punctuation – otherwise the aggregator/solution provider will assume it is false.*

## 4.10 | brand/logos

» node is required

This node allows an organisation to provide various logo assets – graphic imagery – relevant to the brand. So that a solution can provide the correct branding. Since there are a potential infinite number of possible solutions that could use different sizes and formats, we specify here a range of assets that you should provide based on common solutions currently in the industry.

logos is a node that is shared in several documents – the implementation details are available in the Shared Nodes section titled "[organisation|brands/brand]/logos".

## 4.11 | brand/spoken

» node is required

This node allows you to identify the language(s) editorially relevant to this brand (e.g. main language spoken by brand). The text node of spoken should contain languages as defined in RFC 5646 [<http://bit.ly/rfc5646>]. For multiple languages please provide them bar delimited (e.g. 'sl-rozaj|de-DE' - which would mean Resian dialect of Slovenian and the German used in Germany)

Please note that RFC 5646 obsoletes the following chain of RFC documents: RFC 4646 < RFC 3066 < RFC 1766.

## 4.12 | brand/location

» node is required

Identify where the brand resides geographically. The `refid` attribute of `location` should contain an id of a `organisation/locations/location@id` of 'point' type only (see 'organisation/locations').



*Tip: you should pick a particular spot where the brand has its office or headquarters. Please note that this is distinct from the brand/editorialrelevance (below) which is used to describe the location(s) of where this brand is editorially relevant to.*



*Why a location id? As you'll see, locations allow you to specify geographical areas once and reuse them again and again in other xml nodes... like this one. This location may be useful to other specifications. Locations are defined in organisation/locations.*

## 4.13 | brand/editorialrelevance

» node is optional

Identify the editorial relevance of the brand to particular regions by using any type of `organisation/locations/location@id` in the `refid` attribute of `editorialrelevance`. (see 'organisation/locations').

This is useful to solutions which may, at first or on request, present the audience with content which is somehow more relevant to them geographically. For example – if an audience member in Singapore is using a solution then they may be looking for local news services, and so the solution may use the editorial relevance provided here to help it decide whether this content should be shown nearer the top of the results. However this doesn't mean that your content wouldn't be presented, it may just mean that it appears further down the results as it is perhaps to be less relevant.

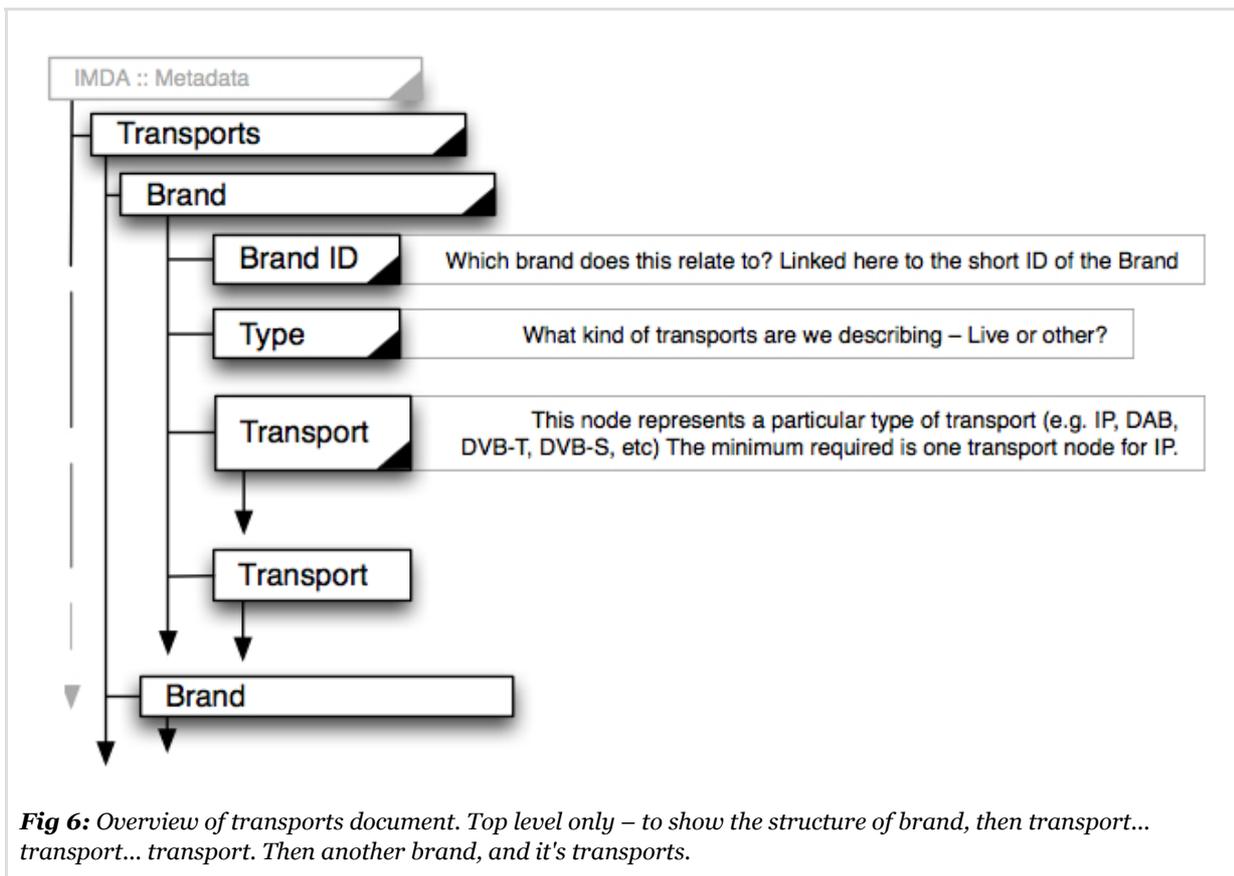
## 5 | Transports Identification

Once the brands themselves have been identified we need to provide information about how a solution can allow an audience member to be able to play back the brand's content. In other words – to allow a solution to provide the correct stream for the organisation's brand. We will call the various technical ways you can get the content as 'transports'.



*The transports document is perhaps one of the more complicated specifications, yet it is hopefully powerful and scalable to future editions of the IMDA Service Identification documents. It should also be noted that this edition of the Service Identification docs assumes your brand's transports describe live, or simulcast, content – we are currently working on how to represent on-demand services, and downloads (e.g. podcasts)*

Each brand identified in the Brands document can have one or more transports. A quick overview of this is as follows:



**Fig 6:** Overview of transports document. Top level only – to show the structure of brand, then transport... transport... transport. Then another brand, and it's transports.

In this overview we have kept the `transport` node without it's various children just to keep the overview simple. Using our favourite example Organisation, the BBC, here's an example of how the shell of the document might look (the content of the transport nodes is left out for simplicity at this stage):

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <transports orgcrid="crid://bbc.co.uk/" modified="2010-01-25T133530Z" owner="imda.servic
3    <brand id="r1" type="live">
4      <transport type="ip_direct">
5        <!-- .... -->
6      </transport>
7      <transport type="ip_indirect">
8        <!-- .... -->
9      </transport>
10   </brand>
11   <brand id="r2" type="live">
12     <transport type="ip_direct">
13       <!-- .... -->
14     </transport>
15     <transport type="ip_indirect">
16       <!-- .... -->
17     </transport>
18   </brand>
19 </transports>

```

The first thing you should notice is that the root `transports` node looks a lot like the same as our `brands` document – with an `orgcrid` attribute to link it to our organisation, a `modified` timestamp and an `owner` email address. The direct descendents of `transports` are `brand` nodes which merely allow us to group subsequent technical transport information together by the `brand` they refer to (from the `brands` document).

That technical transport information is then identified by `transport` nodes.

Let us now look at the structure in detail.

## 5.1 | transports node

» node is required

The `transports` node encloses all our metadata that describes the ways in which a solution could receive the content of our brands. This root node contains information in attributes about the last modified date, an email contact address for the owner of the data and a reference to the organisation that owns the transport(s).

### 5.1.1 | transports@modified

» attribute is required

This attribute contains a time stamp of when you last updated the file. The data type used here follows ISO 8601 [<http://bit.ly/isodate>] recommendations, and we use 'Z' to denote the timezone 'Zulu Time' (UTC). Do not use a local timezone here for simplicity of comparison. Example – 27th of October 2009, at 15 hrs 5 minutes and 30 seconds Zulu would be:

```
2009-10-27T150530Z
```

### 5.1.2 | transports@owner

» attribute is required

Define an email address that can be used to contact the department/person looking after these files – this is to allow solution providers to contact them should there be a problem. We recommend you to use an alias here to allow you to route the email to the correct department/person rather than a real name here for privacy reasons. From the example:

```
imda.services@bbc.co.uk
```

### 5.1.3 | transports@orgcrid

» attribute is required

To show the ownership of these brands, we refer to the `organisation/id@crd` value that we specified in the organisation identification document. This is mainly here for convenience, as it allows a solution provider to pick the file up and immediately the data up appropriately where they already have the rest of your IMDA metadata documents. In our ongoing example, the BBC is the organisation, so we put the `crd` defined in the BBC's organisation identification document here:

```
crd://bbc.co.uk/
```

## 5.2 | transports/brand

» node is required, same number of nodes as brands described in brands document

This node is merely a tool to allow grouping all technical information for a particular brand together. You show the link to the brand by using the `brand@id` attribute and we identify the type of service these transports provide for this brand by entering the `brand@type` attribute. These are as follows:

### 5.2.1 | transports/brand@id

» attribute is required

This attribute should contain the `id` of the brand from your brands document (specifically: `brands/brand@id`) – this is the 'short' id, rather than the full `crid`. For the purposes of showing a link here, a local id is fine rather than the globally unique `crid`.

### 5.2.2 | transports/brand@type

» attribute is required

This attribute allows you to identify the type of transports you are going to define. In this edition of the Service Identification documents we have only 1 currently valid value for `type`:

<i>type</i>	<i>description</i>
live	The live type identifies services of brands which are considered to be simulcast or live services of a linear audio or video service – for example a programmed Television channel, a Radio station/network.

**Table 2:** Current allowable values of `type` for brands.

***The following table is our proposal for future editions of these Service Identification document and we print our initial ideas now to provoke feedback and comment from media organisations and solutions alike. As an organisation or a solution provider you not currently use these types in your document – this will likely change and coalesce into future Guideline's edition.***

<i>type</i>	<i>description</i>
live	The live type identifies services of brands which are considered to be simulcast or live services of a linear audio or video service – for example: a programmed Television channel, a Radio station/network.
ondemand	The ondemand type identifies services of brands which are considered to be provided on request by a user – for example: a 30 minute programme available upon request by the user.

**Table 3:** Table provided as an **example of possible future values** of *type* for brands please providing feedback and comment on these future types.

## 5.3 | transports/brand/transport

» node is required, 1 or more depending on number of transport types

These are used to group technical information about a specific type of transport mechanism together. In this current set of IMDA Service Identification documents we define the first three types of transport that are of use immediately to organisations and solutions:

<i>type</i>	<i>description</i>
<code>ip_direct</code>	a listing of technical information that would allow a solution to select and play a stream through a direct connection. e.g. Shoutcast-compatible HTTP delivered MP3 stream
<code>ip_indirect</code>	a listing of technical information that presents how a solution may link to online services which themselves would allow the audience member to enjoy the brand's content. e.g. XHTML compatible web page accessed via HTTP and appropriate to desktop Web Browsers
<code>broadcast</code>	a listing of technical information that would allow a solution to select and play a linear programmed service. (e.g. Soma FM's Groove Salad, or BBC's Radio 4 Extra).

**Table 4:** Types of *transport* nodes.

We'll now look at these in detail.

### 5.3.1 | transport@type : ip\_direct or broadcast

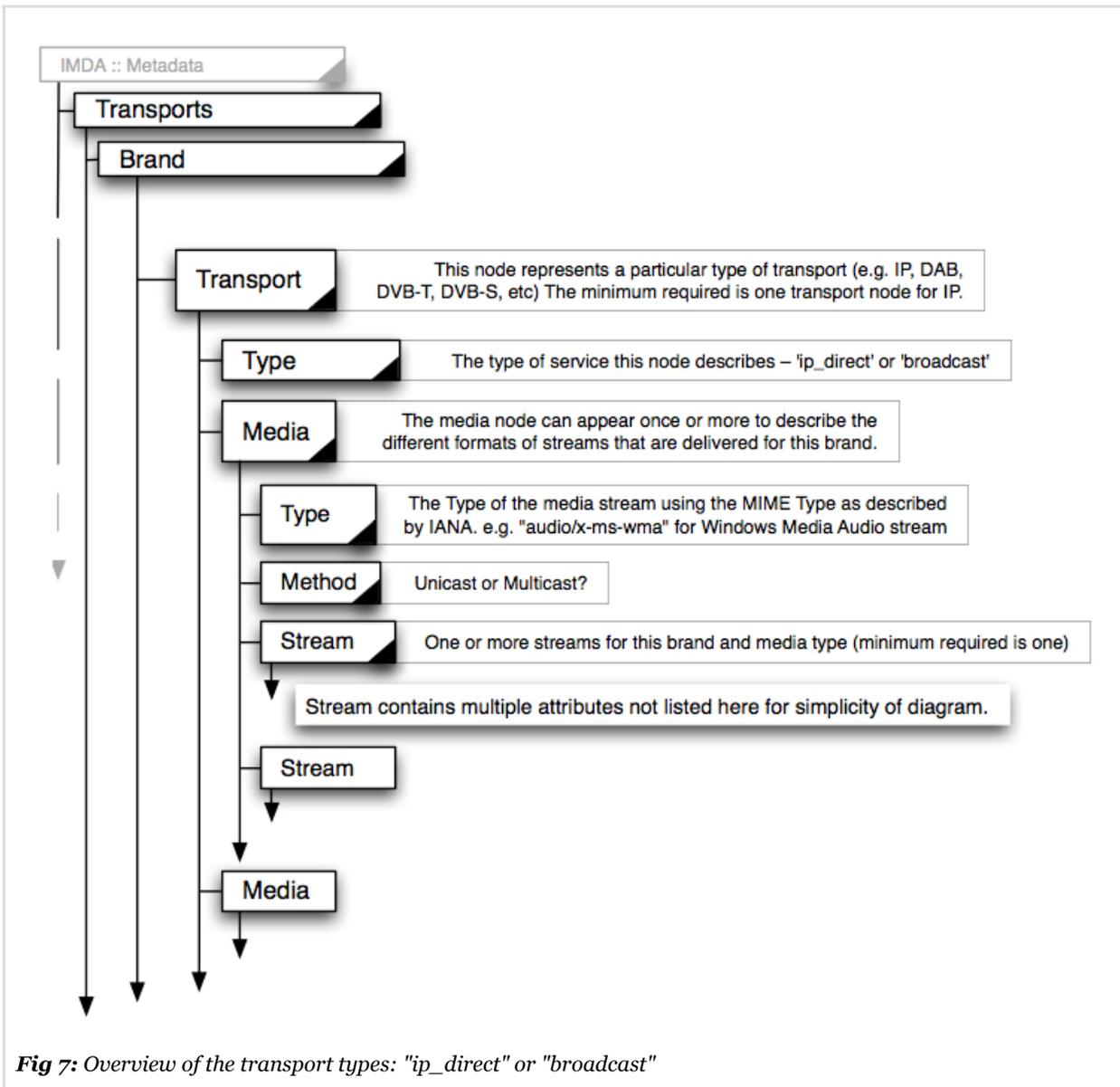
The transport type `ip_direct` or `broadcast` are very similar, allowing for us to keep a consistent method of grouping information together to describe technical information that can be used by a solution to select the most appropriate ways of connecting to and playing back content associated with the brand where the content is available via the public internet ('`ip_direct`') or via a broadcast medium ('`broadcast`').

The children of this `transport` type are one or more `media` nodes.

These `media` nodes are groupings of streaming service types (e.g. Windows Media Services).

Children of `media` nodes are one or more `stream` nodes which actually contain the technical information that would allow a solution to connect and play back.

A quick overview of this follows:



**Fig 7:** Overview of the transport types: "ip\_direct" or "broadcast"

Here is an example of how this node might appear:

```

1  <brand id="r1" type="live">
2    <transport type="ip_direct">
3      <media type="wms">
4        <stream physical="$uk" relevance="$global" url="http://bbc.co.uk/radio/liste
5        <stream physical="$intl" relevance="$global" url="http://bbc.co.uk/radio/lis
6      </media>
7      <media type="real">
8        <stream physical="$uk" relevance="$global" url="http://bbc.co.uk/radio/liste
9        <stream physical="$intl" relevance="$global" url="http://bbc.co.uk/radio/lis
10     </media>
11     <media type="3gp">
12       <stream physical="$uk" relevance="$global" url="http://www.bbc.co.uk/mobile/
13     </media>
14   </transport>
15   <transport type="broadcast">
16     <media type="dab">
17       <stream physical="$uk" url="dab://e1.ce15.c221.0" audiocodec="mp2" audiochan
18     </media>
19   </transport>
20   <transport type="broadcast">
21     <media type="fm">
22       <stream physical="$uk" url="fm://0.d3a6.1007.de0" audiocodec="mp2" audiochan

```

```

23     </media>
24 </transport>
25 <transport type="broadcast">
26   <media type="drm">
27     <stream physical="$uk" url="drm://d407" audiocodec="mp2" audiochannels="2"><
28   </media>
29 </transport>
30 </brand>
    
```

We will now describe the detail of media and stream nodes:

### 5.3.1.1 | transport/media

» nodes is required

To allow grouping streaming service types together our `media` nodes contain a single attribute – `type`. This can contain one of the following values:

<i>type</i>	<i>description</i>
wms	Windows Media Services / Server
shout	Shoutcast-compatible or IceCast-compatible services
real	RealMedia 'cook', G2, etc. Delivered from Real/Helix servers.
3gp	3GPP compatible streaming services.
dab	WorldDMB specified DAB/DAB+ delivery.
drm	Digital Radio Mondial services.
dmb	WorldDMB specified DMB delivery.
dvb-s	DVB satellite delivery of services
dvb-t	DVB terrestrial delivery of services
dvb-c	DVB cable delivery of services
hdradio	Terrestrial 'HDRadio' services
fm	Audio service delivered via frequency modulated analog signal

**Table 5:** Enumeration of possible values for the `type` attribute of `media` nodes .

These types are the most common right now that fit with the IMDA's Technical profiles, we will be adding further types here as new editions of this specification are provided – please get involved with the IMDA and provide feedback to help set how new editions are formed.

 *Why didn't we include Flash Media Server delivered services here, given how common they are? The answer is simply that they are not a priority in this edition, and require further discussion about how to correctly map the required information to allow a technical solution to connect to them. For example – many FMS services vary in one way or another, including having digital rights management that differs from one media organisation to the next.*

***This is clearly an area where we will have more streaming service types in the future – please help us shape future Service Identification document***

**editions by providing feedback and comment about this area.****5.3.1.2 | transport/media/stream**

» nodes is required

The stream node contains much technical information provided by the organisation to help solutions provide or pick the most appropriate stream for the brand. As an organisation you may find that you only need to list one stream node as a child of media – the ability to list multiple streams is required when you have multiple different encode or delivery profiles.

For example – you may have two Windows Media streams at two different bitrates as unique streams (no IntelliStream or Multi-bitrate streaming). Or you may have two streams that have the same encoding profile but are delivered from a different server farm in different locations (geographical areas).

There are quite a few attributes possible within `stream`, but it is important to understand that there are only two required as a minimum: `@url`, `@audiocodec`

The rest either have defaults, or you provide a value for an attribute to provide guidance to a solution.



*Best practice guidance: note that the more optional attributes that you complete, the more likely a solution will be able to easily add you. If you only complete the minimum, the solution will have to investigate your streaming techniques. By completing more of the optional attributes you will help the solution to correctly list you. Some media organisations may not have all this information to hand as they may use a third-party to actually do their streaming for them – if you are one of those organisations we would encourage you to speak to your third-party and show them the optional attributes here and ask them to help you provide more information. Remember – some solutions will balance the work they need to do against the need to list you.*

The attributes within our stream nodes are as follows:

**5.3.1.2.1 | .../stream@physical**

» attribute is optional if transport type is not broadcast, required otherwise

The physical availability of the stream – a backreference to a location id (see `organisation/locations`). The location cannot be a 'point' type location (see "locations/location types"). This is an optional attribute, unless this is a `broadcast` transport type then it is a required attribute. If empty or no attribute (implied) then assume no restriction.



*Note for an organisation this isn't a replacement for managing your rights on your servers, but merely to allow a solution to present appropriate services to a user – if they are not physically going to be able to play it, then don't offer it. To be explicitly clear: this means that a solution may not list you to the audience.*

**5.3.1.3 | .../stream@relevance**

» attribute is optional

The editorial relevance of the stream - a backreference to a location id (see `organisation/locations`). This is an optional attribute. If empty or no attribute (implied) then assume global relevance. Relevance is an opportunity for an organisation to offer guidance to the solution about any editorial relevance issues on a geographic basis.

**5.3.1.4 | .../stream@url**

» attribute is required

Required. Provide the URL where a metafile or stream is exposed that can be 'played' by solutions, or provide a link that describes how to receive a broadcast service. The contents of this attribute are dependent on the `transports/brand/transport@type` value, and the following sections identify the appropriate rules you should follow to complete the `@url` attribute:

**5.3.1.4.1 | @url values where transport@type is ip\_direct or ip\_indirect**

The `@url` attribute should contain the public internet URL of the stream, page or playlist that allows playback of the brand. For example:

```
http://bbc.co.uk/radio/listen/live/r1.asx
```

**5.3.1.4.2 | @url values where transport@type is broadcast**

The `@url` attribute should contain a URL, or similar identifier, which signals to the aggregator or solution how to locate the broadcast within the broadcast standards specified in the `@media` attribute of the parent `transport` node. For many broadcast services there isn't an existing standard for identifying services in a similar way, so we look to providing a URL made up of the media type as the 'protocol' and then identifiers within the broadcast service itself which identify the particular media associated with the brand.

Since there is specific advice depending on the value of the `@media` attribute we have detailed this in the following table:

<i>@media</i>	<i>@url advice</i>
dab	<p>Use the following URL recipe:</p> <pre>dab://scids.sid.eid.ecc</pre> <p>Where the individual elements are specific to DAB/DAB+ broadcasting as defined by the WorldDMB specifications. (scids - Service Component Identifier; sid - Service Identifier; eid - Ensemble Identifier; ecc - Extended Country Code)</p>

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

<p><b>@media</b></p> <p>drm</p>	<p><b>@url advice</b></p> <p>Use the following URL recipe:</p> <pre>drm://&lt;sid&gt;.(&lt;drm&gt; &lt;amms&gt;)</pre> <p>Where the individual elements are specific to DRM broadcasting as defined by the WorldDMB specifications. (sid - Service Identifier)</p>
<p>dmb</p>	<p>Use the following URL recipe:</p> <pre>dmb://scids.sid.eid.ecc</pre> <p>Where the individual elements are specific to DMB broadcasting as defined by the WorldDMB specifications. (scids - Service Component Identifier; sid - Service Identifier; eid - Ensemble Identifier; ecc - Extended Country Code)</p>
<p>dvb-s or dvb-t or dvb-c</p>	<p>ETSI TS 102 851: "Digital Video Broadcasting (DVB); Uniform Resource Identifiers (URI) for DVB Systems"</p>
<p>hdradio</p>	<p>Use the following URL recipe:</p> <pre>hdradio://&lt;tx&gt;.&lt;cc&gt;</pre> <p>Where the individual elements are specific to 'HD Radio' broadcasting as defined by iBiquity Digital's specifications. (tx - transmitter identifier; cc - country code)</p>
<p>fm</p>	<p>Use the following URL recipe:</p> <pre>fm://&lt;freq&gt;.&lt;pi&gt;.(&lt;ecc&gt; &lt;country&gt;)</pre> <p>Where the individual elements are specific to FM radio broadcasting. (ecc - Extended Country Code, the broadcast RDS ECC code, concatenated with the first character of the broadcast RDS PI code (country code), must be supplied if available; country - ISO 3166-1 alpha 2 country code; pi - programme identifier; frequ - the frequency on which the service is received.)</p>

**Table 6:** Enumeration of possible values for the `url` attribute depending on the `media` attribute values.

**5.3.1.5** | `.../stream@min-bitrate`

» attribute is optional

The value of `min-bitrate` should be in bits per second (bps). Specify the lowest logical bit rate of the stream here. This is to allow solutions to offer stream playback where the bandwidth on the client may not be enough for one stream, but is for another – e.g. you

might offer two streams one at a high bitrate, the other at a low bitrate. Note that `min-bitrate` and `max-bitrate` (below) allow you to show the min and max for a multi-bitrate stream, if the stream is only one bitrate then the min and max should be entered as the same. If the stream uses a variable codec, please specify the average value or the minimum value.

This is an optional attribute. You must either specify `min-bitrate` *and* `max-bitrate` - or neither.

Example: 131072

### 5.3.1.6 | `.../stream@max-bitrate`

» attribute is optional

The value of `max-bitrate` should be in bits per second (kbps). Specify the highest logical bit rate of the stream here. This is to allow solutions to offer stream playback where the bandwidth on the client may not be enough for one stream, but is for another – e.g. you might offer two streams one at a high bitrate, the other at a low bitrate. Note that `min-bitrate` (above) and `max-bitrate` allow you to show the min and max for a multi-bitrate stream, if the stream is only one bitrate then the min and max should be entered as the same. If the stream uses a variable codec, please specify the peak value.

This is an optional attribute. You must either specify `min-bitrate` *and* `max-bitrate` - or neither.

Example: 131072

### 5.3.1.7 | `.../stream@audiosamplerate`

» attribute is optional

Specify the audio sample rate, in Hertz, of the stream's audio component.

This is an optional attribute.

Example: 44100

### 5.3.1.8 | `.../stream@protocol`

» attribute is optional

The value of the protocol attribute should be the protocol or protocols used by the streaming servers for this stream. You may list one or more of the following values, where you list multiple values use a '|' (bar) to delimit the list.

In this edition of the specification, the values are: `http`, `rtsp`

Please note that the order of protocols in the list does not convey any meaning to the solution. The solution determines which protocols to use, and it may be limited by the protocols available to it.

For example – a Windows Media stream may be available via RTSP and HTTP protocols so your protocol value would be: `http|rtsp`

This is an optional attribute. If empty or omitted, the solution would need to test which protocols are available.

**5.3.1.9** | `.../stream@method`

» attribute is optional

The method of the stream is merely there to allow you to distinguish unicast streams from multicast streams – and so the possible values here are either unicast or multicast.

This is an optional attribute. If empty or omitted the implied value is `unicast`.

**5.3.1.10** | `.../stream@audiochannels`

» attribute is optional

The value of `audiochannels` is an integer that corresponds to the number of audio channels in the stream – for example, if the stream is mono audio this value would be '1', for stereo audio it would be '2'.

This is an optional attribute. If empty or omitted the implied value is 2.

**5.3.1.11** | `.../stream@audiocodec`

» attribute is required

Required. The `audiocodec` attribute is the IMDAs attempt to create a shared vocabulary around codecs used for audio streaming, or the audio component of an a/v stream.

<i>value</i>	<i>description</i>
wma9	Windows Media Audio v9
wma9.1	Windows Media Audio v9.1
wma9pro	Windows Media Audio 9 Pro
wma10pro	Windows Media Audio 10 Pro
wma10lossless	Windows Media Audio 9 Lossless
wmavoice	Windows Media Audio Voice
realpcj	RealAudio 1
real28_8	RealAudio 2
realdnet	RealAudio 3
realsipr	RealAudio 4/5
realcook	RealAudio 6 (cook and G2 codec variants)
realraac	RealAudio 9
realralf	RealAudio 10 (RealAudio lossless)
3gpamr	AMR (from 3GPP)
3gpamrwb	AMR-WB (from 3GPP)
vorbis	Ogg Vorbis

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

<i>value</i>	<i>description</i>
mp3	MPEG-1 Audio Layer 3
aac-lc	Advanced Audio Coding (AAC) – part of the aacFamily
he-aacv1	High Efficiency AAC version 1 – part of the aacFamily
he-aacv2	High Efficiency AAC version 2, which may include Spectral Band Replication (SBR) and Parametric Stereo (PS) – part of the aacFamily
mp3s	MPEG-1 Audio Layer 3 with multi-channel (5.1) capabilities – aka. MP3 Surround
mp3hd	MPEG-1 Audio Layer 3 encoding giving a 100% (lossless) replication of input – aka. MP3 Lossless
mp2	MPEG-1 Audio Layer 2
mpsaac-lc	MPEG Surround with AAC-LC
mpshe-aacv1	MPEG Surround with HE-AAC version 1
hd-aac	High Definition AAC Lossless
nocodec	This only makes sense where the transport@type is broadcast - where the delivery transport used for the broadcast does not have a logical 'codec'. For example FM uses an analog technique for delivering audio via broadcast.

**Table 7:** Allowed values of *audiocodec* attribute.

### 5.3.1.12 | stream/radiodns

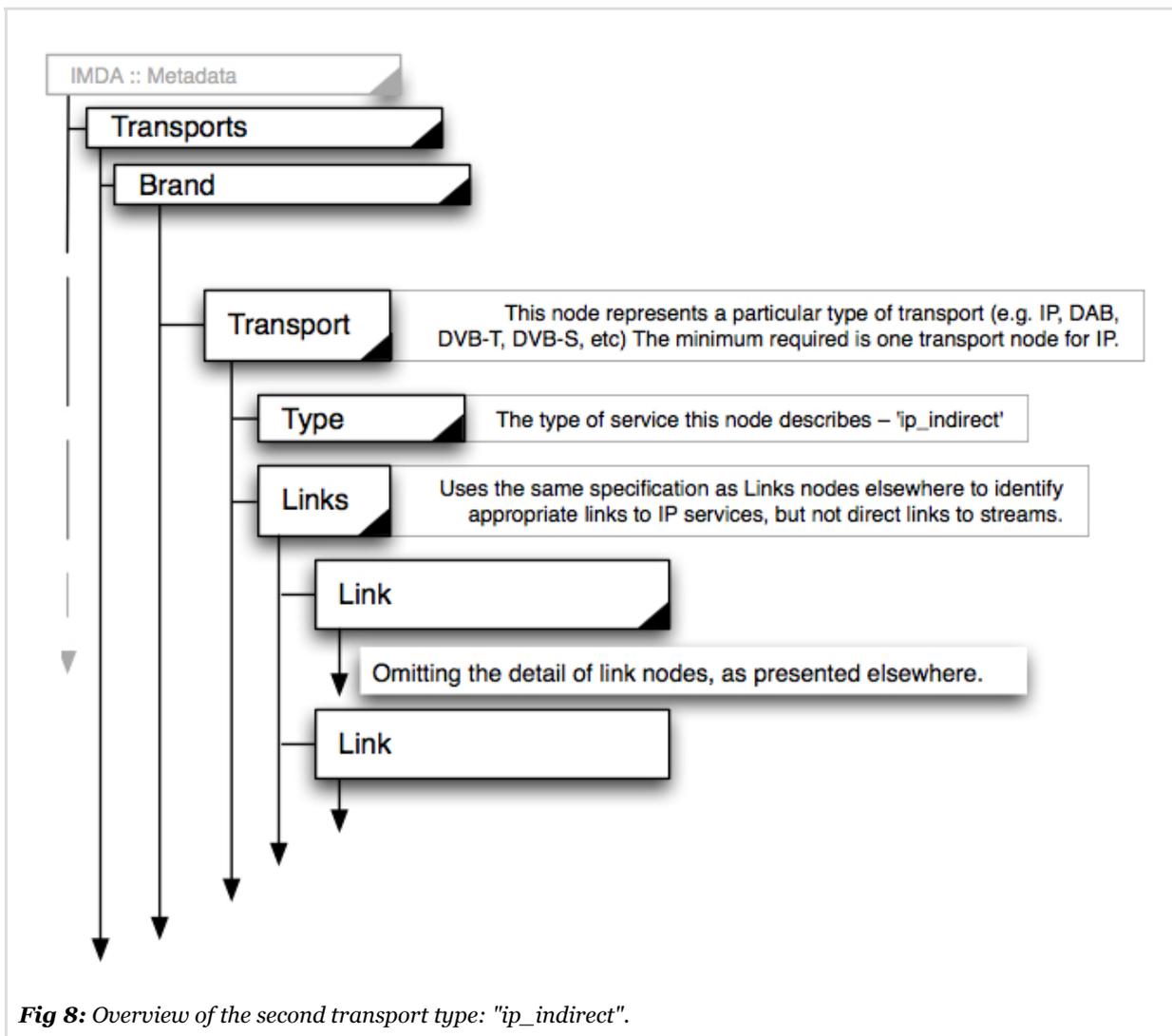
» node is optional

The `radiodns` optional node is provided for media organisations who provide services as defined by the RadioDNS Organisation. For further information on the use of this node please see the section on 'RadioDNS'.

### 5.3.2 | transport@type : ip\_indirect

This should be considered as a secondary transport type. We will use it to describe technical information that can be used by a solution to offer public internet accessible services offered by the brand which may not allow a direct connection and play back via a streaming services. For example – this allows an organisation to offer alternative ways to experience the brand's content on the internet where the service does not lend itself to direct connection by the solution.

There is only one child node type here and it simply uses our shared `links` node as described in '[organisation|brands/brand]/links'.



Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

Here is an example of this transport type, again using the BBC to aid in understanding:

```

1 | <transport type="ip_indirect">
2 |   <links>
3 |     <link devicetype="screen" mimetype="text/html" url="http://www.bbc.co.uk/iplayer/i
4 |     <link devicetype="handheld" mimetype="text/html" url="http://www.bbc.co.uk/mobile,
5 |   </links>
6 | </transport>

```

In this example we use a `links` node to describe a couple of services – one for 'screen' and one for 'handheld'. The `url` attributes contain links to the BBC's web-based iPlayer service formatted for 'screen' and 'handheld'. Because the `links` node, and its children, are defined elsewhere we won't repeat the detail here.

### 5.3.3 | transport@type : the future

***Please provide feedback and comment if there are any additional transport types, or their media types, you would like to see supported in the future.***

## 5.4 | transport/brand/radiodns

» node is optional

The `radiodns` optional node is provided for media organisations who provide services as defined by the RadioDNS Organisation. For further information on the use of this node please see the section on 'RadioDNS'.

## 5.5 | Additional features

This section deals with additional features a Media Organisation may expose in relation to the transports for their brands. They are entirely optional and merely allow a Media Organisation to expose further information. Appearance of these features in the Service Identification does not necessarily mean that an Aggregator, Manufacturer or solution would use this information - it is entirely up to the popularity of the feature whether it is supported.

### 5.5.1 | RadioDNS

Website: [radiodns.org](http://radiodns.org) [<http://bit.ly/radiodns>]

» node is optional, attributes inside node are required if node is used

**"RadioDNS** is a collaborative project to enable the convergence of radio broadcasting and IP-delivered services. It aims to significantly enhance the experience of radio listening using scalable and resilient broadcast technology in tandem with additional information via IP." ~ radiodns.org

For more information on the RadioDNS concepts we refer you to the website. Media Organisations are able to expose RadioDNS functions by signalling them as either:

1. a child node of `transports/brand` – to signify that the RadioDNS details specified are related to the entire `brand`. (See Example A below)
2. a child node of `transports/brand/transport/media/stream` nodes – to signify that the RadioDNS details specified are related to the particular `stream`. (See Example B below)

The RadioDNS node that we provide in the IMDA Service Identification is:

```
<radiodns fqdn="a.fully.qualified.domain.name" sid="serviceid" />
```

We imagine that a fairly simple RadioDNS provision by a Media Organisation would use

the 1st option above – making a single radiodns node a child of transports/brand for each brand node.

Example A – as a child of transports/brand :

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <transports orgcrid="crid://organisation.com/" modified="2010-01-25T133530Z" owner="imda/
3   <brand refid="r1" type="live">
4     <transport type="ip_direct">
5       <media type="wms">
6         <stream url="http://organisation.com/live.asx [http://organisation.com/live
7       </media>
8     </transport>
9     <radiodns fqdn="a.fully.qualified.domain.name" sid="serviceid" ></radiodns>
10  </brand>
11 </transports>

```

Example B – as a child of transports/brand/transport/media/stream :

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <transports orgcrid="crid://organisation.com/" modified="2010-01-25T133530Z" owner="imda/
3   <brand refid="r1" type="live">
4     <transport type="ip_direct">
5       <media type="wms">
6         <stream url="http://organisation.com/live.asx [http://organisation.com/live
7           <radiodns fqdn="a.fully.qualified.domain.name" sid="serviceid" ></ra
8         </stream>
9       </media>
10    </transport>
11  </brand>
12 </transports>

```

Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

For details about values for the fully qualified domain name attribute (`fqdn`) and the service id attribute (`sid`) please examine the specifications provided by the RadioDNS organisation.

RadioDNS specification available here: <http://radiodns.org/documentation/> [<http://radiodns.org/documentation/>] (RDNSo1: Technical Specification, RDNSo2: Administration and Governance, RDNSo3: Potential applications and use cases)

## 5.5.2 | Your feature here!

If you have information on an additional feature which exposes more information about a Media Organisations' brands or transports – please get in touch and provide feedback.

## 6 | Manifests Identification

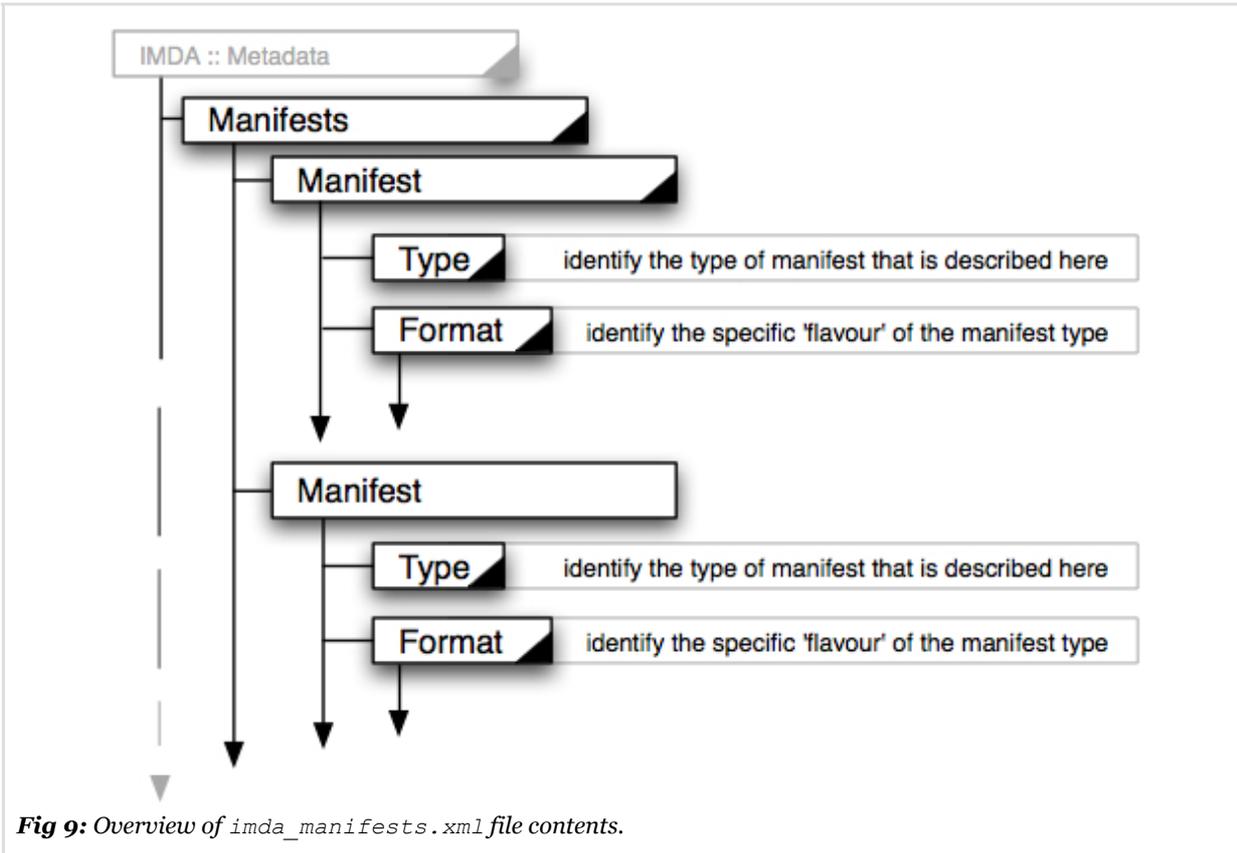
In order to provide information to a solution about schedule information, or more data on what is available for editorial programming in relation to a particular brand, we provide information in a file referenced from the `imda.xml` root file in `imda/manifests@url`. As we said, we don't mind what this file is called or where it is located – as long as it's publicly available. As an example – our manifests identification file could be here:

```
http://imda.organisation.com/imda_manifests.xml
```

The contents of this file allow an organisation to describe data that provides further information about what is available from brands – this could be schedule data regarding programming, or perhaps podcast feeds related to the brand.

If you do not have any manifests - then you don't need a manifests document, it's entirely optional.

Let's take a look at the main elements of this file:



Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

Our manifests document is made up of a `manifests` node, and one or more `manifest` children. Each `manifest` child has a `type` and a `format` to identify the general type of the manifest and the specific format of that type. The specific `type` and `format` combination will dictate the subsequent make up of the child nodes within your `manifest` node – these variants are described below after a quick example of how all this could appear in an XML file example:

```

1  <?xml version="1.0" encoding="UTF-8"?>
2  <manifests orgcrid="crid://bbc.co.uk/" modified="2011-01-20T134032Z" owner="imda@bbc.co.uk">
3    <manifest type="epg">
4      <format name="dabepg" >>/format>
5      <basetimezone tz="Europe/London" >>/basetimezone>
6      <si urlpattern="http://organisation.com/some/location/on/server/ [http://organisation.com/some/location/on/server/]>
7      <pi urlpattern="http://organisation.com/some/location/on/server/ [http://organisation.com/some/location/on/server/]>
8      <gi urlpattern="http://organisation.com/some/location/on/server/ [http://organisation.com/some/location/on/server/]>
9      <mapping type="serviceScope" value="e1.ce15.c221.0" servicefilename="e1_ce15_c221.0">
10     <mapping type="serviceScope" value="e1.ce15.c222.0" servicefilename="e1_ce15_c222.0">
11     <!-- ... -->
12     <!-- .. -->
13     <!-- . -->
14   </manifest>
15   <manifest type="epg">
16     <format name="dabepg_drm" >>/format>
17     <basetimezone tz="Europe/London" >>/basetimezone>
18     <si urlpattern="http://organisation.com/some/other/location/on/server/ [http://organisation.com/some/other/location/on/server/]>
19     <pi urlpattern="http://organisation.com/some/other/location/on/server/ [http://organisation.com/some/other/location/on/server/]>
20     <gi urlpattern="http://organisation.com/some/other/location/on/server/ [http://organisation.com/some/other/location/on/server/]>
21     <mapping type="serviceScope" value="c485" servicefilename="c485" brandid="r1" >>,
22     <mapping type="serviceScope" value="c486" servicefilename="c486" brandid="r2" >>,
23     <!-- ... -->
24     <!-- .. -->
25     <!-- . -->
26   </manifest>
27   <manifest type="podcast">

```

```

28     <format name="atom10" ></format>
29     <basetimezone tz="Europe/London" ></basetimezone>
30     <mapping brandid="r1" url="http://www.organisation.com/r1.xml [http://www.organisat
31     <mapping brandid="r2" url="http://www.organisation.com/r2.xml [http://www.organisat
32     <mapping brandid="r6" url="http://www.organisation.com/radio_opml_atom.xml [http:
33     <mapping brandid="r7" url="http://www.organisation.com/radio_opml_atom.xml [http:
34     </manifest>
35     <manifest type="podcast">
36         <format name="rss20" ></format>
37         <basetimezone tz="Europe/London" ></basetimezone>
38         <mapping brandid="r3" url="http://www.organisation.com/r3.xml [http://www.organisat
39         <mapping brandid="r4" url="http://www.organisation.com/r4.xml [http://www.organisat
40         <mapping brandid="r5" url="http://www.organisation.com/radio_opml_rss.xml [http:/
41         <mapping brandid="r8" url="http://www.organisation.com/radio_opml_rss.xml [http:/
42     </manifest>
43 </manifests>
    
```

Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

In the above example we've tried to keep things simple, to show some of the ways of describing the data. However – the details begin with our root node for this document: `manifests`.

Let us now look at the different types of manifests that you can have, and their structure in more detail.

## 6.1 | manifest@type

» attribute is required

There are currently two manifest types in this version of the specification, more types may become available in future specifications. The different types are defined in the table below:

<i>value</i>	<i>description</i>
epg	An Electronic Programme Guide (EPG) that describes scheduled programming associated with a brand, or brands.
podcast	A series of media files released episodically and associated to a brand, or brands.

**Table 8:** Allowed values of `manifest@type` attribute.

For each of these values, the content of the manifest child differs. The following describes these differences, allowing you to pick and choose which services you wish to include.

## 6.2 | manifest/format

» node is required, attributes required

Each `type` of manifest may have several distinct formats and this is signalled in the `format` node – we have defined several formats in this version of the specification, more formats may become available in future specifications. The node contains one attribute: `name`. In order to signal which format you wish this manifest node to contain - the

`format@name` should contain one of the following values described in the table below:

<i>value</i>	<i>description</i>
dabepg	The WorldDMB Organisation's "DAB EPG XML" - as defined in the ETSI standard TS 102 818 [ <a href="http://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWAhttp://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWA">http://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWA</a> ]
dabepg_drm	The WorldDMB Organsiation's "DAB EPG XML variant for DRM services"- as defined in the ETSI standard TS 102 818 [ <a href="http://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWA">http://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWA</a> ] (DRM is Digital Radio Mondial)
atom10	ATOM 1.0 syndication format - more information available here [ <a href="http://en.wikipedia.org/wiki/Atom_(standard)">http://en.wikipedia.org/wiki/Atom_(standard)</a> ]. As defined in RFC 4287 [ <a href="http://tools.ietf.org/html/rfc4287">http://tools.ietf.org/html/rfc4287</a> ].
rss20	RSS 2.0 syndication format - more information available here [ <a href="http://en.wikipedia.org/wiki/RSS">http://en.wikipedia.org/wiki/RSS</a> ]. This may contain media:rss module extensions [ <a href="http://video.search.yahoo.com/mrss">http://video.search.yahoo.com/mrss</a> ] from Yahoo!. (this is the most common for Podcast services)
xmltv	XMLTV is a specification used by many Open Source projects that use EPG data about Television services, though it can be used for Radio services. More information is available from the XMLTV Wiki [ <a href="http://wiki.xmltv.org/index.php/XMLTVFormat">http://wiki.xmltv.org/index.php/XMLTVFormat</a> ] and also MythTV's XMLTV resources page [ <a href="http://www.mythtv.org/wiki/XMLTV">http://www.mythtv.org/wiki/XMLTV</a> ].
webepg	Allows the Media Organisation with an unstructured HTML page, or pages, which contains schedule data about a brand. Please read the notes below on why, as a Media Organisation, you should avoid using this.

**Table 9:** Allowed values of `manifest/format@name` attribute.

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

For each of these values, the content of the manifest child differs. The following describes these differences, allowing you to pick and choose which services you wish to include.

### 6.3 | `manifest[@type='epg']/format[@name='dabepg']`

This manifest format allows you to link schedule information described in 'DAB EPG XML' documents provided by the media organisation to the brands as defined in the IMDA SI documents. This means that a solution can access data on the editorial programming associated with a live stream in `ip_direct` transport types, or the broadcast services in `broadcast` transport types.

For full details on how the DAB EPG XML should be coded, please refer to the ETSI standard TS 102 818 [<http://pda.etsi.org>]

/pda/home.asp?wki\_id=hUkTWWG26f8ABCDFDRWAh] provided by the WorldDMB organisation. For the IMDA SI manifests document - these XML files containing the data need to be available at URL accessed via HTTP (or HTTPS if the media organisation is using protection).

The following child nodes for this format allow the media organisation to signal where the DAB EPG XML files are stored, and how they link to the brands defined in the IMDA SI Brands document. Here is an example of a manifest node that links to a fictional DAB EPG XML location and maps it to some brands.

```

1  <manifest type="epg">
2  <format name="dabepg" ></format>
3  <basetimezone tz="Europe/London" ></basetimezone>
4  <si urlpattern="http://organisation.com/some/location/on/server/ [http://organisation.c
5  <pi urlpattern="http://organisation.com/some/location/on/server/ [http://organisation.c
6  <gi urlpattern="http://organisation.com/some/location/on/server/ [http://organisation.c
7  <mapping type="serviceScope" value="e1.ce15.c221.0" servicefilename="e1_ce15_c222_0"
8  <mapping type="serviceScope" value="e1.ce15.c222.0" servicefilename="e1_ce15_c222_0"
9  <!-- ... -->
10 <!-- .. -->
11 <!-- . -->
12 </manifest>

```

### 6.3.1 | basetimezone

» node and its attribute is required

In order for solutions working across various timezones, it is important for the media organisation to identify the home timezone for the schedules contained within the DAB EPG XML. This should match the basis used for the 'startTime' values in the DAB EPG XML.

The basetimezone has one required attribute called `tz` and the value of this value should follow the 'Olson' timezone specifications [<http://www.twinsun.com/tz/tz-link.htm>].

### 6.3.2 | si

» node is required

The DAB EPG XML specifies a 'Service Identification' document - this node allows you to link this in to the IMDA SI document. Since the DAB EPG XML files identify a shifting window of time then we allow here for 'macros' which you can use to get the correct DAB EPG XML file based on the current date.

#### 6.3.2.1 | si@urlpattern

» attribute is required

The `urlpattern` value should be an absolute URL to the location of the Media Organisations DAB EPG XML Service Identification (SI) file. Because of the shifting time window specified by the DAB spec, this absolute URL should contain 'macros' where dates are specified in the filename or path. Please see section below for information about what 'macros' are available. This URL may be HTTP or HTTPS depending on whether the media organisation is using protection.



*Reminder: URLs to your data in `si@urlpattern` can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

### 6.3.3 | pi

» node is required

The DAB EPG XML specifies 'Programme Information' documents - this node allows you to link this in to the IMDA SI document. Since the DAB EPG XML files identify a shifting window of time then we allow here for 'macros' which you can use to get the correct DAB EPG XML file based on the current date.

#### 6.3.3.1 | pi@urlpattern

» attribute is required

The `urlpattern` value should be an absolute URL to the location of the Media Organisations DAB EPG XML Programme Information (PI) files. Because of the shifting time window specified by the DAB spec, this absolute URL should contain 'macros' where dates are specified in the filename or path. Please see section below for information about what 'macros' are available. This URL may be HTTP or HTTPS depending on whether the media organisation is using protection.



*Reminder: URLs to your data in `pi@urlpattern` can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

### 6.3.4 | gi

» node is optional

The DAB EPG XML specifies a 'Grouping Information' document - this node allows you to link this in to the IMDA SI document. Since the DAB EPG XML files identify a shifting window of time then we allow here for 'macros' which you can use to get the correct DAB EPG XML file based on the current date.

#### 6.3.4.1 | gi@urlpattern

» attribute is required

The `urlpattern` value should be an absolute URL to the location of the Media Organisations DAB EPG XML Grouping Information (GI) file. Because of the shifting time window specified by the DAB spec, this absolute URL should contain 'macros' where dates are specified in the filename or path. Please see section below for information about what 'macros' are available. This URL may be HTTP or HTTPS depending on whether the media organisation is using protection.



*Reminder: URLs to your data in `gi@urlpattern` can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

## 6.3.5 | mapping

» node is required

Now that we know the `urlpatterns` for the SI, PI and GI DAB EPG XML files we specify the important keys that link the data inside these files with the brands in the IMDA SI document via the brand ids. The mapping node also allows the media organisation to provide the valid values of the '[SERVICEFILENAME]' macro, more about this in the section below.

All attributes of the mapping node are required.

### 6.3.5.1 | mapping@type

» attribute is required

Identifies the node in the DAB EPG XML that we will use to link the value of the `value` attribute to our brand id in the `refid` attribute. The default value of this attribute is 'serviceScope'.

### 6.3.5.2 | mapping@value

» attribute is required

This is the value that should be used to link the node in the DAB EPG XML (by default 'serviceScope') with the brand id in `refid` attribute.

### 6.3.5.3 | mapping@servicefilename

» attribute is required

The value of `servicefilename` defines what to put in the `urlpattern` of the `pi` node to find the data for this service so that it can link to the IDMA SI brand (in the `refid` attribute).

### 6.3.5.4 | mapping@brandid

» attribute is required

The brand id from the IMDA SI Brands document - this is the brand which the media organisation wishes to signal as being related to the Programme Information found by flattening the `urlpattern` of the `pi`, given the `servicefilename`.

## 6.3.6 | urlpattern macros

The `urlpattern` attributes contain URLs which have a number of path or filename elements which may vary depending on the current date, or the service in the DAB EPG XML files. The following table describes what each macro means, and how it should be used. Macros use the style `[VARIABLE]` because square brackets are not allowed in regular URLs, meaning that they are obviously not a valid URL until the macro is swapped out for a valid value. Please note that filename patterns are covered in Annex C of the ETSI DAB EPG XML documentation.

<i>macro</i>	<i>description</i>
[YYYY]	Four digit year. Example: [2011] [ <a href="http://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWAhttp://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWA">http://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWAhttp://pda.etsi.org/pda/home.asp?wki_id=hUkTWWG26f8ABCDFDRWA</a> ]
[MM]	Two digits for the month. Example: [06] which would mean June. Months can be from 1-12.
[DD]	Two digits for the day of the month. Example: [05] which would mean the 5th day of the month. The first day of the month is [01]
[SERVICEFILENAME]	This macro should be replaced with the appropriate mapping@servicefilename value so that the Programme Information file may be recovered from the media organisation and linked to the IMDA SI brand.

**Table 10:** Macros used in `urlpattern` attributes.

## 6.4 | `manifest[@type='epg']/format[@name='dabepg_drm']`

This manifest format is merely a variant of the manifest format `dabepg`. The child nodes are the same as for `dabepg` however the only difference that the `servicefilenames` make up is different in the DAB EPG XML.



*Why duplicate the `dabepg` format? Why have a variant for `drm` services? It's simply because a media organisation may have a set of DAB services **and** DRM services. Because the DRM services are identified slightly differently in the DAB EPG XML, we need to keep it separate in the IMDA SI Manifests document specification.*

## 6.5 | `manifest[@type='epg']/format[@name='xmltv']`

This manifest format is provided to allow a media organisation who publishes their schedule data using the XMLTV schema.

XMLTV is a specification used by many Open Source projects that use EPG data about Television services, though it can be used for Radio services. More information is available from the XMLTV Wiki [<http://wiki.xmltv.org/index.php/XMLTVFormat>] and also MythTV's XMLTV resources page [<http://www.mythtv.org/wiki/XMLTV>].

A quick example of how this node might appear:

```

1 | <manifest type="epg">
2 |   <format name="xmltv" ></format>
3 |   <basetimezone tz="Europe/London" ></basetimezone>
4 |   <mapping brandid="channel1" url="http://www.organisation.com/c1_xmltv.xml [http://www.c
5 |   <!-- ... -->
6 |   <!-- .. -->
```

```
7 | <!-- . -->
8 | </manifest>
```



*Reminder: URLs to your schedule data in the XMLTV format can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

### 6.5.1 | basetimezone

» node and its attribute is required

In order for solutions working across various timezones, it is important for the media organisation to identify the home timezone for the schedules contained within the this manifest.

The basetimezone has one required attribute called `tz` and the value of this value should follow the 'Olson' timezone specifications [<http://www.twinsun.com/tz/tz-link.htm>].

### 6.5.2 | mapping

» node is required

We specify here a URL for a web page that describes the schedule information for a brand – by defining a mapping between our brand's `brandid` and an XPath definition, as defined in the `xpath` attribute, which allows an aggregator to find the root node of the information relating to the particular brand.

All attributes of the mapping node are required.

#### 6.5.2.1 | mapping@brandid

» attribute is required

The `brandid` from the IMDA SI Brands document - this is the brand which the media organisation wishes to signal as being related to the schedule information held at the `url`.

#### 6.5.2.2 | mapping@url

» attribute is required

The `url` attribute contains the uniform resource locator which points to an XML document which follows the XMLTV format.



*Reminder: URLs to your schedule data in the XMLTV format can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

#### 6.5.2.3 | mapping@xpath

» attribute is required

The `xpath` attribute contains the an xpath expression that can be used against the XMLTV document as located in the `url` attribute to select the node of the tree that

contains schedule information that pertains to the brand specified in `brandid`.

## 6.6 | manifest[@type='epg']/format[@name='webepg']

This manifest format is provided to allow a media organisation who publishes their schedule data on publicly accessible pages built up of HyperText Markup Language (HTML). Some media organisations publish their schedule information on these 'web pages', the structure of which typically doesn't follow any other particular structured electronic programme guide standard.

A quick example of how this node might appear:

```

1 | <manifest type="epg">
2 |   <format name="website" ></format>
3 |   <basetimezone tz="Europe/London" ></basetimezone>
4 |   <specification url="http://url.to.format.description.if.available.com [http://url.to.fo
5 |   <mapping brandid="wdr5" url="http://www.wdr.de/programmvorschau/programDateDateSender
6 |   <!-- ... -->
7 |   <!-- .. -->
8 |   <!-- . -->
9 | </manifest>
    
```



*The `webepg` manifest format is included to allow Media Organisations to link to existing web-accessible pages, perhaps on their website, that contains schedule data about a brand's programming. This is classed as unstructured in that it doesn't necessarily conform to any specification that has shared understanding, it is only structured in the sense that it should be written in HTML markup. If a Media Organisation chooses to use this format they should be aware that an aggregator may take longer to list schedule data, or not list the schedule data at all. It is **strongly** recommended that you provide schedules using the other more structured XML manifest formats available here.*

### 6.6.1 | basetimezone

» node and its attribute is required

In order for solutions working across various timezones, it is important for the media organisation to identify the home timezone for the schedules contained within the this manifest.

The `basetimezone` has one required attribute called `tz` and the value of this value should follow the 'Olson' timezone specifications [<http://www.twinsun.com/tz/tz-link.htm>].

### 6.6.2 | specification

» node is optional, but if node is specified the attribute is required

If you have a document that defines the specification of your schedule web pages in HTML, then please provide a link using the

specification node. Put the URL to the specification documentation in the `specification@url` attribute. This node is optional, but please consider that without guidance from a specification an aggregator will be even less likely to list your schedule data.

### 6.6.3 | mapping

» node is required

We specify here a URL for a web page that describes the schedule information for a brand – by defining a mapping between our brand's `brandid` and a webpage `url`.

All attributes of the mapping node are required.

#### 6.6.3.1 | mapping@brandid

» attribute is required

The `brandid` from the IMDA SI Brands document - this is the brand which the media organisation wishes to signal as being related to the schedule information held at the `url`.

#### 6.6.3.2 | mapping@url

» attribute is required

The `url` attribute contains the uniform resource locator which points to a web page built using Hypertext Markup Language (HTML). The web page contains schedule information.

If you have provided a `specification` in this manifest node, this HTML web page should follow what is documented there.



*Reminder: URLs to your schedule data in web pages can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

## 6.7 | manifest[@type='podcast']/format[@name='rss20']

This manifest format is provided to allow a media organisation who publish episodic media via the podcast mechanism using RSS 2.0 [<http://en.wikipedia.org/wiki/Rss>] formatted XML files.

"A podcast is a series of digital media files (either audio or video) that are released episodically and often downloaded through web syndication. The word replaced webcast in common use with the success of the iPod and its role in the rising popularity and innovation of web feeds." ~ Wikipedia entry for 'Podcast' [<http://en.wikipedia.org/wiki/Podcast>]

It is valid to link to RSS 2.0 formatted files that use the Media RSS module

[[http://en.wikipedia.org/wiki/Media\\_RSS](http://en.wikipedia.org/wiki/Media_RSS)] specified by Yahoo!.

Media organisations that use OPML 1.0 (Outline Processor Markup Language [<http://en.wikipedia.org/wiki/OPML>] version 1.0) to aggregate a group of RSS 2.0 syndicated programmes together, can specify the location of the aggregate document in this section.

The RSS 2.0 specification is available at:

```
http://www.rssboard.org/rss-specification [http://www.rssboard.org/rss-specification]
```

The MEDIA:RSS module specification version 1.5.0 is available at:

```
http://video.search.yahoo.com/mrss [http://video.search.yahoo.com/mrss]
```

The OPML 1.0 specification is available at:

```
http://dev.opml.org/spec1.html [http://dev.opml.org/spec1.html]
```

In order to demonstrate the flexibilities of this node – there are at least four different ways it could be used:

1. A single 'podcast' can be related to a particular Brand by providing a mapping between where the RSS is available for the 'podcast' and the brand id that it should relate to.
2. Multiple 'podcasts' can be related to a particular Brand by having more than one of these mappings with the same brand id.
3. Multiple 'podcasts' can be related to a particular Brand by providing a mapping between where the OPML is available for the aggregated list of 'podcasts' and the brand id.
4. A combination of mappings – links to RSS, or OPML files, is possible for complex scenarios.

A quick example of how this node might appear:

```

1 | <manifest type="podcast">
2 |   <format name="rss20" ></format>
3 |   <basetimezone tz="Europe/London" ></basetimezone>
4 |   <mapping brandid="r3" url="http://www.organisation.com/r3.xml [http://www.organisation.c
5 |   <mapping brandid="r4" url="http://www.organisation.com/r4.xml [http://www.organisation.c
6 |   <mapping brandid="r5" url="http://www.organisation.com/radio_opml_rss.xml [http://www.c
7 |   <mapping brandid="r8" url="http://www.organisation.com/radio_opml_rss.xml [http://www.c
8 | </manifest>

```



*Reminder: URLs to your 'podcasts' can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

## 6.7.1 | basetimezone

In order for solutions working across various timezones, it is important for the media organisation to identify the home timezone for the podcasts contained within the this manifest. Although the RSS file linked to will present a timezone in the `rss/channel/item/pubDate` related to each episode, it is useful to know beforehand.

The `basetimezone` has one required attribute called `tz` and the value of this value should follow the 'Olson' timezone specifications [<http://www.twinsun.com/tz/tz-link.htm>].

## 6.7.2 | mapping

» node is required

We specify here a URL to the RSS 2.0 formatted XML file that describes the podcasts for a particular brand – by defining a mapping between our brand's `brandid` the podcasts represented in the linked RSS.

A variant of this node allows for OPML formatted XML files to be linked in the URL attribute. If you wish to use an OPML file here – the mapping node should include the `xpath` attribute which contains the XPath definition which allows an aggregator to find the root `OPML:outline` node of the information relating to the particular brand.

All attributes of the mapping node are required.

### 6.7.2.1 | mapping@brandid

» attribute is required

The `brandid` from the IMDA SI Brands document - this is the brand which the media organisation wishes to signal as being related to the podcast information held at the `url`.

### 6.7.2.2 | mapping@url

» attribute is required

The `url` attribute contains the uniform resource locator which points to an XML document which follows the RSS 2.0, unless the `xpath` attribute is used in which case the XML document should follow the OPML format.



*Reminder: URLs to your podcasts can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

### 6.7.2.3 | mapping@xpath

» attribute is optional – if specified then the contents of `@url` should be formatted as OPML

The `xpath` attribute contains a valid XPath expression that can be used against the OPML document to select the appropriate `outline` node that links to the RSS 2.0 resource related to the brand specified in `brandid`.

## 6.8 | manifest[@type='podcast']/format[@name='atom10']

This manifest format is provided to allow a media organisation who publish episodic media via the podcast mechanism using ATOM Syndication Format 1.0 (known as 'Atom 1.0') formatted XML files.

"A podcast is a series of digital media files (either audio or video) that are released episodically and often downloaded through web syndication. The word replaced webcast in common use with the success of the iPod and its role in the rising popularity and innovation of web feeds." ~ Wikipedia entry for 'Podcast' [<http://en.wikipedia.org/wiki/Podcast>]

Media organisations that use OPML 1.0 (Outline Processor Markup Language [<http://en.wikipedia.org/wiki/OPML>] version 1.0) to aggregate a group of ATOM 1.0 syndicated programmes together, can specify the location of the aggregate document in this section.

The ATOM Syndication Format 1.0 specification is available as RFC 4287:

```
http://tools.ietf.org/html/rfc4287 [http://tools.ietf.org/html/rfc4287]
```

The OPML 1.0 specification is available at:

```
http://dev.opml.org/spec1.html [http://dev.opml.org/spec1.html]
```

In order to demonstrate the flexibilities of this node – there are at least four different ways it could be used:

1. A single 'podcast' can be related to a particular Brand by providing a mapping between where the ATOM feed is available for the 'podcast' and the brand id that it should relate to.
2. Multiple 'podcasts' can be related to a particular Brand by having more than one of these mappings with the same brand id.
3. Multiple 'podcasts' can be related to a particular Brand by providing a mapping between where the OPML is available for the aggregated list of 'podcasts' and the brand id.
4. A combination of mappings – links to ATOM feeds, or OPML files, is possible for complex scenarios.

A quick example of how this node might appear:

```
1 <manifest type="podcast">
2   <format name="atom10" ></format>
3   <basetimezone tz="Europe/London" ></basetimezone>
4   <mapping brandid="r1" url="http://www.organisation.com/r1.xml [http://www.organisation.c
5   <mapping brandid="r2" url="http://www.organisation.com/r2.xml [http://www.organisation.c
```

```

6 |     <mapping brandid="r6" url="http://www.organisation.com/radio_opml_atom.xml [http://www.
7 |     <mapping brandid="r7" url="http://www.organisation.com/radio_opml_atom.xml [http://www.
8 | </manifest>
    
```



*Reminder: URLs to your 'podcasts' can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

### 6.8.1 | basetimezone

» node and its attribute is required

In order for solutions working across various timezones, it is important for the media organisation to identify the home timezone for the podcasts contained within the this manifest. Although the ATOM feed linked to will present a timezone in the feed/entry/updated related to each episode, it is useful to know beforehand.

The basetimezone has one required attribute called `tz` and the value of this value should follow the 'Olson' timezone specifications [<http://www.twinsun.com/tz/tz-link.htm>].

### 6.8.2 | mapping

» node is required

We specify here a URL to the ATOM Syndication Format 1.0 formatted XML file that describes the podcasts for a particular brand – by defining a mapping between our brand's `brandid` the podcasts represented in the linked ATOM feed.

A variant of this node allows for OPML 1.0 formatted XML files to be linked in the URL attribute. If you wish to use an OPML file here – the mapping node should include the `xpath` attribute which contains the XPath definition which allows an aggregator to find the root `OPML:outline` node of the information relating to the particular brand.

All attributes of the mapping node are required.

#### 6.8.2.1 | mapping@brandid

» attribute is required

The `brandid` from the IMDA SI Brands document - this is the brand which the media organisation wishes to signal as being related to the podcast information held at the `url`.

#### 6.8.2.2 | mapping@url

» attribute is required

The `url` attribute contains the uniform resource locator which points to an XML document which follows the ATOM Syndication Format 1.0, unless the `xpath` attribute is used in which case the XML document should follow the OPML 1.0 format.



*Reminder: URLs to your podcasts can be available via HTTP or HTTPS - and basic authentication may be used for protection. See "Protecting your metadata" for further information.*

### 6.8.2.3 | mapping@xpath

The `xpath` attribute contains a valid `xpath` expression that can be used against the OPML document to select the appropriate `outline` node that links to the ATOM 1.0 resource related to the brand specified in `brandid`.

» attribute is optional – if specified then the contents of `@url` should be formatted as OPML

---

## 6.9 | Your favourite manifest type or EPG format here!

If you have information on an additional type or format of manifest, including a common EPG format, which you feel is important we support – please get in touch and provide feedback.

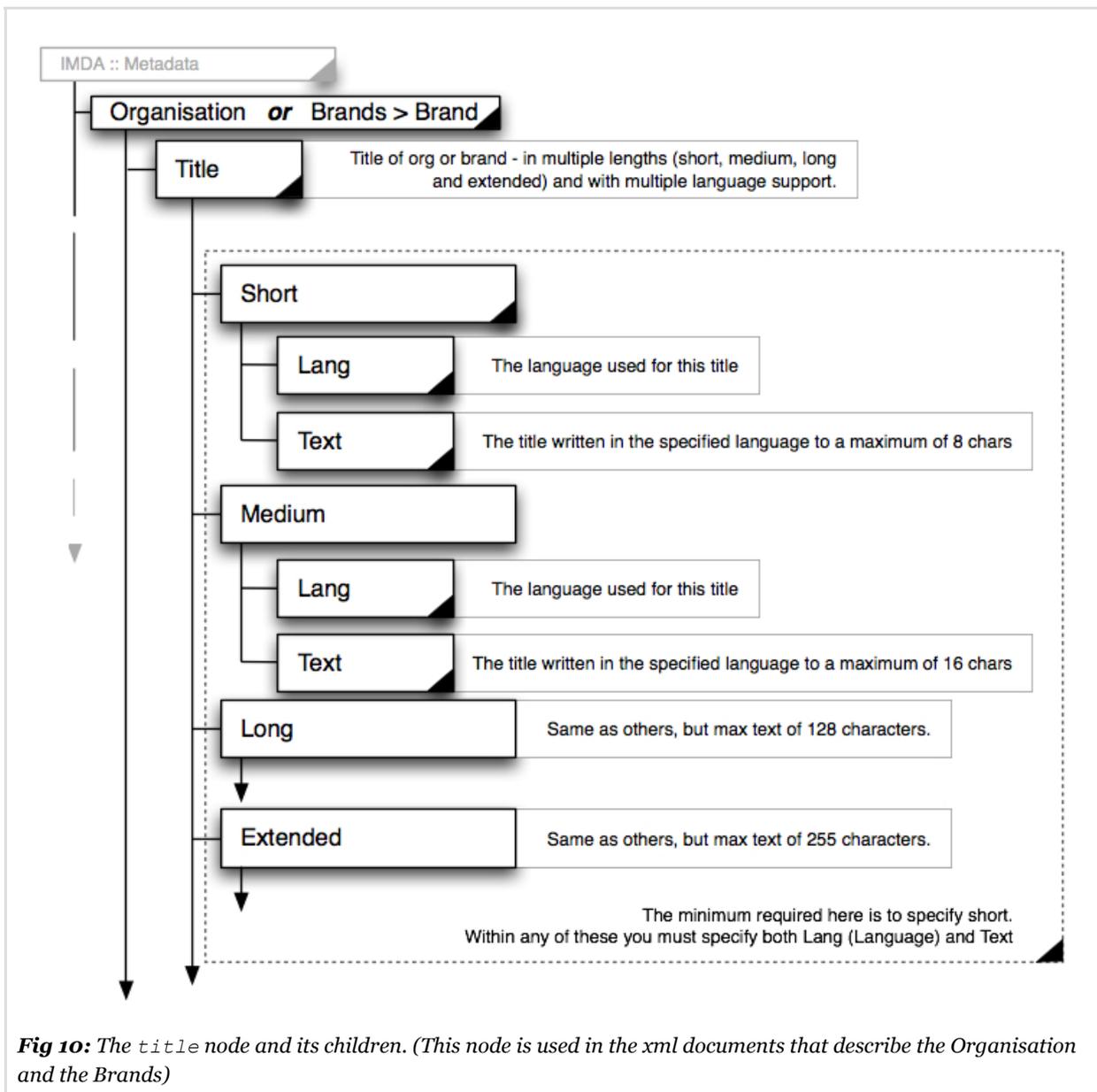
## 7 | Shared Nodes

---

### 7.1 | [organisation|brands/brand]/title

» node is required, see advice about children

The `title` node allows us to present the organisation's title or the organisations brand's name in a set of common sizes that are currently used by devices. It also provides the ability to have alternative language support.



The children of `title` are `short`, `medium`, `long` and `extended` and allow the organisation to provide how they wish their titles to appear where text may be limited to certain sizes. These were decided based on existing specifications that had considerations to how solutions may present an organisation's name. One of the specifications that was used as a good example was the ETSI [http://bit.ly/etsiweb] (European Telecommunications Standards Institute) TS 102.818 [http://bit.ly/9s9mI7], a.k.a. the 'DAB EPG SI' specification.

A summary of the allowable lengths for the various text nodes of the children are:

<i>Title</i>	<i>Number of Characters (inclusive numbers)</i>
Short	1 – 8
Medium	1 – 16

<b>Title</b>	<b>Number of Characters (inclusive numbers)</b>
Long	1 – 128
Extended	1 – 255

**Table 11:** *title children text node lengths.*

The `lang` attribute of those children should contain an entry from RFC 5646 [<http://bit.ly/rfc5646>] that identifies the language of the enclosed text node. A key part of this RFC is section 4 [<http://bit.ly/rfc5646s4>], which includes information about how to choose and how to interpret language tags.

The `title` node can contain one or more children of the same type, if the `lang` is different. This allows you to, for example, have multiple `short` children but in different languages.



*Best practice recommendation: as an organisation you should have at least an English language option for any short, medium, long or extended nodes. This is important as in many cases the solution may have a default display that uses English, or it may be that a hardware limitation means that a display can not support all character sets (quite often this is because a hardware decision was made since the number of services with a minority language doesn't make it sensible to pay for a display which can actually present that language where the hardware is limited). Even if your default language would be non-English you should consider having an English version of your title.*



*Best practice recommendation: don't use a title entry if it contains less characters than the shorter type before it. But you must specify the 'short' title as a minimum. For example – if your title is 15 characters and you have no longer variants then you will provide a 'medium' title here, and you must create a shorter version for 'short' title to comply with the minimum requirements.*

From the example above this could appear in your xml like this – a simple example with just a short and long title all in English (Great Britain):

```

1 <title>
2   <short lang="en-GB">BBC</short>
3   <long lang="en-GB">British Broadcasting Corporation</long>
4 </title>
```

Another example – demonstrating French and English versions of short, medium and long titles:

```

1 <title>
2   <short lang="en">DogRadio</short>
3   <short lang="fr">RDOChien</short>
4   <medium lang="en">Dog Radio</medium>
5   <medium lang="fr">Radio Chien</medium>
6   <long lang="fr">Radio Chien [Nantes]</long>
```

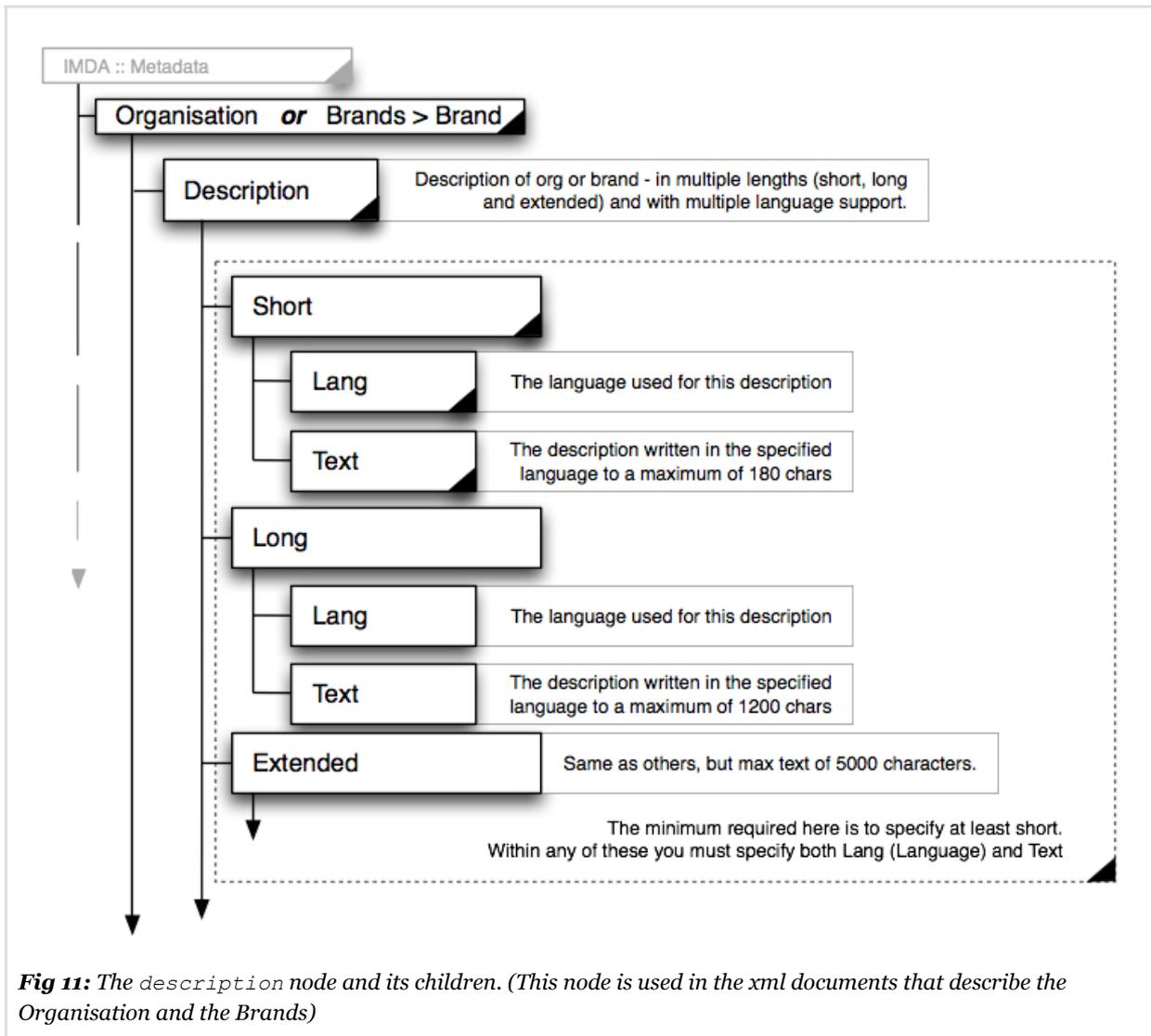
```
7 | <long lang="en">Dog Radio [Nantes]</long>  
8 | </title>
```

---

## 7.2 | [organisation|brands/brand]/description

» See parent document advice for node requirement, child requirements advice can be found below

This node allows us to provide a description of the organisation or the brand that would be visible to an audience member, and it allows us to do this in a set of common sizes that are currently used by devices. It also provides the ability to have alternative language support.



**Fig 11:** The description node and its children. (This node is used in the xml documents that describe the Organisation and the Brands)

The children of description are short, long and extended and allow the organisation to provide a description of the organisation, or brand, where text may be limited to certain sizes. These were decided based on existing specifications that had considerations to how solutions may present an organisation's description. One of the specifications that was used as a good example was the ETSI [http://bit.ly/etsiweb] (European Telecommunications Standards Institute) TS 102.818 [http://bit.ly/9s9mI7], a.k.a. the 'DAB EPG SI' specification.

A summary of the allowable lengths of the for the various text nodes of the children are:

<i>Description</i>	<i>Number of Characters (inclusive numbers)</i>
Short	1 – 180
Long	1 – 1200
Extended	1 – 5000

**Table 12:** description children text node lengths.

The `lang` attribute of those children should contain an entry from RFC 5646 [http://bit.ly/rfc5646] that identifies the language of the enclosed text node. A key part of this RFC is section 4 [http://bit.ly/rfc5646s4], which includes information about how to choose and how to interpret language tags.

The `description` node can contain one or more children of the same type, if the `lang` is different. This allows you to, for example, have multiple `short` children but in different languages.



*Best practice recommendation: as an organisation you should have at least an English language option for any short, long or extended nodes. This is important as in many cases the solution may have a default display that uses English, or it may be that a hardware limitation means that a display can not support all character sets (quite often this is because a hardware decision was made since the number of services with a minority language doesn't make it sensible to pay for a display which can actually present that language where the hardware is limited). The majority of solutions may also be searching content using English words – and providing English versions of the descriptions may help your 'findability' in search engines. In summary - even if your default language would be non-English you should consider having an English version of your description.*



*Best practice recommendation: the minimum requirement is to provide at least a short description. However you should understand that also providing long (and extended) descriptions provides more information for a solution, which may use this information to allow 'findability' of your services by the audience.*

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

From the example above this could appear in your xml like this – a simple example with just a short and long descriptions all in English (Great Britain):

```

1 | <description>
2 |   <short lang="en-GB">To enrich people's lives with programmes and services that inform.
3 |   <long lang="en-GB">In order for the BBC to fulfil its mission to inform, educate and
4 | </description>
    
```

Another example – demonstrating French and English versions of short and long descriptions:

```

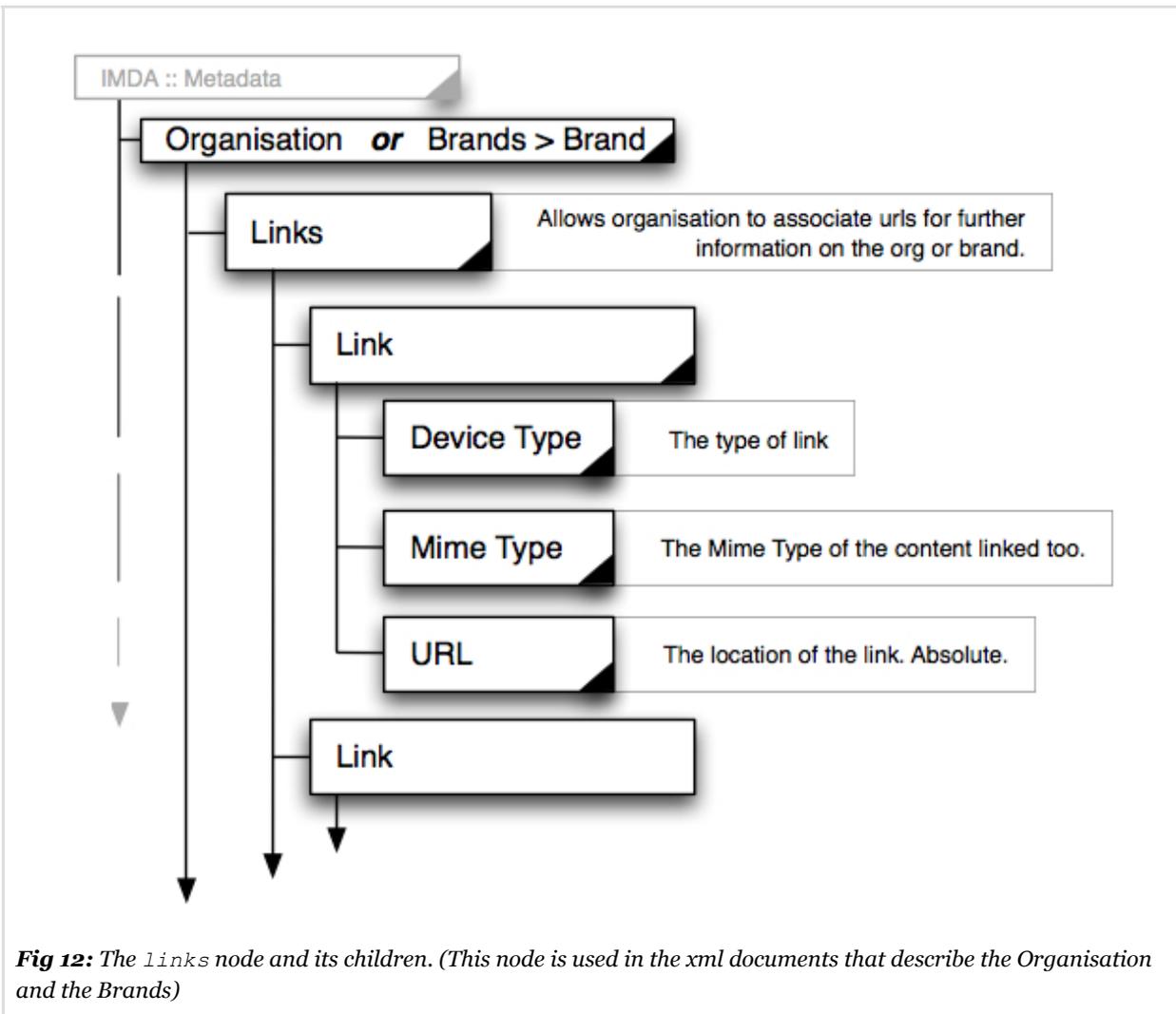
1 | <description>
2 |   <short lang="en">Best radio for our canine friends.</short>
3 |   <short lang="fr">Meilleur radio pour nos amis canins.</short>
4 |   <long lang="en">Music, discussion, recipes, politics and consumer tips for our four l
5 |   <long lang="fr">Musique, discussion, des recettes, la politique et les conseils aux c
6 | </description>
    
```

## 7.3 | [organisation|brands/brand]/links

This node allows an organisation to provide links

See parent document advice for node requirement, child requirements advice can be found below

to their own internet accessible services for the root organisation or their organisation's brands – for example a link to their home page on the web, or perhaps their mobile website.



**Fig 12:** The `links` node and its children. (This node is used in the xml documents that describe the Organisation and the Brands)

`links` is not a required node, but if you have a `links` node you must have at least one `link` child.

As a child node – `link` requires that you provide all attributes – `devicetype`, `mimetype` and `url`.

An example of a `links` node is:

```

1 | <links>
2 |   <link devicetype="screen" mimetype="text/html" url="http://www.bbc.co.uk/ [http://www.t
3 |   <link devicetype="handheld" mimetype="text/html" url="http://www.bbc.co.uk/mobile/ [ht
4 | </links>
    
```

The `devicetype` and `mimetype` attributes are a way of signaling the appropriateness of the linked to content in the `url` attribute.

### 7.3.1 | .../links/link@devicetype

» attribute is required

The value of `devicetype` should be specified using any of the media-types listed in

W3C's Cascading Style Sheet specification. [<http://bit.ly/css2media>]

Media-types are worth thinking about in terms of how appropriate it would be to present this to your user/audience – so we are talking about User Interface Engineering. If you think about this in terms of the rough distance from the user's eyes to the interface you establish the following rule of thumb: television is a '10 foot' experience because 10 feet (about 3meters) is an average of how far a person sits distant to their television; a mobile phone would be a '1 foot' experience (basically - personal - up close); someone using a computer is about a '3 foot' experience (laptop or desktop... it's probably about 3 feet away from the user, possibly 2 feet... but this is just a guide); and a cinema or projector would be more than 10 feet away so we'll call that '10+ foot' experience.

Therefore valid values of `devicetype` are:

<i>devicetype value</i>	<i>usage guidance</i>
screen	use this when showing a link to a web browser accessible website (Consider this to be a '3 foot' experience')
handheld	currently the W3C suggests identifying all mobile handsets and PDAs as 'handheld' because it merely gives overall idea that the content is appropriate for a situation when someone is using a 'handheld' device. (Consider this to be a '1 foot' experience)
tv	when it's appropriate to be on a TV screen (Consider this to be a '10 foot' experience)
projection	again - the idea from the W3C is to identify when something is being displayed in a large format. (Consider this to be a '10+ foot' experience.)
print	this could be used to give a link to a PDF (appropriate mime-type required, obviously) or other things in the future.
braille	for braille readers (tactile feedback device)
embossed	for braille readers (paged)
speech	for an aural experience
tty	?

**Table 13:** Values of the `devicetype` attribute. (This table is ranked by the more common 'devicetype' values first – we currently don't have use cases for the lower items – please provide feedback should you have any pertinent information)

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

### 7.3.2 | `.../links/link@mimetype`

» attribute is required

Here you specify the content type that will be returned should you visit the `url` specified. e.g. `text/html`, `text/vnd.vap.wml`

IANA maintains a list of commonly used mime-types [<http://bit.ly/mimetypes>] - though it

is not exhaustive and we do not place a restriction on the types that can be referenced.

## 7.4 | [organisation|brands/brand]/tags

See parent document advice for node requirement, child requirements advice can be found below

The ability to provide free-text 'tags' allows the organisation to provide some useful keywords that describe itself or the brands that it provides. `Tags` is an optional node.

### 7.4.1 | .../tags/tag

» node is required, 1 or more

The children of the `tags` node are `tag` nodes and you can provide as many as you like. The text node of each `tag` node should contain a string that constitutes a word or short phrase. The concept of 'tagging' something is to allow search functions to allow some categorisation and grouping based on free-text.



*Tagging should be considered in conjunction with the `genres` node – which is a more restricted way of categorising by using common shared vocabularies. Consider correctly categorising by using the `genres` node as much as you can, and where you have things that don't fit – use `tags`.*



*Tags can also be used to provide alternative spellings or common miss-spellings of an organisations name.*

### 7.4.2 | .../tags/tag@weight

» attribute is optional

In order to suggest hierarchy or importance of tags, you can use the `weight` attribute of the `tag` node. The value of this attribute should be an integer signifying the relative 'weight' of this tag in relation to other tags.

The lowest `weight` is zero. Any node that has a higher weight than zero should be considered as having higher importance.

If no `weight` is specified on any `tag` then everything is of equal weight. If only some `tag` nodes have a `weight` specified, then any `tag` nodes without a `weight` should be considered having a weight of zero. Tag nodes with the same `weight` should be considered as having equal weight.



*How a solution interprets the weight given by an organisation to any tag, and the relationship between that tag and others, is up to the solution. It is not something that the IMDA wishes to specify – as all solutions have their own methods for*



determining how to search or browse information. The concept of *weight* is merely to provide assistance to highlight how important the organisation believes a *tag* is to them.

## 7.5 | [organisation|brands/brand]/genres

Categorise your organisation or brands by providing `genres` that describe them. Genres is an optional node.

» see parent document advice for node requirement, child requirements advice can be found below



*Tip: Genres supplied by the you (the organisation) are used by the solution to determine where your brands appear in the solution. However – solutions may not use these verbatim to list your brands, as the genres may not match with their solution – so consider anything you provide here as advice about your brands.*

Genres can contain entries which are either:

*unconstrained ~*

if you have your own concept of how your content should be categorised into genres

*constrained ~*

by constraining your genre types to previously published **shared vocabularies** (e.g. TV-Anytime, iTunes, Media-RSS, EBU...) you can assist solutions to being able to categorise your content alongside content from other organisations.

What is a 'shared vocabulary'? Often referred to as a 'controlled vocabulary'. A vocabulary is a set of terms you are familiar with - a controlled vocabulary is one agreed by a group that allows them to talk about things and know that what one person means when they use a term is the same as what another person means with the same term.

"Controlled vocabulary schemes mandate the use of predefined, authorised terms that have been pre-selected by the designer of the vocabulary, in contrast to natural language vocabularies, where there is no restriction on the vocabulary." - Wikipedia [<http://bit.ly/wpcontvoc>]

We call a controlled vocabulary a 'shared' vocabulary merely because agreed terminology is 'shared' within a group. The two terms are interchangeable.

The `genres` node contains definitions about which shared vocabulary you wish to use (and you can use multiple if you wish), alongside the actual genre identification.

You can mix constrained and unconstrained genres, or have only constrained or

unconstrained genres – which, in summary means, that you can have either:

1. constrained only
2. unconstrained only
3. a mixture of constrained and unconstrained



*Best practice guidance: it is recommended strongly by the IMDA that you **do not provide unconstrained only** as this limits the solution from being able to provide your content listed against other content from other organisations. It also makes it difficult to search for – especially if your genres are very different from the shared vocabularies (see next section about domains and further notes for broadcasters). We recommend strongly that you provide constrained only or both.*



*Language: most common shared vocabularies use English language terms. This is because of the majority of solutions current in service. Should you have non-English genres - please consider mapping these to a shared vocabulary (see domains below) and where not possible, providing them unconstrained. A future version of this specification may add language support to this node. (Please provide feedback if you desire this feature)*

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

## 7.5.1 | .../genres/genre

» node is required, 1 or more

The children of `genres` are `genre` nodes. You can specify 1 or more `genre` nodes. If you haven't provided the `genres` node then you don't have any `genre` nodes.

Here's a quick example – this shows 6 constrained `genre` nodes and 1 unconstrained `genre` node:

```

1  <genres>
2    <genre vocabulary="tva151">3.1.1.8.1</genre>
3    <genre vocabulary="tva151">3.1.1.10.2</genre>
4    <genre vocabulary="tva151">3.1.1.10.3</genre>
5    <genre>jazz fusion hip-hop</genre>
6    <genre vocabulary="tva151">3.1.1.10.4</genre>
7    <genre vocabulary="tva151">3.1.1.10.5</genre>
8    <genre vocabulary="tva151">3.1.1.10.6</genre>
9  </genres>

```

`genre` nodes come in two flavours – so that you can provide constrained or unconstrained genres:

### 7.5.1.1 | Unconstrained `genre` node

The text node of `genre` node is not constrained – so you can put whatever your genre reference is. This could be a text string representing the genre itself, or a numeric reference to some vocabulary you use.

Here are some examples of unconstrained `genre` nodes:

```

1 <genres>
2   <genre>lo-fi swing</genre>
3   <genre>hip 2-step</genre>
4   <genre>tribal death trance</genre>
5   <genre>grime tango</genre>
6 </genres>
    
```



*Please consider providing feedback to us about your vocabulary, and why other shared vocabularies cannot provide what you need – this may allow us both to consider making your vocabulary a shared one, and adding it to this spec.*

### 7.5.1.2 | Constrained `genre` node

The text node of a `genre` node should contain a reference to the genre as specified in a shared vocabulary and is therefore 'constrained'. We define which shared vocabulary we are using by specifying a `genre@vocabulary` attribute. Here are a few example constrained `genre` nodes:

```

1 <genres>
2   <genre vocabulary="tva1.5.1">3.1.1.8.1</genre>
3   <genre vocabulary="itunes052010music">5.1</genre>
4   <genre vocabulary="itunes052010podcast">13.3</genre>
5   <genre vocabulary="id3v2.3.0">(4)Eurodisco</genre>
6   <genre vocabulary="ebu10.12.209">3.2.7.6.2</genre>
7 </genres>
    
```

In this example we have used a few of our `vocabulary` attribute values, the full list is in the table below, and are a short string that identifies a particular shared vocabulary (and the version referred to).

We recommend that you use, at a minimum, the TV-Anytime genre list.

<i>vocabulary value</i>	<i>Description</i>
tva1.5.1	<p>TV-Anytime – is an "association of organizations which seeks to develop specifications to enable audio-visual and other services based on mass-market high volume digital storage in consumer platforms." ~ TV-Anytime website [<a href="http://bit.ly/tvanytime">http://bit.ly/tvanytime</a>]</p> <p>Specifically we are referring to version 1.5.1 of their specification.</p> <p>» Published Document [<a href="http://bit.ly/aty1GW">http://bit.ly/aty1GW</a>]</p> <p>As specified in the published ETSI standard: "ETSI TS 102 822-3-1 V1.5.1 (2009-05)" TV-Anytime refers to 'genres' as 'formats' - look for ContentCS. Full xml document with the options is available in Appendix A.8 (on page 119).</p>

<i>vocabulary value</i>	<i>Description</i>
	<p>The IMDA has republished the xml document here for simplicity:  <a href="#">tva_contentCS_v151.xml</a>                      Use the Term@TermID reference in the text node of our genre node.</p>
ebu10.12.209	<p>European Broadcasting Union (EBU) – " is the largest association of national broadcasters in the world. We promote cooperation between broadcasters and facilitate the exchange of audiovisual content. The EBU works to ensure that the crucial role of public service broadcasters is recognised and taken into consideration by decision-makers." ~ EBU website  <a href="http://bit.ly/ebuweb">[http://bit.ly/ebuweb]</a></p> <p>Specifically we are referring to version 10.12.209 of their specification.</p> <p>» Published Document <a href="http://bit.ly/ebucgenre">[http://bit.ly/ebucgenre]</a>                      EBU have worked on a shared vocabulary which tries to aggregate a number of other popular shared vocabularies (including TV-Anytime, DAB and iTunes). This is a helpful mapping document that shows how the EBU's content format (genre) maps to the others.                      Use the EBU's 'TermID' in the text node <code>genre</code>.</p>
itunes052010music	<p>Apple iTunes - categorisation for Music genres</p> <p>Since Apple doesn't provide a specification for this, we have provided a snapshot of the categories as of May 2010.</p> <p>The IMDA has published an XML document with all of the categorisation here: <a href="#">itunes_music.xml</a></p> <p>Use the 'termid' attribute of the category as the value of <code>genre text node</code>.</p>
itunes052010podcast	<p>Apple iTunes - categorisation for Podcast services.</p> <p>Since Apple doesn't provide a specification for this, we have provided a snapshot of the categories as of May 2010.</p> <p>The IMDA has published an XML document with all of the categorisation here: <a href="#">itunes_podcasts.xml</a></p> <p>Use the 'termid' attribute of the category as the value of <code>genre text node</code>.</p>

<i><b>vocabulary value</b></i>	<i><b>Description</b></i>
id3v2.3.0	<p>ID3 Tags, specifically version 2.3.0 – "ID3 is a very popular audio file data tagging format in active use by software and hardware developers around the world." ... "An ID3 tag is a data container within an MP3 audio file stored in a prescribed format." ID3 Website</p> <p>[<a href="http://bit.ly/id3web">http://bit.ly/id3web</a>]                      » List of Genres [<a href="http://bit.ly/id3genres">http://bit.ly/id3genres</a>]                      Use the genre code from ID3 tags by following the information about the 'Content Type' (TCON) as described in the usage information [<a href="http://bit.ly/id3usage">http://bit.ly/id3usage</a>] given here. For example - the text node of your genre node might be '(21)' or '(51)(39)' or with a refinement comment '(4)Eurodisco'</p>

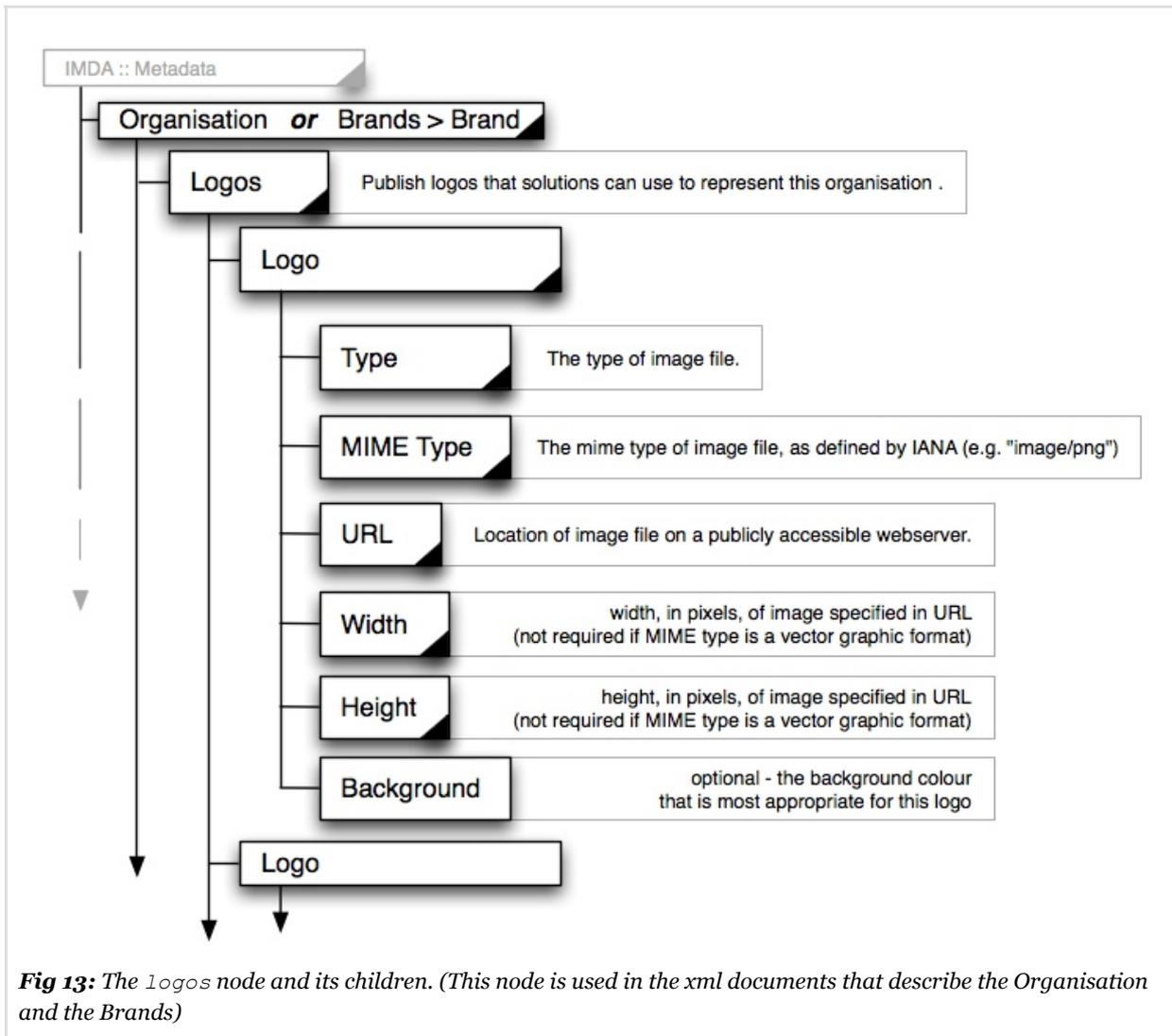
**Table 14:** Value of *vocabulary* attribute and the commonly used shared vocabularies for genre categorisation.

 *Don't see your favourite genre shared vocabulary above? Please contact us and ask us to consider it for inclusion – you'll need to explain why other shared vocabularies cannot provide what you need – this may allow us both to consider making your vocabulary a shared one, and adding it to this spec.*

## 7.6 | [organisation|brands/brand]/logos

see parent document advice for node requirement, child requirements advice can be found below

This node allows an organisation to provide various logo assets – graphic imagery – related to the organisation or the organisation's brands. This is so that a solution can provide the correct branding. Since there are a potential infinite number of possible solutions that could use different sizes and formats, we specify here a range of assets that you should provide based on common solutions currently in the industry.



Logos is a required node. You can specify multiple different sizes or formats of logos by using the child logo node.

As a child node – logo requires that you provide at least attributes – type and url. The width and height are needed when the image type is a bitmap image. The background attribute is optional – and is used to signal the background colour that is appropriate for the logo provided – this allows you to produce multiple logo nodes for the same type, but that would be appropriate for certain background colours.

An example of a logos node is:

```

1 <logos>
2   <logo type="imda_micro_1:1" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_micro_1:1.png" />
3   <logo type="imda_mobile_1:1" background="black" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_mobile_1:1.png" />
4   <logo type="imda_mobile_1:1" background="white" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_mobile_1:1.png" />
5   <logo type="imda_mobile_1:1" background="silver" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_mobile_1:1.png" />
6   <logo type="imda_mobile_3:2" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_mobile_3:2.png" />
7   <logo type="imda_mobile_4:3" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_mobile_4:3.png" />
8   <logo type="imda_tv_4:3" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_tv_4:3.png" />
9   <logo type="imda_tv_16:9" mimetype="image/png" url="http://imda.bbc.co.uk/imda/logos/imda_tv_16:9.png" />
10  <logo type="imda_svg_1:1" mimetype="image/svg+xml" url="http://imda.bbc.co.uk/imda/logos/imda_svg_1:1.svg" />
11  <logo type="imda_svg_3:2" mimetype="image/svg+xml" url="http://imda.bbc.co.uk/imda/logos/imda_svg_3:2.svg" />
12  <logo type="imda_svg_4:3" mimetype="image/svg+xml" url="http://imda.bbc.co.uk/imda/logos/imda_svg_4:3.svg" />

```

```

13 | <logo type="imda_svg_16:9" mimetype="image/svg+xml" url="http://imda.bbc.co.uk/imda/"
14 | </logos>
    
```

## 7.6.1 | .../logos/logo@type

» attribute is required

The logo@type attribute allows us to identify some standard image profiles which would allow a solution to pick the closest image dimensions and type for their solution. There is also an 'unrestricted' profile that allows you to list any assets you may have that you think could be useful. The list of available profiles is as follows:

<i>type value</i>	<i>width</i>	<i>height</i>	<i>colour bit-depth</i>	<i>graphic type : and mime-type</i>	<i>Notes</i>
imda_micro_1:1	32px	32px	24-bit RGB	Portable Network Graphic v1.1/1.2: image/png	This format is intended for devices with very small screens or for logos which are displayed e.g. as part of a browseable station list. The resolution follows the DAB-EPG standard (ETSI TS 102.818). <b>Recommended</b>
imda_mobile_1:1	600px	600px	24-bit RGB	Portable Network Graphic v1.1/1.2: image/png	This format is the recommended size from many online music services and podcast content creators. <b>Mandatory</b>
imda_svg_1:1	n/a (the design should fit within a bounding square)	n/a	n/a	Scalable Vector Graphics v1.1: image/svg+xml	Useful for solutions who scale images. The use of the baseProfile attribute of the root svg node is recommended.

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

<i>type value</i>	<i>width</i>	<i>height</i>	<i>colour bit-depth</i>	<i>graphic type : and mime-type</i>	<i>Notes</i> <b>Optional</b>
imda_mobile_3:2	480px	320px	24-bit RGB	Portable Network Graphic v1.1/1.2: image/png	This format is intended for regular mobile devices with small screens. <b>Recommended</b>
imda_svg_3:2	n/a (the design should fit within a bounding rectangle with ratio 3:2)		n/a	Scalable Vector Graphics v1.1: image/svg+xml	Useful for solutions who scale images. The use of the <code>baseProfile</code> attribute of the root <code>svg</code> node is recommended. <b>Optional</b>
imda_mobile_4:3	640px	480px	24-bit RGB	Portable Network Graphic v1.1/1.2: image/png	This format will possibly be the most important one and therefore the provision should be mandatory for any broadcaster. The QVGA format can be made available by downscaling. <b>Recommended</b>
imda_svg_4:3	n/a (the design should fit within a bounding rectangle with ratio 4:3)		n/a	Scalable Vector Graphics v1.1: image/svg+xml	Useful for solutions who scale images. The use of the <code>baseProfile</code> attribute of the root <code>svg</code> node is recommended. <b>Optional</b>
imda_tv_4:3	768px	576px	24-bit RGB	Portable Network Graphic	This format is possibly the most

Mon, 05 Dec 2011  
 ion version: v2.2 | File version: svn#130 - 4

<i>type value</i>	<i>width</i>	<i>height</i>	<i>colour bit-depth</i>	<i>graphic type : and mime-type</i>	<i>Notes</i>
				v1.1/1.2: image/png	unimportant one as the future is HD ... therefore it could also be dropped from the table. <b>Recommended</b>
imda_tv_16:9	1920px	1080px	24-bit RGB	Portable Network Graphic v1.1/1.2: image/png	This format is aiming at IP-connected TV sets e.g. such as supporting HbbTV etc. Logos in that size shouldn't be just logos. They can include e.g. the stations claim or a background picture as well. <b>Recommended</b>
imda_svg_16:9	n/a (the design should fit within a bounding rectangle with ratio 16:9)	n/a		Scalable Vector Graphics v1.1: image/svg+xml	Useful for solutions who scale images. The use of the <code>baseProfile</code> attribute of the root <code>svg</code> node is recommended. <b>Optional</b>
imda_unrestricted	please specify the width and height of the graphic provide - there are no restrictions for this profile	n/a		Specify the correct image mime-type for the graphic supplied.	Use this if you cannot provide the logo in any of the preferred profiles listed above. Any graphic format could be supplied - but common formats would

05 Dec 2011)

Specification version: v2.2 | File version: svn#130

<i>type value</i>	<i>width</i>	<i>height</i>	<i>colour bit-depth</i>	<i>graphic type : and mime-type</i>	<i>Notes</i>
					be GIF, JPEG, PNG - other formats are unlikely to be supported by third-party solutions. <b>Optional</b>

**Table 15:** Value of *type* and the reasons for choosing them.

Some frequently asked questions about these profiles are available in Appendix: Frequently Asked Questions.

***These are the initial set of logo types as researched by the IMDA based on review of existing specifications on multiple devices and discussion around minimising these to some key size and format profiles. However – should you find that we've missed something or simply got something wrong, please don't hesitate to provide feedback.***

### 7.6.2 | .../logos/logo@url

» attribute is required

This is the Uniform Resource Locator (URL) to the image asset – it should be an absolute fully qualified URL that is publicly available. Organisation's images should be available via HTTP protocol, and publicly available.



*Some tips for the aggregator: your solution shouldn't rely on calling images directly off the organisation's web server it is always preferable for your solution to cache the logo on your local servers, this obviously is required if your solution requires to process a logo and cache it at a different size, colour depth or graphic format (gif, jpeg,...).*

### 7.6.3 | .../logos/logo@background

» attribute is optional

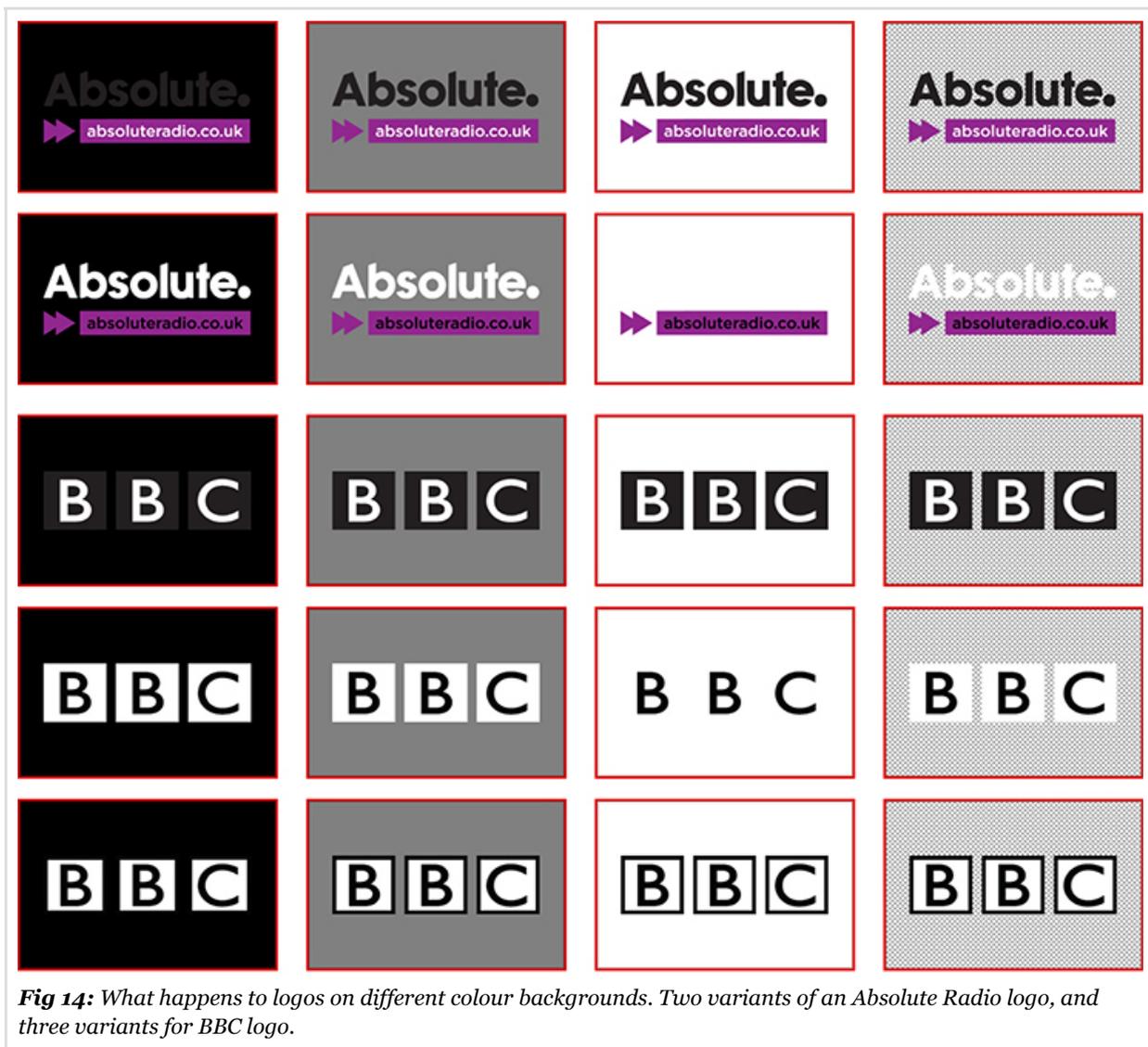
The `background` attribute is an optional attribute. This attribute allows you, as a media organisation, to signal to the aggregator of your data whether a logo is appropriate to be displayed on a particular background. If you do not specify this attribute, the aggregator should assume this means that the logo in this node can be used on any background. If described colour is not available on target solution, then solution will choose closest available colour. The values for `background` are taken from the W3C's Cascading Style Sheets 2.1 (r1) specification [<http://bit.ly/css2colors>], though we currently limit these to

black, silver, gray and white.

<i>background value</i>	<i>CSS colour</i>	<i>Colour swatch</i>	<i>HEX RGB</i>
black	black		#000000
silver	silver		#C0C0C0
gray	gray		#808080
white	white		#FFFFFF

**Table 16:** Valid values of *background*.

Why would you need to provide this attribute? Well, consider the following diagram. Each row represents one particular logo variant, with the columns showing a different background colour in use (the variant on the far right shows a hatched pattern as a background to show which bits of the logo are transparent). Disclaimer – these mock-ups do not necessarily follow brand useage guideines from the owners of the logos, and are merely used here to illustrate the issue.



Specification version: v2.2 | File version: svm#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)

Probably the most obvious from this is when you have the BBC logo on a black background, or the reversed BBC logo on a white background – I hope you'll agree that this is a problem. So, in order to address this – the background attribute is available so that you can provide several logo nodes for one type. Also – we picked four colours for background as these are probably the most common, although in the diagram above we have only used three due to space.

Note that you can have only one logo node per background colour, and that you may have one logo node that doesn't specify background – this would be the default logo to use, and any others specified would override that logo based on the background colour. So, for example, the BBC could specify 3 logo nodes (for one type) for the logo of their 'Media Organisation' – one would have no background attribute and would link to the third BBC variant (on the diagram above - black lettering on white blocks surrounded by black lines), another node would link to the second BBC variant (black letters on white blocks) with a background value of black, another node would link to the first BBC variant (white letters on black blocks) with a background value of white.

---

## *Appendix A* | **XML Schemas**

Schemas are not available as yet – we have draft schemas available but are waiting on feedback regarding this version of the specification before finalising.

---

## Appendix B | Example documents

The example xml documents for each of the main areas used in this specification can be downloaded here:

- Root: imda.xml
- Organisation: imda\_org.xml
- Brands: imda\_brands.xml
- Transports: imda\_transports.xml
- Manifests: imda\_manifests.xml

We also have a set of these documents which contain only the minimum metadata required. Please note that these only contain space for a very simple media organisation that has one brand and only one transport for that brand. We aim to make available, via the IMDA website, a straightforward form-based wizard that allows you to create these xml documents for very simple media organisations.

- Minimum Root: imda.xml
- Minimum Organisation: imda\_org.xml
- Minimum Brands: imda\_brands.xml
- Minimum Transports: imda\_transports.xml

## Appendix C | Frequently Asked Questions

### General issues:

*Why did you decided to use XML documents to share metadata in this way? ~*  
XML documents are commonly used for sharing structured data, it also allows for zero-resource for smaller organisations with minimum data sets that are manually produced.

*Why have you chosen your XML document to use UTF-8 language encoding? ~*  
See "Why UTF-8?" for an answer to this.

*You could have put everything in the one XML document, why have you split it up into multiple documents? ~*  
See "Why split data into multiple XML documents?" for an answer to this.

### Issues relating to Logos (see the shared node information on Logos):

*Why are the bitmap types all Portable Network Graphics (PNG) format? ~*  
Graphics Interchange Format (GIF) and Joint Pictures Expert Group (JPEG) are common, however in order to support all everything they can represent plus including alpha channels properly, the use of PNG is required. Remember - this is what the organisation provides, even if a solution resizes/resamples/transcodes this to another format for their needs. There is a useful comparison of common image types to PNG available on Wikipedia.

*Why bother to provide the Scalable Vector Graphics (SVG) types? ~*  
If your logos can be described as an SVG, then you should consider providing these as the solution could scale these to whatever size they require, with no loss of quality of image – the solution could then cache them as any format of image they wish that's supported in their solution. Unlike with the bitmap formats, where scaling will have an impact on the quality of the resulting image.

*What about file size? ~*  
Clearly the idea here is to have a balance of best quality and file size – it may well be the case that a solution picks the nearest to it's needs and then resizes/resamples/transcodes this to another format. If you are providing an SVG type then you can provide a link to either the uncompressed .svg or the zipped-compressed .svgz

*Why have you specified these colour bit-depths? ~*  
These are guidelines about the colour support of some solutions. However - all solutions vary on their support based on the type and abilities of the display they use. Therefore the solution may choose to resample your logos into a colour palette that works with the output device they are using.

*Should I be worried about 'tv-safe' colour palettes? ~*  
it was important to make sure for Cathode Ray Tube (CRT) based televisions and displays to make sure that the colour palette of any image to be displayed on these were limited to avoid colour bleeding across scan lines. This was only a small number of RGB colours that were too bright, and common image editing software (e.g. Photoshop) has a filter for these. Note that extremes like white and black

were avoided by limiting them to a light grey (RGB - 240, 240, 240) and a dark grey (RGB - 16, 16, 16). However - on HDTVs (LCD and LED based screens) it seems less likely that this sort of problem would persist. We currently do not have enough information regarding this area – if you have any information on this, please do get in touch. Therefore, at this time, we don't require you to use a tv-safe colour palette for your larger images. Should you already have 'tv-safe' images in these sizes, then this is ok.

Issues relating to Geographical reference systems in relation to our `locations` nodes:

*Can GeoRSS polygons define a polygon passing the +180/-180 longitude (International Date Line)? ~*

The specification of GeoRSS does not explicitly state that a polygon can cross this border, but an example is given of a polygon (drawn as rectangle) crossing this border.

*Can GeoRSS polygons define polygons with a hole in them? ~*

No, but using the reference type of location you could specify one polygon location, then another, then use the reference grouping type to 'exclude' (i.e. subtract) one polygon from another.

*Is there any ambiguity in the GeoRSS specification? ~*

Yes, we think so. It seems that GeoRSS polygons have a restriction: given two random distinct coordinate pairs in a polygon, neither their latitudes nor their longitudes may be separated by more than 179 degrees. If they would span more than 179 degrees, then the definition becomes ambiguous about which way they should wrap around the earth. This implies that a single polygon cannot cover more than appr. 50% of the earth – a work around is to group polygon locations together to create an area that covers more than 50%, but individually don't account from 50% or more. However, we feel that this issue is probably unlikely to be a problem for most, if you have an issue with this please do provide us with feedback.

*Why GeoRSS, there are plenty of other Geographical reference systems? ~*

We took a look at some of the common reference systems out there: including OpenGIS and GML as well as GeoRSS (Simple). We currently feel that the simplicity of GeoRSS (Simple) is suitable for the specification in those present in the discussions. If you find you are unable to represent locations using the methods provided then we ask you to provide feedback.

***Do you have a question we don't have an answer for in this document? We'd love to hear from you – help us make things better by providing feedback.***

alan.ogilvie ]

Specification version: v2.2 | File version: svn#130 - 2011-12-05 09:06:20 +0000 (Mon, 05 Dec 2011)