



R&D White Paper

WHP 060

July 2003

A Java API to TV-Anytime data

T. Ferne, T. Sargeant *and* C. Akanbi

A Java API to TV-Anytime data

Tristan Ferne, Tim Sargeant and Chris Akanbi

Abstract

This technical note describes an API designed and written by the authors to allow applications to access and process TV-Anytime data. The API provides a Java representation of TV-Anytime data and an easy-to-use interface to TV-Anytime XML, delivered in a file, a stream or over IP. Note that at the current time the API does not support the entire scope of the TV-Anytime standards.

This document is intended primarily as reference material for those requiring an understanding of the structure of the API, and for those who are intending to make use of it. The API is accompanied by Java API documentation in an electronic form that describes the API in full.

A number of applications written at BBC Research & Development, running on PC and STB platforms have already made use of the API. Some simple illustrative code examples are included in this technical note.

An understanding of TV-Anytime data and Java programming is assumed. The API is written for use with Java version 1.1 and any later releases. XML processing requires the Java API for XML processing (JAXP).

Keywords

Java, TV-Anytime, XML, SAX, API, Metadata, Programme Description, PVR, PDR

White Papers are distributed freely on request.
Authorisation of the Chief Scientist is required for
publication.

© BBC 2003. All rights reserved. Except as provided below, no part of this document may be reproduced in any material form (including photocopying or storing it in any medium by electronic means) without the prior written permission of BBC Research & Development except in accordance with the provisions of the (UK) Copyright, Designs and Patents Act 1988.

The BBC grants permission to individuals and organisations to make copies of the entire document (including this copyright notice) for their own internal use. No copies of this document may be published, distributed or made available to third parties whether by paper, electronic or other means without the BBC's prior written permission. Where necessary, third parties should be directed to the relevant page on BBC's website at <http://www.bbc.co.uk/rd/pubs/whp> for a copy of this document.

A Java API to TV-Anytime data

Tristan Ferne, Tim Sargeant and Chris Akanbi

Contents

1	Introduction	1
1.1	TV-Anytime.....	1
1.2	The BBC R&D TV-Anytime Java API.....	1
2	Licensing	3
2.1	Support and contact details	3
3	The Java TV-Anytime API.....	4
3.1	TVA specification level	4
3.2	TVA XML tables	4
3.3	Scope.....	5
3.4	Java requirements	5
3.5	Coding conventions.....	5
3.5.1	Methods implemented by API objects	5
3.5.2	Setting and retrieving values	6
3.5.3	Recommended and optional fields	6
3.6	The API model	6
4	API Packages	8
4.1	Introduction.....	8
4.2	MPEG7 package (bbc.rd.mpeg7)	9
4.2.1	MPEG7MediaDuration.....	9
4.2.2	MPEG7MediaIncrDuration	9
4.2.3	MPEG7MediaLocator.....	10
4.2.4	MPEG7MediaRelIncrTimePoint.....	10
4.2.5	MPEG7MediaRelTimePoint	10
4.2.6	MPEG7MediaTimePoint.....	10
4.3	TV-Anytime package (bbc.rd.tvanytime)	11
4.3.1	BasicDescription	12
4.3.2	CaptionLanguage	12
4.3.3	ContentReference.....	12
4.3.4	Duration	12
4.3.5	Genre.....	12
4.3.6	MemberOf	12
4.3.7	PromotionalInformation.....	12

4.3.8	RelatedMaterial	12
4.3.9	SignLanguage	12
4.3.10	Synopsis	12
4.3.11	Title	12
4.3.12	URI	13
4.3.13	TVAnytimeException (exception)	13
4.4	Content Referencing package (bbc.rd.tvanytime.contentReferencing)	14
4.4.1	ContentReferencingTable	15
4.4.2	CRIDResult	15
4.4.3	LocationsResult	15
4.4.4	Locator	15
4.4.5	Result	15
4.5	Credits Information package (bbc.rd.tvanytime.creditsInformation)	16
4.5.1	Character	16
4.5.2	CreditsItem	16
4.5.3	CreditsList	17
4.5.4	Name	17
4.5.5	OrganizationName	17
4.5.6	PersonName	17
4.6	Group Information package (bbc.rd.tvanytime.groupInformation)	18
4.6.1	GroupInformation	18
4.6.2	GroupInformationTable	18
4.7	Program Information package (bbc.rd.tvanytime.programInformation)	19
4.7.1	AspectRatio	19
4.7.2	AudioAttributes	19
4.7.3	AVAttributes	20
4.7.4	ProgramInformation	20
4.7.5	ProgramInformationTable	20
4.7.6	VideoAttributes	20
4.8	Program Location package (bbc.rd.tvanytime.programLocation)	21
4.8.1	ProgramLocationTable	22
4.8.2	ProgramURL	22
4.8.3	Schedule	22
4.8.4	ScheduleEvent	22
4.9	Searching package (bbc.rd.tvanytime.search)	23
4.9.1	LocationResolution (interface)	23
4.9.2	MetadataSearch (interface)	23
4.9.3	SearchInterfaceNotSupportedException (exception)	23
4.10	Segment Information package (bbc.rd.tvanytime.segmentInformation)	24
4.10.1	BasicSegmentDescription	25
4.10.2	SegmentGroupInformation	25
4.10.3	SegmentGroupList	25
4.10.4	SegmentGroupType	25
4.10.5	SegmentInformation	25
4.10.6	SegmentInformationTable	25
4.10.7	SegmentList	25
4.10.8	SegmentLocator	25
4.11	Service Information package (bbc.rd.tvanytime.serviceInformation)	26
4.11.1	ServiceInformation	26
4.11.2	ServiceInformationTable	26

4.12	Utilities package (bbc.rd.tvanytime.util).....	27
4.12.1	DVBLocatorToolbox.....	27
4.12.2	GenreToolbox	27
4.12.3	TimeToolbox.....	27
4.12.4	TVAnytimeGenreToolbox	28
4.12.5	TVAnytimeHowRelatedToolbox.....	28
4.13	XML Parser package (bbc.rd.tvanytime.xml).....	29
4.13.1	NonFatalXMLException (exception)	29
4.13.2	SAXXMLParser	29
4.13.3	XMLParser (interface).....	30
5	Code examples	31
5.1	Parsing TV-Anytime data using SAX.....	31
5.2	Generating TV-Anytime XML tables	32
5.3	Searching tables for schedules.....	34
5.4	Retrieving metadata and location information.....	35
6	References	37
7	Bibliography.....	38
	Appendix A.....	39
7.1	Document Conventions.....	39
7.2	Diagrams	39

Index of Figures

Figure 1: TV-Anytime XML table structure.....	4
Figure 2: TVA API.....	7
Figure 3 - bbc.rd.mpeg7	9
Figure 4 - bbc.rd.tvanytime.....	11
Figure 5 - bbc.rd.tvanytime.contentReferencing	14
Figure 6 - bbc.rd.tvanytime.creditsInformation	16
Figure 7 - bbc.rd.tvanytime.groupInformation	18
Figure 8 - bbc.rd.tvanytime.programInformation	19
Figure 9 - bbc.rd.tvanytime.programLocation	21
Figure 10 - bbc.rd.tvanytime.search	23
Figure 11 - bbc.rd.tvanytime.segmentInformation	24
Figure 12 - bbc.rd.tvanytime.serviceInformation	26
Figure 13 - bbc.rd.tvanytime.util	27
Figure 14 - bbc.rd.tvanytime.xml	29

Index of Tables

Table 1: TV-Anytime table contents	5
Table 2: Methods implemented by API objects.....	6
Table 3: XML Generation, code example.....	33
Table 4: XML Generation, output	33

1 Introduction

1.1 TV-Anytime

The *TV-Anytime* Forum is a global association of organizations that develops specifications to enable audio-visual and other services based on mass-market high volume digital storage in consumer platforms.

In addition to an overall systems specification, TV-Anytime have produced two other key specifications, namely:

- Metadata
- Content Referencing and Location Resolution

Metadata is the term used to describe information about TV programmes, radio programmes and other content, which can be human readable (title, genre, etc.) and/or machine-readable (audio type, aspect ratio, etc.). The primary use of TV-Anytime metadata is to attract users to content and to provide users with enhanced information on that content. By delivering metadata in a standard format, the user is able to discover new content in easy, and potentially novel, ways.

The *content referencing* specification presents a standardised way to find (make reference to) a piece of content. In this context, "content" can refer to an individual TV programme, or to a large group of programmes that are in some way related. Each of these entities is assigned a content reference identifier (CRID). By using the CRID, it is possible for a recorder to capture the content by means of the location resolution process. This process allows a device to capture a programme or its repeats, regardless of which channel it is broadcast on and even when the broadcast schedule changes.

1.2 The BBC R&D TV-Anytime Java API

This technical note describes an API designed and written by the authors to allow applications access to TV-Anytime data. The API provides a Java representation of TV-Anytime data and an easy-to-use interface to TV-Anytime XML, delivered in a file, a stream or over IP.

The API is intended to provide a single, stable interface to TV-Anytime data from a number of sources. For example, an application to display an EPG uses the same interface to TV-Anytime data as an application for recommending programmes. As a result, when TV-Anytime specifications change, only one change to the API is required, and all applications will go on functioning as intended. It is hoped that by providing a simple to use interface to TVA tables, which abstracts away from some of the complexity of the data, the use of the TV-Anytime open standard will be encouraged.

A number of applications written at BBC Research & Development, running on PC and STB platforms have already made use of the API, including:

- Programme recommendation agents
- TV-Anytime head-end
- TV-Anytime client PVR
- Segment Authoring application
- MHP STB Segment browser
- 2nd Screen

The document is intended primarily as reference material for those requiring an understanding of the structure of the API, and for those who are intending to make use of it. The API is accompanied by Java API documentation in an electronic form that describes the API in full.

Some simple illustrative code examples are included in this technical note.

An understanding of TV-Anytime data and Java programming is assumed.

2 Licensing

The BBC R&D TV-Anytime Java API is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

The BBC R&D TV-Anytime Java API 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 Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with the BBC R&D TV-Anytime Java API; if not, write to the Free Software Foundation, Inc., 59 TemplePlace, Suite 330, Boston, MA 02111-1307 USA.

2.1 Support and contact details

We do not guarantee to provide any support, bug-fixes or updates to the TV-Anytime Java API however we will do our best to answer any queries, fix any bugs and keep the API up to date with the specifications. For any issues of this kind please contact:

Tristan Ferne: tristan.ferne@rd.bbc.co.uk

Tim Sargeant: tim.sargeant@rd.bbc.co.uk

3 The Java TV-Anytime API

3.1 TVA specification level

The TVA java API is written to the following TV-Anytime specifications:

- SP003 Metadata (Normative), version 1.3 ^[1]
- SP003 Metadata - Corrigenda 2 to S-3 v1.3, 21/03/2003 ^[2]
- SP004 Content Referencing (Normative), version 1.2 ^[3]
- SP004 Content Referencing - Corrigenda 1 to SP004 v1.2 ^[4]

3.2 TVA XML tables

The TV-Anytime data is presented in a number of XML tables, which are (with the exception of the Content Referencing table) contained within the XML root element “TVAMain”, as shown in figure 1.

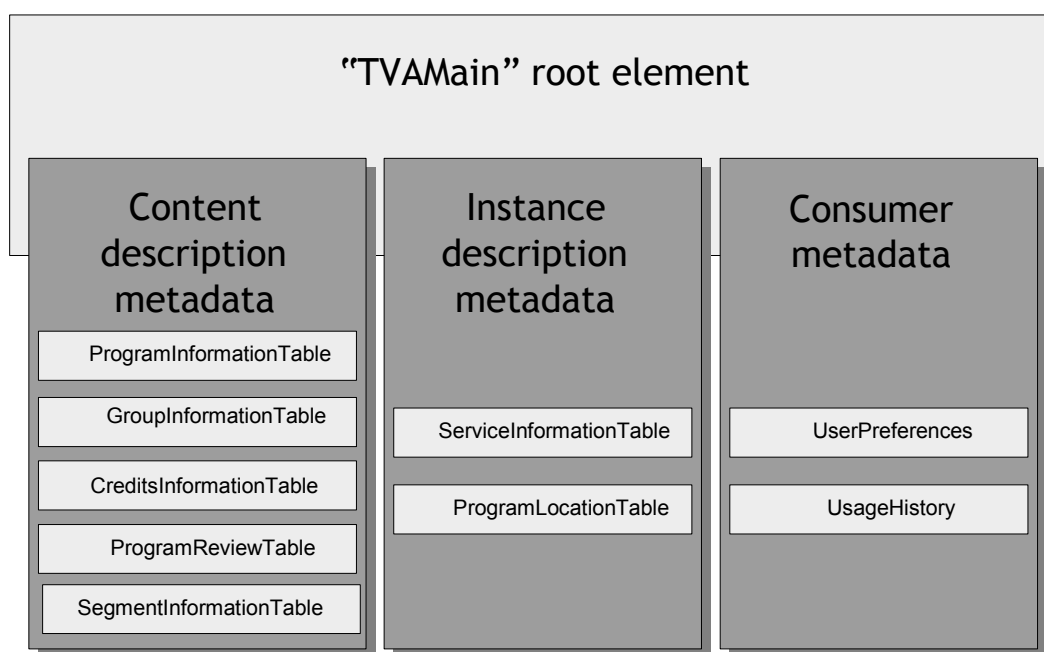


Figure 1: TV-Anytime XML table structure

The data contained in each of the tables is shown in Table 1.

Name of table	Data
<i>ContentReferencingTable</i>	CRID to location and CRID to CRID mappings.
<i>ProgramInformationTable</i>	Attractor metadata (title, synopsis, genres, etc).
<i>ProgramLocationTable</i>	EPG-like information .
<i>ServiceInformationTable</i>	Mapping of locators to human-readable service names.

<i>SegmentInformationTable</i>	Metadata and timing information for programme segments.
<i>GroupInformationTable</i>	Metadata (including descriptions) of TV-Anytime groups.
<i>CreditsInformationTable</i>	Detailed cast and crew (credits) information.
<i>ProgramReviewTable</i>	Contains critical reviews of items of content.
<i>UserPreferences</i>	Part of the consumer metadata, this details rich descriptions of the types of content being requested by the user – their preferences or profile.
<i>UsageHistory</i>	Describes a history of the user's actions (click data).

Table 1: TV-Anytime table contents

3.3 Scope

Not all of the data detailed in the TV-Anytime specification is included in this API at present. (Though it is written in such a way as to be expandable, and additions can be made as required.)

The following tables are included in the API:

- *ContentReferencing*
- *CreditsInformation*
- *GroupInformation*
- *ProgramInformation*
- *ProgramLocation*
- *SegmentInformation*
- *ServiceInformation*

The API provides methods to access all the recommended data fields within these tables, and any optional fields for which data is generated by the BBC at present.

The TVAMain XML root element is supported, in so much as tables contained within this element are parsed successfully by the SAX parser, but within the API, no concept of TVAMain is represented. Similarly, where `toXML()` methods are used for generating TV-Anytime XML tables, these are not placed within the TVAMain root element.

3.4 Java requirements

The API is written for use with Java version 1.1 and any later releases. XML processing requires the Java API for XML processing (JAXP). This is included in JDK/JRE 1.4 and above or is available for download from Sun as the JAXP reference implementation (Xerces2 or Crimson).

3.5 Coding conventions

Whilst the API was authored by three programmers, a coding convention has been strictly followed so that the sections of the API behave in a consistent manner.

3.5.1 Methods implemented by API objects

In particular, for each object representing a TV-Anytime table object, the following methods are provided:

Method name	Purpose
<code>clone()</code>	Create a field-for-field copy of instances of the object (implements <code>java.lang.Cloneable</code>).
<code>toString()</code>	Returns a string representation of the object.
<code>toString(int n)</code>	Returns a string representation of the object, with an indentation of 'n' tabs.
<code>toXML()</code>	Returns an XML representation of the object.
<code>toXML(int n)</code>	Returns an XML representation of the object, with an indentation of 'n' tabs.

Table 2: Methods implemented by API objects

3.5.2 Setting and retrieving values

For objects with single children or variables, `setXxx(...)` and `getXxx()` methods are provided where `Xxx` is the name of the variable or child object.

For objects with a number of children, `addXxx(...)` and `getXxx(int index)` methods are provided. In addition there should be `getNumXxxs()` and `removeAll()` methods.

In most cases, two constructors are provided – one allows the user to create an empty object, and the other allows the user to create an object initialised with all mandatory values. Obviously, the set methods can then be used to configure the object appropriately.

3.5.3 Recommended and optional fields

Currently the API supports all recommended fields in the TV-Anytime tables listed in section 3.3. Additionally, a number of optional fields are supported. Generally, the decision as to which optional fields to support is based on the TV-Anytime data set available at BBC Research & Development at the time of writing. See sections 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11, 4.12 and 4.13 for details of the contents of each package in the API.

3.6 The API model

A structural model of the TVA API is shown in Figure 2.

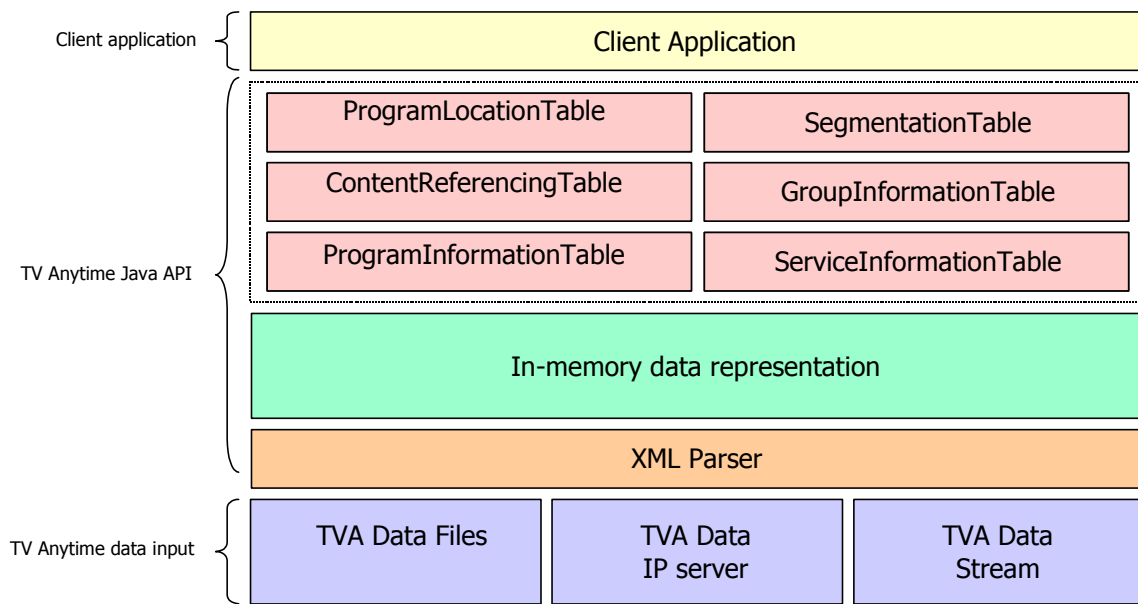


Figure 2: TVA API

A client application is shown at the top of the diagram, using the Java TV Anytime API to access an in-memory representation of TV Anytime data. The bottom of the diagram shows valid TV Anytime data sources, including XML files, responses from an IP server to a given protocol, or data streams.

4 API Packages

4.1 Introduction

The API is structured in several java packages:

Packages containing general objects (TV-Anytime defined and MPEG7 inherited)

- `bbc.rd.mpeg7`
- `bbc.rd.tvanytime`

Packages representing each TV-Anytime table

- `bbc.rd.tvanytime.contentReferencing`
- `bbc.rd.tvanytime.creditsInformation`
- `bbc.rd.tvanytime.groupInformation`
- `bbc.rd.tvanytime.programInformation`
- `bbc.rd.tvanytime.programLocation`
- `bbc.rd.tvanytime.segmentInformation`
- `bbc.rd.tvanytime.serviceInformation`

Packages for sourcing data from XML tables

- `bbc.rd.tvanytime.xml`

Packages offering other features (searching tables and miscellaneous utilities)

- `bbc.rd.tvanytime.search`
- `bbc.rd.tvanytime.util`

The following sections describe the function of each package (in alphabetical order) in more detail.

4.2 MPEG7 package (bbc.rd.mpeg7)

Some MPEG-7 defined datatypes are used within the TV Anytime specifications, in particular within the SegmentInformationTable. These datatypes are contained within this package.

MPEG7MediaDuration	MPEG7MediaTimePoint
+MPEG7MediaDuration(): MPEG7MediaDuration +MPEG7MediaDuration(in duration:long): MPEG7MediaDuration +setTime(in duration:long) +getTime(): long +toXML(): String +toXML(in indent:int): String +toString(): String +toString(in indent:int): String +clone(): Object	+MPEG7MediaTimePoint(): MPEG7MediaTimePoint +MPEG7MediaTimePoint(in timePoint:long): MPEG7MediaTimePoint +setTime(in timePoint:long) +getTime(): long +toXML(): String +toXML(in indent:int): String +toString(): String +toString(in indent:int): String +clone(): Object
MPEG7MediaRelTimePoint	MPEG7MediaRelIncrTimePoint
+MPEG7MediaRelTimePoint(): MPEG7MediaRelTimePoint +MPEG7MediaRelTimePoint(in timePoint:long): MPEG7MediaRelTimePoint +setTime(in timePoint:long) +getTime(): long +toXML(): String +toXML(in indent:int): String +toString(): String +toString(in indent:int): String +clone(): Object	+MPEG7MediaRelIncrTimePoint(): MPEG7MediaRelIncrTimePoint +MPEG7MediaRelIncrTimePoint(in timePoint:long): MPEG7MediaRelIncrTimePoint +setTime(in timePoint:long) +getTime(): long +setTimeUnit(timeUnit:String) +getTimeUnit(): String +toXML(): String +toXML(in indent:int): String +toString(): String +toString(in indent:int): String +clone(): Object
MPEG7MediaIncrDuration	MPEG7MediaLocator
+MPEG7MediaIncrDuration(): MPEG7MediaIncrDuration +MPEG7MediaIncrDuration(in duration:long): MPEG7MediaIncrDuration +setTime(in duration:long) +getTime(): long +setTimeUnit(timeUnit:String) +getTimeUnit(): String +toXML(): String +toXML(in indent:int): String +toString(): String +toString(in indent:int): String +clone(): Object	+MPEG7MediaLocator(): MPEG7MediaLocator +MPEG7MediaLocator(mediaURI:URI): MPEG7MediaLocator +setMediaURI(mediaURI:URI) +getMediaURI(): URI +toXML(): String +toXML(in indent:int): String +toString(): String +toString(in indent:int): String +clone(): Object

Figure 3 - bbc.rd.mpeg7

4.2.1 MPEG7MediaDuration

MPEG7MediaDuration: This is an element specifying the duration of a media time period according to days and day time.

4.2.2 MPEG7MediaIncrDuration

MPEG7MediaIncrDuration: This is an element specifying the duration of a media time period counting time units.

4.2.3 *MPEG7MediaLocator*

MPEG7MediaLocator: This is used to specify the "location" of a particular image, audio or video segment by referencing the media data.

4.2.4 *MPEG7MediaRelIncrTimePoint*

MPEG7MediaTimePoint: This is an element specifying a media time point relative to a time base counting time units.

4.2.5 *MPEG7MediaRelTimePoint*

MPEG7MediaTimePoint: This is an element specifying a media time point using a number of days and day time.

4.2.6 *MPEG7MediaTimePoint*

MPEG7MediaTimePoint: This is an element specifying a media time point using Gregorian date and day time.

4.3 TV-Anytime package (bbc.rd.tvanytime)

This package contains many general objects that are used within TV-Anytime.

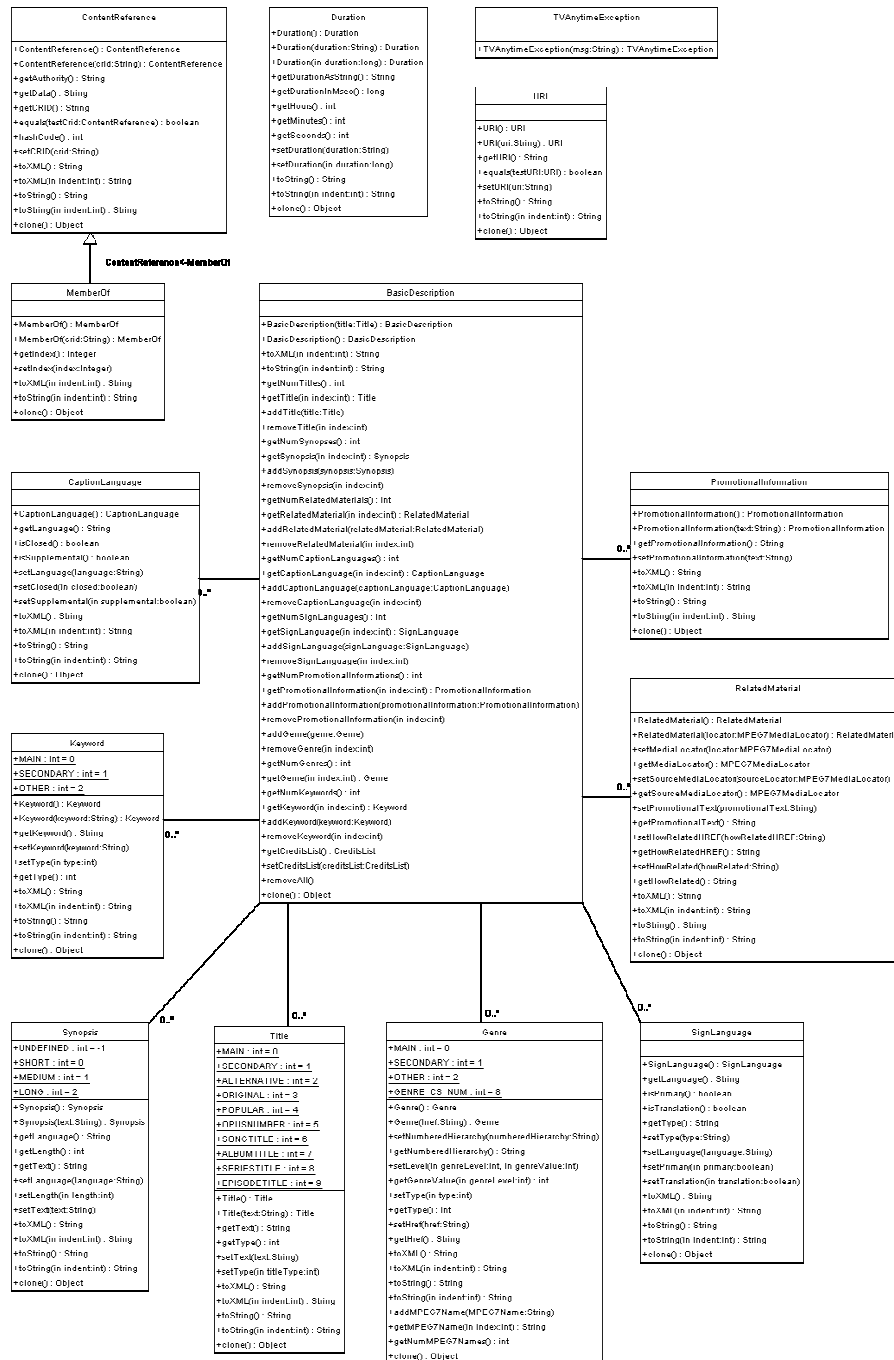


Figure 4 - bbc.rd.tvanytime

4.3.1 BasicDescription

BasicDescription: This contains objects that provide descriptive metadata about a programme, including Title, Synopsis, Genre, SignLanguage, CaptionLanguage, RelatedMaterial and PromotionalInformation.

4.3.2 CaptionLanguage

CaptionLanguage: This represents information about the caption language. Contains information about the caption language used and whether captioning is closed.

4.3.3 ContentReference

ContentReference: This represents a TV-Anytime CRID.

4.3.4 Duration

Duration: This represents a duration (of the form "PT00H25M30S");

4.3.5 Genre

Genre: This represents a TV-Anytime genre. Information includes the numbered hierarchy representation as defined in TV Anytime's genre classification scheme, the name of the genre, and the genre type (main, other or secondary). See also section 4.12.4.

4.3.6 MemberOf

MemberOf: This represents a CRID that is used to identify a group that a programme is a member of.

4.3.7 PromotionalInformation

PromotionalInformation: This represents a PromotionalInformation object, containing any promotional text.

4.3.8 RelatedMaterial

RelatedMaterial: This represents a RelatedMaterial object. Used to refer to other media assets that are related to a programme. This object includes information as to how the material is related (according to TVA's HowRelated classification scheme), any promotional text, and a media locator for the related material. See also section 4.12.5.

4.3.9 SignLanguage

SignLanguage: This represents a SignLanguage object. Includes the sign language used (e.g. BSL), whether signing is a translation from the original language, and whether this is the primary sign language.

4.3.10 Synopsis

Synopsis: This represents a Synopsis object that provides a textual description of a programme. Also contains a 'length' field, indicating 'SHORT', 'MEDIUM' or 'LONG'.

4.3.11 Title

Title: This represents a title of a programme. Also contains a title type, with values including MAIN, ORIGINAL, EPISODETITLE, POPULAR, SERIESTITLE and so on.

4.3.12 *URI*

URI: This represents a URI. Java 1.1 does not contain `java.net.URI`, so we have provided our own, very simple, URI type.

4.3.13 *TVAnytimeException (exception)*

`TVAnytimeException`: This is an exception to be thrown by classes within the TV-Anytime API.

4.4 Content Referencing package (bbc.rd.tvanytime.contentReferencing)

The Content Referencing table provides a means by which CRIDs can be ‘resolved’ into child CRIDs or locators.

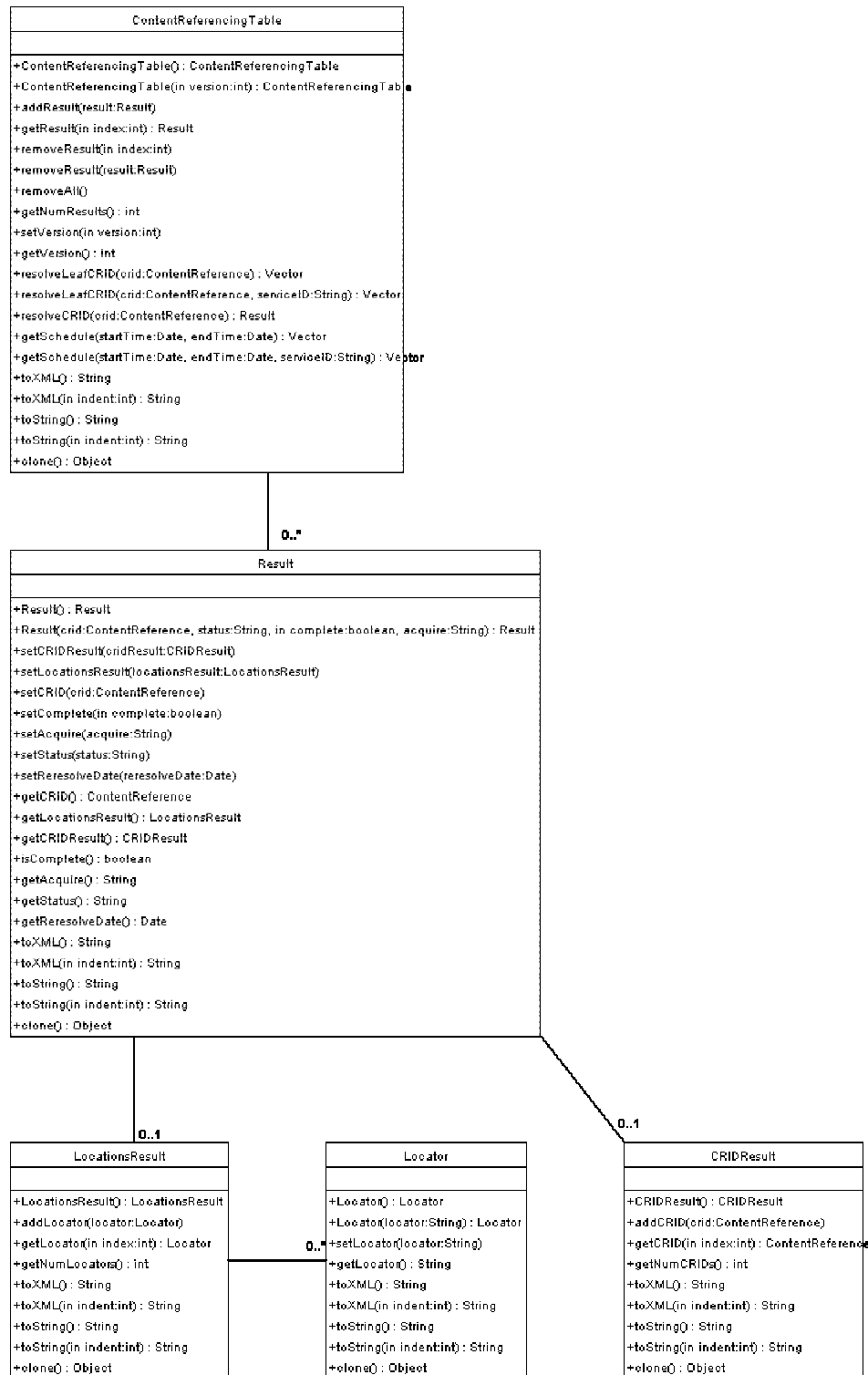


Figure 5 - bbc.rd.tvanytime.contentReferencing

4.4.1 *ContentReferencingTable*

ContentReferencingTable: This represents a table that contains ContentReferencing objects.

4.4.2 *CRIDResult*

CRIDResult: This contains a number of ContentReference objects.

4.4.3 *LocationsResult*

LocationsResult: This contains a number of Locator objects.

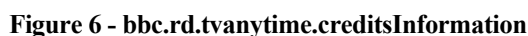
4.4.4 *Locator*

Locator: This represents a generic TV Anytime Locator object. The intention is that utility classes are provided for each specific transport system (see section 4.12.1).

4.4.5 *Result*

Result: This contains may contain either a LocationsResult or CRIDResult object, as well as various attributes about the resolution, including re-resolve date, acquisition instructions, and so on.

Credits information contains data about people acting in or involved with the production of this content. It can be useful for searching for programmes based on actor name, for example.



Character: Represents the name of a character appearing in a programme.

CreditsItem: Represents an item in the CreditsInformation table

4.5.3 CreditsList

CreditsList: This represents a list of CreditsItems

4.5.4 Name

Name: This represents a name.

4.5.5 OrganizationName

OrganizationName: This represents the name of an organisation

4.5.6 PersonName

PersonName: This represents the name of a credited person.

4.6 Group Information package (bbc.rd.tvanytime.groupInformation)

In TV-Anytime programmes can be grouped into “Programme Group” elements such that a group may contain any number of programmes, and a programme can be a member of any number of groups. Furthermore, programme groups themselves can be part of other programme groups. The group information package is used to describe these groups and their relationships.

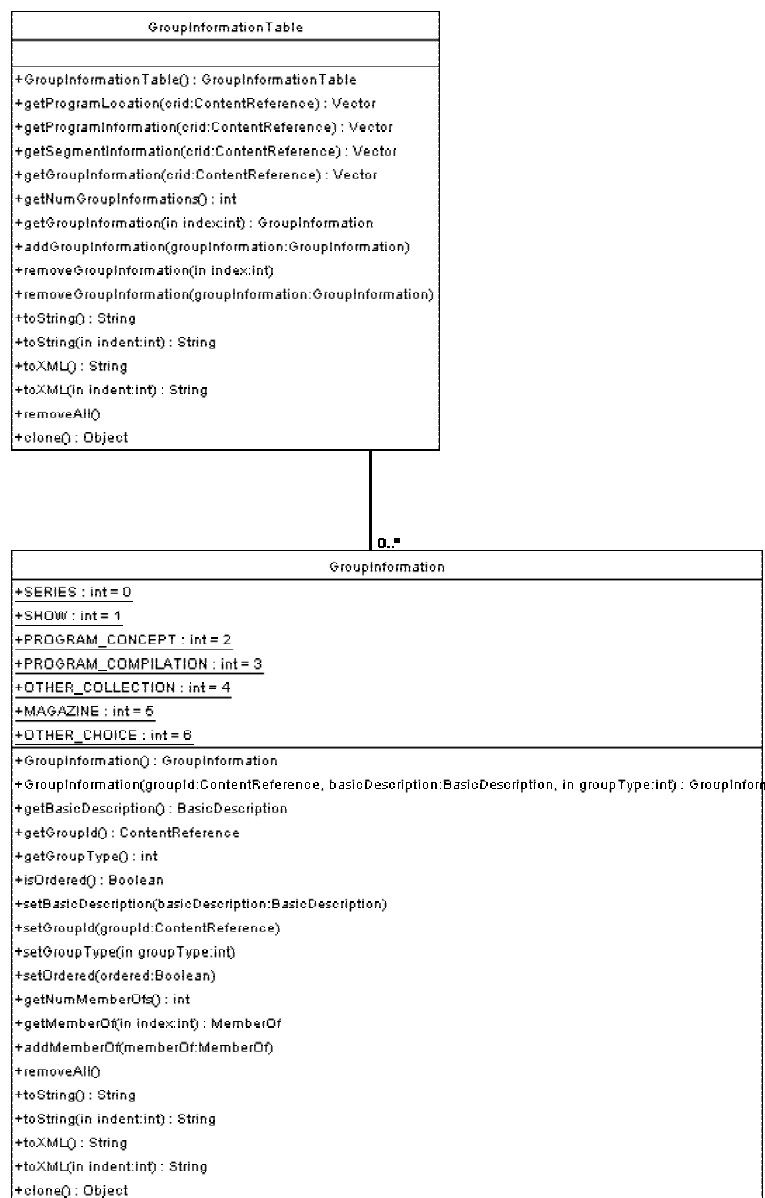


Figure 7 - bbc.rd.tvanytime.groupInformation

4.6.1 GroupInformation

GroupInformation: This is used to describe a TV-Anytime group.

4.6.2 GroupInformationTable

GroupInformationTable: This is a table that contains GroupInformation objects.

4.7 Program Information package (bbc.rd.tvanytime.programInformation)

Program information data is metadata that describes a piece of content. It includes information such as the content's title, textual description, and genre. It is general information about a piece of content that does not change regardless of how the content is published or broadcast.

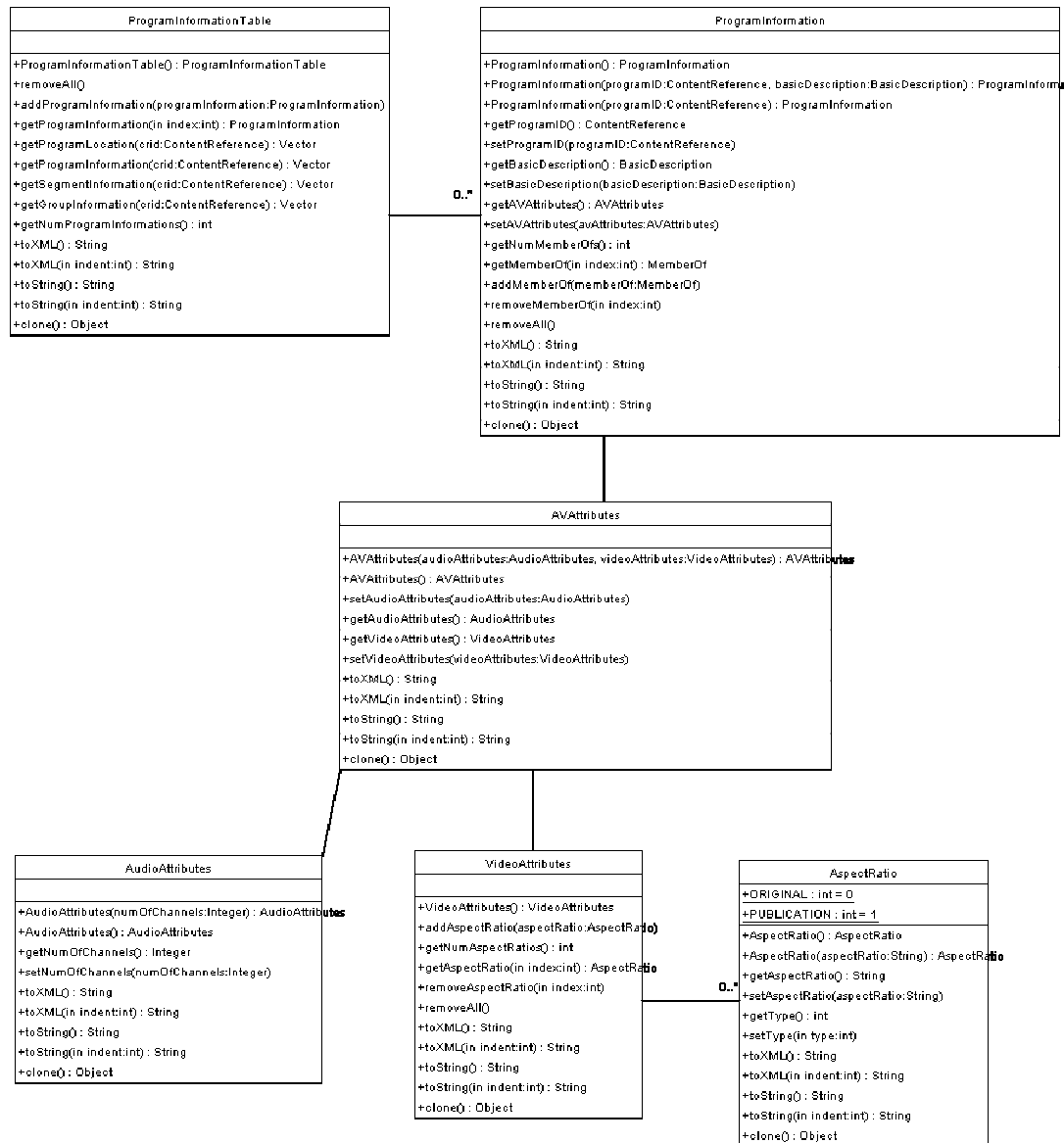


Figure 8 - bbc.rd.tvanytime.programInformation

4.7.1 AspectRatio

AspectRatio: This represents the aspect ratio of the video of a programme.

4.7.2 AudioAttributes

AudioAttributes: This represents technical audio attributes of a programme, including number of audio channels.

4.7.3 AVAttributes

AVAttributes: This represents audio-visual information about a programme. Contains an AudioAttributes object and a VideoAttributes object.

4.7.4 ProgramInformation

ProgramInformation: This represents the descriptive information of a programme.

4.7.5 ProgramInformationTable

ProgramInformationTable: This represents a table that contains ProgramInformation objects.

4.7.6 VideoAttributes

VideoAttributes: This represents technical video attributes of a programme. Contains a number of AspectRatio objects (for use where the original aspect ratio and available aspect ratios are different).

4.8 Program Location package (bbc.rd.tvanytime.programLocation)

Program location data is metadata that gives time, duration and service information for a piece of content. It describes a particular instance of a piece of content.

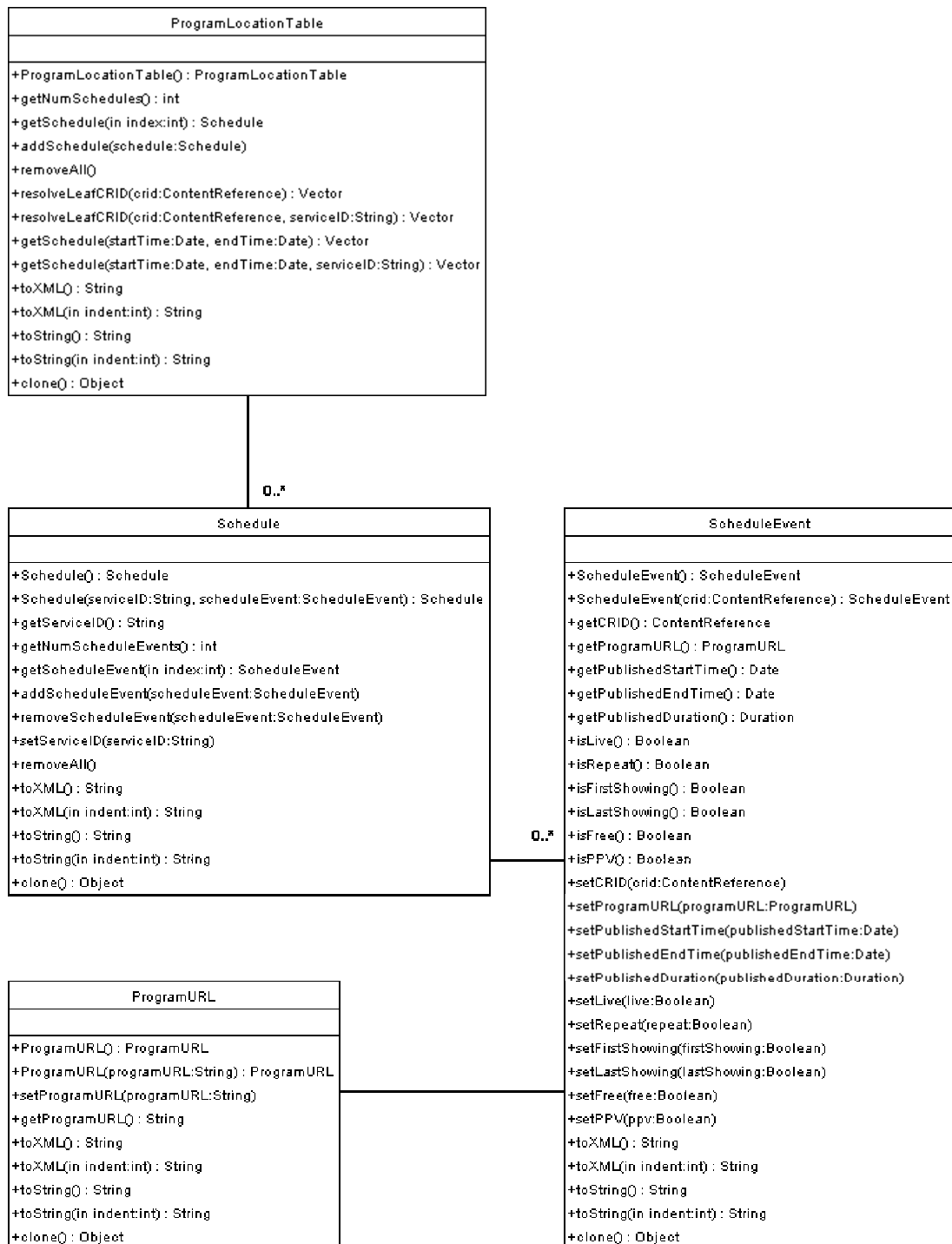


Figure 9 - bbc.rd.tvanytime.programLocation

4.8.1 ProgramLocationTable

ProgramLocationTable: This represents a table that contains Schedule objects. The table contains a number of Schedule objects (see section 4.8.3) and a `getSchedule(int index)` method is provided to retrieve them. Two other 'getSchedule' methods (defined in the `LocationResolution` Interface [section 4.9.1] which is implemented by this object) are provided and are used for retrieving a schedule (or list of `ContentReferences`) for programmes broadcast between a start and end date for on a particular service.

4.8.2 ProgramURL

ProgramURL: This represents a URL for a programme allowing the definition of the location and time of the programme.

4.8.3 Schedule

Schedule: This represents a series of schedule events that are associated with a single `serviceID`.

4.8.4 ScheduleEvent

ScheduleEvent: This can contain a `ProgramURL` object, published start and end times and duration, and information as to whether this showing is pay-per-view, the first showing, the last showing, live, a repeat, etc.

4.9 Searching package (bbc.rd.tvanytime.search)

This package defines interfaces for searching that are implemented by objects representing TV Anytime tables.

SearchInterfaceNotSupportedException
+SearchInterfaceNotSupportedException(msg:String) : SearchInterfaceNotSupportedException

<<interface>> MetadataSearch
+getProgramLocation(crid:ContentReference) : Vector +getProgramInformation(crid:ContentReference) : Vector +getSegmentInformation(crid:ContentReference) : Vector +getGroupInformation(crid:ContentReference) : Vector

<<interface>> LocationResolution

Figure 10 - bbc.rd.tvanytime.search

4.9.1 LocationResolution (interface)

LocationResolution: Interface for location resolution services.

4.9.2 MetadataSearch (interface)

MetadataSearch: Interface defining search functions of TV-Anytime metadata tables: *ProgramInformation*, *ProgramLocation*, *GroupInformation*, *SegmentationInformation*. Anything implementing this interface should search these tables for objects that match the given CRID. If any of the functions are not, or cannot, be supported then a *SearchInterfaceNotSupportedException* should be thrown.

4.9.3 SearchInterfaceNotSupportedException (exception)

SearchInterfaceNotSupportedException: This is an exception to be thrown by implementations of *MetadataSearch* and *LocationResolution* interfaces when they don't support certain search functions.

Segmentation refers to the ability to define, access and manipulate temporal intervals (*i.e.* segments) within an AV stream. The segment information package is used describe this information.



4.10.1 BasicSegmentDescription

BasicSegmentDescription: This contains descriptive information about the Segment, including Title, Synopsis, and any related material. Although this object is defined as being of type 'BasicSegmentDescription' (which matches the name in the TV Anytime specification documents), the XML representation of this object is named 'Description'.

4.10.2 SegmentGroupInformation

SegmentGroupInformation: This represents a group of segments.

4.10.3 SegmentGroupList

SegmentGroupList: This is a list of segment groups.

4.10.4 SegmentGroupType

SegmentGroupType: This denotes a collection of segments that are grouped together, for a particular purpose or due to a shared property. A segment group can contain segments, or other segment groups. Group types (including BOOKMARKS, HIGHLIGHTS, PREVIEW, SHOTS) are defined as static ints.

4.10.5 SegmentInformation

SegmentInformation: This represents information about a segment. A segment is a continuous fragment of a programme. A particular segment can belong to a single programme only, but it can be a member of multiple segment groups.

4.10.6 SegmentInformationTable

SegmentInformationTable: This represents a SegmentInformation table

4.10.7 SegmentList

SegmentList: This represents a list of segments

4.10.8 SegmentLocator

SegmentLocator: This locates a segment within a programme (instance) in terms of start time and duration (optional).

4.11 Service Information package (bbc.rd.tvanytime.serviceInformation)

This package contains classes supporting service information.

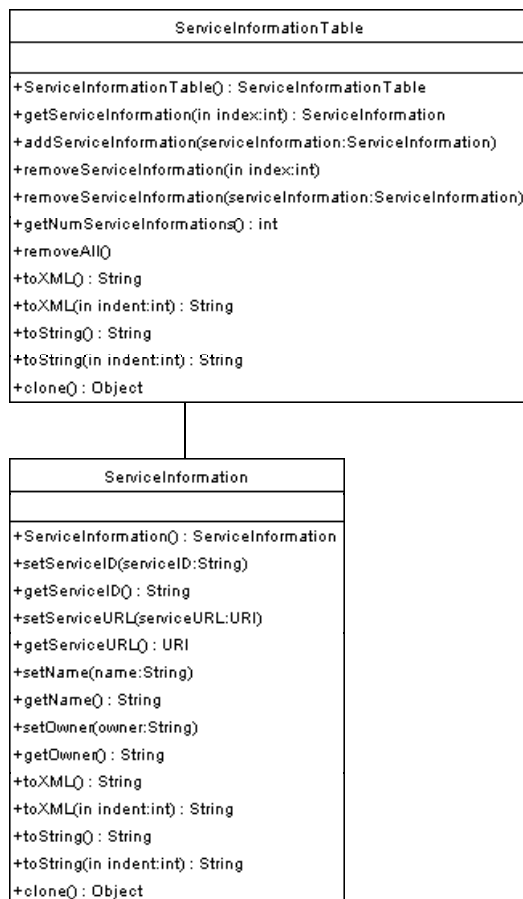


Figure 12 - `bbc.rd.tvanytime.serviceInformation`

4.11.1 ServiceInformation

ServiceInformation: This represents service information about a service, including the name and service ID.

4.11.2 ServiceInformationTable

ServiceInformationTable: This is a table of service information objects.

4.12 Utilities package (bbc.rd.tvanytime.util)

This package contains a number of utility classes providing useful tools for converting between data types used within the TV Anytime API.

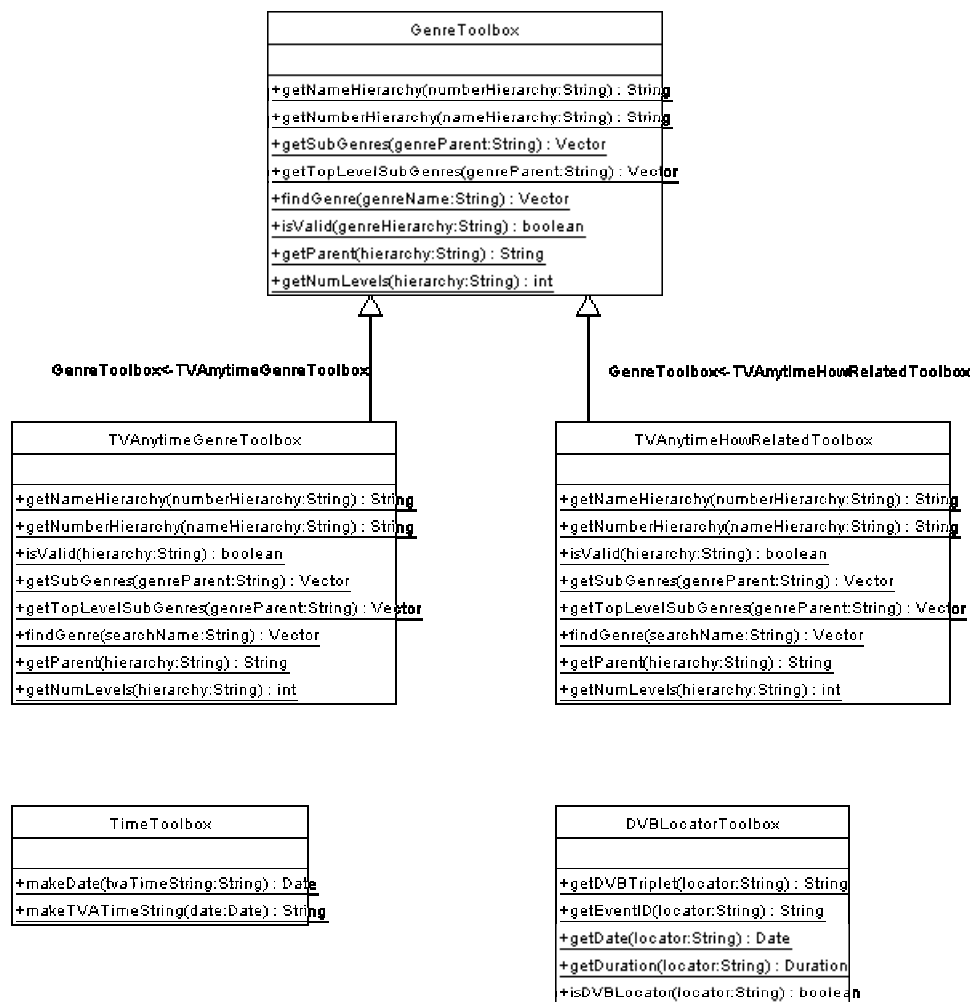


Figure 13 - `bbc.rd.tvanytime.util`

4.12.1 *DVBLocatorToolbox*

`DVBLocatorToolbox`: This contains tools for use with `Locator` objects for DVB. Includes methods for extracting the DVB triplet, the event ID, the date and the duration.

4.12.2 *GenreToolbox*

`GenreToolbox`: This contains tools for use with Genres. Includes searching the genre tree for a particular genre, converting genres between named and numbered forms and providing access to the entire genre tree.

4.12.3 *TimeToolbox*

`TimeToolbox`: This is a class offering useful methods on TV-Anytime time and date types. Includes conversion between dates formatted as TV-Anytime strings and Java Date objects.

4.12.4 TVAnytimeGenreToolbox

TVAnytimeGenreToolBox: This is an implementation of GenreToolbox specifically for TV-Anytime Genres.

4.12.5 TVAnytimeHowRelatedToolbox

TVAnytimeGenreToolBox: This is an implementation of GenreToolbox specifically for TV-Anytime HowRelatedCS.

Note: This class extends GenreToolBox, which has method names specific to 'genres'. HowRelated is not a genre as such, but the method names map across reasonably.

4.13 XML Parser package (bbc.rd.tvanytime.xml)

Used for getting data into the in-memory model from XML files and streams.

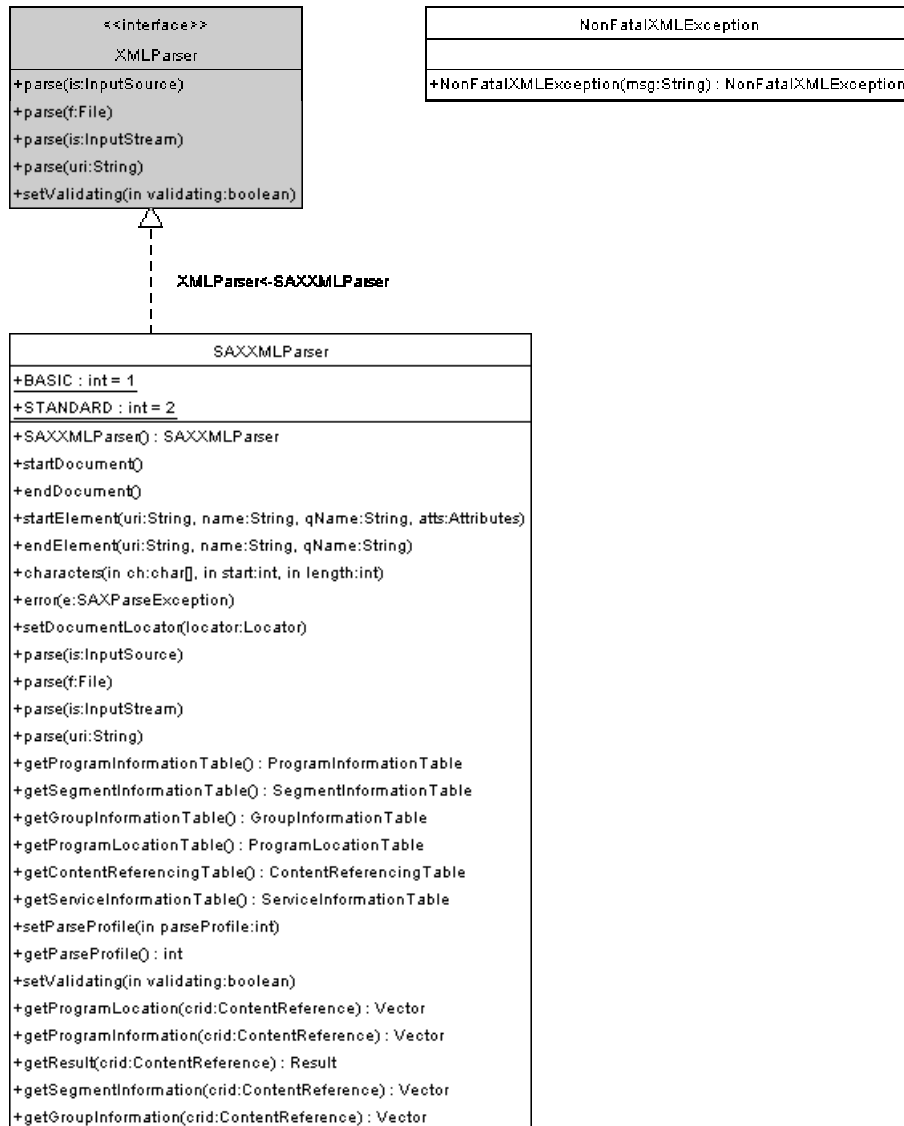


Figure 14 - bbc.rd.tvanytime.xml

4.13.1 NonFatalXMLException (exception)

NonFatalXMLException: This is an exception to be thrown by implementations of XMLParser when XML is valid but invalid TV-Anytime data values are found.

4.13.2 SAXXMLParser

SAXXMLParser: This is an implementation of the XMLParser Interface, using SAX. It uses a state machine to track the current position in document. All TV-Anytime objects that are parsed from files are added to one of the six tables (ContentReferencing, ProgramInformation, ProgramLocation, GroupInformation, SegmentInformation and ServiceInformation).

Error handling:

- Throws SAX XML exceptions
- Catches invalid TV-Anytime data fields, un-sets the variable and throws an error message only after parsing has finished.

Parsing profiles:

- Two profiles are defined for which sections of the XML are parsed.
 - BASIC: Only parses required/recommended elements and attributes.
 - STANDARD: Parses all of the currently used elements and attributes.

4.13.3 XMLParser (interface)

XMLParser: This is an interface for TV-Anytime XML parsers.

5 Code examples

5.1 Parsing TV-Anytime data using SAX

This code illustrates how to parse a TV-Anytime XML file using the API.

```
/**
 * Demonstrates parsing of XML program information table.
 *
 * @param filename Filename of XML file containing program information table.
 */
public void parseXML(String filename) {
    try {
        // Create parser
        SAXXMLParser parser = new SAXXMLParser();
        // Configure the parser to parse the standard profile (ie. everything).
        ((SAXXMLParser)parser).setParseProfile(SAXXMLParser.STANDARD);

        // Do the parsing...
        parser.parse(new File(filename));

        // Print out the contents of the parsed ProgramInformationTable...
        System.out.println(parser.getProgramInformationTable());
    }
    catch (NonFatalXMLException nfxe) {
        // Handle non-fatal XML exceptions
        // Contain any invalid TVAnytime data values from XML source.
        // These are all collated by the parser and thrown at the end to avoid
        // having to abort the parsing.
        nfxe.printStackTrace();
    }
    catch (TVAnytimeException tvae) {
        // Handle any other TVAnytime-specific exceptions that may be generated.
        // E.g. if the XML parser cannot be initialised.
        tvae.printStackTrace();
    }
    catch (IOException ioe) {
        // Handle IOExceptions: things like missing file
        ioe.printStackTrace();
    }
}
```

A fragment of the output looks like this:

```

ProgramInformationTable:
Version: 0
  ProgramInformation:
    CRID: crid://bbc.co.uk/1187011713
    Version: 1
    BasicDescription:
      Title: The Return Of Sherlock Holmes, type = main
      Synopsis: The Solitary Cyclist: A music teacher is being followed whenever she
leaves the house. A shy admirer? Or someone who means her ill?, length = short, language
= null
      Genre href = urn:mpeg:TVAnytime_v0.3IntentionCS:1.1, Genre type = main
        MPEG7Name: ENTERTAINMENT
      Genre href = urn:mpeg:TVAnytime_v0.3FormatCS:2.2, Genre type = main
        MPEG7Name: REPRESENTATION/PLAY
      Genre href = urn:mpeg:TVAnytime_v0.3ContentCS:3.4.1, Genre type = main
        MPEG7Name: General light drama
    AVAttributes:
      AudioAttributes:
        numOfChannels: 2

```

5.2 Generating TV-Anytime XML tables

As well as using TVA XML tables as input, the API can be used to generate TVA XML tables. The commented code below shows how this might be done for a ProgramInformationTable.

```

import bbc.rd.tvanytime.*;
import bbc.rd.tvanytime.programInformation.*;

/* Creating a TV-Anytime ProgramInformationTable using the BBC TVA API.
 * The program information table is populated by Title and Synopsis only.
 */
public class CreateProgInfoTable {
    /* declare the objects in our table */
    private ProgramInformationTable progInfoTable;
    private ProgramInformation progInfo;
    private ContentReference crid;
    private BasicDescription basicDesc;
    private Title title;
    private Synopsis synopsis;

    public CreateProgInfoTable() {
        /* initialise TVA objects to correct values */
        try {
            crid = new ContentReference("crid://bbc.co.uk/123456");
        } catch (TVAnytimeException tvae) {
            System.out.println(tvae);
        }
        title = new Title("The Life Of Mammals");
    }
}

```



```

        synopsis = new Synopsis("Man Eaters: Join Sir David Attenborough as
            he sits beside wild lions in the darkness of the night, meets
            a Siberian tiger face to face and finds out what you need to be a
            true carnivore.");
        basicDesc = new BasicDescription();
        progInfoTable = new ProgramInformationTable();
        progInfo = new ProgramInformation();

        /* add objects to the table */
        basicDesc.addTitle(title);
        basicDesc.addSynopsis(synopsis);
        progInfo.setProgramID(crid);
        progInfo.setBasicDescription(basicDesc);
        progInfoTable.addProgramInformation(progInfo);

        /* output the table as TVA XML*/
        System.out.println(progInfoTable.toXML());
    }

    public static void main (String args[]) {
        CreateProgInfoTable createProgInfo = new CreateProgInfoTable();
    }
}

```

Table 3: XML Generation, code example

The output is shown below:

```

<ProgramInformationTable version="0">
    <ProgramInformation programId="crid://bbc.co.uk/123456" version="0">
        <BasicDescription>
            <Title type="main"><![CDATA[The Life Of Mammals]]></Title>
            <Synopsis><![CDATA[Man Eaters: Join Sir David Attenborough as he
sits beside wild lions in the darkness of the night, meets a Siberian tiger face to face
and finds out what you need to be a true carnivore.]]></Synopsis>
        </BasicDescription>
    </ProgramInformation>
</ProgramInformationTable>

```

Table 4: XML Generation, output

Due to the nature of the API, it is possible to generate non-conformant XML documents, so care must be taken to ensure that all appropriate values are set. Each object has its own `toXML()` method. None of the objects are contained within `<TVAMain>`, so to produce a fully compliant TV-Anytime XML document, the table objects must be placed inside `<TVAMain>` manually.

5.3 Searching tables for schedules

This code shows how to get event information from a previously created program location table (e.g. from an XML file). The events are then sorted into time order.

```
/**
 * Creates a list of events sorted by time for a particular channel.
 *
 * @param programLocationTable ProgramLocationTable to create EPG from.
 */
public void createEPG(ProgramLocationTable programLocationTable) {
    // List of events for a particular channel
    List events = new ArrayList();

    // Search through all schedules in program location table
    for (int schedulect=0; schedulect<programLocationTable.getNumSchedules();
        schedulect++) {
        Schedule schedule = programLocationTable.getSchedule(schedulect);
        if (schedule.getServiceID().equals("BBCSeven")) {
            // Found schedule for a particular service, e.g. BBC 7
            for (int eventct=0; eventct<schedule.getNumEvents(); eventct++) {
                // Add each event to the list
                events.add(schedule.getEvent(eventct));
            }
        }
    }

    // Sort list of events
    Collections.sort(events, new Comparator() {
        // Returns a negative integer, zero, or a positive integer as the first
        // argument is less than, equal to, or greater than the second.
        public int compare(Object o1, Object o2) {
            return (int)((Event)o1).getPublishedTime().getTime() -
                ((Event)o2).getPublishedTime().getTime());
        }
    });

    // Print out the sorted list
    for (int ct=0; ct<events.size(); ct++) {
        System.out.println(events.get(ct));
    }
}
```

A fragment of the output looks like this:

```

Event:
    CRID: crid://bbc.co.uk/1187011715
    BroadcastURL: dvb://233a.1004.46c0;5c8e@2003-02-13T19:00:00Z/PT00H30M
    publishedTime: Thu Feb 13 19:00:00 GMT 2003
    publishedDuration: PT00H30M00S

Event:
    CRID: crid://bbc.co.uk/1187011716
    BroadcastURL: dvb://233a.1004.46c0;5c8f@2003-02-13T19:30:00Z/PT00H30M
    publishedTime: Thu Feb 13 19:30:00 GMT 2003
    publishedDuration: PT00H30M00S

```

5.4 Retrieving metadata and location information

This code demonstrates how to retrieve metadata and location information for a specific programme using the CRID.

```

/**
 * Searches for the program information data and resolves a given CRID.
 *
 * @param programInformationTable ProgramInformationTable to search.
 * @param contentReferencingTable ContentReferencingTable to search.
 */
public void searchForCRID(ProgramInformationTable programInformationTable,
    ContentReferencingTable contentReferencingTable) {
    Vector vector;
    ProgramInformation programInformation;
    ProgramURL programURL;

    try {
        // Create CRID to search for
        ContentReference crid = new ContentReference("crid://bbc.co.uk/1187011715");

        // First get it from the current program information table
        // Note that this is the getProgramInformation() method from the
        // MetadataSearch interface.
        System.out.println("This is the programme from the PI table...");
        vector = programInformationTable.getProgramInformation(crid);
        for (int ct=0; ct<vector.size(); ct++) {
            // For each program found for this CRID
            programInformation = (ProgramInformation)vector.elementAt(ct);
            System.out.println(programInformation);
        }

        // Then get it from the content referencing table
        System.out.println("Searching for CRID in CR table...");
        Result result = contentReferencingTable.resolveCRID(crid);
    }
}

```


6 References

- [1] S-3 TV-Anytime Metadata (Normative) Specification, v1.3, 15/12/2002
- [2] S-3 TV-Anytime Metadata Corrigenda 2 to S-3 v1.3, 21/03/2003
- [3] S-4 TV-Anytime Content Referencing (Normative) Specification, v1.2, 28/06/2002
- [4] S-4 TV-Anytime Content Referencing Corrigenda 1 to S-4 v1.2 16/08/2002

All available from the TV-Anytime website (see next section).

7 Bibliography

- TV-Anytime website (for specifications and further information)
<http://www.tv-anytime.org>

Appendix A

7.1 Document Conventions

The following conventions are used within this document:

Where Java class or method names are given within the body text, they are represented in the `courier` typeface.

Where code examples, XML tables and console output are given, they are represented

on a light grey shaded background, in the `courier` typeface

7.2 Diagrams

The reader may find that, when printed out, the UML diagrams of the API are too small to read comfortably. The diagrams are designed to indicate the structure of the API - if more detail is needed then we suggest that either the electronic version of this document is used or the related javadoc files are consulted.