

# Amarok Design Overview: Timecoding

Casey Link

May 1, 2009

## Abstract

The timecoding feature introduced in Amarok 2.1 allows for two types of position bookmarking: Manual and Automatic. This document serves as a basic developer's primer to the feature's framework. This document is necessary because the feature is spread across several internal Amarok components, including `Meta::Capabilities`, `AmarokUrls`, and the primary `ProgressSlider`.

The timecoding feature is a partial subset of Nikolaj's `AmarokUrls` infrastructure, but includes other necessary components. These include `Meta::Capabilities` for the creating/loading of timecodes, several `QWidgets` that provide visuals, and an `EngineObserver` that handles the creation/deletion of Automatic Timecodes.

## 1 Definitions

*Position Bookmark* – A record that represents a particular position in a track (e.g., 1 minute 5 seconds).

*Timecode* – The technical term for a Position Bookmark, to be used in code/discussions by developers and not presented to the user.<sup>1</sup>

*Manual Timecode* – A timecode created explicitly by the user.

Example: John is listening to a lengthy symphony piece and enjoys a particular solo section, so he creates a timecode at that position.

*Automatic Timecode* – A timecode created by Amarok without any user action.

Example: Mary is listening to a lengthy podcast and is interrupted such that she is forced to stop playback. Amarok timecodes her last listened position so she can easily resume when she returns.

---

<sup>1</sup>When the Position Bookmark feature was implemented there was already another feature that had utilized the term "Bookmark" for a different purpose, so a specific technical term was needed to differentiate the two.

## 2 Classes

### PlayUrlGenerator

- Generates the AmarokUrls that represent the timecode
- The technical definition of a timecode is

```
amarokurl://play/<Base 64 Encoded Meta::Track->playableUrl()/>/<integer seconds>
```

- Timecodes are stored in the AmarokUrl section of the database, which is transparently handled by the AmarokUrl system.

### PlayUrlRunner

- Decodes a timecode `AmarokUrl`
- Can start playback of timecoded track
- Can return a list of bookmarks for a particular `Meta::Track->playableUrl()` (if any exist)

### TimecodeWriteCapability

- A `Meta::Capability` for `Meta::Tracks` that provides an interface to allow tracks to have timecodes written to them.
- It supports the writing of two types of timecodes: Manual and Automatic
- Only one Auto timecode is allowed per track, so this class handles the deletion of old Auto timecodes if necessary

### TimecodeLoadCapability

- A `Meta::Capability` for `Meta::Tracks` that provides an interface to allow tracks to have timecodes read from them.
- It is a read only capability.

### TimecodeObvserver

- Handles the auto timecoding of tracks
- When the current track's played position reaches an arbitrary threshold and playback is stopped a timecode is created.
- The arbitrary threshold is hardcoded to be 10 minutes (600 seconds)
- It will never create an auto timecode if < 1 minute has played

### BookmarkPopup

- A `QWidget` that paints some nice graphical information about a timecode upon a `MouseOver` event.

### BookmarkTriangle

- A `QWidget` that represents the timecode marker visually in the Main slider

## **TimeSlider**

- Handles the drawing of **BookmarkTriangle** and **BookmarkPopup** widgets on screen

## **ProgressWidget**

- When a new track is played this class checks to see if the track has any timecodes, and if so draws the **BookmarkTriangle** on the **TimeSlider**
- Also contains convenience method **ProgressWidget::addBookmark()** if the user adds a timecode to the currently playing track

## **3 Design Notes**

- The technical definition of a timecode might seem clunky, but is the easiest way to record the concept of a track+position into a URL format.
- Timecode Urls should not be displayed to users – it is an internal item only.
- Write and Load capabilities are separated, because it is the case that some tracks are not writable yet have timecodes within them already.
- The timecode denomination needs to be standardized throughout the codebase.

## **4 Discussion Points**

- Should an option to enable/disable Automatic Timecodes be created?
- Should the arbitrary threshold before AutoTimecoding kicks in be user configurable?
- Should a track auto resume at it's Automatic Timecode? Should this be configurable?