jEmplode/RMML's Big Boy Parser

The parser in jEmplode/RMML is much nicer in that it has a proper tokenizer now, so it can support things like functions without catching on fire. It's also much more flexible about how it treats tag names vs string literals, so you can do things like "artist=source" and find all the tunes where the artist is the same value as the album (before, the rvalue always had to be a literal). A subtle resulting change is that if you don't quote your string literals and happen to use a tag name as intending it to be a string literal, it will be treated as the tag variable, not the string literal, so if you mean "source" as a string and not the source tag, put it in quotes.

Tag Names

Tag Name	Data Type	Description
artist	string	the name of the artist of the node
bitrate	string	the bitrate of the node (in [vf][ms][0-9]+ format where v = variable, f = fixed; m = mono, s = stereo; 0-9+ = bitrate in kps
codec	string	the codec of the node ("mp3", "vorbis", "flac", "wma", "taxi", "" for playlists)
colored	boolean	*Runtime Only* - whether or not the node is set to be colored in the UI
comment	string	the comment of the node
copy	boolean	whether or not this node is identified as a copy or an original
copyright	boolean	whether or not this node is identified as copyrighted
ctime	time	the time the node was created (in seconds since epoch)
decade	numeric	the decade of the year of the node, so if year = 1996, decade = 1990
dirty	boolean	whether or not the node has been modified since the last sync
drive	numeric	the drive number that houses the node
duration	numeric	the duration of the song in milliseconds
ext	string	the file extension of the file if it were a file (including the dot, so codec = vorbis returns ext = .ogg)
fid	numeric	the file identifier for the node
fid_generation	numeric	the generation of the node; changes every time the node's fid is recycled
genre	string	the genre of the node
iconType	numeric	*Runtime Only* - returns the type of icon that is used to represent the node
length	numeric	the length of the node in bytes
marked	boolean	whether or not the node has been marked
offset	numeric	the offset in btyes of the music content from the beginning of the file (for instance, this will skip ID3v2 tags)
options	numeric	a bit set of options (the meaning of the bits are different for each device)

pickn	numeric	the number of tracks of the playlist to pick at random
pickpercent	numeric	the percentage of tracks of the playlist to pick at random
pin	string	an assignable PIN number for the node for easy access
play_count	numeric	number of times the node has been played
play_last	time	the date the node was last played (in seconds from epoch)
playlist	binary	an encoded value containing the list of FID numbers of the children of the playlist
pos	numeric	*Runtime Only* - returns the position of this node in the context of whatever playlist you are currently viewing (or 1 if there is no playlist context)
profile	binary	an encoded value containing information about the track for use on the device while playing
refs	numeric	the number of non-soup playlists that contain this node
rid	string	a 32 character hash generated based on the content of the music of the node that should be unique for every tune
samplerate	numeric	the samplerate of the node (in samples per second)
size	numeric	the number of entries in the playlist
skip_count	numeric	the number of times the node has been skipped
soup	string	the encoded format that jEmplode uses to represent the query that defines the soup playlist
source	string	the name of the source of the node (for instance, the album name, "Live", or equivalent)
stereo	boolean	whether or not the tune is stereo
title	string	the title of the node
tracknr	numeric	the track number of the node
tracknrorpos	numeric	if the node does not have a track number specified, returns the position of the tune in its playlist
tracks	numeric	the number of tracks that are contained in the playlist and all of its nested playlists
trailer	numeric	the offset in bytes from the end of the file where the music content ends (for instance, this will skip ID3v1 tags)
type	string	the type of the node ("tune", "playlist", "taxi", or "illegal")
wendy	string	a comma separated list of wendy flag names that the tune is flagged with
year	numeric	the year of the node

Functions

Function Signature	Parameter	Return Type	Description
date(x)	A date string in your local format (i.e. "MM/DD/YYYY" or "DD/MM/YYYY")	time	Returns the specified date in number of seconds from epoch
daysago(x)	Number of days	time	Returns the time x days ago in number of seconds

			from epoch
fibonacci(x)	Iteration number	numeric	Returns the value of the fibonacci sequence at the given iteration
hasoption(x)	The index of the bit in the option set	boolean	Returns whether or not the specified bit is turned on in the option set
now()	n/a	time	Returns the current date in number of seconds from epoch
secondsago(x)	Number of seconds	time	Returns the time x seconds ago in number of seconds from epoch

Operators

x and y, x && y, x & y	Logical and of x and y.
x or y, x || y, x | y	Logical or of x and y.
not x, !x	Logical negation of x.
x = y, x == y, x equals y, x is y	Compares the values of x and y. If x and/or y are tag names, the value of the x tag for the node will be replaced for the variable. For instance, "title = Super" would match tunes with a title of "Super", or "title = artist" would match all the nodes where the title is the same as the artist.
x != y, x <> y	"Not Equals" comparison (inverse results of =)
x < y, x > y, x <= y, x >= y	Less than, Greater than, Less than or equal, Greater than or equal comparison of x and y.
x contains y, x like y	Substring comparison. Returns true if the value of x contains the value of y.
x + y	Returns the sum of x and y.
x - y	Returns the difference of x and y.
-x	Returns the negation of x.
x * y	Returns the multiplication of x and y.
x / y	Returns the division of x by y (integer).
x	If the value of x is not empty, "0", "false", "no", of "off" this will return true, otherwise it will return false.
(...)	Order of operations grouping (can be nested)

Examples
title like "alking" and (artist = "James Taylor" or artist like "James")
Finds all tunes that contain "alking" in their titles (i.e. "Walking Man") and and whose artist is either "James Taylor" or anyone who has the word "James" in the author field.

dirty = "true"
Finds all tunes or playlists that are marked dirty.

refs > 3 and not (tracknr = 2)
Finds all tunes that have a reference count greater than 3 and whose track number

does not equal 2.

refs != 3 && (refs <= 1 || genre = "Rock")
Finds all tunes whose reference count does not equal 3 and whose reference count is either less than or equal to 1 or whose genre is equal to "Rock".

wendy contains "Kirsten" or wendy contains "Mike"
Finds all the tunes that have the Kirsten or Mike wendy flag set.

ctime < date("07/10/96")
Finds all the tunes that have a creation time that is before July 10th, 1996 (or October 7th, 1996 if your locale interprets dates wrong ;))

artist = source
Finds all the tunes that have an artist that is the same as the album.

refs
Finds all the tunes that have a reference.

!refs
Finds all the tunes that do not have a reference.

play_last > daysago(5)
Finds all the tunes that have been played in the last 5 days.

*play_count < fibonacci(10) * 3*
Finds all the tunes that have a playcount that is less than three times the 10th iteration of the Fibonacci sequence :)

These are just samples -- You can mix and match any of these operators to construct any number of queries.