

Appendix B: Media Server Queries

\$Date: 2007-11-23 14:35:44 -0500 (Fri, 23 Nov 2007) \$ for MediaServer V03

Full list of query types (these go in the type field)	
<i>Query Type</i>	<i>Description</i>
themes	get a list of themes. See example query-themes.xml
themesonglist	get a list of songs in a theme. Requires an idTheme parameter to specify which theme. See example query-themesonglist.xml
themeselectionlist	get a list of selections in a theme. Requires an idTheme parameter to specify which theme. See example query-themeselectionlist.xml
whatsnext	get a list of upcoming songs on a given stream. See example query-whatsnext.xml
longestsinceplayed	get a list of songs, ordered by when last played. See example query-longestsinceplayed.xml
recentadditions	get a list of songs, ordered by when added to the system. See example query-recentadditions.xml
recentlyplayed	get a list of songs, ordered by when last played. See example query-recentlyplayed.xml
searchgenres	search for genres whose name matches given search string. See example query-searchgenres.xml
searchartists	search for artists whose name matches given search string. See examples query-searchartists.xml , query-searchartists-anywhere.xml , query-searchartists-invert.xml
searchalbums	search for albums whose name matches given search string. See example query-searchalbums.xml
searchinternetradio	search for Internet radio whose name matches given search string. See example query-searchinternetradio.xml

Full list of query types (these go in the type field)	
searchtitles	search for songs whose name matches given search string. See examples query-searchtitles.xml , query-searchtitles-positive-modulo.xml , query-searchtitles-negative-modulo.xml
searchall	search for songs whose name,artist,album or genre matches given search string. See example query-searchall.xml
searchyears	search for songs whose year matches given search string. See example query-searchyears.xml

Options common to all queries ¹	
<i>Option Name</i>	<i>Description</i>
quantity²	Maximum number of entries to return
offset²	offset says to skip that many entries before beginning to return entries. So an offset of 5 means return entries starting with the 6th. An offset of 0 is the same as omitting the offset option. If both offset and quantity appear, then offset entries are skipped before starting to count the quantity entries that are returned.
modulo	modulo is used to skip entries in the results, and can also be used to reverse the order the items come back (by using a negative modulo). So a modulo of 2 means to return every other entry. A modulo of -1 will return items starting with the item specified by offset, and go backward from there toward 0. A modulo of -2 will go backward, skipping every other entry. A modulo of positive 1 is the same as omitting the modulo option. This option is most useful for fast scrolling through large lists on small devices.
anymoneyoulike³	To allow embedding of session IDs or anything else the caller wants to use to identify queries, all XML fields are echoed back in the query result. (see the anymoneyoulike field in the sample from the XML Protocol

Options common to all queries ¹	
	Overview).
requesttotal	Boolean, when set causes a field named <TotalEntries> to be returned in addition to the query results. This field indicates the total number of entries this query would return if offset and quantity were not specified. This is useful for “Displaying 1-5 of 10” status lines. See example query-searchall.xml .
Only valid in the queries which start with 'search'	
letters	restrict selections to items which start with this text. This is obsolete, replaced by the category specific filters below. See example query-searchartists-invert.xml
genrefilter	restrict selections to items whose genre names start with this text. If matchanywhere is set, then the string matches anywhere in the name. See example query-searchartists-filtergenre.xml
artistfilter	restrict selections to items whose artist names start with this text. If matchanywhere is set, then the text matches anywhere in the name.
albumfilter	restrict selections to items whose album names start with this text. If matchanywhere is set, then the text matches anywhere in the name.
songfilter	restrict selections to items whose song names start with this text. If matchanywhere is set, then the text matches anywhere in the name.
yearfilter	restrict selections to items whose years start with this text. If matchanywhere is set, then the text matches anywhere in the year.
Invert	Flag, if set, then results will be all items which DON'T match search criteria. See example query-searchartists-invert.xml
MatchAnywhere	Flag, if set, allow letters to match anywhere in string instead of anchoring to the beginning. See example query-searchartists-anywhere.xml
idTheme⁴	restrict selections to items which are in the theme idTheme

Options common to all queries ¹	
idGenre ⁵	restrict selections to items which are in the genre idGenre
idArtist ⁵	restrict selections to items which are by the artist idArtist
idAlbum ⁵	restrict selections to items which are in the album idAlbum

Note 1 - a blank search (no letters or id fields) returns all results (so the searches can be used to get a complete list)

Note 2 - **quantity** and **offset** are used to control which entries in the query result list are returned. If a query has 20 results, a quantity of 10 will get the first 10, quantity of 10 with an offset of 5 will return entries 5-15.

Note 3 – The name **anynameyoulike** is just an example, you can name the field any string you wish, as long as it conforms to the XML naming rules, and doesn't conflict with any of our field names. Also note you can have as many custom tags as you want. To prevent conflicts with future fields we suggest prepending a company or product name to your custom tags.

Note 4 - this id is an index into the theme table, this index can be obtained from a themes query. Only valid for searchgenres, searchartists, searchalbums, and searchsongs.

Note 5 - these ids are indexes into the genre/artist/albums, these indexes are given in the results of queries, so they are usually used to restrict an existing query. For example, a list of genres might be shown to the user, then they select one, then a list of artists in that genre would be displayed.