

eyeQ Bundle Implementation Guide

Product Version 1.5.0

Published: 30-Jul-2013 15:44

Gracenote, Inc.
2000 Powell Street, Suite 1500
Emeryville, California
94608-1804
www.gracenote.com

Table of Contents

About the Gracenote eyeQ™ Bundle	4
Service Delivery Platform versus Client-Based Systems	4
Queries and Responses	4
Supported HTTPS Protocols	4
Registration and Authentication	5
Registering a Device	6
Setting the Language Preference	7
Application and SDK Info in Queries	7
Status Reporting	8
General Gracenote Web API Best Practices	9
eyeQ Bundle Query Types and Data Elements	10
Query Types	10
Data Elements	11
Extended Data	12
Associated (Linked) Elements	12
Associated Images	13
Ordering, Numbering, and Ranking	14
Best Practices for TV Program Ranking	15
Best Practices for TV Channel Ranking	15
Using the LANG Attribute in Responses	15
Best Practices for Retrieving and Managing EPG Downloads	16
Managing TV Channel Logos for Different Transmission Types (SD, HD, 3D)	16
Using TVGRID_LOOKUP (What Is On Now)	17
Using TVGRID_LOOKUP with MODE CONTRIBUTOR, AV_WORK, SERIES, or SEASON	18
Using TVGRID_SEARCH	18
Using FIELDVALUES for IPG Category Values	
Best Practices for Image Dimensions	19
Image Resizing Guidelines	19
Image Formats and Dimensions	20
Available Image Dimensions	20
Legacy Image Dimensions	21
Common Media Image Dimensions	21
Music Cover Art	21
Video Cover Art	
Artist Imagery	
Variations in Video Image Dimensions	
AV Work Images	22
Video Product Images	22
Contributor Images	22

About the Gracenote eyeQ™ Bundle

The Gracenote eyeQ Bundle includes the eyeQ, VideoID, and VideoExplore Web APIs. These Web APIs are based on an XML-based HTTP protocol that provides simple, yet highly-functional access Gracenote Media Recognition Service. Using the protocol, client applications send queries and receive responses as XML through HTTP.

Gracenote eyeQ supports Video identification and exploration, and TV IPG (interactive programming guide) functionality. You can use these interfaces to get detailed Gracenote metadata about audio-visual works and TV programs.

Service Delivery Platform versus Client-Based Systems

The Gracenote eyeQ Bundle supports Service Delivery Platform (SDP) systems and client-based (non-SDP) systems. SDP systems consist of one or more customer-hosted distributed servers that process queries and deliver responses to large network of end user devices. Non-SDP systems consist of client applications running on set top, mobile, or other end user devices that interact directly with Gracenote Media Recognition Services. Most of the eyeQ Web API is designed for client-based systems, however special APIs are also available to support the unique requirements of SDP systems. For more information about Service Delivery Platforms, please contact your Gracenote Professional Services representative.

Queries and Responses

The Gracenote Web API is an XML-based protocol: both query and response messages carry payloads in the form of XML documents. The root element for all queries is <QUERIES> containing one <QUERY> element. All responses have a root element of type <RESPONSES> containing one <RESPONSE> element.

Example Queries Element

```
<QUERIES>
<QUERY>
...
</QUERY>
</QUERES>
```

Example Responses Element

```
<RESPONSES>
  <RESPONSE>
    ...
    </RESPONSE>
  </RESPONSE>
</RESPONSES>
```

Supported HTTPS Protocols

Client applications request information by issuing XML queries to the Gracenote Media Recognition Service and receive back XML responses. The Gracenote Media Recognition Service uses HTTPS for secure communication. All transmissions use UTF-8 encoding and the HTTPS Secure Socket Layer (SSL) cryptographic protocol.

Web API client applications should submit all Web API queries using an HTTP POST request. After establishing a secure connection, client applications can retrieve metadata (images, biographies, reviews, and others) using the HTTP GET.



A Important

Contact Gracenote Professional Services for the Gracenote Media Recognition Service URL to use for your client application.

HTTP Compression

To save on bandwidth, standard HTTP compression will be supported on the servers. If the client chooses to use compression, gzip is the recommended compression algorithm. The client can invoke this through the standard method of enabling HTTP compression (specifying the "Accept-Encoding: gzip" line in the request's header).

Posting Queries and Responses

You can send XML queries directly to the Gracenote Media Recognition Service from your browser. This is useful to test the service and learn how to use the Gracenote Web APIs. Two popular browser plug-ins for Firefox are Poster (https://addons.mozilla.org/en-US/firefox/addon/poster/) and RESTclient (https://addons.mozilla.org/en-US/firefox/addon/restclient/). To use these plug-ins, enter the Gracenote Media Recognition Service URL and an XML query. You must include your Gracenote client ID string and User ID string in the AUTH block of each query. See 1.5.0 (see page 5).

Registration and Authentication

Gracenote Media Recognition Service authenticates *all queries* using a combination of two customer-specific identifiers: a Client ID string and a User ID string.

The Client ID string has the following format: 123456-789123456789012312. The first part is a six-digit Client ID, and the second part is a 17-digit Client ID Tag. Gracenote Professional Services provides the Client ID string as part of your licensing agreement.

Each device that sends queries to the Gracenote Media Recognition Service also needs a unique ID. This ID is called a *User ID*, and is generated by the Gracenote Media Recognition Service in response to a REGISTER query. See #Registering a Device (see page 6).



Important

Every query a client application sends to Gracenote must include both of Client ID string and User ID string inside an authorization <AUTH> block.

Example Query with authorization

```
<QUERIES>
    <AUTH>
        <CLIENT>client_id_string</CLIENT>
        <USER>user_id_string</USER>
    </AUTH>
    <OUERY>
    </QUERY>
</OUERIES>
```

Registering a Device

To obtain a User ID for a device, the client application must submit a REGISTER query. This query requires a valid Client ID. The client application needs to register a device only once. If registration succeeds, Gracenote returns a successful registration response that contains the User ID. Your application should store the User ID in a non-volatile location on the device itself, and use it for authentication in all subsequent queries.



Important

Store the User ID in a location accessible only by your application. Users and third-party applications must be blocked from accessing or viewing the User ID unless it is encrypted or obscured in some way.

All subsequent non-registration queries must include an <AUTH> block containing both the Client ID string and the User ID string.

Example of REGISTER query

```
<QUERIES>
   <QUERY CMD="REGISTER">
       <CLIENT>client_id_string</CLIENT>
   </QUERY>
</QUERIES>
```

Example of a Successful Registration response

```
<RESPONSES>
   <RESPONSE STATUS="OK">
       <USER>user_id_string</USER>
   </RESPONSE>
<RESPONSES>
```

If user registration fails, the <RESPONSE> element will be empty and will carry the STATUS attribute ERROR. A separate <MESSAGE> element is included along with the response, containing a descriptive error message. For example, the following is a response for a registration query that was missing a valid client ID-Client ID Tag pair.

Example Registration error response

```
<RESPONSES>
    <MESSAGE>Missing or bad CLIENT information.</MESSAGE>
    <RESPONSE STATUS="ERROR">
     </RESPONSE>
    </RESPONSE>
</RESPONSES>
```

Setting the Language Preference

A query can include an optional <LANG> element to specify a preferred natural language in which to receive response metadata. The body of the <LANG> element should be a three-character language code as defined by the ISO 639-2 standard.

When possible, the Gracenote Media Recognition Service returns localized metadata in the language requested. If no metadata is available in the requested language, or if there is no language preference, metadata is returned in English (eng) by default.

Contact your Gracenote Professional Services representative for a list of supported languages.

Example of Language Preference

Application and SDK Info in Queries

<APP_INFO> and <SDK_INFO> are optional query elements, however Gracenote strongly encourages you to use them. When used, they should contain information sent by both the application and the SDK used (if any), respectively. Information in these two fields must be presented as a comma-separated list of key=value pairs and the value should be double-quoted. Suggested keys are:

```
app Application name and version
```

sdk	SDK name and version
mfg	Manufacturer of device
os	Platform and operating system

Applications and SDKs should populate as many of these fields as possible, and are free to add additional fields, to aid debugging and information gathering.

Status Reporting

Each <RESPONSE> element carries a STATUS attribute indicating whether the corresponding query succeeded or failed. If a query fails because of an error of some kind, the <RESPONSE> element will have a STATUS attribute of ERROR and will be accompanied by a separate <MESSAGE> element describing the error.

Response Status	Description		
OK	Query succeeded and produced at least one match		
NO_MATCH	Query succeeded but did not produce a match		
ERROR	Query failed		

Example of Success Response

If the value of the STATUS attribute is OK, the body of the <RESPONSE> element contains the data returned in response to the query.

Example of No Match Response

If STATUS is NO_MATCH, the element body is empty.

```
<RESPONSES>
    <RESPONSE STATUS="NO_MATCH">
     </RESPONSE>
    </RESPONSES>
```

Example of Failure response

The following example shows the response to a VIDEODISCSET_TOC query that was missing the TOC data to look up. The application should display the message string to the user, to allow for diagnosis of problems in the field; this information is required for Gracenote support representatives to help in solving both developer and end-user issues.

General Gracenote Web API Best Practices

The following are best practices for designing client applications with the Gracenote Web API.

Retries

When an error occurs, your client application should retry once. If it still gets an error, give up and notify
the user. The client application must not keep retrying. Doing so can potentially damage the Gracenote
Media Recognition Service.

Error Strings

- You should ensure that error strings from the Web API are easy to retrieve. Preferably, you should
 display this string to the end user, or at least optionally display it. Error strings are contained in the
 <MESSAGE> element of a response.
- Do not try to interpret the error string, rely on it being in a certain format, or try to re-format or otherwise modify it. Error strings can and will change without notice.

Handling XML Responses

- When dealing with the XML responses, understand that new elements and new attributes will be added to extend the protocol.
- The client application should ignore any XML elements and attributes that are not relevant.

Missing Elements

 Design your client code to handle missing elements gracefully. Even elements that always should be there (such as GN_ID) may be missing due to a bug. Should this happen, the client application should not crash or seriously malfunction.

Ordered Elements

- When dealing with ordered elements (RANK or ORD attributes), the client application must be tolerant of gaps in the enumeration, as well as objects of unknown type that appear in the list.
- The client application should ignore objects of unknown type and skip to the one that has the next higher ordinal or rank and is of a known type.

Working with GN_IDs

• Your client application should treat all Gracenote identifiers (GN_IDs) as opaque strings. Do not try to interpret or parse them. Caching them locally for client application use is permitted.

Queuing Requests

For most standard device configurations, Gracenote has typically recommended a maximum of 5
concurrent threads per core simultaneously querying via the Gracenote Web Services API. This assumes
that the application properly handles conditions such as timeouts, retries, and error handling. Please
contact your Gracenote Professional Services representative for additional guidance on optimizing
performance.

Handling Timeouts

• In the event of a timeout when querying the Gracenote Web Services API, Gracenote recommends one immediate retry. This applies to all non-registration Web Services API queries.

eyeQ Bundle Query Types and Data Elements

Each query has a query type denoting the type of operation requested. You specify the query type using the CMD attribute attached to the <QUERY> element.

Query names follow a standard pattern:

<target_object>_FETCH</target_object>	A Fetch query uses a Gracenote ID (GN_ID) to locate and return an item.
<target_object>_SEARCH</target_object>	A Search query uses a text string to search for and return an item.
<target_object>_LOOKUP</target_object>	A Lookup query uses a Table of Contents (TOC) or fingerprint (FP) to locate and return the item.

Where <Target_Object> is the object being queried.

Query Types

Gracenote eyeQ Bundle Web APIs support the following guery types.

Query Type	Description
AV_WORK_FETCH	Get an AV_WORK by Gracenote identifier (GN_ID)
AV_WORK_SEARCH	Get an audio-visual work by text search
CONTRIBUTOR_FETCH	Get a contributor by GN_ID
CONTRIBUTOR_SEARCH	Get a contributor by text search.
REGISTER	Register user identifier (User_ID).
SEASON_FETCH	Get a season by GN_ID.
SERIES_FETCH	Get a series by GN_ID.

SERIES_SEARCH	Get a series by text search
TVCHANNEL_FETCH	Retrieves a TV Channel based on a TV Channel GN_ID.
TVCHANNEL_LOOKUP	Looks up TV Channels provided by a TV Provider based on a North American locale (country + postal code), DVB (Digital Video Broadcasting) triplets, or ISDB (Integrated Services Digital Broadcasting) triplets.
TVGRID_LOOKUP	Retrieves a dynamically-computed TV grid.
TVPROGRAM_FETCH	Retrieves the full metadata for a TV Program.
TVPROGRAM_SEARCH	Searches TV Programs on the specified channels. Only TITLE search is supported.
TVPROVIDER_LOOKUP	Looks up TV Providers for a given locale (country and postal code)
VIDEODISCSET_FETCH	Get a video disc set based on a GN_ID.
VIDEODISCSET_TOC	Get a video disc set by table of contents (TOC) lookup

Data Elements

Gracenote eyeQ Bundle Web APIs support the following data elements:

Data Element	Description
Audio-Visual Work	A distinct artistic or cerebral creation created by one or more contributors, and expressed through text, audio, video, images, or any combination of these. The term work is conceptual and used to encompass related products. For example, the director Ridley Scott's body of work includes the video products Alien, Blade Runner, Thelma and Louise, Gladiator, and Hannibal. Audio-Visual Works can be grouped by Franchise (for example, all CSI TV shows), Series (all seasons of CSI: Miami), and Season (all episodes of Season 2 of CSI: Miami). In the Gracenote eyeQ Web API, Audio-Visual Works are designated by the XML tag <av_work>. In this document, the terms Audio-Visual Work, Work, and AV Work are used interchangeably.</av_work>
Channel	A TV station and its TV programs.
Contributor	A person, group, or other production entity participating in the creation of a work: for example, Christian Bale and Heath Ledger are contributors of The Dark Knight (as actors), as are Hans Zimmer (score) and Legendary Pictures/Warner Brothers (production).
Episode	A specific instance of a TV Program in a TV Series.
Product	A product is a collection of one or more Works; for video, this is a DVD, a Blu-ray, or some type of digital video. A product is generally either a main feature (the movie) and extras, or a set of episodes. Products have a commercial ID, such as a UPC or Hinban code.
Season	An ordered collection of Works, typically representing a season of a TV series. For example: CSI: Miami (Season One), CSI: Miami (Season Two), CSI: Miami (Season Three)
Series	A collection of related Works, typically in sequence, and often comprised of Seasons (generally for TV series), for example: CSI: Miami, CSI: Vegas, CSI: Crime Scene Investigation.
TV Program	A specific AV_Work that airs on a TV channel.
TV Provider	The company or service that provides TV content.
Work	See Audio-Visual Work.

The following example shows a contributor element for the actor Christian Bale. Each element has a Gracenote identifier (<GN_ID>) uniquely designating that specific element, along with additional child elements containing

information (metadata) about the work, contributor, product, or other entity the element represents. For brevity, the example shows only the <NAME> and <BIOGRAPHY> metadata elements; an actual contributor element would contain many more.

Example of Contributor Element

```
<CONTRIBUTOR>
  <GN_ID>a_gracenote_id</GN_ID>
  <NAME>Christian Bale</NAME>
  <BIOGRAPHY>
    Bale first caught the public eye when he was cast in the starring role of Steven Spielberg's "Empire of the Sun" at the age of 13, playing an English boy who is separated from his parents and subsequently finds himself lost in a Japanese internment camp during World War II. Since then, he has portrayed a wide range of characters.
    </BIOGRAPHY>
  </CONTRIBUTOR>
```

Extended Data

Depending on their specific needs, client applications may wish to restrict downloading certain types of metadata to conserve server resources, minimize bandwidth requirements, and and speed up response times. This data is called extended data. Examples include entity associations, image URLs, commercial links, and full contributor mediagraphies. Extended data is not included in query responses unless explicitly requested.

The specific types extended data available vary depending on the query type. To request one or more of these extended data types, the query must include an <OPTION> element with the parameter value SELECT_EXTENDED. The following example shows an example for a VIDEODISCSET_FETCH query requesting cover art images and associated audio-visual works.



🛕 Important

Requesting extended data results in additional server computation and slower responses. The client application should limit such requests to those data values actually needed for a particular use case.

Example of SELECT_EXTENDED query option

```
<QUERY CMD="VIDEODISCSET_FETCH">
  <GN_ID>a_gracenote_id</GN_ID>
  <OPTION>
  <PARAMETER>SELECT_EXTENDED</PARAMETER>
  <VALUE>COVERART, AV_WORK</VALUE>
  </OPTION>
  </QUERY>
```

Associated (Linked) Elements

The Gracenote eyeQ Web API supports associations among works, contributors, products, seasons, and series through nested XML elements referring to the associated entities by their unique Gracenote identifiers.

A client application could use the Gracenote identifiers given in these elements (for example, 23456 for The Dark Knight) to retrieve and display additional metadata for the associated works. The work elements may in turn contain nested elements identifying specific products (such as video disc sets) associated with the work, the client application could then use these elements to retrieve and present commercial information about these products.

The following example shows a contributor element for Christian Bale containing nested elements identifying audio-visual works (movies) to which he has contributed.

Example of Associated Work elements

Example of Associated product elements

Associated Images

The metadata associated with an entity may include graphical images of various kinds, such as product cover art and images representing works, series, and contributors. Where available, such images are represented by XML elements containing universal resource locators (URLs) from which to retrieve the image data.



Image URLs contained in Gracenote eyeQ Web API responses may become invalid shortly after receipt. They are strictly for immediate use. Do not cache them for later use.

Example of Image URL element

```
<CONTRIBUTOR>
  <GN_ID>a_gracenote_id</GN_ID>
  <NAME>Christian Bale</NAME>
  ...
  <URL TYPE="IMAGE" SIZE="LARGE">a_temporary_url</URL>
  </CONTRIBUTOR>
```

Ordering, Numbering, and Ranking

A standard method of ordering, numbering, and ranking elements is used throughout the Web API protocol. The ORD attribute orders elements into a single list: for example, responses containing multiple matches are ordered. The ordering is expressed as an attribute of the XML elements and not within the body of the elements themselves because the ordering is not intrinsic to the metadata. It is meaningful only in the context of the enclosing list.

Example of Ordered elements

```
<CONTRIBUTOR ORD="1">
    ...
</CONTRIBUTOR>
<CONTRIBUTOR>
</CONTRIBUTOR>
<CONTRIBUTOR ORD="3">
    ...
</CONTRIBUTOR ORD="3">
    ...
</CONTRIBUTOR></CONTRIBUTOR></CONTRIBUTOR></CONTRIBUTOR>
```

By contrast, the following example shows the feature list for a DVD. Here the feature numbering within the disc is part of the metadata itself and is meaningful even if it were removed from the surrounding context. The feature number is included as an actual child element within the <FEATURE> element rather than as an external attribute.

Example of Numbered elements

```
<FEATURE>
    <FEATURE_NUM>1</feature_NUM>
    <TITLE>The Dark Knight</TITLE>

</FEATURE>
</FEATURE>

<FEATURE>
    <FEATURE_NUM>2</FEATURE_NUM>
    <TITLE>The Making of...</TITLE>
</FEATURE>
</FEATURE></PEATURE></PEATURE></PEATURE></PEATURE></PEATURE></PEATURE></PEATURE></PEATURE></PEATURE></PEATURE>
```

ORD and FEATURE_NUM values are unique. That is, no two elements in the list can have the same numerical value. There may be gaps in the numbering, so the client application should handle this possibility.

The RANK attribute is used to rank elements by significance. RANK values are not necessarily unique. For example, a contributor can be primarily an actor (RANK=1) and secondarily a director or producer (both RANK=2).

Example of Ranked elements

Best Practices for TV Program Ranking

The Gracenote eyeQ API exposes a rank element for TV Programs that allows client application to display or highlight popular programs to their end user. This value is noted in the results of a TVPROGRAM_FETCH and TVGRID_LOOKUP call at the PROGRAM level as RANK. The RANK value will range from 0 to 1 billion, with a score of 1 billion indicating the highest ranked item.

Because the RANK value is at the TVPROGRAM level, all airings of a specific program will inherit the same ranking.

When generating highlighted program lists, the client device must choose the airing of a program in the user's preferred time slot.

Best Practices for TV Channel Ranking

The Gracenote eyeQ Web API exposes a rank element for TV Channels that allows a client application to display or highlight popular channels to end users. This value is returned as RANK in the results of TVCHANNEL_LOOKUP and TVCHANNEL_FETCH at the TVCHANNEL level. The RANK value ranges from 0 to 1 billion, with a score of 1 billion indicating the highest ranked item.

Multiple channels may share the same RANK value, and the RANK value for a channel may change. Also, the RANK value is global across all countries/territories. For example, a top-ranked French channel may be available in Germany (where it may not be considered a top-ranked channel). Your application should handle these scenarios. To determine the top-ranked channels, sort the channels by the RANK value. Your application can perform additional sorting and filtering to generate a list of the top-ranked channels appropriate for your end users.

Using the LANG Attribute in Responses

Some responses include the LANG attribute for certain fields. The LANG attribute indicates the language of the field. LANG is returned only when the language is not the same as the default language of the query. For

example, if the default query language is German, and the returned language for the field is German, the response would not include the LANG attribute. However, if the language for the field is English, the response would include the LANG attribute with a value of ENG

LANG is currently supported for these fields:

- SYNOPSIS and TITLE for AV Works
- BIOGRAPHY for Contributors

Best Practices for Retrieving and Managing EPG Downloads

The Gracenote EyeQ API exposes several APIs for retrieving and working with EPG data (TV program listings). This topic presents an overview of these methods to help you choose the best implementation path for your device.

The primary EPG download APIs are:

- TVGRID_LOOKUP (also known as "What Is On Now")
- TVGRID_LOOKUP (MODE: CONTRIBUTOR, AV_WORK, SERIES, SEASON)
- TVGRID SEARCH
- FIELDVALUES

In general, use 1.5.0 (see page 17) for client applications that download EPG data for currently-airing programming. This is sometimes called the "What Is On Now" feature.



Mote

In all cases, your application should use compression when downloading EPG data.

Managing TV Channel Logos for Different Transmission Types (SD, HD, 3D)

TV providers identify TV channels to consumers using TV channel logos. Some providers use different logos based on the quality of transmission (SD, HD, 3D). Others use the same logo for all transmission types, provided that the EPG data is the same.

To support these differences, your client application should implement the following logic:

 If a channel is announced as HDTV or 3DTV, then the client application should display an HD or 3D icon next to the channel name and logo. This allows the consumer to distinguish the SD, HD, and 3D variations for those cases where Gracenote returns the same station ID, name, and logo.

 If a TV program is shown on an SD channel, the client should ignore any HDTV or 3DTV attributes the program may carry.



Mote

Some TV channels are created without EPG data for special purposes, such as importance, testing, coverage, and so on. Your client application should handle these special cases.

Using TVGRID_LOOKUP (What Is On Now)

TVGRID LOOKUP retrieves a dynamically-computed TV grid. This guery is designed for lower memory-devices that do not need to retain the complete listings data available. The device specifies a list of channels and a time frame. If no time frame is specified (the DATE node is missing), the client will receive the default range, which is one hour in the past until five hours in the future.

As a best practice the device should in one query request at most:

- 25 channels
- 6 hours

To use this query, get a list of TV Channel GN IDs from the TV CHANNEL LOOKUP query. Then, send the TVGRID_LOOKUP query with these GN_IDs to the Gracenote Media Recognition Service.



A Important

Send only the channel GN_IDs you need to build a page or two of the on-screen grid.

Optionally, include the DATE in the query. Without a date, the query returns a default range of one hour in the past until five hours in the future from the time the server receives the request. All times are UTC.

To include URLS of TV programs images in the query, use the SELECT_EXTENDED option with a value of TVPROGRAM IMAGE. To retrieve generic images based on the IPG Category of the TV Programs, use the SELECT EXTENDED option with a value of IPGCATEGORY IMAGE.

The query response includes all TV programs in all TV Airings. A TV program does not have to be fully contained in the date range for it to be returned in the response. It is returned even if the TV program only partially falls into the date range. For example, the TV program is returned if it begins before the date range begins but ends before the date range ends. The TV program is also returned if it begins after the date range begins, and ends after the date range ends.

Using RANGE to Page Grid Data

RANGE is an option used to enable paging of grid data. Use the <START> and <END> values to specify the range of grid items to show in the response. Use the COUNT value in the response to get the total number of items in the entire grid for the query. <COUNT> is always returned in the response. To page through grid data:

1. Send the query, with or without the RANGE option.

- 2. In the response, obtain the RANGE <COUNT> value. This is the total number of items in the entire grid for the guery.
- 3. Send another query specifying the RANGE <START> and <END> values. For example, START=1 and END=10.
- 4. To get the next range of items, repeat the query with the same input, and specify the next RANGE. For example, START=10 and END=20.
- Continue sending queries and displaying the results until you have reached or exceeded the value of COUNT.

Example Query

```
<OUERIES>
  <AUTH>
  </AUTH>
  <QUERY CMD="TVGRID_LOOKUP">
   <TVCHANNEL>
     <GN_ID>a_gracenote_id</GN_ID>
     <GN_ID>a_gracenote_id</GN_ID>
      <GN_ID>a_gracenote_id</GN_ID>
     <GN_ID>a_gracenote_id</GN_ID>
     <GN ID>a gracenote id</GN ID>
   </TVCHANNEL>
   <DATE TYPE="START">2011-07-26T05:00</DATE>
   <DATE TYPE="END">2011-07-26T10:00</DATE>
   <OPTION>
      <PARAMETER>SELECT EXTENDED</PARAMETER>
      <VALUE>TVPROGRAM_IMAGE, IPGCATEGORY_IMAGE</VALUE>
   </OPTION>
  </OUERY>
</OUERIES>
```

Using TVGRID_LOOKUP with MODE CONTRIBUTOR, AV_WORK, SERIES, or SEASON

You can also use TVGRID_LOOKUP that returns a grid of upcoming showings of a Contributor, AV Work, Series, or Season. To do this, specify a MODE for the query as CONTRIBUTOR, AV_WORK, SERIES, or SEASON, an include the corresponding GN_ID. For example, to get the grid information of upcoming shows of a Series, you include the Series GN_ID. To include URLS of TV programs images in the query, use the SELECT_EXTENDED option with a value of TVPROGRAM_IMAGE. To retrieve generic images based on the IPG Category of the TV Programs, use the SELECT_EXTENDED option with a value of IPGCATEGORY_IMAGE. For additional information, see "TVGRID_LOOKUP (MODE: CONTRIBUTOR, AV_WORK, SERIES, and SEASON)" in the *Gracenote eyeQ Web API Reference*.

Using TVGRID_SEARCH

You can search for TV Program titles in the specified grid and return matching grid items. Include a full or partial text string of the TV Program title with the TVGRID_SEARCH query. The response is identical to TVGRID_LOOKUP, but only contains grid items referring to TV Programs whose title matched the search. To include URLS of TV programs images in the query, use the SELECT_EXTENDED option with a value of TVPROGRAM_IMAGE. To retrieve generic images based on the IPG Category of the TV Programs, use the SELECT_EXTENDED option with a value of IPGCATEGORY_IMAGE. For additional information, see "TVGRID_SEARCH" in the *Gracenote eyeQ Web API Reference*.

Using FIELDVALUES for IPG Category Values

The FIELDVALUES query retrieves all ID/string pairs for a list- or descriptor-based field. This query is useful for any client application that needs to process all potential IPG Category values. For example, the application might color-code particular values, or use the full list to implement searching or filtering by type. Because the list values may change over time, the FIELDVALUES query is useful to get the current set of available string and integer identifier pairs.

See "FIELDVALUES" in the *Gracenote eyeQ Web API Reference* for a partial list of IPG category values. Also see "IPG Category Types" in that document. Contact your Gracenote Professional Services representative for a complete list.

Best Practices for Image Dimensions

Gracenote music and video images – in the form of cover art, artist images and more – are integral features in many online music services, as well as home, automotive and mobile entertainment devices. Gracenote maintains a comprehensive database of images in dimensions to accommodate all popular applications, including a growing catalog of high-resolution (HD) images. Gracenote carefully curates images to ensure application and device developers are provided with consistently formatted, high quality images – helping streamline integration and optimize the end-user experience. This topic describes concepts and guidelines for Gracenote music and video images including changes to and support for existing image specifications.

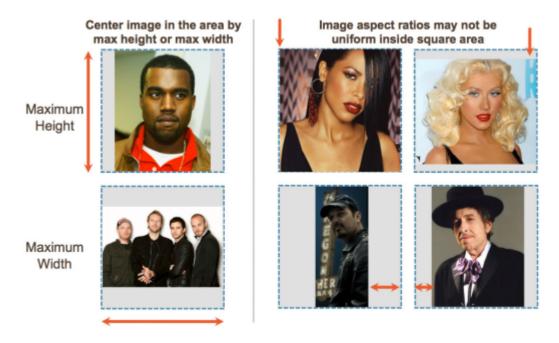
Image Resizing Guidelines

Gracenote music and video images are designed to fit within squares as defined by the available image dimensions. This allows developers to present images in a fixed area within application or device user interfaces. Gracenote recommends applications center images horizontally and vertically within the predefined square dimensions, and that the square be transparent such that the background shows through. This results in a consistent presentation despite variation in the image dimensions. To ensure optimum image quality for end-users, Gracenote recommends that applications use Gracenote images in their provided pixel dimensions without stretching or resizing.

Gracenote resizes images based on the following guidelines:

- 1. **Fit-to-square**: images will be proportionally resized to ensure their largest dimension (if not square) will fit within the limits of the next lowest available image size.
- 2. Proportional resizing: images will always be proportionally resized, never stretched.
- 3. **Always downscale:** smaller images will always be generated using larger images to ensure the highest possible image quality

Following these guidelines, all resized images will remain as rectangles retaining the same proportions as the original source images. Resized images will *fit into squares* defined by the available dimensions, but are not themselves necessarily square images.





For Tribune Media Services (TMS) video images only, Gracenote will upsize images from their native size (288 x 432) to the closest legacy video size (300 x 450) – adhering to the fit-to-square rule for the 450 x 450 image size. Native TMS images are significantly closer to 300 x 450. In certain situations, downsizing TMS images to the next lowest legacy video size (160 x 240) can result in significant quality degradation when such downsized images are later displayed in applications or devices.

Image Formats and Dimensions

Gracenote provides music and video images in six image dimensions to support a variety of applications, from mobile to HDTV. Applications or devices must specify images size when requesting an image from Gracenote. All Gracenote images are provided in the JPEG (.jpg) image format.

Available Image Dimensions

Gracenote provides images to fit within the following six square dimensions.

Image Dimension Name Pixel Dimensions

75	75 x 75
170	170 x 170
300	300 x 300
450	450 x 450
720	720 x 720
1080	1080 x 1080



Mote

Source images are not always square, and may be proportionally resized to fit within the specified square dimensions. Images will always retain their original aspect ratio.

Legacy Image Dimensions

Gracenote will continue to support the following legacy video image dimensions for backward compatibility with existing products.

Image Dimension Name	Pixel Width	Pixel Height
Thumb	40	60 or less
Small	160	240 or less
Medium	300	450 or less
Large	480	720 or less
X-Large	720	1080 or less

Common Media Image Dimensions

Music, video and artist images exist in a variety of dimensions and orientations. Gracenote resizes ingested images according to carefully developed guidelines to accommodate these image differences, while still optimizing for both developer integration and the end-user experience.

Music Cover Art

While CD cover art is often represented by a square, it is commonly a bit wider than it is tall. The dimensions of these cover images vary from album to album. Some CD packages, such as a box set, might even be radically different in shape.

Video Cover Art

Video cover art is most often taller than it is wide (portrait orientation). For most video cover art, this means that images will completely fill the vertical dimension of the requested image size, and will not fill the horizontal dimension. Therefore, while mostly fixed in height, video images may vary slightly in width. For example,

requests for a "450" video image will likely return an image that is *exactly* 450 pixels tall, but *close to* 300 pixels wide.

As with CD cover art, the dimensions of video covers also include packaging variants such as box sets which sometimes result in significant variations in video image dimensions.

Artist Imagery

Artist and Contributor images, such as publicity photos, come in a wide range of sizes and both portrait and landscape orientations. Video contributor images are most often provided in portrait orientation.

Variations in Video Image Dimensions

Video imagery commonly conforms to the shape of a tall rectangle with either a 3:4 or 6:9 aspect ratio. Image dimension characteristics of Gracenote Video imagery are provided in the following sections as guidelines for customers who want to implement Gracenote imagery into UI designs that rely on these aspect ratios. Gracenote recommends, however, that applications reserve square spaces in UI designs to accommodate natural variations in image dimensions.

AV Work Images

AV Work images typically conform to a 3:4 (width:height) aspect ratio.

3:4 (± 10%)	Narrower	Wider	Narrowest	Widest
98%	1%	1%	1:2	9:5

Video Product Images

Video Product images typically conform to a 3:4 (width:height) aspect ratio

3:4 (± 10%)	Narrower	Wider	Narrowest	Widest
90%	5%	5%	1:3	5:1

Contributor Images

Video Contributor images typically conform to a 6:9 (width:height) aspect ratio. Two ranges are provided due to larger variation in image dimensions.

6:9 (± 20%)	Narrower	Wider	Narrowest	Widest
90%	0%	10%	1:3	9:5
6:9 (± 10%)	Narrower	Wider	Narrowest	Widest
,				
69%	1%	30%	1:3	9:5