

TiVo Home Media Option: Music and Photos Server Protocol Specification

1 Introduction

This document describes how the server side of Music and Photos (the "Server") and the TiVo DVR software (the "Client") communicate in order to allow the user to view on one or more TiVo DVRs content stored on one or more PCs. The communication scheme employs standard URL syntax, HTTP protocols, and an XML-based format for "meta-data", which all come together as the "Music and Photos Server protocol".

1.1 Audience

Developers who wish to write a Music and Photos server.

1.2 Disclaimers

The Music and Photos Server protocol specification is provided solely for informational purposes. It is intended solely to enable you to write programs that allow you use your DVR to listen to music files or view photograph files that reside on your home computer for personal, noncommercial use. The Music and Photos Server protocol is not intended to be used for listening to or viewing music or photographs from third party sources including, but not limited to, streaming, webcasting, or any other form of transmission of content. Listening to music or viewing photographs from third party sources could subject your DVR and/or home computer to harm and may infringe the copyrights of third parties.

TIVO IS NOT RESPONSIBLE FOR ANY HARM TO YOUR DVR, COMPUTER, OR OTHER DEVICES RESULTING FROM YOUR USE OF THE MUSIC AND PHOTOS SERVER PROTOCOL. TIVO IS NOT RESPONSIBLE FOR, NOR HAS ANY CONTROL OVER, THE CONTENT OF ANY MUSIC OR PHOTOGRAPHS ACCESSED USING A TIVO DVR. UNAUTHORIZED COPYING OR DISTRIBUTION OF COPYRIGHTED WORKS IS AN INFRINGEMENT OF THE COPYRIGHT HOLDERS' RIGHTS. TIVO RESERVES THE RIGHT TO TERMINATE THE ACCOUNTS OF USERS OF ANY TIVO SOFTWARE WHO INFRINGE THE COPYRIGHTS OF OTHERS.

Redistribution and use of these specifications, with our without modification, are permitted provided that the redistribution contain the above copyright notice and the following disclaimer. Neither the name of TiVo, the TiVo logo, or any other TiVo trademarks or tradenames may be used to endorse or promote products derived from these specifications without specific prior written permission from TiVo Inc.

THESE MATERIALS ARE PROVIDED "AS IS", WITH ALL FAULTS AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. TIVO HEREBY DISCLAIMS ALL SUCH WARRANTIES INCLUDING, BUT NOT LIMITED TO, WARRANTIES AND/OR CONDITIONS OF FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, AND NON-INFRINGEMENT OF THIRD PARTY RIGHTS, EXCLUSIVELY OR RESULTS OBTAINED FROM USE OF THE MATERIAL. TIVO MAKES NO WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

1.3 References

[RFC 1738, Uniform Resource Locators \(URL\)](#)

[RFC 2068, Hypertext Transfer Protocol - HTTP/1.1](#)

[Extensible Markup Language \(XML\) 1.0 \(Second Edition\)](#)

2 Overview

The Music and Photos Server, running on a PC or other platform, serves music or photos to one or more TiVo DVRs. The original audio and graphics files can be in some other format, so long as the Server translates them into MP3 or JPEG, respectively.

Because it is possible to access the Server from any HTTP client program (such as a Web browser), the term "Client" throughout this document should be taken to refer to all such software, not just that running on a TiVo DVR.

Depending on the syntax of the URL used by the Client to request information, the Server returns either an individual document (much like any Web server) or textual "meta-data" describing the Server itself, the available content, etc.. The URL syntax is based on Internet standards for path construction, character encoding, and parameter passing.

While the structure of the meta-data is TiVo-specific, it employs an easily understood XML-based format. The primary purpose of this data is to present to the Client a uniform abstraction of the containers of "media" published from the Server. Where possible, this format has been designed to be "self-describing", allowing to be extended without modification to the existing specification.

2.1 Container Abstraction

When the Server returns information about the content available, it does so by returning meta-data describing a specific "container". Within this data, the individual "items" located in the container are described. These descriptions provide information about regular documents as well as sub-containers, allowing representation of arbitrarily deep containment hierarchies. Basic details such as type, title, and associated URLs are included in the description of each item.

The meta-data describing a container with a very large collection of items (such as a folder on the user's PC full of downloaded MP3 files) may consist of a huge amount of information, which might cause unacceptably long delays before the Client is able to fully receive and/or process it. To combat this, the Server's meta-data request protocol allows the description of any container to be "streamed" to the Client in smaller, more easily digested pieces, under Client control. In proper TiVo tradition, this makes it possible to retrieve just "the data you want, when you want it".

2.2 Extensible Item Details

In addition to the basic details (which are required), an extensible set of optional details is also available for each item in a container. This information is primarily intended to enrich the user-interface presented by the Client.

The complete set of details available for each item is always specific to the item. A particular MP3 music file, for example, might contain additional information about the "artist", "album", and "genre", while the details for a JPEG image created using some fancy digital camera might provide "capture date", "exposure settings", and a user-supplied "caption".

2.3 Application Flexibility

The meta-data format is not tied to any particular organization inherent in the source information, making it possible not only to represent "physical" containers (folders on a hard disk, for example), but also "virtual" ones (a randomized selection of images, a "TiVo Top 40" list based on thumb ratings, a genre/artist/album hierarchy built from ID3 tag data, and so on). This is all up to the Server and the meta-data it decides to generate.

3 Conventions

The following conventions apply throughout this document.

3.1 Syntax

Italicized text denotes optional portions of syntax or formats that may or may not be present in actual data. Curly-brace {enclosed} text denotes a specific "placeholder" in which actual data is inserted.

Consider the following URL syntax.

```
http://www.tivo.com/{document}
```

Here, the "{document}" portion is optional and might not appear in an actual URL. Further, {document} represents the place within the URL where the optional text can be inserted.

3.2 Hierarchy

Classical "dot notation" is used to indicate hierarchical relationships when referring to items located within a particular data structure.

Consider the following structure shown in XML format.

```
<Foo>
  <Bar>
    <Baz>Hello</Baz>
  </Bar>
  <Snarf>Cool</Snarf>
</Foo>
```

Here, the text at Foo.Bar.Baz is "Hello ". Likewise, the text at Foo.Snarf is "Cool". Foo.Crap does not refer to a relationship within this structure, nor does Foo.Bar.Junk.

4 Server Request Syntax

Clients can request two basic types of information from the Server, meta-data (usually representing the contents of some container) and document data (usually the binary image of some media file). Reflecting this are the two forms of URL recognized by the server.

Form 1: `http://{machine}/TiVoConnect?{parameters}`

Form 2: `http://{machine}/TiVoConnect/{document}?{parameters}`

In either form, {machine} should be replaced with the name or IP address of the machine running the target Server. The {parameters} portion of the URL is always present, although its content depends on the specifics of the request.

Regardless of the specifics, the first form is always used to send special "commands" to the server, which produces replies full of meta-data. The second form is always used to request a specific document (media file).

A typical Client, and especially the TiVo DVR-side software, can be fairly oblivious to the Server's request syntax. Instead, the Client normally extracts a complete URL from some meta-data previously received, optionally performs one or two simple string manipulations, and then sends the URL back to the Server, which returns a document or more meta-data, and so on.

4.1 TiVoConnect

The "/TiVoConnect" portion is included in every URL for the following reasons.

- To prevent "false" replies from machines running just any old Web server (that does not adhere to Music and Photos protocols).
- To allow the Music and Photos protocols to be implemented on top of an existing HTTP server, which typically requires that a "keyword" exist in every URL that requires special handling (in order to route the URL to the correct script or plug-in).

Consider what would happen if the QueryServer command request syntax (described in section 4.4.2) didn't contain "/TiVoConnect" and instead looked like:

```
http://{machine}?Command=QueryServer
```

It is very likely that a typical Web server receiving such a request would (ignoring the parameters in the URL) simply return the contents of its "index.html" file (instead of the meta-data expected by the Client).

During the early research phase of this project, we observed this very behavior from an Apache server running on a Linux box somewhere at TiVo.

4.2 URL Encoding

Clients should encode all URLs sent to the Server according to the relevant standards referenced in section 1.3. Particularly, special care should be taken to correctly encode any parameters embedded in the URL.

If the value of a given parameter is itself another URL, note that this has the effect of "double-encoding" some characters within that URL. The server expects this and will perform the proper decoding.

For example, consider the following string be used as the HTTP host name portion of an URL.

```
Me & You
```

To start with, the host name must be encoded in order to construct a valid URL. All characters illegal in the host name portion, such as spaces and ampersands, are escaped. For illustration, assume this URL also has two parameters.

```
http://Me%20%26%20You?Param1=A&Param2=B
```

If this URL is to be passed as the value of a parameter in a second URL, it must first be encoded again. All characters illegal for a parameter, such as slashes, ampersands and equal signs, are escaped.

```
http://SecondUrl?Param=http%3A%2F%2FMe%2520%2526%2520You%3FParam1%3DA%26Param2%3DB
```

However, when the Server receives this URL, extracts the parameter value and decodes it, the first URL (in its original form) is retrieved.

```
http://Me%20%26%20You?Param1=A&Param2=B
```

If needed, the original string used to create the host name (or any other portion of the first URL) can be recovered using a second pass through the decoder.

```
Me & You
```

4.3 Errors

Clients should be prepared to handle errors returned from the Server while processing any request. The list of possible errors is defined by the HTTP/1.1 specification.

4.4 Command Requests

As described previously, requests that issue commands to the Server use the following URL syntax.

```
http://{machine}/TiVoConnect?Command={command}&{parameters}
```

If the Server is not available, or {machine} is invalid, nothing is returned. A standard HTTP error is returned if the specified {command} is invalid or is not successful.

4.4.1 Parameters

Every URL submitted to the Server, and specifically all parameter names and values within every URL, must be properly escaped according to "RFC 1738, Uniform Resource Locators (URL)" referenced in section 1.3.

For the sake of readability, the names and values of parameters appearing in the examples shown throughout this document have not been escaped.

4.4.2 QueryServer

This command returns meta-data representing information about the Server itself, and uses the following URL syntax.

```
http://{machine}/TiVoConnect?Command=QueryServer
```

This command takes no parameters.

The TiVo DVR does not currently use this command. In the future, this command may be supported or even required by the TiVo DVR. Implementation of this command is a requirement for future-safe server design.

The meta-data format is described in section 5.3.

4.4.3 ResetServer

This command clears the Server's internal state.

```
http://{machine}/TiVoConnect?Command=ResetServer
```

Every Client should issue a ResetServer command as soon as possible after it has finished with the Server. This will allow the Server to release memory and any other resources associated with internal "session" state maintained for the Client. This command can also be used to put the Server into a known state before use.

This command has no parameters.

The TiVo DVR does not currently use this command. In the future, this command may be supported or even required by the TiVo DVR. Implementation of this command is a requirement for future-safe Server design.

For more details about sessions, see section 4.6.

4.4.4 QueryContainer

This command returns meta-data describing the contents of one of the containers available from the Server, and uses the following URL syntax.

```
http://{machine}/TiVoConnect?Command=QueryContainer
```

The meta-data format is described in section 5.4.

4.4.4.1 Container

This optional parameter may be set to the relative path of a container available from the Server, as shown here (assume "Foo" is a valid container).

```
http://{machine}/TiVoConnect?Command=QueryContainer&Container=/Foo
```

In this example, meta-data describing the contents of the container named "Foo" would be returned.

When the Container parameter is missing, the default value is "/", which corresponds to the Server's "root container".

4.4.4.2 Recurse

This optional parameter indicates whether or not the meta-data should contain only the items immediately within the container, or also every item contained within all "descendent" containers.

```
http://{machine}/TiVoConnect?Command=QueryContainer&Recurse={flag}
```

The value of {flag} is either "Yes" or "No". When the Recurse parameter is missing, the default value is "No".

For example, assume the following container structure exists.

```
/MyPhotos
  /Birthday
    Surprise.jpg
  /Christmas
    Kids.jpg
    Gifts.jpg
  Dog.jpg
  Cat.jpg
```

The meta-data returned for a QueryContainer command directed at MyPhotos with "Recurse=No" would include the following items.

```
Birthday
Christmas
Dog.jpg
Cat.jpg
```

The meta-data returned for the same command with "Recurse=Yes" would include the following items.

```
Birthday
Surprise.jpg
Christmas
Kids.jpg
Gifts.jpg
```

Dog.jpg
Cat.jpg

While no precise ordering should be implied from the preceding examples, note that the order of the items returned in the meta-data reflects the order in which the container(s) and the items within were traversed. Also note that the order of traversal is affected by the `SortOrder` parameter.

4.4.4.3 SortOrder

This optional parameter indicates the order in which items should appear in the meta-data.

```
http://{machine}/TiVoConnect?Command=QueryContainer&SortOrder={order}
```

`{order}` is a comma-delimited list of sort criteria, indicating the level(s) of sorting desired. Items are sorted according to the first entry in this list, then by the second, then by the third, and so on.

Possible values for the construction of `{order}` are:

- **Type** – Items appear in an order based on their general type. All containers sort before non-container items. Containers representing folders sort before those representing playlists.
- **Title** – Items appear alphabetically based on the text contained in their associated `Item.Details.Title` element (described in section 5.7.1.1).
- **CreationDate** – Items appear sorted from oldest to the newest based on an appropriate "creation" date (for example, the capture date of an image). If no such date is available, the creation date of the file containing the item's data is used instead.
- **LastChangeDate** – Items appear sorted from most- to least-recently modified.
- **Random** – Items appear in "randomized" order based on a "seed" supplied by the Client (described in section 4.4.4.4). This value may not be used with any other.

Any entry in the list can be prefixed with an exclamation point (!) to request sorting in the opposite direction at the corresponding level. For example, `!Date,Title` sorts from newest to oldest and then alphabetically.

With `"SortOrder=Random"`, note that randomization occurs across the entire sequence of items returned in the meta-data.

When the `SortOrder` parameter is missing, items appear in the container's "native" order. For a music playlist, for example, the items would appear in the order authored. For a file system folder, items would appear potentially "at random" according to how the file system is feeling.

When "streaming" sorted meta-data using multiple `QueryContainer` commands, Clients should specify the same `{order}` in each command.

4.4.4.4 RandomSeed

This parameter is required when the `SortOrder` parameter is `"Random"`.

```
http://{machine}/TiVoConnect?Command=QueryContainer&SortOrder=Random&
RandomSeed={seed}
```

{seed} can be any positive, non-zero, 32-bit unsigned integer value chosen by the Client (0-4294967295). Different values result in different item orderings. However, the same {seed} always produces the same ordering (assuming an unchanging set of items).

When "streaming" randomized meta-data using multiple QueryContainer commands, Clients should specify the same {seed} in each command.

This parameter is ignored unless the SortOrder parameter is "Random".

4.4.4.5 RandomStart

This optional parameter indicates which item should appear first in otherwise randomized meta-data.

```
http://{machine}/TiVoConnect?Command=QueryContainer&SortOrder={order}&
RandomSeed={seed}&RandomStart={url}
```

When "streaming" randomized meta-data using multiple QueryContainer commands, Clients should specify the same {url} in each command.

This parameter is ignored unless the SortOrder parameter is "Random".

4.4.4.6 ItemCount

This optional parameter indicates the maximum number of items that should be described in the meta-data, relative to the "anchor" (described in section 4.4.4.7).

```
http://{machine}/TiVoConnect?Command=QueryContainer&ItemCount={count}
```

{count} is a positive or negative integer. When {count} is positive, the items immediately following the anchor are described. When {count} is negative, the items immediately preceding the anchor are described. However, for the remaining discussion, assume {count} refers only to the absolute (non-negative magnitude) integer value.

If the container holds more than {count} items, "partial" meta-data is returned. Note that the meta-data will have fewer than {count} items if the container itself has fewer.

When the ItemCount parameter is missing, all items (preceding or following the anchor) are described.

4.4.4.7 AnchorItem

This optional parameter supplies the URL of the item that should precede (or follow) the first (or last) item described in the meta-data, if possible. This item is known as the "anchor" and its exact meaning depends on the value of the ItemCount parameter (described in section 4.4.4.4).

```
http://{machine}/TiVoConnect?Command=QueryContainer&AnchorItem={url}
```

When the AnchorItem parameter is missing or ignored, the anchor is the imaginary item that precedes (or follows) the first (or last) item in the container.

If the item identified by {url} no longer exists, then the item immediately following (or preceding) the position in the container the missing item would have occupied is used as the anchor instead. If this position cannot be determined, the AnchorItem parameter is ignored.

Note that {url} must be escaped as described in section 4.4.1.

4.4.4.8 AnchorOffset

This optional parameter supplies an "offset" to be applied to the location of the anchor, changing its effective location.

```
http://{machine}/TiVoConnect?Command=QueryContainer&AnchorItem={url}  
&AnchorOffset={offset}
```

{offset} is a positive or negative integer. When {offset} is positive, the effective location of the anchor is shifted downwards. When {offset} is negative, the location is shifted upwards.

When the AnchorOffset parameter is missing, no offset is applied to the anchor.

4.4.4.9 Parameter Precedence

In order to honor requests with multiple parameters, the following precedence scheme is used to resolve conflicts.

- Container – This meta-data always describes the requested container.
- Recurse – The meta-data always includes descendent items or not, as requested.
- SortOrder – The meta-data always describes items in the order requested.
- AnchorItem – The meta-data always describes the item(s) that immediately follow/precede the requested anchor.
- ItemCount – The meta-data always describes the number of items requested, unless also honoring anchorItem, or/and the number of items actually in the container makes this impossible. In this case, a different number of items is returned.

4.4.4.10 Filter

This optional parameter limits the types of items that appear in the meta-data.

```
http://{machine}/TiVoConnect?Command=QueryContainer&Filter={filter}
```

{filter} is a comma-delimited list of MIME types, indicating the desired value(s) for the Details.ContentType element of each item.

Any entry can contain a wildcard (*) in place of the "major" or "minor" portion of the MIME type. For example, "image/*" would match any item with a general MIME type of "image".

Any entry in the list also can be prefixed with an exclamation point (!) to limit the items returned to those NOT of the specified type.

When the Filter parameter is missing, the default value is "*/*". This will result in no restriction of the types of items appearing in the meta-data.

4.4.5 QueryItem

This command returns meta-data describing a single item within a container, and uses the following URL syntax.

```
http://{machine}/TiVoConnect?Command=QueryItem&Url={url}
```

The TiVo DVR does not currently use this command. In the future, this command may be supported or even required by the TiVo DVR. Implementation of this command is a requirement for future-safe Server design.

4.4.5.1 Url

This parameter must be supplied as it specifies what item to retrieve information about, as shown here.

```
http://{machine}/TiVoConnect?Command=QueryItem&Url=http://{machine}/Foo/Bar.jpg
```

In this example, the meta-data returned would describe Bar.jpg.

The full URL for the queried item is passed as a parameter to allow the Client to request information about items that don't necessarily reside on the same Server processing the QueryItem command.

Note that {url} must be escaped as described in section 4.4.1.

4.4.6 QueryFormats

This command returns meta-data describing all the formats in which the Server is able to provide a particular type of data, and uses the following URL syntax.

```
http://{machine}/TiVoConnect?Command=QueryFormats&SourceFormat={mime-type}
```

The TiVo DVR does not currently use this command. In the future, this command may be supported or even required by the TiVo DVR. Implementation of this command is a requirement for future-safe Server design.

4.4.6.1 SourceFormat

This parameter must be supplied as it specifies which format to query, where {mime-type} indicates the type of data to request the available formats for (for example, "audio/wave").

Note that {mime-type} may also contain a "wildcard" (for example, "image/*"). This queries for only those formats in which a general type of data (in this case, images) is available.

In the following example, the list of all audio formats available from the Server would be returned.

```
http://{machine}/TiVoConnect?Command=QueryFormats&SourceFormat=audio/*
```

Also consider this example, where a specific MIME type is used.

```
http://{machine}/TiVoConnect?Command=QueryFormats&SourceFormat=image/jpeg
```

Here, the list of formats in which the Server can provide JPEG images would be returned.

4.4.6.2 Default Formats

Clients can assume that the Server's default format for images is "image/jpeg", and for audio is "audio/mpeg". When necessary, QueryFormats can be used to verify this or to test support for additional formats.

4.4.7 Format

This optional parameter applies to all command requests.

```
http://{machine}/TiVoConnect?{command}&Format={type}
```

{type} can be either "text/xml" or "text/html". When "text/xml" is specified, the Server's normal meta-data format (described throughout section 5) is returned. When "text/html" is specified, a Web page containing the same information is returned, suitable for display in any browser.

For now, the "text/html" option is intended primarily for debugging and development purposes (i.e., so a regular Web browser can be used for testing the Server in place of TiVo DVR software).

When the Format parameter is missing, the default value is "text/xml".

4.5 Document Requests

As described previously, requests that retrieve document data from the Server use the following URL syntax.

```
http://{machine}/TiVoConnect/{document}?{parameters}
```

The {document} portion of the URL is completely up to the Server – Clients should not attempt to interpret it. However, Clients may modify the URL by appending certain parameters (described in the following sections). The set of applicable parameters depends on the type of document requested.

4.5.1 Format

This optional parameter may be set to any MIME type supported for the document requested (according to the meta-data returned by an appropriate QueryFormats command) as shown in the following example (assume "image/png" is a supported format for JPEG source data).

```
http://{machine}/TiVoConnect/Stuff/Picture.jpg?Format=image/png
```

The Server always converts the document to the requested (or implied default) format before returning it. When the Format parameter is missing, a default value is implied by the requested document.

If the requested document is not available in the specified format, a standard HTTP 415 (Unsupported Media Type) error is returned.

4.5.2 Image-specific Parameters

If the requested document's general MIME type is "image" (as in "image/jpeg"), optional parameters named Width, Height, Rotate, and PixelShape apply, when supported. Support for these parameters can be determined using the AcceptsParams element (described in section 5.8.1.1) within the meta-data associated with the URL. If such meta-data is not already available, the Client can retrieve it using the QueryItem command described in section 4.4.5. Clients can append any combination of these parameters when requesting an image needed with a particular size, orientation, and/or pixel shape.

4.5.2.1 Width & Height

These parameters supply the dimensions, in pixels, to which the returned image data should be scaled to fit within. The pixels are calculated according to the value of the PixelShape parameter. The following URL illustrates how an image could be scaled to 640 by 480 pixels.

```
http://{machine}/TiVoConnect/Stuff/ReallyBig.jpg?Width=640&Height=480
```

The aspect ratio of any scaled image is always preserved, which means that the actual dimensions of the image returned may differ from the Width and Height requested. Assuming ReallyBig.jpg in the example above is 1280 by 600 pixels, the image returned would actually be 640 by 300 pixels (not 640 by 480) in order to preserve the original aspect ratio.

4.5.2.2 Rotation

This parameter indicates (in degrees) the amount of clockwise rotation that should be performed on the returned image data. The following URL illustrates how an image could be turned upside down.

```
http://{machine}/TiVoConnect/Oops/WrongWayUp.jpg?Rotation=180
```

For now, the only valid rotations are positive and negative multiples of 90 degrees.

The Server will remember the most recently requested amount of rotation for each image and continue to return the image in the same orientation until another rotation is requested.

Note that the amount of rotation specified in each new request is always added to the last amount requested. In other words, an image could be "spun" counter-clockwise using repeated submissions of the following URL.

```
http://{machine}/TiVoConnect/Dizzy.bmp?Rotation=-90
```

4.5.2.3 PixelShape

This parameter indicates the "shape" of the pixel for the returned image data, expressed as a ratio of width to height.

```
http://{machine}/TiVoConnect/Picture.jpg?PixelShape={width}:{height}
```

Any two non-zero 32-bit unsigned integers can be specified for {width} and {height}.

The following URL illustrates how an image could be requested for distortion-free display on a system employing pixels three times wide as they are tall.

```
http://{machine}/TiVoConnect/Picture.jpg?PixelShape=3:1
```

Because the ratio can be expressed using any two numbers, note that the following example is exactly equivalent to the above.

```
http://{machine}/TiVoConnect/Picture.jpg?PixelShape=22023:7341
```

7341 x 3 = 22023.

When the PixelShape parameter is missing, the default value is "1:1" (which will return image data suitable for display only on a "square" pixel system).

4.5.3 Audio-specific Parameters

If the requested document's general MIME type is "audio" (as in "audio/mpeg"), optional parameters named Seek and Duration apply, when supported. Support for these parameters can be determined using the AcceptsParams element (described in section 5.8.1.1) within the meta-data associated with the URL. If such meta-data is not already available, the Client can retrieve it using the QueryItem command described in section 4.4.5.

4.5.3.1 Seek & Duration

When supported, the parameters can be appended in any combination to an audio request. Alone or together, these parameters affect which portion of the audio data is returned.

```
http://{machine}/TiVoConnect/{document}?Seek={msecs}&Duration={msecs}
```

Although the parameters are specified in milliseconds, the Server may round the values to the nearest unit (seconds, MPEG frames, etc.) required to extract valid data from the file.

If Seek or Duration (or the combination of Seek and Duration) specifies a time that exceeds the actual length of the audio, the data returned is truncated as necessary.

When the Seek parameter is missing, the data returned always starts at the beginning of the audio. When the Duration parameter is missing, the data continues to the end.

For example, the following URL will retrieve the first five seconds of CoolSong.mp3.

```
http://{machine}/TiVoConnect/MyMusic/CoolSong.mp3?Duration=5000
```

This URL will retrieve the third 10-second segment.

```
http://{machine}/TiVoConnect/MyMusic/CoolSong.mp3?Seek=20000&Duration=10000
```

4.6 Session Identification

An optional Session parameter applies to all Server requests, and uses the following URL syntax (assume "..." is any other valid request syntax).

```
http://{machine} ... &Session={id}
```

{id} is caller-defined, and is associated with internal state maintained by the Server. The use of an unknown {id} in any URL submitted to the Server results in the creation of a new instance of this internal state, known as a "session".

```
http://{machine}/TiVoConnect?Command=QueryContainer&Session=MyNewSession
```

A session provides the specific context in which the Server handles multiple, related requests, such as a series of QueryContainer commands. In order to create two or more independent (and possibly concurrent) "sessions" within the Server, the Client need only submit two or more URLs, each with a different {id} value.

Requests are always handled by the Server in the context of a session. When the Session parameter is missing from any URL, a default session is used.

Each {id} is tracked by the Server on a per-machine basis, using the IP address associated with each incoming URL. In this way, use of the same {id} by two or more machines communicating with the same Server results in the creation of two or more sessions, one for each machine. A unique default session is also associated with each machine.

All sessions are subject to being "timed out" after a period of inactivity determined by the Server, allowing any resources used to maintain the associated internal state to be freed.

5 Server Reply Formats

The meta-data returned for Music and Photos command requests use an XML-based syntax representing structured information about the Server itself, available containers, and the items within them. Document request replies consist of data in a standard format appropriate for the type of media contained in the document (e.g., JPEG data for images), although the source data will often be transformed in some way by the Server (converted, rotated, etc.) before the Client ever sees it.

5.1 URL Formats

Each URL that appears in the meta-data can be either "absolute" or "relative". An absolute URL always begins with "http://{machine}". A relative URL always begins with "/TiVoConnect".

Also, the double-encoding scheme described in section 4.2 applies to all URLs that appear in the meta-data. URLs appearing in meta-data will additionally be escaped according to the rules for CDATA sections in XML.

5.2 CDATA Escaping

Note that, for readability, the CDATA sections in the following XML examples have not been properly escaped. In actual meta-data, all proper escaping is required.

5.3 QueryServer

The reply for this command consists of meta-data that has the following format.

```
<TiVoServer>
<Version>{version}</Version>
<InternalName>{text}</InternalName>
<InternalVersion>{text}</InternalVersion>
<Organization>{text}</Organization>
<Comment>{text}</Comment>
</TiVoServer>
```

The TiVo DVR does not currently use this command. In the future, this command may be supported or even required by the TiVo DVR. Implementation of this command is a requirement for future-safe Server design.

5.3.1 TiVoServer

This element represents information about the Server itself.

Only one TiVoServer element appears in the meta-data.

5.3.2 TiVoServer.Version

This element indicates which version of the Calypso Server Protocol is supported by the Server. For now, {version} should always be "1".

Only one TiVoServer.Version element appears in the meta-data.

5.3.3 TiVoServer.InternalName

This element contains an organization-defined name for the Server software. For example, {text} might be "TiVoServer".

Only one TiVoServer.InternalName element appears in the meta-data.

5.3.4 TiVoServer.InternalVersion

This element supplied an organization-defined version for the Server software. For example, {text} might be "1.0.3".

Only one TiVoServer.InternalVersion element appears in the meta-data.

5.3.5 TiVoServer.Organization

This element identifies who created the version. For example, {text} might be "TiVo, Inc.".

Only one TiVoServer.Organization element appears in the meta-data.

5.3.6 TiVoServer.Comment

This element contains arbitrary human-readable text about the version. For example, {text} might be "Created on CORPBUILD01 at 03:00AM on 08/15/2000".

Only one TiVoServer.Comment element appears in the meta-data.

5.4 QueryContainer

This reply for this command consists of meta-data that has the following format.

```

<TiVoContainer>

  <ItemStart>{index}</ItemStart>
  <ItemCount>{count}</ItemCount>

  <Details>
    <Title>{text}</Title>
    <ContentType>{mime-type}</ContentType>
    <SourceFormat>{mime-type}</SourceFormat>
    <TotalItems>{count}</TotalItems>
  </Details>

  <Item>

    <Details>
      <Title>{text}</Title>
      <ContentType>{mime-type}</ContentType>
      <SourceFormat>{mime-type}</SourceFormat>
      <Duration>{msecs}</Duration>
    </Details>

    <Links>
      <Content>
        <Url>{url}</Url>
        :
      </Content>
      :
    </Links>

```

```

    </Item>
    ;

    <SourceChanged>{flag}</SourceChanged>

</TiVoContainer>

```

5.4.1 TiVoContainer

This element represents information about a container of documents and/or other sub-containers.

Only one TiVoContainer element appears in the meta-data.

5.4.2 TiVoContainer.Details

This element provides detailed information about the container itself, and always contains a few "standard" sub-elements and optionally several additional sub-elements, depending on the availability of the associated information.

In order to keep the size of TiVoContainer meta-data from becoming unpredictably large (as the additional information that might exist for the container is arbitrary), the set of details included is limited only to the most basic (title, type, etc.). If needed, the QueryItem command can be used at any time to retrieve the complete set of details for any container.

Note that the information included in TiVoContainer.Details is the same as that included in the TiVoContainer.Item.Details associated with the container in the description of the enclosing container.

Only one TiVoContainer.Details element appears in the meta-data. See section 5.7 for a complete description of the Details element.

5.4.3 TiVoContainer.ItemStart

This element indicates the position within the entire container (as a zero-based index) of the actual item represented by the first TiVoContainer.Item appearing within the meta-data.

5.4.4 TiVoContainer.ItemCount

This element indicates the number of items described in the meta-data.

5.4.5 TiVoContainer.SourceChanged

This element indicates whether any aspect of the container has changed since the last query. The possible values for {flag} are "Yes" and "No".

5.4.6 TiVoContainer.Item

Each instance of this element represents an individual item within the container.

Zero or more TiVoContainer.Item elements may appear in the meta-data, one for each document or sub-container described.

5.4.7 Item.Details

This element provides detailed information about an item. As with TiVoContainer.Details, the set of details included is limited only to the most basic (type, title, etc.). If the general MIME type of the item is "audio", the Duration detail is also included. If needed, the QueryItem command can be used at any time to retrieve the complete set of details for any item.

Only one TiVoContainer.Item.Details element appears per Item. See section 5.7 for a complete description of the Details element.

5.4.8 Item.Links

This element provides one or more URLs that can be used to retrieve data related to the item (for example, audio to accompany an image, or "cover artwork" for a container of music). See section 5.8 for a complete description of the Links element.

Only one TiVoContainer.Item.Links element appears per Item.

5.4.9 Item.Links.Content

This element provides the URL that can be used to retrieve the item itself and is always present.

5.4.10 The Root Container & Root-level Items

The items returned in the meta-data describing the Server's root container represent the "classes" of media available from the Server (music, photos, etc.). The Item.Details.Title element for each such item provides a human-readable description of the class ("Music on SOME_PC", for example), while the TiVoContainer.Details.Title element provides the name of the Server ("SOME_PC", for example).

The general type of media associated with each class can be derived from the Item.Details.ContentType element. For example, if a particular root-level container's SourceFormat is "x-container/tivo-music", the general MIME type of all documents in all sub-containers can be expected to be "audio". A reasonable {title} for such a container might be "Music on SOME_PC". See section 5.7.1.2 for a complete list of the MIME types that can appear in ContentType.

Note that this specification does not preclude the presence of multiple root-level containers with the same ContentType (for example, two "x-container/tivo-music" collections).

5.4.11 Example 1

Meta-data describing the root container of a simple Server supporting "music" might look something like this:

```
<TiVoContainer>

  <Details>
    <Title>SOME_PC</Title>
    <ContentType>x-container/tivo-server</ContentType>
    <SourceFormat>x-container/folder</SourceFormat>
    <TotalItems>1</TotalItems>
  </Details>

  <ItemStart>0</ItemStart>
  <ItemCount>1</ItemCount>

  <Item>
    <Details>
      <Title>Music</Title>
      <ContentType>x-container/tivo-music</ContentType>
      <SourceFormat>x-container/folder</SourceFormat>
    </Details>
    <Links>
      <Content>
        <Url>/TiVoConnect/?Command=QueryContainer&Container=/TheMusic</Url>
      </Content>
    </Links>
  </Item>
</TiVoContainer>
```

```

    </Links>
  </Item>

  <SourceChanged>No</SourceChanged>
</TiVoServer>

```

Armed with this data, the Client knows:

- One class of media is available with MIME type "x-container/tivo-music", named "Music".
- The content of the root-level container for the class can be retrieved using `http://{machine}/TiVoConnect?Command=QueryContainer&Container=/TheMusic` (where {machine} is the same machine the XML was retrieved from).

5.4.12 Example 2

The simplest possible meta-data for a container with two audio files and one sub-container might appear as follows.

```

<TiVoContainer>

  <Details>
    <Title>Songs</Title>
    <ContentType>x-container/folder</ContentType>
    <SourceFormat>x-container/folder</SourceFormat>
    <TotalItems>3</TotalItems>
  </Details>

  <ItemStart>0</ItemStart>
  <ItemCount>3</ItemCount>

  <Item>
    <Details>
      <Title>Cool</Title>
      <ContentType>audio/*</ContentType>
      <SourceFormat>audio/mpeg</SourceFormat>
      <Duration>1203094</Duration>
      <SongTitle>Cool Song</SongTitle>
    </Details>
    <Links>
      <Content>
        <Url>/TiVoConnect/Songs/Cool.mp3</Url>
      </Content>
    </Links>
  </Item>

  <Item>
    <Details>
      <Title>Mary </Title>
      <ContentType>audio/*</ContentType>
      <SourceFormat>audio/mpeg</SourceFormat>
      <Duration>96983</Duration>
      <SongTitle>Mary Had a Little Lamb</SongTitle>
    </Details>
    <Links>
      <Content>
        <Url>/TiVoConnect/Songs/Mary.mp3</Url>
      </Content>
    </Links>
  </Item>

  <Item>
    <Details>
      <Title>More Tunes</Title>
      <Type>x-container/playlist</Type>
      <SourceFormat>audio/mpegurl</SourceFormat>
    </Details>
  </Item>

```

```

    </Details>
    <Links>
      <Content>
        <Url>/TiVoConnect?Command=QueryContainer&Container=/Songs
          /Favorites/More</Url>
      </Content>
    </Links>
  </Item>

</TiVoContainer>

```

Armed with this data, the Client knows:

- This container, called "Songs", contains a total of three items.
- Audio data for the first item, named "Cool Song", can be retrieved using the URL "http://{machine}/TiVoConnect/Songs/Cool.mp3". The second item can be retrieved similarly.
- Meta-data about the sub-container, named "More Tunes", can be retrieved using "http://{machine}/TiVoConnect?Command=QueryContainer&Container=/Songs/Favorites/More".

5.4.13 Example 3

Here, slightly more complicated meta-data for a container with sub-containers representing music albums illustrates some of the (proposed) advanced possibilities.

```

<TiVoContainer>

  <Details>
    <Title>My Albums</Title>
    <ContentType>x-container/folder</ContentType>
    <SourceFormat>x-container/folder</SourceFormat>
    <TotalItems>11</TotalItems>
  </Details>

  <ItemStart>0</ItemStart>
  <ItemCount>2</ItemCount>

  <Item>
    <Details>
      <Title>Thriller</Title>
      <ContentType>x-container/playlist</ContentType>
      <SourceFormat>audio/mpegurl</SourceFormat>
      <ArtistName>Michael Jackson</ArtistName>
      <RecordLabel>Columbia Records</RecordLabel>
      <AlbumYear>1983</AlbumYear>
    </Details>
    <Links>
      <Content>
        <Url>/TiVoConnect/?Command=QueryContainer&Container=/Thriller</Url>
      </Content>
      <CoverArtwork>
        <Url>http://www.cddb.org/covers/943932.jpg</Url>
      </CoverArtwork>
    </Links>
  </Item>

  <Item>
    <Details>
      <Title>Spice World</Title>
      <ContentType>x-container/playlist</ContentType>
      <SourceFormat>audio/mpegurl</SourceFormat>
      <ArtistName>The Spice Girls</ArtistName>
      <RecordLabel>EMI</RecordLabel>
    </Details>
  </Item>

```

```

        <AlbumYear>1995</AlbumYear>
    </Details>
    <Links>
        <Content>
            <Url>/TiVoConnect?Command=QueryContainer&Container=/Spice</Url>
        </Content>
        <AudioClip>
            <Url>/TiVoConnect/Spice/Track3.mp3?Seek=34020&Duration=5234</Url>
        </AudioClip>
    </Links>
</Item>

</TiVoContainer>

```

In addition to the same kinds of information the Client knows in Example 2, in this example the Client also knows:

- The container has a total of 11 items, of which only the first two are described.
- The first album, "Thriller" by Michael Jackson, was released in 1983 on Columbia Records. The second, "Spice World" by The Spice Girls, in 1995 on EMI.
- An image for the first album (in this case, the cover artwork) can be retrieved using "http://www.cddb.org/covers/943932.jpg".
- An audio clip representative of the second album (perhaps the chorus from a "hit" song off the album) can be retrieved using "http://{machine}/TiVoConnect/Spice/Track3.mp3?Seek=34020&Duration=5234".

Note that "CoverArtwork" and "AudioClip" have simply been proposed as possible future sub-elements of Links, and are used here only for purposes of illustration. Likewise, not all the sub-elements of Details shown above are currently required in QueryContainer meta-data.

5.5 QueryItem

The reply for this command consists of meta-data that has the following format, which is very similar to that returned by the QueryContainer command. The main differences include the name of the enclosing element (TiVoItem) and restriction to a single Item sub-element.

```

<TiVoItem>
    <Item>
        <Details>
            <Title>{text}</Title>
            <ContentType>{mime-type}</ContentType>
            <SourceFormat>{mime-type}</SourceFormat>
            <TotalItems>{count}</TotalItems>
            <Duration>{msecs}</Duration>
            :
        </Details>
        <Links>
            <Content>
                <Url>{url}</Url>
                :
            </Content>
            :
        </Links>
    </Item>
</TiVoItem>

```

```

    </Item>

  </TiVoItem>

```

As with `TiVoContainer`, any additional information included in `Item.Details` depends on the general type of item. However, `TiVoItem` meta-data always includes ALL known item details, which (in addition to the differences already mentioned) is primarily what distinguishes it from `TiVoContainer`.

5.6 QueryFormats

This reply for this command consists of meta-data that has the following format.

```

<TiVoFormats>

  <Format>
    <Description>{text}</Description>
    <ContentType>{mime-type}</ContentType>
  </Format>
  :

</TiVoFormats>

```

5.6.1 TiVoFormats

This element represents a list of formats in which the Server can provide source data having the format specified by the `SourceFormat` parameter in the initial `QueryFormats` command.

Only one `TiVoFormats` element appears in the meta-data.

5.6.2 TiVoFormats.Format

Each instance of this element represents a supported format.

Zero or more `TiVoFormats.Format` elements may appear in the meta-data, one for each format supported. If none appear, the Server does not support the specified source data format.

5.6.3 Format.Description

This element provides a human-readable description of the format, such as "Portable Network Graphics Image" or "MPEG Layer 3 Audio".

Only one `TiVoFormats.Format.Description` element appears per `Format`.

5.6.4 Format.ContentType

This element identifies the format. Example values for `{mime-type}` include specific MIME types like "image/png" and "audio/mpeg".

Only one `TiVoFormats.Format.ContentType` element appears per `Format`.

5.7 The Details Element

Unless otherwise stated, each sub-element of `Details` appears only once within a particular instance of `Details`.

5.7.1 General Information

The following self-explanatory elements apply within any `Details` element.

```

<Details>

  <Title>{text}</Title>
  <ContentType>{mime-type}</ContentType>
  <SourceFormat>{mime-type}</SourceFormat>

  <CreationDate>{date}</CreationDate>
  <LastChangeDate>{date}</LastChangeDate>
  <LastAccessDate>{date}</LastAccessDate>
  <SourceLocation>{text}</SourceLocation>
  <SourceSize>{bytes}</SourceSize>
  <Copyright>{text}</Copyright>

</Details>

```

The following details are "basic" and will always be included in `TiVoContainer.Details` and/or `TiVoContainer.Item.Details` when available.

- Title
- ContentType
- SourceFormat

5.7.1.1 Details.Title

This element provides a human-readable description of the data, with no guarantee that {text} is unique in any way.

For an image, the {text} might be "Photo of my dog spot!", which should be displayed to the user as the primary "label" for the image.

5.7.1.2 Details.ContentType

This element indicates the content returned by submitting the URL provided by the associated `Links.Content.Url` element, where {mime-type} can be one of the following.

- x-container/folder – A collection of items for which the "native" order is not significant.
- x-container/playlist – A collection of items for which the "native" order is significant.
- x-container/tivo-server – The Server's root container, which is the only container to have this ContentType.
- x-container/tivo-music – A container of "music", where every item (including those in all descendent containers) has a ContentType with a major MIME type of either "x-container" or "audio".
- x-container/tivo-photos – A container of "photos", where every item (including those in all descendent containers) has a ContentType with a major MIME type of either "x-container" or "image".
- image/* – An image document for which the optional Format parameter can be any of those listed by the QueryFormats command where "SourceFormat=image/*".
- image/{type} – Where {type} is a specific MIME sub-type, such as "jpeg". In this case, the item is not available from the Server in any other format.

- audio/* – An audio document for which the optional Format parameter can be any of those listed by the QueryFormats command where "SourceFormat=audio/*".
- audio/{type} – Where {type} is a specific MIME sub-type, such as "mpeg". In this case, the item is not available from the Server in any other format.

For any ContentType with a general MIME type of "x-container", the data returned is always meta-data using the QueryContainer reply format, described in section 5.4.

5.7.1.3 Details.SourceFormat

This element indicates what kind of "source data" the Server derived its information from. The value of {mime-type} may be any specific MIME type (including ones not mentioned in this document). This information is intended mainly for informational purposes, perhaps affecting UI displayed by the Client.

When it appears within TiVoContainer.Details (or TiVoContainer.Item.Details for container items), SourceFormat indicates the type of information from which the Server actually derives the meta-data. In this case, possible values for {type} include potentially any established MIME type for "container-like" information or one of the following TiVo-defined types.

- x-container/folder – The container was derived from a file system (usually that of the system running the Server) or some other unordered collection of items.

The classic example of an established MIME type that might appear as the SourceFormat for a container is "audio/mpegurl" (indicating that the Server derived the meta-data from a WinAmp Playlist).

5.7.1.4 Details.SourceLocation

This optional element provides a textual description of the location of the source data. This can be a Windows- or UNIX-style path, or an URL, or something else entirely (dependent on the platform running the Server).

For example, the location of a JPEG image from a Server running on a Windows computer might be "C:\My Pictures\Vacation 2002\Photo.jpg".

5.7.1.5 Details.SourceSize

This optional element provides the actual size of the source data (in bytes).

Note that this size will often be different than the size of the data received by the Client from a document request because the Server must almost always transform it in some way before returning it.

5.7.1.6 Details.CreationDate

This optional element indicates the date the source data was created.

Unless otherwise specified, all {date} values are UNIX-style (number of seconds since midnight January 1, 1970 GMT) formatted in hex notation. For example, "0xABCD1234".

5.7.1.7 Details.LastChangeDate & Details.LastAccessDate

This optional elements indicate the dates of the most recent modification and use of the source data, respectively (and not necessarily only via the Server).

5.7.1.8 Details.Copyright

This optional element provides a copyright notice associated with the data.

5.7.2 Audio-specific Information

When Details.ContentType specifies a general mime type of "audio", the following self-explanatory elements also apply.

```

<Details>

  <ArtistName>{text}</ArtistName>
  <AlbumTitle>{text}</AlbumTitle>
  <SongTitle>{text}</SongTitle>
  <AlbumYear>{date}</AlbumYear>
  <MusicGenre>{text}</MusicGenre>
  <SourceBitRate>{bits-per-second}</SourceBitRate>
  <SourceSampleRate>{samples-per-second}</SourceSampleRate>
  <Duration>{msecs}</Duration>
  <RecordLabel>{text}</RecordLabel>

</Details>

```

The following details are "basic" and will always be included in TiVoContainer.Item.Details when available.

- ArtistName
- AlbumTitle
- SongTitle
- MusicGenre
- AlbumYear
- Duration

5.7.2.1 TiVoAccurateDuration

Due to the expense of calculating a completely accurate duration for certain MPEG audio files, the HTTP reply for each audio document request includes a header named "TiVoAccurateDuration". This header uses the same format as Details.Duration, although it may contain potentially more accurate information. This provides a way to defer the necessary calculations until the last possible moment.

Note that, until the initial request for an audio document occurs, the associated Details.Duration value may only be an estimate and so may differ from TiVoAccurateDuration.

5.7.3 Image-specific Information

When Details.ContentType specifies a general mime type of "image", the following self-explanatory elements also apply.

```

<Details>

  <CaptureDate>{date}</CaptureDate>
  <SourceWidth>{pixels}</SourceWidth>
  <SourceHeight>{pixels}</SourceHeight>
  <SourceColors>{count}</SourceColors>
  <SourceResolution>{dpi}</SourceResolution>
  <Caption>{text}</Caption>
  <Keywords>{comma-delimited-text}</Keywords>

</Details>

```


The following details are "basic" and will always be included in `TiVoContainer.Item.Details` when available.

- `CaptureDate`
- `CreationDate`
- `LastChangeDate`

5.7.4 Container-specific Information

When `Details.ContentType` specifies a general MIME type of "x-container", the following self-explanatory elements also apply.

```
<Details>

  <TotalItems>{count}</TotalItems>
  <TotalSize>{bytes}</TotalSize>
  <TotalDuration>{msecs}</TotalDuration>

</Details>
```

Note that `TotalItems` will always be included in `TiVoContainer.Details`.

Note that it may be impractical for the Server to provide certain optional details (`TotalDuration`, for example) for some types of containers at all times.

Unless other specified, note that container-specific information returned in `TiVoContainer.Details` (that is, as the result of a `QueryContainer` command) is affected by the `Recurse` and `Filter` parameters. For example, `TotalItems` may decrease in value if certain types of items are filtered out.

5.7.4.1 Details.TotalItems

`TotalItems` provides the total number of all items in the container.

5.7.4.2 Details.TotalSize

This element indicates the combined size (in bytes) of all items in the container.

5.7.4.3 Details.TotalDuration

`TotalDuration` provides the combined durations of all audio items in the container. This item is optional because, 1) not all containers contain audio items, and 2) the information may not be readily available.

5.8 The Links Element

The common `Links` element can contain one or more URLs referring to information about the item. Following is the general syntax for each sub-element of `Item.Links`.

```
<{link-name} >

  <Url>{url}</Url>
  <AcceptsParams>{flag}</AcceptsParams>
  :
</{link-name}>
```

5.8.1.1 {link-name}.AcceptsParams

This optional element indicates whether or the server serving the link understands the applicable Music and Photos parameters (such as Format, Seek, Width, etc.). Possible values for {flag} are "Yes" and "No".

When the AcceptsParams element is missing, the Client can assume that the server accepts parameters.

5.8.2 Container Links

Following is the full syntax for various elements that can appear within Item.Links.Content where the associated Item.Details.ContentType has a general MIME type of "x-container".

```
<Content>
  <Url>{url}</Url>
  <AcceptsParams>{flag}</AcceptsParams>
</Content>
```

For container links, Content.Url always contains an URL that can be used to retrieve QueryContainer format meta-data describing the container.

5.8.3 Audio Links

Following is the full syntax for various elements that can appear within Item.Links.Content where the associated Item.Details.ContentType has a general MIME type of "audio" (or where the link is defined as returning audio data).

```
<{audio-link}>
  <Url>{url}</Url>
  <AcceptsParams>{flag}</AcceptsParams>
  <LoopCount>{loops}</LoopCount>
</{audio-link}>
```

Typically, {audio-link} will be "Content", but future Links elements may contain addition links defined as returning audio. Possible names for such links are "AudioClip" and "BackgroundAudio".

5.8.3.1 {audio-link}.Url

This element contains an URL that can be used to retrieve the audio data.

5.8.4 Image Links

Following is the full syntax for various elements that can appear within Item.Links.Content where the associated Item.Details.ContentType has a general MIME type of "image" (or where the link is defined as returning image data).

```
<image-link>
  <Url>{url}</Url>
  <AcceptsParams>{flag}</AcceptsParams>
  <RefreshInterval>{msecs}</RefreshInterval>
  <ShowStyle>{style}</ShowStyle>
  <ShowDelay>{msecs}</ShowDelay>
</image-link>
```

Typically, {image-link} will be "Content", but future Links elements may contain addition links defined as returning an image. Possible names for such links are "Icon", "CoverArtwork", and "BackgroundImage".

5.8.4.1 {image-link}.Url

This element contains an URL that can be used to retrieve the image data.
