

A3. RadioActivity.fm User's Manual, Addendum 3

Direct playlist entry via HTTP gateway

Version 1.0 / 10.05.2007

The RadioActivity.fm direct playlist entry gateway

The RadioActivity.fm direct playlist entry gateway lets stations make entries to their station's playlist database by sending specially formatted HTTP GET queries to a RadioActivity.fm URL.

The playlist gateway is intended as a tool to help stations integrate existing automation and systems into their RadioActivity.fm playlist logging. A station using an automation solution that outputs text files, for example, can create scripts to parse these files and update their RadioActivity.fm playlists by using the playlist gateway.

RadioActivity.fm is working with stations to create and document such sample scripts, on a per-automation solution basis. RadioActivity.fm is currently working on **MegaSeg**, **BSI**, and **iTunes** automation integration, but if your station has a specific automation need, please feel free to contact us.

Why would our station need such a thing?

Many stations have a mix of live DJ's and 'automation'. While live DJ's are tasked with entering their playlists into the RadioActivity.fm website by hand, stations still need a way to let station automation systems 'talk' to RadioActivity.fm.

A sample station may have live DJ's from 9am until midnight, but the midnight-to-9am shift plays music from a random playlist in iTunes, Winamp, MegaSeg, and/or any number of other automation solutions.

The rules, so to speak, dictate that we still need to keep track of those plays originating from the automation player. Thus, the direct playlist entry gateway is provided to allow such communication to take place.

Note: If you aren't familiar and comfortable with your automation, as well as basic scripting concepts and light programming, STOP. This particular addendum probably won't be of any use to you.

If you are experienced, and want to get started right now, you can skip ahead to the "Mandatory Variables" and "Gateway URL examples" sections and glean most of the information you need to get started.

Playlist Gateway How-To

First: Make sure your station has a gateway password:

Before your station can begin using the playlist gateway, you will have to request a **playlist gateway password**.

This password is a random six-letter password you will include in your HTTP GET requests for authentication purposes. **Requests with a missing or incorrect password are ignored by the playlist gateway.**

To view your station's gateway password – or to request one –

- 1) Log in to RadioActivity.fm with an administrator account.
- 2) Click **Station Admin -> Gateway Password**
- 3) If your station already has a gateway password, it will be displayed.
- 4) If your station does not yet have a gateway password, click '**Request a gateway password**'. Once your key has been issued, it will display here, and a RadioActivity.fm admin will email you when your password has been set.

Please note: The gateway password is a very light layer of security intended to keep outsiders from abusing the playlist gateway -- and to keep bogus entries out of station playlists. Please keep your gateway password as secret as possible.

If, for some reason, your station would ever like to change your gateway password, please contact RadioActivity.fm via email.

Request Variables

Mandatory and Optional Variables

There are ten variables to include when calling the RadioActivity.fm direct playlist entry gateway, four of which are optional.

Gateway request variables

Mandatory variables:

Fieldname:	Notes:
password	- This is your six-letter gateway password
title	- Title of the track being logged
artist	- Artist name for the track being logged
album	- Album name for the track being logged
label	- Label name for the track being logged
djoid	- The DJ OID of the DJ assigned to this track's logging (See below re: using an existing DJ vs. an automation DJ)

Optional variables:

Fieldname:	Notes:
timestamp	- <i>mm-dd-yyyy hh:mm:ss</i> formatted timestamp of the track's time of play
genre	- The plain-english description of the genre for the entry
playtype	- The plain-english description of the playtype for the entry
showoid	- The show ID for the show you would like to log the playlist under

Variable ordering?

As you will see in the request examples in the following sections, these variables don't have to be entered in any particular order. **As long as the mandatory fields are included - and properly formatted - it doesn't matter what order you put the variables in the HTTP GET request.**

Mandatory variables

On the 'password' variable:

The **password** variable is your six-letter gateway password assigned by RadioActivity.fm, and found under **Admin->Station Admin->Gateway Password**

The password variable is mandatory. Requests without a password, or with an invalid password, will be rejected by the gateway.

On the 'title', 'artist', 'album', and 'label' variables:

The **title**, **artist**, **album**, and **label** fields are - perhaps obviously - the title, artist, album, and label of the track being logged. All are mandatory. Requests missing one or more of these variables will be rejected.

Because these are the variables may include foreign characters, spaces, and the like – and are being sent via HTTP GET – be sure to see the special note in the following section '*Urlencoding your request variables*'.

On the 'djoid' variable:

The **djoid** variable is an integer that ties the play you are entering to a specific DJ. The **djoid**'s you use in your gateway requests must match to a valid, active **djoid** at your station.

You can discover an existing DJ's **djoid** by logging into RadioActivity.fm as an admin, Clicking **User admin**, then their name, and noting the number next to the **DJ #:** field.

It is left to the station whether they want to create a new RadioActivity.fm user for their automation, and use that new 'automation user's djoid - or instead to tie them to an existing (human) DJ's entries.

The playlist gateway will reject any entries without a **djoid**, or with a **djoid** that doesn't belong to your station's list of active DJ's.

Optional variables

On the 'timestamp' variable:

The timestamp field is an optional timestamp for the play you are entering, formatted in

mm-dd-yyyy hh:mm:ss

... style. The 'hh', which represents hours, is a 24-hour format, aka 'military' time. Thus, any 'am/pm' designations are **not** part of the timestamp, and should not be sent.

Your particular automation system may or may not log timestamps. If it does, the *timestamp* variable will allow you to use your automation system's timestamp, considered more accurate, instead of the default, which is the time your gateway entry is sent to RadioActivity.fm.

The timestamp variable is **optional**. If your request doesn't include a timestamp, RadioActivity.fm will record that play with the current time, corrected for your station's local timezone.

Also, timestamps in any other format than the *mm-dd-yyyy hh:mm:ss* mentioned above will be rejected.

On the 'genre' variable:

The **genre** variable is an optional entry for the same plain-english genre your station uses to track its genres – i.e. 'country', 'hip hop', 'rock', etc.

Any specified genre included in your request must match an existing, predefined genre your station has on file. Genres that aren't in your station's list of active genres will be rejected.

The genre field is optional. Omitting the genre won't severely impact your station's recordkeeping, but you won't be able to generate genre-based reports on plays without the genre specified.

On the 'playtype' variable:

Much like the genre variable above, the **playtype** variable is an optional entry for the playtype your station is using to track playtypes – i.e. 'CD', 'Vinyl', 'MP3', etc.

Take care to ensure that the playtype you include with your gateway request is the same as it appears in your station's genres – note their capitalization, any hyphens, etc. Any Playtype that isn't in your station's list of active playtypes will be rejected, so it is important to make sure they match.

It is left up to stations whether or not they want to create a special playtype for their station, perhaps called 'automation', for use only with the RadioActivity.fm gateway entries. Doing so, however, would help distinguish live plays from automated plays.

Just like the genre variable, the playtype field is optional, but omitting the playtype won't necessarily impact your station's recordkeeping, but you won't be able to generate playtype -based reports on plays without the playtype specified.

Optional variables (continued)

On the 'showoid' variable:

The **showoid** variable allows you to assign gateway playlist entries to a specific show.

An entry with a **showoid** must have a corresponding **djoid** – and that **djoid** must belong to a station DJ that has been assigned to that show.

Show id's can be found by clicking **Show admin** -> (the show name) and noting the number next to "**Show ID#**"

The idea of adding automated plays to a show may seem odd, but the option exists. Your station, for example, may wish to create a show called "Automated show" or "Off the air" to be displayed when automates plays are coming in, and then make your automation user the DJ for that show. This would help keep your website's public face properly updated via the "onair.xml" file – and would help further delineate live plays from automated plays inside your playlists.

Urlencoding your request variables

Because your gateway requests will include words that may contain spaces, ampersands, and other non-HTTP GET friendly characters, all the fields you send should be **urlencoded** before including them in your request.

What is urlencoding?

Urlencoding is the process of changing a string for inclusion in a URL, where non-alphanumeric strings are specially encoded to be 'web friendly' by replacing them with different characters.

Consider the following playlist entry:

Title: Tahitian Skies
Artist: Chief O'Neill's Hornpipe Chieftains
Album: Another Country
Label: RCA Victor

When this entry is urlencoded, it will look like this:

title=Tahitian+Skies&artist=Chief+O%27Neill%27s+Hornpipe+Chieftains&album=Another+Country&label=RCA+Victor

The replacement symbols: **+**, **%27**, and so on, are the result of the URLEncoding. This method also handles any foreign characters that may appear – such as **é**, **ñ**, and so on.

How do I urlencode?

The method you use for urlencoding will differ according to whatever scripting language you are using. In PHP, the **urlencode()** function. In ASP, the function is called **Server.URLEncode()**

RadioActivity.fm Gateway Request Examples

Below are a number of valid RadioActivity.fm gateway requests. Note that requests start with

<http://www.radioactivity.fm/gateway/index.php?>

... and have many variables added onto the end of the URL.

Example 1:

This example is for station **KFAKE**, who has the following password : **qqq777**
The **djoid** for this entry, is **12**. The song being logged is **Bob Marley's Exodus**, off the **Legend** album, on the **Island** label. Here, the **timestamp**, **genre**, and **playtype** variables have also been specified.

```
http://www.radioactivity.fm/gateway/index.php?password=qqq777&title=Exodus&artist=Bob+Marley&album=Legend&label=Island&djoid=12&timestamp=12-19-2006+15%3A57%3A00&genre=Alternative&playtype=A-list
```

Because it doesn't matter what order the variables appear in, the following request is also valid, and exactly equal to the request above:

```
http://www.radioactivity.fm/gateway/index.php?playtype=A-list&timestamp=12-19-2006+15%3A57%3A00&password=qqq777&artist=Bob+Marley&album=Legend&label=Island&djoid=12&genre=Alternative&title=Exodus
```

Example 2:

Here's an example that includes foreign characters in the song information. Note how urlencoding turns the foreign characters into numerical codes.

This example is for station **KFAKE**, who has the following password : **qqq777** The Djoid, for this entry, is **12**. The song being logged is **José Flaco's** song **Puerto Peñasco**, off the **Amor** album, on the **Flaco** label. Here, only the optional **timestamp** variable has been specified.

```
http://www.radioactivity.fm/gateway/index.php?password=qqq777&title=Puerto+Pe%F1asco&artist=Jos%E9+Pescado&album=Amor&label=Flaco&djoid=12&timestamp=10-09-2007+15%3A57%3A00
```

Example 3:

Here's another example, for the same station. Because this request doesn't include a timestamp, the play's timestamp will be set to the time that this gateway request was sent to RadioActivity.fm.

```
http://www.radioactivity.fm/gateway/index.php?password=qqq777&title=Big+Love&artist=Kevin+Drew&album= Spirit+If&label= Arts+%26+Crafts&djoid=12
```

Example 4:

Here's an example that includes **djoid** and **showoid** variables to tag this entry to a specific show.

In this example, the show's oid is **111**, and the djoid is **379**. This entry doesn't fail, because DJ# **379** is, indeed, a member of show # **111**.

```
http://www.radioactivity.fm/gateway/index.php?password=qqq777&title=testTitle&
artist=testArtist&album=testAlbum&label=testLabel&genre=Alternative&playtype=
A-list&showoid=111&djoid=379
```

Request Throttling:

Because your scripts may be sending requests that can contain duplicate entries, for various reasons, the RadioActivity.fm gateway has built in **request throttling**.

NOTE: To prevent multiple duplicate entries, RadioActivity.fm will discard any request that is the same as any of the last five gateway requests.

Remember: If your scripts send a request in that is exactly the same as any of the last **five** entries sent in by your station, RadioActivity.fm will discard the request and return an error of "*The song you sent in was listed in the last 5 tracks, and prevented from more entry*".

Error messages and explanations:

If your gateway request is valid, RadioActivity.fm will return a single response: **OK**

If, however, your gateway request is invalid, or contains errors, RadioActivity.fm will return the message **Error:**, followed by a message as to why the request failed.

Listed below are all the possible error messages RadioActivity.fm may return, what they mean, and how to address them.

Message: *Missing mandatory field: password*

Meaning: Your gateway request didn't include a password field.

Fix: Make sure your request includes a `password=nnnnnn` where `nnnnnn` is your station's assigned gateway password.

Message: *Password field: was wrong length*

Meaning: The gateway password you provided wasn't 6 characters long.

Fix: Make sure the gateway password you provided is exactly 6 characters long.

Message: *Password failed: not a valid password*

Meaning: The gateway password you provided wasn't on file at RadioActivity.fm

Fix: Make sure the gateway password you provided is the one found by going to **Station Admin** -> **Gateway Password** in RadioActivity.fm. You will have to login as a station admin.

Error messages and explanations (continued)

Message: *Missing mandatory field: title*

Meaning: The request didn't include a *title* – i.e. the title of the track being logged.

Fix: Make sure your request includes a *title=nnnnnn* where *nnnnnn* is the title. Also, make sure that *title* is urlencoded correctly.

Message: *Missing mandatory field: artist*

Meaning: The request didn't include a *artist* – i.e. the artist of the track being logged.

Fix: Make sure your request includes a *artist=nnnnnn* where *nnnnnn* is the artist. Also, make sure that *artist* is urlencoded correctly.

Message: *Missing mandatory field: album*

Meaning: The request didn't include a *album* – i.e. the album name of the track being logged.

Fix: Make sure your request includes a *album=nnnnnn* where *nnnnnn* is the album name. Also, make sure that *album* is urlencoded correctly.

Message: *Missing mandatory field: label*

Meaning: The request didn't include a *label* – i.e. the name of the label for the track being logged.

Fix: Make sure your request includes a *label=nnnnnn* where *nnnnnn* is the label name. Also, make sure that *label* is urlencoded correctly

Message: *Missing mandatory field: djoid*

Meaning: The request didn't include a *djoid*, or the *djoid* provided wasn't a number.

Fix: Make sure your request includes a *djoid=xxx* where *xxx* is an integer that corresponds to one of your station's DJ id numbers.

Message: *specified DJOID (number) does not belong to station*

Meaning: Your request contained a *djoid* number that does not belong to any of your station DJ's id numbers.

Fix: Make sure that the *djoid* number you have provided does, indeed, belong to one of your DJ's.

Message: *the specified DJOID (number) is not currently marked ACTIVE";*

Meaning: Your request contained a valid *djoid* number for your station – but that particular DJ isn't marked as active.

Fix: Make sure that DJ's account is active by logging in as a station admin and checking their status is **active** under **User Admin**. Conversely, you could use a different DJOID, one that you have confirmed as valid.

Message: *Timestamp date is malformed: month*

Meaning: Your request included a *timestamp* variable, but the *month* portion was somehow malformed, included non-number characters where numbers were expected, or included too few or too many digits.

Fix: Make sure that the *month* variable you are sending is composed of exactly two digits, ranging from 01 to 12, and appears in a timestamp formatted as *mm-dd-yyyy hh:mm:ss*

Error messages and explanations (continued)

Message: *Timestamp date is malformed: day*

Meaning: Your request included a *day* variable, but the *day* portion was somehow malformed, included non-number characters where numbers were expected, or included too few or too many digits.

Fix: Make sure that the *day* variable you are sending is composed of exactly two digits, ranging from 01 to 31, and appears in a timestamp formatted as *mm-dd-yyyy hh:mm:ss*

Message: *Timestamp date is malformed: year*

Meaning: Your request included a *year* variable, but the *year* portion was somehow malformed, included non-number characters where numbers were expected, or included too few or too many digits.

Fix: : Make sure that the *year* variable you are sending is composed of exactly four digits, and appears in a timestamp formatted as *mm-dd-yyyy hh:mm:ss*

Message: *Timestamp date is malformed: hour*

Meaning: Your request included an *hour* variable, but the *hour* portion was somehow malformed, included non-number characters where numbers were expected, or included too few or too many digits.

Fix: Make sure that the *hour* variable you are sending is composed of exactly two digits, ranging from 00-23, and appears in a timestamp formatted as *mm-dd-yyyy hh:mm:ss*.
(Remember, *RadioActivity.fm* is expecting hours in military time format)

Message: *Timestamp date is malformed: minute*

Meaning: Your request included a *minute* variable, but the *minute* was somehow malformed, included non-number characters where numbers were expected, or included too few or too many digits.

Fix: Make sure that the *minunte* variable you are sending is composed of exactly two digits, ranging from 00-59, and appears in a timestamp formatted as *mm-dd-yyyy hh:mm:ss*.

Message: *Timestamp date is malformed: seconds*

Meaning: Your request included a *seconds* variable, but the *seconds* portion was somehow malformed, included non-number characters where numbers were expected, or included too few or too many digits.

Fix: Make sure that the *seconds* variable you are sending is composed of exactly two digits, ranging from 00-59, and appears in a timestamp formatted as *mm-dd-yyyy hh:mm:ss*.

Error messages and explanations (continued)

Message: *specified genre: (genre name) is not one of stations known genres*;
Meaning: Your request included a *genre*, but the *genre* specified didn't match up to any valid, active genres on file for your station.
Fix: Make absolutely sure that the *genre* you are specifying in your request is the same as the genres on file for your station. Pay special attention to capitalization, hyphens, etc.

Message: *specified playtype: (playtype name) is not one of stations known playtypes*;
Meaning: Your request included a *playtype*, but the *playtype* specified didn't match up to any valid, active playtypes on file for your station.
Fix: Make absolutely sure that the *genre* you are specifying in your request is the same as the genres on file for your station. Pay special attention to capitalization, hyphens, etc.

Message: Show oid provided, but not numeric
Meaning: The request included a *showoid* variable, but it wasn't a number.
Fix: make sure the *showoid* variable you are sending consists of only an integer.

Message: Show oid provided, but does not belong to this station.
Meaning: The request included a *showoid* variable, but the *showoid* specified doesn't match any show for the specified station.
Fix: Make sure you are providing the correct *showoid* value – namely, one that belongs to your station.

Message Show oid provided, but missing a djoid.
Meaning: The request included a *showoid* variable, but no *djoid* variable.
Fix: When a request has a *showoid* variable, RadioActivity.fm has to know which DJ is also being assigned for this play, because many different DJ's can take part in any one show. Make sure to include a valid *djoid* for a DJ as well.

Message: Showoid and djoid provided, but the DJ does not belong to that show.
Meaning: The request included a *showoid* variable, and a *djoid* variable, but the *djoid* for the DJ provided doesn't belong to the show specified by *showoid*.
Fix: Make sure the *djoid* and *showoid* you are providing are correct, and if the DJ specified by *djoid* isn't a member of the show, assign them to it by going to **Show admin** -> (show name) -> **Other Dj's**

Message: *ERROR: SQL bombed on: (message)*
Meaning: There was a catastrophic error on RadioActivity.fm's end – unrelated to your request. This is most likely the message you will see if our database has had some sort of catastrophic error.
Fix: If this message appears, it isn't your fault, and the RadioActivity.fm admins will receive an automatic error warning via email.

Other issues and notes:

Question 1: Ok, so I get the gateway concept - but I'm not sure how I'm going to do this. How exactly do I get our (automation setup X) talking to the RadioActivity.fm gateway in order to put our automation plays into RadioActivity.fm?

Answer: Unfortunately, the methodology is going to vary wildly by station. It is left up to each station's techs to figure out how they're going to accomplish this.

Why? Because there are as many unique automation setups out there as there are stations. It's going to depend on

- a) What kind of live/automated mix your station has
- b) What specific kind of automation system your station uses (Winamp? iTunes? BSI? megaSeg? RCS? Operating System? etc.)
- c) The format, location, and accessibility of your automation's logfiles.

For example:

Consider a station that had live DJ's from 9am-midnight, and a MegaSeg automation system running from midnight to 9 am. They have MegaSeg running on an OS X machine connected to their campus internet.

The train of thought is as follows:

- MegaSeg keeps logfiles on the local (OS X machine).
- OS X also supports a wide variety of scriptable commandline programming.
- The station, then, is tasked with writing a script that will, from midnight to 9am,
 - parse these MegaSeg logfiles, perhaps every 5 minutes
 - weed out any entries already sent to RadioActivity.fm gateway
 - send a new request to the RadioActivity.fm gateway

While this sounds daunting, RadioActivity.fm is currently working to integrate this exact kind of setup, and will be releasing scripts for doing so once they have been thoroughly tested.

Question 2: Is secure connection (https://) supported on the playlist gateway?

Answer: No. At the time of this manual's writing, secure (https:) connections are not supported on the playlist gateway.

Question 3: Can we change or delete any entries using the playlistgateway?

Answer: No. Right now the gateway only allows 'inserting' data into your playlists – and not updating or deleting existing data.

Yes, this means that any playlist edits or deletes on the plays coming in via the gateway must be made by logging into the RadioActivity.fm admin back-end. Yes, this is admittedly a little annoying, especially when you're doing beta testing.

The idea of runaway script or malicious hacker being able to alter or delete station playlist information via the gateway, however, vastly outweighs the inconvenience of having to log in to make these changes.

Question 4: Our automation seems to record time, artist, album, and label ... but how would we go about adding in playtype and genre to the files somehow in order to pass these onto RadioActivity.fm?

Answer: While this question isn't completely answered as of yet, we're investigating using extra ID3 fields on the music files to store playtype and genre.