

# Trunking Recorder

## Call Upload API

Version: 3.5

Recorded calls and the associated metadata can be uploaded to Trunking Recorder through the web interface using an API. Uploaded calls can then be accessed through the standard web interface like locally recorded calls.

To setup and enable the Trunking Recorder Call Upload features see

<https://www.scannerbox.us/TrunkingRecorder/CallUploads/>

Trunking Recorder Call Upload API offers basic duplicate call detection by checking if the uploaded call matches one of the last 100 previously recorded or uploaded calls. If a match is found the API will return a duplicate call response. **Note:** Duplicate call detection uses a margin of error system to allow for slight variations of Call Times and Call Length so it may allow/block some unexpected calls.

The Trunking Recorder Call Upload API requires that you first upload the call metadata before sending the call audio file. This allows the server to check for a duplicate call and avoids wasting time and resources on uploading the call audio file if it already exists on the Trunking Recorder server.

### Call Upload API metadata request

To send a new call to the Trunking Recorder Call Upload API you first send a HTTP POST request to “http://trunkingrecserver/api/callupload” (Replace “http://trunkingrecserver” with the appropriate values for the Trunking Recorder server you are uploading too.

Example: <http://192.168.1.100/api/callupload>

The request should have a “Content-Type” header set to “application/json”.

It is also recommended to set the “User-Agent” header to a unique value to help identify your uploading application. This will make it easier to distinguish your application uploads from standard Trunking Recorder uploads.

The body of the request should contain “json” formatted data that includes the call metadata in the following format.

```
{
  "apiAuthID": "P25 Upload",
  "apiKey": "1Bkd498nkj23",
  "callAudioFormat": "mp3",
  "recordedCall": {
    "callText": "Attention all units, reports of an accident at Main St and Maple St",
    "talkGroupInfo": {
      "callTargets": [
        {
          "targetid": "1234",
          "targetlabel": "Police Dispatch North",

```

```

        "targettag": "P911N"
    },
    {
        "targetid": "1235",
        "targetlabel": "Police Dispatch South",
        "targettag": "P911S"
    }
],
"receiver": "RTLVoice1",
"receiverVCO": 0,
"frequency": "858.23750",
"sourceid": "1234567",
"sourcelabel": "City Car 1",
"sourcetag": "CCar1",
"lcn": "1217",
"voiceservice": "P25P1",
"systemid": "92493796",
"systemlabel": "MPSCS",
"systemtype": "APCOP25",
"siteid": "001-010",
"sitelabel": "County Simulcast",
"calltype": "2"
},
"startTime": "2021-05-16T20:50:50.2209511Z",
"callDuration": 2.9,
"startPositionSec": "00:00:00"
}
}

```

## Metadata fields

**apiAuthID** – Authentication ID used by API to identify the upload machine, value will be added to the receiver value stored in Trunking Recorder.

**apiKey** – Authentication Key used by API to verify the upload is authorized, must match the value set on the Trunking Recorder server.

**callAudioFormat** – File format of the recorded call audio, Trunking Recorder only supports WAV and MP3 formats.

**callText** – Transcribed text of the call, Trunking Recorder will display the text on the web interface.

**callTargets** – List of all talkgroups for call. If the call is a patch and involves multiple talkgroups then you should list all of them with the primary talkgroup being first.

**targetid** – ID value of the talkgroup.

**targetlabel** – Label value of the talkgroup, Trunking Recorder will display this value in the web interface.

**targettag** – Tag value of the talkgroup.

**receiver** – Receiver used to pickup and decode call. Trunking Recorder will display this value in the web interface. For uploaded calls the “apiAuthID” will be also be shown as part of the Receiver label.

**receiverVCO** – Used by Unitrunker v1 to identify virtual receivers using the same physical receiver. (Recommended to always set it to zero “0”)

**frequency** – Frequency of the call in megahertz (MHz).

**sourceid** – Call source ID value, normally the radio or console ID.

**sourcelabel** – Label value of the call source, Trunking Recorder will display this value in the web interface.

**sourcetag** - Tag value of the call source.

**lcn** - Logical Channel Number (LCN) of the call.

**voicesservice** – Type of voice system used by the call. Valid options are “A” - Analog, “P” - ProVoice, “1” - P25 Phase 1, “2” - P25 Phase 2”, “?” - Unknown, or “P25P1” - P25 Phase 1.

**systemid** – System ID value of the Trunked Radio system.

**systemlabel** – Label of the Trunked Radio system.

**systemtype** – Type of the Trunked Radio system. (Example: APCOP25, EDACS96, EDACS48)

**siteid** – Site ID value of the Trunked Radio system the call was recorded from.

**sitelabel** – Label of the Trunked Radio system site the call was recorded from.

**calltype** – Type of call that was recorded. Valid options are “0” – Individual, “1” – Group, “2” – Patch.

**startTime** – Time when the call first started. Time should be converted from local to UTC and formatted as an ISO 8601 date

**callDuration** – Length of the recorded call in seconds, should include tenths of a second. (Example: 2.9)

**startPositionSec** –Call offset start time if there is a delay at beginning of the recorded file. **Currently not used.**

If the API request is successful Trunking Recorder will respond with a “json” formatted message with a 200 status. If the call is not a duplicate the “Duplicate” value will be **false** and a “CallAudioID” value will be present

```
{
  "Status": 200,
  "StatusMessage": "Success",
  "Duplicate": false,
  "CallAudioID": "772e1ab3-70e4-4b7f-8232-f0045b2bedcd"
}
```

If the call is detected as a duplicate the “Duplicate” value will be **true**.

```
{
  "Status": 200,
  "StatusMessage": "Duplicate Call found, CallID:15668932",
  "Duplicate": true,
  "CallAudioID": ""
}
```

The “CallAudioID” value should be saved by the sending application since it will need to be used in the next API request to upload the recorded call audio file.

### **Call Upload API call audio upload**

To send a recorded call audio file to the Trunking Recorder Call Upload API you need to first successfully complete the Call Upload API metadata request and receive a “CallAudioID” value.

You should send a HTTP POST request to “<http://trunkingrecserver/api/callaudioupload/CallAudioID>” (Replace “<http://trunkingrecserver>” with the appropriate values for the Trunking Recorder server you are uploading too and replace the “CallAudioID” with the actual value returned in the first request.

Example: <http://192.168.1.100/api/callaudioupload/772e1ab3-70e4-4b7f-8232-f0045b2bedcd>

The request should have a “Content-Type” header set to “audio/mpeg” for MP3 files or “audio/wav” for WAV files.

The request should have a Content-Length” header set that matches the size of the audio file you will be uploading; this is normally set automatically by the application you are using to make the API request.

It is also recommended to set the “User-Agent” header to a unique value to help identify your uploading application. This will make it easier to distinguish your application uploads from standard Trunking Recorder uploads.

The body of the request should contain only the audio file contents.

If the request is successful Trunking Recorder will respond with a “json” formatted message with a 200 status. The “StatusMessage” will report that the call was archived successfully.

```
{
  "Status": 200,
  "StatusMessage": "Call archived successfully, CallID=15668940"
}
```

Once archived the call will be made available on the Trunking Recorder web interface like locally recorded calls.

## Call Upload API errors

The Call Upload API will return a different “Status” number when issues are encountered. The “StatusMessage” value should provide more detail on the error.

The Trunking Recorder log file on the server should also contain details on why the API call failed.

```
{
  "Status": 403,
  "StatusMessage": "Invalid apiKey",
  "Duplicate": false,
  "CallAudioID": ""
}
```

## Call Upload API metadata request with external Audio URL

If the Call Audio file is already available via an external web server you can send the Call MetaData with the full Call Audio URL to the Trunking Recorder Call Upload API.

The only difference is you should add the full URL for the Call audio file in the “filename” field in the body of the request.

The body of the request should contain “json” formatted data that includes the call metadata in the following format.

```
{
  "apiAuthID": "P25 Upload",
  "apiKey": "1Bkd498nkj23",
  "callAudioFormat": "mp3",
  "recordedCall": {
    "callText": "Attention all units, reports of an accident at Main St and Maple St",
    "filename": "https://www.externalWebServer.com/path/to/CallAudioFile.mp3",
    "talkGroupInfo": {
      "callTargets": [
        {
          "targetid": "1234",
          "targetlabel": "Police Dispatch North",
          "targettag": "P911N"
        },
        {
          "targetid": "1235",
          "targetlabel": "Police Dispatch South",
          "targettag": "P911S"
        }
      ]
    },
    "receiver": "RTLVoice1",
    "receiverVCO": 0,
  }
}
```

```

    "frequency": "858.23750",
    "sourceid": "1234567",
    "sourcelabel": "City Car 1",
    "sourcetag": "CCar1",
    "lcn": "1217",
    "voiceservice": "P25P1",
    "systemid": "92493796",
    "systemlabel": "MPSCS",
    "systemtype": "APCOP25",
    "siteid": "001-010",
    "sitelabel": "County Simulcast",
    "calltype": "2"
  },
  "startTime": "2021-05-16T20:50:50.2209511Z",
  "callDuration": 2.9,
  "startPositionSec": "00:00:00"
}
}

```

The Trunking Recorder server will verify that the URL starts with “http” or “https” and will ensure the URL ends with “mp3” or “wav”.

If the request is successful Trunking Recorder will respond with a “json” formatted message with a 200 status. The “StatusMessage” will report that the call was archived successfully.

```

{
  "Status": 200,
  "StatusMessage": "Call archived successfully, CallID=21149181",
  "Duplicate": false,
  "CallAudioID": ""
}

```

**Note:** The Trunking Recorder server will **not** verify if the URL is valid and works.

### **External Audio URL security settings**

If you are accessing Trunking Recorder using “HTTPS” you will also need to ensure that the External Audio URL uses “HTTPS”. If you try to use “HTTP” most browser will block access to the mixed insecure “HTTP” URL and the calls will not play.

Also, to allow access to the External Audio URL a “Content-Security-Policy” header must be modified to include the External Server.

To modify the header.

1. Shutdown the Trunking Recorder application.
2. Open the **settings.config** file located in %LOCALAPPDATA%\Trunking\_Recorder\ using **notepad.exe**

3. Locate the “<AdditionalMediaServers></AdditionalMediaServers>” line.
4. Add the Server Domain and Sub Domain that will house the External Audio files  
“<AdditionalMediaServers>\*.externalserver.com</AdditionalMediaServers>” (Wildcards are supported).
5. Save the **settings.config** file.
6. Start Trunking Recorder.
7. Refresh the Trunking Recorder web interface.

You should now be able to successfully play Calls with External Audio URL through the Trunking Recorder web interface. If the call does not play check the Web Browsers console for any error messages.