NAVIGATING BROADCASTING'S FUTURE

API Documentation

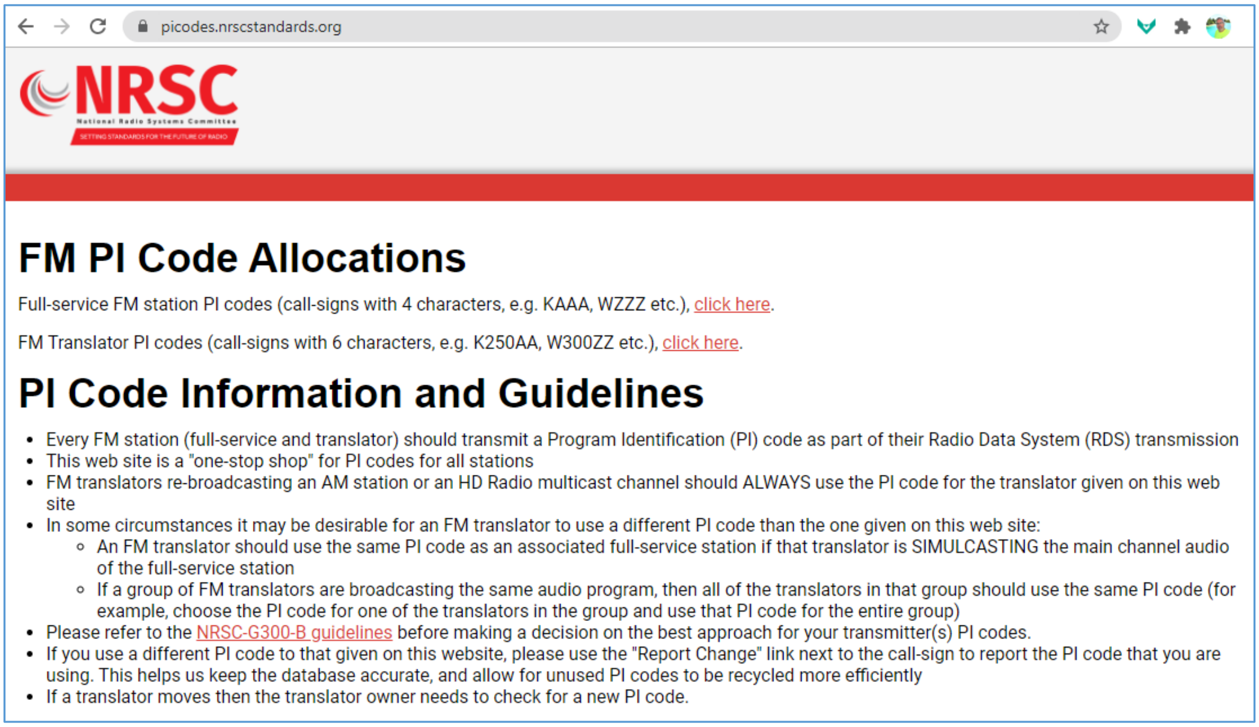
callsigns

Methods to access the NRSC PI Codes Database

V1.01 – 24th August 2020

Introduction

The NRCS PI Codes Database is a list of active FM radio licences and their assigned Radio Data System (RDS) Program Information (PI) codes.[1] PI codes for full-service licences follow the NRSC-4-B standard and PI codes for FM translators are allocated using an algorithm developed jointly by the NRSC and RadioDNS to avoid code collisions with other translators within a 400-mile radius. The NRSC PI Codes Database is the *only* authoritative source of PI codes for FM translators. The database home page is shown here in this screenshot:


picodes.nrscstandards.org
NRSC
National Radio Systems Committee
SETTING STANDARDS FOR THE FUTURE OF RADIO

FM PI Code Allocations

Full-service FM station PI codes (call-signs with 4 characters, e.g. KAAA, WZZZ etc.), click here.

FM Translator PI codes (call-signs with 6 characters, e.g. K250AA, W300ZZ etc.), click here.

PI Code Information and Guidelines

- Every FM station (full-service and translator) should transmit a Program Identification (PI) code as part of their Radio Data System (RDS) transmission
- This web site is a "one-stop shop" for PI codes for all stations
- FM translators re-broadcasting an AM station or an HD Radio multicast channel should ALWAYS use the PI code for the translator given on this web site
- In some circumstances it may be desirable for an FM translator to use a different PI code than the one given on this web site:
    - An FM translator should use the same PI code as an associated full-service station if that translator is SIMULCASTING the main channel audio of the full-service station
    - If a group of FM translators are broadcasting the same audio program, then all of the translators in that group should use the same PI code (for example, choose the PI code for one of the translators in the group and use that PI code for the entire group)
- Please refer to the NRSC-G300-B guidelines before making a decision on the best approach for your transmitter(s) PI codes.
- If you use a different PI code to that given on this website, please use the "Report Change" link next to the call-sign to report the PI code that you are using. This helps us keep the database accurate, and allow for unused PI codes to be recycled more efficiently
- If a translator moves then the translator owner needs to check for a new PI code.

The database can be browsed at **https://picodes.nrscstandards.org/** or it can be accessed programmatically with an API token. An API token can be requested using this form located on the PILOT Radio Callsigns API web page.

The PI Codes Database addresses several use cases:

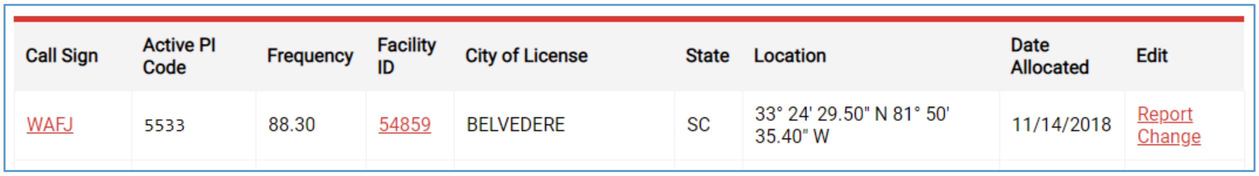
1. Broadcasters wanting to configure a correct PI code into an RDS encoder, for any kind of FM licence including translators;
2. RDS encoder manufacturers wanting to offer the correct PI code for a given call sign in their own device configuration software;
3. Developers constructing RadioDNS service information (SI) files but requiring frequency and PI code information for call signs.

[1] The National Radio Systems Committee (NRSC) is jointly sponsored by the National Association of Broadcasters (NAB) and the Consumer Technology Association (CTA). Its purpose is to study and make recommendations for technical standards that relate to radio broadcasting and the reception of radio broadcast signals.

Configuring RDS Encoders

When configuring PI code values in RDS encoders, it is strongly advised to either use the PI code value shown in the NRSC PI Codes Database or report to the database that a different PI code is in use.

When accessing the database using a browser, there is "Report Change" link (see screen shot below) to report the PI code that is in use. Reporting changed PI codes keeps the database accurate, avoids unintentional PI code collisions and (in the case of translators) allows for unused PI codes to be recycled more efficiently.



Call Sign	Active PI Code	Frequency	Facility ID	City of License	State	Location	Date Allocated	Edit
WAFJ	5533	88.30	54859	BELVEDERE	SC	33° 24' 29.50" N 81° 50' 35.40" W	11/14/2018	Report Change

Programmatic Access to the Database

There are two programmatic methods to access the database:

- Download the entire database as a flat, pipe-delimited file
- Query the database for individual call-signs using the **callsigns** API

PI Codes Database Download

The entire database can be downloaded as a single text file in UTF-8 format. Fields are delimited by pipe ("|") characters. Lines are terminated by the character `0x0A`. There is no header field.

The field order is:

call_sign – string value in the format:
^[K|W][A-Z]{3}(-LP)?(-FM\d*)?$|^[K|W][0-9]{3}[A-Z]{2}$
pi_code – string value representing a hexadecimal value in the format [0-9A-F]{4}
frequency – string value representing a decimal value in the range 88.0 to 108.0 (MHz)
facility_id – string value representing a decimal value in the format [0-9]{5}
city – string value representing the city of licence
state – string value representing the state of licence in format [A-Z]{2}
latitude – string value representing a decimal number for the latitude
longitude – string value representing a decimal number for the longitude
allocation_date – string value representing date value in the format MM/DD/YYYY for the date this pi_code was allocated to this call_sign
override – string value representing a single digit integer in the range 0|1. If 1, the pi_code was reported as different to that originally allocated.

Access is made by making a GET request to:

https://picodes.nrscstandards.org/pi_codes_allocated.dat

The client must present a valid API Authorization Token as a Header to the HTTPS request:

```
Authorization: Bearer <API Authorization Token>
```

Example requests

This request uses cURL to set the authorization header and make the request for the database file.

GET request:

```
curl -H “Authorization: Bearer {authorization token}"
https://picodes.nrscstandards.org/pi_codes_allocated.dat
```

Radio Callsigns API Functionality

The Radio Callsigns API is designed to make accessing the database easier for individual call-signs.

The API is RESTful and accessed at:

https://picodes.nrscstandards.org/callsigns

It may be called using either GET or POST methods. The parameters provided in the request are the same for both methods.

Request

The client must present a valid API Authorization Token as a Header to the HTTPS request:

```
Authorization: Bearer <API Authorization Token>
```

<table>
<tr><th>METHOD</th><th>REQUEST PARAMETERS</th></tr>
<tr><td>GET</td><td><b>format</b> – string value – zero or one instance
<b>call_sign</b> – string value in the format:
^[K|W][A-Z]{3}(-LP)?(-FM\d*)?$|^[K|W][0-9]{3}[A-Z]{2}$
– one or more instances</td></tr>
<tr><td>POST</td><td>Body is a JSON object containing:
<b>format</b> – string value – zero or one instance
<b>call_sign</b> – array of one or more string objects with values in the format:
^[K|W][A-Z]{3}(-LP)?(-FM\d*)?$|^[K|W][0-9]{3}[A-Z]{2}$
– one or more instances</td></tr>
</table>

Response 200 - OK

The request has successfully completed, and results are provided in the body. The structure of the body depends on the **format** parameter provided.

Format	Response Structure
Json or null/empty	A JSON array containing zero or more objects with properties: **call_sign** – string value matching one of the input values **frequency** – string value representing a decimal value in the range 88.0 to 108.0 (MHz) **pi_code** – string value representing a hexadecimal value in the format [0-9A-F]{4} **facility_id** – string value representing a decimal value in the format [0-9]{5} **fm_bearer_id** – string value as defined in TS 103 270 (RadioDNS Hybrid Radio Lookup) **hd_bearer_id** – string value as defined in TS 103 270 (RadioDNS Hybrid Radio Lookup)
Csv	One or more lines of text, starting with the line: `call_sign, frequency, pi_code, facility_id, fm_bearer_id, hd_bearer_id` followed by zero or more lines of values separated by commas: **call_sign** – as above **frequency** – as above **pi_code** – as above **facility_id** – as above **fm_bearer_id** – as above **hd_bearer_id** – as above

The order of responses may not match the order of parameters in the original request.

Responses must be matched to requests using **call_sign** which shall be identical to the request value if it has matched.

Where no match is found for a **call_sign**, no response shall be given.

Response 401 – Unauthorized

The Authorization header was not provided, or the API Authorization Token provided was invalid.

Response 500 – Internal Server Error

The API has encountered an error processing the entire request. The response body shall contain the detail of the error:

Format	Response Structure
Json or null/empty	**error** – a text string describing the error
Csv	Two lines of text starting with one line: `error` and then one line of text describing the error

Implementation Notes

fm_bearer_id and hd_bearer_id

The **fm_bearer_id** and **hd_bearer_id** responses are provided as a convenience for the user but are not authoritative responses. In particular:

- A transmitter may not be transmitting RDS information.
- A transmitter may be transmitting an RDS PI code that differs from that in the PI Codes Database.
- A transmitter may not be transmitting an HD Radio signal
- The ECC component of the GCC value in an **fm_bearer_id** shall always be "a0" (USA)
- The Country Code value in an **hd_bearer_id** shall always be "292" (USA)

Character Encoding

All requests and responses shall be made with character encoding set to : `utf-8`

For the format CSV, a newline shall be defined by the character `0x0A`.

Allowed Call-Signs

Call Signs must match the values in the PI Codes Database, which are of the format:

WAAA	Full service FM
WAAA-LP	Low power FM
WAAA-FMx	FM Booster (x is optional and decimal)
WAAA-LP-FMx	Lower power FM Booster (x is optional and decimal)
W1234AA	FM Translator

Example requests

The following requests use cURL to set the authorization header and make the request to the API.

GET Request for call-sign WHTZ with response in format JSON:

```
curl -H "Authorization: Bearer {authorization token}"
https://picodes.nrscstandards.org/callsigns?call_sign=WHTZ
```

POST Request for call-signs WAMU and KIIS response in the format JSON:

```
curl -X POST -H "Authorization: Bearer {authorization token}" " -d
'{"call_sign":["WAMU"],["KIIS"]}'
https://picodes.nrscstandards.org/callsigns
```

Document History

2020-06-08	Version 1.00 – Release
2020-XX-YY	Version 1.01 – Addition of database file download method