



NOAA FISHERIES

Electronic Vessel Trip Reporting (eVTR) Technical Requirements Guide

For eVTR Application Developers

Rev. 1.3

Greater Atlantic Regional Fisheries Office (GARFO)

Analysis & Program Support Division (APSD)

Gloucester, MA

1/12/2021

Table of Contents

1	Introduction	5
1.1	Audience	5
1.2	Product Support.....	5
1.3	Note Concerning Revision History	5
2	eVTR Composition.....	6
2.1	How GARFO Organizes eVTR Information	6
2.2	Subtrips	7
2.3	eVTR Unique Trip Identifier (UTID).....	7
2.4	File Format	8
2.5	Electronic Signature	9
2.6	File Transfer	9
2.7	Error Checking.....	9
2.8	eVTR User Interface Requirements	10
3	JSON API Requirements	11
3.1	JSON API Generic Sequence.....	11
3.2	API Authentication Protocol	12
3.3	Send eVTR Form to GARFO	14
4	eVTR Reference.....	18
4.1	Support Tables	18
4.2	eVTR Form Specification	19
4.3	Mandatory Use of a Field in eVTR Based on Trip Activity Type	21
4.4	Gear Code and Soak Time Interdependence	23
4.5	Quality Control Audit Rules	27
4.6	Regulatory References.....	32
4.7	Document Revision History	32

1 Introduction

This guide provides the technical information that you need to satisfy the programming requirements for electronic reporting. The guide comprises three principal sections:

- The basics of GARFO's Electronic Vessel Trip Report (eVTR)
- Application Programming Interface (API) requirements for software application developers

NOTE: The Federal Register outlines the regulatory requirements that eVTR software must meet. See the proper references in the section [Regulatory References](#) in this guide.

- Reference information

Use the links in the Table of Contents or the links embedded throughout this guide to quickly navigate to the information you want.

1.1 Audience

This guide is intended for programmers whose expertise ranges from intermediate to advanced. This guide is not intended for novice programmers who do not have a working knowledge of APIs. Novice programmers should refer to API tutorials. More experienced users should feel free to use the Table of Contents in order to look up specific information as needed.

1.2 Product Support

GARFO must endorse your product's adherence to regulatory and technical requirements. To request assistance, contact the IT Service Desk by phone or by e-mail:

- Phone: 978-281-9188
- E-mail: nmfs.gar.helpdesk@noaa.gov

1.3 Note Concerning Revision History

When GARFO changes eVTR technical requirements, we also update the information in this guide to reflect those changes. To obtain important information about updates to the technical requirements at this release, or to obtain information about updates at previous releases, please consult the section [Document Revision History](#) elsewhere in this guide.

2 eVTR Composition

The fishing vessel's owner/operator files a Vessel Trip Report (VTR) after a fishing trip. If the owner/operator files the report electronically, the report is called an eVTR. The report includes but is not limited to information about the trip duration, the fishing effort expended (gear used, location, and duration fished), the species that were caught, and their landed weight. VTR data are the definitive source of fishing effort information (gear and area) which is a critical input to ensure accurate catch monitoring.

Information in this section includes the following topics:

- [How GARFO Organizes eVTR Information](#)
- [Subtrips](#)
- [eVTR Unique Trip Identifier \(UTID\)](#)
- [File Format](#)
- [Electronic Signature](#)
- [File Transfer](#)
- [Error Checking](#)
- [eVTR User Interface Requirements](#)

2.1 How GARFO Organizes eVTR Information

GARFO internally organizes the fishing trip data in an eVTR as follows:

1) Trip

Identifies the vessel, trip type (commercial, party, charter or recreational), number of crew/anglers and the dates and times for when it departed and returned to port (sailed and landed respectively).

2) Effort

Identifies subtrip information (see the section [Subtrips](#)). This portion of the VTR captures the location where the vessel fished, gear information, haul/soak time, number of hauls/sets/strings, etc.

3) Catch

Identifies the species and the amount of the catch (*kept* and *discarded*) for each subtrip.

4) Sales

Identifies the dealer(s) that bought the catch (or the ultimate disposition of the catch), and also the date and port of the offload associated with each subtrip.

5) Signature

Provides the verification by the vessel owner / operator that the information being submitted on the report is correct.

[Jump to: Table of Contents](#)

2.2 Subtrips

A subtrip represents the effort and catch associated with a unique combination of the following:

- Chart area
- Gear type
- Mesh size

If any of these elements change during the course of a trip, a new subtrip starts, and the vessel owner/operator must record additional catch and effort information.

A well-designed eVTR program or application has the ability to significantly reduce the amount of time required for a vessel operator to comply with their VTR reporting requirements by eliminating the need to fill out information that stays the same throughout the trip (for example, vessel permit, date sail and land, trip type).

This means that the application programmer must code the application so that the user supplies data for the fields in the Trip and Signature information only once, since these fields apply to the entire trip.

2.3 eVTR Unique Trip Identifier (UTID)

The eVTR application must build and generate a unique trip identifier (UTID) that serves as the key by which all fisheries data systems are referenced from a specific fishing trip.

2.3.1 UTID Composition

The eVTR application must build the UTID from the following numeric data:

- Vessel Permit Number (6 digits) as listed on the vessel's Federal fishing permit
The Vessel Permit Number is also referred to as the VPN.
- Year (2 digits)
- Month (2 digits)
- Day (2 digits)
- Hour (2 digits, based upon 24 hour clock / military time)

[Jump to: Table of Contents](#)

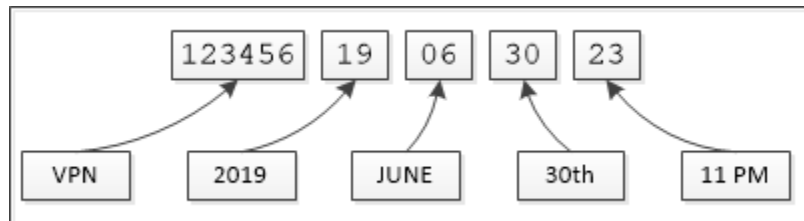
2.3.2 UTID Example

A single numeric string constitutes the eVTR UTID, such as the following example:

12345619063023

[Figure 1](#) shows how to read the eVTR UTID:

Figure 1: Reading the eVTR UTID



NOTE: The year, month, day and hour portion of the eVTR UTID represent the time at which the trip report was initiated.

2.3.3 UTID Usage Notes

Be aware of the following information in accounting for the eVTR UTID:

- Once the eVTR application generates the value of the eVTR UTID, the value cannot change.
- The eVTR UTID shall be readily available to the vessel owner/operator. The UTID:
 - Must be available early in the trip, before the Trip Start Hail.
 - Is required by the seafood dealer before the catch can be purchased from the vessel.
- If two duplicate trip IDs are created by starting trips in rapid succession, it is acceptable to increment the last two digits.
- NMFS uses the eVTR UTID to link a fishing trip to other Fisheries Dependent Data (FDD) systems that collect and process data, such as Dealer reports and Observer reports.

2.4 File Format

An eVTR application must fulfill the following file format requirements:

- 1) Produce a data file in the specified standard format.

Such a format allows a successful file upload into the GARFO databases. Find the valid file format in the section [eVTR Form Specification](#) in this guide.

- 2) Meet GARFO data collection requirements.

GARFO provides support tables for standard industry information that a programmer needs to build into the application. Find a list of the valid support tables in the section labeled [Support Tables](#) in this guide.

NOTE: A value of **NA** in a table cell indicates that there is no value or instruction from GARFO concerning that particular instance.

[Jump to: Table of Contents](#)

2.5 *Electronic Signature*

Immediately upon completion of a trip report, the vessel owner/operator must have the means to digitally sign the eVTR. The eVTR application must provide the following for electronic signature:

- A way for the vessel owner/operator to verify the operator name and Vessel Permit Number (VPN).
- The display of the following text during the signing ceremony:

```
"I certify that the vessel trip information I am providing is true, complete and correct to the best of my knowledge, and made in good faith. Making a false statement is punishable by law (18 U.S.C. 1001)."
```

2.6 *File Transfer*

GARFO requires that you submit trip report files using the HTTPS POST method. Use the API URL to submit the eVTR. For information, refer to the section [JSON API Requirements](#) in this guide.

2.7 *Error Checking*

When the vessel owner/operator submits a trip report to GARFO, the audit engine checks the report against a set of audit rules. The audit engine flags any eVTR that fails one or more audit rules. GARFO subsequently instructs the vessel owner/operator to first correct, then resubmit the report.

GARFO requires your application to contain the same error checking logic as the GARFO QC checks in order to prevent the rejection of the report. Find a list of the GARFO QC audits in the section [Quality Control Audit Rules](#) later in this guide.

2.7.1 *Corrections to a Previously Submitted eVTR*

GARFO can identify most syntax errors in a submitted eVTR before processing that eVTR. However, GARFO can only identify other types of errors, such as discrepancies between VTR and dealer data sets, after processing the eVTR in the audit rules engine.

If the audit returns one or more errors against the submission, GARFO notifies the vessel owner/operator of the problem(s) so that they may correct the eVTR, and then resubmit. This action requires that eVTR vendors provide a method for copying and then modifying, sometimes significantly, a previously submitted eVTR. The vessel owner/operator must then have the ability to submit the amended eVTR in the same manner as the original submission.

CAUTION: The vessel owner/operator *must* use the trip identifier from the original submission when submitting the revised eVTR.

[Jump to: Table of Contents](#)

2.8 *eVTR User Interface Requirements*

The eVTR application shall adhere to the following requirements for user interface design:

- Record fishing data accurately as entered.
- Transmit data to GARFO that is an accurate representation of what was reported by the vessel owner/operator.
- Store data entered by owners/operators even if no internet connection is available.
- Enforce the submission of a valid value for each data element required on a particular fishing trip.
- Allow the owner/operator to enter, choose, or confirm all data elements.
- Prevent data element auto-population, or default value generation, without confirmation.
- Generate a unique trip identifier (UTID) as soon as a new trip report is initiated.
- Provide the ability to display all previously submitted reports submitted on that device.

When requested by authorized personnel, eVTR applications must have the capability to display a facsimile of the paper VTR form, including a separate 'page' for each subtrip. Ideally, the eVTR software should enable print, export, or email VTR facsimiles.

- Allow the manual entry of latitude/longitude coordinates while also optionally supporting the coordinates to be automatically populated from the device or vessel's GPS system.

[Jump to: Table of Contents](#)

3 JSON API Requirements

GARFO uses a JavaScript Object Notation (JSON) based API for 3rd-party communications with the eVTR server. JSON is a lightweight, open standard data exchange format, uses a natural language coding scheme, and can work with any framework or platform. Topics in this section include:

- [JSON API Generic Sequence](#)
- [API Authentication Protocol](#)
- [Send eVTR Form to GARFO](#)

3.1 JSON API Generic Sequence

Data is passed and returned in JSON format over the HTTPS protocol by means of requests and responses. You code a request to obtain a resource that's stored on a remote server, or to transmit data to a remote server, and the remote server responds with the requested information or message. Each action requires a separate request and response.

The following sequence describes a generic request and return exchange between the 3rd-party application and the database server:

- 1) A 3rd-party application submits an authentication request (if needed) from the device to the API by means of a URL.
- 2) The application receives a server STATUS message as a JSON object.
If authentication is successful, that response will include a token string.
- 3) The bundle of the request (for example, an eVTR form) is passed in the POST header of the HTTPS request using the JSON object format.
- 4) The API returns the requested data or response as a JSON object.

The return includes a STATUS indicator.

3.1.1 Base URL

The base URL link is as follows:

https://apps-garfo.fisheries.noaa.gov/base_api/

This URL forms the basis of all requests to the eVTR API module. Create specific requests by appending request strings on this URL.

3.1.2 JSON API Usage Notes

Be aware of the following before you begin to use the GARFO eVTR API module:

- All errors return a STATUS as a JSON object that contains a description of the error.
See the section named [Quality Control Audit Rules](#) for a list of valid codes.

[Jump to: Table of Contents](#)

3.2 API Authentication Protocol

This section describes how to properly submit an authentication request to the system.

3.2.1 Prerequisite: Obtain Valid Username and Password

A Federally-permitted vessel owner / operator must obtain a valid username / password combination from the IT Service Desk **before** being able to submit a VTR electronically:

- IT Service Desk phone: 978-281-9188
- IT Service Desk e-mail: nmfs.gar.helpdesk@noaa.gov

3.2.2 Authentication

The API authenticates the user's credentials and returns an alphanumeric token that represents the validation of credentials. The token is valid for 45 minutes after it is issued, and may be included in a VTR within that time period.

- The authentication values are in the submitted JSON.
- The order of the fields does not matter.

```
{
  "authorization":
  {
    "username": "Thetis",
    "password": "Thetis_password",
    "vendor_id": "1234567",
    "vendor_software_version": "1.1.0.1",
    "vendor_app_name": "ReallyNeat FishingApp",
    "vendor_app_platform": "ANDROID"
  }
}
```

3.2.2.1 App Platform

The field `vendor_app_platform` key should describe the operating system or environment that the app is running on. This might be used to identify and remedy an issue in a certain version of an app, or if the codebase is shared between platforms, it is not necessary to identify the platform to fill out this field. For example, a JavaScript app that runs on multiple platforms could use `BROWSER`. Recommended values:

- BROWSER
- IOS
- IPAD
- IPHONE
- MAC CATALYST
- MAC OS
- ANDROID
- WINDOWS

Or any value deemed reasonable for your application.

[Jump to: Table of Contents](#)

3.2.3 Returns from Authentication Request

Authentication attempt returns one of two possible results.

- Return from correct authorization
- Return from incorrect authorization

3.2.3.1 Return from Correct Authorization

Following is a representation of the return from a correct authorization request. The values themselves are for example purposes only.

```
{
  "user_app_status": "TEST",
  "user_full_name": "user_full_name",
  "Status": "Logged In",
  "token": "a_very_long_cryptic_string",
  "token_expires": 1609941713,
  "client_id": "id of user",
  "operator_key": "operator key"
  "vtr_preferences": [],
  "broadcast_message": [],
  "entitlements": [
    {
      "vessel": "555555"
    },
    {
      "vessel": "555556"
    }
  ]
}
```

3.2.3.2 Return from Incorrect Authorization

Following is an example of the return from an incorrect authorization request.

```
{
  "Status": "Logged Out"
}
```

[Jump to: Table of Contents](#)

3.3 Send eVTR Form to GARFO

This call submits an eVTR form for a logged in, authenticated user to the GARFO eVTR database.

3.3.1 Request Formats

Following are the two possible JSON formats for the eVTR that the GARFO API accepts. Refer to both formats in this section.

- Catch and Sales data as one array
- Catch and Sales data as separate arrays

The values in the examples in this section are placeholders only. The rules for all of the fields are the same. The order of the fields is your choice.

3.3.2 Endpoints

Following are the supported endpoints for each submission request. Be aware that adding the string `test` to the end of either URL directs the request to a test system, and does not affect the vessel owner/operator's reporting compliance in any way. Use the test endpoint to verify your code.

WARNING: *DO NOT* submit test data to the production endpoint!

- Endpoints for Catch and Sales data as one array:
 - https://apps-garfo.fisheries.noaa.gov/base_api/evtr/v_1 (Production)
 - https://apps-garfo.fisheries.noaa.gov/base_api/evtr/v_1/test (Test)
- Endpoints for Catch and Sales data as separate arrays:
 - https://apps-garfo.fisheries.noaa.gov/base_api/evtr/v_1_sales (Production)
 - https://apps-garfo.fisheries.noaa.gov/base_api/evtr/v_1_sales/test (Test)

[Jump to: Table of Contents](#)

3.3.3 JSON Format: Catch and Sales Data Together (Test)

Use the following JSON code block to test the approach of submitting catch and sales data together.

WARNING: *DO NOT* submit test data to the production endpoint!

3.3.3.1 URL

https://apps-garfo.fisheries.noaa.gov/base_api/evtr/v_1/test

3.3.3.2 Description

The user designs the eVTR submit request so that Catch data link directly to the corresponding Sales data. That is, each subtrip includes the disposition of each species caught as part of the subtrip record.

```
{
  "vendor_id": 22222,
  "vendor_software_version": "0.0.0",
  "vendor_app_name": "ReallyNeat FishingApp",
  "vendor_app_platform": "Android",
  "token": "ABC123ABC123ABC123ABC123",
  "client_id": 99999,
  "vendor_message_id": "",
  "trip_id": 11111114091913,
  "vessel_permit_num": 111111,
  "vessel_registration": "11",
  "date_sail": 1409190739,
  "date_land": 1409190739,
  "trip_type": 1,
  "number_of_crew": 10,
  "number_of_anglers": null,
  "operator_first_name": "test",
  "operator_last_name": "operator",
  "operator_permit_number": 90000001,
  "no_effort_trip_reason": 0,
  "trip_comments": "",
  "subtrips": [{
    "vtr_serial_num": 12345619063023,
    "gearcode": "403LLB",
    "mesh_size": "6",
    "gearqty": "120",
    "gearsizes": "1.2",
    "latitude_or_loran1": "42:17",
    "longitude_or_loran2": "-068:51",
    "chart_area": 521,
    "fishing_depth": 50,
    "fishing_depth_uom": "fa",
    "number_of_hauls": 4,
    "tow_soak_time": "5:30",
    "datetime_haul_start": 1409190739,
```

```

    "datetime_haul_end": 1409190739,
    "catch": [{
      "species": "cod",
      "kept": 1000,
      "discarded": 100,
      "uom": "pounds",
      "dealer_permit_number": 1621,
      "dealer_name": "a & a seafood inc",
      "date_sold": 130403,
      "offloading_port": 220101
    }]
  }]
}

```

3.3.4 JSON Format: Separate Arrays for Catch and Sales Data (Test)

Use the following JSON code block to test the approach of submitting catch and sales data together.

WARNING: DO NOT submit test data to the production endpoint!

3.3.4.1 URL

https://apps-garfo.fisheries.noaa.gov/base_api/evtr/v_1_sales/test

3.3.4.2 Description

The user designs the eVTR submit request so that Sales and Catch are self-contained units of data. The separate arrays method means that you don't have to link Sales to a specific Effort.

CAUTION: The total of the *kept* value **must equal** the total *sold* value for the entire trip.

```

{
  "vendor_id": 22222,
  "vendor_software_version": "0.0.0",
  "vendor_app_name": "ReallyNeat FishingApp",
  "vendor_app_platform": "Android",
  "token": "ABC123ABC123ABC123ABC123",
  "client_id": 99999,
  "vendor_message_id": "",
  "trip_id": 11111114091913,
  "vessel_permit_num": 111111,
  "vessel_registration": "11",
  "date_sail": 1409190739,
  "date_land": 1409190739,
  "trip_type": 1,
  "number_of_crew": 10,
  "number_of_anglers": null,
  "operator_first_name": "test",
  "operator_last_name": "operator",
  "operator_permit_number": 90000001,
  "no_effort_trip_reason": 0,

```



```
"trip_comments": "",
"subtrips": [{
  "vtr_serial_num":12345619063023,
  "gearcode": "403LLB",
  "mesh_size": "6",
  "gearqty": "120",
  "gearsizes": "1.2",
  "latitude_or_loran1": "42:17",
  "longitude_or_loran2": "-068:51",
  "chart_area": 521,
  "fishing_depth": 50,
  "fishing_depth_uom": "fa",
  "number_of_hauls": 4,
  "tow_soak_time": "5:30",
  "datetime_haul_start": 1409190739,
  "datetime_haul_end": 1409190739,
  "catch": [{
    "species": "cod",
    "kept": 1000,
    "discarded": 100,
    "uom": "pounds"
  }]
}],
"sales": [{
  "species": "cod",
  "sold": 1000,
  "uom": "pounds",
  "dealer_permit_number": 1621,
  "dealer_name": "a & a seafood inc",
  "date_sold": 130403,
  "offloading_port": 220101
}]
```

[Jump to: Table of Contents](#)

4 eVTR Reference

Use this section to obtain reference information on eVTR format and content. Topics include:

- [Support Tables](#)
- [eVTR Form Specification](#)
- [Quality Control Audit Rules](#)
- [Mandatory Use of a Field in eVTR Based on Trip Activity Type](#)
- [Gear Code and Soak Time Interdependence](#)
- [Regulatory References](#)

NOTE: A value of **NA** in a table cell indicates that there is no value or instruction from GARFO concerning that particular instance.

4.1 Support Tables

The eVTR Support Tables provide standard industry information that you need to build your application. Access the support tables at the following link: [Vessel Trip Reporting in the Greater Atlantic Region](#).

You may also use the following list to find links to individual support tables.

- Link to: [eVTR Dealer Listing](#)
Provides dealer permit number, dealer name, port, and state along with dealer codes for Home Consumption, Bait, etc.
- Link to: [eVTR Gear Codes](#)
Provides valid gear codes along with additional information that can be used for range checking values such as mesh size, quantity, soak time etc. associated with a given gear code.
- Link to: [eVTR Location to Area](#)
Provides for the conversion of a specified latitude, longitude and a ten minute square into a chart area
- Link to: [eVTR Port Listing](#)
Provides the port code for a specified port and state combination.
- Link to: [eVTR Species Codes](#)
Provides synonym(s), species code and common name for species in GARFO. The synonyms provide for lookups using colloquial names. For example, DAB = American Plaice Flounder
- Link to: [eVTR Trip Activity Type](#)
Subcategories used to refine a trip type.
- Link to: [eVTR Trip Types](#)
Valid trip types

[Jump to: Table of Contents](#)

4.2 eVTR Form Specification

The information presented in [Table 1](#) provides the data specifications (such as data type and numeric precision) for each field used in the eVTR form.

Table 1: Data Formats for eVTR Form

FIELD NAME	TYPE	LENGTH OR PRECISION	PARENT	NULLABLE	FORMAT
<i>vendor_id</i>	INTEGER	NA	NA	NO	NA
<i>vendor_software_version</i>	VARCHAR	NA	NA	NO	NA
<i>vendor_app_platform</i>	VARCHAR	NA			
<i>token</i>	VARCHAR	500	NA	YES	NA
<i>client_id</i>	NUMBER	10	NA	NO	NA
<i>vendor_message_id</i>	INTEGER	150	NA	YES	NA
<i>trip_id</i>	INTEGER	14	NA	NO	Exactly 14 digits
<i>vessel_registration</i>	VARCHAR	10	NA	NO	NA
<i>date_sail</i>	DATE	12	NA	NO	YYMMDDHH24MI
<i>date_land</i>	DATE	12	NA	NO	YYMMDDHH24MI
<i>trip_type</i>	INTEGER	1	NA	NO	NA
<i>number_of_crew</i>	INTEGER	2	NA	NO	NA
<i>vessel_permit_num</i>	INTEGER	6	NA	NO	Exactly 6 digits
<i>number_of_anglers</i>	INTEGER	3	NA	YES	NA
<i>operator_first_name</i>	VARCHAR	17	NA	YES	NA
<i>operator_last_name</i>	VARCHAR	17	NA	YES	NA
<i>operator_permit_number</i>	INTEGER	8	NA	YES	NA
<i>no_effort_trip_reason</i>	INTEGER	1	NA	NO	NA
<i>trip_comments</i>	VARCHAR	150	NA	YES	NA
<i>subtrips</i>	Array	NA	NA	YES	NA
<i>gearcode</i>	VARCHAR	3	subtrips	NO	Exactly 3 characters
<i>vtr_serial_num</i>	INTEGER	NA	subtrips	YES	NA
<i>mesh_size</i>	NUMBER	6.1	subtrips	YES	NA
<i>gearqty</i>	NUMBER	5	subtrips	YES	NA
<i>gearsizes</i>	NUMBER	6.1	subtrips	YES	NA

FIELD NAME	TYPE	LENGTH OR PRECISION	PARENT	NULLABLE	FORMAT
<i>latitude_or_loran1</i>	VARCHAR	8	subtrips	NO	(-)DD.DDDD or (-)DD:MM
<i>longitude_or_loran2</i>	VARCHAR	9	subtrips	NO	(-)DDD.DDDD or (-)DDD:MM
<i>chart_area</i>	INTEGER	3	subtrips	NO	NA
<i>fishing_depth</i>	INTEGER	4	subtrips	NO	NA
<i>fishing_depth_uom</i>	VARCHAR	3	subtrips	NO	NA
<i>number_of_hauls</i>	INTEGER	4	subtrips	YES	NA
<i>tow_soak_time</i>	VARCHAR	7	subtrips	YES	9999:59
<i>catch</i>	Array	NA	NA	YES	NA
<i>species</i>	VARCHAR	10	catch	NO	NA
<i>kept</i>	INTEGER	8	catch	YES	NA
<i>discarded</i>	INTEGER	8	catch	YES	NA
<i>uom</i>	VARCHAR	3	catch	YES	NA
<i>dealer_permit_number</i>	INTEGER	5	catch	YES	NA
<i>dealer_name</i>	VARCHAR	40	catch	YES	NA
<i>date_sold</i>	DATE	6	catch	YES	YYMMDD
<i>offloading_port</i>	INTEGER	6	catch	YES	NA

[Jump to: Table of Contents](#)

4.3 Mandatory Use of a Field in eVTR Based on Trip Activity Type

[Table 2](#) lists each field in the eVTR. An **X** indicates that the field must have a provided value, based on the trip activity type, as presented in the eVTR Trip Activity Types support table.

NOTE: A value of NA in a table cell indicates that there is no value or instruction from GARFO concerning that particular instance.

Table 2: Reporting Fields Based on Trip Activity Type

Field Name	MANDATORY FIELDS FOR TRIP ACTIVITY TYPE 0 (FISHING TRIP WITH EFFORT(S))	MANDATORY FIELDS FOR TRIP ACTIVITY TYPES 1-8
<i>trip_id</i>	X	X
<i>vessel_permit_num</i>	X	X
<i>vessel_registration</i>	X	X
<i>date_sail</i>	X	X
<i>date_land</i>	X	X
<i>trip_type</i>	X	X
<i>number_of_crew</i>	X	X
<i>number_of_anglers</i>	Mandatory when <i>trip_type</i> reported as 2 (party), 3 (charter) or 6 (private/recreational)	Mandatory when <i>trip_type</i> reported as 2 (party), 3 (charter) or 6 (private/recreational)
<i>operator_name</i>	X	X
<i>operator_permit_number</i>	NA	NA
<i>no_effort_trip_reason</i> (Trip Activity Type)	X	X
<i>gearcode</i>	X	NA
<i>mesh_size</i>	Mandatory depending on <i>gear_code</i> reported. See Table 3 in this guide.	NA
<i>gearqty</i>	Mandatory depending on <i>gear_code</i> reported. See Table 3 in this guide.	NA
<i>gearsizes</i>	Mandatory depending on <i>gear_code</i> reported. See Table 3 in this guide.	NA
<i>latitude_or_loran1</i>	X	NA
<i>longitude_or_loran2</i>	X	NA
<i>chart_area</i>	X	NA
<i>fishing_depth</i> (UOM = Fathoms)	X	NA
<i>number_of_hauls</i>	Mandatory depending on <i>gear_code</i> reported. See Table 3 in this guide.	NA

Field Name	MANDATORY FIELDS FOR TRIP ACTIVITY TYPE 0 (FISHING TRIP WITH EFFORT(S))	MANDATORY FIELDS FOR TRIP ACTIVITY TYPES 1-8
<i>tow_soak_time</i>	Mandatory but <i>tow_soak_time</i> reported may or may not exceed trip duration, depending upon <i>gear_code</i> reported. See Table 3 for requirements.	NA
<i>species</i>	X	NA
<i>kept</i>	Mandatory unless <i>species</i> reported is NC (no catch). If <i>species</i> value equals NC, <i>kept</i> can be NULL or 0.	NA
<i>discarded</i>	Mandatory when <i>kept</i> is NULL or 0 and <i>species</i> code is not NC (no catch). If <i>kept</i> is 1 or greater, the field can still be populated by the user.	NA
<i>uom</i>	Mandatory; <i>uom</i> for <i>trip_type</i> 1 = pounds, <i>uom</i> for <i>trip_type</i> 2, 3 and 6 = count.	NA
<i>dealer_permit_number</i>	Mandatory unless the value reported for <i>kept</i> is NULL or 0.	NA
<i>dealer_name</i>	Mandatory unless the value reported for <i>kept</i> is NULL or 0.	NA
<i>date_sold</i>	Mandatory unless the value reported for <i>dealer_permit_number</i> is 99998, 1, 4, or 8.	NA
<i>offloading_port</i>	X	NA

[Jump to: Table of Contents](#)

4.4 Gear Code and Soak Time Interdependence

[Table 3](#) shows how the validity of the soak time value depends upon the gear code that was reported.

Table 3: Gear Code and Soak Time Interdependence

GEAR CODE	GEAR NAME	GEAR CODE DEPENDENT MESH_SIZE	GEAR CODE DEPENDENT GEARQTY	GEAR CODE DEPENDENT GEARSIZE	GEAR CODE DEPENDENT NUMBER OF HAULS	Soak Time Evaluation (RULE ID 169)
CST	CASTNET	X	NA	X	X	<i>tow_soak_time</i> cannot exceed time at sea
DIP	"DIP NET, COMMON"	X	X	X	X	<i>tow_soak_time</i> cannot exceed time at sea
DIV	DIVING GEAR	NA	NA	NA	NA	<i>tow_soak_time</i> cannot exceed time at sea
DRM	"DREDGE, MUSSEL"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DRC	"DREDGE, OCEAN QUAHOG/SURF CLAM"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DRO	"DREDGE, OTHER"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DRS	"DREDGE, SCALLOP,SEA"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DSC	"DREDGE, SCALLOP-CHAIN MAT"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DRU	"DREDGE, URCHIN"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DTC	"DREDGE,SCALLOP,CHAIN MAT,MOD"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
DTS	"DREDGE,SCALLOP,TURTLE DEFLECT"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
FYK	FYKE NET	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
GND	"GILL NET, DRIFT,LARGE MESH"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea

GEAR CODE	GEAR NAME	GEAR CODE DEPENDENT MESH_SIZE	GEAR CODE DEPENDENT GEARQTY	GEAR CODE DEPENDENT GEARSIZE	GEAR CODE DEPENDENT NUMBER OF HAULS	Soak Time Evaluation (RULE ID 169)
GNT	"GILL NET, DRIFT,SMALL MESH"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
GNO	"GILL NET, OTHER"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
GNR	"GILL NET, RUNAROUND"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
GNS	"GILL NET, SINK"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
HND	HAND LINE/ROD & REEL	NA	X	NA	NA	<i>tow_soak_time</i> cannot exceed time at sea
RAK	HAND RAKE	NA	NA	NA	NA	<i>tow_soak_time</i> cannot exceed time at sea
HRP	HARPOON	NA	X	NA	Mandatory unless value reported for <i>species</i> is NC (no catch)	<i>tow_soak_time</i> cannot exceed time at sea
LLB	"LONGLINE, BOTTOM"	NA	X	X	X	<i>tow_soak_time</i> can exceed time at sea
LLP	"LONGLINE, PELAGIC"	NA	X	X	X	<i>tow_soak_time</i> can exceed time at sea
OTB	"OTTER TRAWL, BEAM"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
OTF	"OTTER TRAWL, BOTTOM,FISH"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
OTO	"OTTER TRAWL, BOTTOM,OTHER"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
OTC	"OTTER TRAWL, BOTTOM,SCALLOP"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
OTS	"OTTER TRAWL, BOTTOM,SHRIMP"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea
OHS	"OTTER TRAWL, HADDOCK SEPARATOR"	X	X	X	X	<i>tow_soak_time</i> * <i>number_of_hauls</i> cannot exceed time at sea

GEAR CODE	GEAR NAME	GEAR CODE DEPENDENT MESH_SIZE	GEAR CODE DEPENDENT GEARQTY	GEAR CODE DEPENDENT GEARSIZE	GEAR CODE DEPENDENT NUMBER OF HAULS	Soak Time Evaluation (RULE ID 169)
OTM	"OTTER TRAWL, MIDWATER"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
OTR	"OTTER TRAWL, RUHLE"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
TTS	"OTTER TRAWL,BOTTOM,SHRIMP,TWIN"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
OTT	"OTTER TRAWL,BOTTOM,TWIN"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
OBP	"OTTER TRAWL,LGMESH BELLY PANEL"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
PTB	"PAIR TRAWL, BOTTOM"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
PTM	"PAIR TRAWL, MIDWATER"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
PTW	"POT, CONCH/WHELK"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTC	"POT, CRAB"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTE	"POT, EEL"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTF	"POT, FISH"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTH	"POT, HAG"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTL	"POT, LOBSTER"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTO	"POT, OTHER"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTS	"POT, SHRIMP"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
PTX	"POTS, MIXED"	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
SED	"SEINE, DANISH"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea

GEAR CODE	GEAR NAME	GEAR CODE DEPENDENT MESH_SIZE	GEAR CODE DEPENDENT GEARQTY	GEAR CODE DEPENDENT GEARSIZE	GEAR CODE DEPENDENT NUMBER OF HAULS	Soak Time Evaluation (RULE ID 169)
SEH	"SEINE, HAUL"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
PUR	"SEINE, PURSE"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
SES	"SEINE, SCOTTISH"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
STS	"SEINE, STOP"	X	X	X	X	<i>tow_soak_time * number_of_hauls</i> cannot exceed time at sea
TRP	TRAP	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea
TGS	"TROLL LINE, GREEN STICK"	NA	X	NA	NA	<i>tow_soak_time</i> cannot exceed time at sea
WEI	WEIR	X	X	X	X	<i>tow_soak_time</i> can exceed time at sea

[Jump to: Table of Contents](#)

4.5 Quality Control Audit Rules

[Table 4](#) provides a list of audit rules. The audit rules engine checks the input data against the rules and, if there is an error, returns to the user an error notification as well as a text description of the error.

Find the error text descriptions also in [Table 4](#) in the column **Rule Description**.

Table 4: Quality Control Audit Rules

LEVEL	RULE ID	RULE NAME	RULE DESCRIPTION	RULE DETAIL
TRIP	3	DATE LAND BEFORE DATE SAIL	Evaluates a trip record to determine if the land date occurs before the sail date.	<i>date_land</i> must be greater than <i>date_sail</i> .
TRIP	4	DATE LAND AFTER TODAYS DATE	Evaluates a trip record to determine if the land date occurs more than one hour in the future.	<i>date_land</i> must not be greater than one hour past the submission datetime.
TRIP	5	DATE SAIL AFTER TODAYS DATE	Evaluates a trip record to determine if the sail date occurs in the future (after todays date - date trip reported).	<i>date_sail</i> must not be greater than the submission datetime.
TRIP	14	TRIP ID DOES NOT MATCH VESSEL PERMIT NUMBER	Evaluates a trip record to determine if the trip ID matches the vessel permit number reported.	The first six digits of <i>trip_id</i> must be identical to <i>vessel_permit_num</i> .
TRIP	23	TRIP MISSING SUBTRIP RECORDS	Evaluates a trip record to determine if it is missing required subtrip records. All trips where no effort reason (<i>trip_activity_type</i>) equals 0 or 1 or is null require a subtrip record. Trips with an trip activity type equal to 2,3,4,5,6,7 are not required to submit a subtrip record. However, a trip activity type 5 (for Herring Carrier Vessel) may or may not submit a subtrip record.	If <i>no_effort_trip_reason</i> is 0, a subtrip array must exist.
TRIP	27	VESSEL PERMIT NUMBER SUBMITTED INVALID	Evaluates a trip record to determine if the vessel permit number provided is a six-digit integer and has been a valid Federal permit. Does not determine if the permit is valid during the trip dates.	<i>vessel_permit_num</i> must exist in the available list of vessels. This list includes vessels with valid Federal permits or which had valid Federal permits in the past year. (link to the JSON file or the Excel download)
TRIP	32	MISMATCH BETWEEN SUBTRIPS REPORTED AND SUBTRIP RECORD COUNT	Evaluates a trip record to determine the number of subtrips reported by the trip actually equal the number of subtrip records.	<i>effort_record_count_by_trip</i> must be an integer equal to the number of subtrips represented in the trip.

LEVEL	RULE ID	RULE NAME	RULE DESCRIPTION	RULE DETAIL
TRIP	34	DUPLICATE OF EXISTING TRIP WITH SAME PERMIT AND DATE SAIL	Evaluates if a trip with the same permit number and date sail are already entered into the production CID system. Evaluates the unique constraint defined by DOC_PNUM_DSAIL unique index. On 24-AUG-15 added code to account for re-submitted trips - so only valid where there is a duplicate trip but they do not have the same trip IDs.	A submitted trip may not have the same <i>date_sail</i> and <i>vessel_permit_num</i> as a previously submitted trip which does not have the same <i>trip_id</i> . A resubmitted trip must use the same <i>trip_id</i> as the trip record it is meant to update.
TRIP	69	TRIP DATE SAIL OVERLAPS AN EXISTING TRIP	Evaluates a trip record to determine if the vessel permit number submitted a trip that overlaps (<i>date_sail</i> is between another trip's date sail and land dates) with an existing trip in CID.	For the given <i>vessel_permit_num</i> , the timespan described by <i>date_sail</i> and <i>date_land</i> must not in any way overlap a previously submitted trip which does not have the same <i>trip_id</i> . A resubmitted trip must use the same <i>trip_id</i> as the trip record it is meant to update.
TRIP	155	INVALID TEST EVTR SUBMISSION	Evaluates a trip record to determine if it is a test submission.	In the production API, <i>operator_last_name</i> must not be a known test user. Test submissions must only be submitted to the test API endpoints.
TRIP	160	INVALID TRIP TYPE	Evaluates a trip record to determine if a valid trip type has been provided.	<i>trip_type</i> must exist in the available list of trip types. (link to the JSON file or the Excel download)
TRIP	161	INVALID OPERATOR NAME LENGTH	Evaluates operator first and last name for acceptable length.	The combined length of <i>operator_first_name</i> and <i>operator_last_name</i> must not exceed 35 characters.
TRIP	168	MINIMUM TRIP LENGTH NOT MET	Compares <i>date_sail</i> and <i>date_land</i> to determine that the length of the trip is possible	<i>date_sail</i> and <i>date_land</i> must represent a span of time that exceeds 29 minutes.
TRIP	169	SOAK TIME EXCEEDS LENGTH OF TRIP	Evaluates a subtrip and trip record to determine if tow time exceeds length of trip	If <i>gearcode</i> is one of the following: 'CST', 'DIP', 'DIV', 'HND', 'HRP', or 'RAK', then <i>date_sail</i> and <i>date_land</i> must represent a span of time that is greater than or equal to the sum of all <i>tow_soak_time</i> values in the given trip. If <i>gearcode</i> is one of the following: 'DRC', 'DRM', 'DRO', 'DRS', 'DRU', 'DSC', 'DTC', 'DTS', 'OBP', 'OHS', 'OTB', 'OTC', 'OTF', 'OTM', 'OTO', 'OTR', 'OTS', 'OTT', 'PTB', 'PTM', or 'TTS', then <i>date_sail</i> and <i>date_land</i> must represent a value that is greater than or equal to the total value of <i>tow_soak_time</i> multiplied by <i>number_of_hauls</i> for all efforts with the above listed values for <i>gearcode</i> .

LEVEL	RULE ID	RULE NAME	RULE DESCRIPTION	RULE DETAIL
TRIP	177	INVALID OPERATOR NAME	Evaluates a trip record to determine if the operator name contains unexpected characters.	The fields <i>operator_first_name</i> and <i>operator_last_name</i> must not contain special characters other than apostrophe, ampersand, or hyphen.
TRIP	179	INVALID NUMBER OF CREW	Evaluates a trip record to determine if the value of number of crew is not empty or zero.	<i>number_of_crew</i> must be an integer greater than zero.
SUBTRIP	6	GEARCODE INVALID	Evaluates a subtrip record to determine if the gear code is a valid gear code.	<i>gearcode</i> must exist in the available list of gear. (link to the JSON file or the Excel download).
SUBTRIP	7	AREA INVALID	Evaluates a subtrip record to determine if the area is a valid area.	<i>chart_area</i> must exist in the available list of areas. (link to the JSON file or the Excel download)
SUBTRIP	8	GEAR REPORTED WITHOUT A SPECIES	Evaluates a subtrip record to determine if gear has been reported without reporting a species catch - except when <i>gearcode</i> = CAR which is exempt from this requirement OR a set only trip which will report gear and not species.	If <i>no_effort_trip_reason</i> =1, and <i>gearcode</i> is not 'CAR', <i>species</i> may not be NULL. 'NC' is the correct species code for an effort with no catch.
SUBTRIP	21	SUBTRIP/HAUL MISSING CATCH RECORDS	Evaluates a subtrip record to determine if the subtrip is missing catch records. Excludes set-only trips.	If <i>no_effort_trip_reason</i> is not either 1 or 2, catch records must exist. 'NC' is the correct species code for an effort with no catch.
SUBTRIP	22	ORPHAN SUBTRIP RECORD	Evaluates a subtrip record to determine if the subtrip is missing a parent trip record.	A subtrip array must exist as the child of a trip array.
SUBTRIP	33	MISMATCH BETWEEN CATCH REPORTED AND CATCH RECORD COUNT	Evaluates a subtrip record to determine if the number of catch records reported match the actual number of catch records.	<i>catch_record_count_by_subtrip</i> must be an integer equal to the number of catch records represented in the given subtrip.
SUBTRIP	162	INVALID AREA LATITUDE LONGITUDE	Evaluates a subtrip record to determine if the latitude / longitude reported describes a point in an appropriate area	The geographic point represented by <i>latitude_or_loran1</i> and <i>longitude_or_loran2</i> must exist within waters off of the U.S. Eastern Seaboard or within the Gulf of Mexico.
SUBTRIP	166	NUMBER OF HAULS MISSING	Evaluates a subtrip record to determine that <i>number_of_hauls</i> has a value when the gear code does not exempt that field from being required.	If <i>gearcode</i> is not 'HND', 'DIV', 'RAK', or 'HRP', <i>number_of_hauls</i> must not be NULL.

LEVEL	RULE ID	RULE NAME	RULE DESCRIPTION	RULE DETAIL
SUBTRIP	171	INVALID GEAR VALUES FOR PTL	Evaluates all required gear values, including <i>mesh_size</i> , <i>gearsizes</i> , <i>number_of_hauls</i> , and <i>tow_soak_time</i> , are present when gear type is PTL and trip has effort(s).	If <i>gearcode</i> is 'PTL', and <i>no_effort_trip_reason</i> is 0, values for <i>mesh_size</i> , <i>gearsizes</i> , <i>number_of_hauls</i> , <i>gearqty</i> , and <i>tow_soak_time</i> must all be greater than zero and not NULL.
SUBTRIP	180	MESH SIZE MISSING OR INVALID VALUE FOR GEAR TYPE	If mesh size is required, field cannot be 0 or null. if mesh size is not required, field can only be null or 0	Based on the available list of gear, if <i>mesh_size</i> is required for the given <i>gearcode</i> , <i>mesh_size</i> must be greater than zero. (link to the JSON file or the Excel download)
SUBTRIP	181	GEAR SIZE MISSING OR INVALID VALUE FOR GEAR TYPE	If gear size is required, field cannot be 0 or null. if gear size is not required, field can only be null or 0.	Based on the available list of gear, if <i>gear_size</i> is required for the given <i>gearcode</i> , <i>gear_size</i> must be greater than zero. (link to the JSON file or the Excel download)
CATCH	9	SPECIES CODE INVALID	Evaluates a catch record to determine if the species code is invalid.	<i>species</i> must exist in the available list of species. (link to the JSON file or the Excel download)
CATCH	15	INVALID PORT CODE	Evaluates a catch record to determine if the port ID is valid when species reported catch is greater than 0 and the port code value is not null.	If a catch record indicates a value for <i>kept</i> that is greater than zero, the value for <i>offloading_port</i> must exist in the available list of ports. (link to the JSON file or the Excel download)
CATCH	16	PORT CODE IS MISSING	Evaluates a catch record to determine if the port code is missing when species reported catch is greater than 0.	If a catch record indicates a value for <i>kept</i> that is greater than zero, <i>offloading_port</i> must not be NULL.
CATCH	20	ORPHAN CATCH RECORD	Evaluates a catch record to determine if it has a parent subtrip record - if no parent record is found than the entire trip record should be rejected.	A catch array must exist as the child of a subtrip array.
CATCH	25	INVALID DEALER NUMBER	Evaluates a catch record to determine if the dealer number was valid on the date sold.	<i>dealer_permit_number</i> must exist in the available list of dealers. This list includes Federally permitted dealers who own or owned permits valid in the past five years. (link to the JSON file or the Excel download)
CATCH	28	SPECIES CATCH WEIGHT MISSING IN KEPT AND DISCARDED	Evaluates a catch record to determine if the species reported is missing pounds <i>kept</i> and <i>discarded</i> . Requirement is that at least <i>kept</i> or <i>discarded</i> must report a value greater than zero.	If <i>species</i> is not either 'NC' or 'DCP', a value greater than zero must be entered in either or both of <i>kept</i> and <i>discarded</i> .

LEVEL	RULE ID	RULE NAME	RULE DESCRIPTION	RULE DETAIL
CATCH	72	NO CATCH REPORTED A DEALER	Evaluates a catch record to determine if a dealer was reported for a species code of NC - No Catch	For a given catch record reporting <i>species</i> = 'NC' to indicate no catch, <i>dealer_permit_number</i> must be NULL.

[Jump to: Table of Contents](#)

4.6 Regulatory References

In July 2011, the Greater Atlantic Regional Fisheries Office (GARFO) of the National Marine Fisheries Service (NMFS) approved the use of electronic reporting for the completion of Vessel Trip Reports (VTRs) on a limited, voluntary basis.

The Federal Register, in the section indicated, outlines the regulatory requirements that eVTR software must meet:

50 CFR 648.7 (b) (1) (i)

Use the following hypertext link to jump to the appropriate section:

[Link to the Federal Register language about eVTR software requirements](#)

[Jump to: Table of Contents](#)

4.7 Document Revision History

Read this section to obtain revision history information for this guide.

4.7.1 Revision 1.0 Updates

Release Date: November 12, 2019

- No revisions (initial release)

4.7.2 Revision 1.1 Updates

Release Date: December 16, 2019

- Section 4.5: **Quality Control Audit Rules**
 - Removed the following audit rules from the Quality Control Audit Rules table:
Rule IDs 10, 11, 12, 13, 17, 29, 31, 66, 67, 70, 71, 124, 125, 126, 127, 128, 129, 153, 154, 163, 164, 165, 170, 173, 174, 175, 176.
The above audit rules have been removed from use by eVTR Software Development.
 - Added a column to the Quality Control Audit Rules table.
The new column, *Audit Details*, provides additional information about conditions that would violate each referenced audit rule.

4.7.3 Revision 1.2 Updates

Release Date: April 20, 2020

- Section 3.2.3: **Returns From Authentication Request**
 - Converted the following fields to lowercase: *token*, *client_id*, *operator_key*, *vtr_preferences*
Code is case-sensitive; fields must be lowercase. (Exception is *Status*)
- Section 3.3.3: **JSON Format: Catch and Sales Data Together (Test)**
 - Added *client_id* to JSON code example.
The *client_id* is returned with a correct authorization request, and the value needs to be submitted with the eVTR.
- Section 3.3.4: **JSON Format: Separate Arrays for Catch and Sales Data (Test)**
 - Added *client_id* to JSON code example.
The *client_id* is returned with a correct authorization request, and the value needs to be submitted with the eVTR.
 - Removed *operator_password*.
Deprecated.
- Section 4.2: **eVTR Form Specification**
 - Added *client_id* entry to specification.
The *client_id* is returned with a correct authorization request, and the value needs to be submitted with the eVTR.

4.7.4 Revision 1.2.1 Updates

Release Date: July 20, 2020

- Section 4.3: **Mandatory Use of a Field in eVTR Based on Trip Activity Type**
 - Expanded *number_of_anglers* specification
Added Trip Type 6 (private/recreational)

4.7.5 Revision 1.3 Updates

Release Date: January 15, 2021

- **Global edit:** changed value **No Data** to value **NA** throughout the guide.

Context: A value of **NA** in a table cell indicates that there is no value or instruction from GARFO concerning that particular instance.

Affected sections in the guide:

- Section 2.4: **File Format**
- Section 4: **eVTR Reference**
- Section 4.2: **eVTR Form Specification**
- Section 4.3: **Mandatory Use of a Field in eVTR Based on Trip Activity Type**
- Section 4.4: **Gear Code and Soak Time Interdependence**

- **Global edit:** changed base URL to https://apps-garfo.fisheries.noaa.gov/base_api/ throughout the guide.
Context: This URL forms the basis of all requests to the eVTR API module. Create specific requests by appending request strings on this URL.
Affected sections in the guide: Section 3.1.1 forward through the API discussion.
- Section 3.2.2: **Authentication**
 - Added the following fields: *vendor_id*, *vendor_software_version*, *vendor_app_name*, *vendor_app_platform*
Values that more precisely identify the app.
- Section 3.2.2.1: **App Platform**
 - New section
More information about the *vendor_app_platform* field.
- Section 3.2.3.1: **Return From Correct Authorization**
 - Added the following fields: *token_expires*, *entitlements*.
 - *token_expires*
Displays date/time expiration of authorization token.
 - *entitlements*
Displays vessel(s) that the user is entitled to submit eVTRs for.
- Section 3.3.3.2: **Description**
 - Added the following fields: *vendor_app_name*, *vendor_app_platform*
Fields to provide more vendor information.
 - Removed the following fields: *trip_count*, *effort_count*, *catch count*, *effort_record_count_by_trip*, *catch_record_count_by_subtrip*, *species_count*.
Deprecated.
- Section 3.3.4.2: **Description**
 - Added the following fields: *vendor_app_name*, *vendor_app_platform*
Fields to provide more vendor information.
 - Removed the following fields: *trip_count*, *effort_count*, *catch count*, *effort_record_count_by_trip*, *catch_record_count_by_subtrip*, *species_count*.
Deprecated.
- Section 4.2: **eVTR Form Specification**
 - Added *vendor_app_platform* entry to specification.
 - Removed the following fields: *effort_count*, *catch count*, *effort_record_count_by_trip*, *catch_record_count_by_subtrip*, *species_count*.
Deprecated.

[Jump to: Table of Contents](#)