



# **Frame format for Free-Form Events**

**Last Revised October 24, 2011**

**All contents in this manual are copyrighted by JAVAD GNSS.  
All rights reserved. The information contained herein may not be used, accessed, copied,  
stored, displayed, sold, modified, published, or distributed, or otherwise reproduced without express written  
consent from JAVAD GNSS**



# FRAME FORMAT FOR FREE-FORM EVENTS

## Preface

The purpose of this document is to describe a “frame format” for free-form events. Recall that free-form events are supported not only by Tracy™ but also by other JAVAD GNSS software packages as Giodis™, Justin™ and Jps2Rin™. In this document the reader will also find the detailed description of all of the free-form event types supported by Tracy™ and VS software. Not only fully described are these event types, but also recommendations are given as to how program developers should handle every specific free-form event type. We suggest that program developers, whether JAVAD GNSS or “outside” software engineers, should strictly follow the proposed frame format when writing code dealing with free-form events. This will allow them to avoid unnecessary format clashes in the future.

## General format of JAVAD GNSS Free-Form Events

Tracy uses the following format for free-form events: <DATA\_NAME>=<value>

where <DATA\_NAME> designates one of the following names: \_ANT, \_ANH, \_DYM, \_SIT, \_CAN, \_SAV, \_DSC, \_FEA. <DATA\_NAME> determines the “physical” type of the corresponding event.

By convention, all free-form events created by JAVAD GNSS software must start with underscore. To avoid format conflicts, third-party program developers who create their own JAVAD GNSS receiver interfacing software should not use the leading underscore in their own (“non-javad”) <DATA\_NAME> unless these are syntactically and semantically identical to the JAVAD GNSS counterparts.

Below we will use <DATA\_NAME> to refer to specific free-form event types.

## Frame format for Free-Form Events

Currently supported JAVAD GNSS Free-Form Events

\_ANT Event (“ANTENNA TYPE”)

# Currently supported JAVAD GNSS Free-Form Events

## **\_ANT Event (“ANTENNA TYPE”)**

This event indicates JAVAD GNSS antenna type used. The value of an \_ANT event is a non-empty string comprising up to 20 ASCII characters. Although this event can contain an arbitrary string, it is highly recommended to use only antenna type names approved and standardized by IGS. For officially adopted antenna type identifiers, visit the GPS Antenna Calibration page on the IGS web site at <http://www.ngs.noaa.gov/ANTCAL/>.

**Example:** \_ANT=JAV\_TRIUMPH-1 NONE

## **\_ANH Event (“ANTENNA HEIGHT”)**

The value of an \_ANH event has the format <height\_value>[s], where height\_value is an ASCII representation of the measured antenna height in meters.

Antenna height is expressed as a float in the “usual” (not scientific) notation so that the integer part is delimited from the fractional one with a dot. The optional *s* character indicates, if present, that the corresponding value relates to the *measured slant*. If no optional character is used, this indicates the *measured vertical*.

**Example:** \_ANH=2.000  
\_ANH=1.543s

## **\_DYM Event (“DYNAMICS”)**

The value of a \_DYM event can be either STATIC or DYNAMIC.

STATIC indicates that the antenna phase center (APC) is fixed at the moment this event is issued (and that the APC will probably remain static for some time after issuing this event, say, until another \_DYM event appears).

On the contrary, DYNAMIC indicates that the APC will probably not be static after the event is issued.

**Example:** \_DYM=STATIC  
\_DYM=DYNAMIC

## **\_SIT Event ("SITE")**

The value of a \_SIT event is a non-empty string comprising up to 20 ASCII characters. Currently only alpha-numeric characters, hyphen and underscore are allowed for use. In the future, however, we plan to relax the above mentioned limitations on \_SIT <value>.

By JAVAD GNSS convention, \_SIT<value> serves as an occupation (or site) identifier. Note that this kind of information plays a very important part in post-processing applications.

**Example:** \_SIT=P1-34\_aBcD

## **\_CAN Event ("CANCEL")**

The value of a \_CAN event is a non-empty string comprising up to 20 ASCII characters. Currently only alpha-numeric characters, hyphen and underscore are allowed for use. A CANCEL event is issued if the user wants the post-processing software to ignore the earlier entered occupation name. Post-processing software is supposed to take a particular CANCEL event into account if the following two conditions exist:

- the CANCEL event is preceded by at least one SITE event in the log-file;
- the CANCEL event and the latest of the SITE events preceding it have the same value. If it is not the case, the CANCEL event is considered "false" and is discarded.

A CANCEL event, if identified as valid, will instruct the post-processing software to "cancel" the given occupation, i.e., all the data collected over the interval between the corresponding SITE and CANCEL events will not be attributed to any of the measured occupations.

**Note:** The event can be used without =name value or the name value can be empty. In that case the event cancels the last object.

## **\_SAV Event ("SAVE")**

The value of a \_SAV event is a non-empty string comprising up to 20 ASCII characters. Currently only alpha-numeric characters, hyphen and underscore are allowed for use. Post-processing software is supposed to take a particular SAVE event into account if the following condition exist:

- the SAVE event is preceded by at least one SITE or \_DYM event in the log-file;

SAVE events are intended to notify the post-processing software that the corresponding occupation is finished and that all of the data recorded at this occupation are considered valid. Also it is used to allow user modify object name during surveying. That's why name in \_SAV event is treated as final name for the object if it doesn't match with previous \_SIT event. By JAVAD GNSS convention, a SAVE event, if considered "true", indicates that, as a rule, data recorded after issuing this SAVE event do not belong to the occupation whose name is specified in the field \_SAV <value>.

**Example:** \_SAV=P1-34\_aBcD

## Frame format for Free-Form Events

Currently supported JAVAD GNSS Free-Form Events

Site Scope

## Site Scope

Below in this document, we will use the term “named site scope” or just “site scope”. The formal definition of the term is as follows: Named site scope in a JAVADGNSS receiver log-file is a set of messages recorded into this file over the interval between a site scope opening event, which is always a SITE event, and the corresponding site scope closing event, which can be:

Currently supported JAVADGNSS Free-Form Events `_DSC` Event (“DESCRIPTION”)

- another SITE event with a different <value>
- a SAVE event (even with different <value>)
- a CANCEL event with the same <value> or without a <value>
- a DYNAMICS event whose <value> differs from the <value> of the previous DYNAMICS event in this scope.

## `_DSC` Event (“DESCRIPTION”)

This event type enables the user to associate a free format comment with the current occupation. Issuing a `_DSC` event while not within a particular site scope still can be quite useful. Interpreting “out of scope” description events is a software-specific procedure.

A DESCRIPTION event's value can be an arbitrary sequence of 8-bit bytes excluding - <CR> (0x0D), - <LF> (0x0A), - <double quote>” (0x22). Note that double backslash (i.e. \\) is used to represent backslash itself.

**Note:** Currently Tracy™ does not allow the user to enter description text longer than 127 bytes.

**Note:** In DESCRIPTION events, it is assumed that all of the bytes whose values are less than 128 as well as the two-byte sequences `\r`, `\n`, and `\'` will represent ASCII characters. However, no assumptions are made regarding other bytes that can be found in this type of event, i.e., interpreting such bytes is software-specific.

## `_FEA` Event (“FEATURE”)

This event type allows the user to specify a “spatial feature” for an object related to the GIS project being done. This object can have more than one attribute associated with it so that the field operator will have to do work similar to filling in a questionnaire form.

The general format of `_FEA` event is as follows:

`_FEA=EntityName:FieldName|FieldType|Units=ValueAsText`

where EntityName=AuthorityID.

EntityName Properly FieldType and Units are optional (by default, “string” and “meter”).

There exist four field types: s, i, f, d (string, integer, float, date and time, respectively). The symbol “|”(0X7C) serves as a delimiter.

Also, if a feature event contains a numeric value, it may be necessary for the operator to specify the unit of measurement (meters, seconds, etc.). The letter designating the unit of measurement is put before the equality sign (“=”), which is followed by the corresponding numeric value (see the example below).

**Example:** Supposing we want to define a few features for a light post while performing our GIS survey. Assume that the following information needs to be entered into the receiver log file: POINT1 (name of the object that we enter the features for) Height: 4.2 meters Material: aluminum Condition: poor Age: 99 years

In this example, the four feature events will look as follows:

```
_SIT=POINT1
```

```
....
```

```
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Height|f|m=4.2
```

```
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Material|s=aluminum
```

```
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Condition|s=poor
```

```
_FEA=SuperSurveyorLTD.POLE:SuperSurveyorLTD.Age|i|year=99
```

```
●
```

## Additional comments on `_CAN` and `_SAV`

For post-processing software, it would be incorrect to consider the free-form events `CANCEL` and `SAVE` as the only possible end-of-occupation indicators. In some scenarios, the end of an occupation may coincide with the end of the file, or with the beginning of another occupation (`_SIT` event), or with some other events. If the situation is especially involved, the post-processing software may need to prompt the user for additional information in order to correctly recognize the occupation boundaries, etc.

## `_EVT` Event (“Event”)

This event is used to mark objects during trajectory survey. The `<value>` indicate a name for the new object marked during previous started trajectory object (started with `_DYM-DYNAMIC` and `_SIT` events before).

The scope of the object is defined only with the event.

**Example:** `_EVT=OBJ123`

```
●
```

## `_GUI` Event (“GUID”)

This event is used to assign an global unique identifier for object started before with `_DYM` or with `_SIT` events. The event is used to assign the object any external file/database with Post-Processing software.

**Example:** `_GUI={545FC8B1-A2BB-4ab3-9906-20D29038C036}`

```
●
```

## Frame format for Free-Form Events

Additional comments on `_CAN` and `_SAV`

`_MED` Event (“Media”)

## `_MED` Event (“Media”)

This event is used to assign an external media file (photo or audio-note) for object started before with `_DYM` or with `_SIT` events. The event is used to assign the object any external file/database with Post-Processing software.

**Example:** `_MED=file1.jpg`



## `_OFF` and `_OFD` Events (“Offset and Offset Data”)

This event is used for offset surveying during post-processing data collection. It creates new object based on previously measured objects from the file and some data (e.g. length, angles etc.). The new object will have name marked with `_SIT` / `_SAV` events. `_OFF` event mark type of the event by the integer value and `_OFD` events provide data for computation.

**Example:** `_OFF=1 _OFD=P1->Point1 _OFD=P2->Point2 _OFD=Dist1->1.25m _OFD=Dist2->0.3m`



| <code>_OFF</code> value | Task type                                 | <code>_OFD</code> values   |
|-------------------------|---|--|
| 1                       | Standard offset                           | <code>_OFD=P1-&gt;</code> start point name<br><code>_OFD=P2-&gt;</code> finish point name<br><code>_OFD=Dist1-&gt;</code> along line distance<br><code>_OFD=Dist2-&gt;</code> across line distance (negative value means left) |
| 2                       | Along line offset                         | <code>_OFD=P1-&gt;</code> start point name<br><code>_OFD=P2-&gt;</code> finish point name<br><code>_OFD=Dist1-&gt;</code> along line distance from p2 (negative back from p2)  |
| 3                       | Intersection<br>(distances from 2 points) | <code>_OFD=P1-&gt;</code> first point name<br><code>_OFD=P2-&gt;</code> second point name<br><code>_OFD=Dist1-&gt;</code> distance to point p1<br><code>_OFD=Dist2-&gt;</code> distance to point p2                            |
| 4                       | Perpendicular                             | <code>_OFD=P1-&gt;</code> start point name<br><code>_OFD=P2-&gt;</code> second point name<br><code>_OFD=Dist1-&gt;</code> distance for the perpendicular direction from point p2   |

## **Frame format for Free-Form Events**

Additional comments on \_CAN and \_SAV  
\_OFF and \_OFD Events (“Offset and Offset Data”)



900 Rock Avenue, San Jose, CA 95131 USA

Phone: +1(408)770-1770

Fax: +1(408)770-1799

[www.javad.com](http://www.javad.com)

Copyright © JAVAD GNSS, Inc., 2011  
All rights reserved. No unauthorized duplication.