

NASA/TM-2013-216034



TSAFE Interface Control Document v 2.0

*Russell Paielli
Ames Research Center
Moffett Field, California*

*Ralph Bach
Aerospace Computing, Inc.
Moffett Field, California*

January 2013

The NASA STI Program Office . . . in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA Scientific and Technical Information (STI) Program Office plays a key part in helping NASA maintain this important role.

The NASA STI Program Office is operated by Langley Research Center, the Lead Center for NASA's scientific and technical information. The NASA STI Program Office provides access to the NASA STI Database, the largest collection of aeronautical and space science STI in the world. The Program Office is also NASA's institutional mechanism for disseminating the results of its research and development activities. These results are published by NASA in the NASA STI Report Series, which includes the following report types:

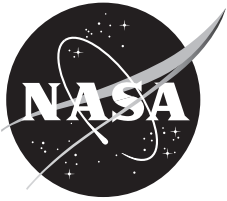
- **TECHNICAL PUBLICATION.** Reports of completed research or a major significant phase of research that present the results of NASA programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA's counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.
- **TECHNICAL MEMORANDUM.** Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.
- **CONTRACTOR REPORT.** Scientific and technical findings by NASA-sponsored contractors and grantees.

- **CONFERENCE PUBLICATION.** Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or cosponsored by NASA.
- **SPECIAL PUBLICATION.** Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.
- **TECHNICAL TRANSLATION.** English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services that complement the STI Program Office's diverse offerings include creating custom thesauri, building customized databases, organizing and publishing research results . . . even providing videos.

For more information about the NASA STI Program Office, see the following:

- Access the NASA STI Program Home Page at <http://www.sti.nasa.gov>
- E-mail your question via the Internet to help@sti.nasa.gov
- Fax your question to the NASA Access Help Desk at (301) 621-0134
- Telephone the NASA Access Help Desk at (301) 621-0390
- Write to:
NASA Access Help Desk
NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320



TSAFE Interface Control Document v 2.0

*Russell Paielli
Ames Research Center
Moffett Field, California*

*Ralph Bach
Aerospace Computing, Inc.
Moffett Field, California*

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, California 94035-1000

Available from:

NASA Center for AeroSpace Information
7115 Standard Drive
Hanover, MD 21076-1320
(301) 621-0390

National Technical Information Service
5285 Port Royal Road
Springfield, VA 22161
(703) 487-4650

TABLE OF CONTENTS

INTRODUCTION	1
Flight Registration	3
Route Amendment	4
Vector Amendment	4
Altitude Amendment	4
Track Update	5
Flight Deletion	6
Wind Data	6
IFR/VFR Status Change	6
TSAFE OUTPUTS	7
Conflict Alerts	7
Conflict Removal	8
Conflict Resolutions	8
Maneuver Release	9

LIST OF FIGURES

Figure 1. TSAFE input/output overview.	2
---	---

LIST OF TABLES

Table 1. TSAFE INPUT RECORD TYPES AND CORRESPONDING METHODS	2
Table 2. TSAFE OUTPUT RECORD TYPES FOR CONFLICT DETECTION	7
Table 3. TSAFE OUTPUT RECORD TYPES FOR CONFLICT RESOLUTION	7

TSAFE INTERFACE CONTROL DOCUMENT V 2.0

Russell Paielli and Ralph Bach¹

Ames Research Center

INTRODUCTION

This document specifies the data interface for TSAFE, the Tactical Separation-Assisted Flight Environment. TSAFE is a research prototype of a software application program for alerting air traffic controllers to imminent conflicts in enroute airspace. It is intended for Air Route Traffic Control Centers (“Centers”) in the U.S. National Airspace System. It predicts trajectories for approximately 3 minutes into the future, searches for conflicts, and sends data about predicted conflicts to the client, which uses the data to alert an air traffic controller of conflicts. TSAFE itself does not provide a graphical user interface.

TSAFE also has an optional conflict-resolution capability that, when enabled, attempts to resolve conflicts within approximately 2 minutes of loss of separation by maneuvering one or both flights with a simple change of altitude or heading. The resolution maneuvers are sent to the client for controller action or, eventually, could be sent directly to the flight deck (possibly using the Mode S datalink with voice synthesis). The resolution function is intended as a tactical backup to a more complex strategic resolver that attempts to resolve conflicts approximately 10 minutes or more in advance.

TSAFE is written in the Scala programming language, which is fully interoperable with Java and adds support for functional programming. TSAFE can operate as a server through a standard TCP/IP socket, or it can be used as a software library that is called directly. The inputs to TSAFE, as shown in figure 1, identify flights and their aircraft types, assigned routes, assigned altitudes, surveillance tracking data, barometric altitude data, and wind data. The basic outputs from TSAFE are the predicted conflicts and, if conflict resolution is enabled, maneuvers to resolve them.

When TSAFE is operated as a server through a socket, the inputs are data records, each consisting of a line of standard ASCII text with data fields delimited by spaces. TSAFE ignores data or text following a “#” symbol. Multiple records can be sent in a single message by separating the records with semicolons. Alternatively, the client can use TSAFE as a library and call its methods directly. The method interfaces are not discussed in this document, but a general method called “processLine” provides a text-based interface identical to the socket interface.

¹ Aerospace Computing, Inc., 465 Fairchild Drive, Suite 224, Mt. View, CA 94043.

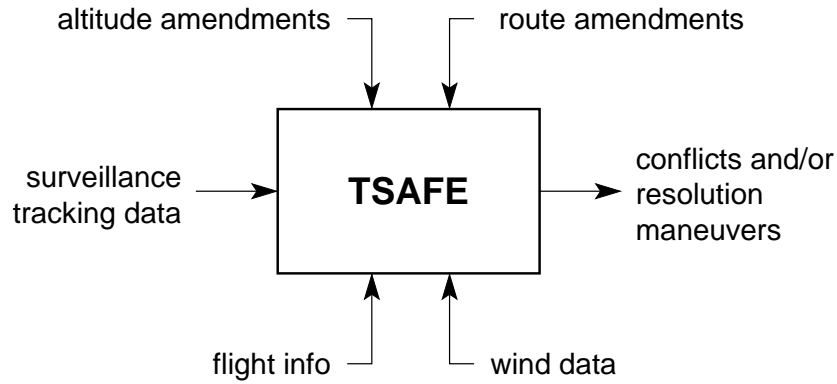


Figure 1. TSAFE input/output overview.

Table 1 lists the TSAFE input record types, their three-letter codes, and the corresponding TSAFE methods. The table shows that three of the record types, the Track Update and Vector and Altitude Amendments, trigger a conflict check between the flight for which the record applies and all the other traffic in the Center. If a conflict is detected, updated, or resolved, TSAFE returns the relevant data from the method call or through the socket.

Each data record begins with a three-letter type code, followed by a record timestamp and an identifier of the flight to which it applies. Here is an example:

FLT 1162849395 JBU349/MCO A320/Q IFR RVSM OVR 360 454

In this example, the record type is FLT, the timestamp is 1162849395, and the flight identifier is JBU349/MCO. The timestamp and flight identifier occupy the same fields for all input record types associated with a particular flight.

TABLE 1. TSAFE INPUT RECORD TYPES AND CORRESPONDING METHODS

Input Record Type	Code	Method	Triggers Conflict Check?
Flight Registration	FLT	addFlight	No
Route Amendment	RTE	amendRoute	No
Vector Amendment	VEC	assignVector	Yes
Altitude Amendment	ALT	amendAltitude	Yes
Track Update	TRK	updateTracks	Yes
Flight Deletion	DEL	delFlight	No
Wind Data	WND	loadWindData	No
IFR Status Change	IFR	setIFRflag	No

The record timestamp is the time at which the data record was sent. It is defined in terms of standard “unix” time (seconds since midnight on 1970-01-01). During a simulated replay of a recorded input file, the timestamp is used to drive the simulation clock. The timestamp for each record should not be earlier than the timestamp of the previous record. It can be given as an integer or as a floating-point number.

The flight identifier is a unique character string that identifies the flight. One convention is to use the standard call sign with the originating airport (and optionally the beacon code) appended after a slash, but the only requirements are that the identifier for each flight must be unique, it must contain no whitespace, and it must be consistent for all data records corresponding to that flight. The TSAFE input records and their remaining data fields are defined in the following subsections.

Flight Registration

Flight registration is represented by the FLT record type, which triggers a call of the addFlight method of TSAFE. This record corresponds to the filed flightplan, except that it does not include the planned route (which is given by the RTE record type discussed in the subsection “Route Amendment” later in this document). This record, which should appear once for each flight, provides basic information about the flight. TSAFE will function without it but not as effectively as it can with the information in this record. The header comment for this record type is shown here, followed by a sample record:

```
# FLT (register flight): (1) time/sec (2) flight ID (3) aircraft type (4) IFR flag (5) RVSM flag  
# (6) ATC type (7) filed altitude/FL (8) filed speed/kn (9) second filed altitude/FL
```

```
FLT 1162849395 JBU349/MCO A320/Q IFR RVSM OVR 360 454
```

As explained earlier, fields 1 and 2 are the record timestamp and the identifier of the flight to which it applies. Field 3 gives the standard Federal Aviation Administration (FAA) aircraft type designation, which TSAFE uses to select the appropriate aircraft model to predict climb and descent rates as functions of altitude. Field 4 indicates the filing status of the flight as either instrument flight rules (IFR) or visual flight rules (VFR). Field 5 is either Reduced Vertical Separation Minimum (RVSM) or non-RVSM, depending on whether or not the aircraft is equipped for RVSM. Field 6 indicates the air traffic control (ATC) type of the flight, which will be one of the following codes:

```
DEP  departure: origin inside Center and destination outside Center  
OVR  overflight: both origin and destination outside Center  
ARR  arrival: origin outside Center and destination inside Center  
INR  internal: both origin and destination inside Center  
UNK  unknown
```

Field 7 gives the filed altitude in flight levels (units of 100 feet (ft)), and field 8 gives the filed true airspeed in knots (kn). If data is unavailable for field 7 or 8, a value of zero should be substituted. The optional field 9 gives the second altitude of a block altitude assignment (for the relatively rare case in which a flight is assigned a range of altitudes rather than a single altitude).

Route Amendment

Route amendments are represented by the RTE record type, which triggers a call of the `amendRoute` method of TSAFE. This record provides information about flightplan routes initially entered by the pilot or airline, and later route amendments entered by the controller or an automated resolution system (other than TSAFE). Because the FLT record does not include the route, an RTE record should immediately follow the FLT record to show the filed or initially assigned route for each flight. Again, the header comment for this record type is shown, followed by a sample record:

```
# RTE (amend route): (1) time/sec (2) flight ID (3) route/nmi
```

```
RTE 1162851129 ASH2606/IAH 675.26,416.78/786.75,480.75/805.29,489.93
```

After the usual timestamp and flight identifier, field 3 gives the route in terms of a series of two-dimensional (2-D) waypoints in standard Center stereographic (x, y) coordinates in units of nautical miles (nmi). The waypoint coordinates are separated by a comma, and each waypoint is separated from the next waypoint by a slash. The entire list of waypoints is compressed into a character string with no spaces to form a single data field. The coordinates should each have at least two digits after the decimal point.

Vector Amendment

A vector amendment is usually a tactical maneuver created by a controller, a client auto-resolver, or TSAFE (see the section “TSAFE Outputs” later in this document). Vector amendments are not currently used, but they could be used in the future. Vector amendments are represented by the VEC record type, which triggers a call of the `assignVector` method of TSAFE:

```
# VEC (amend heading): (1) time/sec (2) flight ID (3) target heading/deg (4) heading spec
```

```
VEC 1272314846 AAL1596/KDFW 115 mag
```

As before, fields 1 and 2 are the record timestamp and the flight to which the maneuver applies, and field 3 specifies the target or assigned heading in degrees. Field 4 specifies whether the target heading is given as magnetic (mag), true (tru), or course (crs) heading.

Each VEC record triggers an immediate check for conflicts between the flight for which the record applies and all other traffic in the Center (unless the flight is under TSAFE control). If TSAFE detects a change in the conflict status, it sends a message (described later) to the client to update its tactical conflict list.

Altitude Amendment

Altitude amendments are represented with the ALT record type, which triggers a call of the `amendAltitude` method of TSAFE. (A method called `amendAltitudeJ` is also provided for compatibility with Java.) The header comment for this record type and a sample record follow:

ALT (amend altitude): (1) time/sec (2) flight ID (3) cleared alt/FL (4) temp altitude flag

ALT 1162849395 JBU349/MCO 360 plan

Field 3 gives the assigned altitude in flight levels (units of 100 ft). Field 4 is “temp” if the amendment is a temporary altitude; otherwise it is “plan” or empty if it is a flightplan altitude. (For backwards compatibility, “1” can be used in place of “temp,” and “0” can be used in place of “plan.”) A temporary altitude is cleared by setting the temporary altitude to 0, which restores the previous flightplan altitude assignment.

Each ALT record triggers an immediate check for conflicts between the flight for which the record applies and all other traffic in the Center (unless the flight is under TSAFE control). If TSAFE detects a change in the conflict status, it sends a message (described later) to the client to update its tactical conflict list.

Track Update

Radar or Automatic Dependent Surveillance - Broadcast (ADS-B) surveillance track updates are represented by the TRK record type, which triggers a call of the updateTracks method of TSAFE. (A method called updateTracksJ is also provided for compatibility with Java.) Again, the header comment for this record and a sample record follow:

TRK (track update): (1) time/sec (2) flight ID (3) relative track time (4) x/nmi (5) y/nmi
(6) alt/ft (7) groundspeed/kn (8) course/deg (9) altrate/fpm (10) sector number

TRK 1162851062 ASH2606/IAH -2.5 670.010 406.535 28500 472.0 38.4 1293.5 86

Field 3 is the relative track time, the track time relative to the timestamp in the first field. Because a delay of a few seconds can occur between the actual track and the recording of the data record, the actual track time can be a few seconds earlier than the record time. To save space, the relative time is given rather than the absolute time. The absolute track time is reconstructed by adding the relative track time to the record timestamp in the first data field.

Fields 4 and 5 give the (x, y) coordinates of the radar position of the aircraft in nautical miles with reference to the standard stereographic coordinate system of the Center. The coordinate system used here must be consistent with the coordinate system used for the flightplan waypoints in the RTE record discussed previously. These coordinates should be given with at least three digits after the decimal place. Field 6 gives the reported barometric altitude in feet.

Fields 7–9 give the estimated velocity of the flight: field 7 gives the groundspeed in knots (kn), field 8 gives the course angle in degrees (deg), and field 9 gives the altitude rate in feet per minute (fpm). The number of digits given should be at least as many as shown in the sample record given in this section. Note that TSAFE has an option to produce its own velocity estimates based on the position data. If that option has been selected, TSAFE will ignore fields 7–9 of this record. Finally, field 10 gives the identification number of the sector that is currently responsible for the flight. If the sector number is unavailable, it should be omitted.

Each TRK record triggers an immediate check for conflicts between the flight for which the record applies and all other traffic in the Center. If TSAFE detects a change in the conflict status, it sends a message (described later) to the client to update its list of predicted conflicts.

Flight Deletion

Flight deletion is represented by the DEL record type, which triggers a call of the delFlight method of TSAFE. It is used to indicate that the flight is no longer active in the Center and can be dropped. The use of this record type is optional. If not used, TSAFE will automatically drop the flight several minutes after track updates end. The header comment for this record type and a sample record follow:

DEL (delete flight): (1) time/sec (2) flight ID

DEL 1162851098 JBU349/MCO

Wind Data

Wind data updates are represented by the WND record type, which triggers a call of the loadWindFile method of TSAFE to tell TSAFE where to find the latest wind data information. The header comment and a sample record follow:

WND (wind data): (1) time/sec (2) wind data file name

WND 1162849395 wind/2010-10-06.11.txt

The current default format for wind data files is a standard format used by the Center/TRACON Automation System (CTAS), but other formats can also be used if a function is provided to read them.

IFR/VFR Status Change

Changes in the IFR/VFR status of a flight are represented by the IFR record type, which triggers a call of the setIFRflag method of TSAFE. This record is needed only if the status of the flight changes after the flight is registered using the FLT record discussed earlier. An example of a record of this type follows:

IFR 1162849395 JBU349/MCO 1

As usual, the first two data fields following the record type code are the record timestamp and the identifier of the flight to which it applies. The “1” at the end of the record indicates that the IFR/VFR status of the flight is set to IFR. A “0” would indicate that the status is set to VFR.

TSAFE OUTPUTS

If TSAFE is operated with text-based input (either as a server through a TCP/IP socket or with direct calls of the “processLine” method), its outputs, like its inputs, are data records that each consist of a line of standard ASCII text with data fields delimited by spaces. Data or text following a “#” symbol are not intended for the client and can be safely ignored. Multiple records can be sent in a single message by separating the records with semicolons. If TSAFE is used as a library and the methods listed in table 1 are called directly, the outputs are the return values of the methods that trigger conflict checks, but those outputs are not discussed in this document. Table 2 lists the TSAFE output record types and their three-letter codes for conflict detection, and table 3 lists the same information for conflict resolution.

TABLE 2. TSAFE OUTPUT RECORD TYPES FOR CONFLICT DETECTION

Output Record Type	Code
Predicted Conflict	pre
Loss of Separation	los
Critical Leveloff	clo
Conflict Removal	rem

TABLE 3. TSAFE OUTPUT RECORD TYPES FOR CONFLICT RESOLUTION

Output Record Type	Code
Altitude Maneuver	alt
Vector (turn) Maneuver	vec
Maneuver Release	rel

Conflict Alerts

TSAFE provides three types of conflict alerts: “pre” represents a predicted conflict, “los” represents a loss of separation that has already occurred, and “clo” represents a critical level-off alert. A critical level-off alert means that an immediate loss of separation is predicted to result if the flight fails to level off at its cleared altitude (these alerts are optional and experimental). The format for these three conflict alert record types is identical except for the record type keyword. The header comment and a sample prerecord follows:

```
# pre (predicted conflict): (1) time/sec (2) conflict pair ID (3) predicted time to LoS/sec  
# (4) min-separation ratio (5) positions at min sep
```

```
pre 1272314846 AAL1596/DFW-AAL2329/ORD 22.0 0.922 550.15,495.59,11256/550.18,500.96,11000
```

Field 1 is the usual record timestamp as defined previously for input records. Field 2 is the conflict identifier, a unique character string that identifies the conflicting pair of flights. The convention is to

concatenate the individual flight IDs (in alphabetical order) with a dash. Field 3 gives the time in seconds, relative to the record time, at which the loss of separation is predicted to occur (or is zero if it already occurred). Field 4 gives the predicted minimum separation ratio, defined as the ratio of the (actual or predicted) separation and the required separation. Field 5 is a character string containing the predicted positions at minimum separation (x/nmi, y/nmi, and h/ft) for each flight, separated by a slash.

The minimum separation ratio in field 4 is calculated for both vertical and horizontal separation, and the maximum of the two is taken. Thus, a separation ratio of zero corresponds to a collision, and a separation ratio of one corresponds to the exact minimum required separation. However, the separation ratio does not apply (or is considered “large”) if the flights are “separated by altitude clearance.” Two flights are separated by altitude clearance if minimum required altitude separation is guaranteed as long as neither flight diverges from its cleared altitude. Thus, if two flights are flying level at adjacent cleared altitudes, the separation ratio is considered large even though it could be 1.0 according to the previous rules. The separation ratio provides a measure of the severity of the predicted conflict.

Conflict Removal

The client is responsible for creating and maintaining a tactical conflict list based on TSAFE conflict alerts. The client should remove a conflict from the list upon receiving a “rem” message from TSAFE, which is of the form:

rem (remove conflict): (1) time/sec (2) conflict pair ID

rem 1272314920 AAL1596/KDFW-AAL2329/KORD

Alternatively, the client can call the “sendConflictList” method (or send that method name through the TCP/IP socket or the processLine method) to get the current list of conflicts.

Conflict Resolutions

There are currently two record types that specify conflict-resolution maneuvers. The first represents an altitude maneuver and is of the form:

alt (altitude maneuver): (1) time/sec (2) flight ID (3) target alt/FL

(4) predicted separation ratio (5) altitude change/FL (6) conflict pair ID

alt 1272315458 EGF3419/DFW 110 1.40 -20 EGF3419/KDFW-EGF3446/KGGG

Fields 1 and 2 are the record timestamp and the flight to which the maneuver applies. Field 3 specifies the target or cleared altitude in units of flight level (100 ft). Field 4 is the separation ratio that is predicted to result from the maneuver, and field 5 is change in altitude in flight levels from the previous assigned altitude for the maneuvered flight. Field 6 is the conflict pair identifier for the conflict being resolved.

The second resolution type is a vector maneuver. The record type is of the form:

```
# vec (vector maneuver): (1) time/sec (2) flight ID (3) target heading/deg
# (4) heading spec (5) bank angle/deg (6) predicted separation ratio
# (7) heading change/deg (8) conflict pair ID
```

```
vec 1272314846 AAL1596/KDFW 115 mag 20 1.35 +18 AAL1596/KDFW-AAL2329/KORD
```

As before, fields 1 and 2 are the record timestamp and the flight to which the maneuver applies. Field 3 specifies the target or assigned heading vector in degrees. Field 4 specifies whether the target vector is given as magnetic heading (mag), true heading (tru), or course (crs). Field 5 is the nominal bank angle in degrees. The bank angle can be either 20 or 30 deg, the former representing a standard turn and the latter representing an “expedited” turn. Field 6 is the separation ratio that is predicted to result from the maneuver. Field 7 is the heading change from the current heading (positive clockwise, as usual). Field 8 is the conflict pair identifier for the conflict that is being resolved.

TSAFE can be operated in several different modes, from fully automated to advisory only. In fully automated mode, resolution maneuvers are relayed to the flight deck without controller intervention. In that mode, the ATC client should take note of the maneuver and not try to maneuver that flight again until TSAFE issues a maneuver “release” message (see the next section). If TSAFE is used in advisory mode, the ATC client is responsible for issuing the maneuver and sending an appropriate message back to TSAFE to indicate that the maneuver has been issued. For example, an ALT message would be used for an altitude maneuver, or a VEC message would be used for a vector maneuver, as explained earlier. If the client should detect a conflict less than approximately 6 minutes ahead while a flight is under TSAFE control, the client should provide a resolution for the flight that has not yet been maneuvered. This resolution should enhance safety, since TSAFE looks ahead only 3 minutes.

Maneuver Release

When the two flights for a resolved conflict are safely separated and diverging, TSAFE sends a release (rel) message for those flights, releasing them from their maneuver. A release message is of the form:

```
# rel (release maneuver): (1) time/sec (2) flight IDs ...
```

```
rel 1272314911 AAL1596/DFW DAL332/SFO
```

The number of flight IDs following the time can be one or more. When the client receives a release message, it can take back control of each flight and return it to its original altitude, heading, or planned route, sending an appropriate RTE or ALT amendment back to TSAFE in the process.

Alternatively, the client can call the “sendManeuverList” method (or send that method name through the TCP/IP socket or the processLine method) to get the current list of active maneuvers.

[last revision on 2012-08-14. Contact author at Russ.Paielli@nasa.gov for updates]