
Surveillance User Interface Specification

March 2003
Version 1.0
formal/03-03-62



An Adopted Formal Specification of the Object Management Group, Inc.

Copyright © 2001, THALES ATM
Copyright © 2001, Object Management Group, Inc.

USE OF SPECIFICATION - TERMS, CONDITIONS & NOTICES

The material in this document details an Object Management Group specification in accordance with the terms, conditions and notices set forth below. This document does not represent a commitment to implement any portion of this specification in any company's products. The information contained in this document is subject to change without notice.

LICENSES

The companies listed above have granted to the Object Management Group, Inc. (OMG) a nonexclusive, royalty-free, paid up, worldwide license to copy and distribute this document and to modify this document and distribute copies of the modified version. Each of the copyright holders listed above has agreed that no person shall be deemed to have infringed the copyright in the included material of any such copyright holder by reason of having used the specification set forth herein or having conformed any computer software to the specification.

Subject to all of the terms and conditions below, the owners of the copyright in this specification hereby grant you a fully-paid up, non-exclusive, nontransferable, perpetual, worldwide license (without the right to sublicense), to use this specification to create and distribute software and special purpose specifications that are based upon this specification, and to use, copy, and distribute this specification as provided under the Copyright Act; provided that: (1) both the copyright notice identified above and this permission notice appear on any copies of this specification; (2) the use of the specifications is for informational purposes and will not be copied or posted on any network computer or broadcast in any media and will not be otherwise resold or transferred for commercial purposes; and (3) no modifications are made to this specification. This limited permission automatically terminates without notice if you breach any of these terms or conditions. Upon termination, you will destroy immediately any copies of the specifications in your possession or control.

PATENTS

The attention of adopters is directed to the possibility that compliance with or adoption of OMG specifications may require use of an invention covered by patent rights. OMG shall not be responsible for identifying patents for which a license may be required by any OMG specification, or for conducting legal inquiries into the legal validity or scope of those patents that are brought to its attention. OMG specifications are prospective and advisory only. Prospective users are responsible for protecting themselves against liability for infringement of patents.

GENERAL USE RESTRICTIONS

Any unauthorized use of this specification may violate copyright laws, trademark laws, and communications regulations and statutes. This document contains information which is protected by copyright. All Rights Reserved. No part of this work covered by copyright herein may be reproduced or used in any form or by any means--graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems--without permission of the copyright owner.

DISCLAIMER OF WARRANTY

WHILE THIS PUBLICATION IS BELIEVED TO BE ACCURATE, IT IS PROVIDED "AS IS" AND MAY CONTAIN ERRORS OR MISPRINTS. THE OBJECT MANAGEMENT GROUP AND THE COMPANIES LISTED ABOVE MAKE NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS PUBLICATION, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF TITLE OR OWNERSHIP, IMPLIED WARRANTY OF

MERCHANTABILITY OR WARRANTY OF FITNESS FOR A PARTICULAR PURPOSE OR USE. IN NO EVENT SHALL THE OBJECT MANAGEMENT GROUP OR ANY OF THE COMPANIES LISTED ABOVE BE LIABLE FOR ERRORS CONTAINED HEREIN OR FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL, CONSEQUENTIAL, RELIANCE OR COVER DAMAGES, INCLUDING LOSS OF PROFITS, REVENUE, DATA OR USE, INCURRED BY ANY USER OR ANY THIRD PARTY IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS MATERIAL, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

The entire risk as to the quality and performance of software developed using this specification is borne by you. This disclaimer of warranty constitutes an essential part of the license granted to you to use this specification.

RESTRICTED RIGHTS LEGEND

Use, duplication or disclosure by the U.S. Government is subject to the restrictions set forth in subparagraph (c) (1) (ii) of The Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 or in subparagraph (c)(1) and (2) of the Commercial Computer Software - Restricted Rights clauses at 48 C.F.R. 52.227-19 or as specified in 48 C.F.R. 227-7202-2 of the DoD F.A.R. Supplement and its successors, or as specified in 48 C.F.R. 12.212 of the Federal Acquisition Regulations and its successors, as applicable. The specification copyright owners are as indicated above and may be contacted through the Object Management Group, 250 First Avenue, Needham, MA 02494, U.S.A.

TRADEMARKS

The OMG Object Management Group Logo®, CORBA®, CORBA Academy®, The Information Brokerage®, XMI® and IOP® are registered trademarks of the Object Management Group. OMG™, Object Management Group™, CORBA logos™, OMG Interface Definition Language (IDL)™, The Architecture of Choice for a Changing World™, CORBA services™, CORBA facilities™, CORBA med™, CORBA net™, Integrate 2002™, Middleware That's Everywhere™, UML™, Unified Modeling Language™, The UML Cube logo™, MOF™, CWM™, The CWM Logo™, Model Driven Architecture™, Model Driven Architecture Logos™, MDA™, OMG Model Driven Architecture™, OMG MDA™ and the XMI Logo™ are trademarks of the Object Management Group. All other products or company names mentioned are used for identification purposes only, and may be trademarks of their respective owners.

COMPLIANCE

The copyright holders listed above acknowledge that the Object Management Group (acting itself or through its designees) is and shall at all times be the sole entity that may authorize developers, suppliers and sellers of computer software to use certification marks, trademarks or other special designations to indicate compliance with these materials.

Software developed under the terms of this license may claim compliance or conformance with this specification if and only if the software compliance is of a nature fully matching the applicable compliance points as stated in the specification. Software developed only partially matching the applicable compliance points may claim only that the software was based on this specification, but may not claim compliance or conformance with this specification. In the event that testing suites are implemented or approved by Object Management Group, Inc., software developed using this specification may claim compliance or conformance with the specification only if the software satisfactorily completes the testing suites.

ISSUE REPORTING

All OMG specifications are subject to continuous review and improvement. As part of this process we encourage readers to report any ambiguities, inconsistencies, or inaccuracies they may find by completing the Issue Reporting Form listed on the main web page <http://www.omg.org>, under Documents & Specifications, Report a Bug/Issue.

Contents

1.	Interface Specification	1-1
1.1	Introduction	1-1
1.2	Interface Introduction	1-1
1.3	Description of Data Items	1-2
1.3.1	Description of Data Items of CATEGORY 030 and IDL Definition	1-2
2.	Interface Definition Source	2-1
	Appendix A - Acronyms and Abbreviations	A-1

Preface

About the Object Management Group

Under the terms of the collaboration between OMG and The Open Group, this document is a candidate for adoption by The Open Group, as an Open Group Technical Standard. The collaboration between OMG and The Open Group ensures joint review and cohesive support for emerging object-based specifications.

Object Management Group

The Object Management Group, Inc. (OMG) is an international organization supported by over 600 members, including information system vendors, software developers and users. Founded in 1989, the OMG promotes the theory and practice of object-oriented technology in software development. The organization's charter includes the establishment of industry guidelines and object management specifications to provide a common framework for application development. Primary goals are the reusability, portability, and interoperability of object-based software in distributed, heterogeneous environments. Conformance to these specifications will make it possible to develop a heterogeneous applications environment across all major hardware platforms and operating systems.

OMG's objectives are to foster the growth of object technology and influence its direction by establishing the Object Management Architecture (OMA). The OMA provides the conceptual infrastructure upon which all OMG specifications are based. More information is available at <http://www.omg.org/>.

The Open Group

The Open Group, a vendor and technology-neutral consortium, is committed to delivering greater business efficiency by bringing together buyers and suppliers of information technology to lower the time, cost, and risks associated with integrating new technology across the enterprise.

The mission of The Open Group is to drive the creation of boundaryless information flow achieved by:

- Working with customers to capture, understand and address current and emerging requirements, establish policies, and share best practices;
- Working with suppliers, consortia and standards bodies to develop consensus and facilitate interoperability, to evolve and integrate specifications and open source technologies;
- Offering a comprehensive set of services to enhance the operational efficiency of consortia; and
- Developing and operating the industry's premier certification service and encouraging procurement of certified products.

The Open Group has over 15 years experience in developing and operating certification programs and has extensive experience developing and facilitating industry adoption of test suites used to validate conformance to an open standard or specification. The Open Group portfolio of test suites includes tests for CORBA, the Single UNIX Specification, CDE, Motif, Linux, LDAP, POSIX.1, POSIX.2, POSIX Realtime, Sockets, UNIX, XPG4, XNFS, XTI, and X11. The Open Group test tools are essential for proper development and maintenance of standards-based products, ensuring conformance of products to industry-standard APIs, applications portability, and interoperability. In-depth testing identifies defects at the earliest possible point in the development cycle, saving costs in development and quality assurance.

More information is available at <http://www.opengroup.org/> .

OMG Documents

The OMG Specifications Catalog is available from the OMG website at:

http://www.omg.org/technology/documents/spec_catalog.htm

The OMG documentation is organized as follows:

OMG Modeling Specifications

Includes the UML, MOF, XMI, and CWM specifications.

OMG Middleware Specifications

Includes CORBA/IIOP, IDL/Language Mappings, Specialized CORBA specifications, and CORBA Component Model (CCM).

Platform Specific Model and Interface Specifications

Includes CORBA services, CORBA facilities, OMG Domain specifications, OMG Embedded Intelligence specifications, and OMG Security specifications.

Obtaining OMG Documents

The OMG collects information for each book in the documentation set by issuing Requests for Information, Requests for Proposals, and Requests for Comment and, with its membership, evaluating the responses. Specifications are adopted as standards only when representatives of the OMG membership accept them as such by vote. (The policies and procedures of the OMG are described in detail in the *Object Management Architecture Guide*.) OMG formal documents are available from our web site in PostScript and PDF format. Contact the Object Management Group, Inc. at:

OMG Headquarters
250 First Avenue
Needham, MA 02494
USA
Tel: +1-781-444-0404
Fax: +1-781-444-0320
pubs@omg.org
<http://www.omg.org>

Typographical Conventions

The type styles shown below are used in this document to distinguish programming statements from ordinary English. However, these conventions are not used in tables or section headings where no distinction is necessary.

Helvetica bold - OMG Interface Definition Language (OMG IDL) and syntax elements.

Courier bold - Programming language elements.

Helvetica - Exceptions

Terms that appear in italics are defined in the glossary. Italic text also represents the name of a document, specification, or other publication.

Acknowledgments

The following company submitted this specification:

- THALES ATM

Reference Documents

OMG, Surveillance Request for Proposal, *transprt/00-01-09*

Eurocontrol, Radar Surveillance in En-Route Airspace and Major Terminal Areas, Edition 1.0 from March 1997, SUR.ET1.ST01.1000-STD-01-01

Demystifying CNS/ATM, CANSO CNS/ATM Working Group, Final Version (June 1999)

Overall CNS/ATM architecture for EATCHIP, ASE.ET1.ST02-ADD-01-00, version 1.0, 18/08/1997

Interface Specification, Application of ASTERIX to ARTAS (DIS/SUR/ARTAS.ASTX.015), version 6.0, 10th Sept. 99)

Surveillance Development Roadmap, Eurocontrol, working draft, edition 0-18, 4 December 2000

System/Segment Specification for ARTAS, version 6.3, 10 October 2000

Interface Specification

1

1.1 Introduction

This document refers to the CORBA ATC architecture white paper. The context of this specification is relative to ATC components identified in the CORBA ATC architecture reference model.

This specification is a subset of the Surveillance Manager Interface, it represents the basic client interface. Enhanced client interfaces along with server side interfaces will be addressed in future specifications.

1.2 Interface Introduction

The interface is based on AXTERIX category 030. This is related to the Exchange of Air Situation Pictures.

The Interface definition is a translation of AXTERIX syntax in IDL syntax. The data CATEGORY 030 : Exchange of Air Situation Pictures defines all items that can be transmitted by the Surveillance Server to its User(s) in the frame of any Track Information Service.

This interface shall be used with typed event COSservice. The Surveillance service is defined according to the service name defined in the COSnaming service.

Data Item Reference Number	Description
I030/050	ARTAS TRACK NUMBER
I030/060	TRACK MODE 3/A
I030/070	TIME OF LAST UPDATE
I030/080	ARTAS TRACK STATUS

I030/90	TRACK QUALITY
	CALCULATED TRACK POSITION (LATLONG)
I030/130	CALCULATED TRACK ALTITUDE
I030/140	LAST MESURED MODE C
I030/180	CALCULATED TRACK VELOCITY (POLAR)
I030/200	MODE OF FLIGHT
I030/220	CALCULATED RATE OF CLIMB/DESCENT
I030/240	CALCULATED RATE OF TURN
I030/270	LOCAL TRACK NUMBER
I030/290	PLOT AGE
I030/360	MESURED POSITION

This subset has been defined in the Avenue research program.

1.3 Description of Data Items

1.3.1 Description of Data Items of CATEGORY 030 and IDL Definition

1.3.1.1 I030/050 :ARTAS TRACK NUMBER

Definition: Identification of an ARTAS track.

```
typedef unsigned long Natural;
    const Natural MIN_NATURAL = 0;
    const Natural MAX_NATURAL = 2147483647;
```

```
typedef Natural TrackId;
    const TrackId NULL_TRACK_ID = 0;
```

1.3.1.2 I030/060 :TRACK MODE 3/A

Definition: Mode 3/A identity associated to the track.

```
typedef octet SsrCode[4];

struct RealModeA
{
    boolean    is_validated;
    boolean    is_garbled;
    boolean    is_track_mode_changed;
    SsrCode    ssr_code;
};
```

1.3.1.3 I030/070 :TIME OF LAST UPDATE

Definition: Absolute time stamping of the information provided in the track message, in the form of elapsed time since last midnight.

```
typedef SFloat ADuration;
    const          ADuration DAY_IN_SECONDS = 86400.0;
```

1.3.1.4 I030/080 :ARTAS TRACK STATUS

Definition: Status of an ARTAS track.

```
enum TargetType
{
    TEST_TARGET,
    LIVE_TARGET,
    UNKNOWN_TARGET
};

enum TrackType
{
    TENTATIVE_TRACK,
    CONFIRMED_TRACK,
    UNKNOWN_TRACK
};

enum RadarUpdate
{
    PR_SSR_TRACK,
    PR_MULTITRACK,
    SSR_MULTITRACK,
    PR_SSR_MONOTRACK,
    SSR_MONOTRACK,
    PR_MONOTRACK,
    UNKNOWN_RADAR_UPDATE
};

enum SlantRangeCode
{
    SLR_USING_MODEC,

    SLR_USING_CALCULATED_HEIGHT,

    SLR_USING_ASSUMED_HEIGHT,

    SLR_NOT_CORRECTED,
    UNKNOWN_SLANT_RANGE_CODE
};
```

```

enum SpecialCode
{
    DEFAULT_SPECIAL_CODE,
    UNLAWFUL_INTERFERENCE,
    RADIOCOMMS_FAILURE,
    EMERGENCY,
    UNKNOWN_SPECIAL_CODE
};

struct TrackStatus
{
    TargetType           target_type;
    TrackType           track_type;
    boolean              uses_aircraft_derived_data;
    boolean              is_coasted;
    boolean              is_coasted_apart_from_lateral_trac;

    RadarUpdate         radar_update;
    boolean              is_terminated;
    boolean              is_created;
    SlantRangeCode      slant_range_code;
    SpecialCode         special_code;
    boolean              is_amalgamated;
    boolean              is_spi_set;
    boolean              is_military_emergency;
};

```

1.3.1.5 130/90 TRACK QUALITY

Definition : track quality.
 unsigned short track_quality;

1.3.1.6 CALCULATED TRACK POSITION

```

// LAT LONG COORDINATES
typedef SFloat Azimuths;
typedef Azimuths LatAzimuths;
typedef Azimuths LongAzimuths;

enum Hemisphere {
    NORTH,
    SOUTH
};

enum LatLongDirection {
    EAST,
    WEST
};

struct Latitude {

```

```

    Hemisphere      lat_hemis;
    LatAzimuths     lat_azim;
};

    struct Longitude {
    LatLongDirection long_dir;
    LongAzimuths     long_azim;
};

// 2D Position in a LAT/LONG coordinates system
    struct LatLong2DPosition {
    Latitude          latit;
    Longitude         longit;
};

```

1.3.1.7 1030/130 : CALCULATED TRACK ALTITUDE

Definition: Calculated altitude of an aircraft.

```

enum CalculationMode
{
    THREED_HEIGHT,
    TRIANGULATED_HEIGHT,
    FROM_COVERAGE_HEIGHT,
    ASSUMED_HEIGHT,
    UNKNOWN_CALCILATION_MODE
};

typedef long Integer;

typedef Integer      Feet;

    struct CalculatedTrackAltitude
    {
    CalculationMode  calculation_mode;
    Feet             track_altitude;
};

```

1.3.1.8 130/140 : LAST MEASURED MODE C

Definition : Mode C code of the last nearest neighbor plot containing a Mode C and used to update the track.

```

typedef boolean     Is_valid_Mode_C_code;
typedef boolean     Is_garbled_information;

    struct LastMesuredModeC
    {
    boolean          Is_valid_Mode_C_code;
    boolean          Is_garbled_information;
};

```

```

    Feet    Mode_C;
};

```

1.3.1.9 I030/180 :CALCULATED TRACK VELOCITY (POLAR)

Definition: Calculated track velocity expressed in polar co-ordinates.

Velocity-heading component is relative to True North.

```

typedef float SFloat;
typedef SFloat Knots;
    typedef SFloat Azimuths;

    struct TrackVelocity
    {
        Knots    groundspeed;
        Azimuths heading;
    };

    union OptTrackVelocity switch (boolean) {
    case TRUE :
        TrackVelocity track_velocity;
    case FALSE :
        Empty field;
    };

```

1.3.1.10 I030/200 :MODE OF FLIGHT

Definition: Calculated Mode-of-Flight of an aircraft.

```

struct Tendencies {
    VerticalTendency    vertical;
    HorizontalTendency  horizontal;
    SpeedTendency       speed;
};

```

1.3.1.11 I030/220 :CALCULATED RATE OF CLIMB/DESCENT

Definition: Calculated rate of Climb/Descent of an aircraft.

1.3.1.12 I030/240 :CALCULATED RATE OF TURN

Definition: Calculated Rate of Turn expressed in degrees per second.

1.3.1.13 I30/270 :LOCAL TRACK

Definition : The local track number is a value representing a unique reference to a track record within the track data-base of a radar local tracker.

```

typedef unsigned long Natural;

```



```
typedef Natural LocalTrackNumber;
```

1.3.1.14 *i130/290 :PLOT AFGE*

Definition : A set of plot related ages.

```
typedef float Duration;
```

```
typedef Duration SecondDuration;
struct PlotAge
```

```
{
    SecondDurationmodeA_age;

    SecondDurationmodeC_age;
};
```

1.3.1.15 *130/360 :MEASURED POSITION*

Definition : Measured position of an aircraft

```
struct MesuredPositon
```

```
{
    Miles           Rho;
    Azimuths       Theata;
};
```

1.3.1.16 *I030 Track*

Definition: State Vector and list of radar track

```
struct StateVector
```

```
{
    ADuration           last_update_time;           // Asterix Equivalent // I030/070
    TrackStatus        track_status;               // I030/080
    unsigned short     track_quality;              // I030/090

    LatLong2DPosition  latlong_position;

    CalculatedTrackAltitude  calculated_track_altitude; // I030/130

    LastMesuredModeC      last_mesured_mode_C;       // I030/140

    Feet                  measured_mode_c;           // I030/150

    Tendencies           tendencies;                 // I030/200
    FeetPerMinute        rate_of_climb_descent;      // I030/220
    DegreesPerSecond     rate_of_turn;               // I030/240
};
```

```
PlotAge          Plot_Age;          // I030/290
MesuredPositon   Mesured_Position; // I030/360
};

struct RadarTrack
{
TrackId          track_id;          // I030/050
LocalTrackNumber local_track_number; // I030/270
StateVector      state_vector;
};

typedef sequence<RadarTrack> RadarTracksList;
```

1.3.1.17 Surveillance Manager interface operation

Definition: Interface to be implemented by a typed push consumer.

```
interface SurveillanceManagerAsterix30
{
void TracksUpdate(in RadarTracksList trackEvent);
};
```

Interface Definition Source

2

```
module SurveillanceAsterix30
{
    //
    // I030/080
    //
    enum TargetType
    {
        TEST_TARGET,
        LIVE_TARGET,
        UNKNOWN_TARGET
    };

    enum TrackType
    {
        TENTATIVE_TRACK,
        CONFIRMED_TRACK,
        Unknown_Track
    };

    enum RadarUpdate
    {
        PR_SSR_TRACK,
        PR_MULTITRACK,
        SSR_MULTITRACK,
        PR_SSR_MONOTRACK,
        SSR_MONOTRACK,
        PR_MONOTRACK,
        UNKNOWN_RADAR_UPDATE
    };

    enum SlantRangeCode
    {
```

```
        SLR_USING_MODEC,  
        SLR_USING_CALCULATED_HEIGHT,  
        SLR_USING_ASSUMED_HEIGHT,  
        SLR_NOT_CORRECTED,  
        UNKNOWN_SLANT_RANGE_CODE  
};  
  
enum SpecialCode  
{  
    DEFAULT_SPECIAL_CODE,  
    UNLAWFUL_INTERFERENCE,  
    RADIOCOMMS_FAILURE,  
    EMERGENCY,  
    UNKNOWN_SPECIAL_CODE  
};  
  
struct TrackStatus  
{  
    TargetType          target_type;  
    TrackType           track_type;  
    boolean             uses_aircraft_derived_data;  
    boolean             is_coasted;  
    boolean             is_coasted_apart_from_lateral_trac;  
  
    RadarUpdate        radar_update;  
    boolean             is_terminated;  
    boolean             is_created;  
    SlantRangeCode     slant_range_code;  
    SpecialCode        special_code;  
    boolean             is_amalgamated;  
    boolean             is_spi_set;  
    boolean             is_military_emergency;  
};  
  
//  
//I030/100  
//  
  
typedef float SFloat;  
typedef SFloat Miles;  
  
struct XY2DPosition  
{  
    Miles x_pos;  
    Miles y_pos;  
};  
  
// LAT LONG COORDINATES  
typedef SFloat Azimuths;
```

```
typedef Azimuths    LatAzimuths;
typedef Azimuths    LongAzimuths;

enum Hemisphere {
    NORTH,
    SOUTH
};

enum LatLongDirection {
    EAST,
    WEST
};

struct Latitude {
    Hemisphere    lat_hemis;
    LatAzimuths   lat_azim;
};

struct Longitude {
    LatLongDirection    long_dir;
    LongAzimuths        long_azim;
};

// 2D Position in a LAT/LONG coordinates system

struct LatLong2DPosition {
    Latitude    latit;
    Longitude   longit;
};

//
// I030/130
//

enum CalculationMode
{
    THREEED_HEIGHT,
    TRIANGULATED_HEIGH,
    FROM_COVERAGE_HEIGTH,
    ASSUMED_HEIGHT,
    UNKNOWN_CALCILATION_MODE
};

typedef long Integer;

typedef Integer Feet;
```

```
struct CalculatedTrackAltitude
{
    CalculationMode    calculation_mode;
    Feet               track_altitude;
};

typedef boolean    Is_valid_Mode_C_code;
typedef boolean    Is_garbled_information;

struct LastMesuredModeC
{
    boolean         Is_valid_Mode_C_code;
    boolean         Is_garbled_information;
    Feet           Mode_C;
};

//
// I030/180
//

typedef SFloat    Knots;

struct TrackVelocity
{
    Knots          groundspeed;
    Azimuths       heading;
};

//
// I030/200
//

enum VerticalTendency
{
    CLIMB,
    STEADY,
    DESCENT
};

enum HorizontalTendency
{
    LEFT,
    STRAIGHT,
    RIGHT
};

enum SpeedTendency
```

```
{
    ACCELERATE,
    DECELERATE,
    MAINTAIN
};

struct Tendencies
{
    VerticalTendency      vertical;
    HorizontalTendency    horizontal;
    SpeedTendency         speed;
};

//
// 1030/220
//

typedef SFloat FeetPerMinute;

//
// 1030/240
//

typedef SFloat DegreesPerSecond;

typedef SFloat ADuration;

typedef unsigned long Natural;

typedef Natural LocalTrackNumber;

typedef float          Duration;
typedef Duration      SecondDuration;

struct PlotAge
{
    SecondDuration      modeA_age;
    SecondDuration      modeC_age;
};

struct MeasuredPosition
{
    Miles              Rho;
    Azimuths          Theata;
};
```

```
struct StateVector
{
    ADuration            last_update_time;           // Asterix Equivalent // I030/070
    TrackStatus          track_status;               // I030/080
    unsigned short       track_quality;              // I030/090
    LatLong2DPosition    latlong_position;           // I030/100
    CalculatedTrackAltitude calculated_track_altitude; // I030/130
    LastMesuredModeC     last_mesured_mode_C;        // I030/140
    TrackVelocity         track_velocity;             // I030/180
    Tendencies           track_tendencies;           // I030/200
    FeetPerMinute         rate_of_climb_descent;     // I030/220
    DegreesPerSecond     rate_of_turn;               // I030/240
    PlotAge               Plot_Age;                  // I030/290
    MesuredPosition       Mesured_Position;          // I030/360
};

//
// I030/50
//

typedef Natural TrackId;

const TrackId NULL_TRACK_ID = 0;

typedef sequence<TrackId> TrackIdList;

struct RadarTrack
{
    TrackId            track_id;           // I030/050
    LocalTrackNumber local_track_number;   // I030/270
    StateVector        state_vector;
};

typedef sequence<RadarTrack> RadarTracksList;

interface SurveillanceManagerAsterix30
{
    void TracksUpdate(in RadarTracksList trackEvent);
};
};
```


Acronyms and Abbreviations

A

Acronym/Abbreviation	Description
°	Degree (angle)
ARTAS	ATC Radar Tracker And Server
ASTERIX	All Purpose STructured Eurocontrol Radar Information EXchange
ATC	Air Traffic Control
AVENUE	<i>ATM Validation ENvironment for Use towards EATMS, TRANSPORT RESEARCH PROGRAMME, DG7 - TRANSPORT/AIR TASK N° 4.1.3/24A</i>
CAT	Data Category
EATCHIP	European Air Traffic Control Harmonisation and Integration Programme
EOP	End of Picture
EWPD	EATCHIP Work Programme Document
f	Scaling factor
FRN	Field Reference Number
FSPEC	Field Specification
FX	Field Extension Indicator
ICAO	International Civil Aviation Organization
LEN	Length Indicator
LSB	Least Significant Bit

NM	Nautical Mile, unit of distance (6 080 feet)
RDP	Radar Data Processing (system)
REP	Field Repetition Indicator
RFS	Random Field Sequencing
RSSP	Radar Systems Specialist Panel
s	second, unit of time
SAC	System Area Code
SIC	System Identification Code
SOP	Start Of Picture
SP	Special Purpose Indicator
SPF	Standard Precision Format
STFRDE	Surveillance Task Force on Radar Data Exchange
UAP	User Application Profile (see Definitions)
UTC	Coordinated Universal Time

A

ARTAS track 1-2
ARTAS track status 1-2
ATC components
Avenue research program 1-1
AXTERIX category 03 01-1
AXTERIX syntax 1-1

C

Calculated altitude of an aircraft 1-4
CATEGORY 030
 Exchange of Air Situation Pictures 1-1
client interface 1-1
Climb/Descent rate 1-6
CORBA
 contributors 1-vii
 documentation set 1-vi
CORBA ATC architecture 1-1
COSService 1-1

E

elapsed time 1-2

I

IDL syntax 1-1

L

Last measured Mode C 1-5

Local track 1-6

M

Measured position 1-7
Mode 3/A 1-2
Mode-of-Flight 1-6

O

Object Management Group 1-v
 address of 1-vii

P

Plot AFGE 1-6

R

Radar track 1-7
Rate of Turn 1-6

S

State Vector 1-7
subset 1-1

T

Track Information Service 1-1
Typed push consume 1-8

V

Velocity expressed in polar co-ordinates 1-5

Surveillance User Interface Specification, v1.0

Reference Sheet

This is the first formal version of this specification.

OMG documents used to create this specification:

- Submission document: `transprt/01-06-01`
- FTF Report: `dtc/02-10-04`
- Proposed available specification: `dtc/02-10-05`

