SAE The Engineering Society For Advancing Mobility Land Sea Air and Space®
INTERNATIONAL
400 Commonwealth Drive, Warrendale, PA 15096-0001

SURFACE VEHICLE STANDARD

SAE J2354

ISSUED NOV1999

Issued 1999-11

Submitted for recognition as an American National Standard

Message Sets for Advanced Traveler Information System (ATIS)

TABLE OF CONTENTS

1. Scope 2

2. References 2

3. Definitions 3

4. Message Descriptions 7
4.1 Introductions 7
4.2 Settings Messages 7
4.3 Directory Services Messages 10
4.4 Mayday Messages 16
4.5 Parking Messages 16
4.6 Traveler Information Messages 21
4.7 Trip Guidance Messages 25

5. ASN.1 Messages 29
5.1 Global Messages 30
5.2 Settings Messages 32
5.3 Directory Services Messages 34
5.4 Mayday Messages 36
5.5 Parking Messages 36
5.6 Traveler Information Messages 38
5.7 Trip Guidance Messages 42

6. Concept of Operations 44

7. Informative Section 61

Appendix A Acknowledged Messages: Request-Reply Message Sequences 70
Appendix B Unacknowledged Messages: By Message Category 72

TO PLACE A DOCUMENT ORDER; (724) 776-4970 FAX: (724) 776-0790
SAE WEB ADDRESS http://www.sae.org

1. ***Scope—***This SAE Standard describes standardized medium-independent messages needed by information service providers for Advanced Traveler Information Systems (ATIS). The messages contained herein address all stages of travel (pre-trip and en-route), all types of travelers (drivers, passengers), all categories of information, and all platforms for delivery of information (in-vehicle, portable devices, kiosks, etc.).

1.1 **Purpose—**The purpose of this document is to enable the rapid development of the market for ATIS products and services.

2. ***References***

2.1 **Applicable Publications—**The following publications form a part of the specification to the extent specified herein. Unless otherwise indicated, the latest revision of SAE publications shall apply.

2.1.1 SAE PUBLICATIONS—Available from SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

SAE J2374—Information Report based on Location Reference Message Specification, Revision B (MDI), May 22, 1997
SAE J2313—Draft Recommended Practice: On-Board Land Vehicle Mayday Reporting Interface

2.1.2 ISO PUBLICATIONS—Available from ANSI, 11 West 42nd Street, New York, NY 10036-8002.

ISO 3166-1—Codes for the representation of names of countries and their subdivisions—Part 1: Country Codes
ISO 3379-1983—Standard vehicle identification numbering (VIN) system

2.1.3 ITE PUBLICATIONS—Available from ITE, 525 School Street, SW, Suite 410, Washington, DC 20024 USA.

TCIP, Version 1.1 of the Recommended Standard for the Transit Communications Interface Profiles.
TMDD, Standard for Functional Level Traffic Management Data Dictionary, Standard Number TM.01

2.2 **Related Publications—**The following publications are provided for information purposes only and are not a required part of this document.

Atlanta Traveler Information Showcase, Appendixes to Final Report, Battelle, 1996
DATEX, Task Force on Traffic and Travel Data Exchange, Technical Documents, Commission of the European Communities, RTD Transport Telematics Application Programme, December 1996
IEEE P1488 Data Dictionary Standard, versions 1-7
IEEE P1489 Draft Standard for Data Dictionary for Intelligent Transportation Systems, version. 0.1.0 (Part 1)
International Traveler Information Interchange Standard (ITIIS), Draft ATIS Data Dictionary, Version 1, June 20, 1996
National ITS Architecture, Federal Highway Administration, U.S. Department of Transportation, 1996
Priority Elements for Traveler Information, Generated from the ATIS Interoperability Summit, May 1-2, 1996
San Antonio TransGuide In-Vehicle Navigation System High Speed FM Subcarrier Communications Protocol, Version 1.0, Texas Department of Transportation, March 26, 1997
Towards Traveler Information Message List Standards, SAE Message List Workshop, March 21-23, 1994.
Traffic Management Data Dictionary, Sections 1 and 2, May 12, 1997
Transcal IRTIS Interface Control Documents, February 3, 1997 and March 11, 1997
TravelTIP Preliminary Design Report, Orange County Transportation Authority, June 21, 1996
TravInfo Bay Area Advanced Traveler Information System, Metropolitan Transportation Commission (Oakland, CA), Detailed Design for Increment 2, January 31, 1996
TravInfo Bay Area Advanced Traveler Information System, Metropolitan Transportation Commission (Oakland, CA), Registered Participants Technical Packet, Version 1.0, March 25, 1997

Yosemite Area Traveler Information Project, Draft Technical Memoranda 2 and 4: Database Architecture Options and System Architecture Alternatives and Evaluation, October 24, 1994. Prepared by National Engineering Technology Corporation
ISO/IEC 8824:1998 Abstract Syntax Notation One (ASN.1)

3. Definitions

3.1 Address Location (or matching)—Translating a user-oriented place specification (e.g., street address, intersection, vanity address, named place) to a specific object (node or link) in a database. (Schuman, 1993).

3.2 Address Range—The range of street numbers associated with a street or a particular name, usually within a given area.

3.3 Altitude—Elevation above or below a reference datum, as defined in FIPSPUB 70-1; the z-value in a spatial address. See also elevation.

3.4 Application-Specific Data Dictionary—A data dictionary specific to a particular implementation of an ITS application.

3.5 Area—A generic term for a bounded, continuous, two-dimensional object that may or may not include its boundary. (NIST, 1992).

3.6 ATIS—Advanced Traveler Information Systems.

3.7 ATMS—Advanced Transportation Management Systems.

3.8 Attribute—Any documenting characteristic of any entity.

3.9 Cartesian Coordinates—A two-dimensional x,y location of a point on a plane in relation to two intersecting straight lines (axes). If the axes are perpendicular to each other, the coordinates are rectangular; if not, they are oblique. By convention, the x-axis measures the horizontal distance and the y-axis measures the vertical distance from the origin point of intersection. An x,y coordinate (or “coordinate pair”) defines every point on the plane. Relative measures of distance, area and direction are constant throughout the Cartesian coordinate plane.

3.10 Classification Scheme—A scheme for the arrangement or division of entities into groups based on properties which the entities have in common.

3.11 Complex Intersection—The intersection of roadways involving offsets, central medians, or some combination of offsets and central medians which can generate multiple nodes for representation.

3.12 Concept—A unit of thought constituted through abstraction on the basis of characteristics common to a group of entities.

3.13 Coordinates—Pairs of numbers expressing horizontal distances along orthogonal axes; alternatively, triplets of numbers measuring horizontal and vertical distances.

3.14 Coordinate System—A reference system for the unique definition of a location of a point in n-dimensional space.

3.15 Data—Representations of static or dynamic entities in a formalized manner suitable for communication, interpretation, or processing by humans or by machines.

3.16 Database—Collection of information structured in an organized way, typically held and maintained in a computer system.

3.17 Data Concept—Any of a group of data dictionary structures defined in this document (e.g., data element, data element concept, entity type, property, value domain) referring to abstractions or things in the natural world that can be identified with explicit boundaries and meaning and whose properties and behavior all follow the same rules.

3.18 Data Dictionary—An information technology for documenting, storing, and retrieving the syntactical form (i.e., representational form) and some semantics of data elements and other data concepts.

3.19 Data Element—A syntactically formal representation of some single unit of information of interest (such as a fact, proposition, observation, etc.) with a singular instance value at any point in time, about some entity of interest (e.g., a person, place, process, property, object, concept, association, state, event). A data element is considered indivisible in a certain context.

3.20 Data Element Concept—An expression of the inherent concept embodied in a data element without regard to the value domain(s) by which it can be physically represented.

3.21 Data Registry—An advanced data dictionary that contains not only data about data elements in terms of their names, representational forms and usage in applications, but also substantial data about the semantics or meaning associated with the data elements as concepts that describe or provide information about real or abstract entities. A data registry may contain abstract data concepts that do not get directly represented as data elements in any application system, but which help in information interchange and reuse both from the perspective of human users and for machine-interpretation of data elements.

3.22 Data Representation—Methods of representing spatial objects in a spatial information system, the most common of which are vectors and tessellations.

3.23 Data Structure—Any construct (including data elements and data concepts) used to represent the contents of a data dictionary.

3.24 Data Type—A classification of the collection of letters, digits, and/or symbols used to encode values of a data element based upon the operations that can be performed on the data element.

3.25 Digital Data—Data represented in a computer-compatible format.

3.26 Digital Map Database—A structured set of digital and alphanumeric data that portray geographic locations and relationships of spatial features. Typically, such structures represent, but are not limited to the digital form of hard copy maps. For example, CAD drawings my be imported into a GIS and considered a form of digital base map.

3.27 Directed Links—Links bounded by start and end points, (i.e., ordered).

3.28 Elevation—A vertical distance below or above a reference surface. Terrain elevation is expressed with reference to mean sea level (MSL).

3.29 Entity—Anything of interest (such as a person, place, process, property, object, concept, association, state, event, etc.) within a given domain of discourse (in this case within the ITS domain of discourse).

3.30 Entity Type—The construct used to represent an entity in the ITS data registry.

3.31 Exchange Standard—A collection of agreements between sender and receiver that enables and assures the receiver's unambiguous understanding of the geographic information that the sender intended.

3.32 Functional-Area Data Dictionary—A data dictionary that is intended to standardize data element syntax, and semantics, within and among application areas within the same functional area.

3.33 Generic Data Element—A data element supertype composed of an entity (type), property, and value domain that remains consistent across its specific application data elements.

3.34 Generic Property Domain—An expression of an approved pairing of a property and a value domain, without regard to any entity type with which it may be associated.

3.35 Geocoding—Process of assigning geographic coordinate locations to objects.

3.36 Geographic Coordinates—The quantities of latitude and longitude which define the position of a point on the Earth with respect to the reference spheroid or ellipsoid.

3.37 Geographic Information System (GIS)—A computerized system for the collection, integration, management, analysis, and display of geographic data.

3.38 Grid—A set of grid cells forming a regular, or nearly regular, tessellation of a surface. The tessellation is regular if formed by repeating the pattern of a regular polygon, such as a square, equilateral triangle, or regular hexagon. The tessellation is nearly regular if formed by repeating the pattern of an “almost” regular polygon such as a rectangle, non-square parallelogram, or non-equilateral triangle.

3.39 Grid Cell—A two-dimensional object that represents the smallest nondivisible element of a grid.

3.40 Ground Control Point—A point of known location that can be recognized on an image or a map and that can be used to calculate the transformation needed for the registration of images or maps. Ground Control Points are related to a known projection for use in geometric transformation.

3.41 Identifier—A means of designating or referring to a specific entity instance.

3.42 Information Service Provider (ISP)—A public or private entity responsible for gathering, fusing, analyzing, and reporting transportation related information to user, including vehicles and non-mobile users.

3.43 Instance—An individual occurrence of an entity which belongs to a particular type of entity.

3.44 Intelligent Transportation Systems (ITS)—Systems that apply modern technology to transportation problems. Another appropriate meaning of the ITS acronym is integrated transportation systems, which stressed that ITS systems will often integrate components and users from many domains, both public and private.

3.45 Interoperability—The ability to share information between heterogeneous applications and systems.

3.46 ITS Databus—An electronic implementation of a device layer where electronics components related to advanced vehicle functions can interoperate.

3.47 Junction—A collection of more than one node that represent a logical feature, such as a complex intersection.

3.48 Legacy Databases—Databases that exist in an organization that must be maintained and used regardless of new technology changes.

3.49 Linear Referencing—Process of identifying location(s) on a transportation network or specific link in a network by specifying a start position, direction, and distance along a particular route.

3.50 Link—A topological connection between two nodes. A link may contain additional intermediate coordinates (shapes points) to better represent the shape of curved features. A link may be direct by ordering its nodes.

3.51 Link ID—An identifier assigned to a link. Link-IDs may be arbitrary, or may be assigned by convention to assure that multiple occurrences of the same ID will not occur within one network or within the universe of similar networks or databases.

3.52 Link Referencing—System which identifies a link in a network, and returns its ID value to an external application.

3.53 Location Referencing System—System of determining the position of an entity relative to other entities or to some external frame of references.

3.54 Map Database—A collection of map data, possibly in digital form, for a region, theme, or sets thereof.

3.55 Media—The physical devices used to record, store, and (or) transmit data.

3.56 Message—A grouping of data elements and message attributes, used to convey information. For the purposes of this document, a message is an abstract description using a message set template; not a specific instance.

3.57 Message Set—A collection of messages based on the ITS functional-area they pertain to.

3.58 Message Set Template—An abstract structure addressing the data and syntax used to specify the requirements and properties of ITS messages, as well as rules for producing message set standards (e.g., conformance statements).

3.59 Meta—A word denoting a description which is one level of abstraction removed from the entity being described.

3.60 Meta Attribute—In a data dictionary or data registry, a documenting characteristic of a data concept.

3.61 Meta Data—Data that defines and describes other data.

3.62 Model Deployment Initiative—One of the joint public-private programs to implement and test integrated ITS systems and infrastructure in the U.S.

3.63 Name—An indexical term used by humans as a means of identifying data elements and other data concepts.

3.64 Point of Interest—A geographic location that is of interest to the transit community.

3.65 Property—A documenting characteristic of an entity type used to group and differentiate individual entities.

3.66 Restricted Maneuver—A prohibition of movement from one roadway (link) to another roadway (link) due to a physical impediment, regional restriction, one-way flow of traffic, or a posted restriction. There may be multiple restrictions pertaining to any link and these restrictions may be limited to a specific time of day and/or day of the week.

3.67 Route—An aggregation of sequentially connected links in a network typically denoting an intended or scheduled path of a transport resource.

3.68 Route Guidance—Delivering real-time driving directions to the driver, based on a determined route and vehicle position or speed.

3.69 Routing—The problem of calculating least-time, least-cost, or other optimized paths (routes) through a road network.

3.70 Semantics—The meaning, including concept(s), associated with a given entity (i.e., any thing).

3.71 Spatial Data—Information about the location, shape, relationships, and attributes of geographic features.

3.72 Syntax—The structure of expressions in a language, and the rules governing the structure of a language.

3.73 Value Domain—An expression of a specific and explicit representation of some information about something of interest within the ITS domain.

4. Message Descriptions

4.1 Introduction—This section describes the messages defined for different categories of ATIS applications and all possible request/reply sequences of these messages. The five groups of ATIS applications are Setting, Directory Services, Parking, Traveler Information, and Trip Guidance. These groups provides travelers and ISPs with the following functionality:

a. Setting—Enables a traveler to send and store information about personal preferences with an ISP; the ISP can then use this information to update and otherwise modify the traveler's request
b. Directory Services—Provides the traveler with electronic "Yellow Pages," a way for ISPs to offer travelers significant, everyday value-added information
c. Parking—Furnishes parking lot and space availability information
d. Traveler Information—Provides information on traffic, incidents, events, weather, environmental conditions (pollution), wide-area travel, and public transit schedules and services
e. Trip Guidance—Enables a traveler to plan a route to a destination, selecting mode of transportation and the dates and specifying the level of information to be returned about all the services and points of interest along a selected route

This section includes a functional description of the messages.

4.2 Settings Messages—Settings are one-way messages from the traveler to the ISP. Settings store information about a traveler's preferences and other static information. Refer to Settings Figure 1.

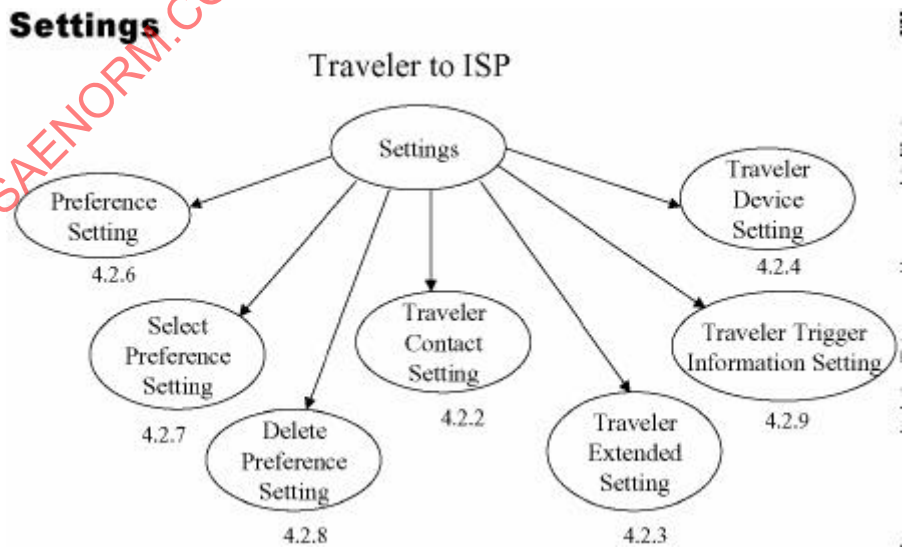

FIGURE 1—SETTINGS

Settings messages can be grouped into three message categories: basic traveler information, traveler preference information, and traveler trigger information.

All settings messages are unacknowledged requests by the traveler. The ISP never replies.

4.2.1 BASIC TRAVELER INFORMATION—Message Sequences A-C shown in Table 1 summarize the three possible Basic Traveler Information request-reply message sequences.

TABLE 1—BASIC TRAVELER INFORMATION REQUEST MESSAGE SEQUENCES.

Message Sequence: (Numbers Indicate Order)	A	B	C
Basic Traveler Information Request Messages (From Traveler To ISP)			
TravelerContactSetting	1		
TravelerExtendedSetting		1	
TravelerDeviceSetting			1
Basic Traveler Information Reply Messages (From ISP To Traveler)			
All requests are unacknowledged. The ISP never replies			

4.2.2 TRAVELER CONTACT SETTING—A Traveler Contact Setting message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting
c. traveler-FirstName—First name of Traveler
d. traveler-LastName—Last name of Traveler
e. contactInformation—The method and means by which the traveler can be contacted
f. atis-DayOfWeek (optional)—Describes the Day of the Week, including a Holiday option
g. contactTimes (optional)—The times for which the Contact Information applies

4.2.3 TRAVELER EXTENDED SETTING—A Traveler Extended Setting message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting
c. setting-ExtendedInformation—ISP-defined additional setting information

4.2.4 TRAVELER DEVICE SETTING—A Traveler Device Setting message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting
c. device-Setting—Device storage capability, operating system, graphics level, etc.
d. device-TransferSpeed—Actual transfer speed, not rated speed

4.2.5 TRAVELER PREFERENCE INFORMATION—Message Sequences A-C shown in Table 2 summarize the three possible Traveler Preference Information request-reply message sequences.

TABLE 2—TRAVELER PREFACE INFORMATION REQUEST MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B	C
Traveler Preference Information Request Messages (From Traveler To ISP)			
PreferenceSetting	1		
SelectPreferenceSetting		1	
DeleteSetting			1
Traveler Preference Information Reply Messages (From ISP To Traveler)			
All requests are unacknowledged. The ISP never replies			

4.2.6 PREFERENCE SETTING—The Preference Setting message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting
c. setting-Type—Allows the user to specify the type of setting, including language, directory services, etc.
d. setting-PreferenceType—Allows the ISP to define additional settings for the user
e. setting-PreferenceSubType—Allows the ISP to define additional settings for the user
f. atis-SearchOperator—Used to specify a conditional operator applying to a key word phrase or list
g. atis-CostPreference—Allows the user to specify a cost preference such as: lowest cost, maximum cost, minimum cost, etc.
h. atis-CostPreferenceAmount—Allows the user to specify the amount to spend on an option

4.2.7 SELECT PREFERENCE SETTING—The Select Preference Setting message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting

4.2.8 DELETE PREFERENCE SETTING—The Delete Setting message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting

4.2.9 TRAVELER TRIGGER INFORMATION—Only one possible Traveler Trigger Information Sequence message sequence exists. The traveler sends a TriggerSetting message to the ISP. There is no ISP acknowledgement. See Table 3.

TABLE 3—TRAVELER TRIGGER INFORMATION REQUEST MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A
Traveler Trigger Information Request Messages (From Traveler To ISP)	
TriggerSetting	1
Traveler Preference Information Reply Messages (From ISP To Traveler)	

TABLE 3—TRAVELER TRIGGER INFORMATION REQUEST MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A
All requests are unacknowledged. The ISP never replies	

The Travel Trigger Information message consists of the following information:

a. traveler-Identity—Identifies a Traveler uniquely—can be used to drive a session and to reference a traveler setting
b. setting-Identity—Unique identity for the user setting
c. location—The Location Reference for a specific trigger event
d. trigger-Event—The event that causes an ISP to contact the traveler
e. start (DateTimePair) (optional)
f. end (DateTimePair) (optional)
g. timeEnforced (optional)—The times during which a trigger should be active
h. trigger-DayOfWeek—Describes the Day of the Week, including a Holiday option (optional)

4.3 Directory Services Messages—Directory Services messages are either a two-way dialog between the traveler and the ISP or a broadcast of summary information by the ISP. A two-way dialog always includes at least one message request by the traveler and an associated message reply from the ISP.

A two-way dialog is typically initiated by a traveler search request message followed by a reply message from the ISP. Based on the content of the ISP reply the traveler may then make an additional request for more detailed information or a request for an appointment.

Refer to the illustrations shown in Directory Services Figures 2 and 3.

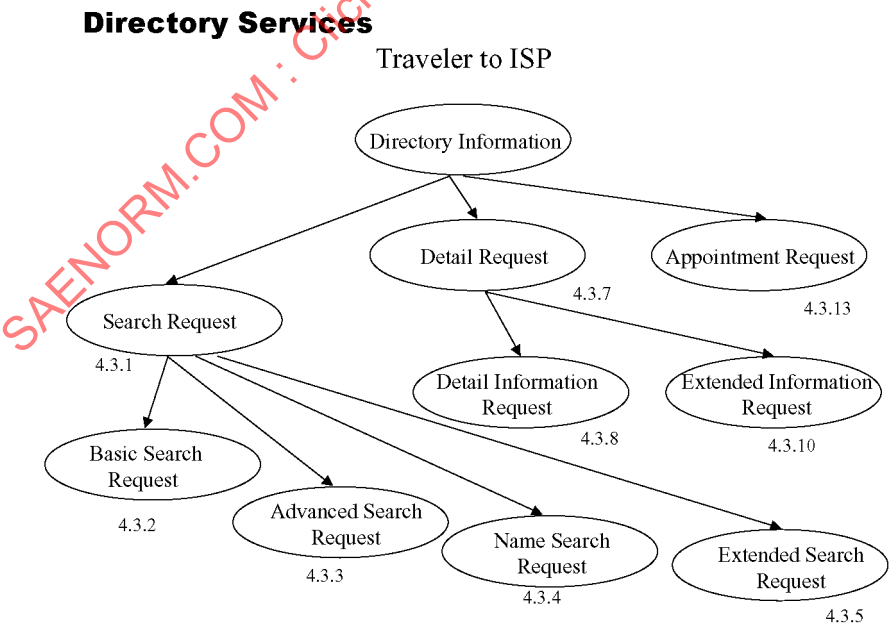

FIGURE 2—DIRECTORY SERVICES

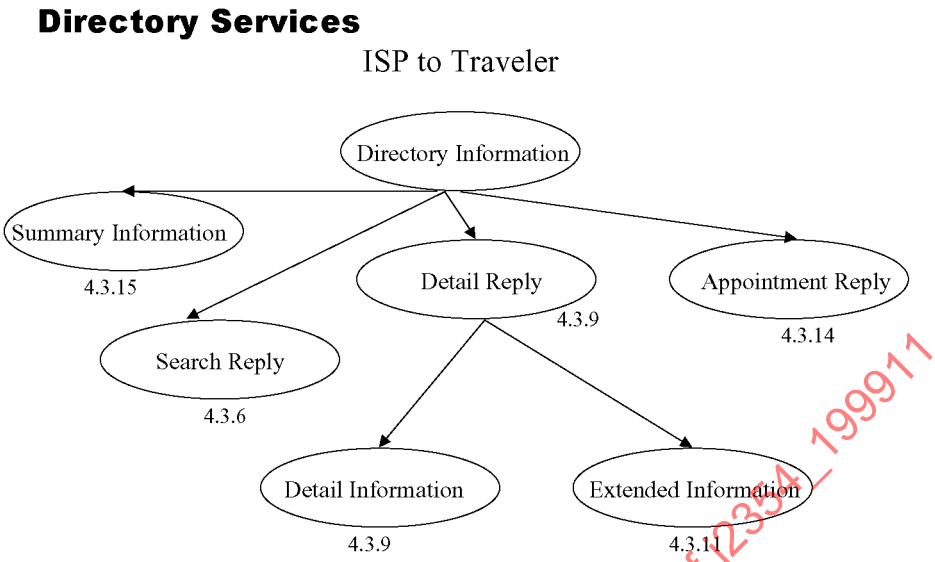

FIGURE 3—DIRECTORY SERVICES

4.3.1 SEARCH MESSAGES—A two-way dialog is typically initiated by the traveler via a search request. Four possible search request message formats are available: Basic Search, Advanced Search, Name Search, and Extended Search.

The ISP responds to a search request with a Search reply if the response contains multiple directory entries – see Message Sequences A to D in Table 4.

The ISP responds to a search request with a Detail reply if the response contains a single directory entry– see Message Sequences E to H in Table 4.

TABLE 4—SEARCH REQUEST-REPLY MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B	C	D	E	F	G	H
Search Request Messages (From Traveler To ISP)								
Basic Search Request	1				1			
Advanced Search Request		1				1		
Name Search Request			1				1	
Extended Search Request				1				1
Search Reply Messages (From ISP To Traveler)								
Search Reply	2	2	2	2				
Detail Reply					2	2	2	2

4.3.2 BASIC SEARCH REQUEST—The Basic Search Request message enables a search to be made on a single directory type and subtype. The provided search boundaries are the radius (about a location) of the search and the maximum number of found entries to return from the search. A Basic Search Request message consists of the following information:

a. DirectoryCoreRequest—A message construct that defines the directory request structure to be used
b. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
c. entry-Location—The geographic location—specified as either a trip route or standard location—of the requestor
d. directorySearch-Radius—Used to describe the permissible radius around a specified location to return Directory entries in a search request
e. requestPair—The type (directoryEntry-Type) and subtype (directoryEntry-SubType) used for the search
f. directorySearch-LimitEntriesReturned—Used to sent a limit on the number of Directory Entries returned by a search request—the ISP may set a lower systemwide limit; this limit is intended to provide a guaranteed limit on the traveler software side
g. directorySearch-Location—Location reference for a search
h. start—(DateTimePair) (optional)
i. end—(DateTimePair) (optional)
j. directoryEntry-LocationType—Type of location reference for a search reply

4.3.3 ADVANCED SEARCH REQUEST—The Advanced Search Request message allows a search to be made on multiple directory types and subtypes. The provided search boundaries are the radius of the search and the maximum number of found entries to return from the search. An Advanced Search Request message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. traveler-Location—Used as part of a Directory Entry Information Request in conjunction with Search Radius—may be current or destination
c. directorySearch-Radius—Used to describe the permissible radius around a specified location to return Directory entries in a search request
d. directorySearch-LimitEntriesReturned—Used to sent a limit on the number of Directory Entries returned by a search request—the ISP may set a lower systemwide limit; this limit is intended to provide a guaranteed limit on the traveler software side
e. requestPair—The type (directoryEntry-Type) and subtype (directoryEntry-SubType) used for the search
f. start—(DateTimePair) (optional)
g. end—(DateTimePair) (optional)
h. directoryEntry-LocationType—Type of location reference for a search reply

4.3.4 NAME SEARCH REQUEST—The Name Search Request message allows a search to be made for a specific directory entry within a single directory type and subtype. The provided search boundaries are the radius of the search and the maximum number of found entries to return from the search. A Name Search Request message consists of the following information:

a. basicRequest—A message construct (DirectoryCoreRequest) that defines the directory request structure to be used
b. directoryEntry-Name—The name of the Directory Entry or its common description for the purposes of describing the Directory Entry to the Traveler
c. start—(DateTimePair) (optional)
d. end—(DateTimePair) (optional)
e. directoryEntry-LocationType—Type of location reference for a search reply

4.3.5 EXTENDED SEARCH REQUEST—The Extended Search Request message allows a search to be made on multiple directory types and subtypes, including Boolean keyword matching. The other provided search boundaries are the radius of the search, the maximum number of found entries to return from the search and a bounding search setting. An Extended Search Request message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. traveler-Location—Used as part of a Directory Entry Information Request in conjunction with Search Radius—may be current or destination
c. directorySearch-NumericRadius—Used to describe the permissible radius around a specified location to return Directory entries in a search request
d. directorySearch-LimitEntriesReturned—Used to sent a limit on the number of Directory Entries returned by a search request—the ISP may set a lower systemwide limit; this limit is intended to provide a guaranteed limit on the traveler software side
e. typePair—The type (directoryEntry-Type) and subtype (directoryEntry-SubType) used for the search
f. keywordPair—The keyword types (directorySearch-Keywords and atis-SearchOperator) used for the search
g. directorySearch-ConstrainByProfile—A flag indicating whether the ISP should use the Traveler Profile to constrain the search
h. directorySearch-OrderEntriesBy—A code indicating the preferred sort variable and sequence
i. start—(DateTimePair) (optional)
j. end—(DateTimePair) (optional)
k. directoryEntry-LocationType—Type of location reference for a search reply

4.3.6 SEARCH REPLY—The Search Reply message consists of the set of directory entries that match the search criteria of the Basic Search request that originated the reply. Each directory entry in the reply set is itself a set that completely describes that directory entry. A Search Reply message consists of the following information:

a. DirectoryReply—The set of DirectoryReplyEntry(s) that defines the reply
b. DirectoryReplyEntry—The following information that describes a single directory entry
c. directoryEntry-Identity—An identifier for the found directory entry
d. directoryEntry-Location—A Location Reference which encapsulates the full location of the Directory Entry for the Traveler—The Location Reference can be requested in a variety of formats depending on the application needs
e. typePair—The type (directoryEntry-Type) and subtype (directoryEntry-SubType) associated with the found directory entry
f. directoryEntry-Name—The Name of the Directory Entry or its common description for purposes of describing the Directory Entry to the Traveler
g. directoryEntry-Phone—Phone number for Directory Entry
h. directoryEntry-ShortDescription—A short text description of the Directory Entry
i. directoryEntry-CapabilityCode—A code used to define the capabilities of the Entry or of the ISP on behalf of the Entry; capabilities include the ability to take electronic reservations and payments

4.3.7 DETAIL REQUEST—The traveler uses Detail message requests to obtain additional information on a directory entry that was previously obtained using one of the search requests. The traveler can make either a basic Detail Information Request or an Extended Information Request (which is not supported for all directory entries). Message Sequences A and B shown in Table 5 summarize the two possible detail message request-replies.

TABLE 5—DETAIL INFORMATION REQUEST-REPLY MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B
Detail Request Messages (From Traveler To ISP)		
Detail Information Request	1	
Extended Information Request		1
Detail Reply Messages (From ISP To Traveler)		
Detail Information Reply	2	
Extended Information Reply		2

4.3.8 DETAIL INFORMATION REQUEST—The Detail Information Request message is sent when a traveler requires detail on a directory. A Detail Information Request message consists of the following information:

a. DirectoryDetailRequest—A message construct that defines the directory request structure to be used
b. directoryEntry-Identity—A code that uniquely identifies the Directory Entry
c. directoryEntry-LocationType—A Location Reference which encapsulates the full location of the Directory Entry for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs.

4.3.9 DETAIL INFORMATION REPLY—The Detail Information Reply message contains basic details about the directory entry specified in the request being replied to. The reply also contains a flag indicating if additional extended detail information is available. A Detail Information Reply message consists of the following information:

a. directoryEntry-Identity—A code that uniquely identifies the Directory Entry
b. directoryEntry-Location—A Location Reference which encapsulates the full location of the Directory Entry for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs
c. typePair—The type (directoryEntry-Type) and subtype (directoryEntry-SubType) used for the search
d. directoryEntry-Phone—Phone number for Directory Entry
e. directoryEntry-HandicapAccess—A code indicating the handicap access level for a business, event, or establishment
f. directoryEntry-BusinessHours—The business hours of the Directory Entry; can be representations for Days of the Week and Holidays
g. directoryEntry-Description—A text description of a Directory Entry; can contain any information desired by the ISP
h. directoryEntry-Cost—Used outside of price schedules to provide text information on rates
i. directoryEntry-ExtendedInformationAvailable—A flag indicating that the ISP has additional special information about the entry; this additional information can be requested by the application

4.3.10 EXTENDED INFORMATION REQUEST—The Extended Information Request is sent when a traveler requires extensive detail on a directory entry. Note that a Detail Information Reply indicates whether extended information is available for a directory entry. An Extended Information Request message consists of the following information:

a. directoryEntry-Identity—A code that uniquely identifies the Directory Entry
b. directoryEntry-ExtendedInformationType—Used by the application to inform an ISP of the types of extended information it can handle; also used by the ISP to attach to an extended information message to define the types of data returned

4.3.11 EXTENDED INFORMATION REPLY—The Extended Information Reply message contains extended detail information on the directory entry specified in the request to which it is replying. The extended detail information is returned as a Binary Large Object (BLOB). The reply may contain multiple BLOBs if there are multiple types of extended information for that directory entry. An Extended Information Request message consists of the following information:

a. DirectoryExtendedInformationReply—The set of DirectoryExtendedInformationEntry(s) that defines the extended detail contained in the reply
b. directoryEntry-Identity—A code that uniquely identifies the Directory Entry
c. directoryEntry-ExtendedInformationType—Used by the application to inform an ISP of the Directories of extended information it can handle; also used by the ISP to attach to an extended information message to define the directory returned
d. directoryEntry-ExtendedInformation—A BLOB used to send extended information to the application from the ISP; may contain pictures, additional text, rich text, sound, etc.

4.3.12 APPOINTMENTS—Appointment requests are used by the traveler to make appointments with a directory entry that was previously obtained using one of the search requests.

Only one possible Appointment message sequence exists—a traveler sends an Appointment Request message and the ISP responds with an Appointment Reply.

4.3.13 APPOINTMENT REQUEST—The Appointment Request message is sent when a traveler wants to make an appointment with an entity identified by a directory entry. An Appointment Request message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. directoryEntry-Identity—A code that uniquely identifies the Directory Entry
c. appointmentTime—The date and time at which the appointment is being requested

4.3.14 APPOINTMENT REPLY—The Appointment Reply message contains the time of the appointment and a confirmation number. An Appointment Reply message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. directoryEntry-Identity—A code that uniquely identifies the Directory Entry
c. AppointmentTime—The date and time at which the appointment is being requested
d. directoryAppointment-Confirmation—A code used to confirm an appointment

4.3.15 SUMMARY INFORMATION—The Summary Information message is used by the ISP to send unacknowledged informational messages, regarding specific directory entries, to travelers. It is a one-way dialog; there is no reply from the traveler. This message identifies the ISP and the summary information:

a. directoryEntry-Identity—A code which uniquely identifies the Directory Entry
b. directoryEntry-Location—A Location Reference which encapsulates the full location of the Directory Entry for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs
c. requestPair—The type (directoryEntry-Type) and subtype (directoryEntry-SubType) used for the search
d. directoryEntry-Name—The Name of the Directory Entry or its common description for purposes of describing the Directory Entry to the Traveler
e. directoryEntry-Phone—Phone number for the Directory Entry
f. directoryEntry-ShortDescription—A short text description of the Directory Entry (optional)
g. directoryEntry-CapabilityCode—A code used to define the capabilities of the Entry or of the ISP on

behalf of the Entry; capabilities include the ability to take electronic reservations and payments

4.4 Mayday Messages—Please see SAE J2313.

4.5 Parking Messages—ATIS parking message sequences fall into two basic types: lot information and space information; price information is returned as part of a space information sequence. Refer to the illustrations shown in Parking Figures 4 through Parking Figure 7.

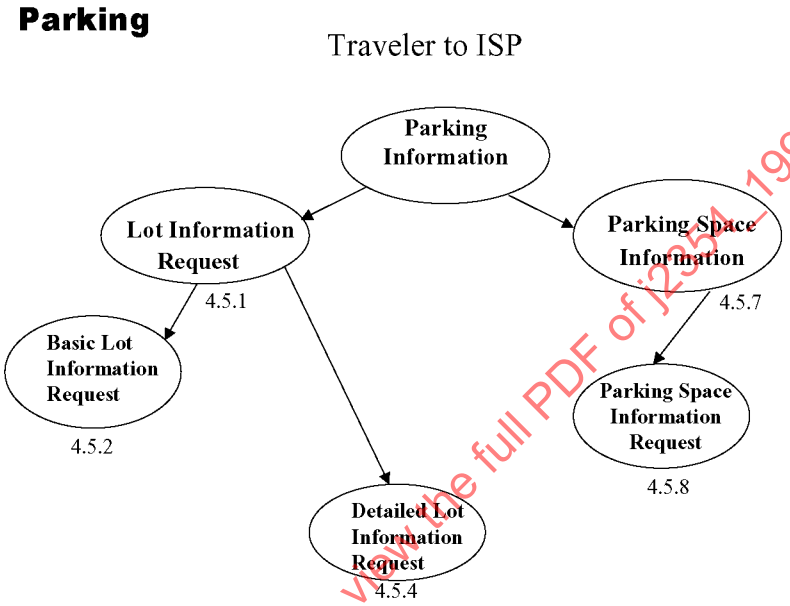

FIGURE 4—PARKING

Parking

ISP to Traveler

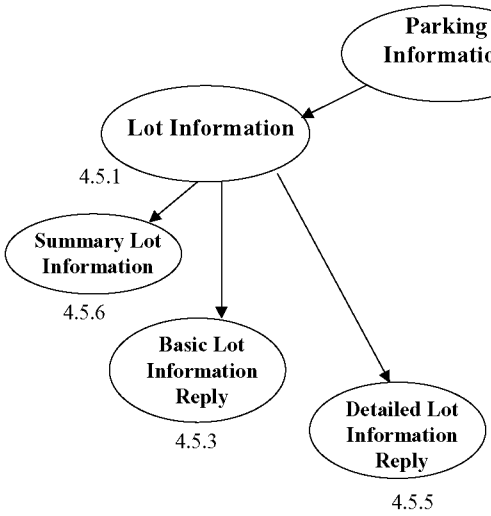

FIGURE 5—PARKING

Parking

ISP to Parking Management

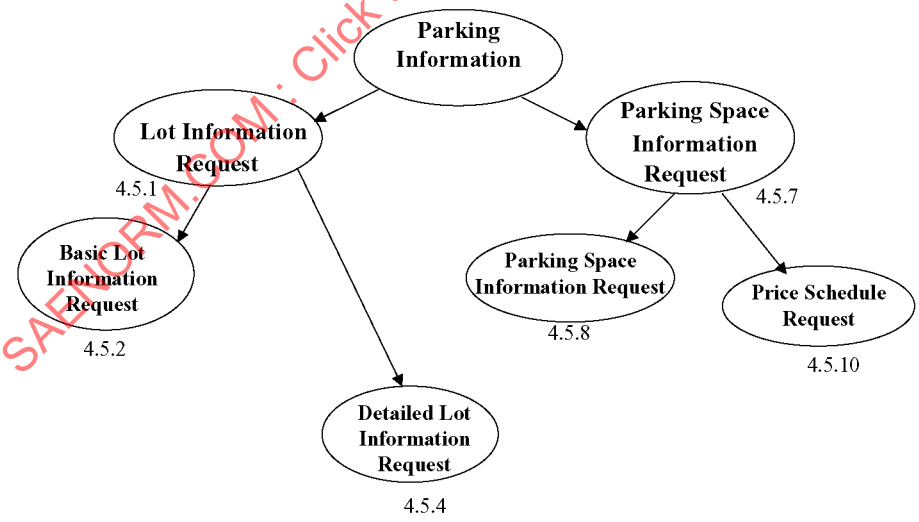

FIGURE 6—PARKING

Parking

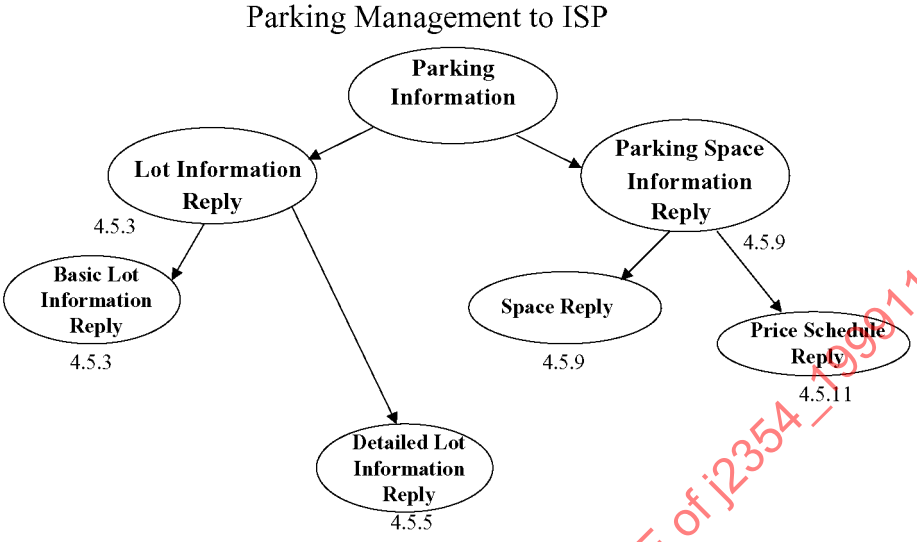

FIGURE 7—PARKING

4.5.1 LOT INFORMATION—Lot Information message requests can be used either by the traveler to obtain lot information from the ISP or by the ISP to obtain lot information from a parking management back office. A request can be made for either Basic Lot Information or more extensive Detailed Lot Information (this request is not supported for all lot entries).

Message Sequences A to D shown in Table 6 summarize the four possible lot information request-reply message sequences. The ISP can also send the traveler an unsolicited and unacknowledged Summary Lot information message (which is shown as Message Sequence E in the table).

TABLE 6—LOT INFORMATION REQUEST-REPLY MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B	C	D	E
Lot Information Request Messages (From Traveler To ISP)					
Basic Lot Information Request	1				
Detailed Lot Information Request		1			
Lot Information Request Messages (From ISP To Parking Management)					
Basic Lot Information Request			1		
Detailed Lot Information Request				1	
Lot Information Reply Messages (From ISP To Traveler)					
Basic Lot Information Reply	2				
Detailed Lot Information Reply		2			
Summary Lot Information Message					1
Lot Information Reply Messages (From ISP To Parking Management)					
Basic Lot Information Reply			2		
Detailed Lot Information Reply				2	

4.5.2 BASIC LOT INFORMATION REQUEST—The Basic Lot Information Request message is sent when a traveler requires basic information about a specific parking lot. A Basic Lot Information Request message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. lot-Location—A Location Reference which encapsulates the full location of the Lot for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs
c. lotRequest-SearchRadius—Search radius for a Lot information request—included with a current or destination LRMS
d. lotRequest-LocationType—Code identifying type of parking—On Road, Public Lot, Private Lot, Enclosed Public Lot, Enclosed Private Lot

4.5.3 BASIC LOT INFORMATION REPLY—The Basic Lot Information Reply message consists of a set of parking lot entries that match the search criteria of the Basic Lot Information Request that originated the reply. Each parking lot entry in the reply set is itself a set that provides basic information about a particular parking lot. A Basic Lot Information Reply message consists of the following information:

a. ParkingLotSearchReply—The set of ParkingLotEntries (s) that defines the reply
b. pI-ParkingFacID—A code which uniquely identifies the Parking Lot
c. lot-Type—Code identifying type of parking—On Road, Public Lot, Private Lot, Enclosed Public Lot, Enclosed Private Lot
d. lot-Location—A Location Reference which encapsulates the full location of the Lot for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs
e. pI-ParkingAvailability—The number of available public spaces in a Lot (optional)

4.5.4 DETAILED LOT INFORMATION REQUEST—The Detail Lot Information Request message is sent when a traveler requires additional detail on a lot for which a Basic Lot Information Reply has been received. A Detail Lot Information Request message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. pI-ParkingFacID—A code which uniquely identifies the Parking Lot
c. lot-InformationType—The specific lot information being requested

4.5.5 DETAILED LOT INFORMATION REPLY—The Detailed Lot Information Reply message consists of a set of parking lot entries that match the search criteria of the Detailed Lot Information Request that originated the reply. Each parking lot entry in the reply set is itself a set that provide basic information about a particular parking lot. A Basic Lot Information Reply message consists of the following information:

a. ParkingLotSpecificDetailReply—The set of lotInformation (s) that defines the reply
b. pI-ParkingFacID—A code which uniquely identifies the Parking Lot
c. pI-parkingSpacesTotal—The total vehicle capacity of the Lot—counts only public spaces
d. pI-ParkingAvailability—The number of available public spaces in a Lot (optional)
e. lot-Name—The name of the Lot or its common description for purposes of describing the lot to the Traveler
f. lot-BusinessHours—The business hours of the Lot—can include separate representations for Days of the Week and Holidays
g. pI-ParkingRates—Used outside of price schedules to provide text information on rates
h. pI-ParkingFillTime—The time at which a parking facility usually (on weekdays) reaches a point where spaces available equals 0
i. pI-ParkingVehicleClass—Vehicle size or height restrictions for a given Parking Lot

4.5.6 SUMMARY LOT INFORMATION—The Summary Lot Information message is used by the ISP to send unacknowledged informational messages—regarding specific parking lots—to travelers. It is a one-way dialog; there is no reply from the traveler. This message contains information identifying the ISP and the summary information itself:

a. lot-Location—A Location Reference which encapsulates the full location of the Lot for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs
b. lot-Type—Code identifying type of parking—On Road, Public Lot, Private Lot, Enclosed Public Lot, Enclosed Private Lot
c. lot-Status—The current status of the Lot in terms of occupancy—used as shorthand for broadcast and basic information

4.5.7 PARKING SPACE INFORMATION—Parking Space Information message requests are used either by the traveler to obtain space information from the ISP or by the ISP to obtain space information, or price schedules, from a parking management back office.

Message Sequences A-C shown in Table 7 summarize the three space information request-reply message sequences.

TABLE 7—SPACE INFORMATION REQUEST-REPLY MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B	C
Space Request Messages (From Traveler To ISP)			
Space Request	1		
Space Request Messages (From ISP To Parking Management)			
Space Request		1	
Price Schedule Request			1
Space Reply Messages (From ISP To Traveler)			
Space Request Reply	2		
Space Reply Messages (From ISP To Parking Management)			
Space Request Reply		2	
Price Schedule Reply			2

4.5.8 PARKING SPACE INFORMATION REQUEST—The Parking Space Information Request message is used by the ISP to request parking space availability information from the Parking Management. The message contains the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. pI-ParkingFacID—A code which uniquely identifies the Parking Lot
c. lotStay-EstimatedDuration—Estimated duration of a Parking Lot stay
d. lotStay-RequestTime—Start time of a vehicle requesting a parking space within a Lot (optional)

4.5.9 SPACE INFORMATION REPLY—The Parking Space Information Reply message is used by the Parking Management to send parking space availability (and its associated cost) information to the ISP. The message contains the following information:

a. pI-parkingFacID—A code which uniquely identifies the Parking Lot
b. pI-ParkingRates—Cost for a Parking Lot stay

4.5.10 PRICE SCHEDULE REQUEST—The Parking Lot Price Schedule Request is used by the ISP to request price information from the Parking Management. The message contains the following information:

a. lot: the identity (DatabaseIdentity) or location (LocationReference) to be used in the search
b. start—(DateTimePair) (optional)
c. end—(DateTimePair) (optional)

4.5.11 PRICE SCHEDULE REPLY—The Parking Lot Price Schedule Reply is used by the Parking Management to send parking space cost information to the ISP. The message contains the following information:

a. pI-ParkingFacID—A code which uniquely identifies the Parking Lot
b. lot-Location—A Location Reference which encapsulates the full location of the Lot for the Traveler; the Location Reference can be requested in a variety of formats depending on the application needs
c. price-DayType—Used as part of Sending a Price Schdeule—The Days of Week to which the following information pertains—also includes designator “Hol” for Holdiay
d. start—(TimePair) (optional)
e. end—(TimePair) (optional)
f. price-FirstPayment—Used as part of Sending a Price Schedule—The first payment where it differs from succeeding payments or the fixed price of a stay
g. price-Maximum—Used as part of Sending a Price Schedule—The Maximum amount payable regardless of stay
h. price-Time Interval—Used as part of Sending a Price Schedule—The Time Interval for the Rate (such as 30 minutes) with all other qualifiers specified; Fixed Rates are specified with the First Payment field
i. price-TimeValue—Used as part of Sending a Price Schedule—The Cost of a space for the given time interval with all other qualifiers specified

4.6 Traveler Information Messages—Traveler Information messages are two-way communications between the traveler and the ISP. Refer to the illustrations in Traveler Information Figures 8 and 9.

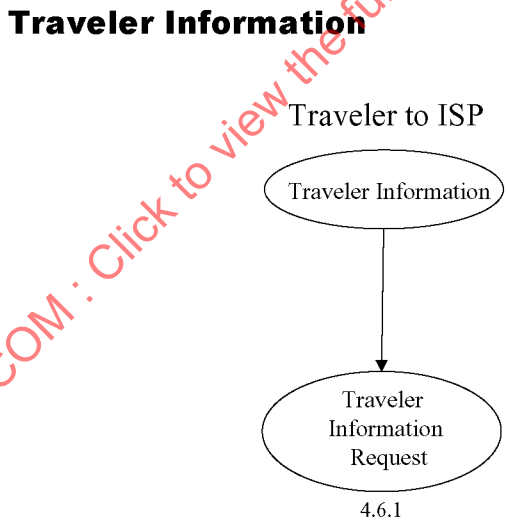

FIGURE 8—TRAVELER INFORMATION

Traveler Information

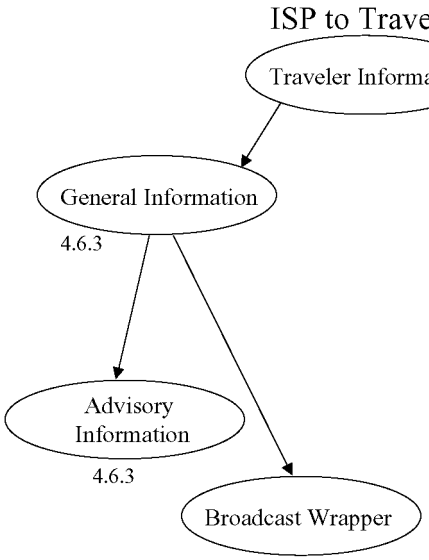

FIGURE 9—TRAVELER INFORMATION

Traveler Information has a single Traveler Information Request message (Table 8, Message Sequence A) that can be parameterized by the traveler to request a wide range of traveler information. A single Traveler Information Reply message is also parameterized to allow the ISP to respond with any of the possible types of information requested by the traveler.

The ISP can broadcast one-way unsolicited messages to multiple travelers using the Advisory Information message.

The ISP can also broadcast unsolicited messages to multiple travelers using the Broadcast Wrapper Message (Table 8, Message Sequence B). A Broadcast Wrapper message consists of information that the ISP will send out to passing travelers via an unacknowledged broadcast.

TABLE 8—TRAVELER INFORMATION REQUEST-REPLY MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B
Traveler Information Request Messages (From Traveler To ISP)		
Traveler Information Request	1	
Traveler Information Reply Messages (From ISP To Traveler)		
Traveler Information Reply	2	
Advisory Information		1
Broadcast Wrapper		1

4.6.1 TRAVELER INFORMATION REQUEST—A Traveler Information Request is sent from the travel to the ISP. It allows the traveler to request a specific category of travel information from within a broad category of possible information types. The message contains the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. informationLocation—Location reference for information request
c. informationRequest-Type—The category of information being requested by the Traveler
d. informationRequest-SubType—The specify item of information being requested by the Traveler
e. start—DateTimePair) (optional)
f. end—(DateTimePair) (optional)

4.6.2 TRAVELER INFORMATION REPLY—The Traveler Information Reply is sent by the ISP to the traveler, and is based upon the Type of information that the traveler requested. The response may be in the form of a One-Time Area Request, an Ongoing Area Request, or an Information Request. The message contains some or all of the following information:

a. advisoryArea—Location reference for information response
b. weatherLocation—The location reference for the weather
c. weather-ForecastOrActual—The weather forecast
d. atis-DayOfWeek: describes the Day of the Week, including a Holiday option
e. weather-Temperature (optional)—The temperature
f. weather-HighTemperature—The high temperature recorded (optional) :
g. weather-LowTemperature—The low temperature recorded (optional)
h. weather-SkyConditions—The condition of the sky (optional) :
i. weather-Probability—The probability assessment for the weather forecast (optional)
j. weather-SpecialConditions: (optional)—Special conditions that elaborate on or extend basic weather information
k. weather-Visibility—The current or forecasted visibility (optional)
l. weatherWindSpeed—The speed of the wind (optional)
m. weatherWindDirection—The wind direction (optional)
n. weather-Humidity—The humidity level (optional)
o. weather-Pressure—Barometric pressure in millibars (optional)
p. weather-SunriseTime—The time for sunrise on a given date (optional)
q. weather-SunsetTime—The time for sunset on a given date (optional)
r. pollution- SmogAlert—Reflects the smog alert level, where 0=No Alert
s. pollution-AirQualityIndex—Pollutant level percentile compared with the National Ambient Air Quality Standard for criteria pollutants (optional)
t. pollution-CarbonMonoxide—The carbon monoxide levels in a region (optional)
u. pollution-HydroCarbon—The hydrocarbon levels in a region (optional)
v. pollution-SulfurDioxide—The level of sulfur dioxide in a region (optional)
w. pollution-NitrousOxide—The nitrous oxide levels in a region (optional)
x. pollution-Particulate—The level of particulate in a region (optional)
y. pollution-Ozone—The ozone levels in a region (optional)
z. linkLocation—Location reference for a specific link
aa. link-Delay—Calculated delay for vehicles driving along a particular Link; this is additional time it will take above that recorded during freeflow conditions to travel from one end of the link to the other (optional)
ab. link-Capacity—The Link maximum capacity in vehicles per hour (optional)
ac. link-Density—Vehicle concentration per kilometer (in vehicles per kilometer) of the Link (optional)
ad. link-LanesNumberOpen—The lowest number at any point of lanes currently open in the link (optional)
ae. link-Occupancy—Percent occupancy measured for the Link (optional)
af. link-Speed—The average Link vehicular speed in kilometers per hour (optional)
ag. link-Status—The Link Status (optional)
ah. link-SurfaceConditions—The surface condition of the Link (dry, wet, ice, snow, rocks, etc.) (optional)
ai. incidentLocation—The Location Reference for a specific incident
aj. event-IncidentSeverity—A code which describes the severity of an incident
ak. event-IncidentStatus—A code which indicates a status of the incident

al. incident-TimelineConfirmedAndResponding (DateTimePair) (optional)
am. incident-TimelineClearedAndRecovering (DateTimePair) (optional)
an. event-IncidentVehiclesInvolvedCount—The total number of vehicles involved in an incident (optional)
ao. event-IncidentVehiclesInvolved—A code list which indicates the types of vehicles involved in a verified incident (optional)
ap. event-DescriptionTypePlannedRoadwayClosure—A code representing the type of planned roadway closure
aq. event-DescriptionTypeSpecialEvent—A code representing the type of special event
ar. event-Description—A textual description of a roadway event (incident, planned roadway, closure, or special event) (optional)
as. event-LanesBlockedOrCLosedCount—The total number of lanes blocked by an event (incident, roadway construction, special event) (optional)
at. event-LanesDirectionOfTravel—A code which indicates the direction where the shoulder(s) or lane(s) are blocked from travel by a roadway event (optional)
au. event-TimeLineStart—The actual start date and time of a roadway event (optional)
av. event-TimeLineEstimatedDuration—The estimated time duration of a roadway event (optional)
aw. event-TimeLineSchduledEnd—The scheduled end date and time of a roadway event (optional)
ax. start—(DateTimePair) (optional)
ay. stop—(DateTimePair) (optional)
az. event-TimeLineScheduleDaysOfWeek—Days of the week during which time a roadway event is in effect (optional)

4.6.3 ADVISORY INFORMATION—An Advisory Information message contains some or all of the following information:

a. advisoryArea—Location reference for information response
b. weatherLocation—The location reference for the weather
c. weather-ForecastOrActual—The weather forecast
d. atis-DayOfWeek—Describes the Day of the Week, including a Holiday option
e. weather-Temperature (optional)—The temperature
f. weather-HighTemperature—The high temperature recorded (optional)
g. weather-LowTemperature—The low temperature recorded (optional)
h. weather-SkyConditions—The condition of the sky (optional)
i. weather-Probability—The probability assessment for the weather forecast (optional)
j. weather-SpecialConditions—(optional)—Special conditions that elaborate or extend basic weather conditions
k. weather-Visibility—The current or forecasted visibility (optional)
l. weatherWindSpeed—The speed of the wind (optional)
m. weatherWindDirection—The wind direction (optional)
n. weather-Humidity—The humidity level (optional)
o. weather-Pressure—Barometric pressure in millibars (optional)
p. weather-SunriseTime—The time for sunrise on a given date (optional)
q. weather-SunsetTime—The time for sunset on a given date (optional)
r. pollution- SmogAlert—Reflects the smog alert level, where 0=No Alert
s. pollution-AirQualityIndex—Pollutant level percentile compared with the National Ambient Air Quality Standard for criteria pollutants (optional)
t. pollution-CarbonMonoxide—The carbon monoxide levels in a region (optional)
u. pollution-HydroCarbon—The hydrocarbon levels in a region (optional)
v. pollution-SulfurDioxide—The level of sulfur dioxide in a region (optional)
w. pollution-NitrousOxide—The nitrous oxide levels in a region (optional)
x. pollution-Particulate—The level of particulate in a region (optional)
y. pollution-Ozone—The ozone levels in a region (optional)
z. linkLocation—Location reference for a specific link
aa. link-Delay—Calculated delay for vehicles driving along a particular Link; this is additional time it will take above that recorded during freeflow conditions to travel from one end of the link to the other

(optional)

ab. link-Capacity—The Link maximum capacity in vehicles per hour (optional)
ac. link-Density—Vehicle concentration per kilometer (in vehicles per kilometer) of the Link (optional)
ad. link-LanesNumberOpen—The lowest number at any point of lanes currently open in the link (optional)
ae. link-Occupancy—Percent occupancy measured for the Link (optional)
af. link-Speed—The average Link vehicular speed in kilometers per hour (optional)
ag. link-Status—The Link Status (optional)
ah. link-SurfaceConditions—The surface condition of the Link (dry, wet, ice, snow, rocks, etc.) (optional)
ai. incidentLocation—The Location Reference for a specific incident
aj. event-IncidentSeverity—A code which describes the severity of an incident
ak. event-IncidentStatus—A code which indicates a status of the incident
al. incident-TimelineConfirmedAndResponding (DateTimePair) (optional)
am. incident-TimelineClearedAndRecovering (DateTimePair) (optional)
an. event-IncidentVehiclesInvolvedCount—The total number of vehicles involved in an incident (optional)
ao. event-IncidentVehiclesInvolved—A code list which indicates the types of vehicles involved in a verified incident (optional)
ap. event-DescriptionTypePlannedRoadwayClosure—A code representing the type of planned roadway closure
aq. event-DescriptionTypeSpecialEvent—A code representing the type of special event
ar. event-Description—A textual description of a roadway event (incident, planned roadway, closure, or special event) (optional)
as. event-LanesBlockedOrCLosed—The total number of lanes blocked by an event (incident, roadway construction, special event) (optional)
at. event-LanesDirectionOfTravel—A code which indicates the direction where the shoulder(s) or lane(s) are blocked from travel by a roadway event (optional)
au. event-TimeLineStart—The actual start date and time of a roadway event (optional)
av. event-TimeLineEstimatedDuration—The estimated time duration of a roadway event (optional)
aw. event-TimeLineSchduledEnd—The scheduled end date and time of a roadway event (optional)
ax. start—(DateTimePair) (optional)
ay. stop—(DateTimePair) (optional)
az. event-TimeLinesScheduleatisDaysOfWeek—Days of the week during which time a roadway event is in effect (optional)

4.7 Trip Guidance Messages—Trip Guidance messages are two-way messages between the traveler and the ISP. The traveler requests various route information, modifying as needed, and the ISP returns the requested data. Refer to the illustrations in Trip Guidance Figures 10 and 11.

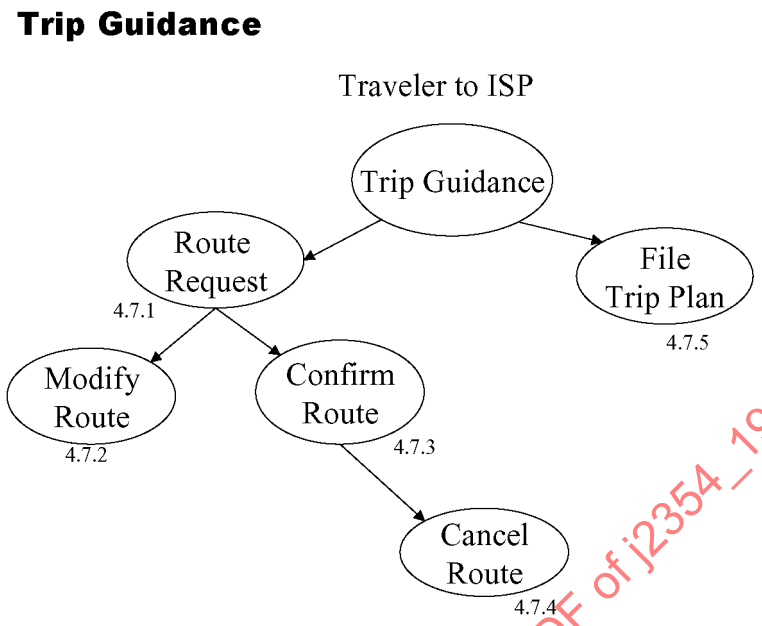

FIGURE 10—TRIP GUIDANCE

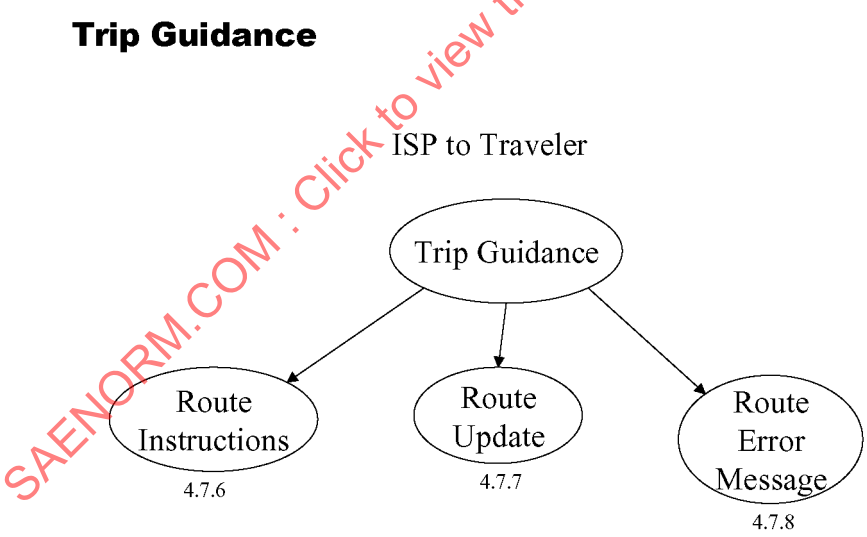

FIGURE 11—TRIP GUIDANCE

4.7.1 ROUTE REQUEST—Route Request messages are used by the traveler to request a route, modify a route, confirm a route (receive route updates), or cancel a previous route request.

Route Reply messages are used by the ISP to return to the traveler either the information requested or an error message indicating why the request failed.

Message Sequences A-H shown in Table 9 summarize the eight possible Route request-reply message sequences.

TABLE 9—ROUTE REQUEST-REPLY MESSAGE SEQUENCES

Message Sequence: (Numbers Indicate Order)	A	B	C	D	E	F	G	H
Route Request Messages (From Traveler To ISP)								
Route Request	1	1						
Modify Route Request			1	1				
Confirm Route Request					1	1		
Cancel Route Request							1	1
Route Reply Messages (From ISP To Traveler)								
Route Instructions	2							
Route Update			2		2			
Route Error Message		2		2		2		

The message contains to following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-StartDate—Date to begin trip
c. trip-StartTime—Time to begin trip
d. trip-OriginLocation—Location reference for origin of trip
e. trip-DestinationLocation—Location reference for destination
f. wayPoint—Location reference for intermediate destination(s)
g. trip-WayPointPrioritizedListFlag—Flag to indicate way points are in priority order
h. trip-PreferenceType—Traveler preference affecting a route choice
i. trip-PreferenceSubType—Values for Traveler Preferences (as required)
j. trip-ConstraintType—Traveler constraints affecting route selection
k. trip-ConstraintSubType—Values to describe constraints (as required)
l. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
m. trip-Guidancelevel—A code to describe the amount of information to be returned to the user, such as: maneuvers only, connected links, etc.
n. trip-MaximumManeuvers—The maximum number of maneuvers to return to the user
o. trip-MaximumLinks—The maximum number of links to return to the user
p. trip-MaximumConnectedLinks—The maximum number of connected links to return to the user

4.7.2 MODIFY ROUTE REQUEST—The Modify Route Request message contains to following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-StartDate—Date to begin trip
c. trip-StartTime—Time to begin trip
d. trip-OriginLocation—Llocation reference for origin of trip
e. trip-DestinationLocation—Location reference for destination
f. wayPoint—Location reference for intermediate destination(s)
g. trip-WayPointPrioritizedListFlag—Flag to indicate way points are in priority order
h. trip-PreferenceType—Traveler preference affecting a route choice
i. trip-PreferenceSubType—Values for Traveler Preferences (as required)
j. trip-ConstraintType—Traveler constraints affecting route selection
k. trip-ConstraintSubType—Values to describe constraints (as required)
l. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
m. trip-GuidanceLevel—A code to describe the amount of information to be returned to the user, such as: maneuvers only, connected links, etc.
n. trip-MaximumManeuvers—The maximum number of maneuvers to return to the user (optional)
o. trip-RouteIdentity—A unique identifier for a route assigned by the ISP

4.7.3 CONFIRM ROUTE—The Confirm Route message contains the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
c. trip-ConfirmRouteFlag—Flag to confirm route

4.7.4 CANCEL ROUTE—The Cancel Route message contains the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
c. trip-RouteIdentity—A unique identifier for a route assigned by the ISP

4.7.5 FILE TRIP PLAN—The File Trip Plan message contains the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-StartDate—Date to begin trip
c. trip-StartTime—Time to begin trip
d. trip-OriginLocation—Location reference for origin of trip
e. trip-DestinationLocation—Location reference for destination
f. wayPoint—Location reference for intermediate destination(s)
g. trip-WayPointPrioritizedListFlag—A flag to indicate way points are in priority order
h. trip-PreferenceType—Traveler preference affecting a route choice
i. trip-PreferenceSubType—Values for Traveler Preferences (as required)
j. trip-ConstraintType—Traveler constraints affecting route selection
k. trip-ConstraintSubType—Values to describe constraints (as required)
l. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system

4.7.6 ROUTE INSTRUCTIONS—A Route Instructions message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-StartDate—Date to begin trip
c. trip-StartTime—Time to begin trip
d. trip-OriginLocation—Location reference for origin of trip
e. trip-DestinationLocation—Location reference for destination
f. wayPoint—Location reference for intermediate destination(s)
g. trip-EstimatedTravelTime—Estimate of travel time returned to the traveler based upon route
h. wayPointTimes—Sequence of travel times
i. trip-TotalDriveMiles—Total mileage of trip returned to the traveler based upon route
j. trip-ManeuverCode—Code describing maneuver required to maintain route
k. trip-ManeuverAngle—Angle of turn calculated by ISP or Route Guidance Provider
l. trip-ManeuverLocation—Location reference where maneuver takes place
m. trip-CompoundManeuverFlag—Flag to indicate next two maneuvers comprise a compound maneuver
n. trip-MilesToNextManeuver—Number of miles to next maneuver
o. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
p. trip-NumberOfTransitStops—The number of transit stops returned for the route (optional)
q. trip-EstimatedRouteCost—Estimated cost for route (tolls + transit) (optional)
r. pI-DepartTimeScheduled—Scheduled departure time for transit
s. trip-LinkLocation—The location reference for a specified link (optional)
t. trip-RouteIdentity—A unique identifier for a route assigned by the ISP

4.7.7 ROUTE UPDATE—A Route Update message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-EstimatedTravelTime—Estimate of travel time returned to the traveler based upon route
c. wayPointTimes—Sequence of travel times
d. trip-TotalDriveMiles—Total mileage of trip returned to the traveler based upon route
e. trip-ManeuverCode—Code describing maneuver required to maintain route
f. trip-ManeuverAngle—Angle of turn calculated by ISP or Route Guidance Provider
g. trip-ManeuverLocation—Location reference where maneuver takes place
h. trip-CompoundManeuverFlag—Flag to indicate next two maneuvers comprise a compound maneuver
i. trip-MilesToNextManeuver—Number of miles to next maneuver
j. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
k. trip-EstimatedRouteCost—Estimated cost for route (tolls + transit) (optional)
l. trip-LinkLocation—The location reference for a specified link (optional)

4.7.8 ROUTE ERROR MESSAGE—A Route Error message consists of the following information:

a. traveler-Identity—Identifies a traveler uniquely—can be used to drive a session and to reference a traveler setting
b. trip-RequestErrorType—Type of trip guidance request error
c. trip-RequestIdentity—Unique identifier allowing traveler to have multiple requests in the system
d. locationAlternatives—Alternative location references (optional)

5. ASN.1 Messages—This section is organized according to the six message categories delineated in Section 4. In addition, messages that are not used exclusively in one category are included in the “Global Messages” section.

5.1 Global Messages

```
LocationReference ::= CHOICE
{
        geometry                        GeometryProfile,
        geocoordinate                   GeographicCoordinate,
        grid                            GridProfile,
        linearReferencing               LinearReferenceProfile,
        crossStreets                    CrossStreetsProfile,
        address                         AddressProfile,
        mdi                             ModelProfile
}

-- Included as an Example Only are CrossStreetsProfile and
-- necessary supporting structures

Altitude  ::=  CHOICE
{
        levelCode                       INTEGER   (-7..7),
        altitude                        INTEGER   (-8191..57344)
}

CrossStreetsProfile ::= SEQUENCE
{
        onStreetName                    IA5String (SIZE(1..255)),
        fromStreetName                  IA5String (SIZE(1..255)),
        toStreetName                    IA5String (SIZE(1..255)),
        side                            ENUMERATED
                                        {
                                        right(0),
                                        left(1)
                                        },
        horizontalDatum                 INTEGER {
                                        wgs-84 (0), wgs-84egm-96 (1),
                                        nad83 (2), nad27 (3),
                                        reservedStandard (7)
                                        },
        verticalDatum                           INTEGER   {
                                                wgs-84(0) , navd-88(1), verticalLevelCode(2), reservedStandard(7)
                                        },
        startLongitude                          INTEGER   (-180000000..180000000),
        startLatitude                           INTEGER   (-90000000..90000000),
        endLongitude                            INTEGER   (-180000000..180000000),
        endLatitude                                     INTEGER   (-90000000..90000000),
        startAltitude                           Altitude,
        endAltitude                                     Altitude,
        distanceOffset1                                 INTEGER   (-8388607..8388607),
        distanceOffset2                                 INTEGER   (-8388607..8388607)
}

ATIS-DayOfWeek ::= BIT STRING
{
        sunday(0),
        monday(1),
        tuesday(2),
        wednesday(3),
        thursday(4),
        friday(5),
        saturday(6),
        includeHolidays(7)
}
```

```
SearchRadius ::= CHOICE
{
        directorySearch-NumericRadius                   INTEGER (0..65535),
        -- unit to be best metric verision of 0.05 mile
        geometry                                        GeometryProfile
}

DatabaseNumber ::= INTEGER (0..255)
Database  ::= SEQUENCE
{
        dbNumber                        DatabaseNumber,
        databaseName                    IA5String (SIZE(1..15)),
        databaseVersion                 OCTET STRING  (SIZE(1..7))
}

DatabaseIdentity ::= SEQUENCE
{
        identity                        OCTET STRING,
        databaseCode                    CHOICE {
                                        dbNumber  DatabaseNumber,
                                        dbName    Database

        }
}

DatabaseSetting ::= SEQUENCE OF Database
IdentityOrLocation ::= CHOICE
{
        location                        LocationReference,
        dbIdentity                      DatabaseIdentity
}

CoreRouteRequest ::= SEQUENCE
{
        trip-OriginLocation                     LocationReference,
        trip-DestinationLocation                LocationReference,
        wayPoint                                SEQUENCE OF LocationReference
}

RouteOrArea ::= CHOICE
{
        route                           CoreRouteRequest,
        area                            IdentityOrLocation
}

DateTimePair   ::= SET
{
        requestDate                     IA5String (SIZE(8)),
        requestTime                     IA5String (SIZE(4..12))
}

PriceSchedule ::= SEQUENCE
{
        start                           DateTimePair OPTIONAL,
        end                             DateTimePair OPTIONAL,
        entries                         SEQUENCE OF PriceScheduleEntry
}

TimePair ::= SEQUENCE
{
        start                           IA5String (SIZE(6..12)),
        end                             IA5String (SIZE(4..12))
}
```

```
TimeInterval ::= SEQUENCE
{
        price-TimeInterval                  INTEGER (0..65535),
        price-TimeValue                     REAL
}
```

5.2 Setting Messages

```
PreferenceSettings ::= SEQUENCE
{
        traveler-Identity                   IA5String (SIZE(1..12)),
        setting-Identity                    INTEGER   (0..255),
        settingsDetail                      SEQUENCE OF TravelerPreference
}

TripPreferencePair ::= SEQUENCE
{
        preferences                         TripPreferences OPTIONAL,
        constraints                         TripConstraints OPTIONAL
}

TypePreferencePair ::= SEQUENCE
{
        setting-PreferenceType                  INTEGER (0..255),
        setting-PreferenceSubType           INTEGER (0..255) OPTIONAL
}

PreferenceExpression ::= CHOICE {
            directory                       DirectoryTypePair,
            trip                            TripPreferencePair,
            other                           TypePreferencePair
}

TravelerPreference ::= SEQUENCE
{
        setting-Type                        INTEGER,
        expression                          PreferenceExpression,
        atis-SearchOperator                 INTEGER   (0..255) DEFAULT 0,
        atis-CostPreference                 INTEGER   OPTIONAL,
        atis-CostPreferenceAmount           INTEGER   (0..65535) OPTIONAL
}

DeleteSetting ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        setting-Identity                        INTEGER   (0..255)
}

SelectPreferenceSetting ::= SEQUENCE
{
        traveler-Identity                   IA5String (SIZE(1..12)),
        setting-Identity                    INTEGER   (0..255)
}

PhoneInformation ::= SEQUENCE
{
        traveler-Phone                                  NumericString  (SIZE(15)),
        traveler-Extension                              NumericString  (SIZE(1..4)) OPTIONAL,
        traveler-InternationalAccessCode                NumericString  (SIZE(1..3)) OPTIONAL
}
```

```
TravelerContactSetting ::= SEQUENCE
{
        traveler-Identity                IA5String           (SIZE(1..12)),
        setting-Identity                 INTEGER             (0..255),
        traveler-FirstName               IA5String           (SIZE(1..25)) OPTIONAL,
        traveler-LastName                IA5String           (SIZE(1..25)) OPTIONAL,
        contact   SEQUENCE OF TravelerContactInformation
}

ContactInformation      ::= CHOICE
{
         traveler-EMail                  IA5String (SIZE(1..40)),
         phone                           PhoneInformation,
         traveler-Pager                  NumericString (SIZE(1..20)),
        device-Identity                  OCTET STRING
}

TravelerContactInformation    ::= SEQUENCE
{
        traveler-Identity                IA5String           (SIZE(1..12)),
        information                      ContactInformation,
        contactDayOfWeek                 ATIS-DayOfWeek  OPTIONAL,
        contactTimes                     TimePair            OPTIONAL
}

TravelerExtendedSetting ::= SEQUENCE
{
        traveler-Identity                        IA5String  (SIZE(1..12)),
        setting-Identity                         INTEGER  (0..255),
        setting-ExtendedInformation              OCTET STRING
}

TriggerSetting ::= SEQUENCE
{
        traveler-Identity                IA5String (SIZE(1..12)),
        setting-Identity                 INTEGER   (0..255),
        location                         RouteOrArea,
        trigger-Event                    INTEGER   (0..255),
        start                            DateTimePair  OPTIONAL,
        end                              DateTimePair OPTIONAL,
        timeEnforced                     TimePair OPTIONAL,
        triggerDayOfWeek                 ATIS-DayOfWeek OPTIONAL
}

TravelerDeviceSetting ::= SEQUENCE
{
        traveler-Identity                IA5String           (SIZE(1..12)),
        setting-Identity                 INTEGER             (0..255),
        device-Setting                   OCTET STRING,
        device-TransferSpeed             INTEGER             OPTIONAL
}
ErrorOrInformationNotification ::= SEQUENCE
{
        error-NotificationType           INTEGER             (0..255),
        error-NotificationCode           INTEGER             (0..65535),
        error-NotificationText           IA5String           (SIZE(1..255)) OPTIONAL,
        locationAlternatives             SEQUENCE OF LocationReference OPTIONAL
}

MessageInformationWrapper ::= SEQUENCE
{
```

```
        message-Version                 INTEGER  (0..255) OPTIONAL,
        message-Priority                INTEGER OPTIONAL,
        messageTime                     DateTimePair OPTIONAL,
        message-Confidence              INTEGER  (0..100) OPTIONAL,
        message-Quality                 INTEGER  (0..100) OPTIONAL
}
```

5.3 Directory Services Messages

```
DirectoryTypePair ::= SEQUENCE
{
        directoryEntry-Type             INTEGER   (0..999999),
        directoryEntry-SubType          INTEGER   (0..65535)
}

DirectoryKeywordPair ::= SEQUENCE
{
        directorySearch-Keywords        IA5String       (SIZE(1..40)),
        atis-SearchOperator             INTEGER         (0..255)
}

DirectoryCoreRequest ::= SEQUENCE
{
        traveler-Identity                                       IA5String (SIZE(1..12)),
        entry-Location                                          RouteOrArea,
        directorySearch-Radius                                  SearchRadius,
        requestPair                                             DirectoryTypePair,
        directorySearch-LimitEntriesReturned                    INTEGER   (0..65535) OPTIONAL,
        directoryEntry-LocationType                     INTEGER   (0..255) OPTIONAL
}

DirectoryRequest ::= SEQUENCE
{
        coreRequest                     DirectoryCoreRequest,
        start                           DateTimePair OPTIONAL,
        end                             DateTimePair OPTIONAL
}

DirectoryAdvancedRequest ::= SEQUENCE
{
        traveler-Identity                               IA5String (SIZE(1..12)),
        traveler-Location                               RouteOrArea,
        typePair SEQUENCE OF                            DirectoryTypePair,
        directorySearch-Radius                          SearchRadius OPTIONAL,
        directorySearch-LimitEntriesReturned            INTEGER   (0..65535) OPTIONAL,
        start                                           DateTimePair OPTIONAL,
        end                                             DateTimePair OPTIONAL,
        directoryEntry-LocationType                     INTEGER   (0..255) OPTIONAL
}

DirectoryNameRequest ::= SEQUENCE
{
        basicRequest                    DirectoryCoreRequest,
        directoryEntry-Name             IA5String (SIZE(1..30)),
        start                           DateTimePair OPTIONAL,
        end                             DateTimePair OPTIONAL,
        directoryEntry-LocationType     INTEGER   (0..255) OPTIONAL
}

DirectoryExtendedRequest ::= SEQUENCE
{
        traveler-Identity               IA5String (SIZE(1..12)),
        traveler-Location               RouteOrArea,
        directorySearch-Radius          SearchRadius OPTIONAL,
```

```
        directorySearch-LimitEntriesReturned        INTEGER   (0..65535) OPTIONAL,
        typePair SEQUENCE OF                        DirectoryTypePair OPTIONAL,
        keywordPair SEQUENCE OF                     DirectoryKeywordPair OPTIONAL,
        directorySearch-ConstrainByProfile          BOOLEAN,
        directorySearch-OrderEntriesBy              INTEGER   (0..255),
        start                                       DateTimePair OPTIONAL,
        end                                         DateTimePair OPTIONAL,
        directoryEntry-LocationType                 INTEGER   (0..255) OPTIONAL
}

DirectoryEntry ::= SEQUENCE
{
        directoryEntry-Identity                     IA5String (SIZE(1..9)),
        directoryEntry-Location                     LocationReference,
        requestPair                                 DirectoryTypePair,
        directoryEntry-Name                         IA5String (SIZE(1..30)),
        directoryEntry-Phone                        PhoneInformation OPTIONAL,
        directoryEntry-ShortDescription             IA5String (SIZE(1..40)) OPTIONAL,
        directoryEntry-CapabilityCode               INTEGER   (0..255) OPTIONAL
}

DirectoryInformation ::= SET OF DirectoryEntry

DirectoryDetailRequest ::= SEQUENCE
{
        directoryEntry-Identity                     IA5String (SIZE(1..9)),
        directoryEntry-LocationType                 INTEGER   (0..255)
}

DirectoryDetailReply ::= SEQUENCE
{
        directoryEntry-Identity                     IA5String (SIZE(1..9)),
        directoryEntry-Location                     LocationReference,
        typePair                                    DirectoryTypePair,
        directoryEntry-Name                         IA5String (SIZE(1..30)),
        directoryEntry-Phone                        PhoneInformation,
        directoryEntry-HandicapAccess               INTEGER   (0..255) OPTIONAL,
        directoryEntry-BusinessHours                IA5String (SIZE(1..12)) OPTIONAL,
        directoryEntry-Description                  IA5String (SIZE(1..200)) OPTIONAL,
        directoryEntry-Cost                         IA5String (SIZE(1..12)) OPTIONAL,
        directoryEntry-ExtendedInformationAvailable                 BOOLEAN OPTIONAL
}

DirectoryExtendedInformationRequest ::= SET
{
        directoryEntry-Identity                     IA5String (SIZE(1..9)),
        directoryEntry-ExtendedInformationType      INTEGER   (0..255)
}

DirectoryExtendedInformationEntry ::= SET
{
        directoryEntry-Identity                     IA5String (SIZE(1..9)),
        directoryEntry-ExtendedInformationType      INTEGER   (0..255),
        directoryEntry-ExtendedInformation          OCTET STRING
}

DirectoryExtendedInformationReply ::= SET OF DirectoryExtendedInformationEntry

DirectoryAppointmentRequest ::= SEQUENCE
{
        traveler-Identity                           IA5String (SIZE(1..12)),
        directoryEntry-Identity                     IA5String (SIZE(1..9)),
        appointmentTime                             DateTimePair
```

```
}

DirectoryAppointmentReply ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        directoryEntry-Identity                 IA5String (SIZE(1..9)),
        appointmentTime                         DateTimePair,
        directoryAppointment-Confirmation       INTEGER   (0..255)
}
```

5.4 Mayday Messages—Please see SAE J2313.

5.5 Parking Messages

```
ISPLotBasicInformationRequest ::= SET
{
        lot                             IdentityOrLocation,
        request-Type                    INTEGER   (0..255) OPTIONAL
}

ISPLotDetailInformationRequest ::= SEQUENCE
{
        lot                             IdentityOrLocation,
        lot-InformationType             SEQUENCE OF INTEGER (0..255) OPTIONAL,
        request-Type                    INTEGER                 (0..255) OPTIONAL
}

ISPLotSpaceRequest ::= SEQUENCE
{
        lot                             IdentityOrLocation,
        lotStay-EstimatedDuration       INTEGER      (0..65535),
        requestTime                     DateTimePair  OPTIONAL
}

ISPLotPriceScheduleRequest ::= SEQUENCE
{
        lot                             IdentityOrLocation,
        start                           DateTimePair OPTIONAL,
        end                             DateTimePair OPTIONAL
}

PriceScheduleEntry ::= SEQUENCE
{
        price-DayType                   INTEGER DEFAULT 0,
        time                            TimePair OPTIONAL,
        price-FirstPayment              REAL ,
        price-Maximum                   REAL ,
        times                           SEQUENCE OF TimeInterval
}

LotPriceSchedule ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        lot-Location                    LocationReference,
        price                           PriceSchedule
}

ParkingLotSearchRequest ::= SEQUENCE
{
        traveler-Identity               IA5String (SIZE(1..12)),
        lot-Location                    RouteOrArea,
        lotRequest-SearchRadius         SearchRadius OPTIONAL,
        lotRequest-LocationType         INTEGER          (0..255) OPTIONAL
```

```
}

ParkingLotDetailRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String       (SIZE(1..12)),
        pI-ParkingFacID                         INTEGER,
        lot-InformationType                     SEQUENCE OF INTEGER (0..255) OPTIONAL
}

ParkingSpaceRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        pI-ParkingFacID                         INTEGER,
        lotStay-EstimatedDuration               INTEGER   (0..65535),
        requestTime                             DateTimePair OPTIONAL
}

ParkingLotSearchReply ::= SEQUENCE OF ParkingLotEntries

ParkingLotEntries ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        lot-Type                        INTEGER         (0..255),
        lot-Location                    LocationReference,
        pI-ParkingAvailability          INTEGER OPTIONAL
}

ParkingLotDetailReply ::= SEQUENCE
{
        pI-ParkingFacID                         INTEGER,
        pI-ParkingType                          INTEGER (0..255),
        lot-Location                            LocationReference,
        pI-ParkingSpacesTotal                   INTEGER OPTIONAL,
        pI-ParkingAvailability                  INTEGER OPTIONAL,
        lot-Name                                IA5String (SIZE(1..30)) OPTIONAL,
        pI-ParkingHoursofOperation              IA5String (SIZE(1..12)) OPTIONAL,
        pI-ParkingRates                         INTEGER OPTIONAL,
        pI-ParkingFillTime                      INTEGER (0..255) OPTIONAL
}

LotInformation ::= CHOICE
{
        pI-ParkingSpacesTotal                   INTEGER,
        pI-ParkingAvailability                  INTEGER,
        lot-Name                                IA5String (SIZE(1..30)),
        pI-ParkingHoursofOperation              IA5String (SIZE(1..12)),
        pI-ParkingRates                         INTEGER,
        pI-ParkingFillTime                      INTEGER OPTIONAL,
        pI-ParkingVehicleClass                  IA5String (SIZE(1..12))
}

ParkingLotSpecificDetailReply ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        lotDetail                       SEQUENCE OF LotInformation
}

ParkingSpaceReply ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        pI-ParkingRates                 INTEGER
}

ParkingLotBasicInformation ::= SEQUENCE
```

```
{
        lot-Location                    IdentityOrLocation,
        pI-ParkingType                  INTEGER   (0..255),
        lot-Status                      INTEGER   (0..255)
}
```

5.6 Traveler Information Messages

```
TravelerInformationRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        informationLocation                     RouteOrArea,
        req SEQUENCE OF SEQUENCE {
            -- one or more requested types of data
            informationRequest-Type                     INTEGER (0..255),
            informationRequest-SubType                  INTEGER (0..255) OPTIONAL
            },
        start                                   DateTimePair OPTIONAL,
        end                                     DateTimePair OPTIONAL
}

WeatherInformation ::= SEQUENCE
{
        weatherLocation                         LocationReference,
        weather-ForecastOrActual                ENUMERATED
            {
            actual (0),
            forecast(1)
            },
        weatherDayOfWeek                        ATIS-DayOfWeek,
        weather-Temperature                     INTEGER   (-128..127) OPTIONAL,
        weather-HighTemperature                 INTEGER   (-128..127) OPTIONAL,
        weather-LowTemperature                  INTEGER   (-128..127) OPTIONAL,
        weather-SkyConditions                   INTEGER   (0..255) OPTIONAL,
        weather-Probability                     INTEGER   (0..100) OPTIONAL,
        weather-SpecialConditions               BIT STRING OPTIONAL,
        weather-Visibility                      INTEGER   (0..255) OPTIONAL,
        weather-WindSpeed                       INTEGER   (0..255) OPTIONAL,
        weather-WindDirection                   ENUMERATED
            {
            north                   (0),
            south                   (1),
            west                    (2),
            east                    (3),
            northwest               (4),
            northeast               (5),
            southwest               (6),
            southeast               (7)
            }   OPTIONAL,
        weather-Humidity                        INTEGER   (0..100) OPTIONAL,
        weather-Pressure                        INTEGER OPTIONAL,
        weather-SunriseTime                     GeneralizedTime OPTIONAL,
        weather-SunsetTime                      GeneralizedTime OPTIONAL
}

LinkTrafficInformation ::= SEQUENCE
{
        linkLocation                            LocationReference,
        link-Delay                              INTEGER         (0..12000) OPTIONAL,
        link-Capacity                           INTEGER         (0..300000) OPTIONAL,
        link-Density                            INTEGER         (0..2000) OPTIONAL,
        link-LanesNumberOpen                    INTEGER         (0..50) OPTIONAL,
        link-Occupancy                          INTEGER         (0..100) OPTIONAL,
        link-Speed                              INTEGER         (0..300) OPTIONAL,
```

```
        link-Status                                 OCTET STRING    (SIZE(8)) OPTIONAL,
        link-SurfaceCondition                       BIT STRING      (SIZE(1)) OPTIONAL,
        link-TravelTime                             INTEGER         (0..10800) OPTIONAL
}

NodeTrafficInformation ::= SEQUENCE
{
        nodeLocation                    LocationReference,
        node-Delay                      INTEGER (0..12000) OPTIONAL,
        node-Status                     OCTET STRING (SIZE(8)) OPTIONAL
}

LinkOrNode    ::= CHOICE
{
        link                            LinkTrafficInformation,
        node                            NodeTrafficInformation
}

TrafficInformation ::= SEQUENCE OF LinkOrNode

IncidentInformation ::= SEQUENCE
{
        incidentLocation                                        LocationReference,
        incident-Type                                           INTEGER (0..255),
        event-IncidentSeverity                                  OCTET STRING (SIZE(8)),
        event-IncidentStatus                                    OCTET STRING,
        incident-TimelineConfirmedAndResponding                 DateTimePair OPTIONAL,
        incident-TimelineClearedAndRecovering                   DateTimePair OPTIONAL,
        event-IncidentVehiclesInvolvedCount                     INTEGER (0..255) OPTIONAL,
        event-IncidentVehiclesInvolved                          OCTET STRING (SIZE(8)) OPTIONAL
}

EventSubType   ::= CHOICE
{
        event-DescriptionTypePlannedRoadwayClosure                      OCTET STRING (SIZE(0..255)),
        event-DescriptionTypeSpecialEvent               OCTET STRING
}

EventInformation ::= SEQUENCE
{
        eventLocation                                   LocationReference,
        event-DescriptionTypeEvent                      OCTET STRING (SIZE(8)),
        subType                                         EventSubType OPTIONAL,
        event-Description                               OCTET STRING (SIZE(0..256)) OPTIONAL,
        event-LanesBlockedOrClosedCount                 INTEGER (0..255) OPTIONAL,
--These two lists will be included in the eventLocation in future drafts of the
-- LRMS specification
--They are included here for completeness only
        laneClosedList                                  SEQUENCE OF    INTEGER(0..32) OPTIONAL,
        laneConfigurationList                           SEQUENCE OF    INTEGER(0..32) OPTIONAL,
        event-LanesDirectionOfTravel                    OCTET STRING (SIZE(8)) OPTIONAL,
        event-TimeLineStart                             DateTimePair OPTIONAL,
        event-TimeLineEstimatedDuration                 INTEGER  (0..4294967296) OPTIONAL,
        event-TimeLineScheduledEnd                      DateTimePair OPTIONAL,
        timeList                                SEQUENCE OF IA5String (SIZE(9)) OPTIONAL,
        dayList                                 SEQUENCE OF IA5String (SIZE(10)) OPTIONAL,
        event-TimeLineScheduleDaysOfWeek                ATIS-DayOfWeek
}

ResponseEntity ::= CHOICE
{
        event-ResponsePlanText                  IA5String (SIZE(1..255)),
        detailManeuverInstructions              SEQUENCE OF ManeuverInstructions
```

```
}

ResponsePlan ::= SEQUENCE
{
        event-ResponsePlanType          OCTET STRING (SIZE(8)),
        response                        ResponseEntity
}

LinkAdditionalInformation ::= SEQUENCE
{
        location                        LocationReference,
        link-Name                       IA5String       (SIZE(1..256)) OPTIONAL,
        link-RoadNumber                 IA5String       (SIZE(1..64)) OPTIONAL,
        link-Length                     INTEGER         (0..160000) OPTIONAL,
        link-Capacity                   INTEGER         (0..300000) OPTIONAL,
        link-LanesMinimumNumber         INTEGER         (0..50) OPTIONAL,
        link-ShoulderWidthLeft          INTEGER         (0..999) OPTIONAL,
        link-ShoulderWidthRight         INTEGER         (0..999) OPTIONAL,
        link-MedianType                 OCTET STRING (SIZE(8)) OPTIONAL,
        link-PavementType               OCTET STRING (SIZE(1)) OPTIONAL,
        link-LevelOfService             OCTET STRING (SIZE(1)) OPTIONAL,
        link-SpeedLimit                 INTEGER         (0..300) OPTIONAL,
        link-TruckSpeedLimit            INTEGER         (0..300) OPTIONAL,
        link-RestrictionClass           BIT STRING      (SIZE(1)) OPTIONAL,
        link-RestrictionHeight          INTEGER         (0..2000) OPTIONAL,
        link-RestrictionAxleCount       INTEGER         (0..8) OPTIONAL,
        link-RestrictionLength          INTEGER         (0..6000) OPTIONAL,
        link-RestrictionWidth           INTEGER         (0..2000) OPTIONAL,
        link-RestrictionWeight          INTEGER         (0..50000) OPTIONAL
}

NodeAdditionalInformation ::=SEQUENCE
{
        location                        LocationReference,
        node-Name                       IA5String (SIZE(1..256)) OPTIONAL,
        node-LinksNum                   INTEGER   (0..999) OPTIONAL
}

LinkOrNodeAdditionalInformation ::= CHOICE
{
        link                            LinkAdditionalInformation,
        node                            NodeAdditionalInformation
}

RoadAdditionalInformation ::= SEQUENCE OF LinkOrNodeAdditionalInformation

PollutionInformation ::= SEQUENCE
{
        pollutionLocation               LocationReference,
        pollution-SmogAlert             INTEGER         (0..255) OPTIONAL,
        pollution-AirQualityIndex       INTEGER         (0..255) OPTIONAL,
        pollution-CarbonMonoxide        INTEGER         (0..255) OPTIONAL,
        pollution-HydroCarbon           INTEGER         (0..255) OPTIONAL,
        pollution-SulfurDioxide         INTEGER         (0..255) OPTIONAL,
        pollution-NitrousOxide          INTEGER         (0..255) OPTIONAL,
        pollution-Particulate           INTEGER         (0..255) OPTIONAL,
        pollution-Ozone                 INTEGER         (0..255) OPTIONAL
}

AirlineTravelInformation ::= SEQUENCE
{
        directoryEntry-Name             IA5String (SIZE(1..30)),
        flight-OriginAirport            IA5String (SIZE(3)),
        flight-DestinationAirport       IA5String (SIZE(3)),
```

```
        pI-DepartTimeScheduled              GeneralizedTime,
        pI-ArriveTimeScheduled              GeneralizedTime,
        pI-OffSchedule                      GeneralizedTime OPTIONAL,
        flight-DepartureGate                IA5String (SIZE(1..6)) OPTIONAL,
        flight-ArrivalGate                  IA5String (SIZE(1..6)) OPTIONAL
}

WideAreaTravelInformation ::= SEQUENCE
{
        service-Mode                        IA5String  (SIZE(1..20)),
        directoryEntry-Name                 IA5String  (SIZE(1..30)),
        origin                              LocationReference,
        destination                         LocationReference,
        pI-DepartTimeScheduled              GeneralizedTime,
        pI-ArriveTimeScheduled              GeneralizedTime,
        pI-OffSchedule                      GeneralizedTime OPTIONAL,
        directoryEntry-Cost                 IA5String (SIZE(1..12)) OPTIONAL
}

Amenity ::= SEQUENCE
{
        pI-AmenityID                INTEGER   (0..65535),
        amenity-Type                INTEGER   (0..255),
        pI-AmenityName              IA5String (SIZE(1..25)),
        pI-AmenityStatus            INTEGER   (0..255) OPTIONAL
}

StopPoint ::= SEQUENCE
{
        cPT-StopPointID             IA5String,
        stopLocation                LocationReference,
        cPT-StopPointName           IA5String,
        pI-ParkingProvided          BOOLEAN   OPTIONAL,
        pI-ADAAccess                INTEGER   (0..255)  OPTIONAL,
        cPT-StopPointDescription            IA5String OPTIONAL,
        pI-MarkerType               INTEGER   (0..255) OPTIONAL,
        amenities                   SEQUENCE OF Amenity
}

TimeSchedule ::= SEQUENCE
{
        pI-DepartTimeScheduled              GeneralizedTime,
        pI-ArriveTimeScheduled              GeneralizedTime,
        pI-NextArrivalCountdown             GeneralizedTime OPTIONAL,
        pI-OffSchedule                      GeneralizedTime OPTIONAL
}

TripWalkingDirections ::= IA5String     (SIZE(0..127))

TransitRouteLegInformation ::= SEQUENCE
{
        route-Identity              IA5String,
        service-Mode                IA5String (SIZE(1..20)),
        route-Name                  IA5String,
        route-Description           IA5String OPTIONAL,
        route-SpecialService        BOOLEAN,
        originStop                  StopPoint,
        destinationStop             StopPoint,
        nearestAvailableTime        TimeSchedule,
        startWalkDirections         TripWalkingDirections OPTIONAL,
        endWalkDirections           TripWalkingDirections OPTIONAL
}
```

```
TransitRoute ::= SEQUENCE OF TransitRouteLegInformation

AdvisoryInformation ::= SEQUENCE
{
        advisoryArea                    LocationReference,
        weather   SEQUENCE OF           WeatherInformation OPTIONAL,
        pollution                       PollutionInformation OPTIONAL,
        traffic   SEQUENCE OF           TrafficInformation OPTIONAL,
        incidents SEQUENCE OF           IncidentInformation OPTIONAL,
        events    SEQUENCE OF           EventInformation OPTIONAL
}

ResponseType  ::= CHOICE
{
        weather SEQUENCE OF                     WeatherInformation,
        pollution SEQUENCE OF                   PollutionInformation,
        traffic SEQUENCE OF                     TrafficInformation,
        incidents SEQUENCE OF                   IncidentInformation,
        events SEQUENCE OF                      EventInformation,
        roads SEQUENCE OF                       RoadAdditionalInformation,
        flights SEQUENCE OF                     AirlineTravelInformation,
        widereatravels SEQUENCE OF              WideAreaTravelInformation,
        routes    SEQUENCE OF                   TransitRoute
}

TravelerInformationResponse ::= SEQUENCE OF ResponseType

TravelerBroadcastWrapper ::= SEQUENCE
{
        wrapper   BIT STRING (SIZE(104))
        -- this is in error, the rds frame is 104 bits long
        -- the TMC portion is 32~37 bits long and represents
        -- a table lookup entry
}
```

5.7 Trip Guidance Messages

```
RouteRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        trip-StartDate                          IA5String (SIZE(8)),
        trip-StartTime                          IA5String (SIZE(6..12)),
        trip-OriginLocation                     LocationReference,
        trip-DestinationLocation                LocationReference,
        wayPoint                                SEQUENCE OF WaypointList,
        trip-WayPointPrioritizedListFlag        BOOLEAN,
        preferences                             SEQUENCE OF TripPreferences,
        constraints                             SEQUENCE OF TripConstraints,
        trip-RequestIdentity                    INTEGER   (0..255),
        trip-GuidanceLevel                      ENUMERATED
            {
            maneuversOnly               (0),
            linksBetweenManeuvers       (1),
            connectedLinks              (2)
            },
        trip-MaximumManeuvers                   INTEGER   (0..255) OPTIONAL,
        trip-MaximumLinks                       INTEGER   (0..255) OPTIONAL,
        trip-MaximumConnnectedLinks             INTEGER   (0..255) OPTIONAL
}

WaypointList ::= SEQUENCE
{
        trip-WayPoint                   LocationReference   OPTIONAL
```

```
}

TripPreferences ::= SEQUENCE
{
        trip-PreferenceType             BIT STRING,
        trip-PreferenceSubType          INTEGER (0..255) DEFAULT 0
}

TripConstraints ::= SEQUENCE
{
        trip-ConstraintType             INTEGER (0..255) OPTIONAL,
        trip-ConstraintSubType          INTEGER (0..255) OPTIONAL
}

ConfirmRoute ::= SEQUENCE
{
        traveler-Identity               IA5String (SIZE(1..12)),
        trip-RequestIdentity            INTEGER   (0..255)
}

ModifyRoute ::= SEQUENCE
{
        traveler-Identity                               IA5String (SIZE(1..12)),
        trip-StartDate                                  IA5String (SIZE(8)),
        trip-StartTime                                  IA5String (SIZE(6..12)),
        trip-OriginLocation                             LocationReference,
        trip-DestinationLocation                        LocationReference,
        wayPoint                                        SEQUENCE OF WaypointList,
        trip-WayPointPrioritizedListFlag                BOOLEAN,
        preferences                                     SEQUENCE OF TripPreferences OPTIONAL,
        constraints                                     SEQUENCE OF TripConstraints OPTIONAL,
        trip-RequestIdentity                            INTEGER   (0..255),
        trip-GuidanceLevel                              ENUMERATED
            {
            maneuversOnly               (0),
            linksBetweenManeuvers       (1),
            connectedLinks              (2)
            },
        trip-MaximumManeuvers                           INTEGER   (0..255) OPTIONAL,
        trip-MaximumLinks                               INTEGER   (0..255) OPTIONAL,
        trip-MaximumConnnectedLinks                     INTEGER   (0..255) OPTIONAL,
        trip-RouteIdentity                              INTEGER   (0..255)
}

CancelRoute ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        trip-RequestIdentity                    INTEGER   (0..255),
        trip-RouteIdentity                      INTEGER   (0..255)
}

FileRoutePlan ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        trip-StartDate                          IA5String (SIZE(8)),
        trip-StartTime                          IA5String (SIZE(6..12)),
        trip-OriginLocation                     LocationReference,
        trip-DestinationLocation                LocationReference,
        wayPoint                                SEQUENCE OF WaypointList,
        trip-WayPointPrioritizedListFlag                BOOLEAN,
        preferences                             SEQUENCE OF TripPreferences   OPTIONAL,
        constraints                             SEQUENCE OF TripConstraints   OPTIONAL,
        trip-RequestIdentity                    INTEGER   (0..255),
        trip-ConfirmRouteFlag                   BOOLEAN
```

```
}

RouteInstructions ::= SEQUENCE
{
        traveler-Identity                   IA5String (SIZE(1..12)),
        trip-StartDate                      IA5String (SIZE(8)),
        trip-StartTime                      IA5String (SIZE(6..12)),
        trip-OriginLocation                 LocationReference,
        trip-DestinationLocation            LocationReference,
        wayPoint                            SEQUENCE OF WaypointList,
        trip-EstimatedTravelTime            INTEGER   (0..65535),
        wayPointTimes                       SEQUENCE OF WayPointTravelTime,
        trip-TotalDriveMiles                INTEGER   (0..65535),
        detailManeuverInstructions          SEQUENCE OF ManeuverInstructions,
        trip-RequestIdentity                INTEGER   (0..255),
        trip-NumberOfTransitStops           INTEGER   (0..255) OPTIONAL,
        trip-EstimatedRouteCost             INTEGER   (0..65535) OPTIONAL,
        pI-DepartTimeScheduled              IA5String (SIZE(4..12)) OPTIONAL,
        trip-LinkLocation                   LocationReference OPTIONAL,
        trip-RouteIdentity                  INTEGER   (0..255)
}

WayPointTravelTime ::= SEQUENCE
{
trip-EstimatedWayPointTravelTime   IA5String (SIZE(4..12))
}

ManeuverInstructions ::= SEQUENCE
{
        trip-ManeuverCode                   INTEGER   (0..255),
        trip-ManeuverAngle                  INTEGER   (0..359),
        trip-ManeuverLocation               LocationReference,
        trip-CompoundManeuverFlag           BOOLEAN   DEFAULT FALSE,
        trip-MilesToNextManeuver            INTEGER   (0..65535)
}

RouteUpdates ::= SEQUENCE
{
        traveler-Identity                   IA5String (SIZE(1..12)),
        trip-EstimatedTravelTime            INTEGER   (0..65535),
        wayPointTimes                       SEQUENCE OF WayPointTravelTime,
        trip-TotalDriveMiles                INTEGER   (0..65535),
        detailManeuverInstructions          SEQUENCE OF ManeuverInstructions,
        trip-RequestIdentity                INTEGER   (0..255),
        trip-EstimatedRouteCost             INTEGER   (0..65535) OPTIONAL,
        trip-LinkLocation                   LocationReference OPTIONAL
}

RouteErrorMessage ::= SEQUENCE
{
        traveler-Identity                   IA5String (SIZE(1..12)),
        trip-RequestErrorType               INTEGER   (0..255),
        trip-RequestIdentity                INTEGER   (0..255),
        locationAlternatives                SEQUENCE OF LocationReference OPTIONAL
}
```

6. Concept of Operations —This standard identifies five data flow groups that are within the scope of ATIS. The data flows represent a communication between the vehicle or traveler and an entity, such as an ISP or a parking management service. The six groups are: Setting, Directory Services, Parking, Traveler Information, and Trip Guidance. The following section provides a concept of operations based on these five message set groups.

6.1 Setting—The Setting messages are designed to communicate traveler preference and interest information to an ISP, and to authorize the ISP to use this information in subsequent information transfers. This provides a way for the traveler to alert an ISP that a particular preference or request is not merely a one-time happening but a regular feature of his/her travel. Note that there is no demand mechanism that an ISP can use to identify a traveler, and the submission of Setting information is always optional. The submission of Setting information must conform to the standard only in cases where a traveler asks for services that require such information.

The structures and messages that comprise the Setting messages can be grouped into three broad categories: basic traveler information, traveler preference information, and traveler trigger information.

6.1.1 BASIC TRAVELER INFORMATION—Basic Traveler Information messages provide important personal information to the ISP through the following messages:

a. TravelerContactSetting
b. TravelerExtendedSetting
c. TravelerDeviceSetting

The TravelerContactSetting includes the traveler and setting identities, name information, and a sequence of one or more TravelerContactInformation structures:

```
TravelerContactSetting ::= SEQUENCE
{
        traveler-Identity                   IA5String       (SIZE(1..12)),
        setting-Identity                    INTEGER         (0..255),
        traveler-FirstName                  IA5String       (SIZE(1..25)) OPTIONAL,
        traveler-LastName                   IA5String       (SIZE(1..25)) OPTIONAL,
        contact   SEQUENCE OF TravelerContactInformation
}
```

The traveler-Identity is used across a number of ATIS functions to uniquely identify the traveler. Since there is no existing standard for assigning unique identities, this field is open to ISP/Device discretion. The setting-Number field is used to uniquely identify a specific setting. This number should be assigned by the sending device and it is the responsibility of the sending software to pick a unique number within its own domain. The setting-Number is not intended to be unique across travelers, only unique within a traveler-Identity. The function of setting-Number is to allow a single traveler (or single traveler device) to have multiple settings that can be invoked with the ISP.

The sequence of TravelerContactInformation structures contained in the setting enable the ISP to communicate with the traveler:

```
TravelerContactSetting ::= SEQUENCE
{
        traveler-Identity                   IA5String       (SIZE(1..12)),
        setting-Identity                    INTEGER         (0..255),
        traveler-FirstName                  IA5String       (SIZE(1..25)) OPTIONAL,
        traveler-LastName                   IA5String       (SIZE(1..25)) OPTIONAL,
        contact   SEQUENCE OF TravelerContactInformation
}

TravelerContactInformation    ::= SEQUENCE
{
        traveler-Identity                   IA5String       (SIZE(1..12)),
        information                         ContactInformation,
        contactDayOfWeek                    ATIS-DayOfWeek  OPTIONAL,
        contactTimes                        TimePair        OPTIONAL
```

```
}

ContactInformation      ::= CHOICE
{
        traveler-EMail          IA5String (SIZE(1..40)),
        phone                   PhoneInformation,
        traveler-Pager          NumericString (SIZE(1..20)),
        device-Identity         OCTET STRING
}
```

The contact-Mode specifies a particular way that the ISP can reach the traveler. Examples of modes would be home phone, business phone, home fax, office fax, in-car device, etc. For each mode there is a contact-Information field that contains the relevant information (such as a phone number, email address, device identity, etc.). Note that there is not a unique form of contact information for every contact mode. Thus, some modes such as business phone and home fax use the same contact-Information type (i.e., phone). Optionally, the user may attach a contact day type and time for which the contact information applies.

The DayOfWeek type is a bit string structure with eight positions. The first 7 bits (0-6) are assigned to each day of the week (Sunday-Saturday). An application can specify any combination of weekdays by turning on the appropriate bit. The final bit (7) is reserved as a flag to indicate whether holidays should be included. Holidays in ATIS are defined as bank holidays. Setting the final bit indicates that the day type specification holds for all days including holidays. If the final bit is 0, then the day type specification is presumed to apply only during “normal” days.

Because there can be more than one TravelerContactInformation for a traveler (it is sent as a SEQUENCE) it is essential that both sender and ISP agree on which setting is to be used first. The order of use (and therefore of transmission) should be the first contact setting sent which is not specifically constrained to be outside the current day-type and current time.

There are some areas within ATIS, such as Settings, that may require some level of customization and/or localization by the ISP (see 5.4). Since there is no way to foresee what kind of customizations might be desired, the ATIS standard supplies an ExtendedInformationSetting Message which is essentially a place-holder for customization:

```
TravelerExtendedSetting ::= SEQUENCE
{
        traveler-Identity               IA5String       (SIZE(1..12)),
        setting-Identity                INTEGER         (0..255),
        setting-ExtendedInformation     OCTET STRING
}
```

Because the setting-ExtendedInformation is simply described as an OCTET, it can contain any binary information in pre-formatted fashion that is desired by the ISP and Traveler. This information could include encrypted payment information, additional personal information, or extended device information.

The ATIS Standard also provides a TravelerDeviceSetting message, intended to provide the ISP with information about the traveler’s current device and its capabilities. This information can be used by the ISP to more intelligently manage the dialog between the ISP and the traveler.

6.1.2 TRAVELER PREFERENCE INFORMATION—Traveler preference information is communicated via the PreferenceSetting message. This message contains the traveler and setting identities (see discussion above) and a sequence of TravelerPreference structures:

```
PreferenceSettings ::= SEQUENCE
{
        traveler-Identity           IA5String       (SIZE(1..12)),
        setting-Identity            INTEGER         (0..255),
        settingsDetail              SEQUENCE OF     TravelerPreference
}
```

Each Traveler Preference contains a single preference expression, consisting of a setting-Type that indicates what kind of preference is being communicated, a preferenceExpression which contains the contents of the preference, an operator (atis-SearchOperator) that specifies how the contents of the preference are to be treated, and, optionally, information about costs within the preference type and expression.

```
TravelerPreference ::= SEQUENCE
{
        setting-Type                INTEGER,
        expression                    PreferenceExpression,
        atis-SearchOperator         INTEGER   (0..255) DEFAULT 0,
        atis-CostPreference         INTEGER   OPTIONAL,
        atis-CostPreferenceAmount     INTEGER             (0..65535) OPTIONAL
}
```

This is a moderately complex structure and it provides a considerable degree of flexibility in the specification of traveler preferences. The setting-Type is to some extent redundant to the preferenceExpression (in ASN.1 the CHOICE variable could be used without the setting-Type since the specification carries sufficient information to differentiate between pair types) but both language(0) and other(3) will presumably share the use of TypePreferencePairs and the use of the setting-Type allows the application to use multiple types of preferenceExpressions with a minimum of effort.

The preferenceExpression allows the user to specify a type and sub-type within an application area. These codes are dependent on the type of preference being specified. For Directory Services, a DirectoryTypePair is specified. The DirectoryTypePair consists of a Directory Type (which is the domain of NAICS codes) and a Sub-Type (a further refinement for certain NAICS categories). Thus, the user could specify Restaurant as Type and familyItalian as a Sub-type. This DirectoryTypePair would then constitute a preferenceExpression. Attached to each preferenceExpression is an atis-SearchOperator. This entity is part of Directory Services and is used to specify how the preferenceExpression is to be interpreted. It is an integer and the defined values are: Must Include, Should Include, Should Not Include, and Must Not Include. These values are similar to those employed on web search engines and allow the user to specify whether a preference is positive or negative, a suggestion (SHOULD) or a constraint (MUST).

Finally, the TravelerPreference structure contains an atis-CostPreference and an atis-CostPreferenceAmount field. These fields form an optional pair in which the user can specify some preference regarding cost. The cost preference allows the user to specify a preference for lowest cost (this could control what is returned - say between two trip routes - or what order items are returned in - as in a listing of restaurants), or a preference for a cost and an amount. When the atis-CostPreference is maximumCost, minimumCost or targetCost then the message must also include the atis-CostPreferenceAmount. Note that the specification includes a lowest-cost preference but not the less-intuitive highest-cost preference. However, a Minimum cost can be specified for cases when a traveler is looking for a certain class of experience (e.g., I want French Restaurants where the Estimated Price is at least $30 per person). Target Cost allows the traveler to indicate a general price range (I want Restaurants where the Estimated Price is around $20 per person). Note that the Standard makes no assumption about what the exact definition of price is in the context of any specific entity.

Taken together, it can be seen that the TravelerPreference can express a wide variety of preference types and attitudes. The PreferenceSetting allows the traveler to provide a sequence of one or more of these structures - potentially representing a wide array of traveler interests and concerns. The Standard does make it possible for the Traveler to communicate contradictory or even impossible requests (A MUST preference for Italian Restaurants followed by a MUST NOT preference for Italian Restaurants). Ideally, the user device should be sensitive to logical or qualitative contradictions and alert the user to their existence. The ISP is not required to do this within the ATIS Standard. However, the ATIS Standard does include a generalized Error/Information Message that could be used for this purpose (ErrorOrInformationNotification). Finally, it should be realized that the setting is always overridden by a specific request where there is a direct conflict. If the user sends a Directory request for Italian Restaurants, this request would be fulfilled even where the setting contains a MUST NOT preference for Italian Restaurants.

Each PreferenceSetting contains a whole set of preferences. Multiple PreferenceSettings exist so that a single device can easily support multiple travelers (as in a one-car family) with specific preferences or where a single device may wish to support multiple types of generic traveler settings (as in a kiosk).

To differentiate between settings, all Settings are assigned a unique identifier by the originating device (the setting-Number). This number can be used to uniquely identify the setting to be used for subsequent requests by the traveler. It is important that the traveler identity and setting identity be unique across devices so that the ISP can use the setting appropriately.

Because multiple settings can exist for a traveler, the ATIS Standard includes some limited messages for controlling the setting currently in force and for eliminating obsolete settings. The SelectPreferenceSetting provides a mechanism for choosing the current setting to be used for subsequent requests. The Selection should remain in force until a further Selection is specified. The DeletePreferenceSetting is used to remove an obsolete Setting. If the currently selected Setting is deleted, then no setting should be in force for subsequent requests even where other settings exist.

```
SelectPreferenceSetting ::= SEQUENCE
{
        traveler-Identity               IA5String       (SIZE(1..12)),
        setting-Identity                INTEGER         (0..255)
}

DeleteSetting ::= SEQUENCE
{
        traveler-Identity               IA5String       (SIZE(1..12)),
        setting-Identity                INTEGER         (0..255)
}
```

The ATIS Standard provides no pure update or modify function for a setting. The user should delete and re-send a new setting to accomplish an update.

6.1.3 TRAVELER TRIGGER SETTINGS—The Traveler Trigger Setting provides a way for a traveler to “register” an interest in a particular route or location with an ISP. Part of the trigger allows the traveler to select a class of events that will activate the trigger. Events can be traffic problems, weather problems, planned events, changes in transit routes, transit delays, or anything that could generate an advisory in the Traveler Information function. In addition to defining a location and an event for a trigger, the user can attach date and time information to qualify when the trigger should be in effect, when it should be ignored, and when it should expire. All of the date and time information is optional in the Trigger Setting; when not specified, the default is to start the trigger immediately and leave in place until further notice.

Once the traveler has registered an interest using a Trigger, the ISP is responsible for monitoring the specified location or area for the trigger event. If it occurs, the ISP should then notify the traveler. Notification is not specifically dealt with by the ATIS Standard. The method of notification should be determined by the Traveler Contact Setting. The Contact Setting may request notification via a traveler device such as an in-vehicle navigation system. In this case, the notification would be via the Message Set Traveler Advisory functions. However, it also feasible that the notification could be via email or fax or other off-system mechanism.

```
TriggerSetting ::= SEQUENCE
{
        traveler-Identity               IA5String (SIZE(1..12)),
        setting-Identity                INTEGER   (0..255),
        location                        RouteOrArea,
        trigger-Event                   INTEGER   (0..255),
        start                           DateTimePair  OPTIONAL,
        end                             DateTimePair OPTIONAL,
        timeEnforced                    TimePair OPTIONAL,
        triggerDayOfWeek                ATIS-DayOfWeek OPTIONAL
}
```

As with other Settings, the Trigger Setting includes both the Traveler Identity and a device generated unique setting number. This allows a user to operate on the Trigger once it is in place - specifically, this allows the user to delete a Trigger Setting using the DeleteSetting message.

All triggers will apply to some geographic area, location or object. The trigger location is specified using the RouteOrArea construct.

```
RouteOrArea ::= CHOICE
{
        route           CoreRouteRequest,
        area            IdentityOrLocation
}

CoreRouteRequest ::= SEQUENCE
{
        trip-OriginLocation             LocationReference,
        trip-DestinationLocation        LocationReference,
        wayPoint                        SEQUENCE OF LocationReference
}

IdentityOrLocation ::= CHOICE
{
        location        LocationReference,
        dbIdentity      DatabaseIdentity
}
```

The RouteOrArea is designed to allow the Traveler considerable flexibility in specifying an area of interest. The structure is a CHOICE between either a RouteRequest or an IdentityOrLocation. For a RouteRequest, only three components are used: a trip Origin, a trip Destination, and sequence of wayPoints. Note that each of these objects is, itself, a LocationReference.

In addition, the flexibility of the IdentityOrLocation structure also provides a great deal of capability to the user. The contents can be a database identity or any one of the LocationReference settings. These settings provide the ability to define a single point, a street, a building, a set of streets, an area around a set of streets, or an area on a map. In fact, the use of a RouteRequest could be replaced by a complex LocationReference. The RouteRequest construct is included in the routeOrArea to make it easier for systems to specify a route in a single, uniform fashion.

The Date/Time specification allows a trigger to be put in place “for all time” or for a specified duration. For the specified start and end dates, it is also possible for the user to specify a particular time when a trigger is enforced (timeEnforced). This can be used, for example, to limit the Trigger to commute hours. The user can also specify which days of the week the Trigger applies to. For a complete discussion of the DayOfWeek structure see above.

Although the TriggerSetting is set up as a single message, a user may send multiple TriggerSetting messages all of which are in force. The TriggerSetting was created as a single message structure as opposed to a SEQUENCE of triggers because of the likelihood that a TriggerSetting will be created singly to meet a specific need.

6.2 Directory Services —Directory Services (also known as “Yellow Pages”) form a core component of the ATIS message set. The Directory Services message set provides three broad types of functionality:

a. One-way Traveler Information messages
b. Two-way Traveler Search Request and Reply messages
c. Two-way Traveler Appointment/Reservation messages

The one-way traveler information message provides a simple way for a multitude of commercial and government entities to interact with traveler devices. The summarized one-way information flow uses the DirectoryEntry structure within the DirectoryInformation message.

```
DirectoryEntry ::= SEQUENCE
{
        directoryEntry-Identity                 IA5String (SIZE(1..9)),
        directoryEntry-Location                 LocationReference,
        requestPair                             DirectoryTypePair,
        directoryEntry-Name                     IA5String (SIZE(1..30)),
        directoryEntry-Phone                    PhoneInformation OPTIONAL,
        directoryEntry-ShortDescription         IA5String (SIZE(1..40)) OPTIONAL,
        directoryEntry-CapabilityCode           INTEGER   (0..255) OPTIONAL
}

DirectoryInformation ::= SET OF DirectoryEntry
```

Most existing (and presumably future) implementations of Directory Services involve a two-way dialog between the traveler and the ISP. This dialog encompasses at least two messages and will sometimes require an extended dialog between traveler and ISP. The dialog is typically initiated by the traveler with one of the SearchRequest messages.

This basic search request message is:

```
DirectoryCoreRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        entry-Location                          RouteOrArea,
        directorySearch-Radius                  SearchRadius,
        requestPair                             DirectoryTypePair,
        directorySearch-LimitEntriesReturned    INTEGER   (0..65535) OPTIONAL,
        directoryEntry-LocationType             INTEGER   (0..255) OPTIONAL
}

DirectoryRequest ::= SEQUENCE
{
        coreRequest             DirectoryCoreRequest,
        start                   DateTimePair OPTIONAL,
        end                     DateTimePair OPTIONAL
}
```

The DirectoryCoreRequest structure includes a traveler identity (allowing setting mechanisms to be applied), a location for the request (RouteOrArea), a search radius, a directory type and sub-type, and an optional limit to the number of entries returned by the ISP.

The location can be specified as a RouteOrArea. This constuct allows the traveler to define a number of different kinds of location request. The RouteOrArea can be a trip route returned by the ISP or generated by the travelers device (see Integrating Trip Guidance for more information). The ISP should interpret the Route as a request for directory services at the Way Points and the Destination. The RouteOrArea may also be a standard location reference.

The directoryEntry-LocationType allows the traveler to specify the type of location reference that the ISP should use when returning results. This is an optional field that can be used by the traveler or the traveler's device to insure that the location settings returned by the ISP are of the most useful type for the receiving device.

The ISP replies to the directory services request with a list of directory entries. By choosing which fields to deliver from the DirectoryEntry structure, the ISP has considerable control over the bandwidth demands of the directory services subsystem.

When there is only a single DirectoryEntry structure to return (possibly because the traveler used a location profile specifying an address), then the ISP should use the DirectoryDetailReply message:

```
DirectoryDetailReply ::= SEQUENCE
{
        directoryEntry-Identity                          IA5String (SIZE(1..9)),
        directoryEntry-Location                          LocationReference,
        typePair                                         DirectoryTypePair,
        directoryEntry-Name                              IA5String (SIZE(1..30)),
        directoryEntry-Phone                             PhoneInformation,
        directoryEntry-HandicapAccess                    INTEGER   (0..255) OPTIONAL,
        directoryEntry-BusinessHours                     IA5String (SIZE(1..12)) OPTIONAL,
        directoryEntry-Description                       IA5String (SIZE(1..200)) OPTIONAL,
        directoryEntry-Cost                              IA5String (SIZE(1..12)) OPTIONAL,
        directoryEntry-ExtendedInformationAvailable                          BOOLEAN OPTIONAL
}
```

ATIS will also support two extensions to this dialog. First, ATIS will provide a mechanism (the directory entry ExtendedInformationAvailable) for the ISP to inform the Traveler that ISP value-added information is available for a directory entry. ATIS will also support a Traveler Request for Extended Information, a Traveler Device Setting for Extended Information Capabilities (formats the Traveler Device can support), and an ISP mechanism for passing any binary object as extended information within the Message Set. This extension provides the ISP a way of sending additional rich text information, database information, graphics, sound, or video to the traveler as appropriate.

```
DirectoryExtendedInformationRequest ::= SET
{
        directoryEntry-Identity                          IA5String (SIZE(1..9)),
        directoryEntry-ExtendedInformationType           INTEGER   (0..255)
}

DirectoryExtendedInformationEntry ::= SET
{
        directoryEntry-Identity                          IA5String (SIZE(1..9)),
        directoryEntry-ExtendedInformationType           INTEGER   (0..255),
        directoryEntry-ExtendedInformation               OCTET STRING
}

DirectoryExtendedInformationReply ::= SET OF DirectoryExtendedInformationEntry
```

As a second extension, ATIS will support the ability to request reservations or appointments for directory entry items where the ISP can provide this service. This extension is implemented through a CapabilityCode on each directory entry (allowing the ISP to support this capability for some subset of ENTRIES or Merchants), a Traveler to ISP message requesting an appointment, and a ISP to Traveler confirmation or response.

The appointment mechanism will include natural hooks allowing for payment mechanisms to be added to the overall message flows. However, in keeping with the determination made during the element selection phase, commercial transactions will not be supported in this phase.

```
DirectoryAppointmentRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        directoryEntry-Identity                 IA5String (SIZE(1..9)),
        appointmentTime                         DateTimePair
}

DirectoryAppointmentReply ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        directoryEntry-Identity                 IA5String (SIZE(1..9)),
        appointmentTime                         DateTimePair,
        directoryAppointment-Confirmation       INTEGER   (0..255)
}
```

In many respects, this operational concept for Directory Services echoes that specified in the National Architecture. The major extensions beyond the architecture are in the variety of search mechanisms and capabilities supported. The concept of a Search Radius was borrowed from the Atlanta showcase and most of the basic flows echo that implementation. However, most existing ATIS implementations and specifications have implementing searching based exclusively on directory type and sub-type. This division mirrors a classic yellow-pages, and forms the core search mechanism implemented here. As powerful and natural as category based searching is, it does not satisfy a number of likely search situations. And since Directory Services is a important area for ISP value-add, the ATIS message set has been designed to support several additional kinds of search.

Most important of these is the ability to search on a name or name fragment. One existing system has implemented a phonetic name. The decision was made to simply allow passing a name or name fragment and the ISP could choose to implement either a phonetic or traditional search or some combination of the two.

```
DirectoryNameRequest ::= SEQUENCE
{
        basicRequest                            DirectoryCoreRequest,
        directoryEntry-Name                     IA5String (SIZE(1..30)),
        start                                   DateTimePair OPTIONAL,
        end                                     DateTimePair OPTIONAL,
        directoryEntry-LocationType             INTEGER   (0..255) OPTIONAL
}
```

ATIS will also support a Search Request allowing multiple Category/Sub-Category pairs to be enumerated:

```
DirectoryAdvancedRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        traveler-Location                       RouteOrArea,
        typePair SEQUENCE OF                    DirectoryTypePair,
        directorySearch-Radius                  SearchRadius OPTIONAL,
        directorySearch-LimitEntriesReturned    INTEGER   (0..65535) OPTIONAL,
        start                                   DateTimePair OPTIONAL,
        end                                     DateTimePair OPTIONAL,
        directoryEntry-LocationType             INTEGER   (0..255) OPTIONAL
}
```

Finally, the Standard provides an extended keyword search. The keyword search model implemented in the Extended Search Message is based on current practice on Internet search engines. The Extended Search Message allows an application to specify one or more keywords or phrases. Each keyword or phrase includes an Operator with four possible values: must include, should include, should not include, and must not include.

```
DirectoryExtendedRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        traveler-Location                       RouteOrArea,
        directorySearch-Radius                  SearchRadius OPTIONAL,
        directorySearch-LimitEntriesReturned            INTEGER   (0..65535) OPTIONAL,
        typePair SEQUENCE OF                    DirectoryTypePair OPTIONAL,
        keywordPair SEQUENCE OF                 DirectoryKeywordPair OPTIONAL,
        directorySearch-ConstrainByProfile      BOOLEAN,
        directorySearch-OrderEntriesBy          INTEGER   (0..255),
        start                                   DateTimePair OPTIONAL,
        end                                     DateTimePair OPTIONAL,
        directoryEntry-LocationType             INTEGER   (0..255) OPTIONAL
}
```

The Extended ATIS Search also includes the ability to specify a return order. This will control both the physical transmission of returned elements in the list and the elements actually included when a limit on entries is in force from either the application or the ISP. It is important that the Order By operation be applied before the entry limit is enforced.

Finally, the ATIS Extended Search includes a constraining setting. Where a user has a registered setting, the search request can ask that it be used as part of a search operation. For instance, a setting might specify a restaurant type and sub-type preference ordering. An extended Search request for a restaurant could request that the setting be used to further prioritize entries. As with the Order By clause, setting preference ordering should take place before any limit on returned entries is enforced. The device setting can be used by the ISP to determine the user application's ability to handle extended information types.

6.3 Mayday—Please see SAE J2313—Draft Recommended Practice: On-Board Land Vehicle Mayday Reporting Interface.

6.4 Parking—The Parking Message Group includes a set of messages designed to provide travelers with extensive information about parking facilities, their current occupancy and availability, and the cost of a prospective parking stay. This group includes a set of messages that standardize the real-time flow of information between ISPs and Parking Management. Parking includes a message designed to support one-way information flows to the traveler:

```
ParkingLotBasicInformation ::= SEQUENCE
{
        lot-Location                    IdentityOrLocation,
        pI-ParkingType                  INTEGER         (0..255),
        lot-Status                      INTEGER         (0..255)
}
```

The lot-Location is the IdentityOrLocation structure described in detail in the section on location referencing. It allows an ISP to describe a parking facility using an identity or as a standard location reference. The standard location reference could include any of the profiles available under ATIS.

The ParkingType field is a code that specifies the type of parking facility being described. The ISP can describe the current status of public, private, and street parking spaces. The lot-Status field describes the percentage of available parking spaces still available. Zero indicates a completely full facility. A one indicates that less than 10% of the spaces in the facility are available. This continues up to a maximum value of 10, which indicates that between 90 to 100% of the spaces in the facility are still available.

This very short Basic message is supplemented by a set of messages that support a more sophisticated dialog between the traveler and the ISP. The dialog is initiated by the traveler with a parking search request:

```
ParkingLotSearchRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String           (SIZE(1..12)),
        lot-Location                    RouteOrArea,
        lotRequest-SearchRadius                 SearchRadius OPTIONAL,
        lotRequest-LocationType                 INTEGER             (0..255) OPTIONAL
}
```

The traveler-Identity is included in the message to allow the ISP to employ settings to refine search results. The lot-Location can be specified as a RouteOrArea. This constuct allows the traveler to define a number of different kinds of location request. The RouteOrArea can be a trip route returned by the ISP or generated by the travelers device (see Integrating Trip Guidance for more information). The ISP should interpret the Route as a request for parking at the Way Points and the Destination. The RouteOrArea may also be a standard location reference. The traveler can specify a location as a point and use the SearchRadius construct to request either a polygon or a simple radius around the point within which a lot should be returned. The traveler can also specify the lot-Location as a Geometric location - potentially a complex polygon using either the Geometry or Grid profiles. When specifying a geometric profile with the RouteOrArea, the traveler should ignore the SearchRadius and allow it to default to 0. Finally, the traveler can specify a single location using the Address or CrossStreets profiles without a Search Radius. In this mode, the traveler requests information on a single lot. This mechanism allows the traveler to obtain detailed information on a lot directly. The lotRequest-LocationType allows the traveler to specify the type of location reference that the ISP should use when returning results.

The ISP can reply to the ParkingLotSearchRequest with the ParkingLotSearchReply or the ParkingLotDetailReply. The ISP should use the first message when more than one parking facility fits the traveler requirements. When just one lot meets the search requirements specified in the ParkingLotSearchRequest then the ISP should use the second message.

```
ParkingLotSearchReply ::= SEQUENCE OF ParkingLotEntries
ParkingLotEntries ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        lot-Type                        INTEGER   (0..255),
        lot-Location                    LocationReference,
        pI-ParkingAvailability          INTEGER OPTIONAL
}

ParkingLotDetailReply ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        pI-ParkingType                  INTEGER   (0..255),
        lot-Location                    LocationReference,
        pI-ParkingSpacesTotal           INTEGER OPTIONAL,
        pI-ParkingAvailability          INTEGER OPTIONAL,
        lot-Name                        IA5String (SIZE(1..30))  OPTIONAL,
        pI-ParkingHoursofOperation              IA5String (SIZE(1..12))     OPTIONAL,
        pI-ParkingRates                 INTEGER OPTIONAL,
        pI-ParkingFillTime              INTEGER   (0..255)                  OPTIONAL
}
```

The sequence of ParkingLotEntry(s) allows the ISP to send a list of potential parking options available to the traveler. The ParkingLotEntry provides an identity, location, type and status for a lot. The location, type and status are as already discussed. Naturally, when an ISP specifies the location for a lot it will be in an appropriate profile such as CrossStreets or Address. If the traveler has requested a specific type of location profile in the request then the ISP should honor that request in the reply.

When the ISP returns a list of parking facilities to a request, the traveler has two options for obtaining additional information. First, the traveler can use the location reference of a returned lot to issue a second ParkingLotSearchRequest with a SearchRadius of zero. Since only a single facility will meet the requirements of the request, the ISP will respond with the ParkingLotDetailReply.

The traveler can also use the pI-ParkingFacID provided in the returned list to issue a simpler request for additional information. The ParkingLotDetailRequest supports this mechanism:

```
ParkingLotDetailRequest ::= SEQUENCE
{
        traveler-Identity                   IA5String                       (SIZE(1..12)),
        pI-ParkingFacID                     INTEGER,
        lot-InformationType         SEQUENCE OF INTEGER (0..255) OPTIONAL
}

ParkingLotSpecificDetailReply ::= SEQUENCE
{
        pI-ParkingFacID             INTEGER,
        lotDetail                   SEQUENCE OF LotInformation
}

LotInformation ::= CHOICE
{
        pI-ParkingSpacesTotal               INTEGER,
        pI-ParkingAvailability              INTEGER,
        lot-Name                            IA5String (SIZE(1..30)),
        pI-ParkingHoursofOperation          IA5String (SIZE(1..12)),
        pI-ParkingRates                     INTEGER,
        pI-ParkingFillTime                  INTEGER OPTIONAL,
        pI-ParkingVehcileClass              IA5String (SIZE(1..12))
}
```

The ParkingLotDetailRequest is both a short-hand request for information on a specific facility and a way for the traveler to request a specific piece of information from the ISP. The lot-InformationType is an enumeration of the fields available from the ISP. The traveler can send one or more these to essentially custom configure the return message about the parking facility.

Once the traveler has identified a particular lot with availability, the second phase of parking dialog can be initiated. The second phase allows the traveler to ascertain the likely cost of a parking stay.

```
ParkingSpaceRequest ::= SEQUENCE
{
        traveler-Identity                   IA5String (SIZE(1..12)),
        pI-ParkingFacID                     INTEGER,
        lotStay-EstimatedDuration           INTEGER   (0..65535),
        requestTime                         DateTimePair OPTIONAL
}
```

The ParkingSpaceRequest provides the traveler identity, lot identity, time of request (not a timestamp but the time at which the traveler expects to arrive at the facility), and the estimated duration of the stay. The request time and estimated duration are essential to determining a lot cost. The requestTime field is optional and if it is not specified the ISP should assume that the request begins at the current time.

The ISP replies to the ParkingSpaceRequest with a ParkingSpaceReply. The reply just includes the lot and the estimated cost of the stay. The cost of the stay should be the best estimate based on the lot, the entrance time and the estimated duration. The ISP can derive the estimate from the detailed price schedule information passed from Parking Management to the ISP.

```
ParkingSpaceReply ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        pI-ParkingRates                 INTEGER
}
```

The ISP to Parking Management messages are similar in content and format to the messages between the ISP and the traveler. The ISP uses the lot location or identity to query Parking Management for information. A basic request will return the current availability of the lot. The ISP can also request detailed information and like the traveler, the request can be custom configured so that the ISP obtains just the desired information in the reply.

The ISP does have an additional field not available to the traveler. The request-Type field allows the ISP to indicate a level of interest or update period on the information. If this field is omitted, the request is assumed to be a one-time request for information. Using this field, the ISP can request ongoing updates at the specified interval.

```
ISPLotBasicInformationRequest ::= SET
{
        lot                             IdentityOrLocation,
        request-Type                    INTEGER         (0..255) OPTIONAL
}

ISPLotDetailInformationRequest ::= SEQUENCE
{
        lot                             IdentityOrLocation,
        lot-InformationType             SEQUENCE OF INTEGER     (0..255) OPTIONAL,
        request-Type                    INTEGER                 (0..255) OPTIONAL
}
```

The ISP can pass a single space request to Parking Management and receive the standard space cost reply. However, in most cases, it is presumed that the ISP will choose to maintain price information for the lot and respond to space requests by calculating an estimated price without having to communicate with Parking Management.

```
ISPLotSpaceRequest ::= SEQUENCE
{
        lot                             IdentityOrLocation,
        lotStay-EstimatedDuration               INTEGER         (0..65535),
        requestTime                     DateTimePair            OPTIONAL
}
```

To support the ability of the ISP to calculate prices, the ATIS Standard includes a fairly complex mechanism that allows the ISP to request and receive pricing schedules. The schedules are complex because of the many kinds of options that can affect price. The basic request is for price schedules between some pair of dates.

```
ISPLotPriceScheduleRequest ::= SEQUENCE
{
        lot                             IdentityOrLocation,
        start                                   DateTimePair OPTIONAL,
        end                             DateTimePair OPTIONAL
}
```

The reply is a sequence of PriceSchedules for that period. The PriceSchedule covers some period of dates (possibly all) within the requested time period. The PriceSchedule, in turn, contains a sequence of PriceScheduleEntry structures. The PriceScheduleEntry includes the day types for which the schedule applies, the start and end times for which the schedule applies, the first payment for situations where there is an initial cost different than for subsequent periods, a maximum cost for situations where there is a cap cost for the specified period (as in maximum daily rates for parking), and then a sequence of TimeInterval structures. Each TimeInterval contains a time period (such as 20 minutes or an hour) and a TimeValue which is the cost for that time period.

```
PriceSchedule ::= SEQUENCE
{
        start                                   DateTimePair OPTIONAL,
        end                             DateTimePair OPTIONAL,
        entries                         SEQUENCE OF PriceScheduleEntry
}

TimeInterval ::= SEQUENCE
{
        price-TimeInterval                      INTEGER (0..65535),
        price-TimeValue                         REAL
}

PriceScheduleEntry ::= SEQUENCE
{
        price-DayType                   INTEGER DEFAULT 0,
        time                            TimePair OPTIONAL,
        price-FirstPayment              REAL ,
        price-Maximum                   REAL ,
        times                           SEQUENCE OF TimeInterval
}

LotPriceSchedule ::= SEQUENCE
{
        pI-ParkingFacID                 INTEGER,
        lot-Location                    LocationReference,
        price                           PriceSchedule
}
```

This complex structure permits Parking Management to communicate very complex pricing schemes to the ISP in a consistent fashion. The ISP, however, does not pass the schedules themselves to the traveler. Instead, the ISP is responsible for using the information in the schedule for the calculations that drive the replies to specific parking space requests.

6.5 Traveler Information—Traveler Information encompasses a wide variety of ATIS data elements. It includes information on traffic, incidents, events, weather, environmental conditions (pollution), wide-area travel, and public transit schedules and services. Because much of this information comes from other centers - especially Traffic Management and Transit Control, a great many of the data elements which support these messages are common to those data dictionaries.

Traveler Information provides two paradigms for communication with the Traveler:

a. One-way Traveler Information messages
b. Two-way Traveler Information messages (including area or route-specific and type-specific messages)

6.5.1 ONE-WAY TRAVELER INFORMATION MESSAGES—One-way Traveler Information messages are intended to provide wide, potentially low-bandwidth access to information likely to be important to Travelers. One-way messages include the ability to provide location reference and one or more pieces of information.

```
AdvisoryInformation ::= SEQUENCE
{
        advisoryArea                    LocationReference,
        weather   SEQUENCE OF           WeatherInformation OPTIONAL,
        pollution                       PollutionInformation OPTIONAL,
        traffic   SEQUENCE OF           TrafficInformation OPTIONAL,
        incidents SEQUENCE OF           IncidentInformation OPTIONAL,
        events    SEQUENCE OF           EventInformation OPTIONAL
}
```

The AdvisoryInformation message contains optional sequences of information structures. Each information structure contains a range of fields relevant to a particular type of information. WeatherInformation contains a location reference and a range of fields that detail temperature, sky conditions, precipitation, sunset/ sunrise times, etc. PollutionInformation contains a location reference and measures of particulate matter for the specified location. TrafficInformation provides a means of detailing specific information about traffic flow in an area, while IncidentInformation and EventInformation provide greater detail about specific traffic problems.

Because most of the information within a structure is OPTIONAL and because each of the structures is itself an optional SEQUENCE, an AdvisoryInformation message could be as short as a location reference plus one field of information (such as pollution-SmogAlert) or could contain multiple instances of traffic information, incidents, events, weather conditions and pollution measurements each containing many different fields.

In addition to the AdvisoryInformation, the ATIS Standard provides a wrapper intended to support backwards compatibility with phrase-based systems.

```
TravelerBroadcastWrapper ::= SEQUENCE
{
        wrapper                 BIT STRING (SIZE(104))
        -- this is in error, the rds frame is 104 bits long
        -- the TMC portion is 32~37 bits long and represents
        -- a table lookup entry
}
```

6.5.2 TWO-WAY TRAVELER INFORMATION MESSAGES—The messages that support two-way traveler information dialogs are structured very much like the AdvisoryInformation message. The basic request message is the TravelerInformationRequest:

```
TravelerInformationRequest ::= SEQUENCE
{
        traveler-Identity                       IA5String (SIZE(1..12)),
        informationLocation                     RouteOrArea,
        req SEQUENCE OF SEQUENCE {
                -- one or more requested types of data
                informationRequest-Type                 INTEGER     (0..255),
                informationRequest-SubType              INTEGER     (0..255) OPTIONAL
                },
        start                                   DateTimePair OPTIONAL,
        end                                     DateTimePair OPTIONAL
}
```

The TravelerInformationRequest provides the ability to request information (based on one or more vaules of the informationRequest-Type) for a RouteOrArea (the Route is a sequence of LocationReferences and the Area is a single location reference). A basic TravelerInformationRequest includes the traveler's identity, the location, and a request type (such as weather). The response to a TravelerInformationRequest with a type specified but no sub-type, start or end should be a one-time reply containing all available information for the requested type(s). The reply is encapsulated in the TravelerInformationResponse: