



## **OpenTravel™ Alliance 2001B Message Specifications**

### **Abstract**

This document presents the specifications for the exchange of messages in the travel industry, covering travel services for airlines, car rentals, hotels, and travel insurance. It uses the Extensible Markup Language (XML) for the exchange of these messages transmitted under Internet protocols.

Section 1 - Introduces the OTA specifications, presenting generic messages for use across all industry verticals.

Section 2 - Car Rental Industry Specifications, describes the messages used for searching for availability with rates and booking a reservation from a car rental supplier.

Section 3 - Airline Industry Specifications presents the basic Booking message as based on the OTA architecture model for messaging.

Section 4- Travel Insurance Specifications, presents a messaging structure for use by the travel insurance industry to provide a rate quote and to purchase travel insurance as an adjunct to travel services.

Section 5- Hotel Industry Specifications, contains a revision of the Hospitality Industry Technology Integration Standards (HITIS) project Version 1.1 standards, which was sponsored by the American Hotel and Lodging Association, (AH&LA). The original HITIS standards that addressed these functions have been integrated into OTA messaging format, and provide an extensive set of messages to serve the hotel environment.

### **OpenTravel Alliance, Inc.**

333 John Carlyle Street, Suite 600

Alexandria, Va. 22314 USA

+1 703-548-7005

Fax +1 703-548-1264

[opentravel@disa.org](mailto:opentravel@disa.org)

<http://www.opentravel.org/>

Prepared in partnership with Data Interchange Standards Association (<http://www.disa.org>)

**January 31, 2002**

**OpenTravel™ Alliance, Inc.**

**License Agreement**

**Copyright, OpenTravel Alliance (OTA), 2001, 2002.**

**All rights reserved, subject to the User License set out below.**

***Authorization to Use Specifications and documentation***

**IMPORTANT:** The OpenTravel™ Alliance (“OTA”) Message Specifications (“Specifications”), whether in paper or electronic format, are made available subject to the terms stated below. Please read the following carefully as it constitutes a binding Agreement, based on mutual consideration, on you and your company as licensee (“You”).

- 1. Documentation.** OTA provides the Specifications for voluntary use by individuals, partnerships, companies, corporations, organizations, and other entities at their own risk. The Specifications and any OTA supplied supporting information, data, or software in whatever medium in connection with the Specifications are referred to collectively as the “Documentation.”
- 2. License Granted.**
  - 2.1.** OTA holds all rights, including copyright, in and to the Documentation. OTA grants to You this perpetual, non-exclusive license to use the Documentation, subject to the conditions stated below. All use by You of the Documentation is subject to this Agreement.
  - 2.2.** You may copy, download, and distribute the Documentation and may modify the Documentation solely to allow for implementation in particular contexts. You may bundle the Documentation with individual or proprietary software and/or sublicense it in such configurations.
  - 2.3.** You must reference, in a commercially reasonable location, the fact that the OTA Documentation is used in connection with any of your products or services, in part or in whole, whether modified or not, and You may include truthful and accurate statements about Your relationship with OTA or other use of the Documentation.
  - 2.4.** However, you may not change or modify the Specification itself, develop a new standard or specification from the Documentation, or state or imply that any works based on the Documentation are endorsed or approved by OTA.
  - 2.5.** You must include the OTA copyright notice in connection with any use of the Documentation. Any uses of the OTA name and trademarks are subject to the terms of this Agreement and to prior review and approval by OTA.
  - 2.6.** Nothing in this Agreement shall be interpreted as conferring on You or any other party any other interest in or right to the Documentation. Nothing in this Agreement shall be interpreted as in any way reducing or limiting OTA’s rights in the Documentation.
- 3. LIABILITY LIMITATIONS. THIS AGREEMENT IS SUBJECT TO THE FOLLOWING LIABILITY LIMITATIONS:**
  - 3.1.** ANY DOCUMENTATION PROVIDED PURSUANT TO THIS NON-EXCLUSIVE LICENSE AGREEMENT IS PROVIDED “AS IS” AND NEITHER OTA NOR ANY PERSON WHO HAS CONTRIBUTED TO THE CREATION, REVISION, OR MAINTENANCE OF THE DOCUMENTATION MAKES ANY REPRESENTATIONS OR WARRANTIES, EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

- 3.1. Neither OTA nor any person who has contributed to the creation, revision or maintenance of the documentation makes any representations or warranties, express or implied, that the use of the documentation or software will not infringe any third party copyright, patent, patent application, trademark, trademark application, or other right.
  - 3.2. Neither OTA nor any person who has contributed to the creation, revision, or maintenance of the documentation shall be liable for any direct, indirect, special or consequential damages or other liability arising from any use of the documentation or software. You agree not to file a lawsuit, make a claim, or take any other formal or informal action against OTA based upon Your acquisition, use, duplication, distribution, or exploitation of the Documentation.
  - 3.3. The foregoing liability limitations shall apply to and be for the benefit of OTA, any person who has contributed to the creation, revision or maintenance of the documentation, and any member of the board of directors, officer, employee, independent contractor, agent, partner, or joint venturer of OTA or such person.
4. **No Update Obligation.** Nothing in this Agreement shall be interpreted as requiring OTA to provide You with updates, revisions or information about any development or action affecting the Documentation.
5. **No Third Party Beneficiary Rights.** This Agreement shall not create any third party beneficiary rights.
6. **Application to Successors and Assignees.** This Agreement shall apply to the use of the Documentation by any successor or assignee of the Licensee.
7. **Term.** The term of this license is perpetual subject to Your compliance with the terms of this Agreement, but OTA may terminate this License Agreement immediately upon your breach of this Agreement and, upon such termination you will cease all use duplication, distribution, and/or exploitation of the Documentation in any manner.
8. **Interpretation and Choice of Forum.** The law of the Commonwealth of Virginia and any applicable Federal law shall govern this Agreement. Any disputes arising from or relating to this Agreement shall be resolved in the courts of the Commonwealth of Virginia, including Federal courts. You consent to the jurisdiction of such courts and agree not to assert before such courts any objection to proceeding in such forum.
9. **Acceptance.** Your acceptance of this License Agreement will be indicated by your affirmative acquisition, use, duplication, distribution, or other exploitation of the Documentation. If you do not agree to these terms, please cease all use of the Documentation now.
10. **Questions.** Questions about the Agreement should be directed to [www.opentravel.org](http://www.opentravel.org).

## Executive Summary

The OpenTravel Alliance 2001B Message Specification addresses the functionality of searching for availability and booking a reservation in the hotel and car rental, booking a reservation in the airline industry, as well as the purchase of travel insurance in conjunction with these services. OTA specifications use XML for structured data messages to be exchanged over the Internet or other means of transport.

This specification relies upon the work of other standards developer organizations; specifically, the ebXML initiative sponsored by UN/CEFACT and OASIS ([www.ebxml.org](http://www.ebxml.org)), and the work of the World Wide Web Consortium (W3C).

This specification also references standards developed by the International Organization of Standards (ISO), and the International Air Transport Association (IATA), that are used by the travel industry to provide standardized message structures and data for the travel industry. OTA realizes that the code dictionary was not present during this specification and is aware of its necessity and is making arrangements to provide this in further specifications.

This document contains five sections that introduce specifications for the travel industry verticals represented. For each respective industry vertical, this publication is the first of the business content specifications that OTA intends to develop.

**Section 1** - Introduction, describes the background for OTA messages and presents a generic cancel message structure and the Point-of-Sale object (POS) for use by all industry vertical working groups.

**Section 2** - Car Rental Industry Specifications, describes the messages used for searching for availability with rates and booking a reservation from a car rental supplier.

**Section 3** - Airline Industry Specifications present a Booking message based on the OTA architecture model for messaging.

**Section 4** - Travel Insurance Specifications, presents a messaging structure for use by the travel insurance industry to provide a rate quote and to purchase travel insurance as an adjunct to travel services.

**Section 5**- Hotel Industry Specifications, contains a revision of the Hospitality Industry Technology Integration Standards (HITIS) project Version 1.1 standards. The HITIS standards were developed under the sponsorship of the American Hotel and Lodging Association, (AH&LA).

The original HITIS standards provided an extensive set of messages to serve the hotel environment. Their functionality has been integrated into OTA messaging format, resulting in OTA Version 1 messages that can be exchanged between a larger audience than hotels alone, allowing for interoperability of well structured XML messages with other industry verticals.

The spectrum of functions addressed includes searching for a hotel property, determining availability, booking a reservation, recording the hotel stay information and statistical analysis, payment of commissions, and publication of rates and static hotel property data. In some cases, such as statistical analysis or payment of agent commissions, these specifications may be applicable to other travel service providers throughout the industry.

## **OTA Infrastructure**

The OTA defined the messaging components and a method for securely and reliably exchanging messages between trading partners in OTA 2001A Infrastructure. That publication defines a structure for common elements and attributes, the syntax of action verbs at the root tag level, and conventions found in all OTA messages.

OTA transactions use a request and response model that provides a matched pair of messages to help account for and manage the data flow. OTA messages use the message structure from the ebXML Header envelope that effectively separates the routing and packaging information from the payload content of the message. This provides for the authentication of parties, security, confidentiality, and integrity of message handling separate from the business content of each message. The Version 2 OTA Infrastructure was a 'snapshot in time' based on a version of the ebXML Messaging Specification available at the time of the publication deadline for OTA 2001A per the HITIS /OTA agreement.

*OTA recognizes that another change in infrastructure will be necessary; and that change is anticipated to take place based upon the completion of the ebXML specification in May 2001. The 2001C OTA Specification makes direct mapping of OTA message sets with ebXML 1.0 and will be ready for public review within the fourth quarter of the 2001 calendar year.*

The OTA 2001A publication defined an infrastructure that separated the messaging component (Header and Manifest) from the payload documents. The infrastructure defined in the OTA 2001A Publication allowed working groups to concentrate in the interim time on payload messages specific to their business needs, and to develop the business content independent of the way in which they exchange messages or connect to trading partners.

Parties exchanging messages are able to validate messages using XML Schema. Logical groups of data elements are represented by schema fragments that provide formal specification of semantics and data typing that allows the schema fragments to be combined into message structures.

A separate schema specifies basic error conditions and administrative messages independent of OTA's specification versions. This definition establishes guidelines for the development of OTA's message sets and for the exchange of OTA messages in the travel industry.

## **Brief Perspective on the Hospitality Industry Technology Integration Standards (HITIS)<sup>1</sup>**

The earliest systems developed for hotels addressed functions for the Property Management Systems. Their primary objective was to replace the laborious task of manually posting all charges to guest accounts. Before the advent of the computerized PMS, the process of managing the guest in a hotel was done by hand. When a guest checked in to a room, the front desk clerk would create a blank ledger sheet (folio). The folio would be placed in a file (bucket) by room number. Accounts to which a posting was permitted, but were not guest folios, were placed in the front or the back of the bucket. These were known as "city ledger" accounts, because typically they were set up for local business to which direct billing was permitted.

Each transaction charged to the guest's room was posted individually to the correct folio. The folio was extracted from the section in the bucket, and placed in the posting machine. The previous balance, the category of the transaction, and the transaction amount were then entered. The posting machine printed the transaction, adding the amount to the total, and printed the new

---

<sup>1</sup> Based on the original whitepaper by William T. Geoghegan.

total on the folio. It also accumulated the category totals, which were then printed at the end of the day. When it came time to post the nightly room and tax charges, the folio for each room was pulled, updated, and returned. The night audit function consisted of ensuring that the total of transactions plus a previous control total would all add up to the correct amount.

The computer changed all this. The need to individually post the nightly charges was eliminated, as the process could be done automatically. Control totals were easily accomplished, and the task of the night auditor was no longer an all night task.

Almost all of the earliest systems had the ability to receive and transmit serial data. The interfaces developed for the hospitality industry converted the output of a serial device to transfer data. A translation could be accomplished both into and out of the computer using the standard ASCII (American Standard Code for Information Interchange) character set as the de facto standard for communication. While ASCII standards existed for transmissions, not all vendors implemented them completely or in the same way. Some vendors believed that they could do better, and eventually, each vendor created its own specifications, and the PMS vendors had no choice but to program to each individually.

Today, the world is changing rapidly. Serial ports are no longer the preferred method of communicating between computers and networks have become the de facto standard for communication within a hotel property. This has been quickly followed by the Internet and communication links to the world outside of the local area network, to include the hotel chain reservations systems and global distribution systems.

At any given time, dozens of vendors proliferate systems to the hospitality industry. In addition, many new services, such as Internet services, travel services, in-room fax services, are being introduced into the hospitality market with the intent to provide additional guest services, and all require interfaces to fully automate these services. To a great extent, however, existing legacy connections still continue to use serial interfaces in the hospitality industry, and the amount of effort required by a vendor to write a new interface has caused many a new vendor to be shut out of the market.

But, despite the method of connectivity, it is still necessary for two parties to agree on the data elements to be exchanged when they attempt to write an interface. One thing that has become evident is that there are only a finite number of data elements within the hotel environment. Unfortunately, there were no standards within the industry as to these elements. For example, something as simple as a guest name could be represented in many different ways. You could have separate elements for first, last, title, etc. or you could have a single field populated with last, title, first. If you include the potential problems with international names, the problem becomes quite complex. If you consider the data elements necessary to represent the address, the problem becomes a monumental project.

Thus, any standard needs to ensure that data will be consistently represented between systems. Without standards, systems have to go through the laborious process of converting proprietary data to communicate with other systems. Vendors who create new releases of their software which adhere to standards will likely find their costs of maintaining their interface library significantly reduced, and their product more widely accepted by the hotel customers.

In an attempt to address these issues, the American Hotel and Lodging Association initiated the Hospitality Industry Technology Integration Standards (HITIS) project. The original technical specifications were written and submitted to HITIS by the Windows Hospitality Interface

Specification (WHIS) organization, under the sponsorship of Microsoft Corporation. The goal of these specifications was to identify the functions served by the interfaces in the hospitality industry and to standardize the implementation of these functions.

In addition to the identification of the functions addressed by interfaces, it was important for a common data dictionary to be developed. This would allow all systems to use the same structures for things like room numbers, names, addresses, etc. All standards would then include only those data elements that have been identified as part of the data dictionary, and all vendors could rely on those data elements being supported by compliant systems.

The HITIS originally created object based standards using the Unified Modeling Language (UML), which provides a platform and programming language independent way of designing system components and is well adapted for the definition of the interfaces. The HITIS interfaces were intended to work in all computer environments, giving the hotel far more flexibility in the selection of guest service and data collection devices.

It was recognized, however, that the HITIS standards, platform neutral in concept, would require two initial platform mappings for implementation; one to the Windows operating system and one to the Java language platform. With the WHIS organization already in place, Sun Microsystems, Inc. and IBM formed the Java for HITIS Implementation Solutions (JHIS) working group to provide assistance in the Java language mapping of the HITIS interface standards.

Fortunately, technology never stands still. Without minimizing the impact that common data definitions and functional specifications will have on legacy systems, it was the goal of HITIS to create standards to be used with future technologies. The arrival of the eXtensible Markup Language (XML), provided a third platform mapping extension for implementation of the HITIS interface standards. After extensive consideration of mapping the standards to the two separate platforms, Microsoft and Sun Microsystems recommended to AH&LA in 1999 that XML be pursued as the primary platform mapping for implementation.

The use of XML will enable rapid adoption and implementation of the standards as well as secure the interoperability between all systems in the hospitality industry. In addition, XML broadens the capabilities of the HITIS Central Reservation Systems (CRS) standards to interconnect with other facets of a travel reservation, such as airline, rental car or tour packages associated with a hotel stay.

HITIS represents a significant contribution to the future of Information Technology within the Hospitality Industry, and the overwhelming benefits to the industry are clear. PMS and Central Reservation Systems (CRS) vendors will be able to concentrate on the capabilities of their core software. Hotels will no longer be locked into the finite list of interfaces from their properties, and new vendors will be able to create systems that can be much more readily accepted by their potential customers.

On June 22, 2000, the AH&LA and the OpenTravel Alliance reached an agreement to develop and maintain specifications technically relevant to their industries. That agreement, finalized by October, 2000, placed the responsibility for further development and revision of the Central Reservation System (CRS) standards on OTA. HITIS continues to maintain standards for the hotel industry not involved in the CRS environment.

These specifications represent the revision of the initial mapping of the HITIS 1.1 standards into XML, incorporating them into the OTA infrastructure, and offering them to the members of the OpenTravel Alliance through the OTA Hotel Working Group for review.



## **Acknowledgments**

This publication is the culmination of numerous efforts by the working groups, individuals and consultants who contributed their time and knowledge to the OTA. I would like to acknowledge the following participants for their outstanding efforts:

As OTA began the development of this specification, a Data Content team was formed to identify the basic data elements to be used in the message sets. Paula Heilig, Worldspan, bravely served as commander-in-chief of that team from its early beginnings in mid-2000 to the fruition of its efforts in this specification. Appreciation is expressed to Brian Smith, CSA Travel Protection, who painstakingly entered all the data elements from every industry vertical into a spreadsheet to provide a basis for comparison, and to Jim deBettencourt, McCord Travel Management, for providing leadership and cohesion between different industry working groups on the team.

Many thanks to John Cross, David Joslin, and Jagdeep Sharma of Perot Systems for providing consultant support to the Car Working Group and for writing the Vehicle Availability with Rate specification. Special thanks go to Melvyn Turner, GetThere, Inc., for filling in the gap at the other end of the spectrum by writing the Vehicle Reservations specification.

Brendan Sherlock, consultant to the Air Working group, wrote the first Data Dictionary for the Air Working Group, survived the experience to write the Air Availability specification, then retired to live in the Cayman Islands. Thanks go to Alan Kotok, former OTA Standards Manager, for recording the requirements of the AWG and writing the original DTDs for the Air Booking specification.

Appreciation is expressed to Sue Domenick, United Airlines, David Brennan, Northwest Airlines, and Victoria Day, Air Transport Association, for hosting Air Working Group meetings, and to Steve Fohl, American Airlines, for hosting the Interoperability Committee, at their respective locations.

Special thanks to Michael Schreiber, Insuractive, who single-handedly took over the development of specifications for the travel insurance industry during the final days of preparation of this specification.

In the Hotel Working Group, thanks go Bill Geoghegan, Consultant, for contributing his vast knowledge of the hotel industry, bringing forward his long history with the HITIS project, and for writing the Hotel Reservation and a few other specifications published in this document. Thanks to Glenda Brady, Cendant Corporation, Lisa Fues, Marriott Corporation, and Jennifer Weisenberger, Hyatt Hotels, who, in conjunction with Bill Geoghegan, accomplished the Herculean task of organizing well over 1,000 data elements into a spreadsheet that forms the basis for a code list of Hotel Descriptive Content. Thanks also goes to Mark Hoare for his diligent analysis of the HITIS standards that made us reach a step beyond the obvious in revising the HITIS Version 1.1 standards.

Thanks goes to Jim Martin, Cendant Corporation, for writing prototypes for the POS object and generic Cancel message. Jim, along with Chris Heinen, Best Western Intl., assisted in the early conversion of numerous HITIS messages into OTA format. Thanks also to Chris Heinen and Mike Kistner, for hosting the OTA Hotel Working Group meeting in Phoenix, AZ. It was nice to be there during the month of February.

OTA appreciates the assistance of Marcus Keyser, Carl Leander, and Bill Duncan, Daylight Software, and Dave Foss, PlanSoft, for writing the meeting planning messages that became a part of the Hotel Property Data sub-committee work.

Appreciation is expressed to Sally Kelly, KPMG, for dedicating the services of KPMG consultants, and to John Petit, KPMG, for participation in the Hotel Property Data effort and sharing his knowledge of the ebXML specifications and XML schema that guided neophytes into schema literacy.

Many thanks goes to Tony Williams, Travel Technology Initiative, Ltd., for representing the Leisure Working Group on the Data Content team and for dutifully writing the Package Holiday Bookings specification despite the limitations of working long-distance across the Atlantic Ocean.

Thanks go to OTA Architecture Chair, Michael Townsie, Galileo, and Co-Chair, Ron Kleinman, Sun Microsystems, along with members of the Architecture team, for overseeing the architectural hurdles as these specifications began to take shape.

Finally, acknowledgements are due to the Chair(s) of each industry working group for keeping track of the progress of the specifications for their working group members, (and for keeping their members on track). These leaders are:

- ◆ Steve Fohl, Air Working Group
- ◆ Glenda Brady and Jennifer Weisenberger (past and current Chairs), Hotel Working Group
- ◆ Kim Lambert, Car Working Group
- ◆ David Trigg and Mick Mott (past and current Chairs), Leisure Working Group
- ◆ Nick Lanyon and Steve Reynolds (past and current Chairs), Non-Supplier Working Group

Many members attended meetings and conference calls, participating in OTA's efforts over the past year. Thank you and this document could not have been developed without your help.

## OTA Code of Conduct

The OTA Code of Conduct governs the OpenTravel Alliance. A reading or reference to the OTA Code of Conduct begins every OTA activity, whether a meeting of the OTA Board of Directors or conference call to resolve a technical issue. The OTA Code of Conduct says:

Trade associations are perfectly lawful organizations. However, since a trade association is, by definition, an organization of competitors, OpenTravel Alliance (OTA) members must take precautions to ensure that we do not engage in activities which can be interpreted as violating anti-trust or other unfair competition laws.

For any activity which is deemed to unreasonably restrain trade, OTA, its members and individual representatives may be subject to severe legal penalties, regardless of our otherwise beneficial objectives. It is important to realize, therefore, that an action that may seem to make "good business sense" can injure competition and therefore be prohibited under the antitrust or unfair competition laws.

To ensure that we conduct all meetings and gatherings in strict compliance with any such laws and agreements in any part of the world, the OTA Code of Conduct is to be distributed and/or read aloud at all such gatherings.

- There shall be no discussion of rates, fares, surcharges, conditions, terms or prices of services, allocating or sharing of customers, or refusing to deal with a particular supplier or class of suppliers. Neither serious nor flippant remarks about such subjects will be permitted.
- OTA shall not issue recommendations about any of the above subjects or distribute to its members any publication concerning such matters. No discussions that directly or indirectly fix purchase or selling prices may take place.
- There shall be no discussions of members' marketing, pricing or service plans.
- All OTA related meetings shall be conducted in accordance with a previously prepared and distributed agenda.
- If you are uncomfortable about the direction that you believe a discussion is heading, you should say so promptly.

Members may have varying views about issues that OTA deals with. They are encouraged to express themselves in OTA activities. However, official OTA communications to the public are the sole responsibility of the OTA Board of Directors. To avoid creating confusion among the public, therefore, the Board must approve press releases and any other forms of official OTA communications to the public before they are released



## Table of Contents

License Agreement.....	ii
Executive Summary.....	iv
OTA Infrastructure .....	v
Brief Perspective on the Hospitality Industry Technology Integration Standards (HITIS) .....	v
Acknowledgments .....	ix
OTA Code of Conduct.....	xi
Section 1 - OTA Generic Messages .....	17
1.0 Introduction.....	18
1.1 Use of XML.....	19
1.2 Status of the Travel Industry.....	19
1.3 Design Goals of OTA.....	20
1.4 OTA's Approach .....	20
1.5 Specification Components .....	21
1.6 Future goals for OTA .....	21
1.7 OTA Versioning.....	22
1.8 The Journey Concept - 2001 and Beyond.....	25
1.9 OTA Generic Cancel Request.....	31
1.10 Point of Sale (POS) .....	36
Section 2 - Car Rental Specifications .....	41
2.0 Introduction.....	42
2.1 Vehicle Availability Rates Messages .....	45
2.2 Vehicle Availability Rates Request Message (OTA_VehAvailRateRQ).....	46
2.3 Vehicle Availability Rates Response Message ( OTA_VehAvailRateRS ) .....	54
2.4 XML Schema Diagrams for Vehicle Availability Rates Request.....	62
2.5 Sample XML Message for Vehicle Availability Rates Request .....	65
2.6 XML Schema Diagrams for Vehicle Availability Rates Response.....	67
2.7 Sample XML Message for Vehicle Availability Rates Response.....	72
2.8 VehAvailRateRQ/RS Revision History.....	75
2.9 Vehicle Reservation Messages.....	78
2.10 Vehicle Reservation Request Message (OTA_VehResRQ) .....	79
2.11 Vehicle Reservation Response Message ( OTA_VehResRS ) .....	87
2.12 XML Schema Diagrams for Vehicle Reservation Request .....	98
2.13 Sample XML Message for Vehicle Reservation Request.....	100
2.14 XML Schema Diagrams for Vehicle Reservation Response .....	102
2.16 VehResRQ/RS Revision History .....	110
Section 3 - Airline Booking Specifications.....	112
3.0 Introduction.....	113
3.1 Limited Scope .....	113
3.2 Point of Sale information.....	113
3.3 POS elements .....	113
3.4 GDS attributes.....	114
3.5 Additional Airline requirements .....	114
3.6 Receive From element.....	<b>Error! Bookmark not defined.</b>
3.7 Air Booking Request message .....	114
3.8.Air Booking Response message.....	119
Section 4 - Travel Insurance Specifications.....	125
4.0 Introduction.....	126

4.1 Insurance Quote Request.....	127
4.2 Insurance Quote Response.....	135
4.3 Insurance Book Request.....	136
4.4 Insurance Book Response.....	136
4.5 Insurance Sample XML Instance Documents.....	138
Section 5 - HITIS Integration - Hotel Industry Messages .....	142
5.1 Synopsis of HITIS Standards.....	143
5.2 Hotel Search Request (HITIS Hotel Search Query).....	147
5.3 Hotel Search Messages.....	149
5.4 XML Sample Messages - Hotel Search Request .....	153
5.5 Hotel Search Response Message.....	156
5.6 Sample XML Messages - Hotel Search Response .....	157
5.7 Hotel Availability Request (HITIS Availability Query).....	161
5.8 Hotel Availability Messages.....	165
5.9 Sample XML Message .....	170
5.10 Hotel Availability Response Message.....	172
5.11 Hotel Reservation Request Messages.....	179
5.12 Hotel Reservation.....	181
5.13 Sample XML Message using the OTA_HotelResRQ XML Schema.....	182
5.14 Hotel Reservation Response Message (OTA_HotelResRS).....	184
5.15 Sample XML Message .....	185
5.16 Hotel Reservation Notification Messages.....	191
5.17 Hotel Reservation Notification Request Message (OTA_HotelResNotifRQ).....	192
5.18 Sample XML Message using the OTA_HotelResNotifRQ XML Schema.....	193
5.19 Hotel Reservation Notification Response Message (OTA_HotelResNotifRS) .....	197
5.20 Sample XML Message .....	198
5.21 XML Schema Diagrams for Reservation Notification Request.....	198
5.22 XML Schema Diagrams for Reservation Notification Response.....	198
5.23 Revision Log for OTA_ResNofitRQ/RS.....	199
5.24 Hotel Reservation Fragment .....	199
5.25 Hotel Reservation Schema Diagrams.....	212
5.26 Get Message Messages.....	217
5.27 Get Message Request Message (OTA_GetMsgRQ) .....	218
5.28 Sample XML Message using OTA_GetMsgRQ XML Schema. ....	218
5.29 Get Message Response Message (OTA_GetMsgRS) .....	219
5.30 Sample XML Message using the OTA_GetMsgRS XML Schema .....	219
5.31 XML Schema Diagram for Get Message Request .....	220
5.32 XML Schema Diagram for Get Message Response.....	220
5.33 Revision History for OTA_GetMsgRQ/RS.....	220
5.34 Get Message Information Messages.....	223
5.35 Get Message Information Request Message (OTA_GetMsgInfoRQ).....	224
5.36 Get Message Information Response Message (OTA_GetMsgRS) .....	224
5.37 XML Schema Diagram for Get Message Information Request.....	225
5.38 XML Schema Diagram for Get Message Information Response.....	225
5.39 Revision History for OTA_GetMsgInfoRQ/RS.....	225
5.40 Commission Messages .....	229
5.41 Commission Notification Request Message (OTA_CommNotifRQ) .....	230
5.42 Commission Notification Response Message (OTA_CommNotifRS) .....	233
5.43 XML Schema Diagram for Commissions Notification Request.....	234
5.44 XML Schema Diagram for Commissions Notification Response .....	234

5.45 Revision History for OTA_CommNotifRQ/RS .....	234
5.46 Stay Information Notification Messages .....	237
5.47 Stay Information Notification Request Message (OTA_StayInfoNotifRQ).....	238
5.48 Stay Information Notification Response Message (OTA_StayInfoNotifRS).....	240
5.49 XML Schema Diagram for Stay Information Notification Request.....	241
5.50 XML Schema Diagram for Stay Information Notification Response .....	241
5.51 Revision History for OTA_StayInfoRQ/RS .....	241
5.52 Statistics Messages.....	244
5.53 Statistics Notification Request Message (OTA_StatsNotifRQ).....	245
5.54 Sample XML Message .....	246
5.55 Statistics Notification Response Message (OTA_StatsNotifRS).....	247
5.56 Sample XML Message .....	247
5.57 Statistics Request Message (OTA_StatsRQ).....	248
5.58 Statistics Response Message (OTA_StatsRS) .....	248
5.59 XML Schema Diagram for Statistics Notification Request.....	251
5.61 XML Schema Diagram for Statistics Request .....	251
5.62 XML Schema Diagram for Statistics Response .....	251
5.63 Revision History for OTA_StatsRQ/RS.....	251
5.64 Meeting Profile Messages.....	253
5.65 MeetingProfile .....	255
5.66 Sample XML Fragment to Create a Meeting Profile Request .....	261
5.67 Revision History for Meeting Profile .....	262
Section 5a - HITIS Availability, Rate and Inventory .....	263
Hotel Availability, Rate and Inventory Messages .....	264
Scope of ARI Messages .....	264
Business Use of ARI Messages .....	265
Availability Rate and Inventory Notification Messages .....	265
Notification Response Messages .....	266
Standard Module for Versioned Message Responses .....	266
Data Elements in Response Messages .....	266
5.68 Hotel Availability Notification .....	269
5.69 OTA_HotelAvailNotifRQ Sample Message:.....	270
5.70 Hotel Booking Rule Notification .....	270
5.71 Hotel Booking Rule Notification Messages.....	274
5.72 OTA_HotelBookingRuleNotifRQ Sample Message:.....	274
5.73 Booking Rule Notification Response .....	274
5.74 Hotel Inventory Count Notification.....	275
5.75 Hotel Inventory Count Notification Messages.....	276
5.76 Off Sell Inventory Messages.....	276
5.77 OTA_HotelInvCountNotifRQ Sample Message.....	276
5.78 Hotel Summary Notification.....	276
5.79 Hotel Summary Notification Messages .....	277
5.80 OTA_HotelSummaryNotifRQ Sample Messages:.....	277
5.81 Hotel Rate Amount Notification .....	278
5.82 Hotel Rate Amount Notification Messages .....	279
5.83 OTA_HotelRateAmountNotifRQ Sample Message.....	279
5.84 Hotel Inventory Notification.....	279
5.85 Hotel Inventory Notification Messages .....	280
5.86 OTA_HotelInvNotifRS Sample Message:.....	<b>Error! Bookmark not defined.</b>
5.87 OTA_HotelInvNotifRQ Sample Message: .....	281

5.88 Hotel Rate Plan Notification .....	281
5.89 Hotel Rate Plan Notification Messages .....	282
5.90 OTA_HotelRatePlanNotifRQ Sample Message: .....	282
5.91 Hotel Rate Plan Notification Response .....	282
5.92 OTA_HotelRatePlanNotifRS Sample Message: .....	<b>Error! Bookmark not defined.</b>
5.93 Hotel Inventory Block Notification .....	284
5.94 Hotel Inventory Block Notification Messages .....	285
5.95 Hotel Inventory Block Notification Response .....	285
5.96 OTA_HotelInvBlockNotifRS Sample Message: .....	<b>Error! Bookmark not defined.</b>
5.97 Hotel Descriptive Content Notification .....	286
5.98 Hotel Descriptive Content Notification Messages .....	287
5.99 Hotel Descriptive Content Notification Response .....	287
5.100 OTA_HotelDescriptiveContentNotifRS Sample Message .....	<b>Error! Bookmark not defined.</b>



# **OpenTravel Alliance Message Specification**

**Publication 2001B**

## **Section 1 - OTA Generic Messages**

## 1.0 Introduction<sup>2</sup>

Created in May of 1999, the OpenTravel Alliance (OTA) is a legally incorporated, non-profit consortium consisting of approximately 150 member companies from all sectors of the travel industry. These sectors include typical travel industry supplier companies from the airline, car rental, hotel and leisure travel verticals, in addition to related companies that provide distribution and technology to support the industry.

The OTA consists of the five following working groups:

- Air Working Group;
- Car Working Group;
- Hotel Working Group;
- Leisure Working Group, and
- Non-Supplier Working Group

Each of the industry travel supplier segments has its own business work group, tasked with the development of technical specifications for its particular vertical, while the non-supplier group works with each of the supplier working groups according to the technology services that their company provides.

OTA's efforts over the past two years have led to the successful organization of the work groups and, more importantly, immediate development of specifications for customer and company profiles, and the all-important functions of requesting the availability of travel products and obtaining those products by booking a reservation.

An Interoperability Committee reconciles the specifications from all industry segments to provide a consistent approach to both customer and content throughout the travel business. The Interoperability Committee is comprised of elected members from each of the five industry work groups, and is chartered to serve as the technical steering group for OTA.

From time to time, the Interoperability Committee commissions cross-industry teams to address the needs of all the working groups. Over the past two years, a Profile Content team was created to develop the original OTA Version 1 Customer Profile, and the Profile Integration team worked to integrate the HITIS and OTA profiles.

Currently, two ongoing teams are in existence:

- Architecture team
- Data Content team

In the creation of these cross-industry teams, OTA realized that there was considerable work to do involving infrastructures and specific vocabularies within each travel industry segment.

---

<sup>2</sup> Thanks go to David Blackistone and Diane Mair, KPMG, for their contributions to this section.

## 1.1 Use of XML

The OTA has chosen Extensible Markup Language (XML) as the vehicle for its specifications, determining that XML is the best means for meeting its goals. XML offers a common framework for creating Internet-compatible messages, but is flexible enough to be used between systems elsewhere in the travel industry.

The use of XML will assist in producing industry-wide specifications that will exploit the opportunities available with the advent of universal access to the low-cost, fast, communications infrastructure that has arrived with the Internet. The successful implementation of these new specifications will result in the facilitation of customer and trip-related information amongst all industry trading partners, regardless of how “net savvy” the participant.

The following are several aspects of XML that made it the preferred language in designing a specification for OTA’s needs:

- XML allows for the exchange of structured data over the web.
- XML messages can include action elements that instruct receiving systems to undertake certain operations.
- XML allows for the definition of data elements meaningful to trading partners or entire industries.
- XML is a powerful tool for reshaping the way information is transmitted and presented over the Internet.

The result of OTA’s stated goals is specifications created on an open platform that will allow trading partners to access standardized information.

## 1.2 Status of the Travel Industry

The travel industry has been a leader in using global electronic distribution of information for decades, but it has developed this technology in a manner that has resulted in several incompatible standards between the different industry members and their lines of business. What has evolved is a series of related, sometimes interconnected, segments trying to provide the user with as much information as possible. Incompatible specifications have made it difficult for various participants in the travel industry to share information and has limited the variety of distribution channels available to travel suppliers. These limitations make it difficult for suppliers to differentiate themselves, reach new customers, create innovative offers or exchange information with their corporate customers.

Customers are also restricted by this situation and are not able to make a completely informed choice in comparing services from different sources. Currently, customers must navigate a complex web of interrelationships to obtain all the information they might desire. Most travelers elect not to do this.

Information is the core of any travel transaction, be it with the airlines, car rental companies, cruise lines, railroads, hotels, or tour operators. The absence of a common standard has fostered distributions through single channels that silo both customer and market data. While the industry has tried to exploit the Internet, it cannot fully do so until a universal standard has been established to exchange information.

### 1.3 Design Goals of OTA

OTA's basic goal is to design industry specifications capable of exploiting the communications systems that are available with Internet connectivity. To achieve these goals, OTA has designed specifications to meet the following criteria:

- *Openness*- OTA specifications are publicly available to all organizations seeking to develop new or enhanced systems. Membership to OTA is open to all organizations interested in developing these specifications.
- *Flexibility*- OTA specifications provide travel service providers with the flexibility they need to develop, test and deploy new services. The specifications outline the minimum necessary functionality to provide reliable interactions between customers and the systems owned and maintained by the companies serving them.
- *Platform Independence*- OTA has developed these specifications to work with any equipment or software that can support the common standards used in the specifications.
- *Security*- OTA places great importance on the need to protect information from unauthorized access and the need to give the customer control over the creation, update and exchange of data with other parties.
- *Extensibility*- OTA plans to add more services and functionality to this specification in a way that minimizes incompatibility for those implementing this or other early versions. OTA work groups that develop the specifications do so with future transitions in mind.
- *International scope*- The initial specification was written in English; however, OTA intends to extend later versions to provide representation in character sets supporting the Unicode standard. When possible, OTA has designed data elements to meet as many global elements as possible.

### 1.4 OTA's Approach

Industry fragmentation has led to little terminology and technology being universal throughout the travel industry. The specifications developed by OTA remove the constraints imposed by the trade-offs required when computer communications and processing were costly. These specifications represent a different approach to exchanging data elements and messages, and intend to provide each data element with a unique identity or tag within each industry sub-segment.

OTA recognizes the challenges in developing specifications and is proactively trying to address this by the belief that the underlying technology has been designed to accommodate changes that will occur as the travel industry evolves, and it is OTA's hope that worldwide availability of OTA's specifications will spur development of new interchanges between partners in the travel industry.

OTA uses the term *specification* rather than *standard*. OTA reserves the word standard for documents such as those from International Organization of Standards (French acronym ISO) or American National Standards Institute (ANSI). This document, therefore, is called a specification, which takes broad standards such as XML and applies them to the travel industry. As a specification, OTA documents focus on travel industry requirements and have been developed in an open consensus process.

## 1.5 Specification Components

OTA's publications consist of several components that should enable companies to take advantage of current tools and methods for implementation.

This set of OTA specifications includes:

- *Specification Document*- MS Word or .pdf file that contains the traditional specification information; along with introduction and explanations of the business content.
- *XML Schema and Schema Fragments* – All OTA XML Schema files are in conformance with the World-Wide Web Consortium (W3C) Recommendation XML Schema 1.0.\*
- *XML Instance Documents* – Included in the 2001B Specification are sample XML instance files that have been validated to OTA XML Schema files.

\* While the text of the Specification document makes every attempt to accurately reflect the XML Schema, in the case of a conflict, the XML Schema takes precedence.

## 1.6 Future goals for OTA

The ever-evolving travel industry will require OTA to provide for the continued growth in the capabilities and functions for future specifications. OTA's goals are to provide specifications specifically written to address the needs of the industry sub-segments in a manner that is unique within the OTA name-space. Future publications and specifications will focus on the vocabulary and the messaging needs of each industry segment designed by the industry work groups.

In 2001C, OTA will include other message sets for example, Air Availability, Golf Reservations, and the Package Holiday Specification. The Package Holiday Specification contains a set of messages to determine availability and book reservations for a package holiday. The products included in package holidays consist of any number of itinerary elements, such as transport, accommodation, car rental, extra products or services, special services, extras etc., packaged together and sold as a single unit.

Additionally, 2001C will contain both the ebXML mapping to the OTA 2001A Infrastructure and a Best Practices document. Spearheading the adoption of OTA Best Practices, the Package Holiday Tour Specifications will represent this initial application, in lieu of its original preparation for release in 2001B.

The HITIS Integration portion of this specification, once it has gone through its OTA's review process and is recommended for adoption and broad implementation by the travel industry, may be submitted as an ANSI standard as a result of an Agreement with the American Hotel & Lodging Association (AH&LA), signed October 18, 2000. This agreement allows the AH&LA, an ANSI-accredited standards developer organization, to submit a specification for approval as an ANSI standard. That process, monitored by the American National Standards Institute, involves the publication of an intended standard for a period of commentary and consists of a canvass of those parties who would be materially affected by the standard.

## 1.7 OTA Versioning

To provide for both implementation stability and managed growth, the OTA has adopted a scheme for versioning of messages. Infrastructure changes, such as the addition of Header elements or changes in the method used to connect to trading partners and initiate authentication or security-related functions will constitute an Infrastructure version change. For the 2001B Specification, OTA assumes the service interface as defined in Chapter 4, Message Structure, of the OTA 2001A Specification. That infrastructure is anticipated to change, potentially adopting the Messaging Specification, Version 1.0, of the ebXML Transport, Routing and Packaging working group.

OTA payload messages are intended to communicate the business functionality. Changes to the schema or the addition of business content elements to the schema constitute a change in the version of the payload message.

A payload document is identified in the Document Reference of the Manifest element (OTA 2001A Infrastructure). The Version attribute in the Schema element indicates the version of message in the payload. The identification of metadata about the version at this level allows a trading partner to immediately identify whether its system can accept the document(s) and process them appropriately, eliminating the need to parse the entire document in order to return an error message about version compatibility.

In order to isolate and contain the scope of change with the addition of new business content and messages, when OTA releases a specification, each new message introduced will be versioned starting with the number 1, and incremented by (n+1) with each subsequent revision. This places the level of versioning at the message interaction, or at the level of the action verb that is the root of the payload document.

OTA will identify publications by the year of publication, and sequential publications within a calendar year will be further identified by the letter A, B, C, or D, etc. For example, the first specification published in the year 2001 was 2001A, the second publication 2001B, etc.

- The OTA 2001A Specification document included the OTA Infrastructure and the integrated HITIS / OTA Profile.
- This publication includes Version 1 messages for all working groups represented in this document.

With each publication, an updated list of valid enumerations supported for the *Type* attribute (strings that match the root tag of the payload and identify the messages) will be included.

### 1.7.1 Versions Supported

The Version numbers of the messages supported by this publication<sup>3</sup> are the following:

### 1.7.2 Infrastructure Version

The Version 2 OTA Infrastructure is the current version of infrastructure used for the exchange of OTA messages. OTA Infrastructure, Version 2, is documented in publication 2001A.

---

<sup>3</sup> This list includes existing messages, new messages, and revisions of existing messages.

As of the date of this publication, OTA payload messages in this document are presumed to use OTA Version 2 Infrastructure.

***Note:** A change in infrastructure is anticipated based upon the completion of the ebXML Messaging Specification in early May, 2001.*

### **1.7.3 UniqueId Types Supported**

Two string values are supported as an enumeration of the *Type=* attribute of the IniqueId.

- *Profile* - Identifies a customer or company profile by UniqueId
- *Reservation* - Identifies a reservation by UniqueId

### **1.7.4 Version of Payload Messages**

<b>Generic Infrastructure Messages</b>
<b>Existing Messages:</b>
OTA_CreateProfileRQ - Version 2
OTA_CreateProfileRS - Version 2
OTA_ReadRQ - Version 2
OTA_ReadProfileRS - Version 2
OTA_UpdateRQ - Version 2
OTA_UpdateRS - Version 2
OTA_DeleteRQ - Version 2
OTA_DeleteRS - Version 2
<b>New Messages in this publication:</b>
OTA_CancelRQ - Version 1
OTA_CancelRS - Version 1

<b>OTA Payload Messages</b>
<b>Car Rental Industry Messages:</b>
OTA_VehAvailRateRQ - Version 1
OTA_VehAvailRateRS - Version 1
OTA_VehResRQ - Version 1
OTA_VehResRS - Version 1
<b>Airline Industry Messages:</b>
OTA_AirBookRQ- Version 1
OTA_AirBookRS- Version 1
<b>Hotel Industry Messages (HITIS Integration):</b>
OTA_HotelSearchRQ- Version 1
OTA_HotelSearchRS- Version 1
OTA_HotelAvailRQ- Version 1
OTA_HotelAvailRS- Version 1
OTA_HotelResRQ- Version 1
OTA_HotelResRS- Version 1
OTA_HotelResNotifRQ- Version 1
OTA_HotelResNotifRS- Version 1
OTA_GetMsgRQ- Version 1
OTA_GetMsgRS- Version 1
OTA_GetMsgInfoRQ- Version 1

OTA_GetMsgInfoRS- Version 1
OTA_CommNotifRQ- Version 1
OTA_CommNotifRS- Version 1
OTA_StayInfoRQ- Version 1
OTA_StayInfoRS- Version 1
OTA_StatsRQ- Version 1
OTA_StatsRS- Version 1
OTA_StatsNotifRQ- Version 1
OTA_StatsNotifRS- Version 1
OTA_CreateMeetingProfileRQ
OTA_CreateMeetingProfileRS
OTA_AvailNotifRQ- Version 1
OTA_AvailNotifRS- Version 1
OTA_RateAmountNotifRQ- Version 1
OTA_RateAmountNotifRS- Version 1
OTA_RatePlanNotifRQ- Version 1
OTA_RatePlanNotifRS- Version 1
OTA_HotelSummaryNotifRQ- Version 1
OTA_HotelSummaryNotifRS- Version 1
OTA_InvCountNotifRQ- Version 1
OTA_InvCountNotifRS- Version 1
OTA_BookingRuleNotifRQ- Version 1
OTA_BookingRuleNotifRS- Version 1
OTA_InvBlockNotifRQ- Version 1
OTA_InvBlockNotifRS- Version 1
OTA_InvNotifRQ- Version 1
OTA_InvNotifRS- Version 1
OTA_HotelDescriptiveContentNotifRQ- v 1
OTA_HotelDescriptiveContentNotifRS- v 1
<b>Travel Insurance Industry Messages:</b>
OTA_InsuranceQuoteRQ- Version 1
OTA_InsuranceQuoteRS- Version 1
OTA_InsuranceBookRQ- Version 1
OTA_InsuranceBookRS- Version 1



## 1.8 The Journey Concept - 2001 and Beyond

The OpenTravel Alliance (OTA) seeks to make better use of Internet technology to provide the travel industry and travelers with a greater array of communications channels and services, and in turn, provide a greater level of service to travelers. OTA's 2001B Specification makes important contributions toward this objective by offering specifications for the travel industry on travel planning, availability, package tours, reservations, and fulfillment of services.

As part of these efforts, OTA also seeks to broaden the concept of travel services, to meet the needs of travelers that extend beyond the industry's traditional definition of the term. To reach this goal, OTA has identified the traveler's *journey* as a basic idea upon which to organize further development of its message and service specifications.

### Scope of these specifications

OTA has specified a set of business processes for planning travel through the fulfillment and analysis of the travel experience, all of which play a vital role in one's journey. During the planning phase, the functions involve a choice of destination (if one has that choice), with the meeting of personal interests or business objectives, such as climate, activities, or recreational services. This planning function can involve searching by name, finding airports or railway stations in the proximity of a geographic location or the final destination, etc., as well as include making a choice among lodging options.

Planning a trip is different from buying other goods or services because of the interrelationship between the individual items in the complete package. Items can be related *commercially* (packages, car rental/flight combinations, etc.) and/or by *time* and *place*. Unifying this relationship is the fact that they all belong to an individual's trip plan.<sup>4</sup>

The journey also involves taking action on those plans, beginning with determining availability of the services required, covering transportation, lodging, car rental, and travel insurance if required. A distinct feature of the travel business is that one can purchase (traditionally as well as over the Internet) from horizontally as well as vertically oriented businesses. If the traveler is fortunate enough to be planning a trip for a holiday, he/she can take advantage of the planning having largely been done for them in a tour package. Business travelers, on the other hand, may address their needs independently, or with the assistance of travel planning software or the services of a travel agent.

A traveler's journey begins in the planning stages by determining the boundaries of time and location within which the travel needs to be fulfilled. These requirements form the foundation of a Journey structure. To add specific information, one would need to fill in a "*segment*" of the journey, which provides a framework for storing individual components of trip-related data throughout all phases of the trip planning.

---

<sup>4</sup> We thank Vincent Buller, formerly with AND Data Solutions, for his contributions to this treatise on the Journey Concept.

Segment, in its broadest definition, includes a portion of the journey, such as the requirement to have a flight from Point A to Point B, but does not necessarily have to be filled in until a selection has been made for the service that will fulfill the need. A Segment defines a piece of the journey that may include the flight, hotel stay, insurance, kennel stay, etc. A segment would have any other information that would apply to the segment in general, such as a start date or end date. Each segment may be in a different stage of completeness with additional segments being added for return flights, hotel stays etc., as needed. At some point, the journey document, made up of individual segments, can be sent to travel suppliers for a quotation request.<sup>5</sup>

Travel providers are then free to respond to a request. If the request has been directed singularly to a supplier, that supplier (e.g.: an airline) may offer the best deal on a flight, while a travel agent may put together a package that includes hotel and car rental within the constraints of the initial journey.

A "service" is the actual product or service that the customer eventually purchases, and it may be desirable to have more than one service in any given segment. For example, only after receiving a first quotation it may become apparent that it is more economical to pay for one or two extra hotel nights in order to get a less expensive flight that stays over the weekend, etc. If the preferred vendor cannot fulfill the request, e.g.: the hotel is not available for the longer length of stay, than an alternative needs to be considered within the same segment.

The relationships between travel services make trip planning very much an interactive process. Determining availability means considering target dates and pricing options or limitations. While events in a trip may occur in chronological order, several events could be running at the same time. For example, a car rental event can be running simultaneously with a hotel stay. The time relationship does communicate important information, however, such as the customer requires a parking space that may need to be reserved along with the hotel reservation.

The responses of the travel providers will assist in identifying the specific services that will ultimately arrive at the point at which the traveler can make the booking request. The commercial relationships link together the items that were previously defined only as time and geographic location, but now attaches essential trip information such as reservation numbers, electronic tickets, vouchers, etc. After all the travel items are finalized, a travel insurance broker could be given a look at the journey information in order to quote for insurance covering the complete trip.

Travel, like any human activity, however, does not always take a direct path from plans through fulfillment. The traveler may wish to book additional facilities, such as reserve a table in a restaurant near the hotel, or plan an extra step into the journey. Changes in flight arrangements, such as a delay or cancellation, requires an update of hotel and car rental information, and other portions of the trip that could be affected as well. Because the Journey concept is valid in all travel phases, not just the booking and reservation phase, it is a concept that fits the interactive nature of travel.

When the traveler faces the need to make changes in itineraries, this means changes in dates, times, destinations, services, and pricing. The traveler's changes affect not only

---

<sup>5</sup> We thank Vic Roberts, OAG, for his contributions to the Journey Concept.

the travel services directly, but the changes also need to be recorded and synchronized with all of the travel supplier's reservations and information services. For example, a cancellation of a room reservation adds to a property's inventory that needs to be communicated quickly and accurately to the parent company's central reservation service and made available to their web site, as well as those run by the individual property.

Travel services need to record in real time the traveler's engagement of travel services throughout the journey. The documentation of these services becomes part of the travel service's record, either for historical purposes or as part of an audit trail. Companies in the travel industry, like companies anywhere, need to get paid for the services they provide, and the specifications include the identification of payment methods and indicators for the financial settlement. OTA will rely on existing vocabularies in the financial services industry for the semantics and mechanics of financial settlements.

### **The Programmer's View of the Journey**

OTA intends to provide a comprehensive set of journey-oriented, loosely-coupled travel services that will enable nontraditional as well as traditional business services. Because the Journey Concept is valid in all travel phases, it is a concept that fits the iterative process of trip planning.

Currently, the OTA specifications contain independent messages from each industry vertical developed by supplier working groups. To the extent possible, the OTA has compared similar data elements across industry verticals so that the same data tag will be expressed similarly where it is encountered in another set of specifications. However, it is inevitable that specifications developed independently from one another will require a cross-industry comparison of the structure of the messages as well as a process of data normalization that recognizes reusability and provides a mapping between them.<sup>6</sup>

The current OTA infrastructure is designed to handle multiple related documents by identifying those documents in the Manifest and referencing them in the reply.

Atomic messages have been retained for both the request and reply. Each message is identified in the Document Reference element of the Manifest based upon the string that is the root tag of the document. Although the EchoToken is an attribute on the root element that can be used to tie the response back to the request, the EchoToken in the OTA\_v2ent.xsd file is optional. Since it is not required, it should not be relied upon to be used or to identify the message in all cases.

Ultimately, it may be possible to develop a structure for a combined message that may include all supplier messages in one payload document. However, complications may arise from a message combining documents intended for separate suppliers as the documents can be encrypted differently depending upon agreements between parties. Multiple documents have been deemed to allow for more flexibility, and require less processing, as a multi-part document must be parsed in order to split out a message and send it to different servers.

### **The Traveler's View of the Journey - 2001 and Beyond**

OTA's work on the Profile specification defines general traveler preferences such as smoking/non-smoking, company travel arrangements and fares for business trips, etc.

---

<sup>6</sup> This task may be an ongoing role for the OTA Data Content team.

These preferences can be extracted using the journey concept to apply a place and time relationship to describe traveler preferences for a particular trip.

For many customers, the journey encompasses needed services that go well beyond the traditional travel services of airlines, passenger rail, lodging, or car rentals. While these basic services are indeed important to getting travelers safely to their destinations and providing for a reasonable level of comfort, many travelers will need other kinds of assistance in order for their travels to be successful. Here are a few examples:

- Business travelers conducting meetings in distant cities may have literature to distribute to their associates. They could carry the materials with their hand baggage on the airplane or check the box as accompanied baggage. However, many of these business travelers would also pay for a reliable package delivery service that could relieve them (and the air carrier) of an extra piece of baggage and deliver the package to their destination. Why can't business travelers arrange for this service while planning their travel?
- Many travelers headed to downtown locations in large cities rarely need a car. For these travelers who need to get from the airport to downtown, however, the only chance they have to make a decision on ground transportation (shared van, airport bus) is when they get to airport and retrieve their luggage. Why can't travelers make arrangements for ground transportation, perhaps other than taxis, when making the air or hotel arrangements?
- Finding a suitable restaurant is a continuous need of travelers. While some hotels provide concierge services to help travelers with meal reservations, sometimes being able to make those arrangements in advance can help a traveler's business as well as culinary interests. While a few restaurant reservation services have begun appearing on the Web, why can't they be integrated with traditional travel services?
- When travelers have pets, they can leave them with friends or family members, but when these options are not available, what are the choices? Finding a suitable boarding kennel that can take a pets on short notice is a definite need of these travelers. Why not have this service as an option for these travelers?

The examples of the journey concept given above are not exhaustive, but they show the kinds of services that travelers consider part of their travel experiences, yet are rarely offered by companies in the travel industry. Including these new kinds of services along with the traditional travel services requires communication methods and protocols that cover the traditional travel services, yet can still extend in a consistent and reliable way to include these non-traditional offerings.

To fulfill the promise of the journey concept will mean bringing in many companies not yet involved in the travel industry, which presents a whole new set of challenges. Many of the companies offering these new kinds of services are smaller in size and do not have the extensive resources needed to develop sophisticated information systems. As a result, any framework adopted by OTA to incorporate these services needs to be

designed for use by companies of all sizes. Any specifications of this kind also must be available worldwide, since the travel industry and the needs of travelers extend worldwide.<sup>7</sup>

### **Meeting the journey's data exchange requirements**

The Electronic Business XML or ebXML initiative offers a new worldwide specification for electronic business messages and services designed for businesses ranging from the smallest operation to multi-national conglomerates. EbXML, a joint enterprise of the Organization for the Advancement of Structured Information Standards (OASIS) and the United Nation's trade facilitation agency (UN/CEFACT) will complete its technical specifications in May 2001.

EbXML is a set of specifications that together enables a modular, yet complete electronic business framework. The ebXML initiative is designed for electronic interoperability, allowing businesses to find each other, agree to become trading partners and conduct business. All of these operations can be performed automatically, minimizing, and in most cases completely eliminating the need for human intervention. This streamlines electronic business through a low cost, open, standard mechanism.

EbXML is designed to meet these needs and is built on three basic concepts: provide an infrastructure that ensures data communication interoperability; provide a semantic framework that ensures commercial interoperability, and also provides a mechanism that allows enterprises to find each other, agree to become trading partners and conduct business with each other.

1. The infrastructure to ensure data communication interoperability is provided through:
  - a standard message transport mechanism with a well defined interface, packaging rules, and a predictable delivery and security model
  - a 'business service interface' that handles incoming and outgoing messages at either end of the transport
2. The semantic framework to ensure commercial interoperability is provided through:
  - a metamodel for defining business process and information models
  - a set of re-useable business logic based on core components that reflect common business processes and XML vocabularies
  - a process for defining actual message structures and definitions as they relate to the activities in the Business Process model.
3. The mechanism to allow enterprises to find each other, agree to establish business relationships, and conduct business, is provided through:
  - a shared repository where enterprises can register and discover each other's business services via partner profile information

---

<sup>7</sup> Many thanks to Alan Kotok, former OTA Standards Manager, for providing this vision of the future.

- a process for defining and agreeing to a formal Collaboration Protocol Agreement (CPA), if so desired or where required.
- a shared repository for company profiles, business process models and related message structures<sup>8</sup>

OTA is an early and long-time supporter of ebXML. Representatives from OTA took part in most of the ebXML development meetings, beginning with its organizational session in November 1999. OTA also contributed the business content of its first proof of concept test in May 2000.

With ebXML as a foundation, OTA can build a set a specifications addressing issues of travel planning, availability, package tours, reservations, and fulfillment for travel services and still leave open the opportunity for other kinds of services anywhere in the world to interact with this industry. Once operational, this extensible framework will provide tangible benefits for travelers and open up new opportunities for the industry as a whole.

---

<sup>8</sup> “Enabling Electronic Business with ebXML,” ebXML White Paper, December 2000.

## **1.9 OTA Generic Cancel Request**

It may be with a sense of irony that a collection of specifications that address the functions of availability and booking a reservation begins with a cancellation. OTA has developed a generic cancel message for use within the travel industry because there are common patterns between industry verticals for canceling a reservation, and it is anticipated that the messages defined here would be used to cancel a reservation made using the specifications in this publication.

The basic pattern for all industry verticals is to identify the reservation by some form of unique identification number, whether that number is called a Record Locator Number, PNR, Reservation ID, Confirmation ID, or called by some another name. While the nomenclature may differ from system to system, the convention is universal: 1) identify the ID unique to the system assigned to perform the cancellation, and 2) receive a confirmation of the cancellation as a response.

If a travel service is not going to be fulfilled, the availability of inventory is affected, and the system holding the inventory will want to release that inventory. Typically, the cancel action is executed by the receiving system, as it is usually the system that holds the reserved inventory. The cancellation allows the inventory to be returned back to the marketplace and enables the supplier to resell it as quickly as possible.

A cancellation could conceptually be considered as a special case of a "modify" transaction because the record of the reservation is generally not removed immediately from a system, but placed into a different status. However, a cancel transaction differs from a modification because the entirety of the services booked will not be consumed. The information exchange differs because the system doing the cancellation does not have to return the bulk of the reservation information as is the case when confirming the modification of reservation information. The process becomes much simpler: send the unique identifier to the system with the indication that the action to be taken is to perform a cancel; receive a response that the action has taken place.

A cancellation also differs from the generic infrastructure verb, OTA\_DeleteRQ. The Delete action removes the record out of the database, while in a cancellation, the record of the reservation is not necessarily deleted. A cancellation is a business process driven by the business rules applied to the reservation, and effectively provides a change of status of that reservation.

### ***1.9.1 Consequences for Canceling***

When a cancel message is sent, two possibilities exist: the reservation may be canceled without penalty, or the cancellation incurs a penalty for doing so. For many travel services, perhaps the majority in this day and age of bargains, promotional fares and special rates, some sort of restrictions may apply. A cancellation of travel services may result in forfeiture of an amount paid for a guarantee, or deposit, etc.

### ***1.9.2 Read Request Prior to Cancel***

The logical steps that occur for a cancellation recognize that prior to sending a cancellation message, the party holding the reservation may wish to retrieve and review that reservation. Reasons for doing so include reviewing associated information that may communicate the cancellation policy, or simply to confirm the identification of the reservation to be cancelled, among a series of reservations held by that party.

An optional *Instance* value returned in a Read response, if implemented as a timestamp, may indicate the most recent record of the reservation and allow for synchronization if the requesting and receiving party do not hold the same reservation.

### ***1.9.3 Security considerations***

Inherent in the business practices of companies exchanging information is the requirement for some kind of qualification that entitles the requestor to receive the reservation information. Sufficient information should be sent for the receiving system to be able to recognize the requesting system, and to qualify it to receive the information.

Systems may assume a point-to-point connection upon receiving a request to retrieve a booking, but may still need to know who is on the terminal, particularly with transactions that may come from travel web sites where the individual has the opportunity to log on and control their own reservation. The booking engine, or application processing the request, is tasked to qualify the requestor so that it can be certain that the party is who they represent themselves to be. This usually involves furnishing some kind of information such as a last name, membership number, confirmation number, or credit card number (at least the last 4 digits of the credit card number) that was used for the reservation. That level of verification may be required even with the identification of the source and booking channel supplied by the Point-of-Sale information.

This generic cancel message assumes that the software asking for the read or cancel action has pre-determined the other party's rights for viewing at either end of the message conversation. The same considerations for the security of the connection, as well as identification of the requesting party, is true for transactions involving reading and modifying a profile.

It is common practice for many systems to keep a summary level view of a reservation, retaining information about the requestor. Therefore, within the remit of information in the request, some level of security can be applied to view or cancel that information. In many cases, sending in the transaction to retrieve a reservation must be made through the channel or source where the original booking took place.

Complications arise when, within a booking or channel, all of the other qualifiers might be right, but the request is not made through the same source. For example, a travel agency branch office could be handling a request for a customer who made the original reservation through another location. In that case, the retrieving application is then tasked to take the qualification to the next level and match the request up to the channel and source to display only those reservations that were booked through that agency.

Travel suppliers may wish to support returning all reservations that can be identified with their company, e.g.: using identifiable confirmation numbers, regardless of where the reservation was booked (e.g.: through their Central Reservation System, or through a web site, etc.). Conversely, the customer who had booked a reservation through a specific web site, could not go on another travel supplier web site, and retrieve a booking made on the competitor's site. The reservation could only be retrieved through the original source.

**Note:** It is likely that within one company, systems will be able to use the generic Read or Cancel request for conversations between trusted sources, such as an interface from their web site to a legacy system. Additional information may be needed to establish a level of confidence between trading partners. As the nature of XML is to be extensible, partners wishing to expand upon



requirements in their environment may use the TPA\_Extensions defined in the OTA Version 2001A publication.

#### ***1.9.4 OTA Cancel Messages***

This specification provides a request/response pair of messages to support the functionality of canceling a reservation.

The following action verbs are used as root elements of a payload document that seeks to cancel a reservation.

- **OTA\_CancelRQ** - Identifies the reservation and requests a cancellation.
- **OTA\_CancelRS** - Returns a list of rules that govern the cancellation, a cancellation number upon execution of the cancel action, or Warnings or Errors if the processing of the request did not succeed.

#### ***1.9.5 OTA Cancel Request***

The root element of the OTA\_CancelRQ contains the standard payload attributes found in all OTA payload documents as well as the attribute *ReqRespVersion*= that requests a specific version of the response message. As this is the first publication of the OTA cancel message set, currently the only valid value is "1".

The cancel request also has an attribute, *CancelType* = " ", that defines the action requested in the cancel message.

Attributes of OTA\_CancelRQ are as follows:

- **OTA\_PayloadStdAttributes** - includes the 5 standard attributes on all OTA messages.
- **ReqRespVersion** - Specifies requested response version.
- **CancelType** - An enumerated type indicating the type of request made for the cancellation. Valid Values are: (Initiate | Ignore | Confirm).

*Initiate* - Indicates the initial request to cancel a reservation.

*Ignore* - Indicates a roll-back of the request to cancel, leaving the reservation intact.

*Confirm* - Indicates a request to complete the cancellation.

#### ***1.9.6 Confirmation of Cancellation***

When a cancellation has been executed, the majority of systems will return a cancellation number. The identification number supplied serves to confirm that the action has taken place, and makes it possible to track monetary ramifications that may be associated with the cancellation.

A single cancel request can result in a transaction that straightforwardly cancels a reservation and returns a confirmation that the cancellation has taken place. Alternatively, the request to cancel a reservation may violate a business rule, subjecting the party to a cancellation penalty. In that case, it is desirable to return information about the cancellation penalty in the response.

The OTA\_CancelRQ message allows for the possibility to perform a two-phase transaction to inform the party wishing to cancel of the penalties that would be incurred, and to allow them to verify their intention to cancel. The *CancelType* attribute is a flag that can be set to the choices of (Initiate | Ignore | Confirm). This enables the message conversation to send in an initial request to

cancel a reservation, anticipate a response that returns the consequences of doing so, and allows for the option of rolling back the transaction and choosing to NOT cancel the reservation.

### 1.9.7 OTA Cancel Response

The <OTA\_CancelRS> message uses the external entity OTA\_v2ent.xsd file that defines the root element standard attributes found in all OTA payload documents, and the response options of returning the indication of Success, Warnings or Errors in processing the request. Additionally, it returns a collection of cancellation rules, with penalty amounts, if incurred, and an indication of the status of the cancellation request, either "Cancelled", "Pending", or "Ignored" if the transaction has been rolled back and the reservation remains intact.

Attributes of OTA\_CancelRS are as follows:

- **OTA\_PayloadStdAttributes** - includes the 5 standard attributes on all OTA messages.
- **Id** - The identification number of the cancellation.
- **Status** - An enumerated type indicating the status of the cancellation request. Valid Values are: (Pending | Ignored | Canceled).

*Pending* - Indicates initial the request to cancel a reservation is pending confirmation to complete the cancel action. Cancel rules may have been returned along with the response.

*Ignored* - Indicates the request to cancel was rolled back, leaving the reservation intact.

*Canceled* - Indicates the cancellation is complete. A cancellation ID may have been returned along with the response.

By providing the option of a two-step process, individual business rules may determine how their system processes the initial request. If there are no penalties involved in the cancellation, the cancel transaction can take place and the response return the cancellation number along with the status that the reservation has been cancelled.

If the processing system determines that a cancellation policy has been invoked, it may choose to send back the OTA\_CancelRS with the Status="*Pending*", accompanied by a collection of cancellation rules, allowing the originating party to determine if the cancellation should proceed. The originating system would then resend the OTA\_CancelRQ. A CancelType="*Ignore*" would anticipate a response with the Status="*Ignored*", thus ending the message conversation with no action being taken to cancel the reservation.

A CancelType = "*Confirm*" indicates a definitive "Yes" to process the cancellation. This message would anticipate the response of Status="*Cancelled*", along with the return of a Cancellation Id, and that transaction would complete the cancellation process. The cancel RQ is the same message in each case, with the CancelType attribute indicating the action to be taken on the request.

### 1.9.8 Cancel Rules

The <OTA\_CancelRS> message has two child elements: the reservation record is identified by the use of the UniqueId element, and the cancellation rules and/ or penalties are communicated by the use of the CancelRules collection that may return one or many CancelRule elements.

The attributes of the CancelRule element are as follows:

- ***CancelByDate*** - Identifies the date by which a cancellation should be made in order to avoid incurring a penalty.
- ***Amount*** - The monetary amount incurred as a result of the cancellation.
- ***CurrencyCode*** - The code that identifies the currency of the penalty amount, using ISO 4217.

The data type of the CancelRule element is #PCDATA, which can be used for text describing the cancellation penalties. Cancel Rule is a repeating element and can be used as many times as needed to communicate the cancel penalties and rules

### **1.10 Point of Sale (POS)**

Point of Sale data defines the identity of the party that is connecting to the system and is making a request for availability or to book a reservation. Aside from the security used in authenticating the connection, information about the connecting party identifies a business entity and determines the roles and privileges assigned by a trading partner relationship. This information is independent from the actual message data. Identification of the connecting party is used to determine eligibility for rates, discounts or promotions on travel products. Their identity, and the channel that they connecting on, also determines viewership and selection of available product offerings.

Determination of roles and privileges is not limited to a singular relationship with the connecting party, or even the last party connecting to a system, but may be a collection of multiple levels of identification. This may include: 1) a personal profile that identifies a membership in a frequent traveler program or private club; 2) a corporation employing that person that has a special negotiated rate for hotel rooms, or car rentals; and 3) a travel agency that is used to book these services, with whom the travel supplier has an agency agreement, etc. Each of these identifications may be necessary to qualify for multi-leveled rate structures.

In addition to identifying each source of travel request, the availability and price of travel products is also affected by the means by which the request is made, i.e. the channel through which the request is made. For example, if a member of an organization wishes to attend a convention, the member must first identify the organization in order to register for travel services at a group rate. The specific rate for the convention in the hotel may be restricted to booking only through the convention bureau, through a specific travel agency, or only via a specific web site that is set up for online registration.

The identification of multiple parties may also involve a list of routing hops of systems through which the request arrived at the supplier, and the path of the request may determine the availability of products and services, as some channels may be privileged to access them, while others are not.

Other business rules may determine that travelers who are residents of certain countries are eligible to receive certain rates, while others are not. In addition, the viewership of those rates may be available only through certain means, such as agreements with specific web sites, or travel agencies licensed to do business in certain geographic localities, etc.

If systems are able to filter requests by location, travel suppliers can provide product offerings based on the demographics of consumers, or do marketing and promotionals based on a specific region or locale.

#### ***1.10.1 Point of Sale Identification***

Point-of-sale data, contained in a **<POS>** element, communicates this information and allows the receiving system to identify the trading partner, or other entity that is making a request for availability or a booking. Because the identification of Point-of-Sale information must be iterative and allow for multiple levels to support the transmission of a message through multiple sources, it can be used recursively to provide the information needed. It is recognized that some systems are capable of supporting only a single-level of POS identification and generally choose the last system to have touched the message for purposes of identification and determination of rights and privileges assigned to a trading partner.

The POS object provides basic Point-of-Sale information from static data contained in a profile of the person or business entity, and utilizes the UniqueId construct assigned to the profile to identify each <Source> element in a recursive fashion. This information may be stored in a database of personal profiles or trading partners with whom the supplier has a relationship to do business. Additional information passed as Point-of-Sale information comprises dynamic POS information and is unique to the transaction.

### 1.10.2 UniqueId for company and personal profiles

Each record that identifies a unique account of trading partner or personal information may be maintained in a profile and used to respond to an availability request or to book a reservation. The Profile record **MUST** have a unique identifier assigned by the system that creates it with the tag name <UniqueId>. The unique identifier on the record **MUST** contain both a *Type* and an *Id* attribute. It **MAY** optionally include a *URL* and an *Instance* attribute.

- *Id* - This represents a unique identifying value assigned by the creating system, using the XML data type String. The *Id* attribute might, for example, reference a primary-key value within a database behind the creating system's implementation. In the POS information the <UniqueId> for example, could be an IATA/ATA assigned number or an ESRP number.
- *Type* - This enumerated attribute references the type of object this <UniqueId> refers to, and gives this element its generality. By convention, the *Type* attribute value is the same as the OTA element tag name for the referenced object, for example, "Profile".
- *URL* - This optional attribute is what makes a <UniqueId> instance globally unique outside the context of a single bilateral conversation between known trading partners. OTA recommends that the URL be referenced by the URL identifying the public HTTP v2 OTA discovery message implementation for each trading partner. Thus, it identifies the certifying body, such as IATA, ARC, ERSP, etc., or in the event that the entity is a business or corporation, the URL would be the public URL of the organization. The specific URL used may include a location for validation of certification.
- *Instance* - This optional attribute represents the record as it exists at a point in time. An *Instance* is used in update messages where the sender must assure the server that the update sent refers to the most recent modification level of the object being updated.

Possible implementation strategies for *Instance* values are:

- a timestamp
- a monotonically increasing sequence (incremented on each update)
- an md5 sum of the binary representation of the object in its persistent store

### 1.10.3 Examples of unique identifiers

A valid unique identifier **MAY** contain only the *Type* and *Id* (a unique string assigned by the system that created it) attributes:

```
<UniqueId Type="Profile" Id="1234567"/>
```

To ensure that a unique identifier is globally unique (in the universal namespace) add a *URL* attribute which includes a fully-qualified domain-name:

```
<UniqueId URL="http://provider1.com/OTAengine/" Type="Profile" Id="1234567"/>
```

This *Id* is assured of being globally unique in any namespace as the URL points to a unique domain name for the vendor assigned via the INTERNIC. **Note:** In the absence of having a public URL, the reference for this attribute could be determined by bilateral agreement.

#### 1.10.4 Enumerations of Profile Types

The OTA Profile contains an attribute, *ProfileType*, that defines what type of profile is stored in a system database. The following is an enumerated list of values for the *ProfileType* attribute:

Customer	CRO (Central Reservations Office)
Corporation	Rep (Representative) Company
Travel Agency	Airline
Wholesaler	Car Rental
Group	Hotel
Tour Operator	Cruise Line
Internet Broker	

#### 1.10.5 POS Element

The Point-Of Sale (POS) object allows for the identification of a single or multiple Source that represents a company or person.

The POS element contains one child element, *Source*, that can have one to many occurrences:

- **Source** - Identifies the source of the connecting party making a request.

The child elements of the *Source* element include the following:

1. **UniqueId** - The four attributes used to identify the unique number of the party from their profile: *Id*, *Type*, *URL* and *Instance*, as described above.
2. **Position** - An element that identifies the geographic position of the server that sends the request. The *Position* element uses the representation defined by ISO Standard 6709 to define a geographic point location.

The three attributes of the *Position* element include:

- *Latitude* – Distance North or South of the equator, expressed in degrees or decimal degrees, using ISO 6709 syntax of + for north of the equator, and – (minus) for locations south of the equator.
- *Longitude* – Distance east or west of the Prime Meridian, expressed in degrees or decimal degrees, using ISO 6709 syntax of + for west, and – (minus) for locations east of the Prime Meridian.
- *Altitude* – Altitude in meters. (This attribute is optional, and would normally not be used for point-of-sale identification)

3. **Booking Channel** - Identifies the type of channel used for connectivity that the request is coming through. Some systems require the identification of a channel type or switch system used to reach the supplier in order to honor a request to book a reservation.

The attributes of *Booking Channel* are as follows:

- *Type* - Indicates the type of booking channel from an enumerated list. Valid values: (GDS | ADS | SCS | PMS | CRS | TOS | All)

*Definitions:* Global Distribution System (GDS), Alternative Distribution System (ADS), Sales and Catering System (SCS), Property Management System (PMS), Central Reservation System (CRS), Tour Operator System (TOS), and ALL.

- *Primary* - Indicates whether the enumerated booking channel used for the connection in this transaction is the primary means of connectivity used by the source. Valid values: (Yes | No).

### 1.10.6 GDS attributes

Since booking requests to suppliers frequently use a Global Distribution System (GDS) for their routing, two attributes are found on the Source element to accommodate those systems:

- *AgentSine* - Identifies the in-house travel agent making the request.
- *PseudoCityCode* - Identifies the location or city code of the office making the request.

**Note:** The Position element can be used by systems that do not generate a PsuedoCityCode.

### 1.10.7 Additional Airline requirements

The OTA Air Work Group has identified additional point-of-sale information required to complete airline bookings.

- *ISOCountry* - Identifies the country of origin of the system making the request.
- *ISOCurrency* - Identifies the currency to be used for payment of the ticket.
- *AgentDutyCode* - Identifies the duty code of the agent.
- *CarrierCode* - Identifies the airline or GDS system making the request.
- *AirportCode* - Identifies the airport or city code of the system delivering the message.
- *FirstDepartPoint* - Identifies the first departure point of the itinerary, by airport code. This attribute is needed only when the request is from an ERSP.
- *ERSPUserId* - User Id for a business entity using the ERSP system or connection to make a booking. This is a second number required of a booking entity that goes through an ERSP provider.

**Note:** The *ERSPUserId* attribute may be a candidate for a recursive UniqueId if systems are set up to handle the recursive pattern of a UniqueId pattern.

### 1.10.8 POS - Sample XML message fragment

The following is an example of a multiple POS segment used in a message:

```
<POS>
  <Source PseudoCityCode= "35789AKW1000" AgentSine= "4678987659jd76">
    <UniqueId URL="http://switch.com/OTAEEngine/" Type="Profile" Id="111111"/>
    <BookingChannel Type = "GDS"/>
  </Source>
  <Source>
    <UniqueId URL="http://company1.com/rbd/" Type="Profile" Id="222222"/>
    <Position Latitude="+41.47" Longitude="+87.45"/>
  </Source>
</POS>
```

```
        <BookingChannel Type= "CRS"/>
    </Source>
    <Source>
        <Uniqueld URL="http://internal\_data\_storage\_loc" Type="Profile" Id="444444"/>
        <Position Latitude="+41.22" Longitude="+87.55"/>
    </Source>
    <Source>
        <Uniqueld URL="http://website.com/sitelicense/" Type="Profile" Id="333333"/>
        <Position Latitude="+39.54" Longitude="+88.3745"/>
        <BookingChannel Type = "ADS" />
    </Source>
</POS>
```



# **OpenTravel Alliance Message Specification**

## **Publication 2001B**

### **Section 2 - Car Rental Specifications**

## 2.0 Introduction<sup>9</sup>

This version of the Car Working Group (CWG) Specifications has been developed with the help of many devoted participants from the membership of the Open Travel Alliance and the CWG.

Under the guidance of the OTA, the car rental industry has come together, for the first in its history, to develop these specifications. With the biggest names in the industry, Alamo, Avis, Budget, Dollar, Enterprise, Hertz, National and Thrifty, the CWG combined efforts and business requirements to produce these messages. Each member company representative worked diligently to ensure their own business and technical needs were incorporated into both the Vehicle Availability and Vehicle Reservation Messages.

In future versions of the OTA Message Specification Document, the CWG plans to create a “suite” of messages that will encompass the entire car rental experience, from making the reservation through the internet to reaching one’s final destination. By utilizing the XML standard, OTA, and the CWG in particular, can unify what is today a fractured process and open communication between all travel business partners. Future message specifications include “message retrieve” and “car cancel” and will eventually incorporate wireless technology.

Bringing together the knowledge of virtually the who’s who of the car rental industry can only spawn a meaningful and comprehensive specification to be used by the entire travel industry.

### **Why Is This Important?**

This section deals with incorporating the use of industry approved common standards.

The free flow of information supporting travel services creates the ability for those travel services to be marketed and sold through multiple distribution channels, unlike today’s predominately single-channel travel services market. This opportunity to move and aggregate information is dramatically enhanced by the near universal access to the Internet.

Creating a standard “language” for travel suppliers and intermediaries to use allows for travel suppliers to differentiate their products and services directly with their true customer base and to move away from the homogenization of service created by today’s single distribution channel. The release of Version 1 of the CWG standards is an early step in the right direction.

### **What Should You Do With This?**

In short, start something! Now. We have provided several ideas below as ways to explore how these standards, and future versions of them, can provide benefit to your organization and position your company as an early-adopter in the travel industry.

- Begin a dialog within your organization regarding adoption. Someone needs to champion this issue. Find pockets of interest. Find sponsorship. Discuss the possible benefits. Get your company thinking about this opportunity.
- Enumerate the potential benefits, both near term and long term, for your organization. Explore what benefits accrue to early adopters, what risks are run by those who wait.

---

<sup>9</sup> Introduction to the Car Working Group specifications courtesy of David Joslin, Perot Systems, Inc.

- Determine what adoption might mean within your organization. Integration with legacy systems, writing new systems, tailoring your website, etc. What costs may be involved? Over what period of time? Can this be done incrementally or all at once?
- Encourage a planning process with both your information technology team and your business community. Encourage dialog.
- Set Goals!
- Provide feedback to the OTA on the thoughts and issues raised by your organization. This dialog can be incredibly valuable in future direction setting.

### **Summary**

A travel industry XML-based data standard will never reach its full potential unless the travel supplier community BEGINS its drive toward adoption and integration both within individual travel companies and within the travel industry as a whole.

You can make a difference. Do your part. Start now.

# **Open Travel Alliance**

## **Vehicle Availability & Rates**

### **Request/Response Message Specifications**

## 2.1 Vehicle Availability Rates Messages

The OTA version 2001B specification of Vehicle Availability Rates provides a request/response pair of messages to support the functionality of searching for available vehicles. The message set provides for two levels of inquiry. A request using the **OTA\_VehAvailRateRQ** may return specific availability for a vehicle or a range of vehicles including rates, vehicle type, descriptions, restrictions, and policies. The request message assumes the 'pull' model, meaning that the querying system will initiate the transaction and expect an immediate response from the pre-determined supplier system.

The syntax of the action verbs that are the root elements of a payload document used to exchange vehicle availability data are enumerated below:

**OTA\_VehAvailRateRQ** - Requests availability of vehicle(s) by specific criteria that may include preferences such as vehicle type, pickup location and date/time, return location and date/time. It may also include special instructions, coverages etc. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_VehAvailRateRS** - Returns information about the available vehicle that meets the criteria supplied at the time of request. The message may or may not include rates. If the rates are available then the various types of rates will be supplied. For example: Hourly, Daily, Weekly, Weekend and Monthly. The response message may include “Warnings” from business processing rules or errors if the request did not succeed. In case the rates are not available, the response will contain zero rates.

**Note:**

All the elements are indicated in **Bold** and are enclosed between < and >.

All the attributes are indicated in *italics* and are bulleted items.

## 2.2 Vehicle Availability Rates Request Message (OTA\_VehAvailRateRQ)

The root tag of <OTA\_VehAvailRateRQ> contains the standard payload attributes entity found in all OTA payload documents. Because the results of the search message could be quite numerous, the request also has an attribute, *MaxResponses*=" ", indicating the number of replies requested in the response message. The root tag also contains *ReqRespVersion* to indicate the version number requested for the response message.

<OTA\_VehAvailRateRQ> - This is the main Request element.

- *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as indicated below:
  - *EchoToken*- a sequence number for additional message identification assigned by the requesting host system. When a request system includes an *EchoToken*, the corresponding response message must include an *EchoToken* with an identical value.
  - *TimeStamp* - indicates the creation date and time of the message in UTC using the following format specified by ISO 8601:YYYY-MM-DDThh: mm: ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
  - *Version*- String value that indicates the version of the message.
  - *Target* - indicates whether the message is a test message or a production message with the default value of Production. Valid values: (Test | Production).
  - *SequenceNmbr* – A sequence number generated by the sending system that allows for tracking of message receipt or retrieval of specific message(s) for purposes of synchronization between systems.
- *ReqRespVersion* – An indication of the version of the response message that is required.
- *MaxResponses* – a positive integer value that indicates the maximum number of responses desired in the return.

<POS> - Point of Sale Identification. Identification number of the vendor that has made the vehicle availability request and agency number assigned by IATA, ARC, ESRP or TID.

Refer to the separate OTA documentation of the Point of Sale element for full details.

<VehRequest> - Contains elements and attributes that form a Vehicle request.

- *QuoteType* - Indicates whether the response should include a single rate or multiple rates. Single rate may be subjective and may vary from vendor to vendor. Valid values: (Single | Multiple).

**<CountryOfResidence>** - Country of residence identified by ISO country code.

- *CountryCode* - ISO Country code.

**<PickupInfo>** - Identifies the location from where the customer will rent the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single pickup location. The location code is decided between the vendors.
- *DateTime* - Indicates the pickup date and time when the customer would pick up the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<ReturnInfo>** - Identifies the location to where the customer will return the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single return location. The location code is decided between the vendors.
- *DateTime* - Indicates the return date and time when the customer would return the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<RateQualifier>** - Indicates the type of rates the customer is shopping for.

- *TravelType* - Indicates purpose of travel for which the customer will rent the vehicle. Valid values: (Business | Personal).
- *ShopRateType* - Indicates the type of rates the customer is shopping for. Valid values: (Business | Leisure | All).
- *CorpDiscountNbr* - Corporate discount number.
- *PromoCoupon* - Promotion/Coupon number honored by the vendor.
- *RateCode* - Vendor specific rate code to be used for Availability.
- *RateType* – Information on the type of rate. Valid values: (Daily | Weekly | Monthly | Weekend | Unknown)

**<Driver>** - Counts of different types of drivers on the rental. All the attributes of this element are optional.

- *AgeQualifyingCode* - A code representing a business rule that determines the charges for a driver based upon an age range. This age qualifying code is an attribute that allows a company to categorize drivers by age, but does not limit the business logic to the three categories of the additional attributes Adult, Young Driver, Younger Driver. Each business may have a different age or way of categorizing, thus pricing what is charged for the driver. For example, senior driver, young driver as second driver accompanied by an adult, etc. Use of this attribute allows greater flexibility than establishing the three fixed categories.
- *Count* - A positive valued integer to indicate the number of drivers based on the age range indicated.
- *Adult* - A positive valued integer to indicate the number of people above 25 years of age that will be driving the vehicle.
- *YoungDriver* - A positive valued integer to indicate the number of people between 21 and 24 years of age that will be driving the vehicle
- *YoungerDriver* - A positive valued integer to indicate the number of people between 18 and 20 years of age that will be driving the vehicle.

**<LuggageCount>** - Indicates the anticipated number of pieces of baggage the customer will be carrying in the vehicle. This is represented by a positive integer value.

**<PassengerCount>** - Number of passengers (excluding driver) for which the rental will be used.

**<FlightArrivalDetails>** - Details of the customer's arrival flight.

- *ArrivalLocation* - Arrival Airport Id. Should be a valid airport code.
- *MarketingAirline* - Identifies the Airline Company. The company name is unique amongst the airline vendors.
- *FlightNumber* - Flight number that the customer will arrive by.
- *ArrivalDateTime* - Indicates the tentative date and time of customer flight. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.

**<VehRentalPref>** - Various preferences requested for the rental. Some of the preferences may override the Customer profile preferences. Vehicle type is required.



- *SmokingInd*- Indicates the customer's preference for smoking. Valid values: (Yes | No).
- *GasPrePayInd* - Indicates customer's preference of option to pre-pay for gasoline with rental. Valid values: (Yes | No).

**<VendorPref>** - Indicates the preferred Vendor Company for car rental. If a company name is supplied, the rates will be supplied for the specific Vendor Company. The company name is unique amongst the vendors.

- *PreferLevel* - Indicates the preference level for the vendor. Valid values: (Only | Unacceptable | Preferred). Default is Preferred.

**<CompanyName>>** - the unique name identifying the company.

- *CompanyCode* – identifies the company using a company code
- *CodeContext*- identifies the context of the identifying code.

**<RentalPayment>** - Indicates preferences for the form of payment that will be used, if the request results in a reservation.

- *PaymentType* - Type of payment that will be used at the time of booking the reservation. Valid values: (CreditCard | DebitCard | BusinessAccount | CentralBill | DirectBill | Voucher | Cash | Check).
- *BillingId* - Card number/Account number associated with the payment type.
- *RefNumber* - Reference number to be used by the supplier for billing purposes. Example: Purchase Order Number.
- *CreditCardCode* – Code indicating the type of credit card code (e.g., MasterCard, AmericanExpress, Visa, etc.), or the type of central bill / debit card.

**<Coverage>** - Insurance Coverage needed for the rental.

- *CoverageType* - Categories of insurance preferred by the customer. Valid values: (VehDamage | Baggage | PersonalEffects | Liability | Theft | YoungDriver)
- *CoverageCode* - Industry code for type of insurance coverage.

**<SpecialReq>** - Special needs or requirements requested by the customer. This is a repeating element.

**<VehiclePref>** - Describe the details of the vehicle that is preferred.

<**VehicleTypePref**> - Major categories of vehicles.

- *PreferLevel* Valid values: (Only)
- *VehicleType* Valid values: (Car | Van | SUV | Convertible | Truck | Motorcycle | Limo | StationWagon | Pickup | MotorHome | Other). This is a required attribute.
- *VehicleDoorCount* This is an optional attribute.

<**VehicleClassPref**> - Details of vehicle types.

- *PreferLevel* Valid values: (Only | Preferred)
- *VehicleClass* Valid values: (Mini | Subcompact | Economy, Compact | Midsize | Intermediate | Standard | Full\_Size | Luxury | Premium | Minivan | 12\_Passenger\_Van | Moving\_Van | 15\_Passenger\_Van | Cargo\_Van | 12\_Foot\_Truck | 15\_Foot\_Truck | 20\_Foot\_Truck | 24\_Foot\_Truck | 26\_Foot\_Truck | Moped | Stretch | Regular | Unique | Exotic | Other).

<**VehicleMakeModelCodePref**> - A code for a specific make and model of vehicle, if supported.

- *PreferLevel* - Indicates the preference level. Valid values: (Only | Preferred).

<**AirConditionPref**> - Indicates the customer's preference for Air Conditioning.

- *PreferLevel* - Indicates the preference level. Valid values: (Only | Unacceptable | Preferred).

<**TransmissionPref**> - Indicates customer's preference for Vehicle Transmission.

- *Type* - Indicates the type of transmission. Valid values: (Automatic | Manual).
- *PreferLevel* - Indicates the preference level. Valid values: (Only | Unacceptable | Preferred).

<**SpecialEquipment**>- Indicates the preference for a specific item of additional equipment. This element may repeat for each unique type of equipment.

- *Quantity*- Number of this type of equipment that is requested.

- *EquipType* - Indicates the type of equipment that is preferred to be a part of this reservation Valid values: (CellularPhone | BikeRack | LuggageRack | SkiRack | TrailerHitch | AutomaticLocks | InfantSeat | ChildToddlerSeat | BoosterSeat | SnowChains | HandControlRight | HandControlLeft | NavigationalSystem | SnowTires | BabyStroller | DVDPlayer | VideoMonitor | SpinnerKnob | FurniturePads | CarDolly | AutoTransport | HandTruck | CargoBarrierFront | CargoBarrierRear | LuggageTrailer | CampingEquipment). This is required.

<**OnlineNews**> - Indicates whether the customer would prefer online news (if available) by the vendor.

- *OnlineNewsInd* - Indicates the choice for online news. Valid values: (Yes | No).

<**Customer**> - Information on the person who will be renting the vehicle. This information is optional.

- *BirthDate*
- *Currency*

<**ProfileRef**> - a reference to a profile for this customer, if such a profile is available.

<**PersonName**> - Full name of the customer renting the vehicle.

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *NameType*

<**NamePrefix**>- Salutation or honorific (e.g., Mr., Mrs., Ms., Miss, Dr., etc.).

<**GivenName**>- Given name, first name or names.

<**MiddleName**>- Middle name or names.

<**Surname**> - Family name/Last name

<**NameSuffix**>- Suffix that follows the surname (e.g., Jr., Sr., III, Ret., Esq., etc.).

<**NameTitle**> - Degree or honors, (e.g.: Ph.D., M.D., etc.) often used with a person's name.

**<TelephoneInfo>** - Telephone number(s) that the customer may be contacted at.

**<Telephone>**

- *PhoneTechType*
- *CountryAccessCode*
- *AreaCityCode*
- *PhoneNumber*
- *Extension*
- *PIN*

**<Email>** - Electronic mail address, in IETF format.

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *EmailType*

**<AddressInfo>** - Information about the physical address of the customer

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *AddressType*

**<Address>**

**<StreetNbr>**

- *PO\_Box*

**<BldgRoom>**

**<AddressLine>**

- *ParsedInd* Valid values: ( Yes | No )

**<CityName>**

- *PostalCode*

**<StateProv>**

- *StateCode*

**<CountryName>**

- *CountryCode*

**<CustLoyalty>** - Loyalty programs in which the customer is enrolled. Some loyalty programs may be affiliated with the vehicle rental vendor.

- *ProgramCode*
- *Privacy*
- *MembershipId*
- *SingleVendorId* Valid values: ( SingleVndr | Alliance )
- *LoyalLevel*
- *SignupDate*
- *EffectiveDate*
- *ExpireDate*

**<Document>** - Information on documents associated with this person, for example, driver license.

- *Privacy*
- *DocId*
- *DocType*
- *Gender*
- *BirthDate*
- *EffectiveDate*
- *ExpireDate*

**<DocHolderName>**

**<DocIssueAuthority>**

**<DocIssueLocation>**

**<DocIssueLimitations>**

### 2.3 Vehicle Availability Rates Response Message ( OTA\_VehAvailRateRS )

The root tag of <OTA\_VehAvailRateRS> uses the external entity that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The message may or may not include rates. If the rates are available then the various types rates will be supplied. For example: Hourly, Daily, Weekly, Weekend and Monthly. The response message may include Warnings from business processing rules or Errors if the request did not succeed. In case the rates are not available, the response will contain zero rates.

Additionally, just as the request message has the MaxResponses=" ", indicating maximum number of replies requested, the response message has an attribute "TotalReturns=" " to indicate the number of selections returned in the response. Use of these attributes enables both the requestor and responder to filter the number of request/responses exchanged.

<OTA\_VehAvailRateRS> - Main Response element.

- *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as indicated below:
  - *EchoToken*- a sequence number for additional message identification assigned by the requesting host system. When a request system includes an *EchoToken*, the corresponding response message must include an *EchoToken* with an identical value.
  - *TimeStamp* - indicates the creation date and time of the message in UTC using the following format specified by ISO 8601:YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour(military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
  - *Version*- String value that indicates the version of the message.
  - *Target* - indicates whether the message is a test message or a production message with the default value of Production. Valid values: ( Test | Production ).
  - *SequenceNmbr* – A sequence number generated by the sending system that allows for tracking of message receipt or retrieval of specific message(s) for purposes of synchronization between systems.
- *TotalReturns* – a positive integer value that indicates the maximum number of responses in the response.

<Success>

<Warnings>

<Warning>

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization |

ProtocolViolation | TransactionModel | AuthenticationModel |  
ReqFieldMissing )

- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

<**POS**> - Point of Sale Identification. Identification number of the vendor that has made the vehicle availability request and agency number assigned by IATA, ARC, ESRP or TID.

Refer to the separate OTA documentation of the Point of Sale element for full details.

<**VendorVehicleInfo**> This element is a child element of the vehicle availability response message. It contains information about the individual vehicle that has been requested. The VendorVehicleInfo shall be a single-occurrence element, reflecting the vehicle that is available.

<**Vendor**> Indicates the Vendor Company Name associated with this specific vehicle. The company name is unique amongst the vendors.

<**Mileage**> - Indicates whether mileage is expressed in Miles or Kilometers.  
Valid values: (MILE | KM).

<**PickupInfo**> - Identifies the location from where the customer will rent the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single pickup location. The location code is decided between the vendors.
- *LocationName* – Provides the name of the location, this is useful when the code does not easily identify the location, such as an off-airport location.
- *DateTime* - Indicates the pickup date and time when the customer would pick up the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

<**VehicleLocation**> – Descriptive information about where the vehicles are parked.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<CounterLocation>** – Descriptive information about where the rental counter is located.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<ReturnInfo>** - Identifies the location to where the customer will return the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single return location. The location code is decided between the vendors.
- *LocationName* – Provides the name of the location, this is useful when the code does not easily identify the location, such as an off-airport location.
- *DateTime* - Indicates the return date and time when the customer would return the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh: mm: ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<VehicleLocation>** – Descriptive information about where the vehicles are parked.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<CounterLocation>** – Descriptive information about where the rental counter is located.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<LengthOfRental>** - Rental duration. The duration may be expressed in Days, Hours, etc., or a combination of Days and Hours.

**<RentalPeriod>** This element can be used to describe individual rental periods. The element can repeat as needed, for example, it would occur twice to indicate a length of rental of 2 days and 1 hour.

- *Duration* - A positive integer value that indicates the length of this rental period.
- *Unit* - Qualifies the duration. Valid values: (Hour|Day|Week|Month).

**<VehAvailRate>** - Vehicle Availability Rate Information. This element is used to indicate the availability and rates associated with a vehicle.



- *AvailabilityInd* - Indicates the status of the vehicle associated with the availability request. Valid values: (Available | Unavailable | OnRequest)

<**Total**> - Specifies the total amounts for the reservation.

<**RateTotalAmount**> - The total base cost for the vehicle reservation.  
This amount excludes all the taxes and surcharges.

- *Amount*
- *CurrencyCode* - the currency code associated with the costs.

<**EstimatedTotalAmount**> - The total estimate of rates that include Base Rates, Taxes, Surcharges and other associated rental items.

- *Amount*
- *CurrencyCode* - the currency code associated with the costs.

<**Vehicle**>

- *AirConditionInd* - An indication if air conditioning is available in this vehicle. Valid values: (Yes | No)
- *TransmissionType* - An indication of the transmission type of this vehicle. Valid values: (Automatic | Manual)
- *GuaranteeInd* - Indicates that a guaranteed form of payment was used to reserve this vehicle.

<**VehicleType**> - Major categories of vehicles.

- *VehicleType* Valid values: (Car | Van | SUV | Convertible | Truck | Motorcycle | Limo | StationWagon | Pickup | MotorHome | Other). This is required.
- *VehicleDoorCount* This is an optional attribute.

<**VehicleClass**> - Details of vehicle types.

- *VehicleClass* Valid values: (Mini | Subcompact | Economy, Compact | Midsize | Intermediate | Standard | Full\_Size | Luxury | Premium | Minivan | 12\_Passenger\_Van | Moving\_Van | 15\_Passenger Van | Cargo\_Van | 12\_Foot\_Truck | 15\_Foot\_Truck | 20\_Foot\_Truck | 24\_Foot\_Truck | 26\_Foot\_Truck | Moped | Stretch | Regular | Unique | Exotic Other).

<**VehicleMakeModel**> - a description of the vehicle associated with this type and class.

- *Code* - A code describing the make and model of this vehicle.

**<RateQualifier>** - Indicates the type of rate that was used when completing the reservation. Data associated with this rate is also presented.

- *TravelType* - Indicates purpose of travel for which the customer is renting the vehicle. Valid values: (Business | Personal).
- *ShopRateType* - Indicates the type of rate associated with the reservation. Valid values: (Business | Leisure | All).
- *CorpDiscountNbr* - Corporate discount number applied to the reservation.
- *PromoCoupon* - Promotion/Coupon number that has been applied to the reservation.
- *RateCode* - The rate code applied to the reservation.
- *RateType* - Information on the type of rate. Valid values: (Daily | Weekly | Monthly | Weekend | Unknown)

**<PromoDesc>** - A description of the promotion that has been applied to the rates that are returned in this response.

**<RateMileage>** - Information on the mileage that is included with this rate.

- *Unlimited* - Valid values: (Yes | No)
- *Quantity* - number of miles or kilometers if not unlimited
- *VehUnitName* (RentalPeriod | Year | Month | Week | Day | Hour)

**<VehRateChargeInfo>** - Charges and rates associated with the vehicle. This element may repeat for each charge, fee, surcharge, etc, that is to be represented.

- *VehRateType* - Information on the type of charge, that is, what is it for? Valid values: (VehicleRental | Drop | Extra | Discount | Insurance | Surcharge | Fee | Tax | AdditionalMileage | AdditionalWeek | AdditionalHour | AdditionalDay | AdditionalDriver | YoungDriver | Fuel | Other )

**<VehCharge>** - This element provides details on the specific charge, including unit cost, quantity and total cost.

- *Total* - the total cost of this charge.

- *CurrencyCode* - the currency code associated with the costs.
- *UnitCharge* - the cost per unit of this charge. The unit has been identified in VehUnitName.
- *VehUnitName* - Information on the unit for which this charge is expressed. Valid values: (RentalPeriod | Year | Month Week | Day | Hour | Mile | Km | Gallon | Liter )
- *UnitQuantity* - the number of units on which the charge is based.
- *Percentage* - a numerical value if the charge is based on a percentage rather than on a unit cost.
- *MaxCharge* - the maximum charge that will apply, for example, a child seat may have a charge of \$5.00 per day with a maximum charge of \$20.00
- *MinCharge* - the minimum charge that will apply, for example, a VehicleRental charge may be \$9.99 per day, but requires a two-day minimum rental. Minimum charge would be \$19.98.
- *Taxable* - An indication if this charge is taxable. Valid values: (Yes | No)
- *AmountInclusive* - An indication if this charge has been included in the base rates. Valid values: (Yes | No).

<**Description**> - A description of the charge, for example, "Airport Surcharge".

<**TaxAmounts**> - Tax charges that are applied to the rate. Note that this object allows for both percentages and currency amounts.

<**TaxAmount**> - provides the amount of the tax

- *Total* - the amount of this tax
- *TaxCode* - Code identifying the tax (e.g., occupancy tax, city tax, etc.).
- *CurrencyCode* - Code of the currency associated with this tax.
- *Percentage* - the tax percentage

<**Description**> - description of the tax.

**<InternetId>** - Vendor's Internet address.

**<VendorMessages>** - allows the vendor to present informational messages relevant to the availability or reservation of vehicles. These may include policy information messages, messages providing special instructions relative to vehicle collection or return, or general marketing messages.

- *Type* - identifies the category of message. Valid values: (Policy | Marketing | SpecialInstructions | Other)
- *Title* - The header of the message.

**<Paragraph>** - An indication of a new paragraph for this section of the message. This element will repeat for each paragraph in this message.

**<Text>** - The text describing the message. This element would be used if the paragraph can be re-formatted to fit the available display space.

**<Line>** A line of information that has been formatted by the vendor and which should be shown in a fixed width font, without reformatting. This element may repeat for each line of formatted text.

**<SpecialEquipment>** - A collection of Special Equipment items that have been associated with the reservation, along with their corresponding charges.

- *Restrictions* - An indication of there are any restrictions associated with this equipment and the type of rental. Valid values: (OneWayOnly | RoundTripOnly | AnyReservation)
- *Quantity* - Number of this type of equipment that was requested
- *EquipType* - Indicates the type of equipment that is preferred to be a part of this reservation Valid values: (CellularPhone | BikeRack | LuggageRack | SkiRack | TrailerHitch | AutomaticLocks | InfantSeat | ChildToddlerSeat | BoosterSeat | SnowChains | HandControlRight | HandControlLeft | NavigationalSystem | SnowTires | BabyStroller | DVDPlayer | VideoMonitor | SpinnerKnob | FurniturePads | CarDolly | AutoTransport | HandTruck | CargoBarrierFront | CargoBarrierRear | LuggageTrailer | CampingEquipment). This is required.

**<VehCharge>** - This element provides details on the specific charge, including unit cost, quantity and total cost.

- *Total* - the total cost of this charge.
- *CurrencyCode* - the currency code associated with the costs.
- *UnitCharge* - the cost per unit of this equipment. The unit has been identified in VehUnitName.

- *VehUnitName* - Information on the unit for which this equipment charge is expressed. Valid values: (RentalPeriod | Year | Month | Week | Day | Hour | Mile | Km | Gallon | Liter )
- *UnitQuantity* - the number of equipment units on which the charge is based.
- *Percentage* - a numerical value if the charge is based on a percentage rather than on a unit cost.
- *MaxCharge* - the maximum charge that will apply, for example, a child seat may have a charge of \$5.00 per day with a maximum charge of \$20.00
- *MinCharge* - the minimum charge that will apply, for example, a child seat may have a charge of \$5.00 per day with a minimum charge of \$10.00.
- *Taxable* - An indication if this equipment charge is taxable.  
Valid values: (Yes | No)
- *AmountInclusive* - An indication if this charge has been included in the base rates. Valid values: (Yes | No)

<**Description**> - A description of the equipment, for example, "Child

<**TaxAmounts**> - Tax charges that are applied to the rate. Note that this object allows for both percentages and currency amounts.

<**TaxAmount**> - provides the amount of the tax

- *Total* - the amount of this tax
- *TaxCode* - Code identifying the tax (e.g., occupancy tax, city tax, etc.).
- *CurrencyCode* - Code of the currency associated with this tax.
- *Percentage* - the tax percentage

<**Description**> - description of the tax.

<**Coverages**> - Information on Insurance Coverage that has been requested.

<**Coverage**> - An indication of one type of coverage. This element may repeat for as many coverages as available.

- *CoverageType* - The category of coverage. Valid values: (VehDamage | Baggage | PersonalEffects | Liability | Theft | YoungDriver)
- *CoverageCode* - Industry code for type of insurance coverage.
- *CoverageCost* - Cost for this type of insurance coverage.
- *CurrencyCode* - Currency in which the cost is quoted.

<CoverSupplement> - Supplemental information

<CoverDescription> - Description of the coverage

<CoverLimits> - Limits of the coverage

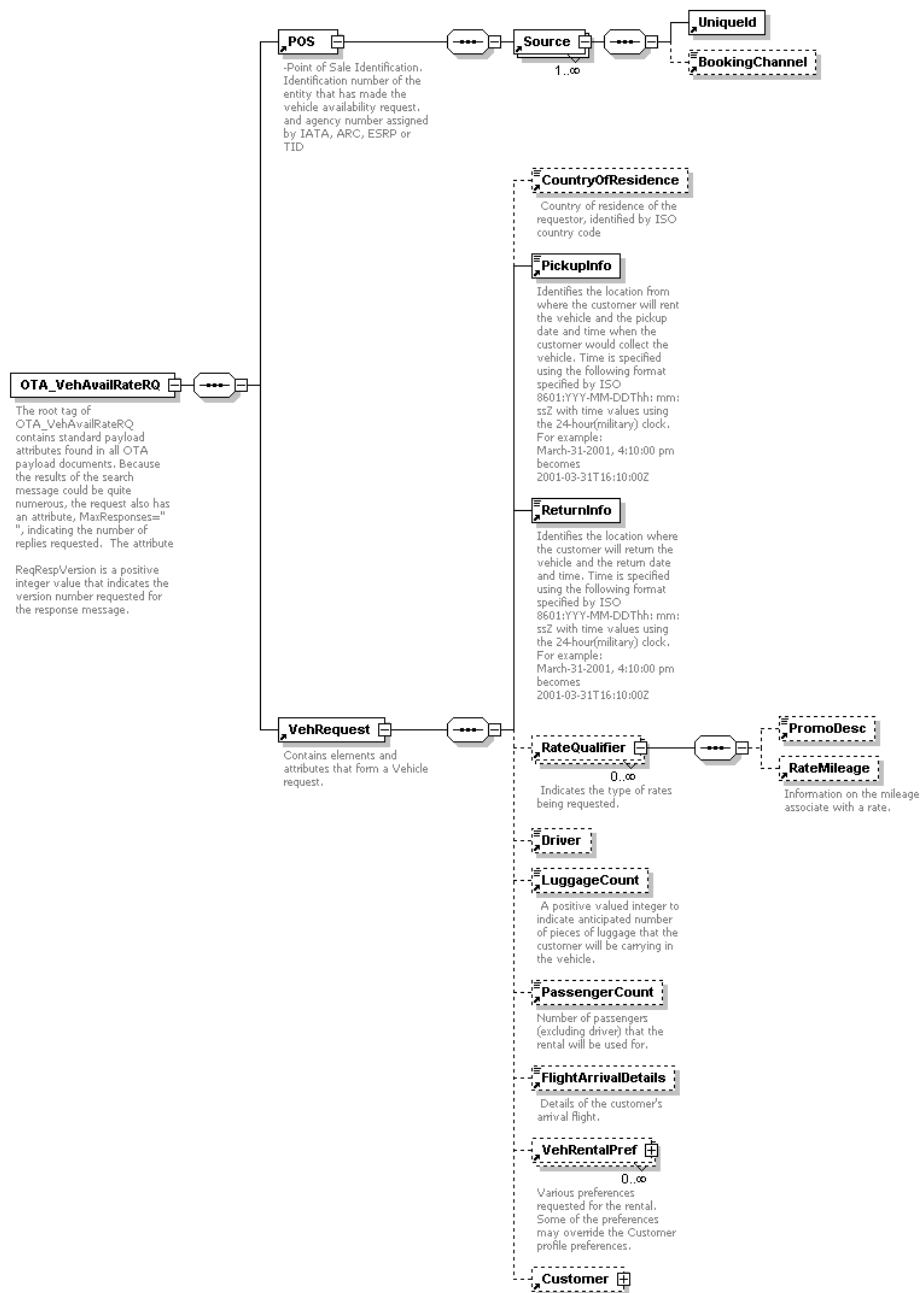
#### <Errors>

##### <Error>

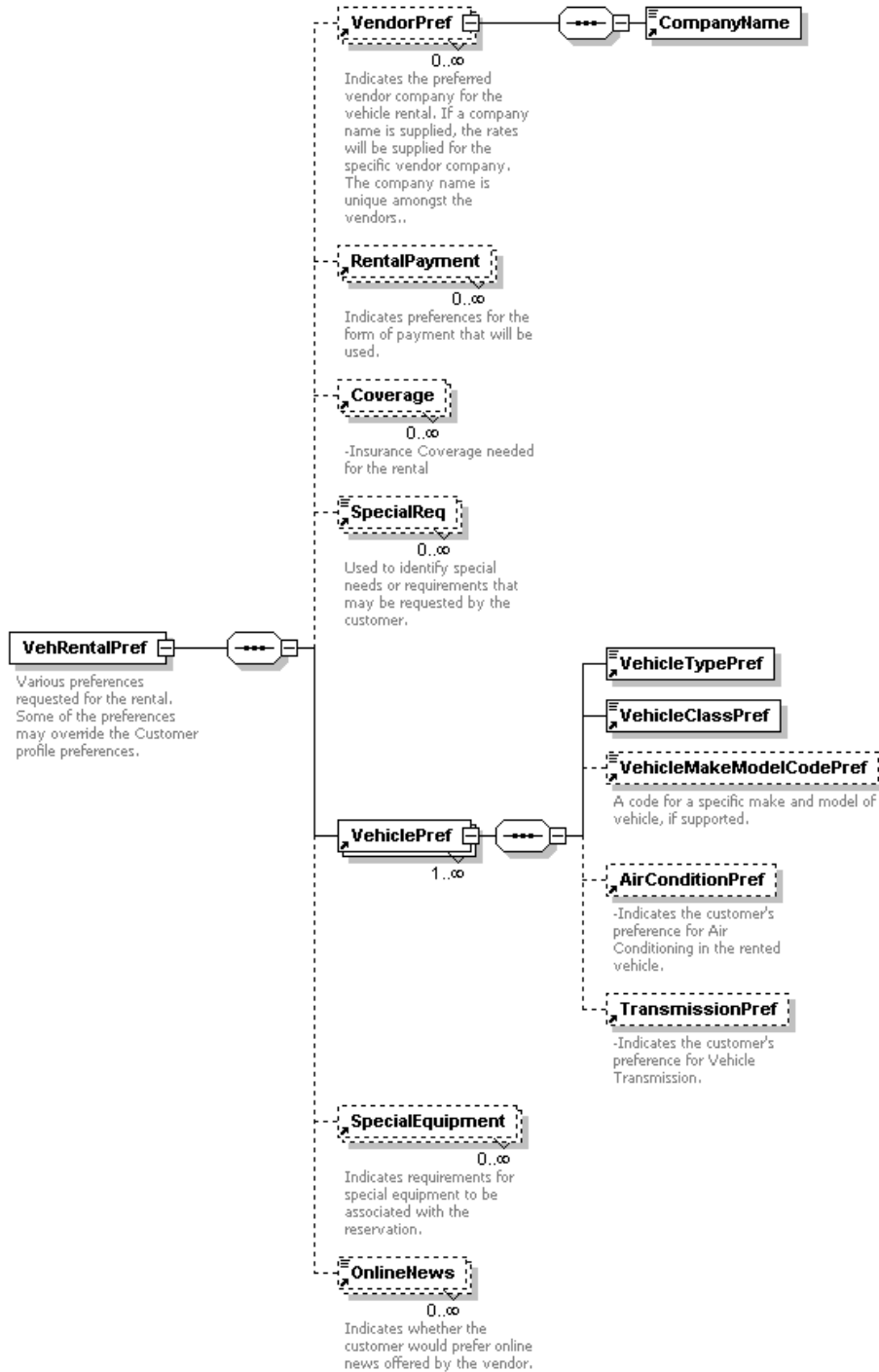
- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the error
- *DocURL*
- *Status*
- *Tag*

## 2.4 XML Schema Diagrams for Vehicle Availability Rates Request

### OTA\_VehAvailRateRQ



## VehRentalPref





## 2.5 Sample XML Message for Vehicle Availability Rates Request

This sample XML Message illustrates a request for Vehicle Availability and Rates for a rental period of 10 days, from May 5<sup>th</sup>, 2001, to May 15<sup>th</sup>, 2001, picking up from and returning to IAD airport. The request is for a car, with a preference of a Compact car. Stating this as a preference allows for information on other cars to be included in the response. Information on the renter is provided, along with the request to rent 2 Child Seats and a Ski Rack. A preferred vendor, XYZ, is indicated.

The example POS element reflects that this is a request for a corporate traveler, and that the request originated from a website. Information on both the corporation and the associated travel agency are provided.

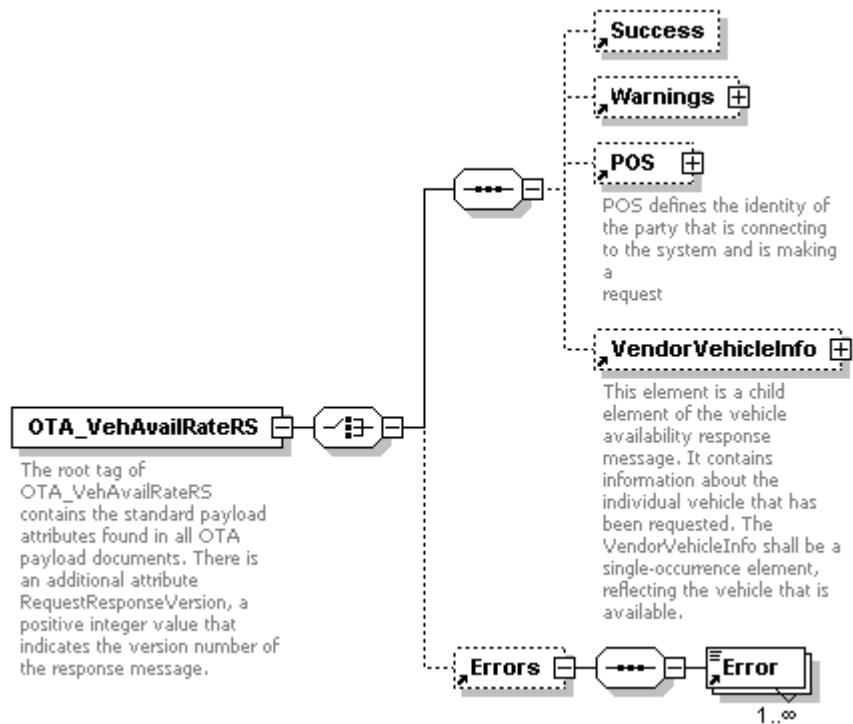
```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_VehAvailRateRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opentravel.org/OTA" xsi:noNamespaceSchemaLocation="OTA_VehAvailRateRQ.xsd"
EchoToken="18723222" TimeStamp="2001-03-31T16:10:00Z" Target="Production" Version="1" SequenceNmbr="1566"
MaxResponses="15">
  <POS>
    <Source>
      <Uniqueld Type="Corporate" Id="ID01"/>
      <BookingChannel Type="ADS" Primary="true"/>
    </Source>
    <Source>
      <Uniqueld Type="ARC" Id="ID02"/>
    </Source>
  </POS>
  <VehRequest QuoteType="Multiple">
    <CountryOfResidence CountryCode="US"/>
    <PickupInfo LocationCode="IAD" DateTime="2001-05-05T10:00:00Z"/>
    <ReturnInfo LocationCode="IAD" DateTime="2001-05-15T10:00:00Z"/>
    <RateQualifier TravelType="Personal" CorpDiscountNmbr="56432" PromoCoupon="B45"
RateCode="7S" ShopRateType="All"/>
    <Driver AgeQualifyingCode="SENIOR" Count="1" Adult="1" YoungDriver="0" YoungerDriver="0"/>
    <LuggageCount>3</LuggageCount>
    <PassengerCount>4</PassengerCount>
    <FlightArrivalDetails ArrivalLocation="IAD" MarketingAirline="AA" FlightNumber="50"
ArrivalDateTime="2001-05-05T09:00:00Z"/>
    <VehRentalPref SmokingInd="No" GasPrePayInd="No">
      <VendorPref PreferLevel="Preferred">
        <CompanyName>XYZ</CompanyName>
      </VendorPref>
      <RentalPayment PaymentType="CreditCard" CreditCardCode="VI"/>
      <Coverage CoverageCode="LIA" CoverageType="Liability"/>
      <SpecialReq>Require a non-smoking car</SpecialReq>
    </VehRentalPref>
    <VehiclePref>
      <VehicleTypePref PreferLevel="Only" VehicleType="Car"/>
      <VehicleClassPref PreferLevel="Preferred" VehicleClass="Compact"/>
      <AirConditionPref PreferLevel="Preferred"/>
      <TransmissionPref PreferLevel="Only" Type="Automatic"/>
    </VehiclePref>
    <SpecialEquipment Quantity="2" EquipType="ChildToddlerSeat"/>
    <SpecialEquipment Quantity="1" EquipType="SkiRack"/>
    <OnlineNews>No</OnlineNews>
  </VehRequest>
  <Customer>
    <PersonName>
      <NamePrefix>Mr</NamePrefix>
      <GivenName>John</GivenName>
      <Surname>Smith</Surname>
      <NameTitle>M.D.</NameTitle>
    </PersonName>
    <Email EmailType="Personal">johnsmith@home.com</Email>
    <AddressInfo>
      <Address>
```

```

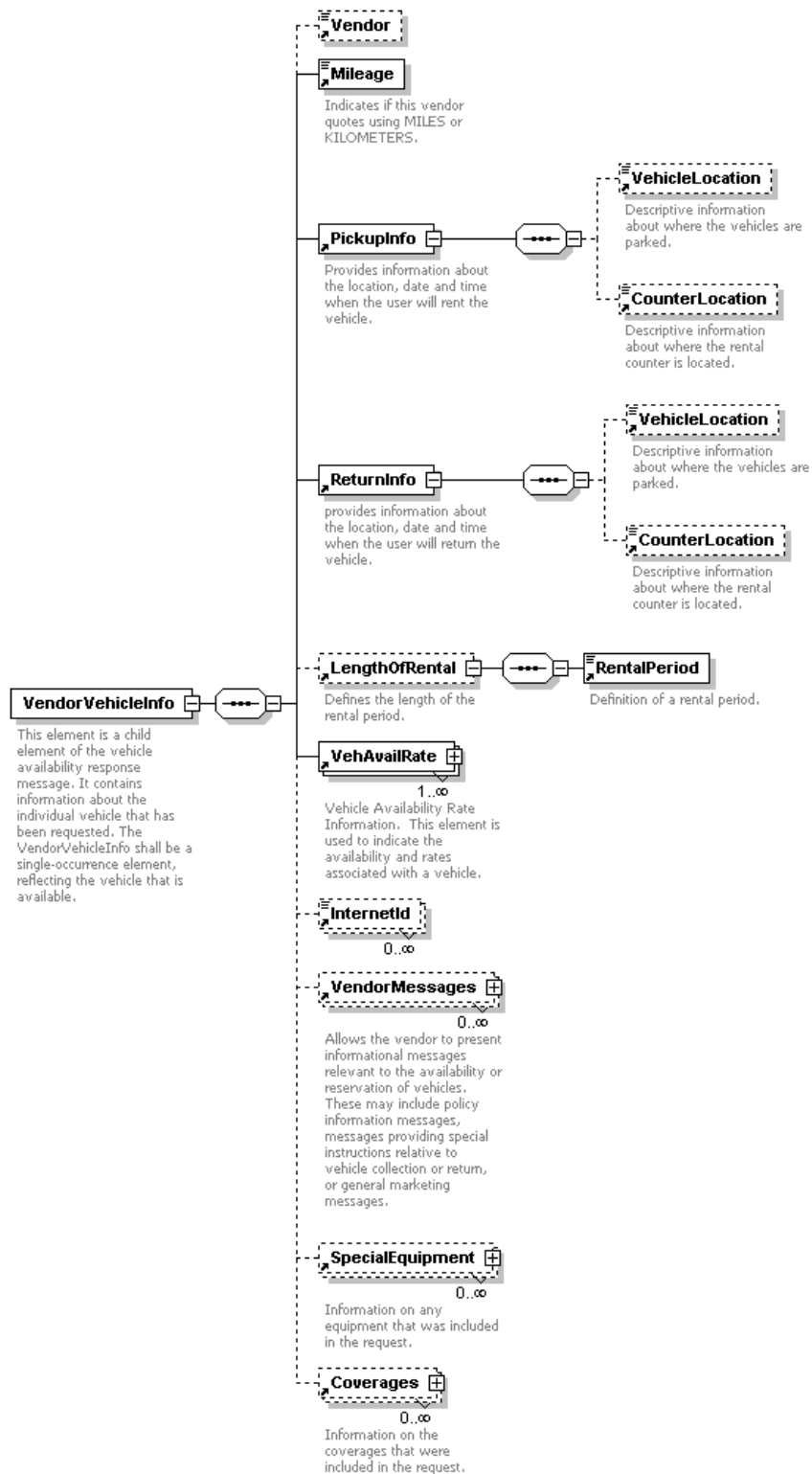
                                <AddressLine>12 Main Street</AddressLine>
                                <CityName PostalCode="75000">Dallas</CityName>
                                <StateProv StateCode="TX"/>
                                <CountryName CountryCode="US"/>
                            </Address>
                        </AddressInfo>
                        <CustLoyalty MembershipId="Q56GTF" ProgramCode="QuickRent"/>
                    </Customer>
                </VehRequest>
</OTA_VehAvailRateRQ>
```

## 2.6 XML Schema Diagrams for Vehicle Availability Rates Response

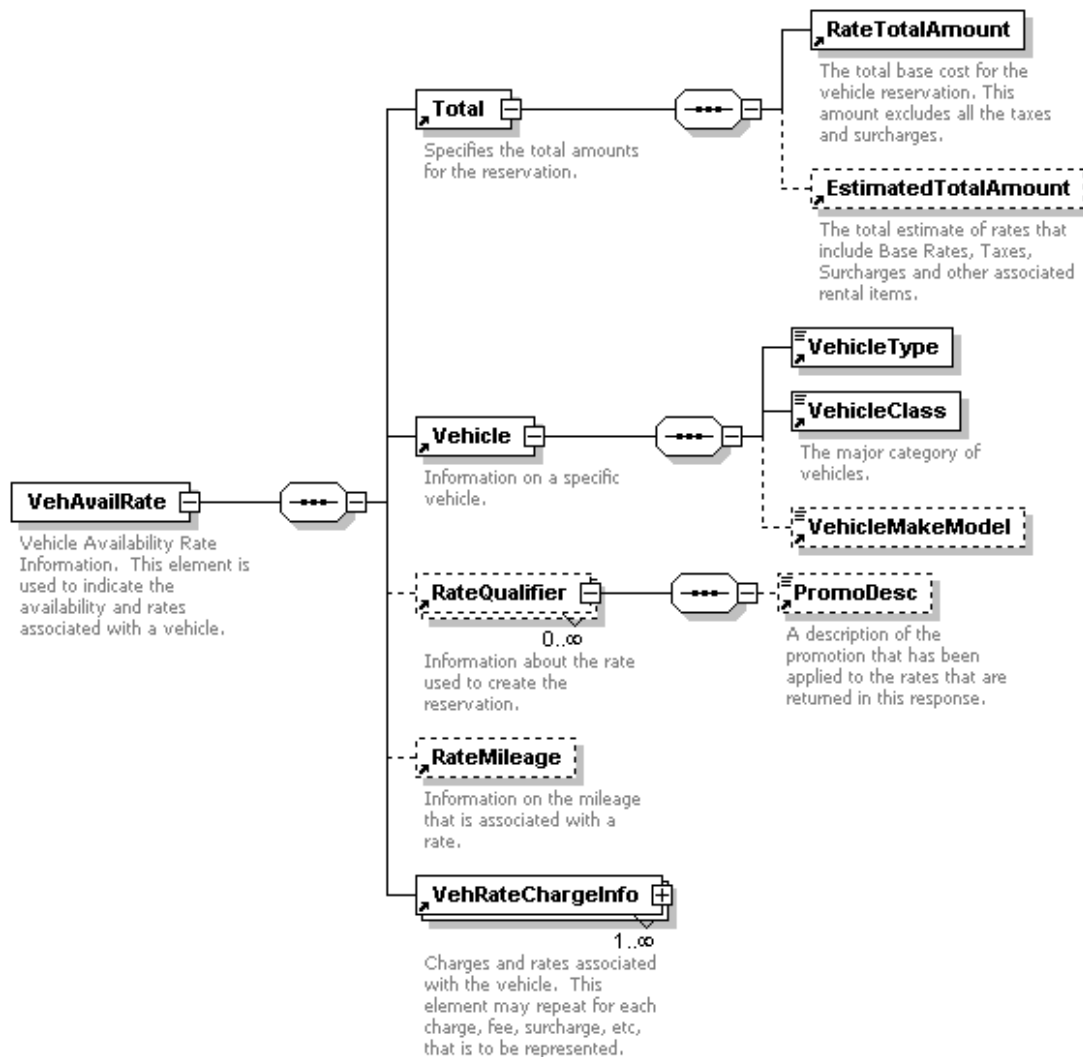
### OTA\_VehAvailRateRS



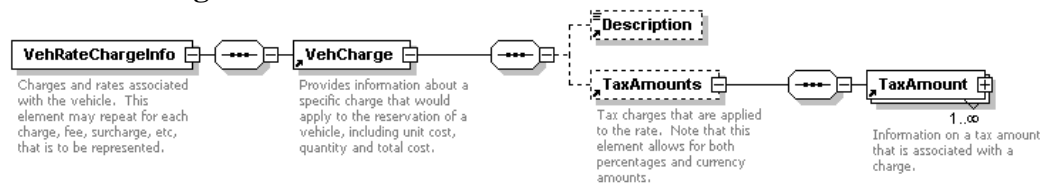
## VendorVehicleInfo



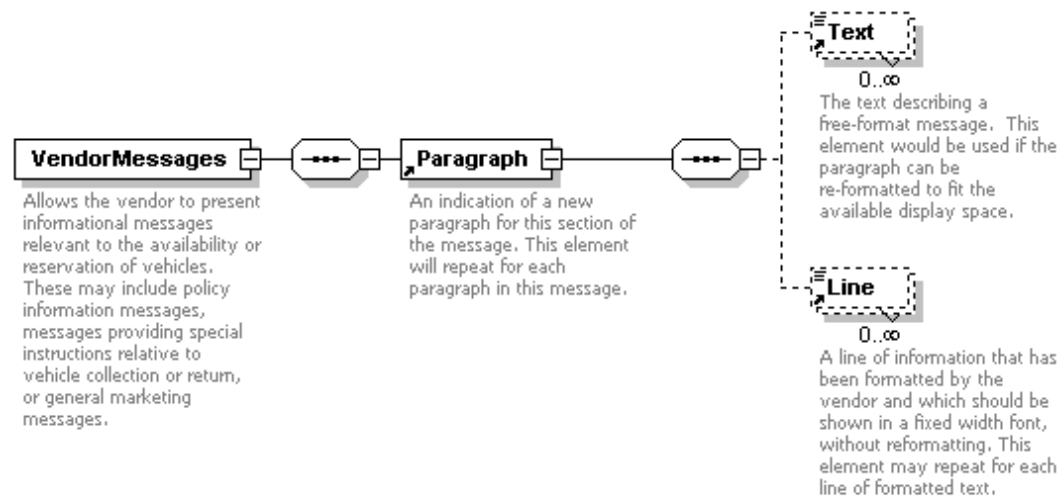
## VehAvailRate



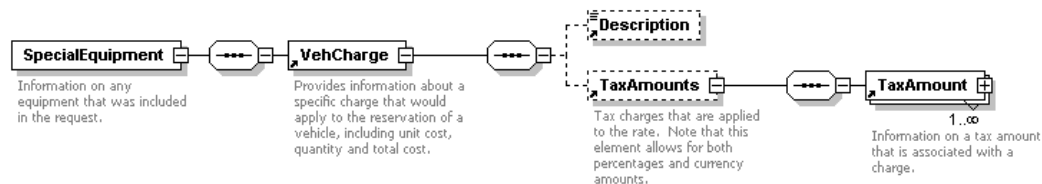
## VehRateChargeInfo



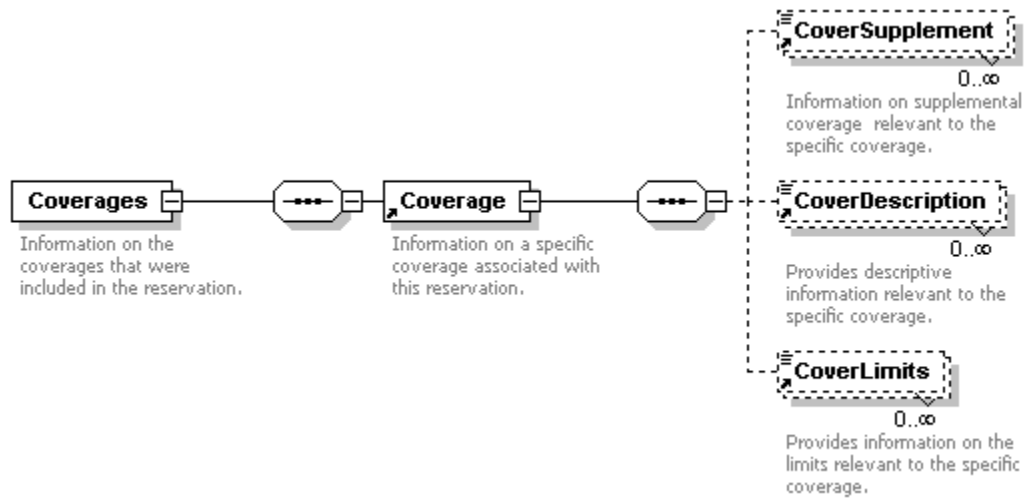
## VendorMessages



## SpecialEquipment



## Coverages



## 2.7 Sample XML Message for Vehicle Availability Rates Response

This sample XML Message illustrates a response for Vehicle Availability and Rates for a rental period of 10 days, from May 5<sup>th</sup> 2001 to May 15<sup>th</sup> 2001, picking up from and returning to IAD airport. This message reflects that there are three car types available, an Economy car, an Intermediate car and a Fullsize car. For each vehicle type the daily rate is provided (\$31.10, \$41.10 and \$51.10) along with the total rate cost and the total cost, inclusive of taxes, equipment rental, etc. Each vehicle type incurs a State Tax and a City Tax, along with a fixed Airport Fee.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_VehAvailRateRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.opentravel.org/OTA" xsi:noNamespaceSchemaLocation="OTA_VehAvailRateRS.xsd"
EchoToken="18723222" TimeStamp="2001-03-31T16:11:00Z" Target="Production" Version="1" SequenceNmbr="1567"
TotalReturns="3">
  <Success/>
  <POS>
    <Source>
      <UniquelId Type="Corporate" Id="ID03"/>
      <BookingChannel Type="ADS" Primary="true"/>
    </Source>
    <Source>
      <UniquelId Type="ARC" Id="ID04"/>
    </Source>
  </POS>
  <VendorVehicleInfo>
    <Vendor>XYZ</Vendor>
    <Mileage>MILES</Mileage>
    <PickupInfo LocationCode="IAD" DateTime="2001-05-05T10:00:00Z" LocationType="Airport">
      <VehicleLocation Location="ShuttleOnAirport">Shared Rental Car Parking Lot,
South</VehicleLocation>
      <CounterLocation Location="Terminal">Baggage Claim Level 2</CounterLocation>
    </PickupInfo>
    <ReturnInfo LocationCode="IAD" DateTime="2001-05-15T10:00:00Z" LocationType="Airport">
      <VehicleLocation Location="ShuttleOnAirport">Shared Rental Car Parking Lot,
South</VehicleLocation>
    </ReturnInfo>
    <LengthOfRental>
      <RentalPeriod Duration="10" Unit="Day"/>
    </LengthOfRental>
    <VehAvailRate AvailabilityInd="Available">
      <Total>
        <RateTotalAmount Amount="310.99" CurrencyCode="USD"/>
        <EstimatedTotalAmount Amount="510.99" CurrencyCode="USD"/>
      </Total>
      <Vehicle AirConditionInd="Yes" TransmissionType="Automatic">
        <VehicleType VehicleType="Car"/>
        <VehicleClass VehicleClass="Economy"/>
        <VehicleMakeModel Code="FE1">Ford Escot</VehicleMakeModel>
      </Vehicle>
      <RateQualifier TravelType="Personal" CorpDiscountNmbr="56432" PromoCoupon="B45"
RateCode="7S" ShopRateType="All" RateType="Weekly"/>
      <RateMileage Unlimited="No" Quantity="200" VehUnitName="Day"/>
      <VehRateChargeInfo VehRateType="VehicleRental">
        <VehCharge Total="310.99" CurrencyCode="USD" UnitCharge="31.10"
VehUnitName="Day" UnitQuantity="10" Taxable="Yes" AmountInclusive="Yes">
          <Description>Vehicle Rental Charge</Description>
          <TaxAmounts>
            <TaxAmount Total="5.00" CurrencyCode="USD"
TaxCode="STT">
              <Description>State Tax</Description>
            </TaxAmount>
            <TaxAmount Total="2.00" CurrencyCode="USD"
TaxCode="CTY">
              <Description>City Tax</Description>
            </TaxAmount>
          </TaxAmounts>
        </VehCharge>
      </VehRateChargeInfo>
    </VehAvailRate>
  </VehAvailRateRS>
```



```

        </TaxAmounts>
      </VehCharge>
    </VehRateChargeInfo>
    <VehRateChargeInfo VehRateType="Fee">
      <VehCharge Total="20.00" CurrencyCode="USD" UnitCharge="20.00"
VehUnitName="RentalPeriod" UnitQuantity="1" Taxable="No" AmountInclusive="No">
        <Description>Airport Fee</Description>
      </VehCharge>
    </VehRateChargeInfo>
  </VehAvailRate>
  <VehAvailRate AvailabilityInd="Available">
    <Total>
      <RateTotalAmount Amount="410.99" CurrencyCode="USD"/>
      <EstimatedTotalAmount Amount="618.49" CurrencyCode="USD"/>
    </Total>
    <Vehicle AirConditionInd="Yes" TransmissionType="Automatic">
      <VehicleType VehicleType="Car"/>
      <VehicleClass VehicleClass="Intermediate"/>
      <VehicleMakeModel Code="NA1">Nissan Altima</VehicleMakeModel>
    </Vehicle>
    <RateQualifier TravelType="Personal" CorpDiscountNmbr="56432" PromoCoupon="B45"
RateCode="7S" ShopRateType="All" RateType="Weekly"/>
    <RateMileage Unlimited="No" Quantity="250" VehUnitName="Day"/>
    <VehRateChargeInfo VehRateType="VehicleRental">
      <VehCharge Total="410.99" CurrencyCode="USD" UnitCharge="41.10"
VehUnitName="Day" UnitQuantity="10" Taxable="Yes" AmountInclusive="Yes">
        <Description>Vehicle Rental Charge</Description>
      <TaxAmounts>
        <TaxAmount Total="5.50" CurrencyCode="USD"
TaxCode="STT">
          <Description>State Tax</Description>
        </TaxAmount>
        <TaxAmount Total="2.25" CurrencyCode="USD"
TaxCode="CTY">
          <Description>City Tax</Description>
        </TaxAmount>
      </TaxAmounts>
    </VehCharge>
  </VehRateChargeInfo>
  <VehRateChargeInfo VehRateType="Fee">
    <VehCharge Total="20.00" CurrencyCode="USD" UnitCharge="20.00"
VehUnitName="RentalPeriod" UnitQuantity="1" Taxable="No" AmountInclusive="No">
      <Description>Airport Fee</Description>
    </VehCharge>
  </VehRateChargeInfo>
  </VehAvailRate>
  <VehAvailRate AvailabilityInd="Available">
    <Total>
      <RateTotalAmount Amount="510.99" CurrencyCode="USD"/>
      <EstimatedTotalAmount Amount="720.79" CurrencyCode="USD"/>
    </Total>
    <Vehicle AirConditionInd="Yes" TransmissionType="Automatic">
      <VehicleType VehicleType="Car"/>
      <VehicleClass VehicleClass="Fullsize"/>
      <VehicleMakeModel Code="FT">Ford Taurus</VehicleMakeModel>
    </Vehicle>
    <RateQualifier TravelType="Personal" CorpDiscountNmbr="56432" PromoCoupon="B45"
RateCode="7S" ShopRateType="All" RateType="Weekly"/>
    <RateMileage Unlimited="No" Quantity="300" VehUnitName="Day"/>
    <VehRateChargeInfo VehRateType="VehicleRental">
      <VehCharge Total="510.99" CurrencyCode="USD" UnitCharge="51.10"
VehUnitName="Day" UnitQuantity="10" Taxable="Yes" AmountInclusive="Yes">
        <Description>Vehicle Rental Charge</Description>
      <TaxAmounts>
        <TaxAmount Total="5.64" CurrencyCode="USD"
TaxCode="STT">
          <Description>State Tax</Description>
        </TaxAmount>
        <TaxAmount Total="2.34" CurrencyCode="USD"
TaxCode="CTY">

```

```

                                <Description>City Tax</Description>
                                </TaxAmount>
                                </TaxAmounts>
                                </VehCharge>
                                </VehRateChargeInfo>
                                <VehRateChargeInfo VehRateType="Fee">
                                    <VehCharge Total="20.00" CurrencyCode="USD" UnitCharge="20.00"
VehUnitName="RentalPeriod" UnitQuantity="1" Taxable="No" AmountInclusive="No">
                                        <Description>Airport Fee</Description>
                                        </VehCharge>
                                        </VehRateChargeInfo>
                                </VehAvailRate>
                                <SpecialEquipment Restrictions="RoundTripOnly" Quantity="2" EquipType="ChildToddlerSeat">
                                    <VehCharge Total="10.00" CurrencyCode="USD" UnitCharge="2.50" VehUnitName="Day"
UnitQuantity="10" MaxCharge="10.00" MinCharge="5.00" Taxable="No" AmountInclusive="No">
                                        <Description>Cost for Rental of Child Seat</Description>
                                        </VehCharge>
                                </SpecialEquipment>
                                <Coverages>
                                    <Coverage CoverageType="VehDamage" CoverageCode="LIA" CoverageCost="100.00"
CurrencyCode="USD">
                                        <CoverSupplement/>
                                        <CoverDescription>All damage except first 100.00</CoverDescription>
                                        <CoverLimits/>
                                    </Coverage>
                                </Coverages>
                                </VendorVehicleInfo>
</OTA_VehAvailRateRS>

```

## 2.8 VehAvailRateRQ/RS Revision History

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
March 30, 2001	0.1	Initial Draft.
April 4, 2001	0.2	<p>Changes resulting from the telephone conference.</p> <ol style="list-style-type: none"> <li>1. Changed Car Availability to Vehicle Availability Rates.</li> <li>2. Renamed &lt;IATA Nbr&gt; attribute to a generic Tag.</li> <li>3. Removed &lt;SummaryInd&gt; as the Car rental vendors will always provide a detailed quote.</li> <li>4. Changed Drop-Off to Pickup.</li> <li>5. Removed the &lt;RentalDuration&gt; element.</li> <li>6. Added an element for Promotion/Coupon Number.</li> <li>7. Reworded the text for Driver and Young Driver count.</li> <li>8. Added an element for Younger Driver.</li> <li>9. Added Arrival Time to the Flight Arrival element.</li> <li>10. Added two additional Payment types: Voucher, Direct Bill.</li> <li>11. Added a reference number attribute to the payment attribute.</li> <li>12. Dropped Date of Birth and Added Age to the Customer element.</li> <li>13. Added Country Code to the Customer element.</li> <li>14. Added 'Request' to the list of values for Availability Indicator.</li> <li>15. Rearranged the Vehicle Type element and added guarantee indicator.</li> <li>16. Added a element for Estimated Total Amount.</li> <li>17. Rehashed the Charge element and renamed to Rates.</li> <li>18. Reorganized the Response elements according to their logical meaning.</li> </ol>
April 10, 2001	0.3	<ol style="list-style-type: none"> <li>1. Restructured OTA_VehAvailRateRQ element.</li> <li>2. Restructured OTA_VehAvailRateRS element.</li> <li>3. Incorporated the changes resulting from the teleconference dated April/4.</li> <li>4. Dropped customer profile element based on the assumption that the information is not required for availability.</li> <li>5. Renamed PaymentForm element to RentalPayment.</li> <li>6. Dropped 'Name' and 'Expiration Date' attributes from RentalPayment element.</li> <li>7. Added AirConditionPref and TransmissionPref elements.</li> </ol>
April 11, 2001	0.3a	<ol style="list-style-type: none"> <li>1. Indented elements and attributes.</li> <li>2. Synched up with Reservation Response structure.</li> </ol>
April 18, 2001	0.4	<p>Changes resulting from document review</p> <ol style="list-style-type: none"> <li>1. Reworked on Charges structure.</li> <li>2. Changed the tag name of Charge structure to &lt;VehRateChargeInfo&gt;.</li> <li>3. Added &lt;Customer&gt; information to the request message structure.</li> <li>4. Reworded the description of existing tags.</li> </ol>
April 19, 2001	0.5	Cosmetic changes to bring in line with Vehicle Reservation.
October 31, 2001	0.6	Updated.

December 11, 2001	<p>Added element RentalPeriod under element LengthOfRental to Allow for multiple units to be used, for example, a rental may be 2 days and 1 hour.</p> <p>Clarified that VendorPref contains an element CompanyName.</p> <p>Renamed element VEHICLE to VEHICLEPREF in the request.</p>
December 14, 2001	<p>Added two more values for attribute VehRateType: AdditionalDriver, YoungDriver. Corrected misspelling of AdditionalDay.</p> <p>Renamed attribute CardIdentifier to CreditCardCode to conform to what has been defined for Profile.</p>
December 20, 2001	<p>Added an attribute LocationName as part of element PickupInfo and element ReturnInfo, to allow the name of the rental facility to be provided.</p> <p>Changed element AmountInclusive within element VehCharge to an attribute to conform to recommendation of use of attributes rather than elements when value is one of a limited set of values.</p>
January 18, 2002	<p>Moved “Success” and “Warnings” elements to start of Response Messages</p>

# **Open Travel Alliance**

## **Vehicle Reservation**

### **Request/Response Message Specifications**

## 2.9 Vehicle Reservation Messages

The OTA version 2001B specification of Vehicle Reservation provides a request/response pair of messages to support the functionality of completing a reservation of a rental vehicle. A request using the OTA\_VehResRQ will provide specific information on the rental locations and the required vehicle. The OTA\_VehResRS will, if successful, return a confirmation number along with specific rate information, vehicle type, description, restrictions, and policies. It is assumed that prior to requesting the booking, the calling system has already identified the system to which the request will be sent. It is not required that a OTA\_VehAvailRateRQ message have been completed prior to using the OTA\_VehResRQ message, if all information that is required for the reservation is already known. The request message assumes the 'pull' model, meaning that the querying system will initiate the transaction and expect an immediate response from the pre-determined supplier system.

The syntax of the action verbs that are the root elements of a payload document used to exchange vehicle availability data are enumerated below:

**OTA\_VehResRQ** - Requests a reservation of a specific vehicle type by providing criteria that shall include required information like vehicle type, pickup location and date/time, return location date/time. User preferences may also be indicated in the request. All the elements and attributes are optional, unless otherwise stated as required. Note that only one vehicle reservation may be requested in the OTA\_VehResRQ message. Multiple vehicle reservations require the sending of multiple OTA\_VehResRQ messages.

**OTA\_VehResRS** - Returns a confirmation number if successful, along with information about the vehicle that has been booked. The message shall include rates. If the rates are available then the various types of rates will be supplied, for example: Hourly, Daily, Weekly, Weekend and Monthly. The response message may include “Warnings” from business processing rules or errors if the request did not succeed. In case the rates are not available, the response will contain zero rates.

**Note:**

All the elements are indicated in **Bold** and are enclosed between < and >.

All the attributes are indicated in *italics* and are bulleted items.

## 2.10 Vehicle Reservation Request Message (OTA\_VehResRQ)

The root tag of <OTA\_VehResRQ> contains the standard payload attributes found in all OTA payload documents. The root tag also contains *ReqRespVersion* to indicate the version number requested for the response message.

<OTA\_VehResRQ> - This is the main Request element.

- *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as indicated below:
  - *EchoToken*- a sequence number for additional message identification assigned by the requesting host system. When a request system includes an *EchoToken*, the corresponding response message must include an *EchoToken* with an identical value.
  - *TimeStamp* - indicates the creation date and time of the message in UTC using the following format specified by ISO 8601:YYYY-MM-DDThh: mm: ssZ with time values using the 24-hour(military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
  - *Version*- String value that indicates the version of the message.
  - *Target* - indicates whether the message is a test message or a production message with the default value of Production. Valid values: ( Test | Production ).
  - *SequenceNmbr* – A sequence number generated by the sending system that allows for tracking of message receipt or retrieval of specific message(s) for purposes of synchronization between systems.
- *ReqRespVersion* – An indication of the version of the response message that is required.

<POS> - Point of Sale Identification. Identification number of the vendor that has made the vehicle reservation request and agency number assigned by IATA, ARC, ESRP or TID.

Refer to the separate OTA documentation of the Point of Sale element for full details.

<VehRequest> - Contains elements and attributes that form a Vehicle request.

<CountryOfResidence> - Country of residence identified by ISO country code.

- *CountryCode* - ISO Country code.

<PickupInfo> - Identifies the location from where the customer will rent the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single pickup location. The location code is decided between the vendors.
- *DateTime* - Indicates the pickup date and time when the customer would pick up the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh: mm: ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<ReturnInfo>** - Identifies the location to where the customer will return the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single return location. The location code is decided between the vendors.
- *DateTime* - Indicates the return date and time when the customer would return the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh: mm: ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<ImmediateConf>** - Indicates if the booking should only be completed if it will result in a confirmation number that can be returned in the response. The requestor may not wish to complete a reservation if the status becomes OnRequest.

- *Required* - Valid values: (Yes | No)

**<RateQualifier>** - Indicates the type of rate to be used when completing the reservation.

- *TravelType* - Indicates purpose of travel for which the customer will rent the vehicle. Valid values: (Business | Personal).
- *ShopRateType* - Indicates the type of rates the customer is shopping for. Valid values: (Business | Leisure | All).
- *CorpDiscountNmbr* - Corporate discount number.
- *PromoCoupon* - Promotion/Coupon number honored by the vendor.
- *RateCode* - The rate code to be used in the reservation.
- *RateType* – Information on the type of rate. Valid values: (Daily | Weekly | Monthly | Weekend | Unknown)



**<Driver>** - Counts of different types of drivers to be included on the rental. All the attributes of this element are optional.

- *AgeQualifyingCode* - A code representing a business rule that determines the charges for a driver based upon an age range. This age qualifying code is an attribute that allows a company to categorize drivers by age, but does not limit the business logic to the three categories of the additional attributes Adult, Young Driver, Younger Driver. Each business may have a different age or way of categorizing, thus pricing what is charged for the driver. For example, senior driver, young driver as second driver accompanied by an adult, etc. Use of this attribute allows greater flexibility than establishing the three fixed categories.

- *Count* - A positive valued integer to indicate the number of drivers based on the age range indicated.
- *Adult* - A positive valued integer to indicate the number of people above 25 years of age that will be driving the vehicle.
- *YoungDriver* - A positive valued integer to indicate the number of people between 21 and 24 years of age that will be driving the vehicle
- *YoungerDriver* - A positive valued integer to indicate the number of people between 18 and 20 years of age that will be driving the vehicle.

**<LuggageCount>** - Indicates the anticipated number of pieces of baggage the customer will be carrying in the vehicle. This is represented by a positive integer value.

**<PassengerCount>** - Number of passengers (excluding driver) for which the rental will be used.

**<FlightArrivalDetails>** - Details of the customer's arrival flight.

- *ArrivalLocation* - Arrival Airport Id. Should be a valid airport code.
- *MarketingAirline* - Identifies the Airline Company. The company name is unique amongst the airline vendors.
- *FlightNumber* - Flight number that the customer will arrive by.
- *ArrivalDateTime* - Indicates the tentative date and time of customer flight. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh: mm: ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.

**<VehRentalPref>** - Various preferences requested for the rental. Some of the preferences may override the Customer profile preferences. Vehicle type is required.

- *SmokingInd*- Indicates the customer's preference for smoking. Valid values: (Yes | No).
- *GasPrePayInd* - Indicates customer's preference of option to pre-pay for gasoline with rental. Valid values: (Yes | No).

**<VendorPref>** - Indicates the preferred Vendor Company for car rental. If a company name is supplied, the reservation is to be created with the specific Vendor Company. The company name is unique amongst the vendors.

- *PreferLevel* - Indicates the preference level for the vendor. Valid values: (Only | Unacceptable | Preferred). Default is Preferred.

**<CompanyName>** - the unique name identifying the company.

- *CompanyCode* – identifies the company using a company code
- *CodeContext*- identifies the context of the identifying code.

**<RentalPayment>** - provides information on the form of payment to be used for the reservation.

- *PaymentType* - Type of payment that will be used at the time of booking the reservation. Valid values: (CreditCard | DebitCard | BusinessAccount | CentralBill | DirectBill | Voucher | Cash | Check).
- *BillingId* - Card number/Account number associated with the payment type.
- *RefNumber* - Reference number to be used by the supplier for billing purposes. Example: Purchase Order Number.
- *CreditCardCode* – Code indicating the type of credit card code (e.g., MasterCard, AmericanExpress, Visa, etc.), or the type of central bill / debit card.
- *ExpirationDate* - Date on which the credit card or debit card expires
- *CardHolder* - Name on the credit or debit card

**<Coverage>** - Insurance Coverage needed for the rental.

- *CoverageType* - Categories of insurance preferred by the customer. Valid values: (VehDamage | Baggage | PersonalEffects | Liability | Theft | YoungDriver)
- *CoverageCode* - Industry code for type of insurance coverage.

**<SpecialReq>** - Special needs or requirements requested by the customer. This is

a repeating element. This element may be used to address special needs not addressed elsewhere within the message definition.

<**VehiclePref**> - Describe the details of the vehicle that is preferred.

<**VehicleTypePref**> - Major categories of vehicles.

- *PreferLevel* Valid values: (Only)
- *VehicleType* Valid values: (Car | Van | SUV | Convertible | Truck | Motorcycle | Limo | StationWagon | Pickup | MotorHome | Other). This is a required element.
- *VehicleDoorCount* This is an optional attribute.

<**VehicleClassPref**> - Details of vehicle types.

- *PreferLevel* Valid values: (Only)
- *VehicleClass* Valid values: (Mini | Subcompact | Economy, Compact | Midsize | Intermediate | Standard | Full\_Size | Luxury | Premium | Minivan | 12\_Passenger\_Van | Moving\_Van | 15\_Passenger\_Van | Cargo\_Van | 12\_Foot\_Truck | 15\_Foot\_Truck | 20\_Foot\_Truck | 24\_Foot\_Truck | 26\_Foot\_Truck | Moped | Stretch | Regular | Unique | Exotic | Other). This is a required element.

<**VehicleMakeModelCodePref**> - indication of the specific make and model to be reserved, if supported by the vendor.

- *PreferLevel* Valid values: (Only | Preferred)

<**AirConditionPref**> - Indicates the customer's preference for Air Conditioning.

- *PreferLevel* - Indicates the preference level. Valid values: (Only | Unacceptable | Preferred ).

<**TransmissionPref**> - Indicates customer's preference for Vehicle Transmission.

- *Type* - Indicates the type of transmission. Valid values: (Automatic | Manual).
- *PreferLevel* - Indicates the preference level. Valid values: (Only | Unacceptable | Preferred ).

<**SpecialEquipment**>- Indicates the preference for a specific item of additional equipment.

- *Quantity*- Number of this type of equipment that is requested.
- *EquipType* - Indicates the type of equipment that is preferred to be a part of this reservation Valid values: (CellularPhone | BikeRack | LuggageRack | SkiRack | TrailerHitch | AutomaticLocks | InfantSeat | ChildToddlerSeat | BoosterSeat | SnowChains | HandControlRight | HandControlLeft | NavigationalSystem | SnowTires | BabyStroller | DVDPlayer | VideoMonitor | SpinnerKnob | FurniturePads | CarDolly | AutoTransport | HandTruck | CargoBarrierFront | CargoBarrierRear | LuggageTrailer | CampingEquipment). This is required.

<**OnlineNews**> - Indicates whether the customer would prefer online news (if available) by the vendor.

- *OnlineNewsInd* - Indicates the choice for online news. Valid values: (Yes | No).

<**Customer**> - Information on the person renting the vehicle.

- *BirthDate*
- *Currency*

<**ProfileRef**> - a reference to a profile for this customer, if such a profile is available.

<**PersonName**> - Full name of the customer renting the vehicle.

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *NameType*

<**NamePrefix**>- Salutation or honorific (e.g., Mr., Mrs., Ms., Miss, Dr., etc.).

<**GivenName**>- Given name, first name or names.

<**MiddleName**>- Middle name or names.

<**Surname**> - Family name/Last name

<**NameSuffix**>- Suffix that follows the surname (e.g., Jr., Sr., III, Ret., Esq., etc.).

<**NameTitle**> - Degree or honors, (e.g.: Ph.D., M.D., etc.) often used with a person's name.

**<TelephoneInfo>** - Telephone number(s) that the customer may be contacted at.

**<Telephone>**

- *PhoneTechType*
- *CountryAccessCode*
- *AreaCityCode*
- *PhoneNumber*
- *Extension*
- *PIN*

**<Email>** - Electronic mail address, in IETF format.

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *EmailType*

**<AddressInfo>** - Information about the physical address of the customer

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *AddressType*

**<Address>**

**<StreetNbr>**

- *PO\_Box*

**<BldgRoom>**

**<AddressLine>**

- *ParsedInd* Valid values: ( Yes | No )

**<CityName>**

- *PostalCode*

**<StateProv>**

- *StateCode*

**<CountryName>**

- *CountryCode*

**<CustLoyalty>** - Loyalty programs in which the customer is enrolled. Some loyalty programs may be affiliated with the vehicle rental vendor.

- *ProgramCode*
- *Privacy*
- *MembershipId*
- *SingleVendorId* Valid values: ( SingleVndr | Alliance )
- *LoyalLevel*
- *SignupDate*
- *EffectiveDate*
- *ExpireDate*

**<Document>** - Information on documents associated with this person, for example, driver license.

- *Privacy*
- *DocId*
- *DocType*
- *Gender*
- *BirthDate*
- *EffectiveDate*
- *ExpireDate*

**<DocHolderName>**

**<DocIssueAuthority>**

**<DocIssueLocation>**

**<DocIssueLimitations>**

## 2.11 Vehicle Reservation Response Message ( OTA\_VehResRS )

The root tag of <OTA\_VehResRS> uses the external entity that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. . The message shall include rates if the reservation was completed successfully. If the rates are available then the various types of rate information will be supplied. For example: Hourly, Daily, Weekly, Weekend and Monthly. The response message may include Warnings from business processing rules or Errors if the request did not succeed. In case the rates are not available, the response will contain zero rates.

<OTA\_VehResRS> - Main Response element.

- *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as indicated below:
- *EchoToken*- a sequence number for additional message identification assigned by the requesting host system. When a request system includes an *EchoToken*, the corresponding response message must include an *EchoToken* with an identical value.
- *TimeStamp* - indicates the creation date and time of the message in UTC using the following format specified by ISO 8601:YYYY-MM-DDThh: mm: ssZ with time values using the 24-hour(military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *Version*- String value that indicates the version of the message.
- *Target* - indicates whether the message is a test message or a production message with the default value of Production. Valid values: ( Test | Production ).
- *SequenceNmbr* – A sequence number generated by the sending system that allows for tracking of message receipt or retrieval of specific message(s) for purposes of synchronization between systems.

<Success>

<Warnings>

<Warning>

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code*> - The code assigned to the warning
- *DocURL*

- *Status*
- *Tag*

**<POS>** - Point of Sale Identification. Identification number of the vendor that has made the vehicle availability request and agency number assigned by IATA, ARC, ESRP or TID.

Refer to the separate OTA documentation of the Point of Sale element for full details.

**<VendorVehicleInfo>** This element is a child element of the vehicle reservation response message. It contains information about the individual vehicle that has been reserved. The VendorVehicleInfo shall be a single-occurrence element, reflecting the vehicle that has been successfully reserved.

**<Vendor>** Indicates the Vendor Company Name associated with this specific reservation. The company name is unique amongst the vendors.

**<ConfirmationNumber>** - a unique reference number that is associated with the newly-created reservation.

**<Mileage>** - Indicates whether mileage is expressed in Miles or Kilometers.  
Valid values: (MILE | KM).

**<PickupInfo>** - Identifies the location from where the customer will rent the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single pickup location. The location code is decided between the vendors.
- *LocationName* – Provides the name of the location, this is useful when the code does not easily identify the location, such as an off-airport location.
- *DateTime* - Indicates the pickup date and time when the customer would pick up the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<VehicleLocation>** – Descriptive information about where the vehicles are parked.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<CounterLocation>** – Descriptive information about where the rental counter is located.



- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<ReturnInfo>** - Identifies the location to where the customer will return the vehicle. This is required.

- *LocationCode* - Uniquely identifies a single return location. The location code is decided between the vendors.
- *LocationName* – Provides the name of the location, this is useful when the code does not easily identify the location, such as an off-airport location.
- *DateTime* - Indicates the return date and time when the customer would return the vehicle. Time is specified in UTC using the following format specified by ISO 8601:YYY-MM-DDThh: mm: ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *LocationType* – Information about the type of location.  
Valid values: (Airport | NonAirport )

**<VehicleLocation>** – Descriptive information about where the vehicles are parked.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<CounterLocation>** – Descriptive information about where the rental counter is located.

- *Location* – Valid values: (Terminal | ShuttleOnAirport | ShuttleOffAirport | Other)

**<LengthOfRental>** - Rental duration. The duration may be expressed in Days, Hours, etc., or a combination of Days and Hours.

**<RentalPeriod>** This element can be used to describe individual rental periods. The element can repeat as needed, for example, it would occur twice to indicate a length of rental of 2 days 1 hour.

- *Duration* - A positive integer value that indicates the length of this rental period.
- *Unit* - Qualifies the duration. Valid values: (Hour | Day | Week | Month).

**<VehAvailRate>** - Vehicle Availability Rate Information. This element is used to indicate the availability and rates associated with the reserved vehicle.

- *AvailabilityInd* - Indicates the status of the vehicle associated with the reservation. Valid values: (Confirmed | OnRequest)

<**Total**> - Specifies the total amounts for the reservation.

<**RateTotalAmount**> - The total base cost for the vehicle reservation.  
This amount excludes all the taxes and surcharges.

- *Amount*
- *CurrencyCode* - the currency code associated with the costs.

<**EstimatedTotalAmount**> - The total estimate of rates that include Base Rates, Taxes, Surcharges and other associated rental items.

- *Amount*
- *CurrencyCode* - the currency code associated with the costs.

<**Vehicle**>

- *AirConditionInd* - An indication if air conditioning is available in this vehicle. Valid values: (Yes | No)
- *TransmissionType* - An indication of the transmission type of this vehicle. Valid values: (Automatic | Manual)
- *GuaranteeInd* - Indicates that a guaranteed form of payment was used to reserve this vehicle.

<**VehicleType**> - Major categories of vehicles.

- *VehicleType* Valid values: (Car | Van | SUV | Convertible | Truck | Motorcycle | Limo | StationWagon | Pickup | MotorHome | Other). This is required.
- *VehicleDoorCount* This is an optional attribute.

<**VehicleClass**> - Details of vehicle types.

- *VehicleClass* Valid values: (Mini | Subcompact | Economy, Compact | Midsize | Intermediate | Standard | Full\_Size | Luxury | Premium | Minivan | 12\_Passenger\_Van | Moving\_Van | 15\_Passenger\_Van | Cargo\_Van | 12\_Foot\_Truck | 15\_Foot\_Truck | 20\_Foot\_Truck | 24\_Foot\_Truck | 26\_Foot\_Truck | Moped | Stretch | Regular | Unique | Exotic | Other).

<**VehicleMakeModel**> - a description of the vehicle associated with this type and class.

- *Code* - A code describing the make and model of this vehicle.

**<RateQualifier>** - Indicates the type of rate that was used when completing the reservation. Data associated with this rate is also presented.

- *TravelType* - Indicates purpose of travel for which the customer is renting the vehicle. Valid values: (Business | Personal).
- *ShopRateType* - Indicates the type of rate associated with the reservation. Valid values: (Business | Leisure | All).
- *CorpDiscountNmbr* - Corporate discount number applied to the reservation.
- *PromoCoupon* - Promotion/Coupon number that has been applied to the reservation.
- *RateCode* - The rate code applied to the reservation
- *RateType* - Information on the type of rate. Valid values: (Daily | Weekly | Monthly | Weekend | Unknown)

**<PromoDesc>** - A description of the promotion that has been applied to the reservation.

**<RateMileage>** - Information on the mileage that is included with this rate.

- *Unlimited* - Valid values: (Yes | No)
- *Quantity* - number of miles or kilometers if not unlimited
- *VehUnitName* (RentalPeriod | Year | Month | Week | Day | Hour)

**<VehRateChargeInfo>** - Charges and rates associated with the reservation. This element may repeat for each charge, fee, surcharge, etc, that is to be represented.

- *VehRateType* - Information on the type of charge, that is, with what is the charge associated? Valid values: (VehicleRental | Drop | Extra | Discount | Insurance | Surcharge | Fee | Tax | AdditionalMileage | AdditionalWeek | AdditionalHour | AdditionalDay | AdditionalDriver | YoungDriver | Fuel | Other )

**<VehCharge>** - This element provides details on the specific charge, including unit cost, quantity and total cost.

- *Total* - the total cost of this charge.

- *CurrencyCode* - the currency code associated with the costs.
- *UnitCharge* - the cost per unit of this charge. The unit has been identified in VehUnitName.
- *VehUnitName* - Information on the unit for which this charge is expressed. Valid values: (RentalPeriod | Year | Month | Week | Day | Hour | Mile | Km | Gallon | Liter )
- *UnitQuantity* - the number of units on which the charge is based.
- *Percentage* - a numerical value if the charge is based on a percentage rather than on a unit cost.
- *MaxCharge* - the maximum charge that will apply, for example, a child seat may have a charge of \$5.00 per day with a maximum charge of \$20.00
- *MinCharge* - the minimum charge that will apply, for example, a VehicleRental charge may be 9.99 per day, but requires a two-day minimum rental. Minimum charge would be 19.98.
- *Taxable* - An indication if this charge is taxable. Valid values: (Yes | No)
- *AmountInclusive* - An indication if this charge has been included in the base rates. Valid values: (Yes | No)

**<Description>** - A description of the charge, for example, "Airport Surcharge".

**<TaxAmounts>** - Tax charges that are applied to the rate. Note that this object allows for both percentages and currency amounts.

**<TaxAmount>** - provides the amount of the tax

- *Total* - the amount of this tax
- *TaxCode* - Code identifying the tax (e.g., occupancy tax, city tax, etc.).
- *CurrencyCode* - Code of the currency associated with this tax.
- *Percentage* - the tax percentage

**<Description>** - description of the tax.

**<InternetId>** - Vendor's Internet address.

**<VendorMessages>** - allows the vendor to present informational messages about the reservation that has been completed. These may include policy information messages, messages providing special instructions relative to vehicle collection or return, or general marketing messages.

- *Type* - identifies the category of message.  
Valid values: (Policy | Marketing | SpecialInstructions | Other)

- *Title* - The header of the message.

**<Paragraph>** - An indication of a new paragraph for this section of the message. This element will repeat for each paragraph in this message.

**<Text>** - The text describing the message. This element would be used if the paragraph can be re-formatted to fit the available display space.

**<Line>** A line of information that has been formatted by the vendor and which should be shown in a fixed width font, without reformatting. This element may repeat for each line of formatted text.

**<SpecialEquipment>** - A collection of Special Equipment items that have been associated with the reservation, along with their corresponding charges.

- *Restrictions* - An indication of there are any restrictions associated with this equipment and the type of rental. Valid values: (OneWayOnly | RoundTripOnly | AnyReservation)
- *Quantity* - Number of this type of equipment that was requested.
- *EquipType* - Indicates the type of equipment that is preferred to be a part of this reservation Valid values: (CellularPhone | BikeRack | LuggageRack | SkiRack | TrailerHitch | AutomaticLocks | InfantSeat | ChildToddlerSeat | BoosterSeat | SnowChains | HandControlRight | HandControlLeft | NavigationalSystem | SnowTires | BabyStroller | DVDPlayer | VideoMonitor | SpinnerKnob | FurniturePads | CarDolly | AutoTransport | HandTruck | CargoBarrierFront | CargoBarrierRear | LuggageTrailer | CampingEquipment). This is required.

**<VehCharge>** - This element provides details on the specific charge, including unit cost, quantity and total cost.

- *Total* - the total cost of this charge.
- *CurrencyCode* - the currency code associated with the costs.

- *UnitCharge* - the cost per unit of this equipment. The unit has been identified in VehUnitName.
- *VehUnitName* - Information on the unit for which this equipment charge is expressed. Valid values: (RentalPeriod | Year | Month | Week | Day | Hour | Mile | Km | Gallon | Liter )
- *UnitQuantity* - the number of equipment units on which the charge is based.
- *Percentage* - a numerical value if the charge is based on a percentage rather than on a unit cost.
- *MaxCharge* - the maximum charge that will apply, for example, a child seat may have a charge of \$5.00 per day with a maximum charge of \$20.00
- *MinCharge* - the minimum charge that will apply, for example, a child seat may have a charge of \$5.00 per day with a minimum charge of \$10.00.
- *Taxable* - An indication if this equipment charge is taxable. Valid values: (Yes | No)
- *AmountInclusive* - An indication if this charge has been included in the base rates. Valid values: (Yes | No)

**<Description>** - A description of the equipment charge, for example, "Child Safety Seat per day rate"

**<TaxAmounts>** - Tax charges that are applied to the rate. Note that this object allows for both percentages and currency amounts.

**<TaxAmount>** - provides the amount of the tax

- *Total* - the amount of this tax
- *TaxCode* - Code identifying the tax (e.g., occupancy tax, city tax, etc.).
- *CurrencyCode* - Code of the currency associated with this tax.
- *Percentage* - the tax percentage

**<Description>** - description of the tax.

**<Coverages>** - Information on Insurance Coverage that has been included in the reservation.

**<Coverage>** - An indication of one type of coverage. This element may repeat for as many coverages as available.

- *CoverageType* - The category of coverage. Valid values: (VehDamage | Baggage | PersonalEffects | Liability | Theft | YoungDriver)
- *CoverageCode* - Industry code for type of insurance coverage.
- *CoverageCost* - Cost for this type of insurance coverage.
- *CurrencyCode* - Currency in which the cost is quoted.

**<CoverSupplement>** - Supplemental information

**<CoverDescription>** - Description of the coverage

**<CoverLimits>** - Limits of the coverage

**<VehLocationDetails>** - Information on a vehicle rental location. This element may be repeated to provide information on both pickup and return locations.

- *Code* - a unique code assigned to the location
- *Name* the full name of the location
- *LocationType* - Valid values: (OnAirport | OffAirport | NonAirport)

**<OperationSchedules>**

**<OperationSchedule>** - this element provides information on one schedule, it may repeat for as many schedules that exist.

**<DateTimeSpan>** - defines a period of time for which several operation times exist.

**<OperationTimes>** - defines one or more operation times within this date and time pan.

**<OperationTime>**

**<DateTimeSpan>**

- *Context*
- *startInstant*
- *duration*

**<DOWPattern>** - Indicates the days of the week on which this date and time span apply.

- *Mon* Valid values: (Yes | No)
- *Tues* Valid values: (Yes | No)
- *Wed* Valid values: (Yes | No)
- *Thurs* Valid values: (Yes | No)
- *Fri* Valid values: (Yes | No)
- *Sat* Valid values: (Yes | No)
- *Sun* Valid values: (Yes | No)

**<LocationServices>**

- *Shuttle* - Valid values: (Yes | No)

**<AddressInfo>** - Information about the physical address of the location

- *DefaultInd* Valid values: (Yes | No )
- *Privacy*
- *AddressType*

**<Address>**

**<StreetNbr>**

- *PO\_Box*

**<BldgRoom>**

**<AddressLine>**

- *ParsedInd* Valid values: ( Yes | No )

**<CityName>**

- *PostalCode*

**<StateProv>**

- *StateCode*



**<CountryName>**

- *CountryCode*

**<VehLocationInfo>** - allows the vendor to present informational messages about the rental location. These may include after-hour messages, messages providing directions to the location, or other similar messages.

- *Type* - identifies the category of message.  
Valid values: (AfterHours | Directions | Other)

- *Title* - The header of the message..

**<Paragraph>** - An indication of a new paragraph for this section of the message. This element will repeat for each paragraph in this message.

**<Text>** - The text describing the message. This element would be used if the paragraph can be re-formatted to fit the available display space.

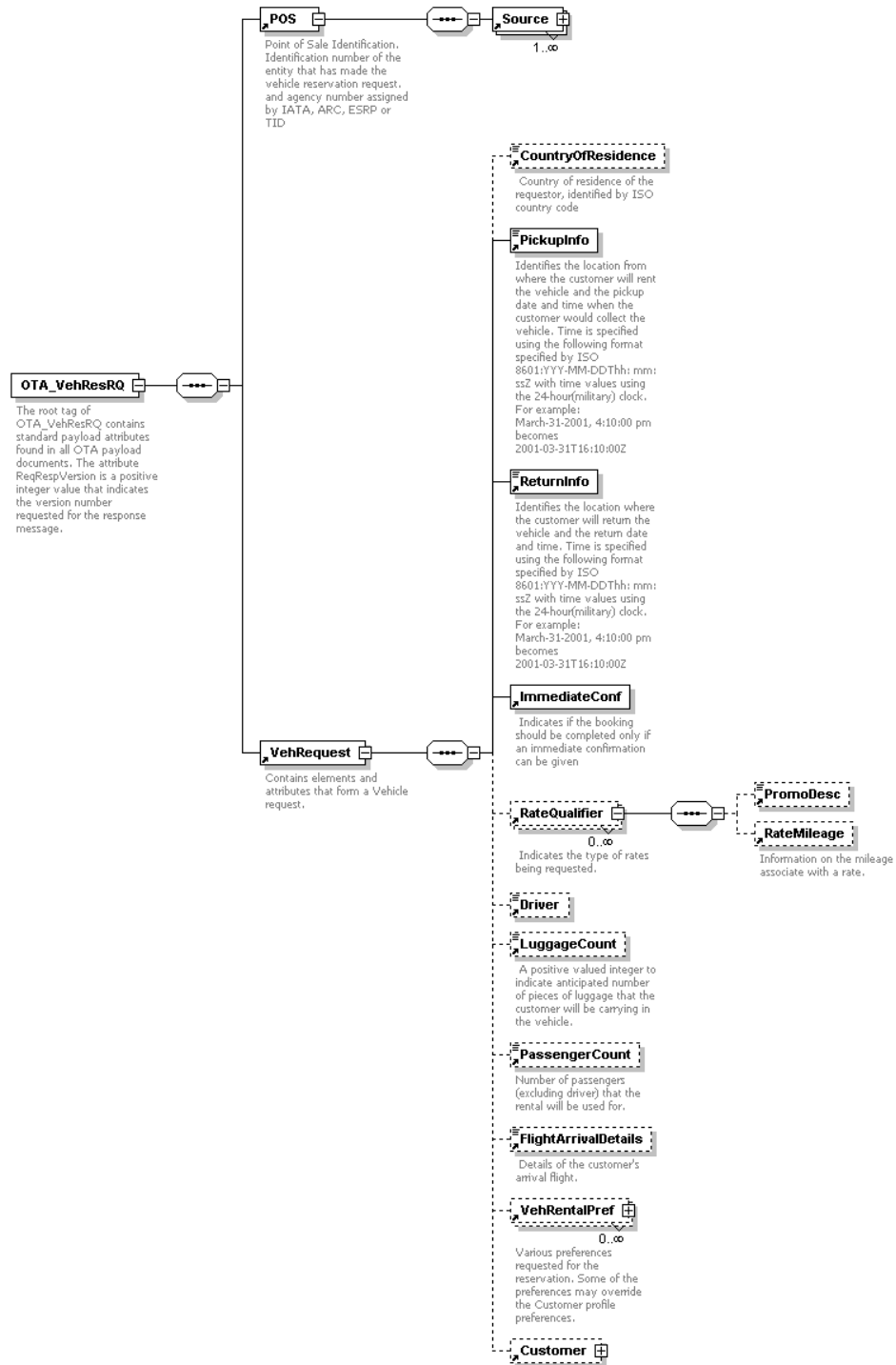
**<Line>** A line of information that has been formatted by the vendor and which should be shown in a fixed width font, without reformatting. This element may repeat for each line of formatted text.

**<Errors>****<Error>**

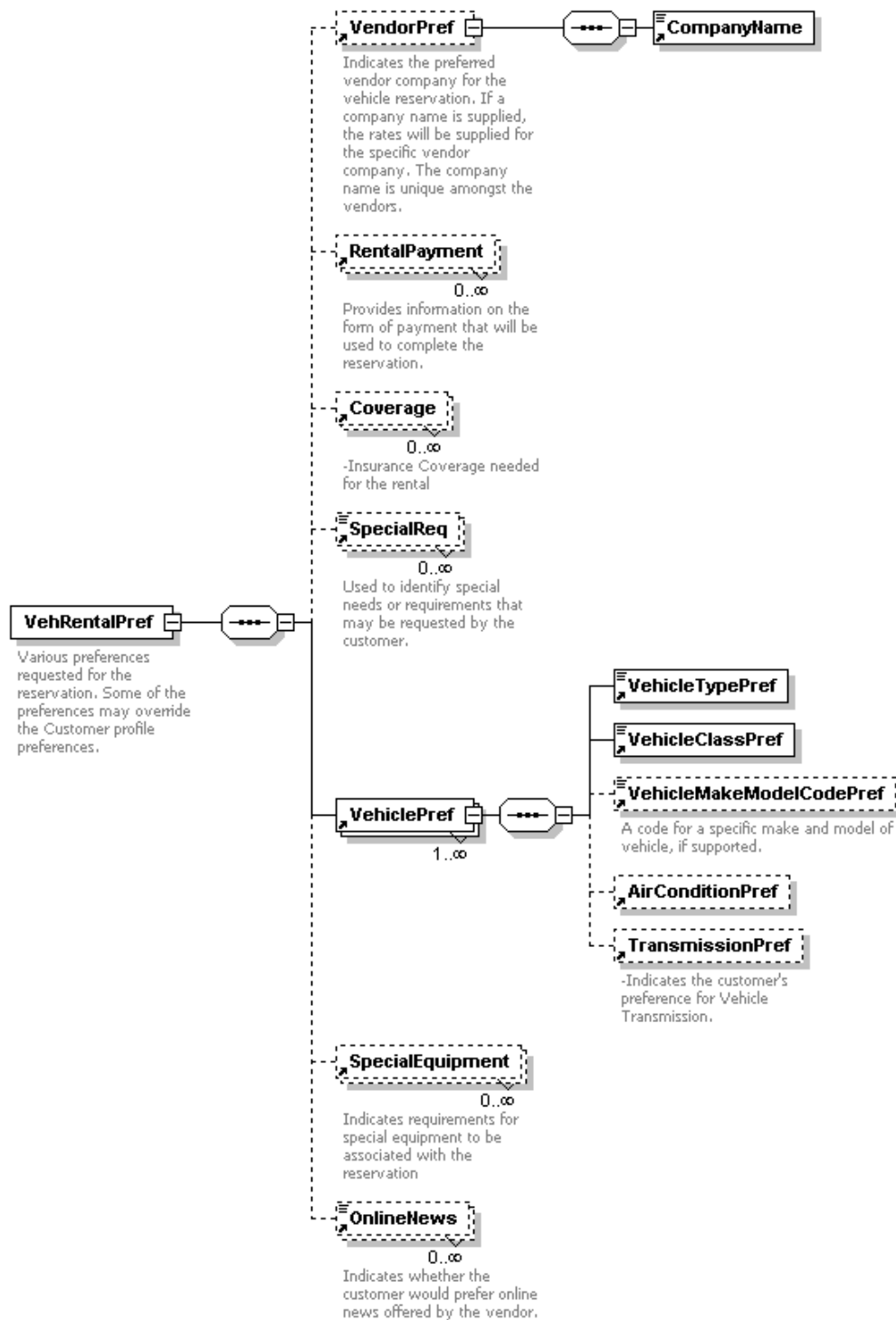
- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the error
- *DocURL*
- *Status*
- *Tag*

## 2.12 XML Schema Diagrams for Vehicle Reservation Request

### OTA\_VehResRQ



## VehRentalPref



## 2.13 Sample XML Message for Vehicle Reservation Request

This sample XML request illustrates a request to create a reservation for a Compact car at IAD airport, renting from May 5<sup>th</sup>, 2001 to May 15<sup>th</sup>, 2001, for a 10 day rental period. Information on the renter and the form of payment is provided. The request includes information on special equipment, specifically two Child Seats and a Ski Rack.

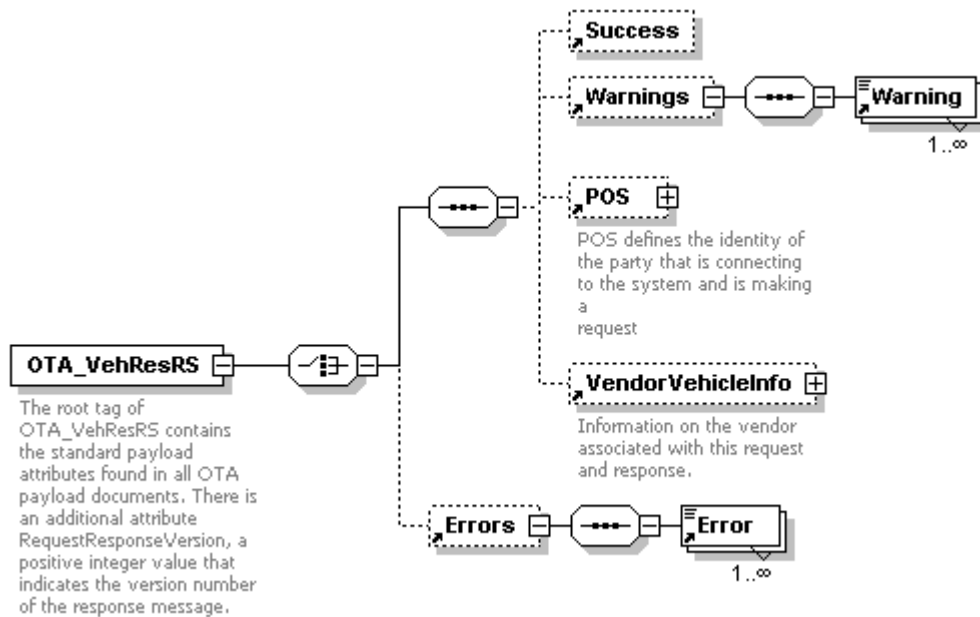
The example POS element reflects that this is a request for a corporate traveler, and that the request originated from a website. Information on both the corporation and the associated travel agency are provided.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_VehResRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opentravel.org/OTA"
xsi:noNamespaceSchemaLocation="OTA_VehResRQ.xsd" EchoToken="18723222" TimeStamp="2001-03-
31T16:12:10Z" Target="Production" Version="1" SequenceNbr="1568">
  <POS>
    <Source>
      <Uniqueld Type="Corporate" Id="ID06"/>
      <BookingChannel Type="ADS" Primary="true"/>
    </Source>
    <Source>
      <Uniqueld Type="ARC" Id="ID07"/>
    </Source>
  </POS>
  <VehRequest>
    <CountryOfResidence CountryCode="US"/>
    <PickupInfo LocationCode="IAD" DateTime="2001-05-05T10:00:00Z"/>
    <ReturnInfo LocationCode="IAD" DateTime="2001-05-15T10:00:00Z"/>
    <ImmediateConf Required="Yes"/>
    <RateQualifier TravelType="Personal" CorpDiscountNbr="56432" PromoCoupon="B45"
RateCode="7S" ShopRateType="All"/>
    <Driver AgeQualifyingCode="SENIOR" Count="1" Adult="1" YoungDriver="0" YoungerDriver="0"/>
    <LuggageCount>3</LuggageCount>
    <PassengerCount>4</PassengerCount>
    <FlightArrivalDetails ArrivalLocation="IAD" MarketingAirline="AA" FlightNumber="50"
ArrivalDateTime="2001-05-05T09:00:00Z"/>
    <VehRentalPref SmokingInd="No" GasPrePayInd="No">
      <VendorPref PreferLevel="Preferred">
        <CompanyName>XYZ</CompanyName>
      </VendorPref>
      <RentalPayment PaymentType="CreditCard" Billing_Id="540000000000045"
CardHolder="John Smith" ExpirationDate="2002-06" CreditCardCode="VI"/>
      <Coverage CoverageCode="LIA" CoverageType="Liability"/>
      <SpecialReq>Require a non-smoking car</SpecialReq>
    </VehRentalPref>
    <VehiclePref>
      <VehicleTypePref PreferLevel="Only" VehicleType="Car"/>
      <VehicleClassPref PreferLevel="Only" VehicleClass="Compact"/>
      <VehicleMakeModelCodePref
PreferLevel="Preferred">FE1</VehicleMakeModelCodePref>
      <AirConditionPref PreferLevel="Preferred"/>
      <TransmissionPref PreferLevel="Only" Type="Automatic"/>
    </VehiclePref>
    <SpecialEquipment Quantity="2" EquipType="DVDPlayer"/>
    <SpecialEquipment Quantity="1" EquipType="SkiRack"/>
    <OnlineNews>No</OnlineNews>
  </VehRequest>
  <Customer>
    <PersonName>
      <NamePrefix>Mr</NamePrefix>
      <GivenName>John</GivenName>
      <Surname>Smith</Surname>
      <NameTitle>M.D.</NameTitle>
    </PersonName>
    <Email EmailType="Personal">johnsmith@home.com</Email>
```

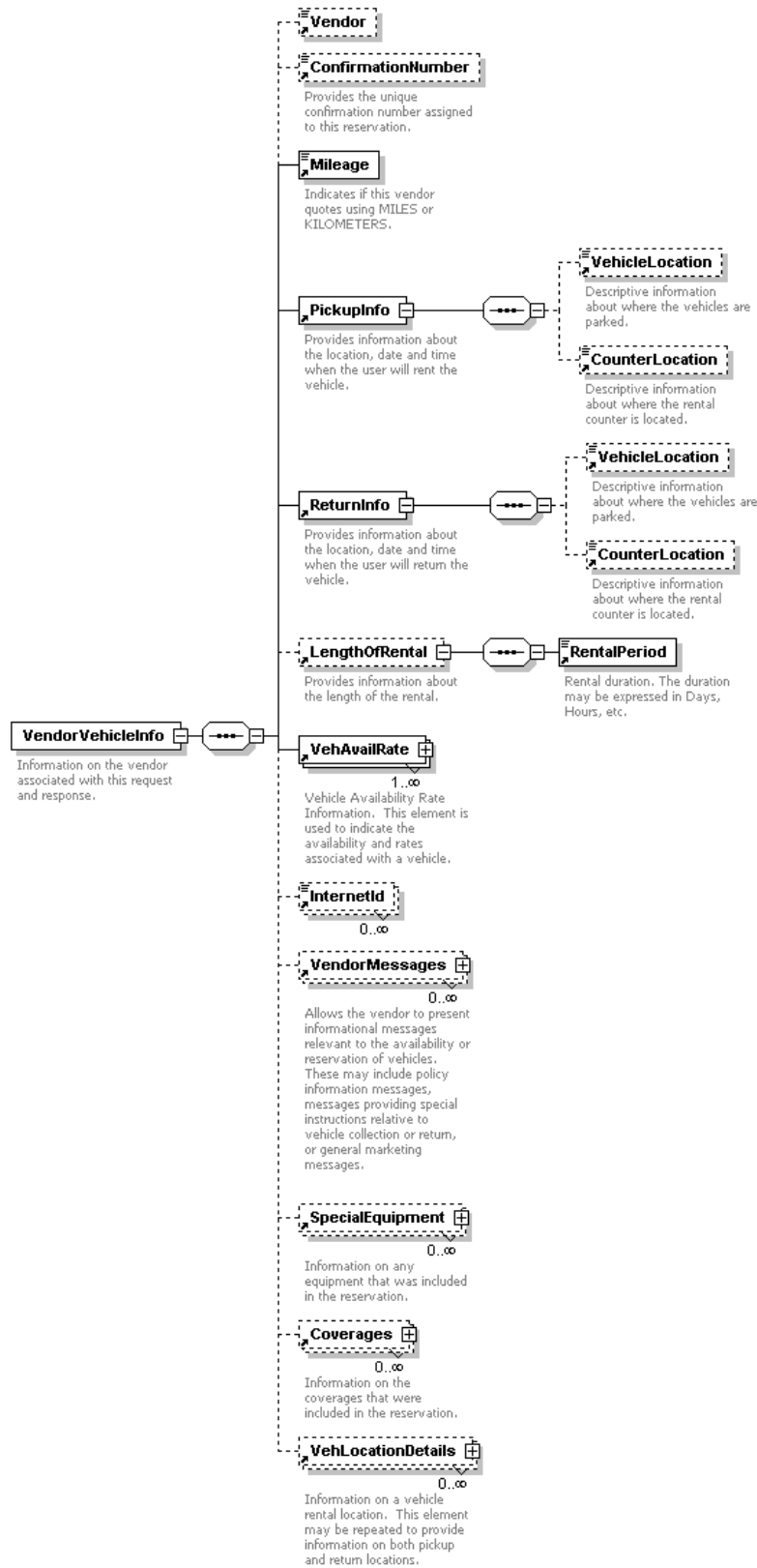
```
        <AddressInfo>
          <Address>
            <AddressLine>12 Main Street</AddressLine>
            <CityName PostalCode="75000">Dallas</CityName>
            <StateProv StateCode="TX"/>
            <CountryName CountryCode="US"/>
          </Address>
        </AddressInfo>
        <CustLoyalty MembershipId="Q56GTF" ProgramCode="QuickRent"/>
      </Customer>
    </VehRequest>
  </OTA_VehResRQ>
```

## 2.14 XML Schema Diagrams for Vehicle Reservation Response

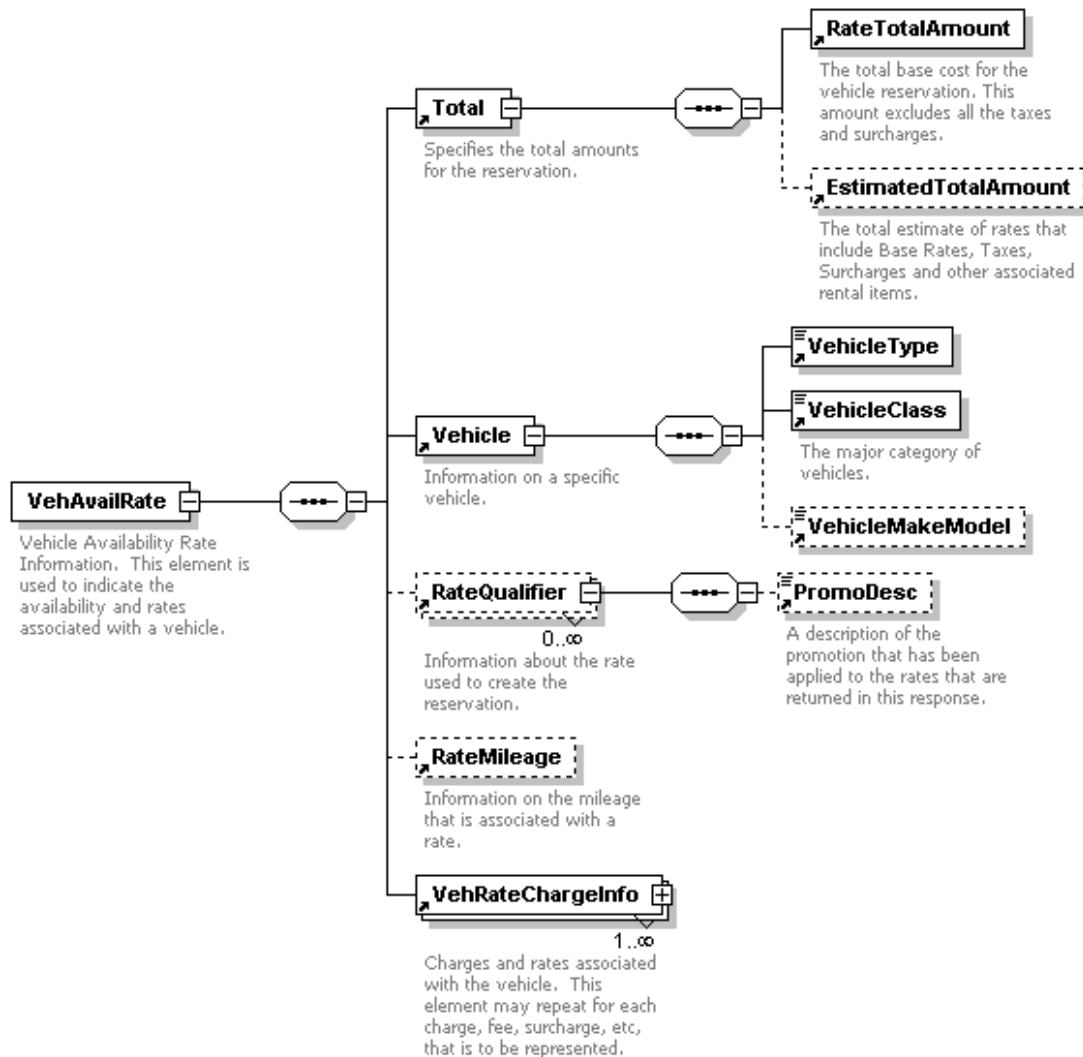
### OTA\_VehResRS



## VendorVehicleInfo

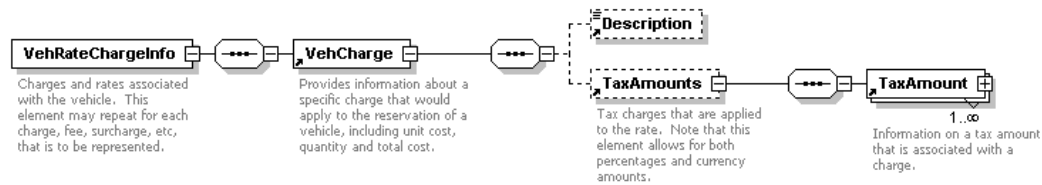


## VehAvailRate

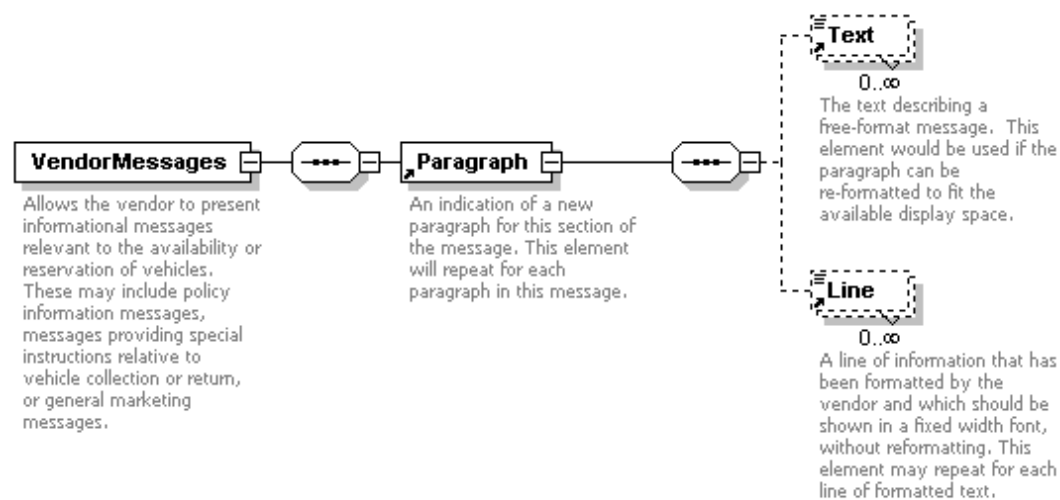




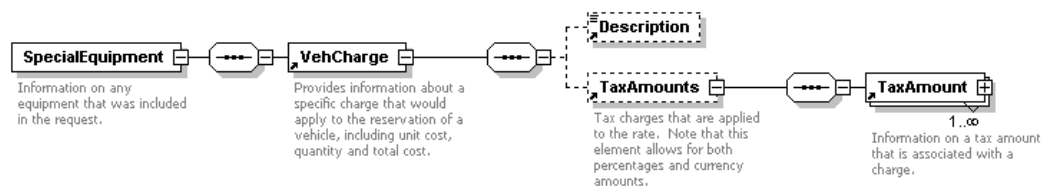
## VehRateChargeInfo



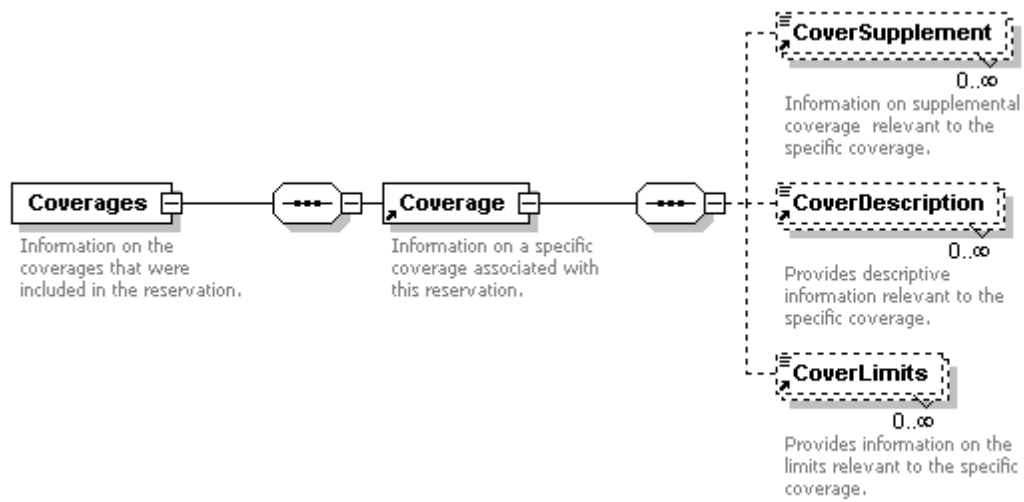
## VendorMessages



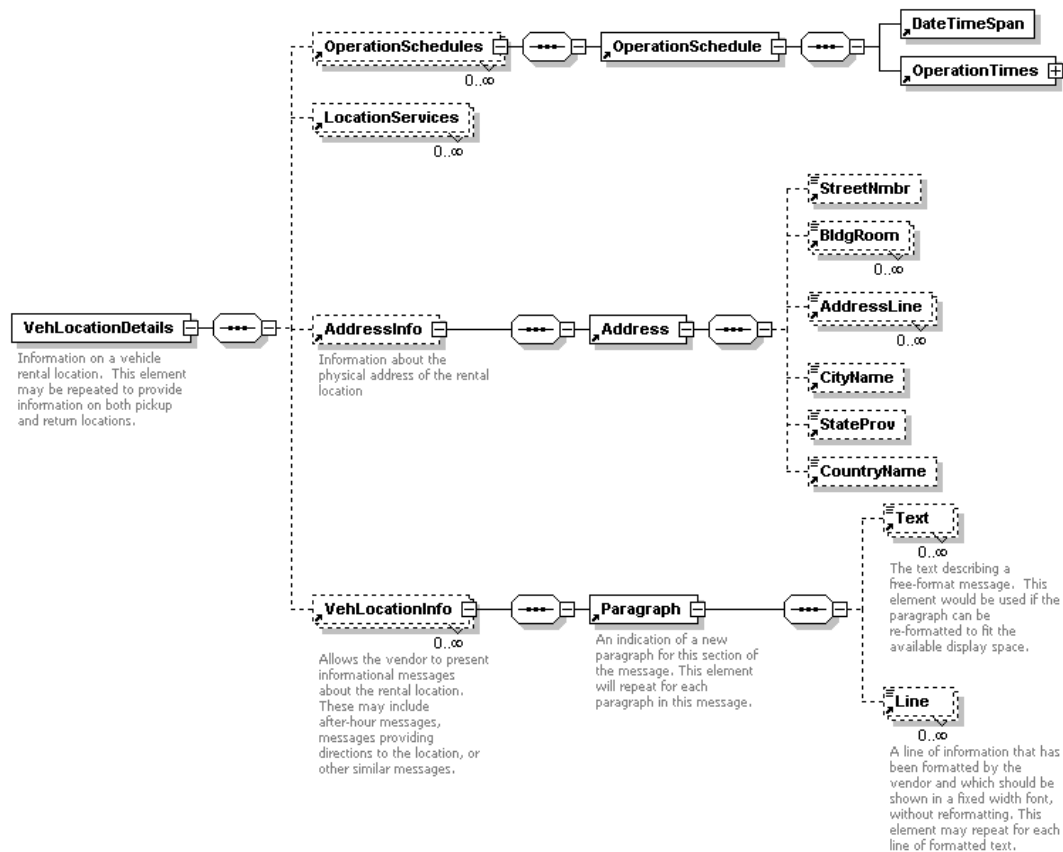
## SpecialEquipment



## Coverages



## VehLocationDetails



## OperationTimes



## 2.15 Sample XML Message for Vehicle Reservation Response

This sample XML request illustrates the confirmation of a reservation for a Compact car at IAD airport, renting from May 5<sup>th</sup>, 2001 to May 15<sup>th</sup>, 2001, for a 10 day rental period. Information on the rates and taxes associated with this reservation are provided, along with additional information, such as the Coverages selected for the rental period and information on how to return the vehicle, etc.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_VehResRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://www.opentravel.org/OTA"
xsi:noNamespaceSchemaLocation="OTA_VehResRS.xsd" EchoToken="18723222" TimeStamp="2001-03-31T16:13:08Z"
" Target="Production" Version="1" SequenceNmbr="1569">
  <Success/>
  <POS>
    <Source>
      <Uniqueld Type="Corporate" Id="ID08"/>
      <BookingChannel Type="ADS" Primary="true"/>
    </Source>
    <Source>
      <Uniqueld Type="ARC" Id="ID09"/>
    </Source>
  </POS>
  <VendorVehicleInfo>
    <Vendor>XYZ</Vendor>
    <ConfirmationNumber>R12128733</ConfirmationNumber>
    <Mileage>MILES</Mileage>
    <PickupInfo LocationCode="IAD" DateTime="2001-05-05T10:00:00Z" LocationType="Airport">
      <VehicleLocation Location="ShuttleOnAirport">Shared Rental Parking Lot,
South</VehicleLocation>
      <CounterLocation Location="Other">Baggage Claim Level 2</CounterLocation>
    </PickupInfo>
    <ReturnInfo LocationCode="IAD" DateTime="2001-05-15T10:00:00Z" LocationType="Airport">
      <VehicleLocation Location="ShuttleOnAirport">Shared Rental Parking Lot,
South</VehicleLocation>
    </ReturnInfo>
    <LengthOfRental>
      <RentalPeriod Duration="10" Unit="Day"/>
    </LengthOfRental>
    <VehAvailRate AvailabilityInd="Confirmed">
      <Total>
        <RateTotalAmount Amount="310.99" CurrencyCode="USD"/>
        <EstimatedTotalAmount Amount="510.99" CurrencyCode="USD"/>
      </Total>
      <Vehicle AirConditionInd="Yes" TransmissionType="Automatic" GuaranteeInd="Yes">
        <VehicleType VehicleType="Car"/>
        <VehicleClass VehicleClass="Compact"/>
        <VehicleMakeModel Code="FE1">Ford Escort</VehicleMakeModel>
      </Vehicle>
      <RateQualifier TravelType="Personal" CorpDiscountNmbr="56432" PromoCoupon="B45"
RateCode="7S" ShopRateType="All" RateType="Weekly">
        <PromoDesc>Free Upgrade at Time of Rental</PromoDesc>
      </RateQualifier>
      <RateMileage Unlimited="No" Quantity="200" VehUnitName="Day"/>
      <VehRateChargeInfo VehRateType="VehicleRental">
        <VehCharge Total="310.99" CurrencyCode="USD" UnitCharge="31.10"
VehUnitName="Day" UnitQuantity="10" Taxable="Yes" AmountInclusive="Yes">
          <Description>Vehicle Rental Charge</Description>
          <TaxAmounts>
            <TaxAmount Total="5.00" CurrencyCode="USD"
TaxCode="STT">
              <Description>State Tax</Description>
            </TaxAmount>
            <TaxAmount Total="2.00" CurrencyCode="USD"
TaxCode="CTY">
              <Description>City Tax</Description>
            </TaxAmount>
          </TaxAmounts>
        </VehCharge>
      </VehRateChargeInfo>
    </RateMileage>
  </VehAvailRate>
</OTA_VehResRS>
```

```

        </TaxAmounts>
        </VehCharge>
    </VehRateChargeInfo>
    <VehRateChargeInfo VehRateType="AdditionalMileage">
        <VehCharge CurrencyCode="USD" UnitCharge="0.05" VehUnitName="Mile"
Taxable="Yes" AmountInclusive="No">
            <Description>Excess Mileage Fee</Description>
            <TaxAmounts>
                <TaxAmount CurrencyCode="USD" Percentage="8"
TaxCode="STT">
                    <Description>State Tax</Description>
                    </TaxAmount>
                <TaxAmount CurrencyCode="USD" Percentage="4"
TaxCode="CTY">
                    <Description>City Tax</Description>
                    </TaxAmount>
            </TaxAmounts>
        </VehCharge>
    </VehRateChargeInfo>
    </VehAvailRate>
    <InternetId>WWW.XYZ-RENT-A-CAR.COM</InternetId>
    <VendorMessages Type="SpecialInstructions" Title="Car Pickup">
        <Paragraph>
            <Text>Use Airport Monorail</Text>
        </Paragraph>
    </VendorMessages>
    <SpecialEquipment Restrictions="RoundTripOnly" EquipType="ChildToddlerSeat">
        <VehCharge Total="10.00" CurrencyCode="USD" UnitCharge="2.50" VehUnitName="Day"
UnitQuantity="10" MaxCharge="10.00" MinCharge="5.00" Taxable="No" AmountInclusive="No">
            <Description>Cost for Rental of Child Seat</Description>
        </VehCharge>
    </SpecialEquipment>
    <Coverages>
        <Coverage CoverageType="VehDamage" CoverageCode="LIA" CoverageCost="10.33"
CurrencyCode="USD">
            <CoverSupplement/>
            <CoverDescription>All damage except first 100.00</CoverDescription>
            <CoverLimits/>
        </Coverage>
    </Coverages>
    <VehLocationDetails Code="IAD" Name="Houston Airport" Type="OnAirport">
        <LocationServices Shuttle="Yes"/>
        <VehLocationInfo Type="Directions" Title="Driving Directions">
            <Paragraph>
                <Text>Return Vehicle to Rental Location by following signs to Rental
Car Return</Text>
            </Paragraph>
        </VehLocationInfo>
    </VehLocationDetails>
    </VendorVehicleInfo>
</OTA_VehResRS>

```

## 2.16 VehResRQ/RS Revision History

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
April 06, 2001	0.1	Initial Draft.
April 09, 2001	0.1a	Added Schema diagrams.
April 10, 2001	0.1b	Updated Schema diagrams.
April 11, 2001	0.2	<p>Changes resulting from the telephone conference.</p> <ol style="list-style-type: none"> <li>1. Changed document references from BOOK to RESERVATION</li> <li>2. Changes throughout to correspond to Vehicle Avail Rates.</li> <li>3. Provided COVERAGE in a wrapper COVERAGES, and expanded this to match Profile element.</li> <li>4. Allowed for a monetary discount amount to be represented.</li> <li>5. Added Station Details</li> <li>6. Changed AvailabilityReq to ImmediateConfirmation</li> <li>7. Updated Customer to match that in Profile.</li> <li>8. Added PromoApplied, reflecting a code and a description.</li> <li>9. Allowed for Marketing Messages to be passed by vendor</li> <li>10. Updated Success/Warning/Error to match OTA guidelines.</li> </ol>
April 16, 2001	0.3	<p>Changes resulting from the Vehicle Avail Rate review</p> <ol style="list-style-type: none"> <li>1. Corrected indentation of elements</li> <li>2. Added DTD, Schema and Sample Messages</li> </ol>
April 18, 2001	0.4	<p>Changes resulting from the document review</p> <ol style="list-style-type: none"> <li>1. Reworked Charges to VehRateChargeInfo</li> <li>2. Reworked Equipment element</li> <li>3. Added RateMileage to RateQualifier</li> <li>4. Added VehicleLocationDetails</li> </ol>
April 19, 2001	0.5	Cosmetic changes to bring in line with Vehicle Availability / Rate.
October 31, 2001	0.6	Updated.
December 11, 2001		<p>Added element RentalPeriod under element LengthOfRental to allow for multiple units to be used, for example, a rental may be 2 days and 1 hour.</p> <p>Clarified that VendorPref contains an element CompanyName.</p> <p>Renamed element VEHICLE to VEHICLEPREF in the</p>

request.

December 14, 2001

Added two more values for attribute VehRateType: AdditionalDriver, YoungDriver. Corrected misspelling of AdditionalDay.

Renamed attribute CardIdentifier to CreditCardCode to conform to what has been defined for Profile.

December 20, 2001

Added an attribute LocationName as part of element PickupInfo and element ReturnInfo, to allow the name of the rental facility to be provided.

Changed element AmountInclusive within element VehCharge to an attribute to conform to recommendation of use of attributes rather than elements when value is one of a limited set of values.

January 18, 2002

Moved “Success” and “Warnings” elements to start of Response Messages

# **OpenTravel Alliance Message Specification**

## **Publication 2001B**

### **Section 3 - Airline Booking Specifications**



### 3.0 Introduction

The OTA Air Work Group has defined a basic airline booking request and response message for these 2001B specifications. As booking a reservation with an airline can run the gamut from the most simple, straightforward transaction to a much more complex iterative message conversation, this specification focuses solely on the basic data elements needed in a simple booking message.

The Request and Response messages defined here draw from the work done by other OTA industry vertical working groups. OTA\_AirBookRQ and OTA\_AirBookRS are defined in this section.

### 3.1 Limited Scope

This specification excludes any of the more complicated transactions, such as booking seat assignments, special meals, special service requests, etc. It also excludes any fare information from the messages. Future OTA specifications may include these options as well as ticketing, gate information, fare structures, and the ability to return different flight options if the flights requested for booking are not available.

### 3.2 Point of Sale information

The first determination that must be made upon receipt of a request to book a reservation is to identify the party that is making the request. The requestor may be a known trading partner with whom the airline has an agreement, such as a travel agent or company, or it may be a customer accessing the airline system through a web site making a direct booking in to the airline system. The provider of the Internet portal is also a party to the reservation transaction and may offer the customer special pricing based on a negotiated rate with the airlines. The identification of the requesting party is of prime importance since it may affect the ability to access exclusive rates or to book a reservation.

The point-of-sale data, contained in the <POS> element, communicates this information and allows the receiving system to identify the trading partner that is making the request for the booking. The <POS> element and sub-elements used by the airline booking request message have been taken directly from the object prepared for this (2001B) specification.<sup>10</sup>

### 3.3 POS elements

The basic elements of the POS object include:

- <**UniqueId**> - Four attributes identify the unique number assigned to the party from their profile. The *Id* is the unique IATA/ATA, TID or ERSP number. The *Type* =Profile, and the *URL* identifies the certifying body, such as IATA, ERSP, etc. The specific URL used may include a location for validation of certification. An optional *Instance* attribute indicates the date of the last information about that party.
- <**Position**> - An element that identifies the geographic position of the server that sends the request. The Position element uses the representation defined by ISO Standard 6709 to define a geographic point location.
- <**BookingChannel**> - Identifies the type of channel used for connectivity that the request is coming through. Some systems require the identification of a channel type or switch system used to reach the supplier in order to honor a request to book a reservation.

---

<sup>10</sup> See Section 1 - Introduction to OTA Messages for a complete explanation of the Point-of-Sale object.

### 3.4 GDS attributes

Since booking requests may arrive at the airline supplier having passed through a Global Distribution System (GDS), two attributes are found on the POS element to accommodate those systems:

- *AgentSine* - Identifies the machine of the in-house travel agent making the request.
- *PseudoCityCode* - Identifies the location or city code of the office making the request.

### 3.5 Additional Airline requirements

The OTA Air Working Group has identified additional attributes needed in the point-of-sale information in order to complete an Airline booking. These attributes have been extracted from the traditional IATA definitions, commonly known as ORG elements.

These include:

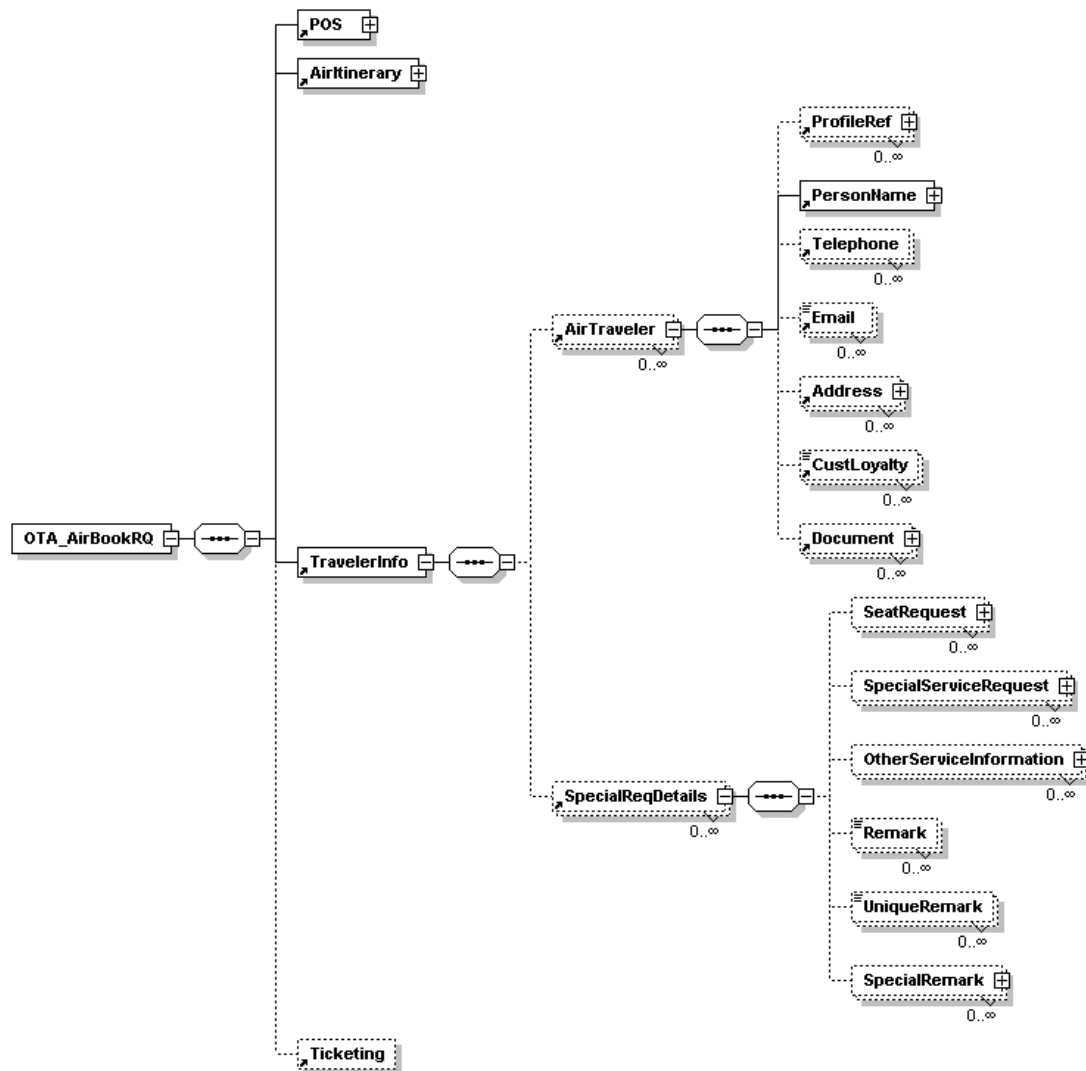
- *ISOCountry* - Identifies the country of origin of the system making the request.
- *ISOCurrency* - Identifies the currency to be used for payment of the ticket.
- *AgentDutyCode* - Identifies the duty code of the agent.
- *CarrierCode* - Identifies the airline or CRS system making the request.
- *FirstDepartPoint* - Identifies the first departure point of the itinerary, by airport code.
- *AirportCode* - Identifies the airport or city code of the system delivering the message.
- *ERSPUserId*

### 3.6 Air Booking Request message

An airline booking <OTA\_AirBookRQ> message is sent to request a reservation on an airline for a flight itinerary, specifying the origin and destination of the traveler. It also contains information about the passenger who will travel on the reservation, and may include a reference to a profile of that passenger or related party, such as the person purchasing the ticket, that is stored on a data base. The booking request also carries minimum information about ticketing required to complete the booking transaction.

Exclusions in bookings, such as avoiding propeller aircraft or connections, may be a part of requesting a reservation, however, items such as this fall under the category of enhancements envisioned for the next version, currently projected for inclusion in OTA publication 2001C.

#### 3.6.1 Diagram of the AirBook Request



### 3.6.2 Air Itinerary

The flight itinerary may be a one-way flight, a round trip, or consist of several segments, or flight legs, including a transfer or connection at an intermediary point along the way. The basic data elements in the request include the flight number, carrier code, flight date, origin, and destination. The request message carries only the marketing carrier, but the response message returns both the marketing and operating carrier codes.

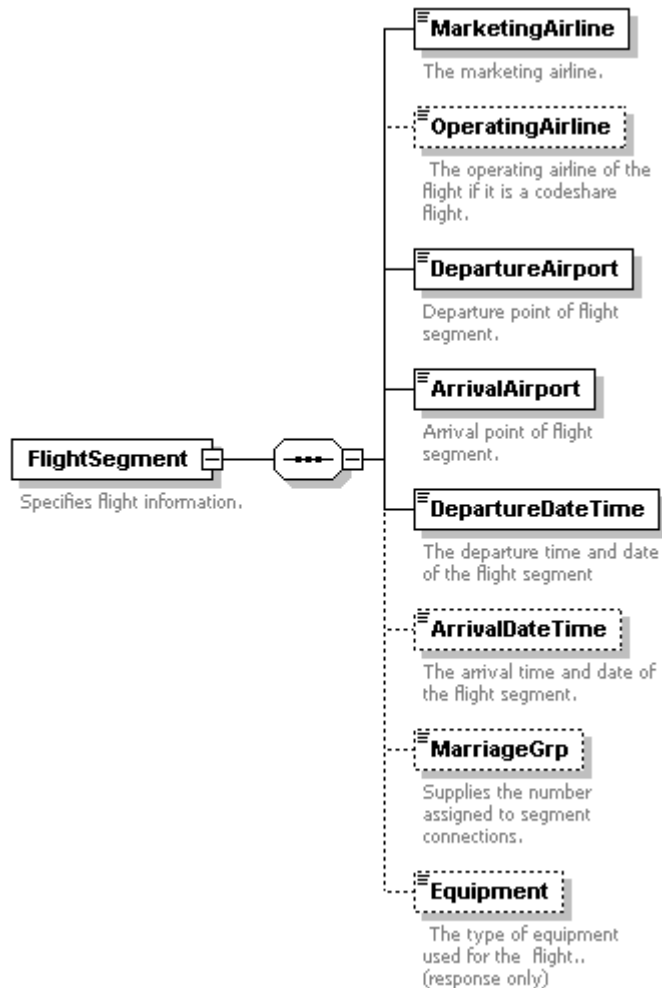
A flight number can be either numeric or alpha, with alpha codes OPEN or ARNK (arrival unknown) as recognized in the industry allowed as a valid value for these elements. Because of different variations in the use of OPEN for city pairs, flight number, and dates in bookings, specific implementations of these values are left up to the trading partners.

Both city/airport names and codes may be used to identify the origin and destination. A city or airport name may be placed in the PCDATA of the element, with its corresponding airport code placed in the attribute of that element.

Three attributes found on the <AirItinerary> element as follows:

- *DirectionInd* - A directional indicator that identifies a type of air booking, either one-way, round-trip, or open-jaw with the enumeration of (OneWay | RT | OpenJaw) respectively.
- *ActionCode* - Indicates the status of the booking, such as OK or Wait-List.
- *NumberInParty* - Indicates the traveler count.

### 3.6.3. Diagram of FlightSegment within the AirItinerary.XSD



### 3.6.4. Customer Loyalty

The **CustLoyalty** entity in the Request and Response messages identifies information about membership in a loyalty program that includes the identification of the loyalty program name and program code, as well as the specific Membership Id of the customer. This object uses the same schema as the OTA 2001A SPECIFICATION and that information may be extracted from a profile to be added to the request if stored by the party (e.g. web site or vendor) that is requesting the booking.

### ***3.6.5. Profile Reference***

Profile information about a customer who is a frequent traveler or a subscriber to a web site who purchases travel services may be stored on a database at the sending or receiving system. That profile may contain readily available information relevant to the booking without having to pass all of the information in the reservation request message itself. The ProfileRef element identifies a profile by UniqueId, and can be used to exchange a reference to the profile and provide access to the data it contains, allowing the application processing the message to assemble the necessary profile information in order to complete the booking.

### ***3.6.6. Passenger Element***

Whenever a reservation for airline travel is booked there is a requirement to identify the passenger (or passengers) who will be travelling. A Customer Profile, attached to a reservation by a Profile Reference, could be used for payment information but not actually identify the party who is traveling; for example, a ticket purchased for travel by another party.

### ***3.6.7. Traveller Info (TIF)***

The Passenger element is used to provide identification of the person(s) who will take the flight. This uses data elements from the OTA Profile specification.

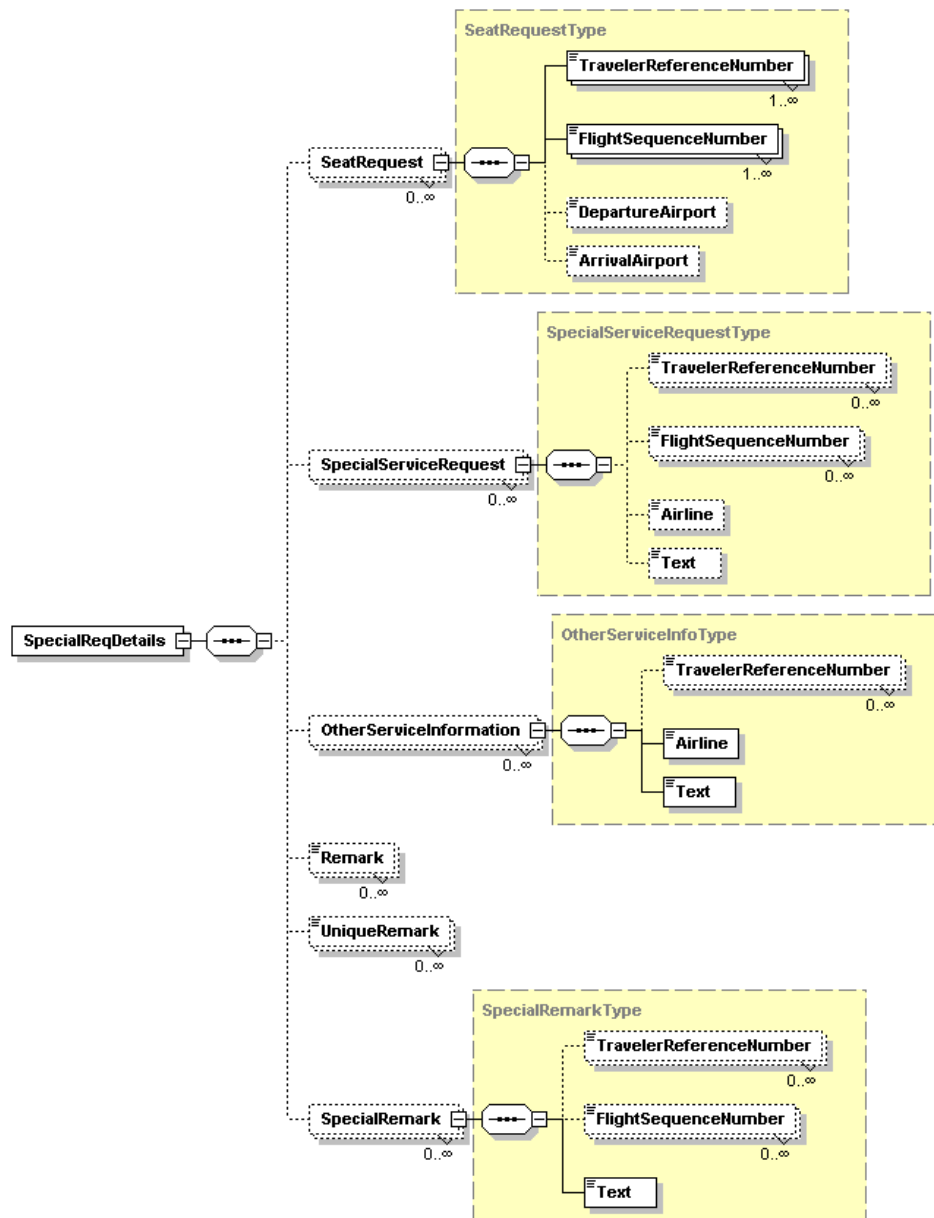
### ***3.6.8. Customer Loyalty Info***

The customer loyalty object is included in the Passenger element to identify information needed to credit travelers with frequent flier award miles. This uses the CustLoyalty data element from the OTA Profile specification.

### ***3.6.9. Special Requirements Details (SSR)***

The Special Requirements Details, (traditional SSR) is attached to the Passenger element in addition to the entity (SpecialReqDetails.XSD) that occurs at the Flight level. When Special Requirement Data appears at the Passenger level, it is used as the default for all flight segments. The data content for the Special Requirements Details is the standard SSR codes.

### ***3.6.10. Diagram of the Special Requirements Details (SSR)***



### 3.6.11. Ticketing

A Ticketing element is included in the Air Booking request and response messages to meet the minimum requirements for completing the reservation as needed for an instant ticketing situation. Ticketing is an empty element that includes two attributes that provide the following information:

- *TicketTimeLimit* - Indicates the ticketing arrangement, and allows for the requirement that an itinerary must be ticketed by a certain date and time.
- *TicketType* - Indicates the type of ticket (paper, electronic). This attribute is identical to the attribute of the same name in the OTA Profile (Airline Preferences section).

The attribute *TicketType* may be empty in a Request message. If a choice is filled in, this may indicate a request for a specific ticket type. In the response message, the data returned indicates the type of ticket issued.

### 3.7. Air Booking Response message

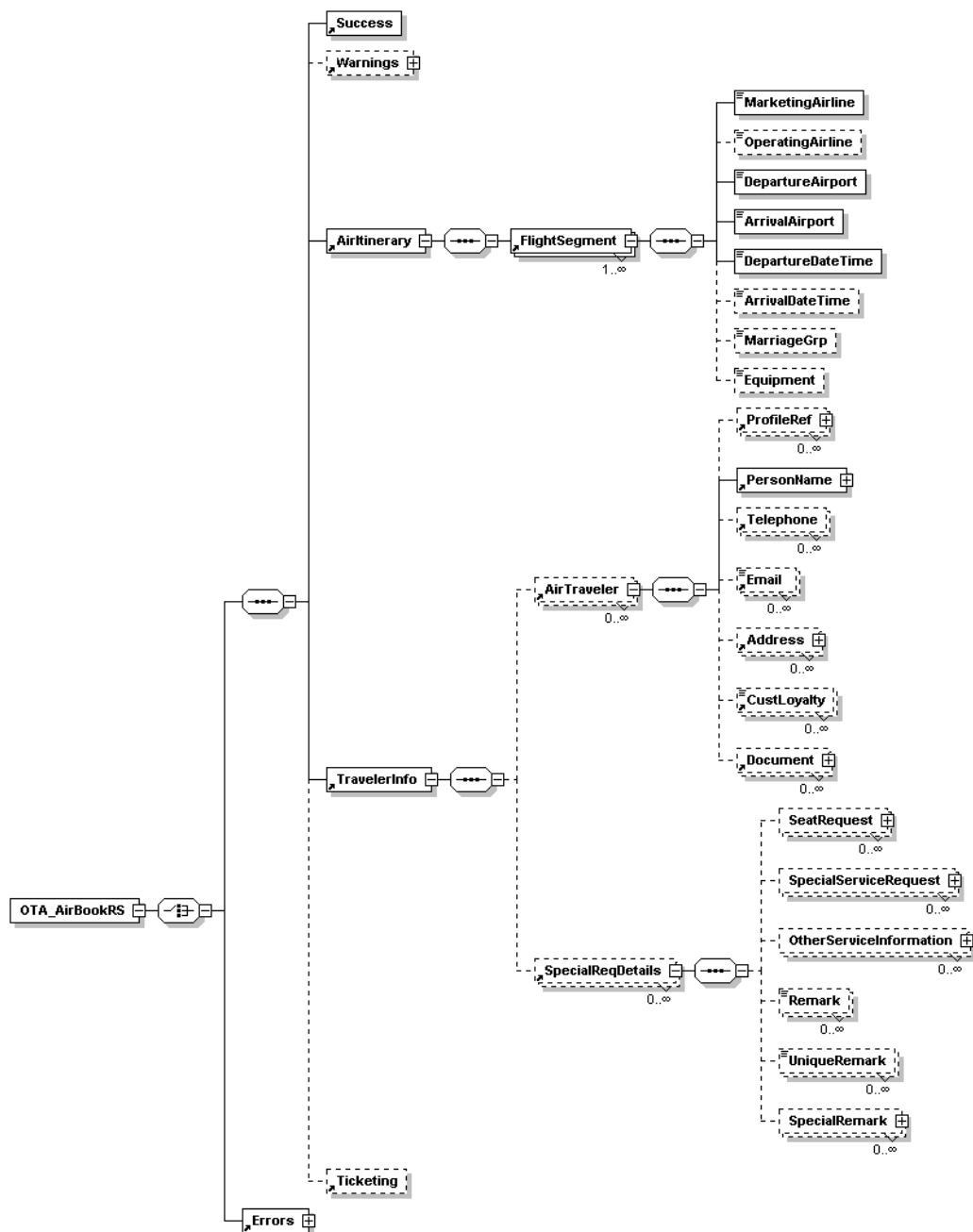
The response message, <OTA\_AirBookRS>, returns the airline reservation along with the Record Locator number and an indication of the status of the booking. The booking response replicates many of the data items sent in the request, including the itinerary, information about the passenger who will be travelling, and the minimum information about ticketing required to complete the booking transaction.

In addition to the OTA Payload Standard Attributes, two other attributes are included in the response message at the root level:

- *RecordLocator* - The identification number of the booking record.
- *Cancel* - Indicate whether this message includes a cancellation of a reservation. The data type is boolean, enumerated as (Yes | No), with the default of "No".

The response message is structured to be able to indicate a cancellation as well as a new booking. For cancellations, the message uses the record locator to identify the precise booking. The Cancel attribute is set by default to "No" and when set to "Yes" indicates that the action taken is a cancellation. The Cancel attribute may be removed from the root element tag if extending the generic OTA\_CancelRQ and RS messages.

### 3.7.1. Diagram of the AirBook Response



### 3.7.2 Air Itinerary

The flight itinerary in the response message includes the flight information, itemizing the Departure Date, Departure Time, Arrival Date, and Arrival Time, Origin Airport, and Destination Airport. All of these elements have been defined as optional rather than mandatory. Air carriers may determine that these items are mandatory between business partners.

The response message identifies both the operating and the marketing carrier. The provision for both operating and marketing carriers in this message allows for the designation of code-shared flights, where the marketing carrier is not the same as the operating carrier. Flight number



changes from marketing to operator carriers are not included and will be incorporated in later versions of the booking messages.

Four attributes found in the Itinerary element in the response message are:

- *DirectionInd* - A directional indicator that identifies a type of air booking, either one-way, round-trip, or open-jaw with the enumeration of (OneWay | RT | OpenJaw) respectively.
- *ActionCode* - Indicates the status of the booking, such as OK or Wait-List.
- *NumberInParty* - An integer field that indicates the Traveler Count.
- *ResBookDesigCode* - Identifies the reservation booking designator (RBD) and refers to the class of service. This is a free-text field rather than enumerated list.

### ***3.7.3. Flight Segments and Marriage Groups***

Each Flight element may indicate one origin-destination pair. For connections, there can be an origin and destination per segment, and the order of the Segments and Marriage Groups can be indicated.

The data type for the FlightNumber element is a string that accepts both numeric for traditional flight numbers and alpha characters to accommodate OPEN and ARNK codes. Again, due to differences among implementations in the use of OPEN in bookings, specific implementations of these text fields are left up to the individual trading partners.

Related flights include a marriage group with a marriage sequence number. A marriage group <MarriageGrp> element, supplies the number assigned to segment connections. A marriage sequence is the order of the travel legs within the marriage group, and a marriage sequence number, attribute *MarriageSeq*, indicates the number of the travel leg within that marriage group

Married segment control is the designation of two or more segments in an itinerary as a marriage, by the Member(s) providing this service. Segments once identified as married are actioned as a single unit (set) (IATA Reso. 1777).

### ***3.7.4. Passenger Element***

When the reservation for airline travel is booked the response message returns the identification of the passenger (or passengers) who will be travelling. The information returned replicates that of the request message, and includes the traditional TravellerInfo (TIF), to confirm the identity of the parties, the FrequentTravellerInfo (FTI), that identifies information needed to credit awards in frequent flier programs, and Special Requirements Details, (SSR) objects.

The SpecialRequirementsDetails, (SSR) attached to the Passenger element is used to supply the special services for all flight segments. Standard SSR codes are used to identify the special services booked. When special services are attached at the level of the Flight element, they indicate that the special service is requested for that flight only.

### ***3.7.5. Ticketing***

A Ticketing element meets the minimum requirements for completing the booking, and provides important information about the requirement to obtain a ticket within a certain time period. The following two attributes should contain data in the response message:

- *TicketTimeLimit* - Indicates the ticketing arrangement, and allows for the requirement that an itinerary must be ticketed by a certain date and time.
- *TicketType* - Indicates the type of ticket (paper, electronic) that applies to this booking.

### 3.7.6. Errors and Warnings

The <OTA\_AirBookRS> response message does not include partial bookings. For the sake of simplicity, it is an all-or-nothing response for individual segments. The response message lists the individual segments requested and indicates whether or not they were booked.

The external entity OTA\_v2ent.xsd defines a standard way for OTA messages to indicate success in the processing of a request, or to return processing errors and warnings. The entity allows for multiple errors to be passed in a response message. The ErrorAttr entity allows for the use of standardized code sets to return an error in processing.

The OTA Air Working Group has indicated the desirability of having authoritative code sets available in a repository for lookup or downloading by trading partners. This service would assist companies with staying current with the most recent versions of the codes for use in their messages.

### 3.7.7. OTA\_AirBookRQ XML Schema Example

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_AirBookRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_AirBookRQ.xsd" EchoToken="18723222" TimeStamp="2001-03-
31T16:13:08Z" Target="Production" Version="1" SequenceNbr="1569">
  <POS>
    <Source AgentSine="BSIA1234PM/GS" PseudoCityCode="2U8" ISOCountry="us"
ISOCurrency="USD" CarrierCode="AA" AirportCode="IAD" ERSPUserId="12345432">
      <Uniqueld URL="http://provider1.org/OTAEngine/" Type="Reservation"
Id="ID0507G4325" Instance="2001-06-03T13:09:21"/>
      <BookingChannel Type="GDS" Primary="1"/>
    </Source>
  </POS>
  <Airtinerary>
    <FlightSegment FlightNumber="150" SequenceNumber="1" ResBookDesigCode="Q">
      <MarketingAirline Code="AA" CodeContext="IATA"/>
      <DepartureAirport LocationCode="IAD" CodeContext="IATA"/>
      <ArrivalAirport LocationCode="SAN" CodeContext="IATA"/>
      <DepartureDateTime>2001-09-11T00:00:00</DepartureDateTime>
    </FlightSegment>
    <FlightSegment FlightNumber="151" SequenceNumber="2" ResBookDesigCode="Q">
      <MarketingAirline Code="AA" CodeContext="IATA"/>
      <DepartureAirport LocationCode="SAN" CodeContext="IATA"/>
      <ArrivalAirport LocationCode="IAD" CodeContext="IATA"/>
      <DepartureDateTime>2001-09-11T00:00:00</DepartureDateTime>
    </FlightSegment>
  </Airtinerary>
  <TravelerInfo>
    <AirTraveler Gender="Male" PassengerTypeCode="ADT">
      <ProfileRef>
        <Uniqueld Type="Profile" Id="ID12345" URL="http://www.usda.gov"
Instance="1999-02-28T09:13:53"/>
      </ProfileRef>
      <PersonName>
        <NamePrefix>Mr</NamePrefix>
        <GivenName>John</GivenName>
        <Surname>Smith</Surname>
        <NameTitle>M.D.</NameTitle>
      </PersonName>
    </AirTraveler>
  </TravelerInfo>
</OTA_AirBookRQ>
```

```

        <Telephone AreaCityCode="703" PhoneNumber="555-5555"/>
        <Email EmailType="Personal">johnsmith@home.com </Email>
        <Address>
            <AddressLine>12 Main Street</AddressLine>
            <CityName PostalCode="75000">Dallas</CityName>
            <StateProv StateCode="TX"/>
            <CountryName CountryCode="US"/>
        </Address>
        <CustLoyalty MembershipId="Q56GTF" ProgramCode="AA"/>
        <Document DocId="T056736" DocType="Passport" Gender="Male" BirthDate="1967-08-
13" EffectiveDate="1997-08-13" ExpireDate="2007-08-13">
            <DocHolderName>John Smith</DocHolderName>
            <DocIssueAuthority>US Passport Office</DocIssueAuthority>
            <DocIssueLocation>Dallas</DocIssueLocation>
        </Document>
    </AirTraveler>
    <SpecialReqDetails>
        <SeatRequest SeatNumber="14C">
            <TravelerReferenceNumber>1</TravelerReferenceNumber>
            <FlightSequenceNumber>1</FlightSequenceNumber>
        </SeatRequest>
        <SpecialServiceRequest SSRCode="WCHR">
            <TravelerReferenceNumber>1</TravelerReferenceNumber>
            <FlightSequenceNumber>1</FlightSequenceNumber>
            <Airline Code="AA" CodeContext="DUNS"/>
        </SpecialServiceRequest>
    </SpecialReqDetails>
</TravelerInfo>
<Ticketing TicketType="Electronic"/>
</OTA_AirBookRQ>

```

### 3.7.8. OTA\_AirBookRS XML Schema Example

```

<?xml version="1.0" encoding="UTF-8"?>
<!--Sample XML file generated by XML Spy v4.1 U (http://www.xmlspy.com)-->
<OTA_AirBookRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_AirBookRS.xsd" xmlns="http://www.opentravel.org/OTA"
EchoToken="18723222" TimeStamp="2001-03-31T16:13:08Z" Target="Production" Version="1" SequenceNmbr="1569"
RecordLocator="L8CD23" Cancel="No">
    <Success/>
    <Airtinerary DirectionInd="RT" ActionCode="OK">
        <FlightSegment FlightNumber="150" SequenceNumber="1" ResBookDesigCode="Q">
            <MarketingAirline Code="AA" CodeContext="IATA"/>
            <DepartureAirport LocationCode="IAD" CodeContext="IATA"/>
            <ArrivalAirport LocationCode="SAN" CodeContext="IATA"/>
            <DepartureDateTime>2001-09-11T12:00:00-05:00</DepartureDateTime>
            <ArrivalDateTime>2001-09-11T14:35:00-08:00</ArrivalDateTime>
            <Equipment AirEquipType="767"/>
        </FlightSegment>
        <FlightSegment FlightNumber="151" SequenceNumber="2" ResBookDesigCode="Q">
            <MarketingAirline Code="AA" CodeContext="IATA"/>
            <DepartureAirport LocationCode="SAN" CodeContext="IATA"/>
            <ArrivalAirport LocationCode="IAD" CodeContext="IATA"/>
            <DepartureDateTime>2001-09-15T08:35:00-08:00</DepartureDateTime>
            <ArrivalDateTime>2001-09-15T16:35:00-05:00</ArrivalDateTime>
            <Equipment AirEquipType="767"/>
        </FlightSegment>
    </Airtinerary>
    <TravelerInfo>
        <AirTraveler Gender="Male" PassengerTypeCode="ADT">
            <ProfileRef>
                <UniqueId Type="Profile" Id="ID12345" URL="http://www.usda.gov"
Instance="1999-02- 28T09:13:53"/>
            </ProfileRef>
            <PersonName>
                <NamePrefix>Mr</NamePrefix>
                <GivenName>John</GivenName>
                <Surname>Smith</Surname>
            </PersonName>
        </AirTraveler>
    </TravelerInfo>
</OTA_AirBookRS>

```

```

        <NameTitle>M.D.</NameTitle>
    </PersonName>
    <Telephone AreaCityCode="703" PhoneNumber="555-5555"/>
    <Email EmailType="Personal">johnsmith@home.com </Email>
    <Address>
        <AddressLine>12 Main Street</AddressLine>
        <CityName PostalCode="75000">Dallas</CityName>
        <StateProv StateCode="TX"/>
        <CountryName CountryCode="US"/>
    </Address>
    <CustLoyalty MembershipId="Q56GTF" ProgramCode="AA"/>
    <Document DocId="T056736" DocType="Passport" Gender="Male" BirthDate="1967-08-
13" EffectiveDate="1997-08-13" ExpireDate="2007-08-13">
        <DocHolderName>John Smith</DocHolderName>
        <DocIssueAuthority>US Passport Office</DocIssueAuthority>
        <DocIssueLocation>Dallas</DocIssueLocation>
    </Document>
    </AirTraveler>
    <SpecialReqDetails>
        <SeatRequest SeatNumber="14C">
            <TravelerReferenceNumber>1</TravelerReferenceNumber>
            <FlightSequenceNumber>1</FlightSequenceNumber>
        </SeatRequest>
        <SpecialServiceRequest SSRCode="WCHR">
            <TravelerReferenceNumber>1</TravelerReferenceNumber>
            <FlightSequenceNumber>1</FlightSequenceNumber>
            <Airline Code="AA" CodeContext="DUNS"/>
        </SpecialServiceRequest>
    </SpecialReqDetails>
    </TravelerInfo>
    <Ticketing TicketType="Electronic" TicketTimeLimit="2001-09-10T00:00:00"/>
</OTA_AirBookRS>

```

# **OpenTravel Alliance Message Specification**

## **Publication 2001B**

### **Section 4 - Travel Insurance Specifications**

## 4.0 Introduction<sup>11</sup>

Travel insurance exists to protect the traveler and the traveler's investments in the journey. Travelers usually find four reasons to invest in travel insurance.

- Travel medical insurance protects the travelers' health and safety while traveling outside of their primary medical insurance coverage area. Travel medical insurance often covers the costs of medical treatment and hospitalization in the event of an accident or illness and may provide monetary compensation to a beneficiary in the unfortunate event of the insured's death.
- Travel protection insurance protects the travelers' travel investment (money, time) in the journey. Travel protection insurance usually covers the costs of the traveler's journey and/or belongings under certain coverage situations.
- Emergency evacuation insurance helps insure the travelers' safety by transporting them out of a dangerous or medically ill-equipped region.
- Travel assistance services provide travelers with aid and resources that they might not normally have access to on their journey. Such resources may include translation services, legal services, and communication services.

Insurance plans are usually pre-set packages of benefits that may include one or more of the above types of insurance. Plan benefits and costs are pre-determined by the plan's underwriters and are usually based on traveler age, trip cost, and specific journey elements (cost, destination, etc). Depending on the plan, coverage may be purchased for the duration of a single trip or may cover multiple trips within a specified period of time.

Customers can search for and book insurance through the OTA specification. The OTA specifications dedicate a schema for travel insurance quoting and booking.

The XML Schema file contains the structure and contents of four separate messages for travel insurance:

- Quote request
- Quote response
- Booking request
- Booking response

The Insurance schema makes full use of objects in the travel profile specifications defined earlier by OTA, to maximize interoperability with those specifications. A description of these shared elements has been left out of this section, but may be found in the OTA Insurance schema. The use of data from customer profiles assumes that the customer has given permission to extract data from those files and share the data with insurance vendors.

---

<sup>11</sup> The OpenTravel Alliance wishes to thank Michael Schreiber of Insuractive for his contributions.

## 4.1 Insurance Quote Request

Unlike other travel services offered by traditional suppliers of travel products (hotels, airlines, etc.), insurance availability is not affected by a limited, or finite inventory. Instead, availability is determined by qualification factors (age of travelers, cost of trip, destination, etc.). An insurance availability search is equivalent to a request to an insurance vendor to provide a price for insurance services. The quote response returns pricing information for specific insurance plans carried by the vendor that meet the customer's requirements.

### OTA\_InsuranceQuoteRQ

The root element for the insurance quote request message is <OTA\_InsuranceQuoteRQ>. It consists of one or more <PlanForQuoteRQ> elements.

### PlanForQuoteRQ elements

The <PlanForQuoteRQ> element contains all of the customer requirements for receiving an insurance quote. The elements of the <PlanForQuoteRQ> object are:

<**InsuranceCustomer**> – an object containing data on the buyer and others covered by the requested insurance services.

<**PlanType**> – an object containing details about the insurance services requested.

<**POS**> – an object containing details for point-of-sale, in this case the immediate seller of the service that can be a broker, travel agent, automated kiosk or direct customer inquiry; this is the same point-of-sale object as used elsewhere in the OTA specifications.

The attributes of the <PlanForQuoteRQ> object are:

- *ISOCurrency* – Identifies the currency to be used for generating the quote.
- *PlanID* – Identifies the specific plan to be quoted (if known).
- *Type* - Identifies the sub-plan designation of the specific plan to be quoted (if known).

### Insurance Customer

The <InsuranceCustomer> element contains information about the individual requesting the quote/booking, the method of purchase, and information on all travelers to be covered under the insurance plan. The <InsuranceCustomer> element is re-used within the insurance booking request message. It contains many of the same identity and contact elements as used in the OTA personal profile (PersonName, Email, Address, etc.).

The elements of the <InsuranceCustomer> object are:

< **PersonName** > – Identifies the individual requesting the insurance product (same as PersonName element in personal profile).

<**Email**> – Email address of the individual requesting the insurance product (same as in personal profile).

**<Address>** – Address of individual requesting the insurance product (same as in personal profile).

**<Telephone>** – Telephone/Fax/Data line number for the individual requesting the insurance product (same as in personal profile).

**<CoveredTraveler>** – Identifies individuals requesting insurance coverage and their insurance-related information such as beneficiaries and special requirements (TravelerCoverageReqs) that can affect the quote.

**<Contact>** – Identifies contact information for the individual requesting the insurance product. (same as in personal profile).

**<Payment>** – Contains payment information for the individual requesting the insurance product. This includes the form of payment as well as the cost of the plan (if known).

**<Affiliations>** – Identifies the affiliations of the individual requesting the insurance product. These may include organization memberships, employer information, etc. (Same as personal profile).

### **CoveredTraveler**

The <CoveredTraveler> element contains information about each individual requesting to be covered by the insurance policy (identifying information as well as coverage requirements and journey information). The <CoveredTraveler> element is also used in the insurance booking response message.

The elements of the <CoveredTraveler> object are:

**<CoveredPerson>** – Identifies the individual requesting to be covered by the insurance policy.

**<Email>** – Email address(es) for the individual requesting to be covered by the insurance policy (same as in personal profile).

**<Address>** – Address information for the individual requesting to be covered by the insurance policy (same as in personal profile).

**<Telephone>** - Telephone/Fax/Data line number for the individual requesting to be covered by the insurance policy (same as in personal profile).

**<CitizenCountryName>** – Country of citizenship for the individual requesting to be covered by the insurance policy (same as in personal profile).

**<Document>**– Document information for the individual requesting to be covered by the insurance policy. This may include passport information, social security number, etc. (same as in personal profile).

**<Contact>** – Contact information for the individual requesting to be covered by the insurance policy. This may include emergency contact person (same as in personal profile).



**<Beneficiaries>**— Identifies the individuals that the traveler would like to name as beneficiaries for their insurance coverage.

**<TravelerCoverageReqs >**— Contains traveler-specific information necessary for generating an insurance quote or booking. Such information may include cost of trip, traveler's pre-existing conditions, as well as plan options unique to the particular plan (if known).

### **CoveredPerson**

The <CoveredPerson> element contains identifying information about the individual requesting to be covered by the insurance. The element uses the <PersonName> entity defined in the personal profile.

The <CoveredPerson> element contains one attribute:

- *Relation* – Relation of the traveler to the other covered individuals.

### **Beneficiaries**

The <Beneficiaries> element contains information about individuals that the traveler would like to name as beneficiaries for the insurance plan. It is a collection of individual beneficiary information contained within a <Beneficiary> tag.

### **Beneficiary**

The <Beneficiary> element contains information about a single beneficiary.

**<PersonName>**— Identifies the individual to be named as beneficiary (Same as in the personal profile).

**<Address>**— An optional element containing the beneficiary's address (Same as in the personal profile).

The attributes of the <Beneficiary> element are:

- *Relation* – Relation of the beneficiary to the traveler
- *BenefitPercent* – Percent of the death benefit to be paid to the beneficiary. The percentages for all beneficiaries named by a traveler must add up to 100%.

### **TravelerCoverageReqs**

The <TravelerCoverageReqs> element contains certain traveler-specific information required for quoting or booking an insurance plan.

The elements of the <TravelerCoverageReqs> element are:

**<InsureTripCost>** – Estimated trip cost, or the total price of the journey for the traveler (often required by insurance plans that offer trip protection benefits).

**<FAP\_Amt>** – Amount of flight accident protection (FAP) requested by the traveler (if offered by the insurance plan).

**<CoveredLuggage>** – Contains descriptions of luggage or equipment for which the traveler has requested additional baggage/equipment coverage (if offered by the insurance plan).

**<PreX\_Conds>** – Contains a description of the traveler's pre-existing conditions, the diagnosis date of the condition, and the date of last treatment.

**<TravelerAge>** – Age of the individual traveler. Often obtaining an insurance quote may be accomplished from a relatively small amount of the traveler's personal information. Age of the traveler is often used as a substitute for traveler birth date and the effective date of his/her plan.

**<PlanOption>** – The PlanOption element allows for the traveler to indicate information or requirements that may be unique to the particular plan.

### **InsureTripCost**

The Estimated Trip Cost <InsureTripCost> element contains the total price of the journey for the traveler.

### **FAP\_Amt**

The <FAP\_Amt> element contains the amount of flight accident protection (FAP) requested by the traveler. Some protection package plans offer additional FAP as an add-on to the normal coverage benefits.

### **CoveredLuggage**

The <CoveredLuggage> element contains information on luggage items that the traveler would like to have insured by additional luggage/equipment insurance.

The <CoveredLuggage> element includes one or more <LuggageItem> elements:

**<LuggageItem>** – Information on a single piece of luggage/equipment.

### **LuggageItem**

The <LuggageItem> element contains a description and estimated value for each piece of luggage/equipment for which the traveler would like additional insurance.

The <LuggageItem> elements include:

**<LuggageDescription>** – A detailed description of what the piece of luggage/equipment contains.

**<LuggageValue>** – Estimated value of the piece of luggage/equipment.

**<LuggagePremium>** – Additional cost to insure the piece of luggage/equipment.

The <LuggageItem> element contains one attribute:

- *LuggageType* – The type of luggage to be insured. Suggested values are: Luggage, Equipment, Photo, Perishable.

### **PreX\_Conds**

The Pre-Existing Conditions, <PreX\_Conds> element, defines the traveler's pre-existing medical conditions

The <PreX\_Conds> element contains a collection of <PreX\_Cond> elements:

**<PreX\_Cond>** – contains information on a single pre-existing condition.

#### **PreX\_Cond**

The <PreX\_Cond> element contains a description of one of the traveler's pre-existing conditions as well as the date of diagnosis and the date last treatment was received.

The <PreX\_Cond> attributes include the following:

- *DiagnosisDate* – Date condition was first diagnosed.
- *LastTreatmentDate* – Date condition was last treated medically (medication, hospitalization, etc).

#### **TravelerAge**

The <TravelerAge> element contains the age of the traveler at time of travel.

#### **PlanOption**

In certain instances, some plans may have unique options or requirements. The <PlanOption> element allows for the traveler to communicate this information to the vendor by passing a very general name/value pair.

The attributes of the <PlanOption> element are:

- *OptionName* – Name of the plan option
- *OptionValue* – Value of the plan option

#### **Payment**

The <Payment> element contains payment information and plan cost (if known) for the individual requesting the insurance plan quote/booking.

The child elements of the <Payment> element are:

**<PaymentForm>** – Contains all the necessary payment information to for purchasing the insurance plan (Same as in personal profile).

**<PlanCost>** – Contains information about the total price of the insurance plan (used for booking).

#### **PlanCost**

The <PlanCost> element is a container for the cost breakdown of the insurance plan.

The child elements of the <PlanCost> element are:

**<Premium>** – Cost for the insurance coverage alone (minus any fees/taxes)

**<AdditionalCharges>** – A container for any additional charges to the cost of the insurance plan.

**<TotalPlanCost>** – Total cost of the insurance plan (premium + any additional charges).

**Premium**

The <Premium> element contains the base cost of the insurance plan.

**AdditionalCharges**

The <AdditionalCharges> element is a container for any additional charges to the cost of the plan.

The child element of the <AdditionalCharges> element is:

<Charge> – Contains a description and value of a single additional charge to the cost of the plan.

**Charge**

The <Charge> element identifies the type and value of an additional charge

The child elements of the <Charge> element is:

<ChargeAmount> – Currency amount of the additional charge

<ChargeDescription> – Description of the charge.

The attribute of the <Charge> element is:

- *ChargeName* – Short name/code for the additional charge for use by the insurance vendor.

**TotalPlanCost**

The <TotalPlanCost> element contains the total cost of the plan (premium and additional charges included).

**PlanType**

The <PlanType> element is used to provide the vendor with information about the type of insurance coverage the traveler(s) are interested in quoting/booking. When the specific insurance plan is unknown, the information contained within the <PlanType> element may serve a very basic discovery function to determine what plans are available. The <PlanType> element requires that the customer specify either an annual or package plan.

The child elements of the <PlanType> element are:

<Annual> – Specifies an annual insurance plan. These plans cover all trips taken during a 1 year period.

<Package> – Specifies any non-annual insurance plan (including both single and multi-trip plans).

**Annual**

The <Annual> element contains information required for quoting and booking an annual insurance plan.

The child elements the <Annual> element are:

**<Coverage>** - Information about the type of insurance coverage that applies to all travelers.

**<InsureTripQty>** - How many trips the traveler(s) plan on taking during the year.

**<MaxTripLength>** - The maximum trip length the traveler(s) plan on taking during the year.

**<InsureTripCost>** - Combined cost of trip for all travelers.

**<TripFeatures>** - Information about the traveler(s) journey that is necessary for quoting/booking the insurance.

The attributes of the **<Annual>** element are:

- *EffectiveDate* - The date the insurance policy becomes effective.
- *AutoRenew* - For some annual plans, the customer has the option of automatically renewing their coverage after their policy expires. The AutoRenew attributes accepts a Yes/No value to indicate the customer's preference.

### **Coverage**

For either annual or package plan types, the customer can identify the types of coverage and features using the **<Coverage>** element.

The child elements of the **<Coverage>** entity are:

**<Deductible>** - The amount of applicable deductible.

**<PolicyLimit>** - The dollar amount of available coverage for the policy.

**<IndividualLimit>** - The dollar amount of available coverage for an individual.

**<DeliveryType>** - Method by which the confirmation materials will be sent to the customer. Suggested values are ExpressMail, USMail, Fax.

The **<Coverage>** element has three attributes:

- *CoverageType* - Indicates the types of coverage requested or included in the quoting/booking process.
- *EffectiveDate* - The date coverage becomes effective.
- *Covered* - Indicates whether the coverage is included in the quoting/booking process.
- *Unlimited* - Identifies coverage with an unlimited benefit level.

### **InsureTripQty**

The **<InsureTripQty>** element contains the total number of trips the traveler(s) plan on taking during the coverage period.

### **MaxTripLength**

The <MaximumTripLength> element contains the maximum length of trip to be taken during the period of coverage

### **TripFeatures**

The <TripFeatures> element contains information about the traveler(s) journey that is necessary for quoting/booking the insurance (e.g. destination, trip/tour operator, date the trip deposit was made).

The child elements of <TripFeatures> are:

<**DepartureDate**> – Date traveler(s) first depart on their journey

<**ReturnDate**> - Date traveler(s) return home from their journey

<**DepositDate**> – Date traveler(s) made their initial monetary deposit towards their journey.

<**FinalPayDate**> – Date traveler(s) made their payment towards their journey.

<**Destinations**> – Container for any number of destinations that the traveler(s) will reach during their journey.

<**Activities**> – Activities that traveler(s) will engage in while on their journey

<**TripOperator**> – Company or organization operating the trip/tour/cruise

### **Destination**

The <Destination> element identifies a destination point of the traveler(s) journey as well as the dates of arrival and departure from the destination.

The attributes of the <Destination> element are:

- *DestArrivalDate* – Date traveler(s) intends on arriving at the destination
- *DestDepartureDate* – Date traveler(s) intent on departing from the destination

### **Package**

The <Package> element contains information required for quoting and booking a single/multi-trip insurance plan.

The child elements of <Package> are:

<**Coverage**> - Information about the type of insurance coverage that applies to all travelers.

<**InsureTripCost**> - Combined cost of trip for all travelers.

<**TripFeatures**> – Contains information about the traveler(s) journey that is necessary for quoting/booking the insurance.

The attributes of <Package> are the following:

- *EffectiveDate* – Date policy becomes effective (for most plans this effective date is the date of purchase).
- *ExpireDate* – Date policy expires.

## 4.2 Insurance Quote Response

The insurance quote response returns to the requestor a price quotation, as well as details about the insurance company providing the quote, contact people/numbers if the requestor needs more information, any restrictions on the policy, and booking details.

The quote response is contained within the root element <OTA\_InsuranceQuoteRS> which contains one or more <PlanForQuoteRS> elements and uses the entity OTA\_v2ent.xsd to indicate whether the request was successfully processed, or returns errors or warnings if it was not.

### PlanForQuoteRS

The business data in the quote response message is contained in one or more <PlanForQuoteRS> elements. Each <PlanForQuoteRS> element represents a quote for a single insurance plan.

The child elements for the <PlanForQuoteRS> element are:

<QuoteDetail> – Contains the information about the plan, such as the name of the insurance company, any restrictions, etc.

<PlanCost> - Container for the cost breakdown of the insurance plan.

<Contact> – Contains contact information for an insurance agent call center if the customer has any questions regarding the plan.

The attributes of <PlanForQuoteRS> are the following:

- *ISOCurrency* – Identifies the currency type of the plan price quotation.
- *PlanID* – ID code for specific plan for which the quote has been generated.
- *Type* – Identifies the sub-plan designation of the specific plan for which the quote has been generated.
- *PlanName* – The full name of the plan for which the quote has been generated.

### QuoteDetail

The <QuoteDetail> element contains information about the insurance company offering the plan, any plan restrictions, and the URL for booking details.

The child elements for the <QuoteDetail> element are:

<CompanyName> – The name of the insurance company offering the plan (Same as in the personal profile).

<PlanRestrictions> – Container for any restrictions on the plan's coverage.

<BookingURL> – Location for booking details (uses the URL entity from the personal profile).

The <QuoteDetail> element contains one attribute:

- *URL* – Location of a document containing details about the plan quote.

### **PlanRestriction**

Each <PlanRestriction> element contains a restriction on coverage for the current plan.

The <PlanRestriction> element contains two attributes:

- *PlanRestrictCode* – Identifies a plan restriction by the restriction code
- *PlanRestrictCodeContext* – Identifies the context of the restriction code

## **4.3 Insurance Book Request**

The insurance book request message resembles the insurance quote request in structure and contents. The insurance book request is contained within the <OTA\_InsuranceBookRQ> root element and contains one or more <PlanForBookRQ> elements.

### **PlanForBookRQ**

Each <PlanForBookRQ> element is used to request the booking for a single insurance plan. It contains all of the customer requirements, payment information and data for booking an insurance plan.

The child elements of the <PlanForBookRQ> element are:

<**POS**> – an object containing details for point-of-sale, in this case the immediate seller of the service that can be a broker, travel agent, automated kiosk or direct customer inquiry; this is the same point-of-sale object as used elsewhere in the OTA specifications.

<**InsuranceCustomer**> – an object containing data on the buyer and travelers covered by the requested insurance services.

<**PlanType**> – an object containing details about the insurance services requested.

The <PlanForBookRQ> element contains three attributes:

- *CurrencyCode* – Identifies the currency to be used for purchasing for the plan.
- *PlanID* – Identifies the specific plan to be booked.
- *Type* – Identifies the sub-plan designation of the specific plan to be booked.

## **4.4 Insurance Book Response**

The insurance book response returns to the requestor the details about the insurance plan(s) booked as well as confirms the information that was sent with the insurance book request message.

The booking response is contained within the root element <OTA\_InsuranceBookRS> which contains one or more <PlanForBookRS> elements and uses the entity OTA\_v2ent.xsd to indicate whether the request was successfully processed, or returns errors or warnings if it was not.

### **PlanForBookRS**



The <PlanForBookRS> element represents the booking of a single insurance policy and contains the booking details of that policy.

The child elements of the <PlanForBookRS> element are:

**<PolicyDetail>** – Contains such policy details such as policy number, reference number, effective and expiration dates

**<PlanType>** – Returns coverage information about the policy just booked. (Same as the PlanType element described in the other messages above).

**<CoveredTraveler>** – Identification and contact information for the one or more persons covered under the policy, as well as document identifiers (e.g. passport number), contact names in case of emergencies, and beneficiaries.

**<PlanCost>** – Contains detailed pricing information about the policy just booked (premium, additional charges, total costs)

**<Contact>** – Contains information for important contacts at the insurance company (assistance, claims), as well as vendor support.

The attributes of the <PlanForBookRS> element are:

- *CurrencyCode* – Identifies the currency that the policy was booked with
- *PlanID* – ID code for specific plan which has been booked.
- *Type* – Identifies the sub-plan designation of the policy which has been booked.
- *PlanName* – The full name of the plan for which has been booked.

### **PolicyDetail**

The <PolicyDetail> element contains information that identifies the insurance policy such as the policy number and the dates that it becomes effective and expires.

The child elements of the <PolicyDetail> element are:

**<PolicyNumber>** – Unique policy number assigned by the insurance company upon issuing the policy.

**<PolicyRefNumber>** – Internal vendor reference number (if different from policy number).

**<PolicyEffectiveDate>** – Date policy becomes effective.

**<PolicyExpireDate>** – Date policy coverage expires.

The attributes of the <PolicyDetail> element are:

- *OrderDate* – Date the policy was processed
- *DetailURL* – Location of a document giving more detail about the policy coverage booked through the insurance company.

## 4.5 Insurance Sample XML Instance Documents

Below are examples of the four insurance messages outlined in this document

### 4.5.1 Insurance Quote Request Message

```
<OTA_InsuranceQuoteRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_InsuranceQuoteRQ.xsd" EchoToken="18723222"
TimeStamp="2001-03-31T16:13:08Z" Target="Production" Version="1" SequenceNmbr="1569">
  <PlanForQuoteRQ ISOCurrency="String" PlanID="ACME " Type="protection">
    <InsuranceCustomer>
      <PersonName>
        <GivenName>John</GivenName>
        <Surname>Smith</Surname>
      </PersonName>
      <Email>JSmith@someServer.com</Email>
      <CoveredTraveler>
        <CoveredPerson Relation="Traveler 1">
          <PersonName Gender="Male" BirthDate="19480613">
            <NamePrefix>String</NamePrefix>
            <GivenName>John</GivenName>
            <Surname>Smith</Surname>
          </PersonName>
        </CoveredPerson>
        <TravelerCoverageReqs>
          <InsureTripCost>480</InsureTripCost>
          <FAP_Amt>100000</FAP_Amt>
        </TravelerCoverageReqs>
      </CoveredTraveler>
      <Affiliations>
        <Organization>
          <OrgName>
            <CompanyName CompanyCode="Agent1">M.I.S.C. Insurance
Agency</CompanyName>
          </OrgName>
        </Organization>
      </Affiliations>
    </InsuranceCustomer>
    <PlanType>
      <Annual>
        <MaxTripLength>15</MaxTripLength>
      </Annual>
    </PlanType>
  </PlanForQuoteRQ>
</OTA_InsuranceQuoteRQ>
```

### 4.5.2 Insurance Quote Response Message

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_InsuranceQuoteRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_InsuranceQuoteRS.xsd" EchoToken="18723222" TimeStamp="2001-
03-31T16:13:08Z" Target="Production" Version="1" SequenceNmbr="1569">
  <Success/>
  <PlanForQuoteRS ISOCurrency="USD" PlanID="ACME" Type="protection" PlanName="ACME Insurance
Protection Plan">
    <QuoteDetail>
      <CompanyName>ACME Insurance Services</CompanyName>
      <PlanRestrictions>
        <PlanRestriction PlanRestCode="111">This plan is only available to US
citizens</PlanRestriction>
      </PlanRestrictions>
    </QuoteDetail>
    <PlanCost>
      <Premium>50</Premium>
      <AdditionalCharges>
        <Charge ChargeName="tax">
          <ChargeAmount>1.25</ChargeAmount>
          <ChargeDescription>There is a 2.5% insurance tax on this
policy</ChargeDescription>
```

```

                                </Charge>
                                <Charge ChargeName="fee">
                                <ChargeAmount>4</ChargeAmount>
                                <ChargeDescription>There is a US$4 processing fee included in the
price of this policy</ChargeDescription>
                                </Charge>
                                </AdditionalCharges>
                                <TotalPlanCost>55.25</TotalPlanCost>
                                </PlanCost>
                                <Contact ContactType="Callcenter">
                                    <ContactEmail>callcenter@miscinsurance.com</ContactEmail>
                                    <ContactCompany>
                                        <CompanyName>M.I.S.C Insurance Agency</CompanyName>
                                    </ContactCompany>
                                </Contact>
                                </PlanForQuoteRS>
</OTA_InsuranceQuoteRS>

```

### 4.5.3 Insurance Book Request Message

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_InsuranceBookRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_InsuranceBookRQ.xsd" EchoToken="18723222" TimeStamp="2001-
03-31T16:13:08Z" Target="Production" Version="1" SequenceNmbr="1569">
  <PlanForBookRQ PlanID="ACME" Type="protection">
    <InsuranceCustomer>
      <PersonName>
        <GivenName>John</GivenName>
        <Surname>Smith</Surname>
      </PersonName>
      <Email>JSmith@someserver.com</Email>
      <Address>
        <AddressLine>555 Delivery Lane</AddressLine>
        <CityName PostalCode="22314">Alexandria</CityName>
        <StateProv StateCode="VA">Virginia</StateProv>
        <CountryName CountryCode="US">United States</CountryName>
      </Address>
      <CoveredTraveler>
        <CoveredPerson Relation="Traveler 1">
          <PersonName BirthDate="19480613">
            <GivenName>John</GivenName>
            <Surname>Smith</Surname>
          </PersonName>
        </CoveredPerson>
        <TravelerCoverageReqs>
          <InsureTripCost>480</InsureTripCost>
          <FAP_Amt>100000</FAP_Amt>
        </TravelerCoverageReqs>
      </CoveredTraveler>
      <Contact ContactType="Emergency">
        <ContactPerson>
          <PersonName BirthDate="19480613">
            <GivenName>Jane</GivenName>
            <Surname>Smith</Surname>
          </PersonName>
        </ContactPerson>
        <ContactTelephone>
          <Telephone AreaCityCode="703" PhoneNumber="555-5554"></Telephone>
        </ContactTelephone>
      </Contact>
      <Payment>
        <PaymentForm>
          <CreditCard CreditCardCode="Visa" CardNumber="4111111111111111" ExpireDate="200610">
            <CardHolderName>John P. Smith</CardHolderName>
            <Address>
              <StreetNmbr PO_Box="1000">333</StreetNmbr>
              <BldgRoom>1</BldgRoom>
              <AddressLine ParsedInd="Yes">John Carlyle Street</AddressLine>
              <CityName PostalCode="String">Alexandria</CityName>
            </Address>
          </CreditCard>
        </PaymentForm>
      </Payment>
    </PlanForBookRQ>
  </InsuranceCustomer>
</OTA_InsuranceBookRQ>

```

```

        <StateProv StateCode="VA">Virginia</StateProv>
        <CountryName CountryCode="US">USA</CountryName>
    </Address>
    </CreditCard>
    </PaymentForm>
    <PlanCost>
        <TotalPlanCost>55.25</TotalPlanCost>
    </PlanCost>
    </Payment>
    <Affiliations>
    <Organization>
        <OrgName>
            <CompanyName CompanyCode="Agent1">M.I.S.C. Insurance
Agency</CompanyName>
        </OrgName>
    </Organization>
    </Affiliations>
</InsuranceCustomer>
<PlanType>
    <Annual>
        <MaxTripLength>15</MaxTripLength>
    </Annual>
</PlanType>
</PlanForBookRQ>
</OTA_InsuranceBookRQ>

```

#### 4.5.4 Insurance Book Response Message

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_InsuranceBookRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_InsuranceBookRS.xsd" EchoToken="18723222" TimeStamp="2001-
03-31T16:13:08Z" Target="Production" Version="1" SequenceNmbr="1569">
    <Success/>
    <PlanForBookRS PlanID="ACME" Type="Protection">
        <PolicyDetail OrderDate="20011001">
            <PolicyNumber>ACME20011001P</PolicyNumber>
            <PolicyRefNumber RefContext="M.I.S.C. Invoice Number">1011</PolicyRefNumber>
            <PolicyEffectiveDate>20010403</PolicyEffectiveDate>
            <PolicyExpireDate>20010430</PolicyExpireDate>
        </PolicyDetail>
        <PlanCost>
            <Premium>50</Premium>
            <AdditionalCharges>
                <Charge ChargeName="tax">
                    <ChargeAmount>1.25</ChargeAmount>
                    <ChargeDescription>There is a 2.5% insurance tax on this
policy</ChargeDescription>
                </Charge>
                <Charge ChargeName="fee">
                    <ChargeAmount>4</ChargeAmount>
                    <ChargeDescription>There is a US$4 processing fee included in the price of this
policy</ChargeDescription>
                </Charge>
            </AdditionalCharges>
            <TotalPlanCost>55.25</TotalPlanCost>
        </PlanCost>
        <Contact ContactType="Callcenter">
            <ContactTelephone DefaultInd="Yes">
                <Telephone PhoneTechType="Home" CountryAccessCode="1" AreaCityCode="800"
PhoneNumber="555-5555" Extension="123"/>
            </ContactTelephone>
            <ContactEmail>callcenter@miscinsurance.com</ContactEmail>
            <ContactCompany>
                <CompanyName>M.I.S.C Insurance Agency</CompanyName>
            </ContactCompany>
        </Contact>
        <Contact ContactType="ACME Claims">
            <ContactTelephone DefaultInd="Yes">

```

```
        <Telephone PhoneTechType="Home" CountryAccessCode="1" AreaCityCode="800"
PhoneNumber="555-5556"/>
      </ContactTelephone>
      <ContactAddress>
        <Address>
          <StreetNmbr PO_Box="String">String</StreetNmbr>
          <BldgRoom>String</BldgRoom>
          <AddressLine ParsedInd="Yes">Acme Claims Department</AddressLine>
          <CityName PostalCode="22314">Alexandria</CityName>
          <StateProv StateCode="VA">Virginia</StateProv>
          <CountryName CountryCode="US">United States</CountryName>
        </Address>
      </ContactAddress>
      <ContactCompany>
        <CompanyName>ACME Insurance Services</CompanyName>
      </ContactCompany>
    </Contact>
  </PlanForBookRS>
</OTA_InsuranceBookRS>
```

# **OpenTravel Alliance Message Specification**

## **Publication 2001B**

### **Section 5 - HITIS Integration - Hotel Industry Messages**

## 5.1 Synopsis of HITIS Standards<sup>12</sup>

The following HITIS standards were integrated into the OTA messaging format, creating request and response message pairs based on the functionality of the original specification.:

### **Available Query & Booking Request**

The Hotel Search Request message provides the capability of doing a universal wide area search to identify the hotels that meet the search criteria defined. It returns a list of hotels to be used for the Availability request.

The Hotel Availability Request message queries the system for detailed availability and pricing information for both room and non-room products. The Hotel Availability Request typically precedes the booking process as it returns available options that can be reserved.

The Hotel Reservation Request message is used to request a reservation from the hotel Central Reservation System or Property Management System. This message is used in the seamless environment by a system that does not hold inventory to hold a tentative reservation until it can be committed.

### **Reservation Synchronization**

The Hotel Reservation Notification message is used by systems that have the authority to book a reservation at a property. When the reservation is sent, the property is contractually obligated to accept it, and the booking authority generates a confirmation number for the reservation. This message is also used to synchronize reservation information between different systems in the event of a change.

The Get Message and the Get Message Info messages provide a request/response pair of messages that permit a system that normally receives notifications to ask for a re-transmission of a message by sequence number.

### **Guest Stay History**

The Guest Stay History messages are used to deliver history information about an individual guest's stay at a property. After a guest has checked out of a hotel, data gathered for the duration of the stay, such as type of room, length of stay, arrival and departure dates and additional services, is accumulated and sent back to a centralized guest data warehouse or the hotel's management company.

### **Agent Commissions**

The Agent Commissions messages are used to transfer data about commissions between systems, so that a commission can be paid centrally by the hotel chain, or by services that pay on behalf of hotel chains. The messages enable a hotel to send information about a guest's stay, such as information about the revenue categories for the commission, the amount and currency of payment, and information about the event that generated the commission.

---

<sup>12</sup> All of the OTA Hotel messages reflect a unanimous decision to be designed for maximum use. To achieve this, the OTA Hotel Work Group believes that all XML components should be of type "optional". If industry feels that certain components should be "required", then a comment should be submitted to the OTA web site at <http://www.opentravel.org>

**Statistics**

The Statistics messages are used for sending information to a hotel management company that analyzes statistical data. It serves as the mechanism whereby these system(s) can be updated on a regular basis (typically daily) on the statistics achieved by a hotel. The statistics include information about revenue categories for room stays, bar, foodservice, meeting rooms and other revenue sources.

**Availability, Rate & Inventory**

The Availability, Rate & Inventory messages publish the availability of the room and rate inventory, rate amounts, and restrictions to systems that book reservations. These messages allow a hotel property to update a CRS with information about rates, stay restrictions, rate hurdles, and supplementary charges; etc. within a date range for rate plans. In addition, a hotel system is able to notify designated reservation systems of the creation of new rates, inventory blocks created for groups, and new inventory items, such as a new room type, depending upon the business agreement for distribution.

The Hotel Descriptive Content contains detailed descriptive information about a hotel property data in a format that can be published or queried from the requesting system.

Note: In order to account for the best possible set of schema documents, the OTA Hotel Work Group has provided XML Schema's with as much industry useful information as possible. For this reason, the XML Schemas and XML instance documents will be provided in separate files and the Schema picture diagrams will be removed. Through the combined use of the schema and sample XML files, this should offer the reader a useful set of source documents to begin OTA-ready operations.



# **OTA Hotel Search Request**

## **Request/Response message specifications**



## **5.2 Hotel Search Request (HITIS Hotel Search Query)**

The Hotel Search Request message provides the ability to search for a list of hotel properties that meet certain criteria, supplying information in return that is related to the specific request.

### ***5.2.1 Shopping Trip***

This type of request message is often referred to as a 'wide-area search' because it typically searches for a list of hotels within a geographic area that may be fairly constrained or quite broad. For example, a list of all the hotels within New York City would be an extensive property search, potentially yielding a list in excess of 1,000 hotels (figure is not based on any statistical data). Other geographic data, such as proximity to a specific location, landmark, attraction or destination point, could be used to constrain the summary response to a limited number of hotels.

A hotel search may take place on several levels and may involve a message conversation of several request/response pairs. At the first level, the query, identifies the hotels based on the initial search, usually a geographic area. Once that list of hotels has been returned, the traveler is then able to select from that list. The search criteria must have been fashioned in such a way that the response fulfills the criteria and returns enough data to add value, potentially a means for marketing a hotel. The information returned needs to be more than just the name of the chain and the hotel, but should include sufficient information to be able to choose a specific property.

A wide area search for general availability of properties that meet the search criteria can be implemented across system boundaries; outside of a single hotel chain or a GDS. Multiple queries can be made with different search criteria. However, one fundamental issue that affects the capability of doing a universal wide area search is that there must be a contractual agreement between the hotel and the booking source in order to list the property.

### ***5.2.2 Hotel Use Case - Identify Property List***

The business use case that supports this message identifies a customer or agent (person or system acting on behalf of the customer), that requests a list of properties based on some criteria. The first step in this use case is for the traveler or requesting party to identify the criteria to be used.

The steps of the use case proceed as follows:

- The Customer or agent requests a list of properties based on the desired criteria.
- The system returns a list of properties that meet the criteria.

The requirement to identify the search criteria needed is effectively a system-level precondition for the message to be formulated, since a search for all the hotels in the world would not be feasible nor a reasonable request.

Further steps could provide an additional refinement of the search by repeating the steps:

- The Customer or agent refines the desired criteria to narrow the list of properties.
- The system returns a list of properties that meet the refined criteria.

Possible business processing errors include:

- No properties are returned that meet the input criteria.
- The input criteria must be changed in order to return the desired information.

Additional data that accompanies the response message assists the individual traveler, the travel agent or other booking source in selecting a target hotel. In addition to identifying the hotel by

name and location, that data could include the type of hotel, its rating, a brief description of its services and facilities and any promotions as a means of marketing the property.

The data returned is used to perform a specific availability query on a specific property or multiple properties selected from the list. This functionality is supported today by Central Reservations Systems that are able to do detailed queries once the requestor narrows their choice to the property level.

***Scenario***

An OTA member is looking for a hotel for one night in order to attend the meeting that begins at 9:00 am the next day. While the primary request is for "hotels in Alexandria, VA," the member may wish to include some other interest factor, such as distance from DISA offices, distance from Reagan National Airport, proximity to the King Street Metro station, etc. In addition, he may prefer to select a hotel from among those chains that honor a frequent guest membership. When the list of hotels that meet those preferences are returned, the final choice of hotel may be influenced by the attraction of restaurants or art galleries in nearby Old Town Alexandria that are marketed in conjunction with the hotel listing.

### 5.3 Hotel Search Messages

The OTA version 2001B specification of Hotel Availability provides a request/response pair of messages to support the functionality of searching for a list of hotel properties.

The syntax of the action verbs that are the root elements of a payload document used to exchange hotel search data are enumerated as follows:

- **OTA\_HotelSearchRQ** - Identifies the initial request for hotel property data.
- **OTA\_HotelSearchRS** - Returns a list of hotel properties that meet the requested criteria and/or Warnings, or Errors if the request did not succeed.

#### 5.3.1 OTA\_HotelSearchRQ

The root tag of the OTA\_HotelSearchRQ contains the standard payload attributes entity found in all OTA payload documents. Because the results of the search message could be quite numerous, the request also has an attribute, `MaxResponses` = " ", indicating the maximum number of replies requested.

Attributes of OTA\_HotelSearchRQ are as follows:

- *OTA\_PayloadStdAttributes* - includes the 5 standard attributes on the root tag of all OTA messages as indicated below.
- *MaxResponses* - A positive integer value that indicates the maximum number of responses desired in the return.

#### 5.3.2 Payload Standard Attributes

The meaning and usage of each of these standard attributes is as follows:

- **EchoToken** - a sequence number for additional message identification assigned by the requesting host system. When a request message includes an *EchoToken*, the corresponding response message MUST include an *EchoToken* with an identical value.
- **TimeStamp** - indicates the creation date and time of the message in UTC using the following format specified by ISO 8601:YYYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. e.g.: 20 November 2000, 1:59:38pm UTC becomes 2000-11-20T13:59:38Z
- **Target** - indicates if the message is a test or production message, with a default value of Production. Valid values: (Test | Production)
- **Version** – String value that indicates the version of the message. [For this publication, the version of the OTA\_HotelSearchRQ is one (1).]
- **SequenceNمبر** - A sequence number generated by the sending system that allows for tracking of message receipt or retrieval of specific message(s) for purposes of synchronization between systems.

### 5.3.3 Hotel Search Criterion

The <OTA\_HotelSearchRQ> may have one to many <HotelSearchCriterion> child elements that identify a single search criterion by criteria type. Because many of the types include partial matches to string values such as partial addresses (street names without a number) or partial telephone numbers (area code or three-digit prefix area, etc.) a <MatchType> attribute indicates whether the match to the string value must be exact or only partial.

Finally, to allow the responding system to search for appropriate hotels and respond to the preference criteria in the order of importance to the customer, an <ImportanceType> attribute is included. It indicates whether the input criteria is Mandatory, or of High, Medium or Low priority. Thus, systems can use this attribute to determine how to weight the criteria in its internal processing of the request.

Attributes of HotelSearchCriterion are as follows:

- *Type* - (Search Criteria Type) an enumerated list, indicating the type of search criteria used to process the request. *Type* is a REQUIRED attribute.
- *MatchType* - an enumerated list, with values of "Exact" or "Partial", indicating whether the string of the search value must be an exact match. *MatchType* is a REQUIRED attribute.
- *ImportanceType* - an enumerated list, indicating the level of importance of the search criterion. Acceptable values are "Mandatory", "High", "Medium" or "Low".

The enumerated values of the *Type* attribute are as follows:

- *Area* - Locates a hotel property by geographical position, using the latitude/longitude values as defined in ISO standard 6709. Area may be a bounded area defined by the use of two latitude/longitude values, or a radius from a single point defined by distance.
- *HotelName* - A string value that uses the hotel name or a partial hotel name to find a hotel.
- *HotelReference* - May search for ChainCode or BrandCode only, with HotelCode optional.
- *Address* - Uses any part of address information, such as street name, postal code, or country code to find a hotel in a specific locale.
- *Telephone* - Uses a country code, area code, or other portion of a telephone number to find a hotel in a specific locale.
- *RefPoint* - (Reference Point) searches by proximity to a major attraction, landmark, train station, airport, building such as a convention center, etc., indicated by distance and direction
- *CodeRef* - (Code Reference) searches by category codes for location, type of property, class of accommodations, facilities, amenities, services, etc. The category codes are defined in the Hotel Descriptive Content specification.

**Note:** The Hotel Descriptive Content Notification specification has defined an extensive list of searchable values for hotel properties. While the messages of the Availability, Rate and Inventory specification define the publication of hotel property data, many items refer to code tables or to recommended string values that can be used in a hotel search.

### 5.3.4 Hotel Search Value

The <HotelSearchValue> element is a REQUIRED child element of HotelSearchCriterion and contains the value indicated by the *Type* attribute. Where that value is the name or partial name of a hotel (anticipated to be a string less than 64 characters) that value is expressed as an attribute of the Hotel Search Value element.

The one attribute of the <HotelSearchValue> element is as follows:

- *HotelName* - An optional attribute used to search for a hotel by name or part of a name, when *Type*="HotelName". The data type is a string.

When the *Type* attribute indicates a search value or one of the specific codes defined by the Hotel Descriptive Content specification,<sup>13</sup> that contains attributes or child elements of its own, they are contained in the child elements of the <HotelSearchValue> element.

The child elements of <HotelSearchValue> are expressed as a choice branch as indicated in the following fragment:

```
<xs:element name="HotelSearchValue">
  <xs:complexType>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element ref="Area"/>
      <xs:element ref="HotelRef"/>
      <xs:element ref="Address"/>
      <xs:element ref="Telephone"/>
      <xs:element ref="RefPoint"/>
      <xs:element ref="CodeRef"/>
    </xs:choice>
    <xs:attribute name="HotelName" type="xs:string"/>
  </xs:complexType>
</xs:element>
```

### 5.3.5 Search by Geographical Area

When the request is a search for a hotel by geographical area, as indicated by the Search Criteria Type [*Type*= "Area"], a specific bounded area can be specified by the use of the Position element, which has two attributes, Latitude and Longitude. Typically, two Position values can be specified (the upper left and lower right corners) to delineate the area of the search. Alternatively, a circular area can be specified for the search, expressed by using a single Position (Latitude/Longitude), with a radius indicated by the distance from the single point accompanied by an indication of the unit of measure used to express the distance (e.g.: miles, kilometers, etc.).

<Area> - An element specifying a search by geographical location. Area contains two child elements; Position and Radius.

<Position> - The Position element contains three attributes, *Latitude*, *Longitude*, and *Altitude*, used to indicate the geographic location(s) requested by the search, expressed in notation specified by ISO standard 6709. It is likely that only the first two attributes, *Latitude* and *Longitude*, would be needed to define a geographic area and locate a hotel.<sup>14</sup>

<Radius> - The Radius element contains two attributes, *Distance* and *DistanceMeasure*, that indicate the distance from the center of the radius [the center of the radius is specified

<sup>13</sup> The Hotel Content Description specification is complete and within Section 5a.

<sup>14</sup> See <http://www.ftp.uni-erlangen.de/pub/doc/ISO/english/ISO-6709-summary>, for the notation prescribed by ISO standard 6709.

by the latitude and longitude of the Position], combined with the unit of measure used to express the Distance.

When the request is a search for a hotel by geographical proximity to a specific reference point, such as a major tourist attraction, airport, train station, a building such as a convention center, etc., as indicated by the Search Criteria Type [*Type*= "RefPoint"], the <HotelSearchValue> is a RefPoint element.

**<RefPoint>** - The Reference Point <RefPoint> element allows for a search by proximity to a designated reference point by name. The data type of the Reference Point element is #PCDATA, as the element can be populated by a string that is the name of the reference point used in the search.

Three attributes allow for precise definition of the search by reference point. The attributes of <RefPoint> are as follows:

- *Distance* - An optional attribute indicating the distance to/from a reference point.
- *DistanceMeasure* - When the Distance attribute contains a value, (presumably a numerical value), the unit of measure is a string value that indicate what units are used for the value.
- *Direction* - An optional string value used to indicate the compass point(s) direction, e.g.: S, SE (South, Southeast), FROM the Reference Point TO the hotel location if the search is not a full circumference from the reference point.

### ***5.3.6 Search by Address or Telephone Information***

Another type of search that involves a geographical area uses address or telephone information to locate a hotel. This search allows the requestor to specify a city name or street name, to locate a hotel with that address. Business rules (and good sense) may dictate that when a street name is specified as the search criteria, the city name should also accompany the request. Likewise, when a city name is specified, a state must also be indicated.

If searching by telephone number, the input criteria could include partial telephone information, such as an area code, or three-digit prefix of a telephone number that identifies a specific locale. When searching within large metropolitan area, any search should logically contain enough information that the search can be limited to identify an address or telephone exchange as closely as possible.

**<Address>** - Uses the elements of the Address entity, such as StreetNmbr, AddressLine, CityName, StateProv, etc., to find a hotel in a specific locale using partial address information.

**<Telephone>** - Uses the Telephone element to supply partial telephone information, such as the country code, area code, or other portion of a telephone number to find a hotel in a specific locale.

### ***5.3.7 Search by Hotel Reference***

Another way to search for a hotel involves limiting the search request to a hotel chain, hotel brand, or a specific hotel. Between two trading partners who have agreed upon the codes that identify a hotel chain, brand, or individual property, the search for a hotel the search can be conducted using the Hotel Reference element.



**<HotelReference>** - The Hotel Reference element has three attributes *ChainCode*, *BrandCode*, and *HotelCode*, allowing a search for a hotel by chain or brand only, or by specifying a single hotel identified by its code.

### 5.3.8 Search by Code Reference

A search by Code Reference **<CodeRef>** element allows for a search for a hotel using a code or a string to identify a specific characteristic of the hotel, such as an amenity, facility, or service, etc., by name or by code.

**<CodeRef>** - The data type of the **<CodeRef>** element is a string that can be populated by the name of the characteristic.

Two attributes allow for precise definition of the search by a code when agreed upon between trading partners:

- **Code** - The value of the Code attribute is a code agreed upon between trading partners that represents the criteria of the search.
- **CodeContext** - The CodeContext attribute allows for a reference to the external code table used as the source for the code.

## 5.4 XML Sample Messages - Hotel Search Request

The following sample messages address various ways to use the OTA\_HotelSearchRQ message to locate a hotel:

Example 1 - Search by location, using the attribute *Type* = Area, and defining a bounded area:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="Area" MatchType="Exact" ImportanceType="Mandatory">
    <HotelSearchValue HotelName="String">
      <Area>
        <Position Latitude="+41.47" Longitude="+87.45" Altitude="String"/>
        <Radius Distance="200.5" DistanceMeasure="km">Navigational Assistance</Radius>
      </Area>
    </HotelSearchValue>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>
```

It should be noted that for comprehensive search capabilities, three different paradigms of location should be considered. One is a singular location, another is a bounded area, and the third is a drive path, defined as 10 miles either side of the driving route. The requesting system can use the *Type* = "Area" search to identify a single location by latitude/ longitude, a pair of latitude/ longitude values to define a general vicinity, or a series of latitudes/ longitudes to define a linear route.

Example 2 - Search by Hotel Reference, looking for hotels in the Best Western chain:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="HotelRef" MatchType="Exact" ImportanceType="Mandatory">
    <HotelSearchValue>
      <HotelReference ChainCode="BW"/>
    </HotelSearchValue>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>
```

```

    </HotelSearchCriterion>
</OTA_HotelSearchRQ>

```

Example 3 - Search by Partial Address, looking for hotels on Cactus Rd. in Phoenix, AZ:

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="Address" MatchType="Partial" ImportanceType="High">
    <HotelSearchValue>
      <Address>
        <AddressLine>Cactus</AddressLine>
        <CityName>Phoenix</CityName>
        <StateProv>AZ</StateProv>
      </Address>
    </HotelSearchValue>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>

```

Example 4 - Combined search by Partial Name and Partial Telephone, looking for a hotel that includes the name 'Empress' within in the 623 area code:

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="Telephone" MatchType="Exact" ImportanceType="Mandatory">
    <HotelSearchValue>
      <Telephone AreaCityCode="623" PhoneNumber=""/>
    </HotelSearchValue>
  </HotelSearchCriterion>
  <HotelSearchCriterion Type="HotelName" MatchType="Partial" ImportanceType="Medium">
    <HotelSearchValue HotelName="Empress"/>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>

```

Example 5 - Search by partial Address and Telephone, looking for a hotel on Union Street within in the 623 area code:

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="Address" MatchType="Partial" ImportanceType="High">
    <HotelSearchValue>
      <Address>
        <AddressLine>Union</AddressLine>
        <CityName>Phoenix</CityName>
        <StateProv>AZ</StateProv>
      </Address>
    </HotelSearchValue>
  </HotelSearchCriterion>
  <HotelSearchCriterion Type="Telephone" MatchType="Exact" ImportanceType="Mandatory">
    <HotelSearchValue>
      <Telephone AreaCityCode="623" PhoneNumber=""/>
    </HotelSearchValue>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>

```

Example 6 - Search by Reference Point, looking for hotels within five miles NorthWest of Sky Harbor Airport:

```

<?xml version="1.0" encoding="UTF-8"?>

```

```

<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="Area" MatchType="Exact" ImportanceType="Mandatory">
    <HotelSearchValue>
      <RefPoint Distance="5" DistanceMeasure="mile" Direction="NW">Sky Harbor
Airport</RefPoint>
    </HotelSearchValue>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>

```

Example 7 - Combined search by Reference Point, Partial Address, and CodeRef looking for a deluxe hotel suite within 5 miles NW of Sky Harbor Airport, Phoenix, AZ.

```

<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSearchRQ.xsd" EchoToken="12345" TimeStamp="2001-05-
31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0" MaxResponses="1">
  <HotelSearchCriterion Type="RefPoint" MatchType="Exact" ImportanceType="Mandatory">
    <HotelSearchValue>
      <RefPoint Distance="5" DistanceMeasure="mile" Direction="NW">Sky Harbor
Airport</RefPoint>
    </HotelSearchValue>
  </HotelSearchCriterion>
  <HotelSearchCriterion Type="CodeRef" MatchType="Partial" ImportanceType="High">
    <HotelSearchValue HotelName="">
      <CodeRef Code="S" CodeContext="www.opentravel.org">suite</CodeRef>
    </HotelSearchValue>
  </HotelSearchCriterion>
  <HotelSearchCriterion Type="CodeRef" MatchType="Partial" ImportanceType="High">
    <HotelSearchValue HotelName="">
      <CodeRef Code="D" CodeContext="www.opentravel.org">deluxe</CodeRef>
    </HotelSearchValue>
  </HotelSearchCriterion>
  <HotelSearchCriterion Type="HotelRef" MatchType="Exact" ImportanceType="Medium">
    <HotelSearchValue>
      <HotelReference ChainCode="HH"/>
    </HotelSearchValue>
  </HotelSearchCriterion>
</OTA_HotelSearchRQ>

```

## 5.5 Hotel Search Response Message

The <OTA\_HotelSearchRS> message returns a list of hotel properties that presumably meet the criteria of the request. The response message at the wide area search level does not attempt to provide any data about availability of a property, but supplies an identification of the hotel(s) selected, and descriptive information that may allow a final selection to be made in a more logical (or personalized) fashion.

### 5.5.1 OTA\_HotelSearchRS

The root tag of the <OTA\_HotelSearchRS> uses the OTA\_v2ent.xsd file that defines the root element standard attributes found in all OTA payload documents, and the response options of returning the indication of Success, Warnings or Errors in processing the request. Additionally, just as the request message has the attribute, `MaxResponses = " "`, indicating the maximum number of replies requested, the response message has the attribute, `TotalReturns = " "` to indicate the number of selections returned in response to the request. The use of these attributes enables both the requestor and the responder to filter the number of requests/ responses exchanged.

<OTA\_HotelSearchRS> elements are as follows:

- *OTA\_PayloadStdAttributes* - includes the 5 attributes that occur on the root tag of all OTA messages.
- *TotalReturns* - "integer" A positive integer value that indicates the number of responses returned.

The original wide area search returns the hotel identification, but is not designed to be an extensive list of hotel data. Including hotel content in the original request may result in receiving a large amount of unwanted data for all the hotels in the list. The typical business transaction sequence is to stage requests in phases, getting a list of hotels first and then requesting additional property information on a more granular basis.

### 5.5.2 Hotel Search Record

The <HotelSearchRecord> element is a child element of the hotel search response message and contains information about an individual hotel identified by the search. The collection of hotels returned generally falls into one of two categories: either a geographic collection, or a marketing collection. The Hotel Search Record may be a repeating element to accommodate the number of responses returned.

The attributes of HotelSearchRecord are as follows:

- *HotelName* - the short name of the hotel.
- *Relevance* - an optional attribute, expressed as a decimal value, representing a percentage of 100%, used to indicate the degree to which the property identified meets the search criteria.

The Relevance attribute is used as a way to weight search criteria by percentages and indicate whether the hotel meets the criteria, and to what extent it does, such as an exact match of all the input criteria. Ideally, this would allow a system to calculate the order of the response based on meeting mandatory requirements, and secondly, the other requests of lower importance. In some cases, it may be incumbent upon the searching system order the criteria in the query in order for the responding system to calculate the order of the response.

Some agreements with destination CRO's contain a requirement to return unbiased responses returning all the hotels with the same rating, for example, all hotels rated at 70%. Other systems are required, by contract, to return a list of all their hotels even the Relevance percentage is zero. If the destination CRO returns an unbiased list of equal candidates, the originating system is then tasked with prioritizing the list, and determining how to trim the list of hotels that do not fulfill the requirements of the request.

The child elements of <HotelSearchRecord> are the following:

**<Hotel Reference>** <HotelReference> - uniquely identifies the hotel property to the receiving system.

- *ChainCode* - the code that identifies a hotel chain or management group.
- *BrandCode* - the code that identifies a hotel brand or flag.
- *HotelCode* - the code that uniquely identifies a single hotel property.

**<LocationDescription>** - A string value describing the location of a hotel. LocationDescription has one child element, **<Position>** that uses latitude and longitude to convey a precise location and enables a graphic presentation in systems that can display a map.

**<Position>** - An optional element indicating the latitude and longitude of the geographic location(s) requested by the search, expressed in notation specified by ISO standard 6709. See <http://www.ftp.uni-erlangen.de/pub/doc/ISO/english/ISO-6709-summary>, for a summary of the notation prescribed by the standard.

**<SearchValueMatch>** - The string value used to search for a property is returned. The SearchValueMatch element returns the input value and offers systems the ability to generate text in the Warnings that communicate the reason for failure to locate a hotel. For example, a Reference Point search may return the following processing message; "No hotels found within 5 miles of Oswego Regional Airport".

SearchValueMatch has one attribute, *Match* = " ".

- *Match* - indication of whether a match was found. The datatype is Boolean (true | false).

**<MarketingText>** - An optional string value used to return information about a hotel that may attract guests This element provides additional information to assist in hotel selection. If the requesting system holds a database of hotel content, it can perform an internal query for additional information based upon the stored hotel content data.

## 5.6 Sample XML Messages - Hotel Search Response

The following sample message shows an example of a response to the Hotel Search request in which three hotels were returned that successfully met the input criteria.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelSearchRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_HotelSearchRS.xsd" xmlns="http://www.opentravel.org/OTA"
EchoToken="18723222" TimeStamp="2001-09-11T09:30:47-05:00" Target="Production" Version="0001"
SequenceNmbr="1">
  <Success/>
  <HotelSearchRecord HotelName="Fairfield Inn South Phoenix" Relevance="100">
    <HotelReference ChainCode="MT" BrandCode="FF" HotelCode="PHX123"/>
    <LocationDescription>Two miles East of Phoenix Sky Harbor Airport</LocationDescription>
    <SearchValueMatch Match="true">Deluxe</SearchValueMatch>
    <MarketingText>Pool, Spa, and Health Club on premises</MarketingText>
  </HotelSearchRecord>
  <HotelSearchRecord HotelName="Phoenix Sky Harbor Best Western" Relevance="90">
    <HotelReference ChainCode="BW" BrandCode="" HotelCode="PHXW2"/>
```

```
<LocationDescription>Nestled in the Apache Mountains only five miles NW of Phoenix Sky Harbor
Airport</LocationDescription>
  <SearchValueMatch Match="true">Deluxe</SearchValueMatch>
</HotelSearchRecord>
<HotelSearchRecord HotelName="Hampton Inn Downtown" Relevance="75">
  <HotelReference ChainCode="HH" BrandCode="HP" HotelCode="D2-5689"/>
  <LocationDescription>Downtown Phoenix in shopping and theatre district</LocationDescription>
  <SearchValueMatch Match="true">Deluxe</SearchValueMatch>
  <MarketingText>Deluxe hotel featuring the "Espanola" Southwestern style Full Service restaurant for
your dining pleasure, or snack at our coffee shop serving 24 hours daily</MarketingText>
</HotelSearchRecord>
<TotalReturns>1</TotalReturns>
</OTA_HotelSearchRS>
```

# **OTA Hotel Availability Request**

## **Request/Response message specifications**





## **5.7 Hotel Availability Request (HITIS Availability Query)**

The Hotel Availability Request message is designed to provide the ability to search for a hotel product, and to determine that the hotel product that meets the defined criteria can be obtained. Most commonly, a search for availability is looking for a room that may be available at certain rates, have certain room amenities, be of a specific room type, etc., A request can also be made for a non-room product, such as banquets and meeting rooms. Presumably, an availability request is made with the intent to ultimately book a reservation for an event or for a room stay.

The Hotel Availability Request allows a system to query another system for detailed availability and pricing information for both room and non-room products. An availability request is often preceded by a Hotel Search Request to obtain a list of all hotels, usually within a specific geographic area that are candidates to accommodate a guest's needs. Once a property (or multiple) properties have been identified, then a request for availability is made by sending the property reference in a Hotel Availability Request message to get back more detailed information.

This specification addresses the functionality of a traditional request for availability at a hotel property. It allows a request for 'static' property data published by the hotel, that includes information about the hotel facilities, amenities, services, etc., as well as 'dynamic' (e.g. rate oriented) data. For example, a hotel may have a AAA or AARP rate, but it may not necessarily offer it at all times, which affects the availability of the rate.

### **5.7.1 Shopping Trip**

There is another business model of 'shopping' searches; a type of search that requires a significant number of systems to be queried in order to accomplish the paradigm of comparing offerings from multiple properties. A 'shopping' transaction differs from the traditional request /response transactions that assume a one-to-one relationship between identification of the hotel property and its availability. The shopping search is not well-supported by many systems, including the GDS's, at the current time. This functionality may be addressed in the future as systems are built that will aggregate data that can be used by search systems at one time.

This availability request limits the functionality to the individual property level, and requires that the hotel has been identified, in order to be able to perform an availability request and determine the rate and availability at a specific property.<sup>15</sup> It is presumed that the wide area search, or Hotel Search Query, has preceded the availability message to obtain a list of eligible properties. However, a request for availability could be performed on multiple properties simultaneously by specifying multiple hotels.

### **5.7.2 Hotel Use Cases**

The business use cases that this Hotel Availability Request message supports are the following:

- Availability / Single Property - determines the availability within the constraints of specified criteria for a single identified property.
- Availability / Multiple Properties - determines the availability for multiple properties identified by a Hotel Reference along with additional specified criteria.

---

<sup>15</sup> An availability query for a specific hotel property incorporates the scope of the original HITIS AQ&BR standard. An availability query for hotels in a bounded geographic area is projected for the next version.

- Alternate Availability - retrieves a list of properties (with availability) that are alternates to a property that may not be available. [While the specifications enable the capability to return alternate choices, the qualifications of the actual returns are dependent upon the application processing the request.]
- Rate Quotation / Single Property - obtains rate quotes for a room or non-room product or products at a specific property. Returns a list of the rates available at the hotel for the desired dates.
- Rate Quotation / Multiple Property - obtains rate quotes for a room or non-room product or products at multiple properties. Returns a list of rates for the products specified that are available at the hotels for the desired dates.

### 5.7.3 Source Identification

When a hotel supplier receives an availability request, the source of that request can be identified by the Point-Of-Sale (POS) information. The source of business will have a great impact on what information is returned. When the request is received, a system qualifies the requestor, and makes a judgement about what products to offer depending on who they are. For example, identification of the requestor might open up several available rates in addition to the standard rack rate.

It is not just only one requestor that needs to be identified. In the hotel environment, a rate offering is dependent on not only the identification of the person who will be sleeping in the bed, but other Point-of-Sale information; such as a Corporate Id that establishes a corporate rate. Additionally, that rate may be available only if booked through American Express travel, etc. Therefore, the availability request should carry multiple ID's, with the receiving application having the ability to pass through the identity of all of the entities involved in the transaction.

A customer, guest, or passenger, etc. can be identified using specific information from their customer profile. For business entities, various means of identification are possible: For a TravelAgent, there is an IATA or ARC number; for a corporate entity there is another source of identification, and each has its own unique ID. Often these are DUNS numbers, but not every hotel chain or corporation subscribes to Dun and Bradstreet, which results in different numbers related to a different context. This problem is not unique to hotels, as similarities can be found across specific industry verticals in the travel industries, as well.

The UniqueId in the OTA 2001A Specification provides the foundation for a recognizable means of identification. The type of profile received can be identified using the URL to provide the context, in relation to the following enumeration of ProfileType:

Customer	Internet Broker
Corporation	Rep (Representative) Company
Travel Agency	Airline
Wholesaler	Hotel
Group	Car Rental
Tour Operator	Cruise Line
CRO (Central Reservations Office)	

### 5.7.4 Booking Channels

In addition to knowing the source of the availability request, it is important to identify the booking channel. Source identification is not singular, but can have many variable business cases. Often messages arrive through a switch. For example, a concentrator may have 8 messages arrive that are sent to one receiver. That entity may get a request, hold on to it (aggregate it) before channeling it to the receiver. The concentrator acts like a switch; channeling information (e.g.: to Channel A, etc).

The receiving system needs to know who the business entities involved in the transaction are to the lower levels, in order to enable the routing to various CRS systems. When the CRS ultimately receives the information, it will subjectively look at who is involved in the chain of requests, and apply a myriad of controls that are qualified to a hotel brand, or chain, specific attributes, etc. In order to process the request it is important to consider both 'who they are' and 'who are they making the inquiry on behalf of'. Every request must have as much data about the source as possible which should be found in the header for the switch.

The major categories of data items used to identify the privileges and qualifications include:

1. A Booking Channel - the way in which the activity is going to get through, such as GDS, Internet, dot-com sites, dot-coms with other dot-coms behind them, etc. (these may have no set number or limit)
2. A Booking Source - multiple different entities making inquiries through the identified channels.
3. Additional Qualifiers - identification of the individual guest travel agent, corporation, etc.

The <POS> element described in this specification addresses the categories above. It is mentioned here, but not included in the content model (Schema) of the Hotel Availability RQ and RS messages because there are two alternatives available for a messaging component:

- The POS object is a separate document - the first document in the payload that identifies the multiple sources and booking channels for all documents in the payload, (it would follow the Control document currently identified in OTA specifications for authentication and connection to the trading partner)
- Every availability query request may need to contain a section of Point-of-Sale data about the source as part of the RQ document. (This would restructure the root tag to have two elements, POS and Hotel Availability [HotelAvail], as in the following fragment:<sup>16</sup> If this change were to occur, the current attributes on the root tag of OTA\_HotelAvailRQ would be moved to the HotelAvail element, with the exception of the OTA\_PayloadStdAttributes)

### 5.7.5 Routing Hops

The Routing Hops class used in these HITIS/OTA specifications is used to provide a record of all the systems that a request has passed through. It carries not only the system identification but the ReservationId, or ProfileId (if the message relates to a profile), that ties it to a specific business transaction. Routing Hops can be used to supply information about the source of business and the intermediary parties to a given transaction.

---

<sup>16</sup> This determination will be made upon completion of the review period and publication of the proposed Point-Of-Sale element by OTA as a final specification. In the interim, the POS element may undergo changes in the content model that would affect validation of this specification.

In order to do an analysis on the amount of activity coming through a specific channel, the collection of Routing Hops transactions is also saved in the Guest Stay History.

## 5.8 Hotel Availability Messages

The OTA version 2001B specification of Hotel Availability provides a request/response pair of messages to support the functionality of searching for availability of hotel properties. The message set provides for two levels of inquiry. The <OTA\_HotelAvailRQ> may return specific availability for a hotel, including rates, descriptions, etc., at a Summary level. If that provides enough information, a booking can be made immediately unless greater detailed information is needed.

If more detailed information is needed, a second query can be sent, turning off the *SummaryOnly* indicator, to obtain the additional information. In some systems, the initial response to a request returns all information and policies intact, cached and accessible, which avoids sending the query to the supplier system a second time.

This message requires that the candidate hotel must be specified, but by using the HotelReference within a Booking Segment, multiple <Booking Segments> can be sent in one message to different hotels. However, it is presumed that the request message is sent to a single server.

The syntax of the root elements of the payload document used to exchange a hotel availability request are enumerated as follows:

<OTA\_HotelAvailRQ> - Requests availability of hotel properties by specific criteria that may include: dates, date ranges, price range, room types, regular and qualifying rates, and/or services and amenities.

<OTA\_HotelAvailRS> - Returns information about hotel properties that meet the requested criteria, indicating whether the requested service, rate, room stay, etc. is available within the date(s) specified. The response message may include Warnings from business processing rules, or Errors if the request did not succeed.

### 5.8.1 OTA\_HotelAvailRQ

The root tag of the OTA\_HotelAvailRQ contains the standard payload attributes entity found in all OTA payload documents. In addition, several other attributes are found on the root tag of the Availability Request that govern all of the responses anticipated. The Hotel Availability Request message assumes a pull model, and anticipates that the querying system will send an immediate response.

Attributes of OTA\_HotelAvailRQ are as follows:

- *OTA\_PayloadStdAttributes* - includes the 5 standard attributes on the root tag of all OTA messages.<sup>17</sup>
- *LanguageId* - The ISO code representing the language in which the confirmation will be delivered.
- *SummaryOnly*
- *SortOrder* - enumeration values are "A" (ascending), "D" (descending), or "N" (none).
- *AvailRatesOnly* - A boolean flag that indicates the response should include ONLY those rates that are available in the date range specified, or whether rates that are NOT available should be shown. "Yes" returns only available rates; "No" returns all rates.

---

<sup>17</sup> The Payload Standard Attributes are described in the Hotel Search Request specification

- *BestOnly* - A boolean flag that indicates the response message should send back only the best rate available at the hotel property, as there could be multiple room types and rates that meet the requested criteria (e.g.: rate range). Allowing that the term "best rate" is subjective, the intent means the lowest rate that meets the category of the input criteria, and may have the effect of limiting the return to one rate quote as opposed to multiples.
- *RateRangeOnly* - A boolean flag that indicates the response message should send back only those rates that are available within the range of the minimum and maximum amount specified.

In the message sequence of requesting availability, many systems require a second query to get to product specifics. The first request may be a generalized query, or high-level transaction. Upon narrowing down the request to the single property level, a hotel can be investigated further. On a detailed request, the "Summary Only" flag is set to "No" in order to indicate a request for product availability that returns more information on a specific product or set of products.

### 5.8.2 Booking Segment

To accommodate the ability to perform multiple requests within one message, the availability request contains the repeating element, <BookingSegment>. Each booking segment includes a collection of criteria that requests a bookable entity, which may include designated rate plans, room types, amenities or services, and the request can be used for guest rooms or other inventory items for which availability is sought. Each Booking Segment would be presumed to have a unique date range for each request.

The <BookingSegment> element has three attributes:

- *AvailReqType* - (AvailRequestType) An enumerated type that indicates whether the availability request is for a room or non-room product, such as amenities or other activities (e.g.: banquet, conference room, etc.). Values: 1 = Room, 2 = Non-Room, 3 = Both.
- *BookingSegmentRPH* - (Reference Place Holder) An integer value that provides an index to a Booking Segment among multiples within the same message.
- *MoreIndicator* - The availability response returns the attribute if there were additional rates that could not fit within the availability response. The text value should be echoed in this availability request to indicate where to begin the next block of availability data.

### 5.8.3 Hotel Reference

The <HotelReference> identifies a lodging facility, using a combination of chain, brand, and hotel codes to uniquely identify the hotel property. Whether to use the sending/querying system's or the receiving/responding system's codes depends on which is doing the translating.

The <Hotel Reference> element has three attributes as follows:

- *ChainCode* - The code that identifies a hotel chain or management group. The hotel chain code is decided between vendors. This attribute is optional if the hotel is an independent property that can be identified by the HotelCode attribute.
- *BrandCode* - A code that identifies the brand or flag of a hotel, often used for independently-owned or franchised properties who are known by a specific brand.

- *HotelCode* - The code that uniquely identifies a single hotel property. The hotel code is decided between vendors.
- *AreaCode*
- *HotelCityCode*

#### 5.8.4 Availability by Dates / Date Range

The availability of hotel rooms or non-room services of a hotel, such as banquets or meeting rooms, is dependent upon the dates requested. Generally, the beginning and ending dates or times are known when the request is made. If they are known, the requestor enters the starting date/time and ending date/time in order to enable a system to determine whether the requested product is available during the specified period.

A query for availability by dates defines a "hard date" search in which the guest plans to arrive on a specific date, and depart on a specific date. By contrast, a 'soft date' search involves a start date/time within a specified time duration and an end date / time within a specified duration. A soft date search looks for an 'interest period' or length of stay (LOS) between flexible start and end dates.

Classically, with availability in a hotel being specific to the day of arrival, and based on the day of the week (DOW) pattern, a soft-date search may need to have a 7-day array (one for each day of the week). This type of search is much more difficult to accomplish and tasks the receiving system to work out what is being asked for (hard date vs. soft date), possibly executing separate queries for the LOS for every day of the week within the dates specified.

In order to accommodate the above models of requesting available dates, the Hotel Availability Request message uses a `<StayDateRange>` element to communicate both the start and end dates, as well as the duration of the stay (LOS) and the starting day of the week.

**<StayDateRange>** - Total time span covered by this availability request (from the earliest arrival to the latest departure).

- *StartDate* - the beginning date/time for the availability requested, expressed in `dateTime` format as prescribed by ISO 8601.
- *EndDate* - the ending date/time for the availability requested, expressed in `dateTime` format as prescribed by ISO 8601.
- *LOS* - the Length of Stay or duration expressed as `timeDuration` in the format prescribed by *W3C Candidate Recommendation 24 October 2000XML Schema Part 2: Datatypes*.

If a greater 'soft date search' is required, the `<StayDateRange>` element has one child element, `StayPattern`, `<StayPattern>`, that defines a minimum and maximum length of stay, and can be repeated as many times as needed to and the arrival day of week (e.g.: 7 times, one for each day of the week).

**<StayPattern>** Total time span covered by this availability request (from the earliest arrival to the latest departure).

- *MinLOS* - the minimum Length of Stay requested expressed as `timeDuration` in the format prescribed by *W3C Candidate Recommendation 24 October 2000XML Schema Part 2: Datatypes*.

- *MaxLOS* - the maximum Length of Stay requested expressed as timeDuration in the format prescribed by *W3C Candidate Recommendation 24 October 2000XML Schema Part 2: Datatypes*.
- *ArrivalDOW* - the Day of Week of the starting date for the availability requested. Enumerated values of *StartDOW* are the seven days of the week: Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, or Sunday.

### 5.8.5 Availability by Rates / Rate Ranges

One factor that places limits on an availability request may be the rate amount for the hotel stay or service, defined as an acceptable range between a certain minimum and a maximum. The Rate Range elements allows for specifying the monetary amounts and the currency that the rate amount is requested in.

**<RateRange>** - A range of monetary values within which the cost of the available products are requested. Rate Range has the following three attributes:

- *MinRate* - A decimal value that indicates the minimum monetary value to be considered in a request for an available product.
- *MaxRate* - A decimal value that indicates the maximum monetary value to be considered in a request for an available product.
- *CurrencyCode* - The currency that the monetary value is expressed in, as defined by ISO 4217 - *Codes for the representation of currencies and funds*.

### 5.8.6 Availability by Rate Plans

Another factor that affects availability is the type rate plans offered for a hotel stay or service. These may be grouped into generic categories to provide uniformity across all systems being queried. The *Type* attribute provides the ability to select a rate plan type, such as business, premium, super-saver, etc., from an enumerated list of major categories. The *Value* attribute is a string that can be used to specify a code that is the result of a negotiated rate such as corporate affiliation, memberships, etc., a collection of codes (e.g.: comma delimited), or a string value of a special promotion of offer.

The Type/Value pair for **<RatePlanCandidate>**, can be repeated in a Hotel Availability Request, nested under the collection of **<RatePlanCandidates>**

**<RatePlanCandidate>** - element used to identify available products and rates.

- ***RatePlanType*** - An enumerated type that allows the query to specify a rate category type, and provides major categories for comparison across brands. Valid values: (Standard, Club, Corporate/Business, Deluxe/Premium, Economy, Family, Government, Industry/Travel, Military, Senior, Special/Promotion, Super-Saver, Volume, Weekend, Group, or All.)
- ***RatePlanCode*** - A string value may be used to request a particular code or an ID if the guest qualifies for a specific rate, such as AARP, AAA, a corporate rate, etc., or to specify a negotiated code as a result of a negotiated rate. This is not a persistent value, therefore no length is defined for the string.



- **ID** - A text field used to indicate the associated ID number that is associated with the rate and required in the reservation request in order to obtain the rate.

A combination of CandidateType = "All" and CandidateValue = "AAA" would indicate a request to send all AAA rates (presumably for designated room types) or all government rates, etc. This request presumes a translation layer to convert codes, especially room types and rate codes, exists between the querying system and the CRS or PMS answering the request.

### 5.8.7 Availability by RoomTypes

A request for rooms by category, room type, group block, or a request for a specific room (one unique room in a property) is indicated in the <RoomStayCandidate> element. Additional information about the number of rooms requested, and the number of guests that will occupy a room can also be communicated, particularly as the number of guests and ages of guests often determines the rate for a room or room type.

<RoomStayCandidate> - element used to identify available room products and rates.

- **RoomType** - (formerly, RoomInventoryCode) A code value that indicates the type of room for which this request is made, e.g.: double, king, etc. Values may use the Hotel Descriptive Content table or a codes specific to the property or hotel brand.
- **RoomTypeCode**
- **RoomId** - A string value representing the unique identification of a room if the request is looking for a specific room.
- **RoomCategoryCode** - A value that allows the query to specify a class of room, and provides major categories for comparison across brands. Recommended values are listed in the Hotel Descriptive Content table {See footnote}.
- **PromoCode** - A code that identifies a special promotion or publicity announcement.
- **InvBlockCode** - A code or identification number that identifies the room stay as part of a group, package tour, or block of rooms designated in the inventory.
- **Quantity** - Integer value that represents the number of units of the inventory, or number of rooms requested. \*

\*In a general availability query request and response scenario, the number of rooms desired, indicated by the *Quantity* attribute is required. However, there may be the desire to send in a request with the number of rooms left at null. This may be used to indicate several things. If the number of rooms is unknown at this time, the request may simply be looking to determine if the hotel has ANY available rooms.

In a group or wholesale scenario, a trading partner agreement may designate that leaving the number of rooms at null dictates a response for the total number of rooms for the requested rate.

<GuestCount> - A recurring element that identifies the number of guests and ages of the guests in the request that determines the rates based on business rules for occupancy at each property.

- **AgeQualCode** - (AgeQualifyingCode) A code that indicates the ages of the guests. The code is specific to a property of hotel chain and represents a business rule that may determine the charges for a guest based upon age range. For example, Adult, Child, Senior, Child With Adult, Child Without Adult, etc. This attribute allows for an increase in rate by occupant class.

- **Count** - The number of guests in one AgeQualifyingCode. The total number of guests in a room is the total of the Count of guests in all AgeQualifyingCodes.

If a guest count is required but not filled in on the request, the responding system may assume a default. Generally, the default is one adult.

Since GuestCount is a repeating element that indicates the count of each Age Qualifying Code, it is assumed that the total of all GuestCounts is the sum total of guests in all rooms.

Static data is published by a hotel and should be able to be queried by category. Dynamic information about the availability of a hotel property with respect to effective rates and availability of rooms is sent periodically to the hotel CRS or to an external booking source to update the current status.

**Note:** For hotels adopting the OTA/HITIS specifications, the source of the data for detailed descriptive information about a hotel is published in the Hotel Descriptive Content specification. The specification identifies the fields, such as amenities, facilities, etc. that may be used in the query and notes where code tables can be used to find these values. e.g.: Location Category Codes; describing a property located on a beach, near an airport, etc., An additional document categorizes these codes, and publishes acceptable or recommended values that can be used by trading partners.<sup>18</sup>

The reference to code tables allows for flexibility between trading partners to reference proprietary code tables, with the CodeContext attribute identifying the code table to be used. If the value of the CodeContext attribute is a URI, the location of the code table may be specified. This reference may be used to identify other code tables from outside sources as well, e.g.: IATA/ARC code tables, etc.

## 5.9 Sample XML Message

The following sample message illustrates an instance of the OTA\_HotelAvailRQ message.

Example 1 - Combined summary message to request availability for a 2 night room stay of 1 room for 1 Adults at a government rate, along with a request for a 2 or 3 night room stay at a weekend rate for 1 room for 2 Adults and 2 Children:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_HotelAvailRQ.xsd" EchoToken="12345" TimeStamp="2001-05-31T17:19:42-
05:00" Target="Production" Version="1" SequenceNbr="0" LanguageId="Eng" SummaryOnly="true" SortOrder="A"
AvailRatesOnly="true" BestOnly="true" RateRangeOnly="true">
  <BookingSegments>
    <BookingSegment AvailReqType="Room" BookingSegmentRPH="011">
      <HotelReference ChainCode="HQ" BrandCode="Flagship" HotelCode="PHX-W"/>
      <StayDateRange StartDate="2001-09-11T09:30:47-05:00" EndDate="2001-09-
11T09:30:47-05:00" LOS="P2D">
        <StayPattern MinLOS="P1Y2M3DT10H30M" MaxLOS="P1Y2M3DT10H30M"
ArrivalDOW="Wed"></StayPattern>
      </StayDateRange>
      <RateRange CurrencyCode="USD" MinRate="89.00" MaxRate="129.00"/>
      <RatePlanCandidates>
        <RatePlanCandidate RatePlanType="Govt" RatePlanCode="USDA"/>
      </RatePlanCandidates>
      <ProfileRefs>
        <ProfileRef>
```

<sup>18</sup> The list of Hotel Descriptive Content values is currently an Excel spreadsheet available on the OTA Members-Only web site. Codes will be added to the list of published values at a later date.

```

                                <Uniqueld Type="Profile" Id="11223344" URL="http://www.usda.gov"
Instance="1999-02-28T09:13:53"/>
                                </ProfileRef>
                                </ProfileRefs>
                                <RoomStayCandidates>
                                    <RoomStayCandidate RoomCategoryCode="Deluxe" RoomType="K"
RoomId="Presidential" Quantity="1" InvBlockCode="PDK1" PromoCode="QXYYZ13">
                                        <GuestCount AgeQualCode="Adult" Count="1"/>
                                    </RoomStayCandidate>
                                </RoomStayCandidates>
                                <SearchCodes>
                                    <CodeRef Code="1276GF" CodeContext="www.hyatt.com"/>
                                    <CodeRef Code="34875" CodeContext="www.opentravel.org"/>
                                </SearchCodes>
                                </BookingSegment>
                                <BookingSegment AvailReqType="Room" BookingSegmentRPH="012">
                                    <HotelReference ChainCode="HQ" BrandCode="Flagship" HotelCode="PHX-W"/>
                                    <StayDateRange StartDate="2001-09-11T09:30:47-05:00" EndDate="2001-09-
11T09:30:47-05:00" LOS="P2D">
                                        <StayPattern MinLOS="P0Y0M2D" MaxLOS="P0Y0M3D"
ArrivalDOW="Fri"></StayPattern>
                                        <StayPattern MinLOS="P0Y0M1D" MaxLOS="P0Y0M2D"
ArrivalDOW="Sat"></StayPattern>
                                    </StayDateRange>
                                    <RateRange CurrencyCode="USD" MinRate="59.00" MaxRate="109.00"/>
                                    <RatePlanCandidates>
                                        <RatePlanCandidate RatePlanType="Weekend"/>
                                    </RatePlanCandidates>
                                    <ProfileRefs>
                                        <ProfileRef>
                                            <Uniqueld Type="Profile" Id="1234" URL="http://www.ibm.com"/>
                                        </ProfileRef>
                                    </ProfileRefs>
                                    <RoomStayCandidates>
                                        <RoomStayCandidate RoomCategoryCode="Standard" RoomType="DD"
RoomId="" Quantity="2" InvBlockCode="PDK1" PromoCode="QXYYZ13">
                                            <GuestCount AgeQualCode="Adult" Count="2"/>
                                            <GuestCount AgeQualCode="Childw/Adult" Count="2"/>
                                        </RoomStayCandidate>
                                    </RoomStayCandidates>
                                    <SearchCodes>
                                        <CodeRef Code="1276GF" CodeContext="www.hyatt.com"/>
                                        <CodeRef Code="34875" CodeContext="www.opentravel.org"/>
                                    </SearchCodes>
                                </BookingSegment>
                            </BookingSegments>
                        </OTA_HotelAvailRQ>

```

The following sample message illustrates an instance of the OTA\_HotelAvailRQ message.  
Example 2. Generic response using the OTA\_HotelAvailRS message.

```

<OTA_HotelAvailRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="C:\Download\OTA\OTA2001B\Oct25Data\HotelWG\OTA_HotelAvailRS.xsd"
EchoToken="12345" TimeStamp="2001-05-31T17:19:42-05:00" Target="Production" Version="1" SequenceNmbr="0">
    <SummaryResponse HotelName="String" IsRoom="Yes" IsConverted="Yes" IsAlternate="Yes"
ReqdGuaranteeType="None" RatePlanCode="String" RoomTypeCode="String">
        <HotelReference ChainCode="String" BrandCode="String" HotelCode="String"/>
        <MarketingText>Text</MarketingText>
        <RateQuote>
            <DateRange EndDate="String" StartDate="String"/>
            <QuotedRate CurrencyCode="String" RateAmount="String" RateBasisUnitsQty="String"
TimeUnitType="Year" MinLOS="String" MaxLOS="String" StayOverDate="String"/>
            <MaxOccupancy Count="String" AgeQualCode="String"/>
            <AddlOccupancyCharge Amount="String" CurrencyCode="String" AgeQualCode="String"/>
        </RateQuote>
        <TPA_Extensions>Text</TPA_Extensions>
    </SummaryResponse>
    <OTA_SucceedDef>
        <Success/>
    </OTA_SucceedDef>

```

</OTA\_HotelAvailRS>

## 5.10 Hotel Availability Response Message

The response to the Availability Query request message returns available product offerings, including rates, type of rooms available, and potentially any information about the hotel. The structure of the response message is the same whether or not a "Summary Only" was requested. The requestor will undoubtedly wish to see information about extra charges, minimum stay restrictions, or deposits or guarantees that are required. Additional data elements may be suppressed because they not considered needed.

### OTA\_HotelAvailRS

The root tag of the OTA\_HotelAvailRS message contains the standard payload attributes entity found in all OTA payload documents.

- **OTA\_PayloadStdAttributes** = includes the 5 standard attributes on the root tag of all OTA messages.

The OTA\_HotelAvailRS element has one child element, SummaryResponse, which can be repeated many times to return as many responses as met the criteria of the request.

#### 5.10.1 SummaryResponse

The content model of the <SummaryResponse> element includes the identification of the hotel, by Hotel Reference, marketing text as determined by the supplier, and a RateQuote element that returns the information about the rate, room type, stay date range and any guarantees required. Each summary response element is for ONE rate plan code, for ONE room type, in other words, one inventoriable item.

The attributes of Summary Response are the following:

- **HotelName** - A text field used to communicate the proper name of the hotel.
- **IsRoom** - A boolean flag that if the product queried is a room or other services. Valid values: No indicate a Non-Room product or service, Yes indicates a Room.
- **IsConverted** - A boolean flag that indicates whether the rate amount quoted has been converted from another currency.

**Note:** Some hotels in International locations, particularly in South America, will not quote rates in any currency other than local because of the frequent fluctuation of their currency. While the rate amount may be displayed in the currency of the request, this flag indicates whether that display is a conversion of the actual quoted rate.

- **IsAlternate** - A boolean flag that indicates if the product is a match for the search criteria or is a suggested alternative product. Valid values: No (Not an Alternative), Yes (Alternative Product).
- **ReqdGuaranteeType** - (Required Guarantee Type) An enumerated type indicating what type of required guarantee/ deposit policy is required with the reservation. Valid values: None (No Guarantee Required), Guarantee Required, Deposit Required, Immediate Payment Required.
- **RatePlanCode** - Identification of the rate plan for which the RateQuote is made.
- **RoomTypeCode** - Identifies the room type associated with the rate plan.

- *RateIndicator* - An enumerated type indicating special conditions with the rate. Valid values: ChangeDuringStay, MultipleNights, Exclusive, OnRequest, LimitedAvailability.
- *MoreIndicator* - A text field used to indicate that there are additional rates that could not be fit within the availability response. The text returned should be meaningful in identifying where to begin the next block of data and sent in the availability request.
- *CurrencyCode* - The currency that the monetary value is expressed in, as defined by ISO 4217 - Codes for the representation of currencies and funds.
- *CorporateID* - A text field used to indicate the associated corporate ID number that is associated with the rate and required in the reservation request in order to obtain the rate.

The <SummaryResponse> element has these child elements: HotelReference, Marketing Text, and Rate Quote, AdditionalDetails, and TPA\_Extensions.

### **5.10.2 Hotel Reference**

The <HotelReference> identifies the lodging facility, using a combination of chain, brand, and hotel codes to uniquely identify the hotel property.

### **5.10.3 Marketing Text**

The <MarketingText> element is a field that allows for any free-form text used to promote the hotel property. This field may include embedded graphics or other multimedia files to describe the hotel property.

### **5.10.4 Rate Quote**

The rate quote element returns the amount quoted for one room stay, but it can be used for fixed stay dates or for sliding soft dates for each rate plan. If a system is searching for availability with soft dates, it is possible to return different rates for room stays that start on different days of the week e.g.: 3 days with one rate, 3 days with another rate, etc. A guarantee type is associated with a specific rate in a one-to-one relationship.

The <RateQuote> element has four child elements: StayDateRange, QuotedRate, MaxOccupancy and AddlOccupancyCharge. These elements allow for the accommodation of rate fluctuation within length of stay, specifying a date range for each quoted rate, as well as adding other charges related to additional occupancy of the room stay on the same rate plan.

### **5.10.4 AdditionalDetails**

A collection of AdditionalDetail elements which contain the “AdditionalDetail” element that serves as a vehicle for textual information from property description and location to business rules for guarantees, deposits, cancellation fees, etc

### **5.10.5 TPA\_Extensions**

The TPA\_Extensions element is allowed as part of each Summary Response as a means for trading partners to add extensions to the Hotel Availability response message. Trading partners will need to write their own schemas to cover items not included within conforming implementations of the schema. Trading partners are encouraged to submit extensions widely used between multiple trading partners for consideration to become part of future revisions of OTA specifications.

### 5.10.6 Date Range

The <DateRange> element defines the start and end dates of the quoted rate with the following two attributes:

- *StartDate* - the beginning date/time for the availability requested, expressed in dateTime format as prescribed by ISO 8601.
- *EndDate* - the ending date/time for the availability requested, expressed in dateTime format as prescribed by ISO 8601.

### 5.10.7 Quoted Rate

The Quote Rate indicates the amount of the rate and currency for the rate plan and room type identified in the Summary Response, as well as other restrictions that may apply to that rate, such as the minimum or maximum length of stay, or any required stay-over dates (such as a Saturday night). It indicates the number of units that the quoted rate is based upon, as well as the TimeUnits type used that the rate is based upon, e.g.: 3days at \$100.00 per day.

The Quoted Rate element has the following attributes:

- *RateAmount* - A decimal value indicating the monetary amount quoted for the hotel stay.
- *CurrencyCode* - The currency that the monetary value is expressed in, as defined by ISO 4217 - *Codes for the representation of currencies and funds*.
- *RateBasisUnitsQty* - The number of units that the quoted rate was based on.
- *TimeUnitType* - An enumerated type indicating the interval of time that the rate is based upon.
- *MinLOS* - The minimum length of stay required by this rate plan.
- *MaxLOS* - The maximum length of stay allowed by this rate plan.
- *StayOverDate* - Indicates a specific day on which the guest must stay over in order to be eligible for the quoted rate plan.

### 5.10.8 Maximum Occupancy

The <MaxOccupancy> element defines the maximum number of guests that are allowed in the room type for the quoted rate plan. The MaxOccupancy element may be repeated for each AgeQualifyingCode to define the maximum number of occupants at each age.

Max Occupancy has the following two attributes:

- *Count* - An integer value indicating the maximum number of guests of that age range that are allowed to occupy the room.
- *AgeQualifyingCode* - a code indicating the age qualification of the occupants of the room.

### 5.10.9 Additional Occupancy Charges

The <AddlOccupancyCharge> element defines the amount charge for additional occupants per age range. The AddlOccupancyCharge element may be repeated for each AgeQualifyingCode to define the amounts charged for occupants at each age level.

AddOccupancyCharge has the following three attributes:

- *Amount* - A decimal value indicating the monetary amount that will be charged for each additional guest of the specified age range.
- *CurrencyCode* - The currency that the monetary value is expressed in, as defined by ISO 4217 - *Codes for the representation of currencies and funds*.
- *AgeQualifying Code* - a code indicating the age qualification of the additional occupants of the room.





# **OTA Hotel Reservation Request**

## **Request/Response message specifications**



## 5.11 Hotel Reservation Request Messages

The Hotel Reservation Request message is used to send a request from one booking source to another booking source requesting a hotel reservation. Typically the Hotel Reservation Request message would be used by a Central Reservation System (CRS) Global Distribution System (GDS), Internet bookers, or other travel service providers that does not have the authority to book a reservation directly, but must determine the status of a property prior to booking a reservation. In the travel industry, allotments of inventory become difficult to manage if dispersed to multiple parties, so the control of inventory is usually held by the hotel property or the CRO of the hotel chain.

The Hotel Reservation Request message is often preceded by an Availability Request message. Upon querying the system that holds the inventory and learning that inventory is available at a chosen hotel property, the request is sent to book the hotel services.

The Hotel Availability Request/Response messages do not hold inventory when the response of availability is received. The availability query response only provides a snapshot at the time that the request is made. Depending upon the time between determining availability and sending the request to book a reservation, it cannot be assumed that a booking request will be approved.

### OTA\_HotelResRQ

There is no requirement to determine availability prior to sending a reservation request. Travel agencies, or individual guests may send a request to book a reservation from an internet site if all the information required for booking is known. The OTA\_HotelResRQ message can initiate the first message in the sequence of booking a reservation.

**<OTA\_HotelResRQ>** – Sends a request for a reservation to another system. All the elements and attributes that constitute the reservation that are known are sent with the request.

**<OTA\_HotelResRS>** - Returns confirmation that the reservation has been successfully booked, and includes a confirmation or reservation number to identify the reservation. Warnings from business processing rules or errors are returned if the request did not succeed. It may optionally include the updated reservation data.

The OTA\_HotelResRQ root tag, in addition to the standard payload attributes, contains a Reservation Request Type attribute that indicates whether the request is for a new reservation, or whether the request is to ignore a request, modify a request, or commit the transaction.

- *OTA\_PayloadStdAttributes* - the standard attributes on the root tag of all OTA messages.
- *ResRequestType* - An enumeration that indicates the action to be taken on the request. Valid values: (Initiate | Ignore | Modify | Commit).

The message conversation may involve several request/response pairs before the final reservation is booked. During the process, a reservation can be rolled back or cancelled until the point at which the reservation is committed. In the seamless environment, the reservation system makes a commitment at an interim point but must retract that commitment if the reservation is not completed. For reservations that carry deposit penalties, refund penalties, or are non-cancelable, an interim commitment cannot be made.

The reservation request is an atomic request that can either be approved or denied depending on the status of the hotel inventory or whatever other business reasons that the hotel might have for declining the request.

The first three enumerations of the *ResRequestType* attribute; 1) Initiate, 2) Ignore, 3) Modify, indicate a tentative message and are used before a commitment is made or a reservation contractually incurred. The purpose of the Modify attribute is to change what is being requested. It does not modify an already confirmed booking. A cancellation cannot be made, and no cancellation penalties can be applied, until a message indicating a Commit has taken place. It is incumbent on the receiving system to periodically clean up tentative transactions, particularly in cases where the Ignore is never successfully received.

Once the Commit is specified and a ConfirmationID and/or ReservationID returned in the Reservation Booking Response message, a reservation exists from that point forward. A Committed reservation requires a new message request be initiated in order to change the reservation. By starting with the confirmation number or ReservationID of the existing reservation, the current reservation has been identified.

When a system requests a new tentative reservation that modifies a confirmed reservation, it would not want to cancel the original commitment before being able to confirm the change. The requesting system would need to retain the original reservation while making changes, and the receiving system would be tasked to process the modification request according to business rules.

## 5.12 Hotel Reservation

The Reservation Booking Request <OTA\_HotelResRQ> and the Reservation Notification Request, <OTA\_HotelResNotifRQ> both use the Reservation class, which is strictly the data for the information associated with the reservation.

<**HotelReservation**> – The HotelReservation is defined in a separate document.

### 5.12.1 Routing Hops

Routing hops are relevant to both the Reservation Notification message and the Reservation Request message. Routing hops trace the path the Reservation object has travelled, passing through a number of systems to reach the reservation destination.

While this is dynamic information on the transaction, the collection of Routing Hops transactions is also saved along with the Reservation in the Guest Stay History. It can be used to do an analysis on the amount of activity coming through a specific channel. The original booking source is identified in the attribute ReservationOriginatorCode, and the OriginalRoutingHops element persists the path of the original reservation transaction.

<**RoutingHops**> – A collection of RoutingHop Objects.

<**RoutingHop**> - A RoutingHop object conveys information about the path that this notification update took; e.g., how many and which systems it passed through.

The attributes of the Routing Hop element are the following:

- *SystemCode* – A system code agreed upon by trading partners
- *LocalRefId* - Identifier within the System that refers to this specific document (reservation)
- *TimeStamp*- The date and time that the reservation passed through a routing hop
- *Comment* – Information that the system adding this routing hop may use upon return of the data in a future response
- *SequenceNmbr* - Sequential number assigned to hops, Base 0.
- *Data* - This attribute is provided so that each system can put in whatever data it would like (e.g., auditing information)

<**WrittenConfInst**> – This object describes the method by which the confirmation is to be delivered and to whom it will be delivered.

The attributes of Written Confirmation Instruction are as follows:

- *LanguageID* – The ISO code representing the language in which the confirmation will be delivered.
- *AddresseeName*– The name of the individual or representative to whom the confirmation will be addressed.

- *Address* – The physical address to which the confirmation will be sent.
- *Telephone* – The telephone number to which a facsimile (fax) will be sent.

**<Email>** – The Electronic mail address of the party receiving the confirmation. This element uses the Email element from the OTA 2001A.

**<SupplementalData>** - Additional data that will be sent with the confirmation. This could be used to include a map, pictures, or any other information that the reservation source wishes to include with the confirmation.

### 5.13 Sample XML Message using the OTA\_HotelResRQ XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<!--Hotel Reservation Request for a hotel room for one person for two nights, guaranteed by credit card-->
<OTA_HotelResRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA OTA_HotelResRQ.xsd" EchoToken="00006#1P#42251#AB345"
Target="Production" Version="SS">
  <POS>
    <Source AgentSine="GS" PseudoCityCode="XYZ" ISOCountry="US">
      <Uniqueld Type="BSA" Id="I234567"/>
    </Source>
    <Source>
      <Uniqueld Type="BSM" Id="I765432"/>
    </Source>
  </POS>
  <Uniqueld Type="Reservation" Id="I56789"/>
  <HotelReservations>
    <HotelReservation RoomStayReservation="1" ReservationOriginatorCode="XYZ"
OriginalBookingDate="2002-01-09T16:13:36.000-05:00">
      <ResComments>
        <ResComment ResCommentRPH="1" CommentOriginatorCode="HSB"
GuestViewable="1">
          <Comment>Hotel Source Booking</Comment>
        </ResComment>
        <ResComment ResCommentRPH="2" CommentOriginatorCode="ARR"
GuestViewable="1">
          <Comment>DL670-10AM-ORD</Comment>
        </ResComment>
      </ResComments>
      <ResCreditCards>
        <ResCreditCard ResCreditCardRPH="1">
          <CreditCard CreditCardCode="AX" CardNumber="1234567890123"
ExpireDate="10-03"/>
        </ResCreditCard>
      </ResCreditCards>
      <ResProfiles>
        <ResProfile ResProfileRPH="1">
          <Profile ProfileType="Customer">
            <Customer>
              <PersonName>
                <NamePrefix>DR.</NamePrefix>
                <GivenName>Jason</GivenName>
                <MiddleName>M</MiddleName>
                <Surname>Jones</Surname>
                <NameTitle>M.D.</NameTitle>
              </PersonName>
              <CustLoyalty ProgramCode="FRT"
MembershipId="1234567890123456789012"/>
              <CustLoyalty ProgramCode="FRG"
MembershipId="12345678901234567"/>
              <CustLoyalty ProgramCode="CID"
MembershipId="1234567890"/>
            </Customer>
          </Profile>
        </ResProfile>
      </ResProfiles>
    </HotelReservation>
  </HotelReservations>
</OTA_HotelResRQ>
```

```

        </ResProfile>
      </ResProfiles>
    <RoomStays>
      <RoomStay NumberOfRooms="1" RoomTypeCode="IJ2ODOC2"
InvBlockCode="IT12345678901">
        <HotelReference ChainCode="XX" HotelCityCode="ORD"
HotelCode="XX103"/>
        <TimeSpan>
          <DateTimeSpan StartInstant="2002-01-16T16:00:00"
EndInstant="2002-01-18T12:00:00"/>
        </TimeSpan>
        <GuestCounts>
          <GuestCount AgeQualifyingCode="Adult" Count="1"/>
        </GuestCounts>
        <RatePlans>
          <RatePlan RatePlanRPH="1">
            <Rates>
              <Rate RateRPH="1">
                <Amount
CurrencyCode="USD"/>
                <TimeSpan>
                  <DateTimeSpan/>
                </TimeSpan>
              </Rate>
            </Rates>
          </RatePlan>
        </RatePlans>
        <GuaranteeDeposits>
          <GuaranteeDeposit>
            <Guarantee GuaranteeCode="GCC"
GuaranteeType="CC/DC/Voucher" ResCreditCardRPH="1"/>
          </GuaranteeDeposit>
        </GuaranteeDeposits>
      </RoomStay>
    </RoomStays>
    <SpecialRequests>
      <SpecialRequest SpecialRequestRPH="1" RequestCode="EXP"
RequestComments="Extra person rate"/>
      <SpecialRequest SpecialRequestRPH="2" RequestCode="EXC"
RequestComments="Extra child rate"/>
      <SpecialRequest SpecialRequestRPH="3" RequestCode="EBA"
RequestComments="Extra adult bed rate"/>
      <SpecialRequest SpecialRequestRPH="4" RequestCode="EBC"
RequestComments="Extra child bed rate"/>
      <SpecialRequest SpecialRequestRPH="5" RequestCode="CRB"
RequestComments="Crib rate"/>
      <SpecialRequest SpecialRequestRPH="6" RequestCode="SUP"
RequestComments="Requests Pool Side Room"/>
    </SpecialRequests>
    <HotelReservationIDs>
      <HotelReservationID ResIDType="PNR" ResIDValue="9X78G2"
ResIDSource="XYZ" ForGuest="1"/>
    </HotelReservationIDs>
    <HotelReservation>
      <RoutingHops>
        <RoutingHop SystemCode="" LocalRefId="" TimeStamp="2002-01-10T11:17:56"/>
      </RoutingHops>
      <WrittenConflnst Address="12345 S Main St., Big Town USA 12345-1212">
        <SupplementalData/>
        <Email/>
      </WrittenConflnst>
    </HotelReservations>
  </OTA_HotelResRQ>

```

### 5.14 Hotel Reservation Response Message (OTA\_HotelResRS)

The response to a booking request is either Yes or No based upon the availability. The hotel PMS or CRO system responds back attaching a confirmation number or additional information such as the reservation ID, etc. when the response is affirmative. Additional information, such as the count of Loyalty Program miles or points to be awarded for the hotel stay, can be added to the Reservation object in the return. Supplementary data for the reservation can be added later, once the reservation has been confirmed and the inventory held.

The response message replies with an indication of the status of the reservation request. If the reservation was committed and accepted by the PMS or CRS that makes the reservation, the confirmation number will be returned along with the reservation.

The enumerations of the *ResRequestType* attribute are echoed in the response message by the *ResResponseType* attribute that indicates whether the reservation request was 1) Initiated, 2) Ignored, 3) Modified, or 4) Committed. The first response to a reservation message that indicates a tentative reservation has been made is likely to be "Initiated". Subsequent requests to change the tentative reservation will return a status of "Modified" until the reservation is either discarded, or "Ignored" or confirmed, which would return a response type of "Committed".

The response message uses the external entity OTA\_v2ent.xsd that defines Success, Warnings or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if processing the request did not succeed.

**<OTA\_HotelResRS>** - The response message returns the reservation information along with the indication of the status of the reservation request.

- *OTA\_PayloadStdAttributes* - the standard attributes on the root tag of all OTA messages.
- *ResResponseType* - An enumeration that indicates the status of the reservation request. Valid values: (Initiated | Ignored | Modified | Committed).

**<UniqueID>**- the unique identifier for this reservation, using the OTA UniqueID element.<sup>19</sup>

**<HotelReservations>** - The HotelReservation is returned with the response message in order to add the Confirmation Number or Reservation Id as well as include data that may have been added or changed by the supplier.

**<RQRouting>**— The Routing Hops that were sent in the request message are echoed back in the response message.

**<RoutingHops>** – A collection of RoutingHop Objects.

**<RoutingHop>** - A RoutingHop object conveys information about the path that this notification update took; e.g., how many and which systems it passed through.

The attributes of the Routing Hop element are the following:

- *SystemCode* – A system code agreed upon by trading partners

---

<sup>19</sup> The UniqueId contains four attributes: *Id*, *Type*, *URL* and *Instance*.



- *LocalRefID* - Identifier within the System that refers to this specific document (reservation)
- *TimeStamp* - The date and time that the reservation passed through a routing hop
- *Comment* – Information that the system adding this routing hop may use upon return of the data in a future response
- *SequenceNumber* - Sequential number assigned to hops, Base 0. This attribute is REQUIRED.
- *Data* - This attribute is provided so that each system can put in whatever data it would like (e.g., auditing information)

**<Success>****<Warnings>****<Warning>**

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*
- *RecordNmbr*

**<Errors>****<Error>**

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the error
- *DocURL*
- *Status*
- *Tag*
- *RecordNmbr*

**5.15 Sample XML Message**

Sample response using the OTA\_HotelResRS XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<!--Hotel Reservation Response with a rate change on the second night-->
<OTA_HotelResRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA OTA_HotelResRS.xsd" EchoToken="00006#1P#42251#AB345"
Target="Production" Version="SEVER">
  <Uniqueld Type="Reservation" Id="I56789"/>
  <Success/>
  <HotelReservations>
    <HotelReservation RoomStayReservation="1" ReservationOriginatorCode="XYZ"
OriginalBookingDate="2002-01-09T16:13:36.000-05:00">
      <ResComments>
        <ResComment ResCommentRPH="1" CommentOriginatorCode="HSB"
          <Comment>      Hotel Source Booking</Comment>
        </ResComment>
        <ResComment ResCommentRPH="2" CommentOriginatorCode="RTD"
          <Comment>Rate description text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="3" CommentOriginatorCode="RMD"
          <Comment>Room description text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="4" CommentOriginatorCode="CMV"
          <Comment>Commission text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="5" CommentOriginatorCode="RML"
          <Comment>Room location text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="6" CommentOriginatorCode="TTL"
          <Comment>Total rate text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="7" CommentOriginatorCode="MKT"
          <Comment>Marketing text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="8" CommentOriginatorCode="GRQ"
          <Comment>Guarantee requirements text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="9" CommentOriginatorCode="CXP"
          <Comment>Cancellation penalty text</Comment>
        </ResComment>
        <ResComment ResCommentRPH="10" CommentOriginatorCode="CKP"
          <Comment>Early checkout penalty text</Comment>
        </ResComment>
      </ResComments>
      <ResProfiles>
        <ResProfile ResProfileRPH="1">
          <Profile ProfileType="Customer">
            <Customer>
              <PersonName>
                <NamePrefix>DR.</NamePrefix>
                <GivenName>Jason</GivenName>
                <MiddleName>M</MiddleName>
                <Surname>Jones</Surname>
                <NameTitle>M.D.</NameTitle>
              </PersonName>
              <CustLoyalty ProgramCode="FRT"
                <CustLoyalty ProgramCode="FRG"
                  <CustLoyalty ProgramCode="CID"
                    </CustLoyalty ProgramCode>
              </CustLoyalty ProgramCode>
            </Customer>
          </Profile>
        </ResProfile>
      </ResProfiles>
    </HotelReservation>
  </HotelReservations>
  <MembershipId="1234567890123456789012"/>
  <MembershipId="12345678901234567"/>
  <MembershipId="1234567890"/>

```

```

        </ResProfiles>
        <RoomStays>
            <RoomStay NumberOfRooms="1" RoomTypeCode="IJ2ODOC2"
InvBlockCode="IT12345678901">
                <HotelReference ChainCode="XX" HotelCityCode="ORD"
HotelCode="XX103"/>
                <TimeSpan>
                    <DateTimeSpan StartInstant="2002-01-16T16:00:00"
EndInstant="2002-01-18T12:00:00"/>
                </TimeSpan>
                <GuestCounts>
                    <GuestCount AgeQualifyingCode="Adult" Count="1"/>
                </GuestCounts>
                <RatePlans>
                    <RatePlan RatePlanRPH="1" RatePlanCode="Weekend">
                        <Rates>
                            <Rate RateRPH="1"
AmountGuaranteed="1">
                                <Amount
CurrencyCode="USD">85.66</Amount>
                                <TimeSpan>
                                    <DateTimeSpan
StartInstant="2002-01-16T16:00:00" EndInstant="2002-01-17T16:00:00"/>
                                    </TimeSpan>
                                </Rate>
                                <Rate RateRPH="2"
AmountGuaranteed="1">
                                    <Amount
CurrencyCode="USD">99.99</Amount>
                                    <TimeSpan>
                                        <DateTimeSpan
StartInstant="2002-01-17T16:00:00" EndInstant="2002-01-18T12:00:00"/>
                                        </TimeSpan>
                                    </Rate>
                                </Rates>
                            </RatePlan>
                        </RatePlans>
                        <GuaranteeDeposits>
                            <GuaranteeDeposit>
                                <Guarantee GuaranteeCode="GRQ">
                                    <ResCommentRPHs>
                                        <ResCommentRPH>8</ResCommentRPH>
                                    </ResCommentRPHs>
                                </Guarantee>
                            </GuaranteeDeposit>
                        </GuaranteeDeposits>
                        <CancelPenalties>
                            <CancelPenalty>
                                <ResCommentRPHs>
                                    <ResCommentRPH>9</ResCommentRPH>
                                </ResCommentRPHs>
                                <Amount CurrencyCode="USD">25.00</Amount>
                                </CancelPenalty>
                                <CancelPenalty>
                                    <ResCommentRPHs>
                                        <ResCommentRPH>10</ResCommentRPH>
                                    </ResCommentRPHs>
                                <Amount CurrencyCode="USD">25.00</Amount>
                                </CancelPenalty>
                            </CancelPenalties>
                        </RoomStay>
                    </RoomStays>
                <SpecialRequests>
                    <SpecialRequest SpecialRequestRPH="1" RequestCode="EXP"
RequestComments="10.00"/>
                    <SpecialRequest SpecialRequestRPH="2" RequestCode="EXC"
RequestComments="5.00"/>

```

```

RequestComments="10.00 "/>
RequestComments="5.00"/>
RequestComments="6.00"/>
RequestComments="Call for Limo"/>
    </SpecialRequests>
    <HotelReservationIDs>
        <HotelReservationID ResIDType="Confirmation" ResIDValue="1234567890"
ResIDSource="" ForGuest="true"/>
        <HotelReservationID ResIDType="PNR" ResIDValue="9X78G2"
ResIDSource="" ForGuest="true"/>
    </HotelReservationIDs>
</HotelReservation>
<RQRouting>
    <RoutingHop SystemCode="" LocalRefId="" TimeStamp="2002-01-10T11:17:56"/>
</RQRouting>
<RoutingHops>
    <RoutingHop SystemCode="" LocalRefId="" TimeStamp="2002-01-10T11:17:56"/>
</RoutingHops>
</HotelReservations>
</OTA_HotelResRS>

```

# **OTA Hotel Reservation Notification**

## **Request/Response message specifications**



## 5.16 Hotel Reservation Notification Messages

The OTA version 2001B specification of Hotel Reservation Notification provides a request/response pair of messages to support the functionality of updating other systems with reservation data. The message set assumes a push model, with the originating system pushing the data to another system. The originating system would usually be a booking source, such as a Global Distribution System (GDS), a Central Reservation System (CRS) or some other agent of the hotel.

The business model assumes that the originating system either has the authority to take a reservation, or is passing along a message from such a system. The message is a notification of the creation, modification, or cancellation of a reservation, and does not require the receiving system to confirm the booking, only the receipt of the message. The responding system may add its own data (such as its own confirmation ID) and include that data in the response message

The originating system will send a report using the OTA\_HotelResNotifRQ message. The receiving system will acknowledge its receipt of that report using the OTA\_HotelResNotifRS message.

**OTA\_HotelResNotifRQ** – Sends a reservation to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_HotelResNotifRS** - Returns acknowledgement that the reservation has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed. It may optionally include the updated reservation data.

### Note:

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description

### 5.17 Hotel Reservation Notification Request Message (OTA\_HotelResNotifRQ)

The root tag of OTA\_HotelResNotifRQ contains the standard payload attributeGroup OTA\_PayloadStdAttributes, found in all OTA payload documents.

**OTA\_HotelResNotifRQ** - This is the main Request element.

- *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages. These attributes are defined in the OTA Version 2 Master Document.
- *ResStatusType* – Indicates the status of the reservation represented by the message. This is an enumeration with possible values of Initiate, Ignore, Modify, Commit.

**UniqueID**- Unique Identifier for this reservation. UniqueID is defined in the OTA Version 2 Master Document.

**HotelReservations** – The collection of HotelReservation elements is defined in a separate document.

**RoutingHops** – A collection of RoutingHop Objects.

**RoutingHop** - A RoutingHop object conveys information about the path that this notification update took; e.g., how many and which systems it passed through.

- *SystemCode* – A system code agreed upon by trading partners
- *LocalRefID* - Identifier within the System that refers to this specific document (reservation)
- *TimeStamp*- The date and time that the reservation passed through a routing hop
- *Comment* – Information that the system adding this routing hop may use upon return  
*SequenceNumber* - Sequential number assigned to hops, Base 0
- *Data* - This attribute is provided so that each system can put in whatever data it would like (e.g., auditing information)

**WrittenConfInst** – This object describes the method by which the confirmation is to be delivered and to whom it will be delivered.

- *LanguageID* – The ISO code representing the language in which the confirmation will be delivered.
- *AddresseeName*– The name of the individual or representative to whom the confirmation will be addressed.
- *Address* – The physical address to which the confirmation will be sent.
- *Telephone* – The telephone number to which a facsimile (fax) will be sent.

**SupplementalData** - Additional data that will be sent with the confirmation. This could be used to include a map, pictures, or any other information that the reservation source wishes to include with the confirmation.



**Email** – The Electronic mail address of the party receiving the confirmation.

## 5.18 Sample XML Message using the OTA\_HotelResNotifRQ XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelResNotifRQ EchoToken="6802409" Target="Production" ResStatusType="Commit" Version="1"
  TimeStamp="2001-05-23T06:13:57.000-00:00" xmlns="http://www.opentravel.org/OTA"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opentravel.org/OTA OTA_HotelResNotifRQ.xsd">
  <Uniqueld Type="SELL" Id="1" Instance=""/>
  <HotelReservations>
    <HotelReservation RoomStayReservation="1" ReservationOriginatorCode="AA" OriginalBookingDate="2001-
05-23T06:13:37.000-07:00">
      <ResComments>
        <ResComment CommentOriginatorCode="RateCodeDescription" ResCommentRPH="1"
          GuestViewable="1">
            <Comment>CORPORATE RATE</Comment>
          </ResComment>
        <ResComment CommentOriginatorCode="Guarantee" ResCommentRPH="2"
          GuestViewable="1">
            <Comment>FIT CENTRAL BILL@TTL RM/TX BKD AUTO REIMBURSED ON
            BW STMT</Comment>
          </ResComment>
        <ResComment CommentOriginatorCode="CancelPolicyCode" ResCommentRPH="3"
          GuestViewable="1">
            <Comment>4PM</Comment>
          </ResComment>
        <ResComment CommentOriginatorCode="RESERVATION" ResCommentRPH="4"
          GuestViewable="1">
            <Comment>adv cxl by 4pm on 05/29/02 /cd</Comment>
          </ResComment>
        <ResComment CommentOriginatorCode="RESERVATION" ResCommentRPH="5"
          GuestViewable="1">
            <Comment>2 OF 2 RES REQ NON SMOKING AND GROUND FLOOR NOT
            GTD REQ WRITTEN CONFIRMATION NO</Comment>
          </ResComment>
        <ResComment CommentOriginatorCode="RESERVATION" ResCommentRPH="6"
          GuestViewable="1">
            <Comment>NON SMOKING ROOM PLEASE @ @ WOULD LIKE ROOM NEAR
            MY FRIEND RONALDO CIRLENCIONES @ @</Comment>
          </ResComment>
        <ResComment CommentOriginatorCode="ForcedSellInd" ResCommentRPH="7"
          GuestViewable="1">
            <Comment>Y</Comment>
          </ResComment>
      </ResComments>
      <ResCreditCards>
        <ResCreditCard ResCreditCardRPH="1">
          <CreditCard CardType="Credit" CreditCardCode="VA"
            CardNumber="123456788934949"
              ExpireDate="04/2002">
                <CardHolderName>DAVID BRASHEARE</CardHolderName>
              </CreditCard>
            </ResCreditCard>
          </ResCreditCards>
        <ResProfiles>
          <ResProfile ResProfileRPH="1">
            <Profile ProfileType="Customer">
              <Customer>
                <PersonName NameType="DirectGuestPrimary">
                  <GivenName>DAVID</GivenName>
                  <Surname>BRASHEARE</Surname>
                </PersonName>
                <TelephoneInfo PhoneUseType="Home">
                  <Telephone AreaCityCode="" PhoneNumber="860
599 2090"/>
                </TelephoneInfo>
                <TelephoneInfo PhoneUseType="Fax">
```

```

        <Telephone AreaCityCode=""
PhoneNumber="4072906000"/>
    </TelephoneInfo>
    <TelephoneInfo PhoneUseType="Business">
        <Telephone AreaCityCode=""
PhoneNumber="7138870782"/>
    </TelephoneInfo>
    <Email
EmailType="Unknown">bshear@wherever.com</Email>
    <AddressInfo AddressType="Home">
        <Address>
            <AddressLine>2099 E. PATRICK
LN.</AddressLine>
            <AddressLine>APT. 244</AddressLine>
            <CityName PostalCode="85024-
2025">PHOENIX</CityName>
            <StateProv
StateCode="AZ">ARIZONA</StateProv>
            <CountryName
CountryCode="US">UNITED STATES</CountryName>
        </Address>
    </AddressInfo>
    </Customer>
    </Profile>
    </ResProfile>
    <ResProfile ResProfileRPH="2">
        <Profile ProfileType="Customer">
            <Customer>
                <PersonName NameType="FrequentGuest">
                    <GivenName>DAVID</GivenName>
                    <Surname>BRASHEARE</Surname>
                </PersonName>
                <TelephoneInfo PhoneUseType="Fax">
                    <Telephone AreaCityCode=""
PhoneNumber="4072895433"/>
                </TelephoneInfo>
                <TelephoneInfo PhoneUseType="Business">
                    <Telephone AreaCityCode=""
PhoneNumber="491981 2"/>
                </TelephoneInfo>
                <Email
EmailType="Unknown">bshearsen@wherever33.com</Email>
                <AddressInfo>
                    <Address>
                        <AddressLine>2099 E PATRICK
LN.</AddressLine>
                        <AddressLine>APT 244</AddressLine>
                        <CityName PostalCode="85024-
2025">PHOENIX</CityName>
                        <StateProv
StateCode="AZ">ARIZONA</StateProv>
                        <CountryName
CountryCode="US">UNITED STATES</CountryName>
                    </Address>
                </AddressInfo>
                <CustLoyalty MembershipId="87019169"
ProgramCode="GCC"/>
            </Customer>
            </Profile>
            </ResProfile>
            <ResProfile ResProfileRPH="3">
                <Profile ProfileType="Trvl_Agnt">
                    <CompanyInfo>
                        <CompanyName CompanyCode="57922200"
CodeContext="IATA/ARC">NIKKEY TRAVEL SERVICE TURISMO</CompanyName>
                        <AddressInfo AddressType="Business">
                            <Address>
                                <AddressLine>PRACA DA LIBERDADE
NO.272</AddressLine>
                                <AddressLine>5 ANDAR-
LIBERDADE</AddressLine>

```

```

010">SAO PAULO/SAO PAULO</CityName>
CountryCode="BR">BRAZIL</CountryName>
0 485104912"/>
11 254-6000"/>
CodeContext="CorporateAccount">TRAVEL TRUST INTERNATIONAL</CompanyInfo>
HAUPTBAHNHOF</AddressLine>
PostalCode="80539">MUNICH</CityName>
StateCode="GR">GERMANY</StateProv>
CountryCode="DE">GERMANY</CountryName>
PhoneNumber="GR2414702850"/>
PhoneNumber="00493088753801"/>
EmailType="Business">ens.yursuv@aquafina.com</Email>
Duration="P5D"/>
StartInstant="2002-05-29T00:00:00.000-00:00"

<CityName PostalCode="01503-
<StateProv StateCode="XX"/>
<CountryName
</Address>
</AddressInfo>
<TelephoneInfo PhoneUseType="Fax">
<Telephone AreaCityCode="" PhoneNumber="42
</TelephoneInfo>
<TelephoneInfo PhoneUseType="Business">
<Telephone AreaCityCode="" PhoneNumber="55
</TelephoneInfo>
<Email EmailType="Business">travelrus@inet.com</Email>
</CompanyInfo>
</Profile>
</ResProfile>
<ResProfile ResProfileRPH="4">
<Profile ProfileType="Corporation">
<CompanyInfo>
<CompanyName CompanyCode="00555780"
<AddressInfo AddressType="Business">
<Address>
<AddressLine>AM
<CityName
<StateProv
<CountryName
</Address>
</AddressInfo>
<TelephoneInfo PhoneUseType="Business">
<Telephone AreaCityCode=""
</TelephoneInfo>
<TelephoneInfo PhoneUseType="Fax">
<Telephone AreaCityCode=""
</TelephoneInfo>
<Email
</CompanyInfo>
</Profile>
</ResProfile>
</ResProfiles>
<RoomStays>
<RoomStay NumberOfRooms="1" RoomTypeCode="K|NSM|2" ReservationStatusType=""
RoomStayRPH="1" PromoCode="CO" MarketSegmentCode="PROMO">
<HotelReference ChainCode="BW" HotelCode="03071" BrandCode=""/>
<TimeSpan>
<DateTimeSpan StartInstant="2002-05-29T00:00:00"
</TimeSpan>
<GuestCounts GuestOccupancyInd="AllRooms">
<GuestCount Count="1" AgeQualifyingCode="Adult"/>
<GuestCount Count="1" AgeQualifyingCode="Children"/>
</GuestCounts>
<RatePlans>
<RatePlan RatePlanRPH="1">
<Rates>
<Rate RateRPH="1" AmountGuaranteed="1">
<Amount
<TimeSpan>
<DateTimeSpan

```

```

Duration="P3D"/>
                                </TimeSpan>
                                </Rate>
                                <Rate RateRPH="2" AmountGuaranteed="1">
                                <Amount
CurrencyCode="USD">54</Amount>
                                <TimeSpan>
                                <DateTimeSpan
StartInstant="2002-06-01T00:00:00.000-00:00"
Duration="P2D"/>
                                </TimeSpan>
                                </Rate>
                                </Rates>
                                </RatePlan>
                                </RatePlans>
                                <GuaranteeDeposits>
                                <GuaranteeDeposit>
                                <Guarantee GuaranteeCode="CC-DEP"/>
                                </GuaranteeDeposit>
                                <GuaranteeDeposit>
                                <Deposit>
                                <PaymentInstructions>
                                <PaymentInstruction>
                                <PaymentDue DueDate="2002-04-29T00:00:00.000-
00:00">
                                <Amount
CurrencyCode="USD">65.7</Amount>
                                </PaymentDue>
                                </PaymentInstruction>
                                </PaymentInstructions>
                                </Deposit>
                                </GuaranteeDeposit>
                                </GuaranteeDeposits>
                                <CancelPenalties>
                                <CancelPenalty CancelByDate="2002-05-29T16:00:00.000-00:00"/>
                                </CancelPenalties>
                                </RoomStay>
                                </RoomStays>
                                <SpecialRequests>
                                <SpecialRequest RequestCode="CRIB" RequestComments="Req Crib at 10.00."
SpecialRequestRPH="1"/>
                                <SpecialRequest RequestCode="ROLLAWAY" RequestComments="Req 2 Rlwys at 15.00
each"
                                SpecialRequestRPH="2"/>
                                </SpecialRequests>
                                </HotelReservation>
                                </HotelReservations>
</OTA_HotelResNotifRQ>

```

### 5.19 Hotel Reservation Notification Response Message (OTA\_HotelResNotifRS)

The root tag of OTA\_HotelResNotifRS uses the external *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_HotelResNotifRS** - Main Response element.

- *OTA\_PayloadStdAttributes*

**HotelReservation** - The collection of HotelReservation elements is defined in a separate document. The HotelReservation as defined separately may be echoed back with the response in order to include data that may have been added or changed by the supplier. For example, the confirmation identification given by the Property Management System may be returned to the reservation source.

**RQRouting** – The Request (OTA\_HotelResNotifRQ) routing hops are copied to this object for the response message.

**RoutingHop** – as defined in OTA\_HotelResNotifRQ message.

**RoutingHops** – Routing hops are the routing of the return message.

**RoutingHop** – as defined in OTA\_HotelResNotifRQ message.

#### Success

#### Warnings

##### Warning

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warnin
- *DocURL*
- *Status*
- *Tag*

#### Errors

##### Error

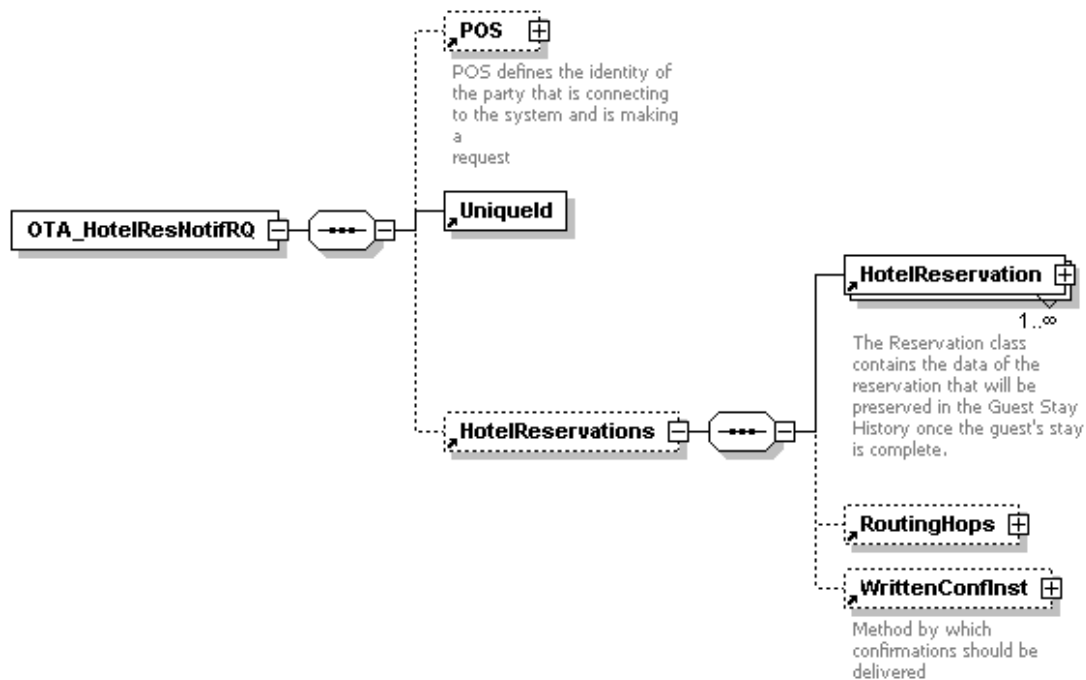
- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

## 5.20 Sample XML Message

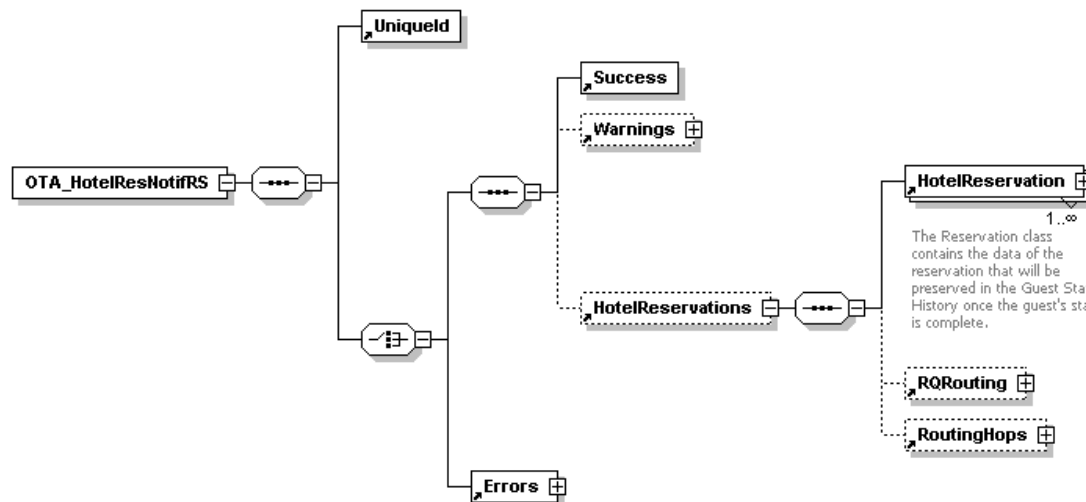
Generic use of OTA\_HotelResNotifRS XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelResNotifRS EchoToken="6802409" ResResponseType="Commit" Version="1"
  TimeStamp="2001-05-23T06:14:03.000-00:00" xmlns="http://www.opentravel.org/OTA"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opentravel.org/OTA OTA_HotelResNotifRS.xsd">
  <Uniqueld Type="SELLR" Id="1" Instance="V6009T"/>
  <Success/>
</OTA_HotelResNotifRS>
```

## 5.21 XML Schema Diagrams for Reservation Notification Request



## 5.22 XML Schema Diagrams for Reservation Notification Response



### 5.23 Revision Log for OTA\_ResNofitRQ/RS

Date	Version	Revisions
May 6, 2001	0.1	Initial Draft.
October 30, 2001	0.2	Updated.

### 5.24 Hotel Reservation Fragment

**HotelReservation** - The Reservation class contains the data of the reservation. This fragment is used in other messages that relate to the reservation itself. These messages include Hotel Reservation Request and Response, Hotel Reservation Notification Request and Response, Stay Info Notification Request and Response.

- *RoomStayReservation* – boolean - True if this reservation is reserving rooms. False if it is only reserving services.
- *ReservationOriginatorCode* - Identifying code of the originating system for this reservation
- *OriginalBookingDate* - Date of the creation of this reservation
- *CancelOriginatorCode* - Identifying code of the system that initiated the cancel of this reservation
- *CancellationDate* - Date this reservation was cancelled.

**StayDateRange** - Total time span covered by this reservation (earliest arrival to latest departure).

**DateTimeSpan** - The DateTimeSpan represents specific durations of time.  
(DateTimeSpan.xsd)

- *StartInstant* – The DateTime at which a time span begins.
- *Duration* – The duration of the time span.

- *EndInstant* - The DateTime at which a time span ends.

**GuestCounts** - Categorization of all guests in this reservation. The total number of guests will equal the sum of the Count attribute of all GuestCount objects in this collection and assumes that the AgeQualifyingCode will be used to determine how each guest should be billed. Each roomstay and service has its own GuestCount attribute to help determine billing, headcount, etc.

**GuestCount** - The GuestCount object indicates the number of guests attached to a specific reservation, room stay, or service. The total number of guests in a room is the total of the Count of guests in all AgeQualifyingCodes.

- *AgeQualifyingCode* - A code representing a business rule that determines the charges for a guest based upon age range. For example, Adult, Child, Senior, Child With Adult, Child Without Adult, etc. This attribute allows for an increase in rate by occupant class.
- *Count* - The number of guests in one AgeQualifyingCode.

**ResComments** - A collection of ResComment objects. Comments which apply to any part of the reservation (the reservation in its entirety, one or more guests, one or more services, or one or more roomstays). Which comments apply to which part is determined by each object's ResCommentsRPHs collection.

**ResComment** - The ResComment object is used to store a further information (comments) pertaining to a reservation. Comments appended to the reservation may be retained as the reservation passes through multiple channels

- *ResCommentRPH* - This is a reference placeholder, used as an index for this object
- *CommentOriginatorCode* - Unique identifier for the system which created the comment
- *GuestViewable* - Whether or not this comment should be shown to the guest

**Comment** - The actual comment in free-form text.

**ResCreditCards** - A collection of ResCreditCard objects. The collection of credit cards which apply to any part of the reservation (any GuaranteeInfo or PaymentInstruction object). Which credit card applies to which part is determined by each object's ResCreditCardRPH attribute. This object can be used to indicate vouchers or credit/debit cards attached to a specific reservation (e.g., as a guarantee) as well as traditional credit cards.

**ResCreditCard** - The ResCreditCard object derives from the CreditCard object and inherits all of its attributes. The ResCreditCard object identifies a credit card (or debit card) associated with a reservation. The CreditCard object is defined in the CRS Profile Synchronization standard.

- *ResCreditCardRPH* - Used as an index to the specific ResCreditCard object.

**CreditCard** – As defined in OTA Profile.



**ResGuests** – A collection of ResGuest objects, identifying the guests associated with this reservation. Which guests are in which room is determined by each RoomStay's ResGuestRPHs collection.

**ResGuest** - The ResGuest object contains the information about a guest associated with a reservation.

- *ResGuestRPH* – This is a reference placeholder, used as an index for this guest in this reservation. In the ResGuest object it is used like all other RPH attributes to send the delta of a reservation. It is used by the RoomStay and Service objects to indicate which guests are associated with that room stay or service.
- *AgeQualifyingCode* - A code representing a business rule that determines the charges for a guest based upon age range. For example, Adult, Child, Senior, Child With Adult, Child Without Adult, etc. This attribute allows for an increase in rate by occupant class.
- *ArrivalTime* - This TimeInstant represents the date and time of the guest's arrival at the hotel. For example, this could be used to indicate a late arrival.
- *DepartureTime* - This TimeInstant represents the date and time of the guest's departure from the hotel.
- *GroupEventCode* - The identification of a group meeting or convention. Used to track a guest who is part of an inventory block (e.g.: group), and can be used for a pick-up when the guest calls in to reserve a room as part of the block. Whether this is taken from inventory depends upon the InventoryBlockCode in the Room Stay class. This code is associated with a group or event as agreed between two parties.

**ProfileRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the ResProfile object. The ResProfileRPH attribute in the ResProfile object is an indexing attribute that identifies other profiles attached to the reservation.

**ProfileRPH** - This is a reference placeholder used as an index for the profiles associated with this reservation, e.g., a corporate account.

**SpecialRequestRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the SpecialRequest object. The SpecialRequestRPH attribute in the Special Request object is an indexing attribute that identifies the special requests of a specific guest or of the reservation.

**SpecialRequestRPH** - This is a reference placeholder used as an index for a special request to be associated with this reservation.

**ResCommentRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the ResComment object. The ResCommentRPH attribute in the ResComment object is an indexing attribute that identifies a specific comment attached to the reservation.

**ResCommentRPH** - This is a reference placeholder used as an index for the comments associated with this reservation.

**ServiceRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the Service object. The ServiceRPH attribute in the Service object is an indexing attribute that identifies the services attached to a specific guest or to the reservation.

**ServiceRPH** - This is a reference placeholder used as an index for a service to be associated with this stay

**InHouseTimeSpan** - The time span during which this guest will be at the hotel, for billing purposes.

**DateTimeSpan** - The DateTimeSpan represents specific durations of time.

**ArrivalTransport** - The details regarding how and when this guest will arrive, e.g., flight info.

**TransportInfo** - A TransportInfo object is used to indicate transportation information for a guest.

**DepartureTransport** - The details regarding how and when this guest will Depart. e.g., flight info.

**TransportInfo** - A TransportInfo object is used to indicate transportation information for a guest.

**Guest Counts** - The total number of guests will equal the sum of the Count attribute of all GuestCount objects in this collection and assumes that the AgeQualifyingCode will be used to determine how each guest should be billed. Each roomstay and service has its own GuestCount attribute to help determine billing, headcount, etc.

**ResProfiles** – A collection of ResProfile objects. The collection of all profiles associated with any part of the reservation. This could include the reservation in its entirety (e.g., a travel agent), one or more guests, or any GuaranteeInfo object. Which profiles are associated with which part is determined by each object's ResProfileRPHs collection or ResProfileRPH attribute.

**ResProfile** - The ResProfile object derives from the Profile object and inherits all of its attributes. The ResProfile object identifies a Profile associated with a reservation. The Profile object is defined in the CRS Profile Synchronization standard.

- *ResProfileRPH* - This is a reference placeholder, used as an index for this object

**RoomStays** - A collection of RoomStay objects. Room stays associated with this reservation.

**RoomStay** - The RoomStay object is a representation of one reserved room for a contiguous period. If there is a change of room or room occupancy, a new RoomStay object must be created and added to the collection. A simple example follows: A guest is staying in a standard king room for three nights (Tuesday the 3rd through Thursday the 5th). His wife and two children are joining him for the weekend (Friday the 6th through Saturday the 7th) when they will stay in a suite. This reservation will have two room stays. The first will be for one standard king room, starting on the 3rd for three nights. The second will be for one suite, starting on the 6th for two nights.

- *ReservationStatusType* - An enumerated type that defines the status of the reservation for this service. Values are:Reserved,Requested,Request denied,No-show,Cancelled,In-house,Checked out,Waitlisted.
- *RoomStayRPH* - This is a reference placeholder, used as an index for the RoomStay object.
- *RoomType* - Long form textual description of room type
- *RoomTypeCode* – Code representing the room inventory from which this roomstay is taken.
- *RoomID* – The identification of the specific room for this room stay.
- *InvBlockCode* - Code representing the inventory block from which this room stay is taken.
- *MarketSegmentCode* – The market segment code for this room stay.
- *PromoCode* – The promotion code for this room stay.
- *TACommissionPlanCode* – The Travel Agent Commission Plan Code for this room stay
- *NumberOfRooms* – The number of rooms that are represented in this room stay. By convention, unless it is agreed otherwise by trading partners, this will be one (1).

**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility. (HotelReference.xsd)

- *ChainCode* - The code for the Chain to which this hotel belongs.
- *BrandCode* - The Brand or Flag under which this hotel operates .
- *HotelCode* - A unique code within the Chain and/or Brand.
- *AreaCode*
- *HotelCityCode*

**TimeSpan** - The time span covered by this service.

**DateTimeSpan** - The DateTimeSpan represents specific durations of time. Time periods, i.e. specific durations of time, can be represented by supplying two items of information: a start instant and a duration or a start instant and an end instant or an end instant and a duration.

**GuestCounts** - Categorization of all guests in this reservation. The total number of guests will equal the sum of the Count attribute of all GuestCount objects in this collection and assumes that the AgeQualifyingCode will be used to determine how each guest should be billed. Each roomstay and service has its own GuestCount attribute to help determine billing, headcount, etc.

**GuestCount** - The GuestCount object indicates the number of guests attached to a specific reservation, room stay, or service. The total number of guests in a room is the total of the Count of guests in all AgeQualifyingCodes.

- *AgeQualifyingCode* – A code representing the category into which this guest count falls. Example: Senior, Child, Infant, etc. This code is agreed between trading partners.
- *Count* – The count of guests within the AgeQualifyingCode.

**RatePlans** - A collection of RatePlan objects indicating the rate plans in effect during this room stay

**RatePlan** - One of the rate plans in effect for a room stay.

- *RatePlanRPH* - This is a reference placeholder used as an index for rateplans in effect for this room stay
- *RatePlanCode* - The representation of the rate plan under which this service was booked

**TimeSpan** - The time span covered by this service.

**DateTimeSpan** - The DateTimeSpan represents specific durations of time.

**Rates** - A collection of Rate objects indicating the rates (prices) in effect during this rate plan

**Rate** - One of the rates in effect for a rate plan within a room stay.

- *RateRPH* - This is a reference placeholder used as an index for rates in effect for this rate plan.
- *AmountGuaranteed* - Whether the amount for this rate is guaranteed instead of quoted and subject to change. Values: 0 = False or No, and 1 = True or Yes
- *IsComp* - IsComp : Boolean Whether the amount for this rate was comped out (zero), such as a manager's free night for an old friend.
- *IsOverride* - IsOverride : Boolean Whether the amount for this rate was overridden by the person making the reservation.
- *IsTaxExempt* - IsTaxExempt : Boolean Whether the amount for this rate has a tax exemption status.

**Amount** - The actual payment amount in the appropriate type of currency.

- *CurrencyCode* – The ISO code representing the currency in which the amount is listed.

**RateBasisUnits** - Number of units for this rate basis

- *RateBasisTimeUnitType* – RateBasisTimeUnitType : TimeUnitType Interval of time upon which the RateBasisUnits are based

**TimeSpan** - The time span covered by this service.

**DateTimeSpan** - The DateTimeSpan represents specific durations of time.

**AnticipatedSettlementRPHs** - A collection of unsigned integers serving as reference placeholders, and used as an index for the anticipated billing/payment arrangements for this service.

**AnticipatedSettlementRPH** - This is a reference placeholder used as an index for the settlement or payment arrangements associated with this stay.

**ResGuestRPHs** - A collection of unsigned integers serving as reference placeholders, and used as an index identifying which guests occupy this room. Note that the Add operation on this collection should create a new index (UInt), then copy the passed parameter into it. Application code can call the Add operation repeatedly, using the same variable as the parameter, and just changing its value each time.

**ResGuestRPH** - This is a reference placeholder, used as an index for this guest in this reservation. In the ResGuest object it is used like all other RPH attributes to send the delta of a reservation. It is used by the RoomStay and Service objects to indicate which guests are associated with that room stay or service.

**SelectedMembershipRPHs** - A collection of unsigned integers that reference the RPH (Reference Place Holder) attribute in the SelectedMembership object. Note that the Add operation on this collection should create a new index (UInt), then copy the passed parameter into it. Application code can call the Add operation repeatedly, using the same variable as the parameter, and just changing its value each time.

**SelectedMembershipRPH** - A collection of unsigned integers that reference the RPH (Reference Place Holder) attribute in the SelectedMembership object.

**ResCommentRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the ResComment object. The ResCommentRPH attribute in the ResComment object is an indexing attribute that identifies a specific comment attached to the reservation.

**ResCommentRPH** - A collection of unsigned integers serving as reference placeholders, and used as an index for the comments associated with this service.

**SpecialRequestRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the SpecialRequest object. The SpecialRequestRPH attribute in the Special Request object is an indexing attribute that identifies the special requests of a specific guest or of the reservation.

**SpecialRequestRPH** - This is a reference placeholder used as an index for a special request to be associated with this reservation.

**ServiceRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the Service object. The ServiceRPH attribute in the Service object is an indexing attribute that identifies the services attached to a specific guest or to the reservation.

**ServiceRPH** - This is a reference placeholder used as an index for a service to be associated with this stay

**GuaranteeDeposits** - A collection of GuaranteeDeposit objects. This links all guarantees and/or deposits with Service and RoomStay objects

**GuaranteeDeposit** - Refers to the PaymentDue information. This attribute is used if the GuaranteeType = Deposit.

**Deposit** - Deposit : PaymentsInstructions Refers to the PaymentsInstructions information.

**Guarantee** - Type of Guarantee.

**CancelPenalties** - The information about cancellation penalties that relate to this service.

**CancelPenalty** - A CancelPenalty object specifies textual information pertaining to cancellation penalties that apply to a product or set of products.

- *CancelByDate* – The date on which a payment or guarantee must be received to avoid the penalty.

**Amount** - The actual amount in the appropriate type of currency.

**PaymentInstructions** - The information about payment instructions that relate to this service.

**PaymentInstruction** - A PaymentInstruction object specifies textual information pertaining to policies and/or restrictions that apply to a product or set of products.

- *PaymentMethodType* - An enumerated type that defines a method of payment for this stay. Values are: Cash, Credit Card, Debit Card, Voucher, Check, Other.
- *ResCreditCardRPH* – The pointer to the Reservation Credit Card that will be used for this payment.

**PaymentDue** - This object defines the terms of payment for the reservation, including the due date, the amount due, and any cancellation terms.

- *DueDate* - This is the actual due date for the payment

- *CancellIfNotReceived* -: Boolean. If Yes, the reservation is to be canceled automatically if payment not received. If No, and payment is not received, the reservation is not canceled, but is no longer guaranteed for late arrival. Values: False or No, and True or Yes.

**Amount** - The actual payment amount in the appropriate type of currency.

- *CurrencyCode* – The ISO code representing the currency in which the amount is listed

**SelectedMemberships** - A collection of SelectedMembership objects to be applied to any part of the reservation (any service or roomstay). Selected memberships associated with this reservation that provides a list of reward programs which may be credited with points accrued from the guest's stay. Which memberships are to be applied to which part is determined by each object's SelectedMembershipRPHs collection.

**SelectedMembership** - The SelectedMembership object identifies the frequent customer reward program and (optionally) indicates points awarded for stay activity.

- *SelectedMembershipRPH* - This is a reference placeholder used as an index for the membership program to be associated with this stay
- *ProgramCode* - The code or name of the membership program (e.g., 'Hertz', 'AAdvantage', etc).
- *BonusCode* - The code or name of the bonus program. BonusCode can be used to indicate the level of membership (e.g.; Gold Club, Platinum member, etc).
- *AccountID* - The account identification number for this particular member in this particular program.
- *PointsEarned* - The total number of points earned through the selected membership.

**AnticipatedSettlements** – A collection of AnticipatedSettlement objects, including credit card or voucher information used to indicate anticipated billing/payment arrangements for this service. This is the collection of all anticipated settlements for any part of this reservation (one or more roomstays or one or more services). Which settlements are associated with which part is determined by each object's AnticipatedSettlementRPHs collection.

**AnticipatedSettlement** - An AnticipatedSettlement object is used by the RoomStay and Service objects to indicate anticipated billing and payment arrangements.

- *SettlementType* - Values: Direct bill, City ledger, Bill to master.
- *AnticipatedSettlementRPHs* - A collection of unsigned integers serving as reference placeholders, and used as an index for the anticipated billing/payment arrangements for this room stay. NOTE: Where new items are to be added to any RPH collection, the OTA X-Path operation should be used.
- *MasterRoomStayRPH* - This is a reference placeholder, used as an index for the RoomStay object that is the master (e.g., SettlementType = BillToMaster).

**ApplicableRevenueCodes** - Collection of strings indicating the revenue codes this settlement arrangement applies to.

**RevenueCode** - Code defining the category to be used for accumulation of revenue

**SpecialRequests** – A collection of SpecialRequest objects. The collection of all special requests associated with any part of the reservation (the reservation in its entirety, one or more guests, or one or more room stays). Which special requests belong to which part is determined by each object's SpecialRequestRPHs collection.

**SpecialRequest** - The SpecialRequest object indicates special requests for a particular guest, service or reservation. Each of these may be independent of any that are tied to the profile (see Profile Synchronization standard).

- *SpecialRequestRPH* - This is a reference placeholder used as an index for a special request to be associated with this reservation
- *RequestCode* - Identifies a request for this reservation from the Special Request code list for Hotels, Amenities, etc. Refers to the Hotel Descriptive Content Notification in the CRS Availability, Rate and Inventory standard
- *RequestComments* - Further information about this request

**Services** - A collection of Service objects. This is the collection of all services associated with any part of this reservation (the reservation in its entirety, one or more guests, or one or more room stays). Which services are attributable to which part is determined by each object's ServiceRPHs collection.

**Service** - A Service object represents a non-room product provided to guests. Service products may have associated inventory and charges.

- *ServicePricingType* - An enumerated type that defines how a service is priced. Values: Per stay, Per person, Per night, Per person per night, Per use.
- *ReservationStatusType* - An enumerated type that defines the status of the reservation for this service
- *ServiceRPH* - This is a reference placeholder used as an index for a special request to be associated with this reservation.
- *ServiceInventoryCode* - The representation of the specific service being reserved
- *RatePlanCode* - The representation of the rate plan under which this service was booked.
- *InventoryBlockCode* - The representation of the block code under which the inventory for this service is categorized.
- *PriceGuaranteed* : Boolean - Whether the price for this service is guaranteed or quoted and subject to change. Values: False or No, (Price NOT guaranteed) and True or Yes, (Price is guaranteed.)
- *Inclusive* – Boolean - Whether the price for this service is included in the room rate. Values: False or No, and True or Yes.



- **Quantity** - The number of tickets, rounds of golf, etc. Also serves as the number of persons when pricing class is per person or per person per night.

**AnticipatedSettlementRPHs** - A collection of unsigned integers serving as reference placeholders, and used as an index for the anticipated billing/payment arrangements for this service.

**AnticipatedSettlementRPH** - This is a reference placeholder used as an index for the settlement or payment arrangements associated with this service.

**ResCommentRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the ResComment object. The ResCommentRPH attribute in the ResComment object is an indexing attribute that identifies a specific comment attached to the reservation.

**ResCommentRPH** - This is a reference placeholder used as an index for the comments associated with this service.

**SelectedMembershipRPHs** - A collection of unsigned integers that reference the RPH (Reference Place Holder) attribute in the SelectedMembership object. Note that the Add operation on this collection should create a new index (UInt), then copy the passed parameter into it. Application code can call the Add operation repeatedly, using the same variable as the parameter, and just changing its value each time.

**SelectedMembershipRPH** - This is a reference placeholder used as an index for the SelectedMembership objects associated with this service.

**TimeSpan** - The time span covered by this service.

**DateTimeSpan** - The DateTimeSpan represents specific durations of time.

**Price** - The price charged for this service. Whether it is the total or per person/per night/per use charge depends on the setting of the PricingType attribute. Each system vendor pair must decide what to do if the price doesn't match what its system calculates the correct price should be.

- **CurrencyCode** – The ISO code representing the currency in which the amount is listed

**GuaranteeDeposits** - A collection of GuaranteeDeposit objects. This links all guarantees and/or deposits with Service and RoomStay objects

**GuaranteeDeposit** - Refers to the PaymentDue information. This attribute is used if the GuaranteeType = Deposit.

**Deposit** - Deposit : PaymentsInstructions Refers to the PaymentsInstructions information.

**Guarantee** - Type of Guarantee.

**CancelPenalties** - The information about cancellation penalties that relate to this service.

**CancelPenalty** - A CancelPenalty object specifies textual information pertaining to cancellation penalties that apply to a product or set of products.

- *CancelByDate* – The date on which a payment or guarantee must be received to avoid the penalty.

**Amount** - The actual amount in the appropriate type of currency.

- *CurrencyCode* – The ISO code representing the currency in which the amount is listed

**PaymentInstructions** - The information about payment instructions that relate to this service.

**PaymentInstruction** - A PaymentInstruction object specifies textual information pertaining to policies and/or restrictions that apply to a product or set of products.

- *PaymentMethodType* - An enumerated type that defines a method of payment for this stay. Values are: Cash, Credit Card, Debit Card, Voucher, Check, Other.
- *ResCreditCardRPH* – The pointer to the Reservation Credit Card that will be used for this payment.

**PaymentDue** - This object defines the terms of payment for the reservation, including the due date, the amount due, and any cancellation terms.

- *DueDate* - This is the actual due date for the payment
- *CancelIfNotReceived* -: Boolean. If Yes, the reservation is to be canceled automatically if payment not received. If No, and payment is not received, the reservation is not canceled, but is no longer guaranteed for late arrival. Values: False or No, and True or Yes.

**Amount** - The actual payment amount in the appropriate type of currency.

- *CurrencyCode* – The ISO code representing the currency in which the amount is listed

**GuestCounts** - Categorization of all guests in this reservation. The total number of guests will equal the sum of the Count attribute of all GuestCount objects in this collection and assumes that the AgeQualifyingCode will be used to determine how each guest should be billed. Each roomstay and service has its own GuestCount attribute to help determine billing, headcount, etc.

**GuestCount** - The GuestCount object indicates the number of guests attached to a specific reservation, room stay, or service. The total number of guests in a room is the total of the Count of guests in all AgeQualifyingCodes.

- *AgeQualifyingCode* - A code representing a business rule that determines the charges for a guest based upon age range. For example, Adult, Child, Senior, Child With Adult, Child Without Adult, etc. This attribute allows for an increase in rate by occupant class.
- *Count* - The number of guests in one AgeQualifyingCode.

**OriginalRoutingHops** – A collection of RoutingHop objects. This details the path that the original request or notification traveled.

**RoutingHop** - A RoutingHop object conveys information about the path that this notification update took; e.g., how many and which systems it passed through.

- *SystemCode* – A system code agreed upon by trading partners
- *LocalRefID* - Identifier within the System that refers to this specific document (reservation)
- *TimeStamp*- The date and time that the reservation passed through a routing hop
- *Comment* – Information that the system adding this routing hop may use upon return of the data in a future response
- *SequenceNumber* - Sequential number assigned to hops, Base 0
- *Data* - This attribute is provided so that each system can put in whatever data it would like (e.g., auditing information)

**ResCommentRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the ResComment object. The ResCommentRPH attribute in the ResComment object is an indexing attribute that identifies a specific comment attached to the reservation.

**ResCommentRPH** - This is a reference placeholder used as an index for the comments associated with this reservation.

**ResProfileRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the ResProfile object. The ResProfileRPH attribute in the ResProfile object is an indexing attribute that identifies other profiles attached to the reservation.

**ResProfileRPH** - This is a reference placeholder, used as an index for this object.

**SpecialRequestRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the SpecialRequest object. The SpecialRequestRPH attribute in the Special Request object is an indexing attribute that identifies the special requests of a specific guest or of the reservation.

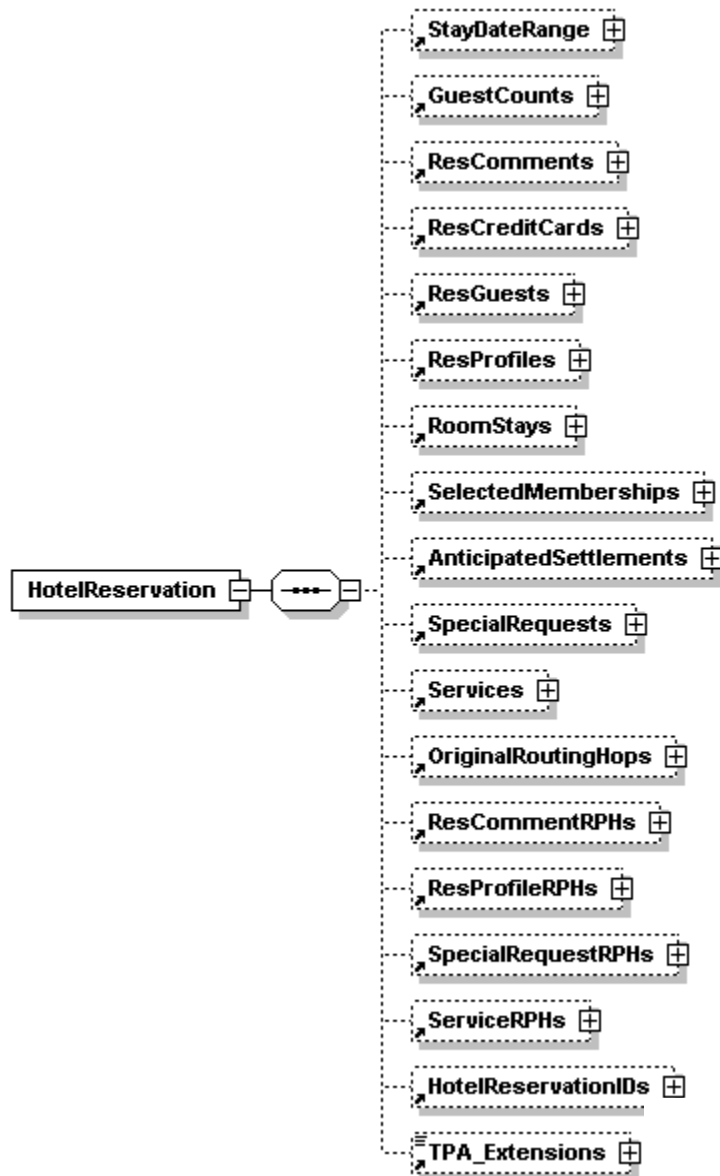
**SpecialRequestRPH** - This is a reference placeholder used as an index for a special request to be associated with this reservation.

**ServiceRPHs** - A collection of unsigned integers that reference the RPH (Reference Place holder) attribute in the Service object. The ServiceRPH attribute in the Service object is an indexing attribute that identifies the services attached to a specific guest or to the reservation.

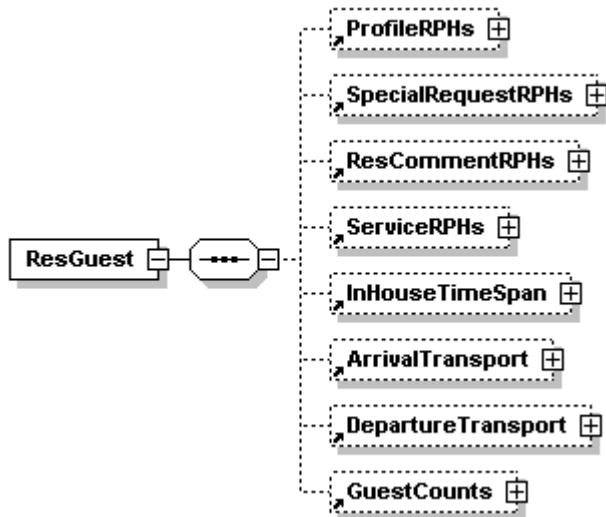
**ServiceRPH** - This is a reference placeholder used as an index for a service to be associated with this stay

## 5.25 Hotel Reservation Schema Diagrams

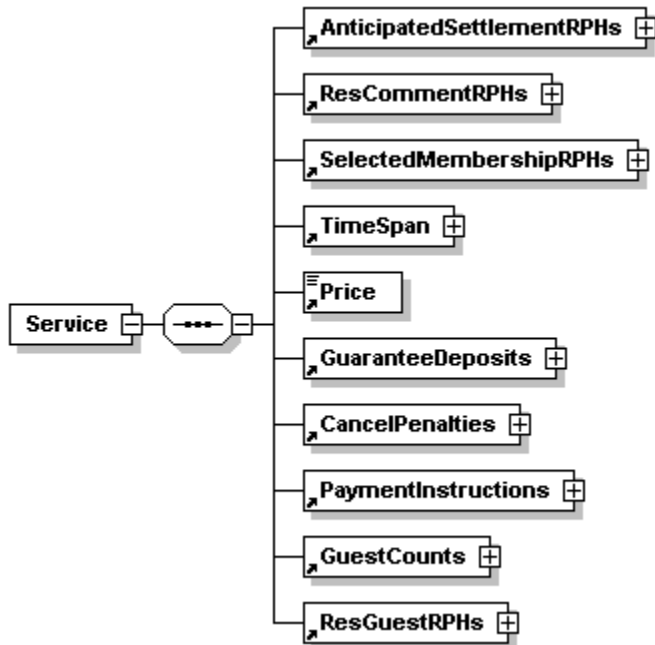
### Hotel Reservation



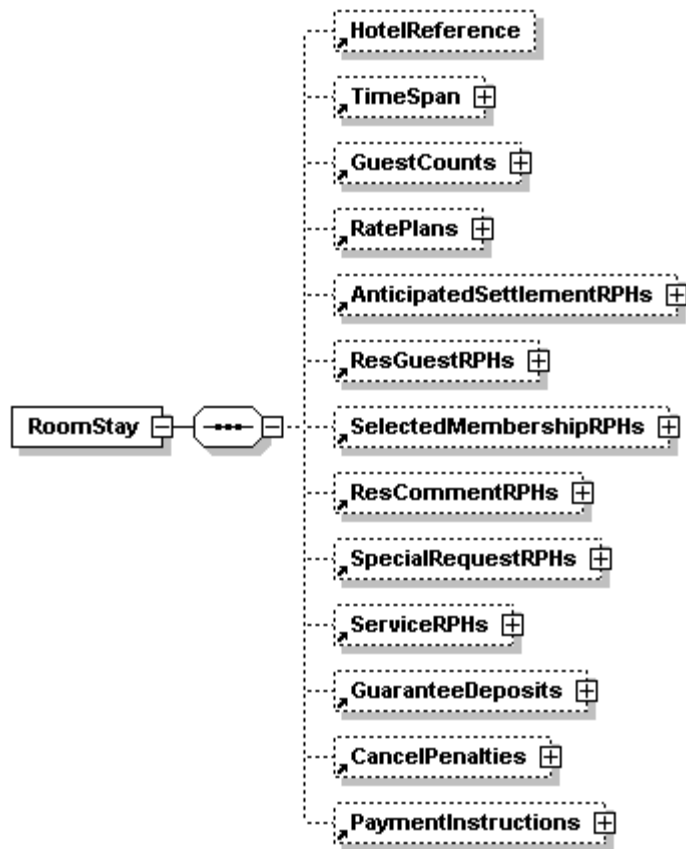
## ResGuest



## Service



## Room Stay



# **OTA Get Message**

## **Request/Response message specifications**





## 5.26 Get Message Messages

The OTA version 2001B specification of Get Message provides a request/response pair of messages to permit a system that normally receives notifications to ask for a re-transmission of a message.

The business model assumes that the requesting system either receives messages that are numbered sequentially, and may ask for a message to be re-sent. In the event that the receiving system receives a message that is not in contiguous numerical sequence, this message set can be used to retrieve missing messages, or to ask for a retransmission of data that for some reason was not cleanly received.

The originating system will send a request using the OTA\_GetMsgRQ message. The receiving system will acknowledge and respond with the OTA\_GetMsgRS message.

**OTA\_GetMsgRQ** – Sends a request to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_GetMsgRS** - Returns acknowledgement that the request has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed. It will additionally include the re-submission of the requested data as it was sent in its original form. Since the request could be for any original message, the response data is include in MessageContent.

**Note:**

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description

### 5.27 Get Message Request Message (OTA\_GetMsgRQ)

The root tag of OTA\_GetMsgRQ contains the standard payload attributeGroup OTA\_PayloadStdAttributes, found in all OTA payload documents.

**OTA\_GetMsgRQ** - This is the main Request element.

**OTA\_PayloadStdAttributes**- includes standard attributes of the root tag of all OTA messages. These attributes are defined in the OTA Version 2 Master Document.

**Message**- The message for which this request is based.

- *MessageType* – This attribute describes the message type on which this request is based
- *SeqNmbr*- Attribute describing the sequence number for that type of message to be re-sent.

**OriginalPayloadStdAttributes**– Where known, this data will help identify the original message. The attributes are identical to those described in the OTA Version 2 Infrastructure document for PayloadStdAttributes.

**Request** –.

### 5.28 Sample XML Message using OTA\_GetMsgRQ XML Schema.

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_GetMsgRQ EchoToken="V12345" Version="1" TimeStamp="2002-01-15T15:35:18" Target="Production"
  xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opentravel.org/OTA OTA_GetMsgRQ.xsd">
  <Uniqueld Type="PGNR" Id="2" Instance=""/>
  <Message MessageType="PGNR">
    <Request RequestCode="PGNR" ReasonForRequest="REVIEW" ConfirmationID="912345678"
      ReservationID="1" >
      <HotelReference HotelCode="03071" ChainCode="BW" BrandCode=""/>
    </Request>
  </Message>
</OTA_GetMsgRQ>
```

## 5.29 Get Message Response Message (OTA\_GetMsgRS)

The root tag of OTA\_GetMsgRS uses the *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_GetMsgRS** - Main Response element.

**OTA\_PayloadStdAttributes** - See above

**Message** - The message included in this response data.

- *MessageType* – This attribute describes the message type for this response.
- *ResponseValue*- Attribute describing the sequence number for that type of message to be re-sent.

**OriginalPayloadStdAttributes** – See Above.

**MessageContent** – The original message including all tags and data.

**Success**

**Warnings**

**Warning**

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

**Errors**

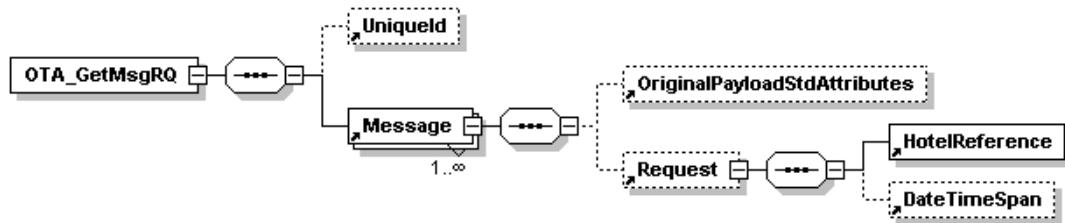
**Error**

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

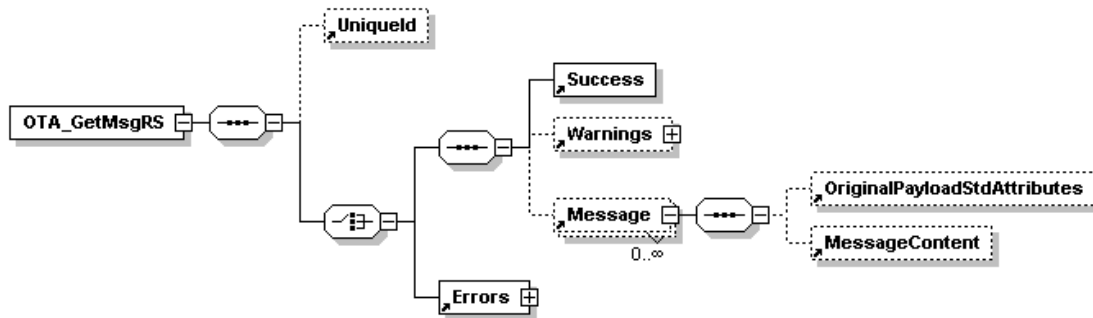
## 5.30 Sample XML Message using the OTA\_GetMsgRS XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_GetMsgRS EchoToken="V12345" Version="1" TimeStamp="2001-10-31T14:19:41"
  xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opentravel.org/OTA OTA_GetMsgRS.xsd">
  <UniqueId Type="PGNRR" Id="1" Instance="7315990"/>
  <Success/>
</OTA_GetMsgRS>
```

### 5.31 XML Schema Diagram for Get Message Request



### 5.32 XML Schema Diagram for Get Message Response



### 5.33 Revision History for OTA\_GetMsgRQ/RS

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
May 6, 2001	0.1	Initial Draft.
October 30, 2001	0.2	Updated.

## **OTA Get Message Information**

### **Request/Response message specifications**



### 5.34 Get Message Information Messages

The OTA version 2001B specification of Get Message provides a request/response pair of messages to permit a system that normally receives notifications to ask for a re-transmission of a message.

The business model assumes that the requesting system either receives messages that are numbered sequentially, and may ask for a message to be re-sent. In the event that the receiving system receives a message that is not in contiguous numerical sequence, this message set can be used to retrieve missing messages, or to ask for a retransmission of data that for some reason was not cleanly received.

The originating system will send a request using the OTA\_GetMsgInfoRQ message. The receiving system will acknowledge and respond with the OTA\_GetMsgInfoRS message.

**OTA\_GetMsgInfoRQ** – Sends a request to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_GetMsgInfoRS** - Returns acknowledgement that the request has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed. It will additionally include the original Payload Standard Attributes for the messages to which the response applies.

**Note:**

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description

### 5.35 Get Message Information Request Message (OTA\_GetMsgInfoRQ)

The root tag of OTA\_GetMsgInfoRQ contains the standard payload attributeGroup OTA\_PayloadStdAttributes, found in all OTA payload documents.

**OTA\_GetMsgInfoRQ** - This is the main Request element.

**Message-** The message for which this request applies

- *StartSeqNmbr*– This attribute describes beginning sequence number for this request.
- *EndSeqNmbr*- Attribute describing the ending sequence number for that type of message.

### 5.36 Get Message Information Response Message (OTA\_GetMsgRS)

The root tag of OTA\_GetMessageRS uses the external *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_GetMsgInfoRS** - Main Response element.

**OTA\_PayloadStdAttributes** - See above

**Message** - A repeating element describing the associated messages.

- *MessageType* – This attribute describes the message type for this response.
- *ResponseValue*- Attribute describing the sequence number for that type of message to be re-sent.

**OriginalPayloadStdAttributes** – The Payload Standard Attributes associated with the original transmission of these messages

**Success**

**Warnings**

**Warning**

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warnin
- *DocURL*
- *Status*
- *Tag*

**Errors**

**Error**

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning

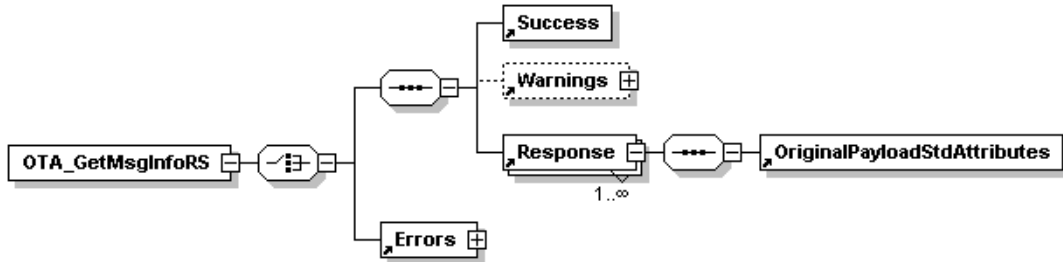


- *DocURL*
- *Status*
- *Tag*

### 5.37 XML Schema Diagram for Get Message Information Request



### 5.38 XML Schema Diagram for Get Message Information Response



### 5.39 Revision History for OTA\_GetMsgInfoRQ/RS

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
May 6, 2001	0.1	Initial Draft.
October 30, 2001	0.2	Upgrade.



## **OTA Commission**

### **Request/Response message specifications**



## 5.40 Commission Messages

The OTA version 2001B specification of Commissions provides a request/response pair of messages to support the functionality of updating other systems with commissions to be paid. The message set assumes a push model, with the reporting system (typically a Property Management System – PMS) pushing the data to the Management Company or Central Reservation Office that is responsible for paying the commissions, or one of these entities pushing the data to a consolidator contracted to pay commissions.

In the push model, the originating system will send a report using the OTA\_CommNotifRQ message. The receiving system will acknowledge its receipt of that report using the OTA\_CommNotifRS message. All message responses include the request identification. Responses may be returned in any order.

**OTA\_CommNotifRQ** – Sends a report to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_CommNotifRS** - Returns acknowledgement that the report has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed.

### Note:

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description

### 5.41 Commission Notification Request Message (OTA\_CommNotifRQ)

The root tag of OTA\_CommNotifRQ contains the standard payload attributeGroup OTA\_PayloadStdAttributes, found in all OTA payload documents.

**OTA\_CommNotifRQ** - This is the main Request element.

**OTA\_PayloadStdAttributes** - includes standard attributes of the root tag of all OTA messages. These attributes are defined in the OTA Version 2 Infrastructure Document.

**CommissionEvents** - CommissionEvents are prepared by the PMS as a result of a guest stay, or by systems such as an activity vendor's system as a result of a completed activity. The report is then sent to the entity responsible for making payments to those parties that are to receive commissions for the transactions.

**CommissionEvent** - The CommissionEvent element is used to identify the commission to be paid to a central server. The Commission Event is initiated on the schedule that the hotel typically pays commissions (e.g.: following night audit), and a CommissionEvent occurs for every check out.

- *ConfirmationID* – The PMS unique identifier for this reservation that the original booking agent was told (or generates), and is a confirmation from the source of the reservation, as is the when a CRO does the booking.
- *ConfirmationOriginatorCode* – Identifying code of the originating system of this reservation. This is an optional attribute.
- *CommissionOriginatorCode* – The originator's code of the commission (e.g. hotel, activity, etc.) This attribute is tied to the reservation (ReservationID) and identifies the originator of the commission i.e.: hotel, chain, activity vendor, etc. This is an optional attribute.
- *ReservationID*- The PMS unique identifier for this reservation. There may be multiple reservations with the same numberso ConfirmationID is a separate attribute. This is an optional attribute.
- *OriginalBookingDate* – Datetime of the creation of this reservation. This is an optional attribute.
- *MarketCode* - This code is associated with this reservation according to market segmentation according to rules determined by the trading partners.
- *PreferredPaymentCurrency*
- *BookingSource* – The entity that made the booking. This is an optional attribute.
- *NumberOfRooms*
- *RoomInventoryCode*
- *PromotionCode*
- *InitialRate*

**UniqueID**- Unique identifier assigned to an individual customer profile.

**StayDateRange** - Total time span covered by this reservation (earliest arrival to latest departure).

**GuestNames** - A collection of PersonName elements associated with the commissionable event. PersonName elements are consistent with the name stored in the reservation or guest stay information and can be used to reconcile records with the original reservation.

**PersonName** – As defined in the Profile Specification

**ProfileCertification** - The ProfileCertification element identifies the travel agency to be paid by providing the IATA number or ARC certifying number. (Implementation note: This element is optional, but if it is not set, the Profile element must be present.)

- *CertificationType* – IATA, ARC, CLIA, etc.
- *CertificationID* – Unique Identifier associated with this payee within the certifying

**ProfileReference**- A reference to the travel agent's Profile. (Implementation note: This element is optional, but if it is not set, the ProfileCertification element must be present.)

**Profile** – As defined in the Profile Specification

**Payee** - Name of Payee for commission. Optional where profiles are used.

**PayeeAddress** - Payee Address for commission. This is optional where profile is used.

**Commissions** - A collection of multiple Commission elements.

**Commission** - The Commission element contains the data required for determining the commission amount to be paid to an agent. The Commission Amount is calculated by applying a Percent to the CommissionableAmount and adding any flat amount that may apply.

- *Status Type*- This is a mandatory, enumerated attribute that indicates the status of this commission event. Full, Partial, Non-paying, Cancel, No-show, Adjustment.
- *Percent* - Percent used to calculate the commission amount. Clients may set this attribute and/or the FlatCommission attribute. Percent is represented using the base datatype, decimal.
- *ReasonCode* - Code determined between vendors that indicates the reason for this commission: (i.e.; the commissionable event that determines the rates in effect during this room stay.) e.g.: special event, banquet meeting, convention, group, other, etc.
- *BillToID* – Identification of the entity to which the payment of this commission will be billed

**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility.

**UniqueID** – Unique identifier assigned to an individual customer profile

**CommissionableAmount** - Basis amount used to determine the commission amount to be paid to the agent. This element is optional (but if is not set, the Profile element must be present) in the form of Currency as defined in the OTA Version 2 Master Document

**PrepaidAmount** - Amount pre-deducted by the agent. This amount would be calculated and subtracted from the amount paid to the agent. This field would also be

used where the hotel has already paid a commission, but is notifying the central system of that payment.

- *CurrencyCode* – The ISO standard code for the currency in which an amount is represented.

**FlatCommission** - Flat rate amount to be paid to agent if applicable. Clients may set this element and/or the *CommissionPercent* element. This is an element in the form of *Currency* as defined in the OTA Version 2 Infrastructure Document.

**CommissionPayableAmount** - This amount overrides all other calculation of commissions. If this value is included, it will be the amount to be paid on this transaction. This is an element in the form of *Currency* as defined in the OTA Version 2 Master Document.

- *CurrencyCode* – The ISO standard code for the currency in which an amount is represented.

**Comment** - A text field that allows for additional explanation for the reason; e.g.: a percent deducted from the commission, guest paid with a coupon, etc.



## 5.42 Commission Notification Response Message (OTA\_CommNotifRS)

The root tag of OTA\_CommNotifRS uses the *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_CommNotifRS** - Main Response element.

**OTA\_PayloadStdAttributes**- includes standard attributes of the root tag of all OTA messages. These attributes are defined in the OTA Version 2 Master Document.

### Success

#### Warnings

##### Warning

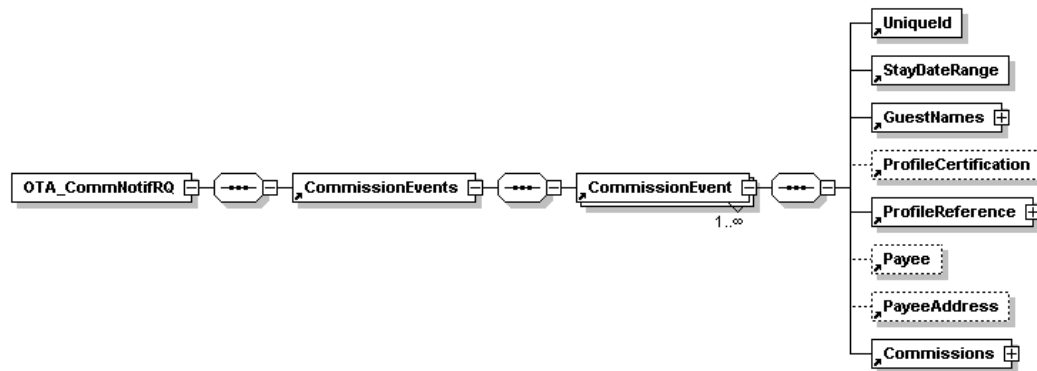
- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

#### Errors

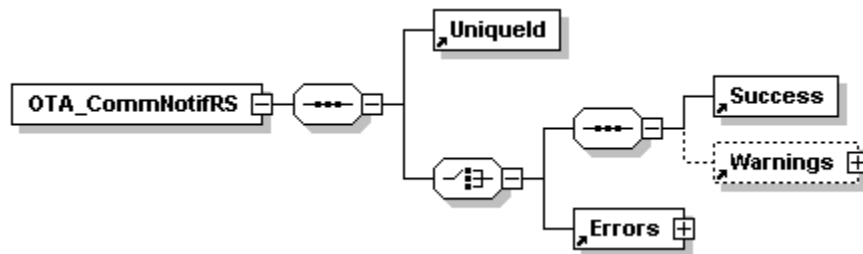
##### Error

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

### 5.43 XML Schema Diagram for Commissions Notification Request



### 5.44 XML Schema Diagram for Commissions Notification Response



### 5.45 Revision History

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
May 4, 2001	0.1	Initial Draft.
October 30, 2001	0.2	Updated.

# **OTA Stay Information Notification**

## **Request/Response message specifications**



### 5.46 Stay Information Notification Messages

The OTA version 2001B specification of Stay Information Notification provides a request/response pair of messages to support the functionality of updating other systems with Guest Stay Information. The message set assumes a push model, with the reporting system (typically a Property Management System – PMS) pushing the data to the Management Company or Central Reservation Office that is responsible for accumulating the information

In the push model, the originating system will send a report using the OTA\_StayInfoNotifRQ message. The receiving system will acknowledge its receipt of that report using the OTA\_StayInfoNotifRS message. All message responses include the request identification. Responses may be returned in any order.

**OTA\_StayInfoNotifRQ** – Sends guest stay information to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_StayInfoNotifRS** - Returns acknowledgement that the report has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed.

**Note:**

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description

### 5.47 Stay Information Notification Request Message (OTA\_StayInfoNotifRQ)

**OTA\_StayInfoNotifRQ** - This is the main Request element that broadcasts guest stay information to the Central Reservation System or the hotel management company. It is sent unsolicited as an update of guest data and provides a snapshot of individual guest records.

The root tag of OTA\_StayInfoNotifRQ contains the standard payload attributeGroup OTA\_PayloadStdAttributes, found in all OTA payload documents.

- *OTA\_PayloadStdAttributes* - includes standard attributes of the root tag of all OTA messages. These attributes are defined in the OTA Version 2 Infrastructure Document.

**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility.

**StayInfos** - A collection of StayInfo objects.

**StayInfo** - The StayInfo class contains the guest revenue and stay data to be sent to the central server.

- *SequenceNumber* - Indicates the date for which the statistics have been compiled. This typically is not a calendar date, as the hotel may compile its data beginning and ending at a time other than midnight.
- *RoomStayRPH* - Indicates a specified report that has been agreed upon by the trading partners. Examples might be Rooms Report, Restaurant Report, etc.

**LoyaltyPointsAccrual** - A collection of SelectedLoyalty classes that are used to report earned bonuses.

**SelectedLoyalty** - SelectedLoyalty communicates the Loyalty program and points to be credited for a specific stay. The SelectedLoyalty class originates in the CRS Reservation Synchronization standard.

- *ReservationActionType* - Action identifier provided by the property management system or the CRS indicating the type of action to be applied (ex. 0 is DoesNotApply, 1 is New, 2 is Change, 3 is Delete)
- *SelectedLoyaltyRPH* - Index number of loyalty programs in Customer Profile. This is a required attribute.
- *ProgramCode* - The code or name of the membership program (e.g. "Hilton Honors", "Hyatt Gold Passport"). This is a required attribute.
- *BonusCode* - Level of program membership (e.g. "Platinum")
- *AccountID* - The account or membership identifier for this particular member in this particular program. This is a required attribute.
- *PointsEarned* - Credit earned for this stay. This is a required attribute.

**RevenueCategories** - A collection of RevenueCategory classes that detail the categories of revenue data associated with the StayInfo report.

**RevenueCategory** - The classifications of revenue data associated with the StayInfo report. A RevenueCategory provide a way to classify guest financial stay data and analyze guest spending for a certain category (e.g., food and beverage, room, etc.)

- *RevenueCategoryCode* – A representation of a revenue category code. This is a required attribute.

**SummaryAmount** - The total of all RevenueDetail amounts. The sending system may choose to send only this value and not sent the line item detail contained in the RevenueDetail class.

- *CurrencyCode* – ISO 3166 Currency Code.

**RevenueDetail** - The RevenueDetail class contains line item detail about specific revenue transactions.

- *ReferenceID*
- *TransactionDate*
- *RatePlanCode*
- *Description*
- *PMSRevenueCode*
- *Amount*

**Amount** - The actual payment amount in the appropriate type of currency.

**FolioIDs** - Collection of strings (recommended length - 20 characters) indicating the PMS folio identifiers associated with this revenue detail item.

**FolioID** - String(recommended length - 20 characters) indicating the PMS folio identifier associated with this revenue detail item.

**ReservationID** - Unique Identification for the reservation associated with this stay information.

**HotelReservation** - The Reservation class contains the data of the reservation that will be preserved in the Guest Stay History once the guest's stay is complete. The HotelReservation class is described in a separate section within this document.

### 5.48 Stay Information Notification Response Message (OTA\_StayInfoNotifRS)

The root tag of OTA\_StayInfoNotifRS uses the *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_StayInfoNotifRS** - Main Response element.

**OTA\_PayloadStdAttributes**- includes standard attributes of the root tag of all OTA messages. These attributes are defined in the OTA Version 2 Master Document.

#### Success

#### Warnings

##### Warning

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*
- *RecordId*

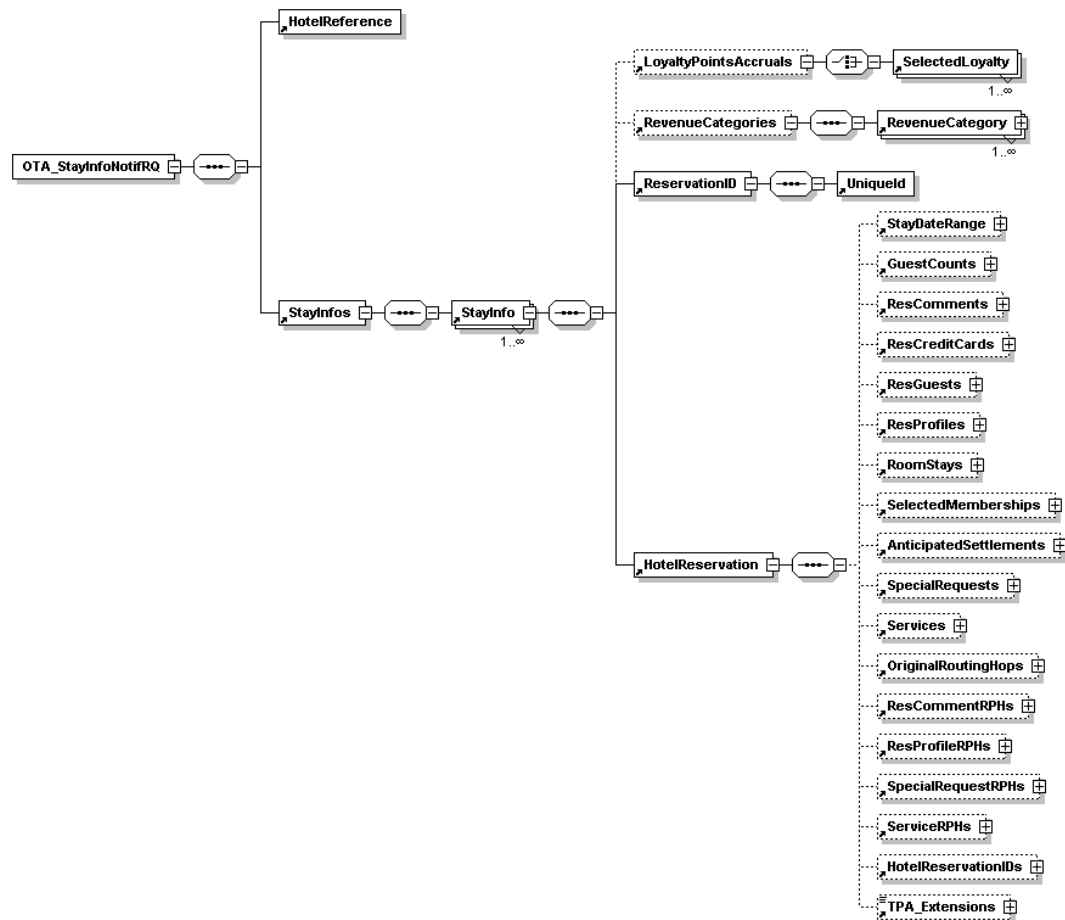
#### Errors

##### Error

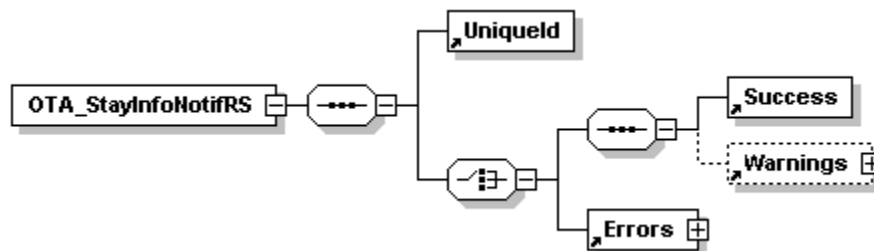
- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*
- *RecordId*



### 5.49 XML Schema Diagram for Stay Information Notification Request



### 5.50 XML Schema Diagram for Stay Information Notification Response



### 5.51 Revision History for OTA\_StayInfoRQ/RS

Date	Version	Revisions
May 7, 2001	0.1	Initial Draft.
October 30, 2001	0.2	Updated.



## **OTA Statistics**

### **Request/Response message specifications**

## 5.52 Statistics Messages

The OTA version 2001B specification of Statistics provides two separate request/response pairs of messages to support the functionality of updating other systems with statistical data. The first message set assumes a push model, with the reporting system (typically a Property Management System – PMS) pushing the data to the Management Company or Central Reservation Office. The second message set assumes a pull model, where the centralized system requests a specific report (as agreed by trading partners) for a specific fiscal date.

In the push model, the originating system will send a report using the **OTA\_StatsNotifRQ** message. The receiving system will acknowledge its receipt of that report using the **OTA\_StatsNotifRS** message.

In the pull model, the central system will request a report using the **OTA\_StatsRQ** message. In this message, the report and fiscal date are identified. The receiving system (typically a PMS) responds with the **OTA\_StatsRS** message, which includes the report itself. All messages assumes the no state, meaning that the querying system will initiate the transaction and expect an response from the queried system. All message responses include the request identification. Responses may be returned in any order.

**OTA\_StatsNotifRQ** – Sends a report to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_StatsNotifRS** - Returns acknowledgement that the report has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed.

**OTA\_StatsRQ** – Sends a request for a report to another system. All the elements and attributes are optional, unless otherwise stated as required.

**OTA\_StatsRS** - Returns the requested report if the request can be processed, or includes Warnings from business processing rules or Errors if the request did not succeed.

### Note:

All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description

### 5.53 Statistics Notification Request Message (OTA\_StatsNotifRQ)

**OTA\_StatsNotifRQ** - This is the main Request element.

- *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as defined in the OTA Version 2 Infrastructure Document.

**Statistics** - The report that sends the guest revenue statistical data and other predetermined information to a central server or management system. It will most likely include multiple Statistic elements.

**Statistic** - The Statistic class contains the guest revenue stay data to be included in the report.

- *FiscalDate* - Indicates the date for which the statistics have been compiled. This typically is not a calendar date, as the hotel may compile its data beginning and ending at a time other than midnight.
- *ReportCode* - Indicates a specific report that has been agreed upon by the trading partners. Examples might be Rooms Report, Restaurant Report, etc.

**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility.

**RevenueCategorySummaries** - A collection of RevenueCategorySummary information sets that report categories of hotel financial statistics.

**RevenueCategorySummary** - A revenue category class that reports the total amount of revenue in a certain category. Categories provide a way to classify hotel financial statistics to show all spending in a specific category (e.g., food and beverage, room, etc.)

- *RevenueCategoryCode* - The representation of a revenue category. The following basic Revenue Categories are suggested:  
RoomRevenue : Total of room revenue at property. FoodRevenue : Total of restaurant and room service revenue at property.  
MeetingRevenue : Total of meeting room revenue at property.  
BarRevenue : Total of bar revenue at property. OtherRevenue : Total of other miscellaneous revenue at property. TotalRevenue : Total of all revenue at property.

**SummaryAmount** - The numerical value of the quantity of money expressed by the currency. The lexical representation of a decimal value consists of a string of digits separated by a period as a decimal indicator, with an optional leading sign to indicate a negative number. If the sign is omitted, "+" is assumed.

- *CurrencyCode* - Indicates the currency in which the Summary Amount is being reported.

**CountCategorySummaries** - A collection of CountCategorySummary information sets that report categories of hotel financial statistics.

**CountCategorySummary** - A class that shows the total of all counts for a certain category. These categories provide a way to show counts such as guests, rooms occupied, etc.

- *SummaryCount* - The total count for that category. The data type is a signed integer.
- *CountCategoryCode* - The representation of a count category. The following basic Count Categories are suggested: Guests : Number of guests at a property. RoomsSingleOcc : Count of rooms with one guest. RoomsMultipleOcc : Count of rooms with more than one guest. RoomsDoNotRent : Count of rooms not available for rent at a property. GtdNoShow : Count of guaranteed no shows for date. NonGtdNoShow : Count of non-guaranteed no shows for date. RoomsAvailable : Count of total rooms available at property.

#### 5.54 Sample XML Message

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_StatsNotifRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_StatsNotifRQ.xsd" EchoToken="12345" TimeStamp="2001-05-31T17:19:42-
05:00" Target="Production" Version="1" SequenceNmbr="0"> <Statistics>
  <Statistic FiscalDate="2000-01-22" ReportCode="2804FOOD">
    <HotelReference ChainCode="HI234" BrandCode="HE223" HotelCode="1234STL"/>
    <RevenueCategorySummaries>
      <RevenueCategorySummary RevenueCategoryCode="4209RMSERV">
        <SummaryAmount CurrencyCode="USD">2184.67</SummaryAmount>
      </RevenueCategorySummary>
      <RevenueCategorySummary RevenueCategoryCode="4212REST">
        <SummaryAmount CurrencyCode="USD">5329.57</SummaryAmount>
      </RevenueCategorySummary>
    </RevenueCategorySummaries>
    <CountCategorySummaries>
      <CountCategorySummary CountCategoryCode="4733ORDRS" SummaryCount="89"/>
      <CountCategorySummary CountCategoryCode="4785PERS" SummaryCount="236"/>
    </CountCategorySummaries>
  </Statistic>
  <Statistic FiscalDate="2000-01-22" ReportCode="2809ROOM">
    <HotelReference ChainCode="HI234" BrandCode="HE223" HotelCode="1234STL"/>
    <RevenueCategorySummaries>
      <RevenueCategorySummary RevenueCategoryCode="2270SGLRM">
        <SummaryAmount CurrencyCode="USD">12276.75</SummaryAmount>
      </RevenueCategorySummary>
      <RevenueCategorySummary RevenueCategoryCode="2246MTGRM">
        <SummaryAmount CurrencyCode="USD">6450.00</SummaryAmount>
      </RevenueCategorySummary>
    </RevenueCategorySummaries>
    <CountCategorySummaries>
      <CountCategorySummary CountCategoryCode="2286GST" SummaryCount="127"/>
      <CountCategorySummary CountCategoryCode="2264MTGCT" SummaryCount="8"/>
    </CountCategorySummaries>
  </Statistic>
</Statistics>
</OTA_StatsNotifRQ>
```

### 5.55 Statistics Notification Response Message (OTA\_StatsNotifRS)

The root tag of <OTA\_StatsNotifRS> uses the *include* <OTA\_v2ent.xsd> that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_StatisticsNotifRS** - Main Response element.

- *OTA\_PayloadStdAttributes* - See above

#### Success

#### Warnings

##### Warning

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*
- *RecordId*

#### Errors

##### Error

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*
- *RecordId*

### 5.56 Sample XML Message

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_StatsNotifRS xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_StatsNotifRS.xsd" EchoToken="12345" TimeStamp="2001-05-31T17:19:42-
05:00" Target="Production" Version="1" SequenceNmbr="0">
  <Success/>
</OTA_StatsNotifRS>
```

### 5.57 Statistics Request Message (OTA\_StatsRQ)

The root tag of OTA\_StatsRQ contains the standard payload attributeGroup OTA\_PayloadStdAttributes, found in all OTA payload documents.

**OTA\_StatsRQ** - Main Response element.

*OTA\_v2ent.xsd* - includes the infrastructure elements including payload standard attributes and the choice of returning an indication of Success, Warnings or Errors.

*OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages as defined in the OTA Version 2 Infrastructure Document.

**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility.

**Report** - The Statistic report for which this request is made.

- *StatisticsDate* - Indicates the date for which the statistics have been compiled. This typically is not a calendar date, as the hotel may compile its data beginning and ending at a time other than midnight.
- *ReportCode* - Indicates a specific report that has been agreed upon by the trading partners. Examples might be Rooms Report, Restaurant Report, etc.

### 5.58 Statistics Response Message (OTA\_StatsRS)

The root tag of OTA\_StatsRS uses the external *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_StatsRS** - Main Response element.

*OTA\_v2ent.xsd* - includes the infrastructure elements including payload standard attributes and the choice of returning an indication of Success, Warnings or Errors.

*OTA\_PayloadStdAttributes* - See above

**Statistics** - The report that sends the guest revenue statistical data and other predetermined information to a central server or management system. It will most likely include multiple Statistic elements.

**Statistic** - The Statistic class contains the guest revenue stay data to be included in the report.

- *FiscalDate* - Indicates the date for which the statistics have been compiled. This typically is not a calendar date, as the hotel may compile its data beginning and ending at a time other than midnight.
- *ReportCode* - Indicates a specific report that has been agreed upon by the trading partners. Examples might be Rooms Report, Restaurant Report, etc.



**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility.

**RevenueCategorySummaries** - A collection of RevenueCategorySummary information sets that report categories of hotel financial statistics.

**RevenueCategorySummary** - A revenue category class that reports the total amount of revenue in a certain category. Categories provide a way to classify hotel financial statistics to show all spending in a specific category (e.g., food and beverage, room, etc.)

- *RevenueCategoryCode* - The representation of a revenue category. The following basic Revenue Categories are suggested:  
RoomRevenue : Total of room revenue at property. FoodRevenue : Total of restaurant and room service revenue at property.  
MeetingRevenue : Total of meeting room revenue at property.  
BarRevenue : Total of bar revenue at property. OtherRevenue : Total of other miscellaneous revenue at property. TotalRevenue : Total of all revenue at property.

**SummaryAmount** - The numerical value of the quantity of money expressed by the currency. The lexical representation of a decimal value consists of a string of digits separated by a period as a decimal indicator, with an optional leading sign to indicate a negative number. If the sign is omitted, "+" is assumed..

- *CurrencyCode* - Indicates the currency in which the Summary Amount is being reported.

**CountCategorySummaries** - A collection of CountCategorySummary information sets that report categories of hotel financial statistics.

**CountCategorySummary** - A class that shows the total of all counts for a certain category. These categories provide a way to show counts such as guests, rooms occupied, etc.

- *SummaryCount* – The count associated with this category. An example would be the number guests, number of rooms occupied, etc.
- *CountCategoryCode* – The representation of a count category. The following basic Count Categories are suggested:  
Guests : Number of guests at a property.  
RoomsSingleOcc : Count of rooms with one guest.  
RoomsMultipleOcc : Count of rooms with more than one guest.  
RoomsDoNotRent : Count of rooms not available for rent at a property.  
GtdNoShow : Count of guaranteed no shows for date.  
NonGtdNoShow : Count of non-guaranteed no shows for date.  
RoomsAvailable : Count of total rooms available at property.

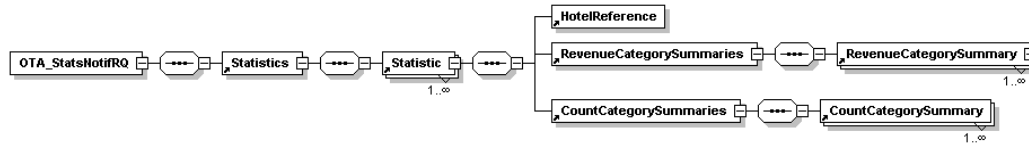
**Success****Warnings****Warning**

- *Type* - Type of warning. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

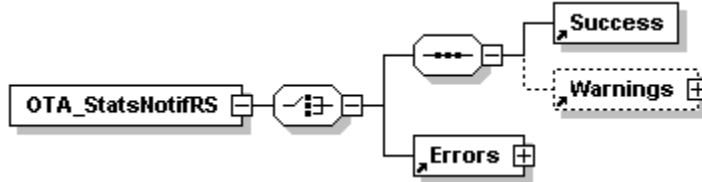
**Errors****Error**

- *Type* - Type of error. Valid values: (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel | ReqFieldMissing )
- *Code* - The code assigned to the warning
- *DocURL*
- *Status*
- *Tag*

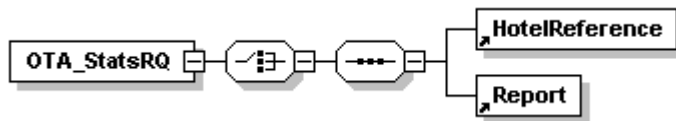
### 5.59 XML Schema Diagram for Statistics Notification Request



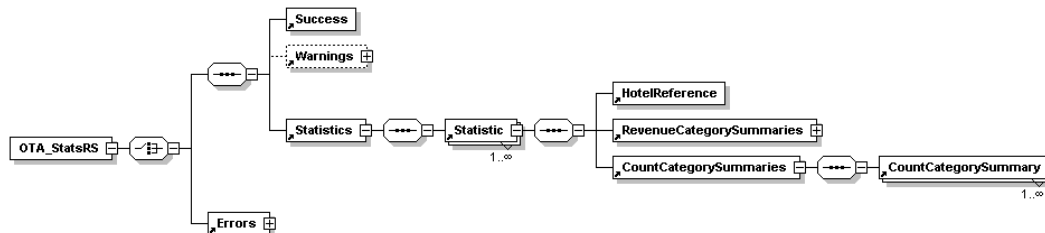
### 5.60 XML Schema Diagram for Statistics Notification Response



### 5.61 XML Schema Diagram for Statistics Request



### 5.62 XML Schema Diagram for Statistics Response



### 5.63 Revision History for OTA\_StatsRQ/RS

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
May 3, 2001	0.1	Initial Draft.
October 30, 2001	0.1	Updated.

# **OTA Create Meeting Profile**

## **Request/Response message specifications**

## 5.64 Meeting Profile Messages<sup>20</sup>

The OTA version 2001B specification of the Meeting Profile provides a request/response pair of messages to support the functionality of creating and updating other systems with Meeting Profile or Group business data. The message sets assumes a push model, with the originating system pushing the data to another system. The originating system would usually be a meeting source, such as a Sales and Catering system or an RFP site, with the receiving system being a PMS or another Sales and Catering system.

The business model assumes that the originating system either has the authority to take a reservation, or is passing along a message from such a system. The message is a notification of the creation, modification, or cancellation of a meeting, and does not require the receiving system to confirm the booking, only the receipt of the message. The responding system may add its own data (such as its own confirmation ID) and include that data in the response message

The syntax of the root elements of the payload document used to exchange a create meeting profile request are enumerated as follows:

**<OTA\_CreateMeetingProfileRQ>** - Requests the receiving system to accept creation of a meeting. All pertinent details about the meeting are included.

**<OTA\_CreateMeetingProfileRS>**<sup>21</sup> - Returns acknowledgement of the request and can include additional information (reservationID). The response message may include Warnings from business processing rules, or Errors if the request did not succeed. This generic OTA class has been extended by adding the MeetingProfile object to allow synchronous data exchange .

```
<xsd:include schemaLocation = "OTA_v2ent.xsd"/>
<xsd:include schemaLocation = "MeetingProfile.xsd"/>
```

An additional request/response pair of messages is supported that provides the functionality to update a previously received **<OTA\_CreateMeetingProfileRQ>** request. This message pair supports sending only the information that needs to be added or changed.

**<OTA\_UpdateMeetingProfileRQ>**<sup>22</sup> - Requests the receiving system to accept a modification to a previously sent create meeting profile. Only the information that is being added or changed needs to be sent. This message utilizes the XPATH syntax to indicate where or what data should be added/changed.

**<OTA\_UpdateMeetingProfileRS>**<sup>23</sup> - Returns acknowledgement of the request and can include additional information (reservationID). The response message may include Warnings from business processing rules, or Errors if the request did not succeed.

The root tag of the **<OTA\_CreateMeetingProfileRQ>** and **<OTA\_UpdateMeetingProfileRQ>** contains the standard payload attributes on the root tag of all OTA messages.

- *OTA\_PayloadStdAttributes*

<sup>20</sup> The OpenTravel Alliance wishes to thank the collective efforts of Daylight Software for their contributions.

<sup>21</sup> The OTA\_CreateRS is a generic message defined in the OTA 2001A Specification.

<sup>22</sup> The OTA\_UpdateRQ is a generic message defined in the OTA 2001A Specification.

<sup>23</sup> The OTA\_UpdateRQ is a generic message defined in the OTA 2001A Specification.

**UniqueID**- Unique Identifier for this reservation. UniqueID is defined in the OTA Version 2001A.

Following the standard payload attributes entity is the core class, called the “MeetingProfile”. Please refer to the diagram below for an overview of the different elements and attributes of the “MeetingProfile” class.

What follows is an itemized explanation of the attributes and elements of the “MeetingProfile” object.

**Note:**

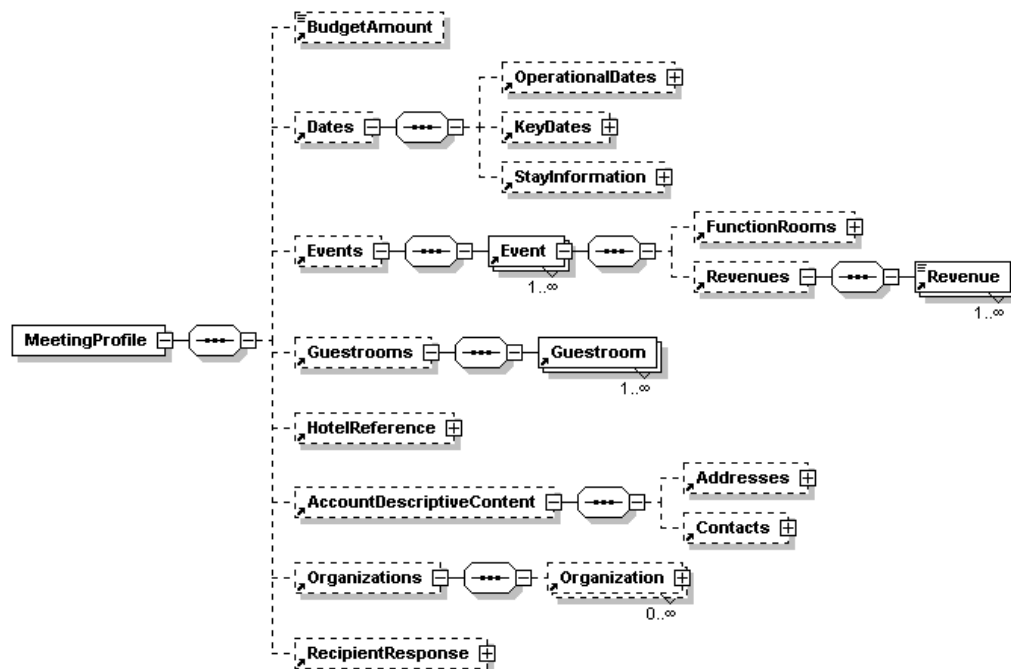
All the elements are indicated in **Bold**.

All the attributes are indicated in *italics*.

Example:

**Element** – Element Description

- *Attribute* – Attribute Description



## 5.65 MeetingProfile

Container for the Meeting Profile. Meeting Profiles messages can describe the following business objects: RFP,Lead,Booking,FutureBooking, and History.

- *MeetingID* - Vendor specific unique Identifier
- *MeetingProfileTypeCode* - Meeting Profile can be: RFP,Lead,Booking,FutureBooking, and History
- *MeetingAbbrevText* - Booking Code
- *MeetingControlID* - Booking Control ID (18 char unique ID)
- *ReservationID* - This may be an ID assigned by a PMS system.
- *PostAsText* - Name or the posting name of the meeting.
- *ReservationTypeCode* - This code describes the type of reservation used when creating this meeting (code values are TBD<sup>24</sup>).
- *MarketSegmentCode* - The code of the market segment that the meeting belongs to (code values are TBD).
- *StatusCode* - Code describing the status of the booking.
- *ContractTypeCode* - Code describing the type of contract employed in creating the meeting.
- *StatusReasonText* - A free form text field to describe the disposition of the status for the meeting.
- *GuestQuantity* - Total amount of guests attending the meeting.
- *SpecialRequestText* - Free form text to communicate any special requests for the meeting.
- *CommentText* - Free form text to communicate additional information about the meeting.

**BudgetAmount** - The amount that the meeting sponsors are willing to spend on the meeting.

- *CurrencyCode* – The ISO standard code representing currency amounts.

This element houses all the dates associated with this meeting. The date classes contained within are Operational dates, Key Dates, and the actual meeting stay information.

**OperationalDates** - Collection of OperationalDate.

**OperationalDate** – An operational date is defined as an action (Booked By, Created By, etc.) taken by an individual. The contact information defines that person and the date component defines when that action happened.

- *Date* - A date and/or time in the following format "2000-05-21T00:00:00"
- *OperationalDateCode* - This code describes the nature of the operational date (code values are TBD).

**Contact** - See Contact class under Account

**KeyDates** - Collection of KeyDate

**KeyDate** – A date and/or time in the following format "2000-05-21T00:00:00"

- *KeyDateCode* - Code defining the usage of the KeyDate (DecisionDue | CutOff | DateBooked | HoldSpaceUntil | OfferIsValidUntil)

<sup>24</sup> All MeetingProfile code fields are enumerated types. Many of these types have not yet been defined at the time of publication.

**StayInformation** - This element describes the beginning, ending, and duration of a stay in units of time.

**ArrivalInfo** – This is the preferred start and end date of the meeting

- *ArrivalDate* – Preferred specific arrival date.
- *DepartureDate* - Preferred specific departure date.

**AlternateDates** – This is a collection of alternate arrival dates.

**AlternateDate** – This represents optional starting and ending dates.

**AltDate** – Altdate specifies a particular start and end date. Multiples can be specified along with an indication of the desirability of the alternative

- *Date* - Specific alternative date when the preferred arrival date is not available.
- *TotalStayDaysQuantity* - The total amount of days for entire stay.
- *PreferenceID* - An alphanumeric field used to identify ranking of an alternative arrival date. (First Option, Second Option, etc...)

**AltRange** – If a particular date alternative is too specific then a range of desired alternative dates can be supplied

- *DOWCode* - Day of the week code works in conjunction with alternative start/end dates offering arrivals that extend for a specific period beginning on the day of the week represented by this code; (Mon | Tue | Wed | Thu | Fri | Sat | Sun).
- *AltStartDate* - Alternative start date used when the preferred dates are deemed unacceptable.
- *TotalStayDaysQuantity* - The total amount of days for entire stay.
- *AltEndDate* - Alternative end date used when the preferred dates are deemed unacceptable.
- *PreferenceID* - An alphanumeric field used to identify ranking of an alternative arrival date. (First Option, Second Option, etc...)

**Events** – a collection of catering events

**Event** – catering events that are part of this meeting (breakfast, breakout, dinner, etc.)

- *EventID* – a unique identifier for this event.
- *PostAsText* - Name or the posting name of the event.
- *EventNameText* - The name of the event. This name may appear different than the posting name.
- *EventStatusCode* - The status code of the event normally defined by the acceptance of a proposal or contract. (The code values are TBD).
- *StartDateTime* - Start time of the event using the format specified by ISO 8601:YYY-MM-DDThh:mm:ssZ with time values using the 24-hour (military) clock. For example: March-31-2001, 4:10:00 pm UTC becomes 2001-03-31T16:10:00Z.
- *EndDateTime* - End time of the event. Same time format as StartTime.
- *EventTypeCode* - The code that defines the type of event.



- *SetupTypeCode* - The code that defines the setup of the event.
- *SpecialSetupTypeCode* - The code that defines the special setup of the event. A special setup is normally defined as a setup that decreases a setup's capacity. (eg. Rear Projection)
- *AgreedAttendanceQuantity* - Attendance for an event as agreed upon with the client.
- *ActualAttendanceQuantity* - The actual attendance of an event as defined during/after an event is taking place.
- *ExpectedAttendanceQuantity* - Expected attendance is the attendance figure that is subjective and usually derived from the agreed figure and past experiences.
- *GuaranteedAttendanceQuantity* - Guarantee attendance is the attendance figure that is contractually agreed upon by the client and the property.
- *Moveable* - Can the event be moved to another function room? (Yes|No)
- *SetupTime* - Total amount of time that is required to Setup an event. This time is considered above and beyond the event start and end times.
- *TeardownTime* - Total amount of time that is required to teardown an event. This time is considered above and beyond the event start and end times.
- *SpecialRequestText* - This field allows for a free form textual description of any Special Requests regarding the setup of the event.
- *ExhibitBoothsQuantity* - Total amount of booths required.

**FunctionRooms** – collection of FunctionRoom

**FunctionRoom** - a Hotel Function room class that represents an area.

**Revenues** – collection of revenues

**Revenue** - A revenue amount, denominated in the defined currency.

- *RevCode* – A Revenue category code that describes the source of the revenue. Categories provide a way to classify hotel financial statistics to show all spending in a specific category (e.g., food and beverage, room, etc.)
- *RevTypeCode* - Type of revenue (Actual | Projected | Forecast).
- *CurrencyCode* – The ISO standard code representing currency amounts.

This collection of Guestroom classes holds all the guest room block details for the meeting. The attributes on the base collection class represent the aggregates across all days of the meeting.

- *TotalAgreedQuantity* - Total agreed quantity of guestrooms for the meeting.
- *TotalBlockQuantity* - Total blocked quantity of guestrooms for the meeting.
- *TotalPickUpQuantity* - Total picked up quantity of guestrooms for the meeting.
- *TotalForecastQuantity* - Total forecast quantity of guestrooms for the meeting.
- *TotalGuestQuantity* - Number of guests in the corresponding guestrooms for the meeting.
- *RoomsQuotedCurrencyCode* – The ISO standard code representing currency amounts.
- *AverageRateAmount* - Average rate for the guestrooms for the entire meeting defined in the Rooms Quoted Currency code.
- *PeakRoomsQuantity* - The highest number of rooms, by night, for this booking. If the pattern is 50/50/65/50 the peak rooms are 65.

**Guestroom** - this represents detailed information about particular guestroom inventory on a specific date.

- *StayDate* - Date specific to a guestroom that is blocked. This is optional if the sender specifies the *DateNumber* instead.
- *DateNumber* - This number represents the actual day within the stay. For example, if the number was 1, this would indicate the guestroom block for the first day of the stay. This is also optional if the actual date is specified in
- *RoomTypeCode* - Code defining the room type specified for this meeting.
- *RoomTypeCodeContext* - Text name of the room specified for this meeting.
- *OccupancyCode* - Occupancy code for the guestrooms in this meeting. (Single, Double, Triple, Quad, XChild, XAdult, Total)
- *AgreedQuantity* - The number of guestrooms agreed to by the client and the property. Commonly displayed on a contract.
- *BlockQuantity* - The number of guestrooms blocked by the property. This number is based on the agreed number and client history.
- *PickUpQuantity* - Number of guestrooms that were actually utilized or "picked up" by the client.
- *ForecastQuantity* - ForecastQuantity a field where a user can define a subjective forecast for the number of guestrooms.
- *RateAmount* - Currency amount assigned to each of the rooms by type. This number is expressed in a currency defined by the *RoomsQuotedCurrencyCode*.
- *GuestNumber* - Number of guests in the corresponding guestrooms, by date.

**HotelReference** - A Hotel Reference that contains the hotel chain, hotel brand or flag, and individual hotel code that identifies a lodging facility this meeting is targeted for.

**Contacts** - This is an optional collection of contacts and will contain people who are connected to the hotel (Sales Manager, Service Manager, etc.) and connected to this meeting profile.

**AccountDescriptiveContent** - The Account is the sponsor or "client" for whom the meeting is held.

- *OrgID* - a unique identifier used to represent the account.
- *OrgNameText* - Name of the Account.
- *URLText* - Name of the URL that uniquely identifies the Account.
- *DandBCode* - Dun and Bradstreet number.
- *DivisionCode* - The code of the division that the account belongs to (code values are TBD).
- *IATACode* - International Assoc. of Travel Agencies reference number.
- *IndustryCode* - The code used to define a type of industry as represented by the SIC and NAIC codes (code values are TBD).
- *GeoMarketCode* - The code of the geographic market that the account belongs to (code values are TBD).
- *LanguageCode* - The code of the language used by the account (code values are TBD).
- *MarketSegmentCode* - The code of the market segment that the account belongs to (code values are TBD).

## Addresses

**Address**

- *UsageCode* - A code that identifies what kind of Address this is. (eg. Home, Office, ...)
- *Address1Text* - This first line of address information.
- *Address2Text* - This second line of address information.
- *CityText* - The city portion of the address.
- *StateText* - The state portion of the address.
- *PostalCodeText* - The zip code portion of the address.

**Contacts** – Collection of Contacts

**Contact** - People associated with the Hotel, Account, or related Organizations.

- *ContactID* – a unique identifier used to represent the account.
- *PrefixText* - eg. "Mrs, Mr, Dr, ..."
- *GivenNameText* - A person's first name.
- *MiddleNameText* - A person's middle name.
- *SurnameText* - A person's last name.
- *SuffixText* - eg. "Esq, ..."
- *TitleText* - A person's professional title.
- *UsageCode* - A code that describes the contextual relationship of this person.
- *UsageCodeContextText* - The context by which the Usage code is defined.

**Addresses****Phones****Emails****Organizations** – A collection of Organization

**Organization** - These are related organizations such as agencies, vendors, suppliers

- *OrgID* – a unique identifier used to represent the account.
- *OrgNameText* - The name of the Organization.
- *OrgCode* - A unique code that identifies an organization.
- *UsageCode* - A code that describes the contextual relationship of this organization.
- *URLText* - Name of the URL that uniquely identifies the Organization.
- *UsageCodeContextText* - The context by which the Usage code is defined.

**Addresses**

**Contacts** - People associated with the Hotel, Account, or related Organizations.

**RecipientResponse** – Response generated after receiving an RFP

- *StatusCode* - The status code of the recipient response defines the acceptance or rejection of the Lead (code values are TBD).
- *StatusReasonText* - The recipient responders reason for choosing a particular status code.
- *ResponseExplanationText* - Text field providing additional information about the status reason.

**Dates** – date class as defined in the meeting record, however these dates represent a change to the base meeting dates.

**Commission** - The Commission element contains the data for a commission amount to be paid. The amount represents the total amount.

- *CommCode* - Code determined between vendors that indicates the reason for this commission.
- *CurrencyCode* – Indicates the currency in which the commission amount is expressed
- *Percent* - Percent is used to calculate the commission amount. Percent is represented using the base datatype, decimal.

## 5.66 Sample XML Fragment to Create a Meeting Profile Request

```
<?xml version="1.0" encoding="UTF-8"?>
<MeetingProfile xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="MeetingProfile.xsd" MeetingID="D0916BD6DED411D393321564854"
MeetingProfileTypeCode="RFP" MeetingAbbrevText="VFW Annual Meeting"
MeetingControlID="D0916BD6DED411D393" ReservationID="1234567890" PostAsText="stringVFW Annual Meeting"
ReservationTypeCode="abc" MarketSegmentCode="CORP" StatusCode="TENT" ContractTypeCode="N/A"
StatusReasonText="N/A" GuestQuantity="50" SpecialRequestText="Handicap accessible" CommentText="">
  <BudgetAmount CurrencyCode="USD">100000</BudgetAmount>
  <Dates>
    <OperationalDates>
      <OperationalDate Date="2000-05-21T00:00:00" OperationalDateCode="Created">
        <Contact ContactID="123456879" UsageCode="12345" UsageCodeContextText="ContactInfo"/>
      </OperationalDate>
    </OperationalDates>
    <KeyDates>
      <KeyDate KeyDateCode="CutOff">2000-05-21T00:00:00</KeyDate>
      <KeyDate KeyDateCode="DecisionDue">2000-05-21T00:00:00</KeyDate>
    </KeyDates>
    <StayInformation>
      <ArrivalInfo ArrivalDate="2000-04-01T00:00:00" DepartureDate="2000-04-03T00:00:00"/>
      <AlternateDates>
        <AlternateDate>
          <AltDate Date="2000-04-03T00:00:00" TotalStayDaysQuantity="2" PreferenceID="1"/>
        </AlternateDate>
      </AlternateDates>
    </StayInformation>
  </Dates>
  <Events>
    <Event PostAsText="Dinner" EventNameText="Dinner" EventStatusCode="" StartDateTime="13:20:00-05:00"
EndDateTime="2001-09-11T09:30:47-05:00" EventTypeCode="Dinner" SetupTypeCode="Rounds8"
SpecialSetupTypeCode="" AgreedAttendanceQuantity="50" ActualAttendanceQuantity="50"
ExpectedAttendanceQuantity="50" GuaranteedAttendanceQuantity="50" Moveable="true" SetupTime="13:20:00-05:00"
TeardownTime="13:20:00-05:00" SpecialRequestText="" ExhibitBoothsQuantity="" EventID="9878797987987">
      <FunctionRooms>
        <FunctionRoom><!--HITIS Standard Class--></FunctionRoom>
      </FunctionRooms>
      <Revenues>
        <Revenue RevCode="Food" RevTypeCode="Actual" CurrencyCode="USD">5000</Revenue>
        <Revenue RevCode="Beverage" RevTypeCode="Actual" CurrencyCode="USD">5000</Revenue>
      </Revenues>
    </Event>
  </Events>
  <Guestrooms TotalAgreedQuantity="" TotalBlockQuantity="100" TotalPickUpQuantity="" TotalForecastQuantity=""
TotalGuestQuantity="" AverageRateAmount="" PeakRoomsQuantity="40" RoomsQuotedCurrencyCode="">
    <Guestroom DateNumber="1" RoomTypeCode="Run of House" RoomTypeCodeContext="DSC"
OccupancyCode="Single" AgreedQuantity="" BlockQuantity="10" PickUpQuantity="" ForecastQuantity="" RateAmount=""
GuestNumber=""/>
    <Guestroom DateNumber="1" RoomTypeCode="Run of House" RoomTypeCodeContext="DSC"
OccupancyCode="Double" AgreedQuantity="" BlockQuantity="10" PickUpQuantity="" ForecastQuantity=""
RateAmount="" GuestNumber=""/>
    <Guestroom DateNumber="1" RoomTypeCode="Run of House" RoomTypeCodeContext="DSC"
OccupancyCode="Triple" AgreedQuantity="" BlockQuantity="10" PickUpQuantity="" ForecastQuantity="" RateAmount=""
GuestNumber=""/>
    <Guestroom DateNumber="1" RoomTypeCode="Run of House" RoomTypeCodeContext="DSC"
OccupancyCode="Quad" AgreedQuantity="" BlockQuantity="10" PickUpQuantity="" ForecastQuantity="" RateAmount=""
GuestNumber=""/>
  </Guestrooms>
  <HotelReference ChainCode="" BrandCode="" HotelCode="jkh123jkh123">
    <Contacts>
      <Contact PrefixText="" GivenNameText="Anne" MiddleNameText="" SurnameText="McDermott" SuffixText=""
TitleText="" UsageCode="SalesMgr" UsageCodeContextText="" ContactID="123456879"/>
    </Contacts>
  </HotelReference>
  <AccountDescriptiveContent OrgNameText="Daylight Software" URLText="DaylightSoftware.com" DandBCode=""
DivisionCode="" IATACode="" IndustryCode="" GeoMarketCode="NE" LanguageCode="ENG"
MarketSegmentCode="CORP" OrgID="54654223218795220">
```

```

<Addresses>
  <Address UsageCode="Main" Address1Text="100 Main Street" Address2Text="Suite 200" CityText="Dover"
StateText="NH" PostalCodeText="03824"/>
</Addresses>
<Contacts>
  <Contact PrefixText="" GivenNameText="Marcus" MiddleNameText="" SurnameText="Keyser" SuffixText=""
TitleText="" UsageCode="" UsageCodeContextText="" ContactID="123456879"/>
</Contacts>
</AccountDescriptiveContent>
<Organizations>
  <Organization OrgNameText="Samantha's Travel" OrgCode="Samantha's Travel" UsageCode="Agency"
URLText="Samantha.com" UsageCodeContextText="DSC">
    <Addresses>
      <Address UsageCode="" Address1Text="" Address2Text="" CityText="" StateText="" PostalCodeText=""/>
    </Addresses>
  </Organization>
</Organizations>
<RecipientResponse StatusCode="Rejected" StatusReasonText="Rate to low" ResponseExplanationText="We expect
a rate over 200">
  <Commission CommCode="" CurrencyCode="" Percent=""/>
</RecipientResponse>
</MeetingProfile>

```

### 5.67 Revision History for Meeting Profile

<u>Date</u>	<u>Version</u>	<u>Revisions</u>
May 6, 2001	0.1	Initial Draft.
May 10, 2001	0.2	revised (Marcus Keyser – Daylight Software)
October 29, 2001	0.3	revised (Marcus Keyser – Daylight Software)
October 30, 2001	0.4	revised (Carl Leander – Daylight Software)
October 30, 2001	0.5	revised (Marcus Keyser – Daylight Software)
October 31, 2001	0.6	Updated.

# **OpenTravel Alliance Message Specification**

## **Publication 2001B**

### **Section 5a - HITIS Availability, Rate and Inventory**

## **Hotel Availability, Rate and Inventory Messages**

The HITIS standards were originally defined as object-oriented specifications, subsequently converted to a messaging format using the eXtensible Mark-up Language (XML) to accommodate a Wide Area Network environment, and to provide interoperability between disparate platforms and message-based legacy systems. As an object-oriented specification, the Availability, Rate & Inventory standard included atomic messages that could be combined into one object, or transmission, to communicate information about the status of a hotel property to reservations systems.

The XML message set consists of individual messages that push notification data and anticipate a response message that confirms successful processing of the information, or an indication of error(s), to complete the message conversation between systems.

In addition, the Availability, Rate & Inventory standard provided a means by which a hotel could publicize descriptive information, Hotel Descriptive Content, in a standardized format.

### **Scope of ARI Messages**

The Availability, Rate and Inventory messages are used by hotel systems to synchronize information about their rates and to communicate the availability of that hotel's inventory with another system. The request message is a push interface that broadcasts the status of the hotel to update its information on a periodic basis. It allows a hotel Property Management System (PMS) to update a Central Reservation System (CRS) with rate amounts, supplementary charges; such as extra bed charges or additional adults or children, and availability affected by rate hurdles, stay restrictions, etc., within a date range for currently negotiated rate plans. Additionally, a hotel may use these messages to notify reservation systems of the creation of new rates, inventory blocks created for groups, and new inventory items, such as a new room type, in accordance with their business agreements for distribution.

It is the responsibility of the sending system to keep track of all the notifications sent and to ensure that each of its individual subscribers has received the notifications. Since the sending system may be pushing out an enormous quantity of transactions, it is presumed that the originating system would resend notifications as necessary if a system were unable to be contacted.

In the hotel environment, systems may choose to send one or several selected notification messages together in one message to communicate the status of their availability, rates and inventory at a given time, with each individual notification message providing a specific category of information about the hotel property.

Viewership given to the Notification messages includes the authorized distribution channel for the inventory, and the profile of the authorized booker for the inventory. Authorized distribution channels are determined by a collection of destination codes in the StatusApplicationControl that allows for controls to be turned on or off. Viewership is generally established when a new rate plan code is negotiated.

The scope of these specifications is limited to the creation, update and/or deletion of information sent via Notifications from the hotel to the designated systems in its distribution channel.



## Business Use of ARI Messages

When a new hotel opens for the first time, the Property Management System would use the Availability, Rate, and Inventory messages to notify reservations systems about the number of rooms in the hotel, the amounts charged for those rooms, and the rate plans available for that hotel, etc. In addition, the complete descriptive information that would be used to advertise and sell that hotel's property and services is broadcast to the negotiated distribution list.

When a hotel has been in operation for a period of time, and the inventory of rooms, services and amenities offered at the hotel well-established, the type of messages sent would more frequently include updates of previously broadcast information. Status messages are sent frequently (often daily) to reservation sources to notify them of the availability of the hotel for booking purposes.

## Availability Rate and Inventory Notification Messages

Nine separate Notification messages are defined in these Availability, Rate, and Inventory specifications. Multiple notifications may sent within one MIME multipart wrapper as a "push" transmission to update the availability, rates and inventory of a hotel to a booking entity or reservation system. The hotel being statused is identified within each individual message.

The response messages returned after processing the notification messages indicate success in processing the notification, or specific warnings or error conditions. All of the response messages use an identical construction, with the exception of the root tag element that identifies the context of the response message. The corresponding request/response pairs for the notification messages are the following:

### REQUEST MESSAGES

- OTA\_HotelAvailNotifRQ
- OTA\_HotelBookingRuleNotifRQ
- OTA\_HotelRateAmountNotifRQ
- OTA\_HotelSummaryNotifRQ
- OTA\_HotelInvCountNotifRQ
- OTA\_HotelInvNotifRQ
- OTA\_HotelInvBlockNotifRQ
- OTA\_HotelRatePlanNotifRQ
- OTA\_HotelDescriptiveContentNotifRQ

### RESPONSE MESSAGES

- OTA\_HotelAvailNotifRS
- OTA\_HotelBookingRuleNotifRS
- OTA\_HotelRateAmountNotifRS
- OTA\_HotelSummaryNotifRS
- OTA\_HotelInvCountNotifRS
- OTA\_HotelInvNotifRS
- OTA\_HotelInvBlockNotifRS
- OTA\_HotelRatePlanNotifRS
- OTA\_HotelDescriptiveContentNotifRS

OTA specifications provide for sending error and warning messages as part of versioned messages. Typically, if a Notification message fails for a reason at the business content level, the response message <OTA\_HotelxxxRS> indicates the reason for the failure.

The response has meaning only in the context of the Notification message that was sent. The failure of any single message may not necessarily result in the failure of all messages, however, in order to have the complete status information about a hotel, the receiving systems would need to update all notification messages successfully.

## Notification Response Messages

Every response message **MUST** have an optional `<Success/>` element. The presence of the empty `<Success/>` element explicitly indicates that the Notification message succeeded. In the absence of `<Success/>`, an implementation may return `<Warnings>` in the event of one or more business context errors, OR `<Errors>` in the event of a failure to process the message altogether.

If a response message returns an error condition that has occurred in the processing of the Notification Request, attributes in the `%ErrorAttr` entity indicate the type or error and other information about the error or warning returned. Systems may independently maintain an external table for reference to the text description of the error associated with the *Type* and/or *Code* attribute(s) unless the *DocURL* attribute is used to provide a location of the description about the error / warning.

## Standard Module for Versioned Message Responses

All Notification response messages contain the set of Payload Standard Attributes on the root element. The `OTA_PayloadStdAttributes` schema is part of the `OTA_v2ent.xsd`. The `OTA_v2ent.xsd` is included in all OTA payload messages.

Error and Warning elements share a common definition (with the exception of the tag name). These common elements and parameter entities are provided for defining a message response. The attributes of a `<Warning>` or `<Error>` element are identical and defined in the Error Attributes of the `OTA_v2ent.xsd` file.

## Data Elements in Response Messages

With the exception of the syntax of the root element that mirrors the individual notification request message (such as `OTA_HotelBookingRuleNotifRS`, `OTA_HotelInvBlockNotifRS`, or `OTA_HotelDescriptiveContentNotifRS`, etc.), the following data elements and attributes are found in all OTA response messages, as illustrated by the `OTA_HotelAvailNotifRS`:

### ***OTA\_HotelAvailNotifRS* <Element>**

The `OTA_HotelAvailNotifRS` message returns an indication of the status of processing the `OTA_HotelAvailNotifRQ` message.

#### **Attributes:**

##### **EchoToken : CDATA**

The EchoToken is a unique identifying string assigned by the host system used for message identification. When a request message includes an EchoToken, the corresponding response message returns an EchoToken with an identical value.

##### **TimeStamp : CDATA (xsd:timeInstant)**

The Timestamp indicates the creation date and time of the message using the ISO format `y:YYYY-MM-DDThh:mm:ss` with time values using the 24-hour (military) clock. The timestamp attribute should be represented in UTC (Universal Coordinated Time). Example: `2001-05-29T14:45:30Z`.

##### **Target : (Test | Production)**

The Target attribute indicates if the message is a test or production message, with the default value of Production. Valid values: (Test | Production).

**SequenceNmbr : CDATA (xsd: integer)**

A sequence number assigned by the host system for a series of messages. The presence of a sequence number allows for the ordering of messages for processing, or to request a resynchronization of messages in the event that a system has been offline and needs to retrieve messages that were missed.

**Version : CDATA (xsd: integer)**

The version of the payload document indicated by a string value.

**Success**

An optional, empty element that explicitly indicates an OTA versioned message has succeeded. In the absence of Success, an implementation may return Warnings in the event of one or more business context errors, or Errors in the event of a failure to process the message altogether.

**Warnings**

In the absence of Success, an implementation may return an Errors element that indicates a failure to process the message altogether.

**Warning**

A Warning identifies the reason for a business context error.

**Attributes:**

**Type : (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel)**

A required attribute that indicates the error type from among a list of enumerated values. A validating DTD may accept values of the Type = "Unknown and process them in an acceptable way.

**Code : CDATA**

Refers to a table of coded values exchanged between applications to identify errors or warnings.

**DocURL : CDATA**

Supplies the URL of an online description of the error that occurred.

**Tag : CDATA**

This attribute is used to identify an unknown or misspelled tag that caused an error in processing. It is recommended that the Tag attribute use XPath notation to identify the location of a tag in the event that more than one tag of the same name is present in the document. Alternatively, the tag name alone can be used to identify missing data [Type=ReqFieldMissing]

**Status : CDATA**

Identifies the status of processing the message. Although the data type is designated as CDATA, recommended values are those enumerated in the non-versioned Standard Error message (NotProcessed | Incomplete | Complete | Unknown). Versioned message responses may identify additional status conditions between trading partners that are not included in the standard enumeration.

**RecordID : CDATA**

This attribute allows for the identification of the record that failed amongst a group of records during batch processing.

**Errors**

In the absence of Success, an implementation may return the Error element to indicate errors in processing.

**Error**

An Error that indicates the reason for failure to process a message.

**Attributes:**

**Type :** (Unknown | NoImplementation | BizRule | Authentication | AuthenticationTimeout | Authorization | ProtocolViolation | TransactionModel | AuthenticationModel)

A required attribute that indicates the error type from among a list of enumerated values. A validating DTD may accept values of the Type = "Unknown and process them in an acceptable way.

**Code :** CDATA

Refers to a table of coded values exchanged between applications to identify errors or warnings.

**DocURL :** CDATA

Supplies the URL of an online description of the error that occurred.

**Tag :** CDATA

This attribute is used to identify an unknown or misspelled tag that caused an error in processing. It is recommended that the Tag attribute use XPath notation to identify the location of a tag in the event that more than one tag of the same name is present in the document. Alternatively, the tag name alone can be used to identify missing data [Type=ReqFieldMissing]

**Status :** CDATA

Identifies the status of processing the message. Although the data type is designated as CDATA, recommended values are those enumerated in OTA non-versioned Standard Error messages (NotProcessed | Incomplete | Complete | Unknown). Versioned message responses may identify additional status conditions between trading partners that are not included in the standard enumeration.

**RecordId :** CDATA

This attribute allows for the identification of the record that failed amongst a group of records during batch processing.

## 5.68 Hotel Availability Notification

The Availability Notification message, <OTA\_HotelAvailNotifRQ>, notifies a booking source of the status of availability at a specific hotel property. Booking a reservation at a hotel is often affected by systems using yield management tables to determine the availability of a specific rate at a given time. Therefore, the Availability Notification message is often sent in conjunction with two other messages: a Rate Amount Notification message, that communicates the rates that apply to the availability, and a Booking Rule Notification message, that communicates the restrictions that apply to the availability and rates.

These messages include a complex set of controls that indicate whether the hotel has available inventory; that is, closed or open for booking. The RateHurdleStatusMessage element establishes an open/closed situation based upon the number of units available. If a hotel is open, status messages communicate the rate at which those bookings can be made. In addition, booking restrictions that apply to each individual rate, such as a minimum length of stay (LOS) must also be communicated to the booking agent so that the hotel guest is informed of all the regulations that govern their reservation.

Inventory is generally considered a physical count, and availability a commitment to sell a room at a specific rate or plan. The physical inventory is the basis by which counts are assigned to the availability. But availability may also depend upon rate plans, as a system may carry a discrete inventory, or an inventory count in association with different rates. Thus, the superset of the inventory may be greater than the physical count, with the actual number of rooms counted down when they are sold.

The status messages in the Availability Notification message also communicate inventory (booking) limits set by Yield and Revenue management systems such as the number of reservations that can be taken for a certain day, and the threshold at which the hotel is closed. A Booking Limit Status Message may even define what can be done after a status is set, such as "Take four more reservations after this status is set."

A system may choose not to synchronize with actual inventory numbers, but with a threshold. Nevertheless, it is critical that booking systems are synchronized with common thresholds, regardless of whether they are derived from virtual or real inventory.

The Availability Notification message uses the StatusApplicationControl to set the status for an inventory block, a rate plan or an inventory code. The attributes InventoryCodeType, RatePlanCodeType, and InventoryBlockCodeType determine whether the message involves a single code, or a grouping of codes.

The Override attribute allows a reservation system to make a change on controls applied at the level of the Property Management System. For example, a CRS may allowed to make manual changes while processing bookings during the day, but when full optimization is done, typically during the night, this Boolean attribute determines whether to retain the changes made. This could be applied to override all status messages and is found in the Status Application Control class.

## Availability Notification Messages

**OTA\_HotelAvailNotifRQ** – Sends an availability notification to another system. . The server must successfully update all messages. The failure of any single status message will result in the failure of all messages.

**OTA\_HotelAvailNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not able to be processed.

## Rate Hurdle Status Messages

The RateHurdleStatusMessage element is used to define the settings for rate hurdle controls at the reservations system. A basic implementation of a rate hurdle control accepts reservations with a net room rate (Reservation rate amount minus sum of internal costs from inclusive services) equal to or higher to the sum of the Rate Hurdle for each day and InventoryCode of the reservation.

A more complete implementation adds the usage of Delta values to add (up to an upper Ceiling) or subtract to the RateHurdle as transient reservations are made or cancelled in between RateHurdleNotification sessions and checks to see that the number of transient reservations made in between sessions does not exceed an upper MaxSold limit.

These additional steps require the reservation system to implement a Rate Hurdle Table for each date and InventoryCode under RMS control with an internal counter for transient reservations to be reset with each notification. The goal is to allow the reservation system to react to small changes between RateHurdleNotification sessions.

### 5.69 OTA\_HotelAvailNotifRQ Sample Message:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelAvailNotifRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelAvailNotifRQ.xsd" EchoToken="12345" TimeStamp="1999-05-
31T13:20:00-05:00" Target="Test" Version="1" SequenceNmbr="123456">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <StatusMessages>
    <StatusMessage>
      <BookingLimitStatusMessages>
        <BookingLimitStatusMessage BookingLimitMessageType="AdjustLimit" BookingLimit="12">
          <StatusApplicationControl InvBlockCodeType="BlockCode" RatePlanCode="wewe"
InvCodeType="InvCode" InvCode="wewe" InvBlockCode="wewe" Override="true">
            <DestinationSystemCodes>
              <DestinationSystemCode>375689</DestinationSystemCode>
              <DestinationSystemCode>394285</DestinationSystemCode>
              <DestinationSystemCode>417350</DestinationSystemCode>
            </DestinationSystemCodes>
            <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
            <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
          </StatusApplicationControl>
        </BookingLimitStatusMessage>
      </BookingLimitStatusMessages>
      <FullPatternStatusMessages>
        <FullPatternStatusMessage StatusCode="Open" ArrivalsOnlyFlag="true" Override="false">
```

```

        <StatusApplicationControl InvBlockCodeType="DoesNotApply"
RatePlanCodeType="RatePlanCode" RatePlanCode="RT2FK" InvCodeType="InvCode" InvCode="BWTV82K"
InvBlockCode="0" Override="false">
        <!--Needs to be a reused fragment. -->
        <DestinationSystemCodes>
            <DestinationSystemCode>375689</DestinationSystemCode>
            <DestinationSystemCode>394285</DestinationSystemCode>
            <DestinationSystemCode>417350</DestinationSystemCode>
        </DestinationSystemCodes>
        <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
        <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
        </StatusApplicationControl>
        <LengthsOfStay TimeUnitType="Day">
            <LOS>2</LOS>
            <LOS>3</LOS>
        </LengthsOfStay>
        </FullPatternStatusMessage>
    </FullPatternStatusMessages>
    <MinMaxLOSStatusMessages>
        <MinMaxLOSStatusMessage MinMaxMessageType="RemoveMinLOS" ArrivalsOnlyFlag="true">
            <StatusApplicationControl InvBlockCodeType="DoesNotApply"
RatePlanCodeType="RatePlanCode" RatePlanCode="RT2FK" InvCodeType="InvCode" InvCode="BWTV82K"
InvBlockCode="0" Override="false">
            <!--Needs to be a reused fragment. -->
            <DestinationSystemCodes>
                <DestinationSystemCode>375689</DestinationSystemCode>
                <DestinationSystemCode>394285</DestinationSystemCode>
                <DestinationSystemCode>417350</DestinationSystemCode>
            </DestinationSystemCodes>
            <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
            <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
            </StatusApplicationControl>
            <LengthsOfStay TimeUnitType="Day">
                <LOS>2</LOS>
                <LOS>3</LOS>
            </LengthsOfStay>
            </MinMaxLOSStatusMessage>
        </MinMaxLOSStatusMessages>
        <RateHurdleStatusMessages>
            <RateHurdleStatusMessage Ceiling="3" MaxSold="6">
                <StatusApplicationControl InvBlockCodeType="DoesNotApply"
RatePlanCodeType="RatePlanCode" RatePlanCode="RT2FK" InvCodeType="InvCode" InvCode="BWTV82K"
InvBlockCode="0" Override="false">
                <!--DestinationSystemCodes is a reused schema fragment. -->
                <DestinationSystemCodes>
                    <DestinationSystemCode>375689</DestinationSystemCode>
                    <DestinationSystemCode>394285</DestinationSystemCode>
                    <DestinationSystemCode>417350</DestinationSystemCode>
                </DestinationSystemCodes>
                <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
                <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
                </StatusApplicationControl>
                <HurdleRate CurrencyCode="USD">180.00</HurdleRate>
                <Delta CurrencyCode="USD">39920.00</Delta>
            </RateHurdleStatusMessage>
        </RateHurdleStatusMessages>
        <SimplePatternStatusMessages>
            <SimplePatternStatusMessage StatusType="Open" ArrivalsOnlyFlag="false">
                <StatusApplicationControl InvBlockCodeType="DoesNotApply"
RatePlanCodeType="RatePlanCode" RatePlanCode="RT2FK" InvCodeType="InvCode" InvCode="BWTV82K"
InvBlockCode="0" Override="false">
                <DestinationSystemCodes>
                    <DestinationSystemCode>375689</DestinationSystemCode>
                    <DestinationSystemCode>394285</DestinationSystemCode>
                </DestinationSystemCodes>
            </SimplePatternStatusMessage>
        </SimplePatternStatusMessages>
    </RateHurdleStatusMessages>
    </FullPatternStatusMessages>

```

```

                                <DestinationSystemCode>417350</DestinationSystemCode>
                                </DestinationSystemCodes>
                                <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
                                <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
                                </StatusApplicationControl>
                                </SimplePatternStatusMessage>
                                </SimplePatternStatusMessages>
                                </StatusMessage>
                                </StatusMessages>
</OTA_HotelAvailNotifRQ>

```

## 5.70 Hotel Booking Rule Notification

The Hotel Booking Rule Notification message, `<OTA_HotelBookingRuleNotifRQ>`, communicates the rules and restrictions associated with the general availability or rates at a hotel to a booking source. The application of a booking rule may narrow the availability of inventory at a specific hotel property. For example, a hotel may be accepting reservations for a two-night or three-night stay, but will not accept a reservation for a one-night stay. This situation may be driven by the use of a yield management system that affects the availability of a specific rate at a given time. The Booking Rule Notification message is often sent in conjunction with two other messages: the Availability Notification, `<OTA_HotelAvailNotifRQ>`, that communicates the status of availability at a specific hotel property, and the Rate Amount Notification message, `<OTA_HotelRateAmountNotifRQ>`, that communicates the rates that the booking restrictions must be applied to.

The Booking Rule Notification message uses the `StatusApplicationControl` to indicate the inventory block, rate plan or inventory code that the booking rules apply to. Each `BookingRule` is potentially a set of different types of booking restrictions. The attributes `InventoryCodeType`, `RatePlanCodeType`, and `InventoryBlockCodeType` determine whether the message involves a single code, or a grouping of codes. In addition, the booking restrictions that apply to each individual rate may include such factors as a minimum length of stay (LOS), or specific days of the week that they are applicable (`DOWPattern`).

These messages may be used to define multiple rules and restrictions applied to a rate plan. For example, it can set absolute dates during which a restriction is to be applied. Alternatively, a Booking Rule can define the minimum offset of time as well as the maximum offset of time required prior to a guest's arrival prior when a restriction is to be applied, or during which a booking can be made. The minimum and maximum advance requirements are not mutually exclusive, and can be used in combination. The Absolute Deadline and/or the Advance Booking attributes may be used to set applicable restrictions to booking dates.

The Booking Rule Notification message can be used to communicate the types of guarantees that are accepted for a booking, to indicate whether a reservation can be modified or cancelled, and if a refund of a deposit is allowed in the case of a cancellation. The `GuaranteeType` is an enumerated type that indicates whether a guarantee is required, or if it is required, the form of the guarantee, such as a credit card, debit card or voucher. In some cases, an actual deposit is required. In other cases, supplying a `Profile`, that provides the identification of a frequent guest by membership or loyalty program number, may be sufficient for a guarantee.

The `CancelPenalties` element defines a collection of restrictions and policies for payments made to a hotel in case of cancellation. It is also used to specify the cancellation fee or penalties imposed by the booking restrictions that would be applied when a reservation is NOT canceled,



as in the case of a no-show. Cancellation penalties may be applied within a specified time frame either prior to arrival, or after the booking has been made. Likewise, the Required Payments <RequiredPaymts> element is used to specify a payment obligation, such as a deposit due, along with the deadline for the payment. The RetributionType indicates the action taken when the deadline has been exceeded, such as cancellation of a reservation when a required payment is not made.

## 5.71 Hotel Booking Rule Notification Messages

**OTA\_HotelBookingRuleNotifRQ** – Sends a booking rule notification to another system. .  
The server must successfully update all messages sent in a single transmission. The failure of any single status message will result in the failure of all messages.

**OTA\_HotelBookingRuleNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not able to be processed.

## 5.72 OTA\_HotelBookingRuleNotifRQ Sample Message:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelBookingRuleNotifRQ xmlns="http://www.opentravel.org/OTA"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_HotelBookingRuleNotifRQ.xsd" EchoToken="12345" TimeStamp="1999-05-
31T13:20:00-05:00" Target="Production" Version="1" SequenceNmbr="123456">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <RuleMessages>
    <RuleMessage>
      <StatusApplicationControl InvBlockCodeType="DoesNotApply"
RatePlanCodeType="RatePlanCode" RatePlanCode="RT2FK" InvCodeType="InvCode"
InvCode="BWTV82K" InvBlockCode="0" Override="false">
        <!--Needs to be a reused fragment. -->
        <DestinationSystemCodes>
          <DestinationSystemCode>375689</DestinationSystemCode>
          <DestinationSystemCode>417350</DestinationSystemCode>
        </DestinationSystemCodes>
        <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
        <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
      </StatusApplicationControl>
      <BookingRules>
        <BookingRule MaxAdvancedBookingOffset="P1Y2M15DT12H30M"
MinAdvancedBookingOffset="P1Y2M15DT12H30M" MinTotalOccupancy="1" MaxTotalOccupancy="4"
AbsoluteDropTime="1999-06-30T24:20:00">
          <DropTimeOffset DropTimeOffset="P1Y2M15DT12H30M"
DropTimeOffsetType="BeforeArrival"/>
          <BookingRuleCodes>
            <BookingRuleCode>IsCancellable</BookingRuleCode>
            <BookingRuleCode>IsModifiable</BookingRuleCode>
          </BookingRuleCodes>
          <EffectiveBookingDateRange>
            <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
            </EffectiveBookingDateRange>
            <GuaranteesAccepted>
              <GuaranteeAccepted
GuaranteePolicyType="NoGuaranteesAccepted"/>
              <GuaranteeAccepted
GuaranteePolicyType="GuaranteesRequired">
                <AcceptedPayments>
                  <AcceptedPayment
PaymentType="CreditCard" CurrencyCode="USD">67.50</AcceptedPayment>
                </AcceptedPayments>
              </GuaranteeAccepted>
            </GuaranteesAccepted>
          <AcceptedCards>
            <AcceptedCard>MC</AcceptedCard>
            <AcceptedCard>VISA</AcceptedCard>
          </AcceptedCards>
        </BookingRule>
      </BookingRules>
    </RuleMessage>
  </RuleMessages>
</OTA_HotelBookingRuleNotifRQ>
```

```

        </AcceptedCards>
        <ProfileTypes>
            <ProfileType ProfileType="Wholesaler">
            </ProfileType>
        </ProfileTypes>
        <Profiles>
            <Profile ProfileType="Customer"/>
            <!--Profile to be inserted as external entity-->
            <Profile ProfileType="Customer"/>
            <!--Profile to be inserted as external entity-->
        </Profiles>
        <CancelPenalties>
            <CancelPenalty AbsoluteDeadline="1999-06-03T13:00:00"
TaxInclusive="true" FeesInclusive="false">
                <DeadlineOffset
DeadlineOffset="P1Y2M15DT12H30M" DeadlineOffsetType="BeforeArrival"/>
                <Amount CurrencyCode="USD">67.50</Amount>
                <AmountPercent NmbrOfNights="2"
BasisType="FullStay" Percent="1.5"/>
            </CancelPenalty>
        </CancelPenalties>
        <RequiredPaymts>
            <RequiredPaymt RetributionType="ResNotGuaranteed"
AbsoluteDeadline="1999-06-03T13:00:00" TaxInclusive="true" FeesInclusive="false">
                <DeadlineOffset
DeadlineOffset="P1Y2M15DT12H30M" DeadlineOffsetType="BeforeArrival"/>
                <Amount CurrencyCode="USD">67.50</Amount>
                <AmountPercent NmbrOfNights="2"
BasisType="Nights" Percent="0.5"/>
            </RequiredPaymt>
        </RequiredPaymts>
        <Description>"A guarantee payment must be received 24 hours prior
to arrival in the amount of the first night's stay is required or your reservation will be cancelled. Cancellation less
that 24 hours before arrival, will result in forfeit of 50% of the guarantee amount. If you do not arrive at the hotel by
midnight of the first day of your stay, the full amount of your guarantee is not refundable."</Description>
        </BookingRule>
    </BookingRules>
</RuleMessage>
</RuleMessages>
</OTA_HotelBookingRuleNotifRQ>

```

### 5.73 Booking Rule Notification Response

The <OTA\_HotelBookingRuleNotifRS> returns an indication that the notification message was successfully processed, and may include Warnings from business processing rules or Errors if the notification was not able to be processed.

### 5.74 Hotel Inventory Count Notification

The Inventory Count Notification message, <OTA\_HotelInvCountNotifRQ>, notifies a booking source of the amount of inventory available at a specific hotel property. It allows the Property Management System and Central Reservation Systems or other booking sources to synchronize the number of inventory items available for sale between them.

When a new hotel is opened for the first time, the <OTA\_HotelInvNotifRQ> Inventory Notification message would be used to supply the reservations systems with descriptions of

rooms in the hotel, as well as non-room products that are subject to inventory as well. The Inventory Count Notification is used to send base inventory levels by inventory code, (e.g.: room type code) to establish the physical inventory count. An Inventory Notification should always precede an Inventory Count Notification to establish the existence of inventory codes in the receiving system.

The physical inventory is the basis by which availability is determined. However, additional calculations figure into assigning the inventory counts for availability. Availability is a commitment to sell a room at a specific rate or plan. Since the same rooms may be sold under different rate plans, as a system may carry a discrete inventory, or an inventory count in association with different rates. The superset of inventory may be greater than the physical count, with the actual number of rooms counted down when they are sold.

The Inventory Count Notification message can be used to communicate to revenue management systems how many rooms are available to sell during a specific period. A reservation system may choose not to synchronize with actual inventory numbers, rather with a threshold. Properties and booking sources need to agree on common thresholds, whether they are derived from virtual or real inventory, as well as a way to accommodate overbooking.

This Notification message allows for communicating both base and off-sell inventory. The base inventory message accommodates changes in the base inventory levels, such as adding a new wing of the hotel. The off-sell inventory message sends a count of the inventory that is not available for sale. The off-sell messages indicating whether that inventory is temporarily out of order or has been taken off the market, as well as whether the inventory count is an adjustment to a current off-sell value, or a replacement of a previously determined amount.

## 5.75 Hotel Inventory Count Notification Messages

**OTA\_HotelInvCountNotifRQ** – Sends a notification inventory count to another system to synchronize the amount of inventory available.

**OTA\_HotelInvCountNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not able to be processed.

## 5.76 Off Sell Inventory Messages

### *OffSellInvMessages*

The set of changes in inventory that is not available for sale to be made on the server. The server must successfully update all messages. Failure to update any single status message will result in the failure of all messages.

### *OffSellInvMessage*

Communicates information about inventory that is not available for sale.

## 5.77 OTA\_HotelInvCountNotifRQ Sample Message

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<OTA_HotelInvCountNotifRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelInvCountNotifRQ.xsd" EchoToken="12345" TimeStamp="1999-
05-31T13:20:00-05:00" Target="Production" Version="1" SequenceNmbr="123456">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <BaselInvMessages>
    <BaselInvMessage InvCode="Y2KD" Inv="46">
      <DateTimeSpan StartInstant="1999-06-03T13:00:00" Duration="P1Y2M15DT12H30M"/>
    </BaselInvMessage>
  </BaselInvMessages>
  <OffSellInvMessages>
    <OffSellInvMessage InvCode="Y2KD" OffSellReasonType="Renovation" OffSellValueType="Total"
OffSellValue="5">
      <DateTimeSpan StartInstant="1999-06-03T13:00:00" Duration="P1Y2M15DT12H30M"/>
    </OffSellInvMessage>
  </OffSellInvMessages>
</OTA_HotelInvCountNotifRQ>

```

## 5.78 Hotel Summary Notification

The Hotel Summary Notification message, `<OTA_HotelSummaryNotifRQ>`, notifies a booking source of the general availability status of the hotel; indicating whether it is Open, Closed, or OnRequest, which means that a hotel is available to take reservations but is limited by restrictions. This notification can be used to update the status of the hotel and may be coupled with other notifications, such as the Booking Rule, Availability, or Rate Amount notifications to convey the general availability, rates, and restrictions in effect at a given time.

The availability status of a hotel may be affected by Yield Management System calculations. On a historical basis, a certain period of time may support higher rates or greater occupancy and thus limit the general availability of the hotel. Rate hurdles establish an open/closed situation based upon the number of units available. If a hotel is open, the Hotel Summary Notification message communicates the minimum and maximum rate at which bookings can be made. As the rates and availability of a hotel property change, status messages are sent frequently (often daily) to reservation sources to notify them of the availability of the hotel for booking purposes.

During a particularly busy time, a hotel may be partially booked with only a few rate plans or room types remaining available. When a travel agent contacts that hotel to book a reservation for the guest, a message may be returned indicating that the hotel is "On Request". This means that the property has some availability and the requesting system needs to make another request using a Hotel Availability Request `<OTA_HotelAvailRQ>` to determine the specific availability. A return of "On Request" indicates that a hotel is not closed, but is sufficiently full that a booking request may fail depending upon what is requested.

## 5.79 Hotel Summary Notification Messages

**OTA\_HotelSummaryNotifRQ** – Sends notification of the macro level availability status of a hotel to another system.

**OTA\_HotelSummaryNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not able to be processed.

## 5.80 OTA\_HotelSummaryNotifRQ Sample Messages:

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<OTA_HotelSummaryNotifRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelSummaryNotifRQ.xsd" EchoToken="12345" TimeStamp="1999-
05-31T13:20:00-05:00" Target="Production" Version="1" SequenceNmbr="123456">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <HotelSummaryMessages>
    <HotelSummaryMessage StatusType="OnRequest">
      <DateTimeSpan StartInstant="1999-06-03T13:00:00" Duration="P1Y2M15DT12H30M"/>
      <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false" Sat="false"
Sun="false"/>
      <RateMin CurrencyCode="USD">86.00</RateMin>
      <RateMax CurrencyCode="USD">124.00</RateMax>
    </HotelSummaryMessage>
  </HotelSummaryMessages>
</OTA_HotelSummaryNotifRQ>

```

## 5.81 Hotel Rate Amount Notification

The Hotel RateAmount Notification message, <OTA\_HotelRateAmountNotifRQ>, notifies a booking source of changes in the rates charged for room and non-room products of a hotel.

The creation of a new rate plan is done through the Rate Plan Notification message. When the rate amount of an active (bookable) rate plan changes, an update is made through the Rate Amount Notification. The Status Application Control is used to identify the inventory item (or inventory block), and the rate plan that the change in rate amount applies to.

The Hotel Rate Amount Message defines the amount of the base rate, as well as the maximum number of adults permitted in a room at the rate, along with the charges for additional adults and children. Tax amounts that apply to the rate are also communicated, indicating the type of tax, and how it is calculated, whether a flat amount or percentage. In short, the Rate Amount Notification should convey all of the information needed by a reservation system to book a hotel room (or non-room product) at the newly-established rate amount.

Using the Status Application Control, rate changes can be made based on dates, days of week, rate plan codes and/or inventory and inventory block codes. The following are examples of different types of rate amount changes that could be applied through this message:

- *Dates* - the rate changes from \$89.00 per night to \$99.00 per night from May 21st through July 31st for double bed rooms and king bed rooms (inventory code).
- *Days of Week* - The rate for all rooms on this property change from \$69.00 per night to \$59.00 per night on Fridays and Saturdays.
- *RatePlan Codes* - AAA and AARP rates are increased from \$79.00 to \$89.00 per night.
- *Inventory Codes* (Room product) - Suites and apartment room rates are increased by 10% (using inventory codes that define these inventory types).
- *Inventory Code* (Non-room product) - Rates for ballrooms and meeting rooms are increased from May 1<sup>st</sup> through July 1<sup>st</sup>.
- *Inventory Block Code* - The room rate for a convention group (identified by inventory block code) is \$95.00 per night.
- *Additional occupancy* - Rates are \$ 9.00 per night for additional adults. Rates for additional children are \$5.00 per night.

When a rate amount is changed, the new rate amount must be populated up through the distribution system. The Viewership element defines the authorized distribution channel for the inventory, and the profile of the authorized booker for the inventory. Viewership is generally set

up when a new rate plan code is negotiated. The authorized distribution channels are determined by the collection of destination codes in the Status Application Control.

### 5.82 Hotel Rate Amount Notification Messages

**OTA\_HotelRateAmountNotifRQ** – Sends a notification of a rate amount change to other systems.

**OTA\_HotelRateAmountNotifRS** - Returns an acknowledgement that the notification was successfully processed, or Warnings from business rules or Errors if the notification was not able to be processed.

### 5.83 OTA\_HotelRateAmountNotifRQ Sample Message

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelRateAmountNotifRQ xmlns="http://www.opentravel.org/OTA"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OTA_HotelRateAmountNotifRQ.xsd" EchoToken="12345" TimeStamp="1999-05-
31T13:20:00-05:00" Target="Production" Version="1" SequenceNmbr="123456">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <RateAmountMessages>
    <RateAmountMessage>
      <StatusApplicationControl InvBlockCodeType="DoesNotApply"
RatePlanCodeType="RatePlanCode" RatePlanCode="RT2FK" InvCodeType="InvCode" InvCode="BWTV82K"
InvBlockCode="0" Override="false">
        <DestinationSystemCodes>
          <DestinationSystemCode>375689</DestinationSystemCode>
          <DestinationSystemCode>394285</DestinationSystemCode>
        </DestinationSystemCodes>
        <DateTimeSpan StartInstant="1999-06-03T13:00:00"
Duration="P1Y2M15DT12H30M"/>
        <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true" Fri="false"
Sat="false" Sun="false"/>
      </StatusApplicationControl>
      <RateAmounts>
        <RateAmount RateAmountUnitType="Day" NmbrOfUnits="1" MaxAdults="2"
AgeQualifyingCode="A2C">
          <Amount CurrencyCode="USD">86.00</Amount>
          <DOWPattern Mon="true" Tues="true" Wed="true" Thurs="true"
Fri="false" Sat="false" Sun="false"/>
          <AdditionalAdultAmounts>
            <AdditionalAdultAmount MaxAdults="2">
              <Amount CurrencyCode="USD">12.00</Amount>
            </AdditionalAdultAmount>
          </AdditionalAdultAmounts>
          <AdditionalChildAmounts>
            <AdditionalChildAmount AgeQualifyingCode="U12Y"
MaxChild="4">
              <AmountWithAdult
CurrencyCode="USD">00.00</AmountWithAdult>
              <AmountWithOutAdult
CurrencyCode="USD">8.00</AmountWithOutAdult>
            </AdditionalChildAmount>
          </AdditionalChildAmounts>
          <TaxAmounts>
            <TaxAmount TaxType="Inclusive" TaxCode="OC-12H"
TaxPercent="0.08">
              <Amount CurrencyCode="USD">0.0</Amount>
            </TaxAmount>
          </TaxAmounts>
          <FeeAmounts>
            <FeeAmount FeeType="Inclusive" FeeCode="MN-016"
FeePercent="0.32" Description="MunicipalityFee">
              <Amount CurrencyCode="USD">15.00</Amount>
            </FeeAmount>
          </FeeAmounts>
        </RateAmount>
      </RateAmounts>
    </RateAmountMessage>
  </RateAmountMessages>
</OTA_HotelRateAmountNotifRQ>
```

```

                                </FeeAmount>
                            </FeeAmounts>
                        </RateAmount>
                    </RateAmounts>
                </RateAmountMessage>
            </RateAmountMessages>
        </OTA_HotelRateAmountNotifRQ>

```

## 5.84 Hotel Inventory Notification

The Hotel Inventory Notification message, <OTA\_HotelInvNotifRQ> is the message that sends the notification of the creation of a new inventory item, such as a room type or service type that did not previously exist at a hotel property. When the data base of a reservation system or booking source is populated for the first time, the hotel inventory notification message would be used to send descriptions of the inventory in the hotel, both room and non-room products.

A Hotel Inventory Notification establishes the existence of inventory codes in the receiving system. As the exchange of inventory information is not always a simple process, as the sending system and the receiving system may assign different codes to the same inventory item, requiring the use of a translation table to identify the inventory item in one system with the same item in another system.

For that reason, the Hotel Inventory Notification message should precede the Inventory Count Notification, <OTA\_HotelInvCountNotifRQ>, and Rate Plan Notification messages. The Inventory Count Notification establishes the physical inventory count by inventory code, and a Rate Plan Notification <OTA\_HotelRatePlanNotifRQ>, assigns a rate plan to the inventory item.

While the Hotel Inventory Notification message provides the building block that populates or initializes the hotel for any reservation system to establish the number of rooms, etc., that can be sold, inventory restrictions that are associated with a rate can be set on the rate itself. Restrictions associated with a rate are sent using the Hotel Booking Rule Notification. Individual notification messages may be sent as separate transmissions or combined together within a MIME multipart envelope as each notification contains a Hotel Reference that identifies the hotel property.

When a hotel has been in operation for a period of time, the rooms, services and amenities that are part of inventory may change or be discontinued. The Inventory Notification allows for the update of an active inventory item, or the deletion of an inventory item altogether, indicating the current status of the inventory.

The response message returned for a new inventory item differs from other Availability, Rate and Inventory notification messages in that the receiving system may return the inventory code(s) assigned by that system that cross-reference with the codes received along with the confirmation that the message was processed successfully.

## 5.85 Hotel Inventory Notification Messages

**OTA\_HotelInvNotifRQ** – Sends a notification of the creation of an inventory item to another system.



**OTA\_HotelInvNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not able to be processed.

### 5.86 OTA\_HotelInvNotifRQ Sample Message:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelInvNotifRQ EchoToken="12345" TimeStamp="1999-05-31T13:20:00-05:00" Target="Production"
Version="1" SequenceNmbr="123456" InvNotifType="New" InvStatusType="Initial" InvCodeType="InvCode"
InvCode="Y2DK" InvGroupingCode="4698DK" InvName="Double Occupancy King Deluxe" IsRoom="true"
BeddingCode="King" Smoking="false">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <SellableProducts>
    <SellableProduct SellableProductRPH="1" InvCodeType="InvCode" InvCode="Y2DK">
      <ProductDescriptions>
        <ProductDescription ProductDescriptionRPH="1" LanguageCode="EN_US">
          <ShortDescription>One King bed with dual lighting control, 2 dressing tables, double
closets, sitting area, separate bath and boudoir with 2 sinks</ShortDescription>
          <LongDescription>Spacious room furnished with one size King bed with dual lighting
control,radio and alarm clocks, 2 dressing tables, double closets, ironing board, coffe maker in room, hair dryer,
dual connection telephone connections allow simultaneous use of modem and telephone line. L-shaped room has
sitting area furnished with coffee table and convenient mini-bar, refrigerator. Separate bath and boudoir with 2
sinks; luxurious 'his and her' terry cloth bath robes.</LongDescription>
        </ProductDescription>
      </ProductDescriptions>
    </SellableProduct>
  </SellableProducts>
</OTA_HotelInvNotifRQ>
```

### 5.87 Hotel Rate Plan Notification

The Hotel Rate Plan Notification message, <OTA\_HotelRatePlanNotifRQ>, is used to notify a booking source of a new rate plan created for a hotel, or to modify and synchronize existing rate plans between systems.

When a hotel creates a new rate, whether that hotel is new or has been in operation for some period of time, the synchronization of rate plans can be a complicated process. A translation table may be required to identify the rate plan in one system with the same rate plan that is stored in another system. The Hotel Rate Plan Notification message is sent to a booking agent, indicating whether this is 1) the initial announcement of a new rate plan, 2) an update of an active (bookable) rate plan, or 3) a notification that a rate plan is no longer in effect and should be deactivated in their system.

With the creation of a new rate plan, a business process must also take place to ensure that the rate plan is populated up through the distribution system. New rate plans and group blocks are broadcast through authorized channels of distribution determined by negotiated business agreements.

Viewership is usually set up when a new rate plan code is negotiated and it defines the distribution channel for the rate plan, and the profile of the authorized booker(s). The distribution channels are indicated by a collection of System Codes.

When a hotel system sends out a status message to notify systems of the availability of a hotel, the StatusApplicationControl uses the rate plan codes that have been established by the Hotel

Rate Plan Notification to determine which rate plans are available. The RatePlanCodeType indicates whether the rate plan(s) available are a single rate plan, or a grouping of rate plans.

The RatePlan Shoulders, Sellable Products and Viewerships contain a Reference Place Holder (RPH) element that can be used for indexing to identify a specific rate plan among a group of items of the same name.

## 5.88 Hotel Rate Plan Notification Messages

**OTA\_HotelRatePlanNotifRQ** - The Hotel Rate Plan Notification message is used to provide notification of a new rate plan created for a hotel, or to modify (synchronize) rate plans between systems

**OTA\_HotelRatePlanNotifRS** - Returns an acknowledgement that the notification was successfully received, Warnings from business processing rules, or Errors if the notification was unable to be processed.

### 5.89 OTA\_HotelRatePlanNotifRQ Sample Message:

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_HotelRatePlanNotifRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance" xsi:noNamespaceSchemaLocation="OTA_HotelRatePlanNotifRQ.xsd" EchoToken="12345" TimeStamp="1999-
05-31T13:20:00-05:00" Target="Production" Version="1" SequenceNmbr="123456" NmbrRequiredNights="2"
RatePlanNotifType="Delta" RatePlanStatusType="Active" GenerallyBookable="true" PriceViewable="true"
RatePlanName="Senior 55" ValidBookingMinOffset="P1Y2M15DT12H30M"
ValidBookingMaxOffset="P1Y2M15DT12H30M">
  <HotelReference HotelCode="1234STL" ChainCode="HI234" BrandCode="1235XTL"/>
  <RatePlanCode RatePlanCodeType="RatePlanCode">98TD4S</RatePlanCode>
  <RatePlanGroupingCode RatePlanCodeType="RatePlanGroupingCode">77846TD</RatePlanGroupingCode>
  <AvailableDaysOfWeek>
    <DOWPattern Mon="true" Tues="false" Wed="false" Thurs="false" Fri="true" Sat="true" Sun="true"/>
  </AvailableDaysOfWeek>
  <ArrivalDaysOfWeek>
    <DOWPattern Mon="false" Tues="false" Wed="false" Thurs="false" Fri="true" Sat="true"
Sun="false"/>
  </ArrivalDaysOfWeek>
  <RequiredDaysOfWeek>
    <DOWPattern Mon="false" Tues="false" Wed="false" Thurs="false" Fri="true" Sat="true"
Sun="false"/>
  </RequiredDaysOfWeek>
  <StayDateRange>
    <DateTimeSpan StartInstant="1999-06-01T09:00:00" Duration="P1Y2M15DT12H30M"/>
  </StayDateRange>
  <InvBlockCodes>
    <InvBlockCode InvBlockCodeType="BlockGroupingCode">Y2TD</InvBlockCode>
    <InvBlockCode InvBlockCodeType="BlockGroupingCode">Y2TS</InvBlockCode>
  </InvBlockCodes>
  <RatePlanShoulders>
    <RatePlanShoulder ShoulderRPH="1" PreShoulderSellLimit="6" PostShoulderSellLimit="6"
PreShoulderStartDate="1999-05-31T13:00:00" PostShoulderEndDate="1999-06-06T13:00:00">
      <SellableProducts>
        <SellableProduct SellableProductRPH="1" InvCodeType="InvCode"
InvCode="Y2TD">
          <ProductDescriptions>
            <ProductDescription ProductDescriptionRPH="1"
LanguageCode="EN_US">
              <ShortDescription>Two twin beds rentable to
double occupancy, 2 dressing tables, single closet, 30" color TV, separate bath and boudoir with 2
sinks</ShortDescription>
              <LongDescription>Moderate, newly-renovated
room furnished with two twin beds and individual night tables and alarm clocks, 2 dressing tables, single closet, ironing
board, coffe maker in room, hair dryer, dual connection telephone connections allow simultaneous use of modem and
```

```

telephone line. Convenient mini-bar, refrigerator. Separate bath and boudoir with 2 sinks; 30" color TV with dual remote
controls.</LongDescription>
                                </ProductDescription>
                                </ProductDescriptions>
                                </SellableProduct>
                                <SellableProduct SellableProductRPH="2" InvCodeType="InvCode"
InvCode="Y2TD">
                                <ProductDescriptions>
                                <ProductDescription ProductDescriptionRPH="1"
LanguageCode="EN_US">
                                <ShortDescription>Two twin beds rentable to
single occupancy, 2 dressing tables, single closet, 30" color TV, separate bath and boudoir with 2
sinks</ShortDescription>
                                <LongDescription>Moderate, newly-renovated
room furnished with two twin beds, night tables, alarm clock, dressing tables, single closet, ironing board, coffe maker in
room, hair dryer, dual connection telephone connections allow simultaneous use of modem and telephone line.
Convenient mini-bar, refrigerator. Separate bath and boudoir with 2 sinks; 30" color TV with dual remote
controls.</LongDescription>
                                </ProductDescription>
                                </ProductDescriptions>
                                </SellableProduct>
                                </SellableProducts>
                                </RatePlanShoulder>
                                </RatePlanShoulders>
                                <SellableProducts>
                                <SellableProduct SellableProductRPH="1" InvCodeType="InvCode" InvCode="Y2TD">
                                <ProductDescriptions>
                                <ProductDescription ProductDescriptionRPH="1" LanguageCode="EN_US">
                                <ShortDescription>Two twin beds rentable to double occupancy, 2
dressing tables, single closet, 30" color TV, separate bath and boudoir with 2 sinks</ShortDescription>
                                <LongDescription>Moderate, newly-renovated room furnished with two
twin beds and individual night tables and alarm clocks, 2 dressing tables, single closet, ironing board, coffe maker in
room, hair dryer, dual connection telephone connections allow simultaneous use of modem and telephone line.
Convenient mini-bar, refrigerator. Separate bath and boudoir with 2 sinks; 30" color TV with dual remote
controls.</LongDescription>
                                </ProductDescription>
                                </ProductDescriptions>
                                </SellableProduct>
                                <SellableProduct SellableProductRPH="2" InvCodeType="InvCode" InvCode="Y2TD">
                                <ProductDescriptions>
                                <ProductDescription ProductDescriptionRPH="1" LanguageCode="EN_US">
                                <ShortDescription>Two twin beds rentable to single occupancy, 2
dressing tables, single closet, 30" color TV, separate bath and boudoir with 2 sinks</ShortDescription>
                                <LongDescription>Moderate, newly-renovated room furnished with two
twin beds, night tables, alarm clock, dressing tables, single closet, ironing board, coffe maker in room, hair dryer, dual
connection telephone connections allow simultaneous use of modem and telephone line. Convenient mini-bar,
refrigerator. Separate bath and boudoir with 2 sinks; 30" color TV with dual remote controls.</LongDescription>
                                </ProductDescription>
                                </ProductDescriptions>
                                </SellableProduct>
                                <SellableProduct SellableProductRPH="2" InvCodeType="InvCode" InvCode="Y2DD">
                                <ProductDescriptions>
                                <ProductDescription ProductDescriptionRPH="2" LanguageCode="EN_US">
                                <ShortDescription>Double-occupancy room for Senior citizens age 55
or older. Moderate room with two twin beds rentable to double occupancy, 2 dressing tables, single closet, single bath and
boudoir with 2 sinks</ShortDescription>
                                <LongDescription>Moderate, newly-renovated room furnished with two
twin beds, night tables, alarm clock, dressing tables, single closet, ironing board, coffe maker in room, hair dryer, dual
connection telephone connections allow simultaneous use of modem and telephone line. Convenient mini-bar,
refrigerator. Separate bath and boudoir with 2 sinks; 30" color TV with dual remote controls.</LongDescription>
                                </ProductDescription>
                                </ProductDescriptions>
                                </SellableProduct>
                                </SellableProducts>
                                <Viewerships>
                                <Viewership ViewershipRPH="1" ViewOnly="false">
                                <ViewershipCodes>
                                <ViewershipCode>SystemCodesInclusive</ViewershipCode>
                                <ViewershipCode>ProfilesInclusive</ViewershipCode>
                                <ViewershipCode>ProfileTypesInclusive</ViewershipCode>

```

```

        <ViewershipCode>LocationCodesInclusive</ViewershipCode>
        <ViewershipCode>ViewOnly</ViewershipCode>
    </ViewershipCodes>
    <SystemCodes SystemCodesInclusive="true">
        <SystemCode>123289</SystemCode>
        <SystemCode>267458</SystemCode>
        <SystemCode>387648</SystemCode>
        <SystemCode>379289</SystemCode>
        <SystemCode>440827</SystemCode>
        <SystemCode>650827</SystemCode>
    </SystemCodes>
    <ProfileTypes>
        <ProfileType ProfileType="Corporation"></ProfileType>
        <ProfileType ProfileType="CRO"></ProfileType>
        <ProfileType ProfileType="Airline"></ProfileType>
    </ProfileTypes>
    <LocationCodes LocationCodesInclusive="false">
        <LocationCode>EA_US</LocationCode>
        <LocationCode>TN</LocationCode>
        <LocationCode>AR</LocationCode>
        <LocationCode>MW_US</LocationCode>
        <LocationCode>LA</LocationCode>
        <LocationCode>SO_US</LocationCode>
    </LocationCodes>
    <BookingChannelCodes ChannelCodesInclusive="true">
        <BookingChannelCode>XYZ.com</BookingChannelCode>
        <BookingChannelCode>ABC.com</BookingChannelCode>
    </BookingChannelCodes>
</Viewership>
</Viewerships>
</OTA_HotelRatePlanNotifRQ>

```

## 5.90 Hotel Rate Plan Notification Response

The <OTA\_HotelRatePlanNotifRS> returns a response to the Hotel Rate Plan Notification request message, indicating that the notification message was successfully processed, Warnings from business processing rules or Errors if the notification was not able to be processed.

Additionally, the response message may return the RatePlanCode(s) and /or Rate Plan Grouping Codes assigned to the rate plan by the receiving system in response to a new rate plan notification. These values would only be returned when the notification is of *RatePlanCodeType*=New and the sender is translating rate plan codes. If this is the case, the values sent in the RatePlanCode or RatePlanGroupingCode attributes could be empty, and in subsequent transactions for the inventory item, the sender would be able to populate the rate plan code with the value returned by the receiver.

## 5.91 Hotel Inventory Block Notification

The Inventory Block Notification message, <OTA\_HotelInvBlockNotifRQ>, is used to notify a booking authority of the creation of a group block that can be sold against inventory, and to subsequently modify or synchronize an existing inventory block between systems.

In order to accommodate reservations for a group of guests in one party, a hotel may assign an inventory block and notify the Central Reservation Systems of the code and the allotment that can be used. Travel agents that are authorized to book against the allotment may then contact the hotel or Central Reservations Office to pick up a reservation within the block of rooms.

Viewership of the inventory block is also a negotiated item. Some blocks may be created with agents having only a read-only capability because reservations for the block must be made

through a single convention bureau, or market segment. In this case, certain rates are packaged together and typically booked by a group of agents. Viewership defines the distribution channels for the block by using the profiles of the authorized booking agents, and assigning distribution channels through the collection of System Codes.

The Hotel Rate Plan Notification and the Hotel Inventory Block Notification messages can be combined to create a group block specifying inventory types, and rate plans, indicating the date range that the group block can be booked, including shoulder periods on either side of the stay dates. The Hotel Rate Amount Notification can be used to indicate the amount charged for the group plan, and any booking restrictions can be sent via the Hotel Booking Rule Notification if needed.

Thus, the Hotel Inventory Block Notification creates the foundation for communicating the rate and inventory of a block, as well as the rules associated with creation of the block. This message includes rates, room types, and hard rules that apply to the booking block, e.g.: 3-night stay required, etc. Although the Hotel Inventory Block Notification is a message that establishes the foundation for a block of inventory, it does not assume any booking activity.

Once the selling process is underway, the synchronization of inventory blocks can be a complicated process. A translation table may be needed to identify an inventory block in one system with the same inventory block that is stored in another system. The Hotel Inventory Block Notification message tells a booking agent whether this is an initial announcement of a new Inventory Block, an update of an active (bookable) block, or a notification of a block that is no longer in effect and should be deactivated in their system. The Booking Limit Status Message, a part of the Hotel Availability Notification, can be used to set new limits and report the utilization of the block in order to pass information, such as the Guest Count, and to synchronize the information on both sides of the interface.

When a hotel system sends out a status message to notify systems of the availability of a hotel, the Status Application Control uses the Inventory Block codes to determine the status of availability for the block. The InventoryBlockCodeType indicates whether the inventory block(s) available are a single Inventory Block code, or a grouping of Inventory Block codes.

The RatePlan Shoulders, Sellable Products and Viewerships contain a Reference Place Holder element (RPH) that can be used for indexing to identify a specific Inventory Block in a collection.

## 5.92 Hotel Inventory Block Notification Messages

**OTA\_HotelInvBlockNotifRQ** - The Hotel Inventory Block Notification message is used to provide notification of a new inventory block created for a hotel, or to modify (synchronize) inventory blocks between systems

**OTA\_HotelInvBlockNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not processed.

## 5.95 Hotel Inventory Block Notification Response

The <OTA\_HotelInvBlockNotifRS> returns a response to the Hotel Inventory Block Notification, indicating that the message was successfully processed, Warnings from business

processing rules, or Errors if the notification was not able to be processed.

Additionally, the response message may return the *InvBlockCode* and /or the *InvBlockGrouping Code(s)* assigned by the receiving system for the inventory block in response to a new inventory block notification. These values would only be returned when the notification is of *InvBlockCodeType*= New and the sender is translating the inventory block code values. In this case, the *InvBlockCode* attribute would be empty and in subsequent transactions for the Inventory Block, the sender would populate the *InvBlockCode* attribute with the values returned by the receiver.

#### *OTA\_HotelInvBlockNotifRS*

The *OTA\_HotelInvNotifRS* message returns an indication of the status of processing the *OTA\_HotelInvNotifRQ* message.

### **5.93 Hotel Descriptive Content Notification**

The Hotel Descriptive Content Notification is a broadcast message used to publicize detailed descriptive information about a hotel property by standardized data categories. Likewise, static information about a hotel property can be obtained by using the Hotel Search Request and/or Hotel Availability Request to search for static information by category, using codes agreed upon between trading partners to request more detail about a hotel.

The Hotel Descriptive Content interface enables accessing hotel data in both a push and pull format in order to avoid storing the data at multiple locations. In most cases, the hotel property is the owner of the data and is in charge of updating it, and sends out a broadcast message as a full overlay replacing previous information or a partial update message modification to make changes or portions of the data, using the *<OTA\_HotelDescriptiveContentNotifRQ>*.

When a new hotel opens for business, the complete descriptive information used to advertise and sell the hotel's property and services is broadcast in a standardized format to the negotiated distribution list. In this initial broadcast of property information, the sending system will be pushing out an enormous quantity of information. The PMS and remote systems must be able to buffer messages during any downtime. It is presumed that the system would continue to republish subsequent updates as necessary if a subscriber is unable to be contacted.

In the hotel environment, when a guest wishes to book a hotel, two basic search criteria often include the location of the hotel, and the price of the rooms. Beyond this, many factors can influence the guest's ultimate choice when booking a reservation. To assist the guest in making their choice, a booking agent looks further for descriptive information about a hotel, such as describing recreational or business services, or the hotel facilities or amenities. In many cases, the description of hotel static information may be more valuable than a percentage or weighting number given by the querying system in response to the hotel search. The Hotel Descriptive Content Specification defines the categories and fields that will allow the agent to search by code to answer the myriad of specific needs of the guest.

The descriptive content data is structured by categories of text data, and enables a query using a category code -either published by the OpenTravel Alliance or as agreed upon between trading partners. The transaction for pulling hotel data in granular sections using the Hotel Search Request *<OTA\_HotelSearchRQ>* is the Search Criterion *Type*="CodeRef". When performing an availability query using the message, *<OTA\_HotelAvailRQ>*, the element *<SearchCodes>* can include multiple *<CodeRef>* elements to obtain detailed information.

The data returned is determined by the category code sent in the request. A detailed query response may return a collection of descriptive content for each category.

### **5.94 Hotel Descriptive Content Notification Messages**

**OTA\_HotelDescriptiveContentNotifRQ** - The Hotel Descriptive Content Notification message is used to send hotel descriptive content information to a server from the PMS in a standardized format.

**OTA\_HotelDescriptiveContentNotifRS** - Returns an acknowledgement that the notification was successfully received, or Warnings from business processing rules or Errors if the notification was not processed.

### **5.95 Hotel Descriptive Content Notification Response**

The <OTA\_HotelDescriptiveContentNotifRS> returns a response to the request message, indicating that the notification message was successfully processed, or Warnings from business processing rules or Errors if the notification was not processed.