



OpenTravel Alliance

# Message Users Guide

2014A, Version 1, June 2014

The OpenTravel Alliance (OpenTravel) offers this publication for use at the discretion of the individual or company. Each individual or company that uses this material does so at its own risk and with the acknowledgment that the individual or company is solely and fully responsible for its use of this material.

**Intellectual Property Statement:** OpenTravel is a travel industry community effort to develop voluntary XML-based communications specifications and thereby facilitate the use of electronic commerce in our industry. These specifications are developed through a consensus building process that encourages the active participation of all Member Companies' representatives. Suggestions made at OpenTravel meetings are therefore likely to be included in OpenTravel specifications. Member companies and potential member companies should be aware of this and closely review the membership agreement. OpenTravel is the owner of OpenTravel specifications, and each member company grants the OpenTravel Alliance the right to sublicense to other member companies, their affiliates, and end users, a non-exclusive, royalty-free license as defined in the membership agreement.

## Contents

<b>Introduction and Getting Started with the OpenTravel Specification.....</b>	<b>21</b>
Introducing the OpenTravel Alliance .....	21
OpenTravel Alliance Resources .....	21
OpenTravel Specification Components .....	22
<b>Basic OpenTravel Schema Concepts.....</b>	<b>23</b>
OpenTravel Schema Design Best Practices .....	23
OpenTravel Message Exchange Patterns.....	23
Request/Response Pattern.....	23
Notif Pattern .....	23
Generic Messages .....	24
Payload Transaction Management.....	24
OpenTravel File Naming Conventions .....	26
OpenTravel Code Lists and Enumerations .....	26
External Code Lists .....	27
External Codes Lists & Other References in OpenTravel Schema .....	27
OpenTravel Message Success, Warnings & Errors Elements .....	28
Schema Architecture .....	29
Understanding OpenTravel Schema Architecture.....	29
OpenTravel Message Level Schema.....	30
OpenTravel Industry Common Types Schema .....	30
OpenTravel Common Types Schema.....	30
OpenTravel Simple Types Schema.....	30
<b>OpenTravel Message Transport.....</b>	<b>31</b>
Introduction.....	31
Definitions & Conventions .....	31
State Maintenance.....	31
SOAP Messaging .....	32
Purpose.....	32
The SOAP Transport Protocol .....	32
The Need for Interoperability .....	33
Guidelines for OpenTravel Documents & Messages .....	35
SOAP Version 1.1 and 1.2 .....	35
SOAP Messaging and SOAP RPC .....	36
Differences between SOAP Messaging and SOAP RPC .....	36
WSDL Document/ Literal Binding .....	36

SOAP Messaging Benefits .....	37
SOAP Messaging vs. RPC Guidelines .....	37
SOAP Action URI .....	37
SOAP Messaging and the SOAP Action URI.....	37
SOAP Intermediaries .....	37
SOAP Action URI Guidelines .....	38
SOAP Envelope Content .....	38
SOAP Header Content.....	38
SOAP Body Content .....	39
SOAP Attachments .....	39
SOAP Fault vs. OpenTravel Error.....	39
Examples .....	40
HTTP Messaging.....	50
Background.....	50
The Need for Interoperability .....	50
<b>GENERIC MESSAGES.....</b>	<b>56</b>
Messages Overview .....	56
Messages List .....	56
OTA_AuthorizationRQ/RS .....	57
Message Pair Overview .....	57
OTA_CancelRQ/RS.....	57
Message Pair Overview .....	57
OTA_DeleteRQ/RS.....	58
Message Pair Overview .....	58
OTA_ErrorRS .....	58
Message Pair Overview .....	58
OTA_FileAttachmentNotifRQ/RS .....	59
Message Pair Overview .....	59
OTA_NotifReportRQ/RS.....	59
Message Pair Overview .....	59
OTA_PingRQ/RS.....	60
Message Pair Overview .....	60
OTA_PurchaseItemRQ/RS.....	60
Message Pair Overview .....	60
OTA_ReadRQ/OTA_ResRetrieveRS .....	63
Message Pair Overview .....	63
OTA_ReviewsNotifRQ/RS.....	64
Message Pair Overview .....	64
OTA_ReviewsRQ/RS .....	64
Message Pair Overview .....	64

OTA_ScreenTextRQ/RS .....	64
Message Pair Overview .....	64
<b>AIR MESSAGES .....</b>	<b>65</b>
Messages Overview .....	65
Messages List .....	65
OTA_AirAvailRQ/RS .....	67
Message Pair Overview .....	67
OTA_AirBaggageRQ/RS .....	67
Message Pair Overview .....	67
OTA_AirBookRQ/RS .....	68
Message Pair Overview .....	68
OTA_AirCheckInRQ/RS .....	68
Message Pair Overview .....	69
OTA_AirDemandTicketRQ/RS .....	69
Message Pair Overview .....	69
OTA_AirDisplayQueueRS .....	69
Message Pair Overview .....	69
OTA_AirFareDisplayRQ/RS .....	69
Message Pair Overview .....	69
OTA_AirFifoRQ/RS .....	69
Message Pair Overview .....	69
OTA_AirDetailsRQ/RS .....	70
Message Pair Overview .....	70
OTA_AirLowFareSearchRQ/RS .....	70
Message Pair Overview .....	70
OTA_AirPriceRQ/RS .....	71
Message Pair Overview .....	71
OTA_AirRulesRQ/RS .....	71
Message Pair Overview .....	71
OTA_AirScheduleRQ/RS .....	71
Message Pair Overview .....	71
OTA_AirSeatMapRQ/RS .....	72
Message Pair Overview .....	72
OTA_AirBookModifyRQ/OTA_AirBookRS .....	72
Message Pair Overview .....	72
OTA_AirGetOfferRQ/RS .....	72
Message Pair Overview .....	72
<b>CAR MESSAGES .....</b>	<b>74</b>
Messages Overview .....	74
Messages List .....	75

OTA_VehAvailRateRQ/RS .....	76
Message Pair Overview .....	76
OTA_VehCancelRQ/RS .....	76
OTA_VehCheckOutRQ/RS.....	77
Message Pair Overview .....	77
OTA_VehCheckInRQ/RS .....	77
Message Pair Overview .....	77
OTA_VehLocDetailsNotifRQ/RS .....	78
Message Pair Overview .....	78
OTA_VehExchangeRQ/RS.....	78
Message Pair Overview .....	78
OTA_VehLocDetailRQ/RS.....	78
Message Pair Overview .....	78
OTA_VehLocSearchRQ/RS .....	79
Message Pair Overview .....	79
OTA_VehModifyRQ/RS .....	79
Message Pair Overview .....	79
OTA_VehRateNotifRQ/RS.....	80
Message Pair Overview .....	80
OTA_VehRateRuleNotifRS/RS.....	80
Message Pair Overview .....	80
OTA_VehRateRuleRQ/RS.....	80
Message Pair Overview .....	80
OTA_VehResRQ/RS .....	81
Message Pair Overview .....	81
OTA_VehResStatusNotifRQ/RS.....	81
Message Pair Overview .....	81
OTA_VehRetResRQ/RS.....	81
Message Pair Overview .....	81
OTA_VehResNotifRQ/RS .....	82
Message Pair Overview .....	82
<b>CRUISE MESSAGES .....</b>	<b>83</b>
Messages Overview .....	83
Messages List .....	84
OTA_CruiseBookingDocumentRQ/RS .....	85
Message Pair Overview .....	85
OTA_ReadRQ/OTA_CruiseBookingHistoryRS .....	85
Message Pair Overview .....	85
OTA_CruiseCabinAvailRQ/RS .....	85
Message Pair Overview .....	85

OTA_CruiseCabinHoldRQ/RS.....	86
Message Pair Overview .....	86
OTA_CruiseCabinUnholdRQ/RS.....	86
Message Pair Overview .....	86
OTA_CruiseCancellationPricingRQ/RS .....	86
Message Pair Overview .....	86
OTA_CruiseCategoryAvailRQ/RS .....	86
Message Pair Overview .....	86
OTA_CruiseBookRQ/RS .....	87
Message Pair Overview .....	87
OTA_CruiseDiningAvailRQ/RS .....	87
Message Pair Overview .....	87
OTA_CruiseFareAvailRQ/RS .....	87
Message Pair Overview .....	87
OTA_CruiseFastSellRQ .....	88
Message Pair Overview .....	88
OTA_CruiseInfoRQ/RS .....	88
Message Pair Overview .....	88
OTA_CruiseItineraryDescRQ/RS .....	88
Message Pair Overview .....	88
OTA_CruisePaymentRQ/RS .....	89
Message Pair Overview .....	89
OTA_CruisePkgAvailRQ/RS .....	89
Message Pair Overview .....	89
OTA_CruisePriceBookingRQ/RS .....	89
Message Pair Overview .....	89
OTA_CruisePNR_UpdateNotifRQ/RS.....	89
Message Pair Overview .....	89
OTA_CruiseSailAvailRQ/RS.....	90
Message Pair Overview .....	90
OTA_CruiseShorexAvailRQ/RS .....	91
Message Pair Overview .....	91
OTA_CruiseSpecialServiceAvailRQ/RS.....	91
Message Pair Overview .....	91
<b>DESTINATION ACTIVITY MESSAGES .....</b>	<b>92</b>
Messages Overview .....	92
Messages List .....	92
OTA_DestActivityCapabilitiesRQ/RS .....	93
Message Pair Overview .....	93
OTA_DestActivityResRQ/RS.....	93

Message Pair Overview .....	93
<b>DYNAMIC PACKAGE MESSAGES .....</b>	<b>94</b>
Messages Overview .....	94
Messages List .....	94
OTA_DynamicPkgAvailRQ/RS .....	94
Message Pair Overview .....	94
OTA_DynamicPkgBookRQ/RS .....	95
Message Pair Overview .....	95
OTA_ReadRQ/OTA_DynamicPkgBookRS.....	95
Message Pair Overview .....	95
OTA_DynamicPkgModifyRQ/ OTA_DynamicPkgBookRS .....	95
Message Pair Overview .....	95
OTA_CancelRQ/OTA_DynamicPkgBookRS .....	95
Message Pair Overview .....	95
OTA_DynamicPkgModifyNotifRQ/RS.....	95
Message Pair Overview .....	95
<b>GOLF MESSAGES .....</b>	<b>97</b>
Messages Overview .....	97
Messages List .....	97
OTA_GolfCourseAvailRQ/RS .....	98
Message Pair Overview .....	98
OTA_GolfFacilityInfoRQ/RS .....	98
Message Pair Overview .....	98
OTA_GolfRateRQ/RS.....	99
Message Pair Overview .....	99
OTA_GolfCourseResRQ/RS .....	99
Message Pair Overview .....	99
OTA_GolfCourseResModifyRQ/RS.....	100
Message Pair Overview .....	100
OTA_GolfCourseSearchRQ/RS .....	100
Message Pair Overview .....	100
<b>HOTEL MESSAGES .....</b>	<b>101</b>
Messages List .....	101
OTA_HotelAvailRQ/RS.....	103
Message Pair Overview .....	103
OTA_HotelAvailGetRQ/RS .....	104
Message Pair Overview .....	104
OTA_HotelAvailNotifRQ/RS .....	104
Message Pair Overview .....	104
OTA_HotelBookingRuleRQ/RS .....	105

Message Pair Overview .....	105
OTA_HotelBookingRuleNotifRQ/RS.....	105
Message Pair Overview .....	105
OTA_HotelCacheChangeRQ/RS .....	106
Message Pair Overview .....	106
OTA_HotelInvChangeRQ/RS .....	106
Message Pair Overview .....	106
OTA_HotelCacheChangeNotifRQ/RS .....	106
Message Pair Overview .....	106
OTA_HotelCommNotifRQ/RS .....	107
Message Pair Overview .....	107
OTA_HotelDescriptiveContentNotifRQ/RS .....	107
Message Pair Overview .....	107
OTA_HotelDescriptiveInfoRQ/RS.....	108
Message Pair Overview .....	108
OTA_HotelEventRQ/RS .....	109
Message Pair Overview .....	109
OTA_HotelGetMsgRQ/RS.....	109
Message Pair Overview .....	109
OTA_HotelInvBlockRQ/RS.....	110
Message Pair Overview .....	110
OTA_HotelInvBlockNotifRQ/RS .....	110
Message Pair Overview .....	110
OTA_HotelInvCountRQ/RS .....	111
Message Pair Overview .....	111
OTA_HotelInvCountNotifRQ/RS.....	111
Message Pair Overview .....	111
OTA_HotelInvNotifRQ/RS .....	112
Message Pair Overview .....	112
OTA_HotelPostEventRQ/RS .....	112
Message Pair Overview .....	112
OTA_HotelPostEventNotifRQ/RS.....	113
Message Pair Overview .....	113
OTA_HotelRateAmountNotifRQ/RS .....	113
Message Pair Overview .....	113
OTA_HotelRatePlanRQ/RS.....	114
Message Pair Overview .....	114
OTA_HotelRatePlanNotifRQ/RS .....	114
Message Pair Overview .....	114
OTA_HotelResModifyRQ/RS .....	115

Message Pair Overview .....	115
OTA_HotelResModifyNotifRQ/RS .....	115
Message Pair Overview .....	115
OTA_HotelResNotifRQ/RS.....	115
Message Pair Overview .....	115
OTA_HotelResRQ/RS .....	116
Message Pair Overview .....	116
OTA_HotelRFP_MeetingRQ/RS .....	117
Message Pair Overview .....	117
OTA_HotelRFP_MeetingNotifRQ/RS .....	117
Message Pair Overview .....	117
OTA_HotelRFP_TransientRQ/RS .....	117
Message Pair Overview .....	117
OTA_HotelRFP_TransientNotifRQ/RS .....	118
Message Pair Overview .....	118
OTA_HotelRoomListRQ/RS .....	118
Message Pair Overview .....	118
OTA_HotelSearchRQ/RS .....	118
Message Pair Overview .....	118
OTA_HotelStatsRQ/RS.....	119
Message Pair Overview .....	119
OTA_HotelStatsNotifRQ/RS .....	120
Message Pair Overview .....	120
OTA_HotelStayInfoNotifRQ/RS .....	120
Message Pair Overview .....	120
OTA_HotelSummaryNotifRQ/RS.....	120
Message Pair Overview .....	120
<b>TRAVEL INSURANCE MESSAGES .....</b>	<b>122</b>
Messages Overview .....	122
Messages List .....	123
OTA_InsurancePlanSearchRQ/RS .....	124
Message Pair Overview .....	124
OTA_InsuranceQuoteRQ/RS .....	124
Message Pair Overview .....	124
OTA_InsuranceBookRQ/RS .....	124
Message Pair Overview .....	124
<b>PACKAGE TOUR/ HOLIDAY BOOKING MESSAGES .....</b>	<b>125</b>
Messages Overview .....	125
Messages List .....	125
OTA_PkgAvailRQ/RS .....	126

Message Pair Overview .....	126
OTA_PkgBookRQ/RS .....	126
Message Pair Overview .....	126
OTA_PkgExtrasInfoRQ/RS .....	127
Message Pair Overview .....	127
OTA_PkgCostRQ/RS .....	127
Message Pair Overview .....	127
<b>TOUR/ACTIVITY MESSAGES .....</b>	<b>128</b>
Messages Overview .....	128
Messages List .....	128
OTA_TourActivityAvailRQ/RS .....	129
Message Pair Overview .....	129
OTA_TourActivityBookRQ/RS .....	129
Message Pair Overview .....	129
OTA_TourActivityCancelRQ/RS .....	129
Message Pair Overview .....	129
OTA_TourActivityModifyRQ .....	129
Message Pair Overview .....	129
OTA_TourActivityResRetrieveRQ .....	129
Message Pair Overview .....	129
OTA_TourActivitySearchRQ/RS .....	129
Message Pair Overview .....	129
<b>TRAVEL ITINERARY MESSAGES .....</b>	<b>130</b>
Messages Overview .....	130
OTA_TravelltinineraryReadRQ/OTA_TravelltinineraryRS .....	131
Message Pair Overview .....	131
<b>RAIL MESSAGES .....</b>	<b>132</b>
Messages Overview .....	132
Messages List .....	132
OTA_RailAvailRQ/RS .....	133
Message Pair Overview .....	133
OTA_RailBookRQ/RS .....	133
Message Pair Overview .....	133
OTA_RailConfirmBookingRQ/RS .....	134
Message Pair Overview .....	134
OTA_RailConfirmBookingRQ Request Message Overview .....	134
OTA_RailConfirmBookingRS Request Message Overview .....	134
OTA_RailFareQuoteRQ/RS .....	134
Message Pair Overview .....	134
OTA_RailIgnoreBookingRQ/RS .....	135

Message Pair Overview .....	135
OTA_RailPaymentRQ/RS .....	136
Message Pair Overview .....	136
OTA_RailPriceRQ/RS .....	136
Message Pair Overview .....	136
OTA_ReadRQ/ OTA_RailResRetrieveDetailRS .....	137
Message Pair Overview .....	137
OTA_ReadRQ/ OTA_RailResRetrieveSummaryRS .....	138
Message Pair Overview .....	138
OTA_RailScheduleRQ/RS .....	138
Message Pair Overview .....	138
OTA_RailShopRQ/RS .....	139
Message Pair Overview .....	139
<b>LOYALTY MESSAGES .....</b>	<b>140</b>
Messages Overview .....	140
Messages List .....	140
OTA_LoyaltyAccountCreateRQ/OTA_LoyaltyAccountRS .....	141
Message Pair Overview .....	141
OTA_LoyaltyCertificateCreateRQ/RS .....	141
Message Pair Overview .....	141
OTA_LoyaltyCertificateCreateNotifRQ/RS .....	141
Message Pair Overview .....	141
OTA_LoyaltyCertificateRedemptionRQ/RS .....	141
Message Pair Overview .....	141
OTA_ReadRQ/OTA_LoyaltyAccountRS .....	141
Message Pair Overview .....	141
<b>PROFILE MESSAGES .....</b>	<b>142</b>
Messages Overview .....	142
Messages List .....	142
OTA_ProfileCreateRQ/RS .....	143
Message Pair Overview .....	143
OTA_ProfileMergeRQ/RS .....	143
Message Pair Overview .....	143
OTA_ProfileModifyRQ/RS .....	143
Message Pair Overview .....	143
OTA_ReadRQ/OTA_ProfileReadRS .....	143
Message Pair Overview .....	143
<b>GROUND TRANSPORTATION MESSAGES .....</b>	<b>144</b>
Messages Overview .....	144
Messages List .....	145

Implementation Best Practices .....	145
OTA_GroundAvailRQ/RS .....	147
Message Pair Overview .....	147
OTA_GroundResRetrieveRQ/RS .....	148
Message Pair Overview .....	148
OTA_GroundBookRQ/RS.....	149
Message Pair Overview .....	149
OTA_GroundCancelRQ/RS.....	150
Message Pair Overview .....	150
OTA_GroundModifyRQ/OTA_GroundBookRS.....	150
Message Pair Overview .....	150
OTA_GroundResRetrieveRQ/RS .....	151
Message Pair Overview .....	151
OTA_GroundResNotifRQ/RS .....	152
Message Pair Overview .....	152
<b>VACATION RENTAL MESSAGES .....</b>	<b>153</b>
Messages Overview .....	153
Messages List .....	153
Implementation Best Practices.....	153
OTA_XXXXXXRQ/RS .....	154
Message Pair Overview .....	154



# Introduction and Getting Started with the OpenTravel Specification

## Introducing the OpenTravel Alliance

The OpenTravel Alliance is passionate about solving the problems inherent in connecting multiple systems within the complex travel distribution arena. OpenTravel's mission is to engineer specifications that make data transmission flow smoothly throughout travel, tourism and hospitality. OpenTravel creates, expands and drives adoption of open universal data specifications, including but not limited to the use of XML, for the electronic exchange of business information among all sectors of the travel industry.

OpenTravel is comprised of companies representing airlines, car rental firms, hotels, cruise lines, railways, leisure suppliers, service providers, tour operators, travel agencies, solutions providers, technology companies and distributors. OpenTravel is a not-for-profit trade association, founded in 1999 by travel companies, with a primary focus on the creation of electronic message structures to facilitate communication between the disparate systems in the global travel industry. Tens of thousands of OpenTravel message structures are in use, carrying tens of millions of messages between trading partners every day.

## OpenTravel Alliance Resources

More information about OpenTravel can be found at:

**[The OpenTravel Corporate Website \(http://www.OpenTravel.org\)](http://www.OpenTravel.org)**

[Download an OpenTravel Specification](#)

[Sign-up for an OpenTravel Webinar](#)

[View OpenTravel News & Events](#)

[Submit a Comment](#)

**[The OpenTravel Forum Website \(http://www.OpenTravelForum.com/\)](http://www.OpenTravelForum.com/)**

[Ask a Planning/ Implementation Question](#)

[Access OpenTravel Documentation](#)

[Subscribe to an OpenTravel Mailing List](#)

[See OpenTravel-Enabled Tools/ Products](#)

**[The OpenTravel Wiki](#)**

[Access Specification Development & Implementation Resources](#)

[Access the OpenTravel Calendar & Publication Schedules](#)

**[The OpenTravel Alliance on Linked In \(http://www.linkedin.com/groups?gid=1053987\)](http://www.linkedin.com/groups?gid=1053987)**

[Join the OpenTravel Alliance on Linked in](#) (comprised of individuals from the travel industry)

## OpenTravel Specification Components

The OpenTravel Message Specification includes the following components:

- The following documents are included with the OpenTravel 2013A Publication:

----- OpenTravel\_CodeTable (folder) -----

- [OpenTravel\\_CodeList\\_2012\\_8\\_27.zip](#) - A zip file containing the OpenTravel Code Table in Microsoft Excel format. The OpenTravel Codes List spreadsheet, included with the specification download, includes an alphabetized, point and click list of all OpenTravel Code List names (with 3 character abbreviations) and errors and warnings to help implementers quickly find code list values.

----- OpenTravel2013A\_XML (folder) -----

- XML Schema files- OpenTravel XML Schema messages.

----- OpenTravel\_2013A\_Schema\_Versioning.pdf -----

- OpenTravel began tracking versions in the 2003B release cycle. Prior to the 2008A release, the version table contained 2001 and 2002 data for newly created messages. This document contains a table with each OpenTravel message version from 2006A through 2013A.

----- OpenTravel\_2013A\_Comments\_Report.pdf -----

- This document contains comments that were entered against the OpenTravel 2012B specification and resulted in changes to the 2013A specification.

----- OpenTravel\_SchemaDesignBestPracticesV3.08.pdf -----

- The OpenTravel XML Schema Best Practices document contains documentation about the standards and best practices used in creating the XML Schemas. Typically, this document is used within the OpenTravel Alliance to ensure that the creation of XML Schemas is consistent in design across the organization and across releases.

----- OpenTravel\_2013A\_XMLMessageSuite\_ReleaseNotes.pdf -----

- The release notes detail the latest information and changes for any given release. This file contains detailed documentation on changes made between the 2012B release and the 2013A release.

----- OpenTravel\_ImplementationGuide\_v1.2\_ExecSum.pdf -----

- This is an executive summary of the OpenTravel Implementation Guide. The purpose of the OpenTravel Implementation Guide is to provide invaluable information to both analysts and implementers of the OpenTravel specification on how to more easily understand and build software systems that are interoperable with other travel systems using the OpenTravel schema. The full Guide is available to members of the OpenTravel Alliance via the Wiki <http://wiki.opentravel.org/index.php/Public:Resources>.

----- OpenTravel\_FastRez (folder) -----

- OpenTravel\_FastRez.zip: A zipped file containing OpenTravel FastRez Schema, WSDL and User Documentation.

----- OpenTravel\_Other (folder) -----

- OpenTravel Hotel Schema Disability Audit Results: An OpenTravel Hotel Disability Project Brief artifact has been included with the OpenTravel 2011B 1.0 XML Schema Message publication. The OpenTravel Hotel Disability project was focused on a subset of the new 2010 ADA legislation that specifically applies to electronic hotel reservations and includes identifying and describing accessible features; reserving, upon request, accessible guest rooms or specific types of guest rooms and ensuring that the guest rooms requested are blocked and removed from all reservations systems; and guaranteeing the specific accessible guest room. Per the audit results, relatively few changes were made to OpenTravel Hotel schema, with one change made to one common hotel schema (OTA\_HotelCommonTypes) to provide a more consistent and structured way to pass measurement information, and annotation changes made to three hotel schema (OTA\_HotelCommonTypes, OTA\_RailCommonTypes and OTA\_RailPreferences) to accommodate renaming the OpenTravel Code List Physically Challenged Feature code table. The slide show included with the publication includes an overview of the project.

# Basic OpenTravel Schema Concepts

## OpenTravel Schema Design Best Practices

Each XML schema within the OpenTravel specification is built according to a well-defined set of best practices and design guidelines. These guidelines are described in the [OpenTravel Schema Design Best Practices document](#) that is included with each OpenTravel publication.

Developers of OpenTravel schema should get familiar with this document because it provides specific instructions on designing and developing schema enhancements (that meet your specific business processes) and how to submit them as comments (or as a Project Team Proposal if you are a member company.)

OpenTravel develops and maintains its library of XML schema based on its own specific guidelines so that the specification as a whole is built against the same set of design principles. This provides several benefits, including:

- A coherent OpenTravel specification. Since all OpenTravel XML schema are built according to a single, coherent set of design principles, the OpenTravel specification as a whole is easier to maintain and use.
- Increased interoperability among travel systems. Since software programs are the ultimate consumers of the OpenTravel specification, alignment to a shared XML schema design promotes software systems that are easier to develop and maintain.
- Increased reuse of XML content throughout the specification. Since a consistent set of XML content types (complex types, simple types, attribute groups, etc.) recurs throughout the specification, the level of reuse of the XML schema is greatly increased.

## OpenTravel Message Exchange Patterns

To accommodate the variations in which data exchanges between trading partners are done, OpenTravel XML schema support several distinct types of trading partner interactions, known as message exchange patterns. Each pattern serves a different purpose. For example, one pattern supports requests for information, while another supports the delivery of information to a recipient as shown below:

### *Request/Response Pattern*

This is the most prevalent message exchange pattern supported by the OpenTravel specification and is known as the request/response pattern. Any transaction that conforms to this pattern will be invoked by one trading partner requesting information explicitly from another party. The requesting party sends a message to the receiver indicating what data are being requested; this is the request part of the transaction. The receiver then processes the request and returns the relevant data to the requester; this is the response part of the transaction. You'll find that most of the message-level XML schema in the OpenTravel specification conform to this exchange pattern and an example is OTA\_CruiseBookRQ (which represents the request) and OTA\_CruiseBookRS which represents the response.

### *Notif Pattern*

This is another message exchange pattern supported by the OpenTravel specification and is known as the notification or “notif” pattern. Any transaction that conforms to this pattern involves the delivery of data from one trading partner to another party without the request to do so. Therefore, this pattern can be understood as a “push” of data to another party as opposed to a “pull” of data, which describes the request and response pattern. The sender delivers (pushes) a message to one or more predefined recipients. Any recipient of that message will then process the data as needed and provide a response back to the sender indicating that the payload was received.

## Generic Messages

In addition to the industry-specific message-level XML schema that constitute the core of the OpenTravel specification, several other message pairs support generic functions that are not tied to a specific industry or business process, but, rather, are shared by all industries. Some examples of generic OpenTravel Messages are:

OTA\_AuthorizationRQ/RS—which is typically used to verify that a customer’s bank account is valid for payment, and OTA\_CancelRQ/RS—which are used to cancel all or portions of reservations. This Message Users Guide contains a list of OpenTravel generic messages with sample use cases.

## Payload Transaction Management

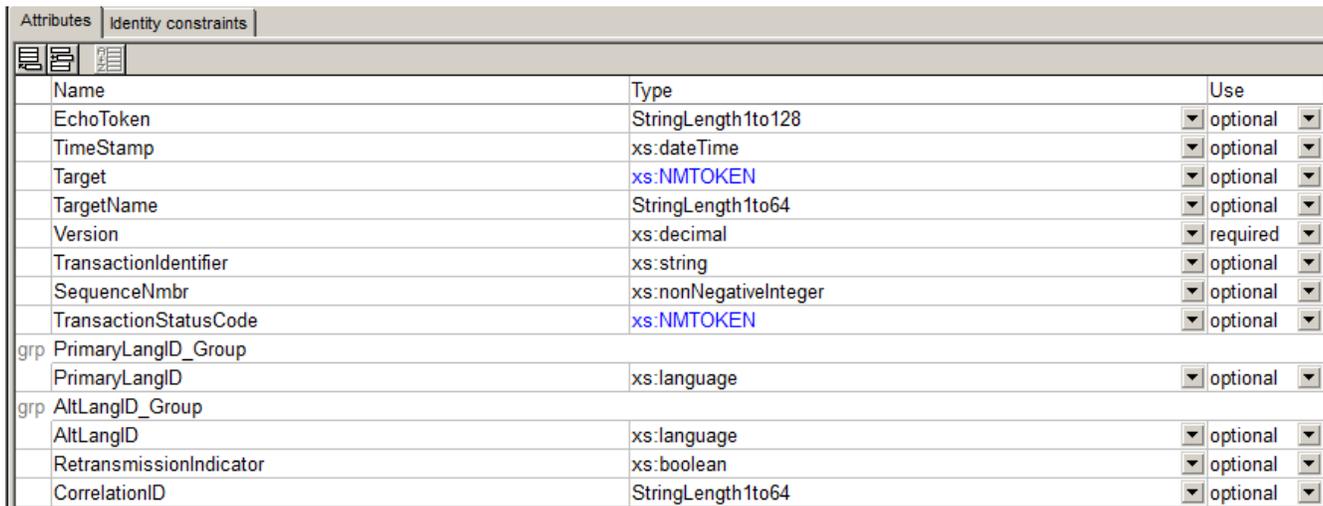
Commonly, the first encounter that an OpenTravel specification user has with non-functional requirements is within the root element of each message. The root element of each message level schema refers to the OTA\_PayloadStdAttributes attribute group. This attribute group is defined in the OTA\_CommonTypes file and contains attributes that, collectively, support common transaction management requirements.

At run-time, these attributes operate within the XML payload rather than the envelope layer. Implementers should consider carefully their use of these fields as compared to fulfilling the same functionality by way of the message envelope. The image below depicts the appearance of this attribute group when viewed in an XML schema editor.

**EchoToken.** The value of this attribute is assigned by a requesting system and serves as a reference for messages that may follow an initial request. When a request message includes an echo token, the corresponding response message must include an echo token with an identical value. This makes it possible for two or more messages to be correlated together as part of a single transaction.

**TimeStamp.** This date and time value specifies the creation date and time of a message. This value is UTC format specified by ISO 8601—YYYY-MM-DDThh:mm:ssZ—with time values using the 24-hour clock (e.g., 20 November 2003, 1:59:38 p.m. UTC becomes 2003-11-20T13:59:38).

**Target.** This enumerated attribute indicates whether a request message is targeted at a test or production system.



Name	Type	Use
EchoToken	StringLength1to128	optional
TimeStamp	xs:dateTime	optional
Target	xs:NMTOKEN	optional
TargetName	StringLength1to64	optional
Version	xs:decimal	required
TransactionIdentifier	xs:string	optional
SequenceNmbr	xs:nonNegativeInteger	optional
TransactionStatusCode	xs:NMTOKEN	optional
grp PrimaryLangID_Group		
PrimaryLangID	xs:language	optional
grp AltLangID_Group		
AltLangID	xs:language	optional
RetransmissionIndicator	xs:boolean	optional
CorrelationID	StringLength1to64	optional

**Version.** This decimal value identifies the specific version of the OpenTravel XML schema associated with an XML instance document. This value makes it possible for trading partners to maintain several versions of the same schema and correlate instances to the appropriate version.

**TransactionIdentifier.** This value uniquely identifies a transaction and all messages that make up that transaction. This value would be sent in all request and response messages that are part of an ongoing transaction.

**SequenceNumber.** This integer value identifies the sequence number of a transaction. This allows for an application to process messages in a certain order or to request a resynchronization of messages in the event that a system has been offline and needs to retrieve messages that were missed.

**TransactionStatusCode.** This enumerated attribute indicates where this message falls within a sequence of messages (Start, End, Rollback, InSeries).

**PrimaryLangID.** The language code associated with this attribute indicates the primary language for content provided within an XML instance document. The ability for a message sender to request a particular language within the response message is not supported by OTA\_PayloadStdAttributes.

**AltLangID.** The language code associated with this attribute specifies the secondary language for content provided within an XML instance document.

**RetransmissionIndicator.** This boolean attribute indicates whether a message is being re-sent due to a previously failed attempt to transmit a message.

The section below depicts a sample XML instance document and its use of several attributes contained within the OTA\_PayloadStdAttributes attribute group.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- OPENTRAVEL SAMPLE INSTANCE -->
<!-- USE CASE SCENARIO: #1: One Passenger Ride to Airport for Non-Corporate Account -->
<!-- USE CASE DETAILS:
1. Jane Smith needs ground transportation to an airport and goes to the Academy Limousine site because
they have sent her a 10% off coupon for an airport transfer.
2. From the Academy Limousine website, Jane does an availability request for June 15 at 9:00 AM. (THIS
INSTANCE)
3. Jane receives two ground transportation availability responses, one for an Executive Sedan and one
for a Luxury Sedan. She reviews the pricing, fees, deposit and cancellation information for each
service. (OTA GroundAvailRS AirportTransfer.xml)
4. Jane selects the Executive Sedan and books the ground transportation service.
(OTA_GroundBookRQ_AirportTransfer.xml)
5. She receives a booking response that includes the details of her ground transportation reservation.
(OTA_GroundBookRS AirportTransfer.xml)
6. A week after making the ground transportation reservation, Jane has had to change her flight to be
two hours later (11:00 AM). She phones Academy Limousine to change her reservation, but does not know
her reservation number.
7. The Academy Limousine representative searches for Jane's reservation by entering her last name and
date of service (OTA_GroundResRetrieveRQ_AirportTransferQuery.xml). She receives 2 responses and
identifies Jane's reservation (OTA_GroundResRetrieveRS AirportTransferQuery.xml)
8. The Academy Limousine representative then issues a reservation retrieve request using the
confirmation number returned in step 7 above and receives a detailed reservation response
(OTA_GroundResRetrieveRQ_AirportTransferDetail.xml/ OTA_GroundResRetrieveRS_AirportTransferDetail.xml)
-->
<OTA GroundAvailRQ xmlns="http://www.opentravel.org/OTA/2003/05"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opentravel.org/OTA/2003/05 OTA_GroundAvailRQ.xsd" EchoToken="12345"
TimeStamp="2011-05-01T09:44:52" Target="Production" Version="1.000" PrimaryLangID="en-us">
  <!-- Point of sale information for the sender of the availability request. -->
  <POS>
    <Source>
      <RequestorID Type="43" ID="Academy Limousine"/>
      <!-- Type 43 = "Ground Transportation Supplier (branded website)". See
OpenTravel Code List Unique ID Type (UIT). -->
    </Source>
  </POS>
...

```

## OpenTravel File Naming Conventions

Each OpenTravel schema is named according to a standard naming convention that dictates that each file name should be identical to the root element name within that schema. Therefore, each file begins with an “OTA\_” prefix and concludes with the remainder of the root element name. For example, the root element name of the “OTA\_VehResRQ.xsd” schema file is “OTA\_VehResRQ”.

OpenTravel mandates a naming convention for all XML schema that model a request and response transaction. For example, the name of all request messages ends with an “RQ” suffix and the name of all response messages ends with an “RS” suffix.

Because of this, newcomers to the OpenTravel specification usually find it relatively easy to identify the XML schema that pertain to a particular message exchange pattern.

One thing to remember is that OpenTravel schema are named so that the files associated with each industry (like air, car, hotel and cruise) are categorized and easily identified for each industry.



## OpenTravel Code Lists and Enumerations

XML message exchanges commonly require the ability to restrict a field to a limited set of values.

Address Use Type	AUT	Code List Acronym
	1	Delivery
	2	Mailing
	3	Billing
	4	Credit card
	5	Other
	6	Contact
	7	Physical
	8	Pre-opening office
	9	Collection
	10	Chain
		Deposit
	12	Hotel
	13	Permanent

Diagram annotations: A blue arrow labeled 'Code List Name' points to the 'Address Use Type' header. A blue arrow labeled 'Code Value' points to the 'AUT' column. A blue arrow labeled 'Code Name' points to the description column.

Business requirements may control this, but it is often preferred to limit the allowable values for an element or attribute to as few as possible to facilitate validation of the values. To support this need, the OpenTravel specification uses XML enumerations and code lists.

The OpenTravel XML schema contain many elements and attributes that define the allowable values for the respective field. Enumerations are used when the list of values is static or when it is unlikely that values will be added. Some examples of this are: Days of the week and gender.

In addition to complex, simple, and native data types (string, integer), the OpenTravel XML schema contain many code values.

Most of these values refer to code lists main-

tained by OpenTravel and each code refers to a specific value contained in one of the code lists. Code lists are preferred over enumerations when the collection of values they represent are likely to change and grow. The OpenTravel code lists are included with each OpenTravel Specification publication and, as a whole, are referred to as the OpenTravel code table. Below you’ll find a sample OpenTravel XML code list reference:

```
<xs:attribute name="UseType" type="OTA_CodeType" use="optional">
  <xs:annotation>
    <xs:documentation xml:lang="en">Describes the use of the address (e.g. mailing, delivery, billing, etc.) Refer to OTA Code List Address Use Type (AUT).</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

All code list references in the OpenTravel schema are typed with the OTA\_CodeType simple type and the specific code list associated with an XML attribute is defined within the XML schema annotation for that attribute. For your benefit, the code table is provided in spreadsheet format as well as in an XML instance.

## External Code Lists

In addition to codes maintained by OpenTravel, the OpenTravel XML schema contain some references to code lists maintained by other standards organizations. The snippet below shows an excerpt from the OTA\_AirCommonTypes schema, which refers to an International Air Transport Association (IATA) location code:

```
<xs:attribute name="LocationCode" type="StringLength1to8" use="optional">
  <xs:annotation>
    <xs:documentation xml:lang="en">A 3 character ATA/IATA city code of telephone
location.</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Notice that the data type associated with this attribute is not the OTA\_CodeType simple type that is used for any code maintained by OpenTravel. Rather, the StringLength1to8 simple type is used to allow any value of 1 to 8 characters in length. Individual trading partners would be responsible for ensuring that each code passed is a valid IATA code for example.

### *External Codes Lists & Other References in OpenTravel Schema*

[ABTA \(Association of British Travel Agents\)](#)

[ATA \(Air Transport Association\)](#)

[DUNS \(Dun & Bradstreet\)](#)

[ERSP \(Electronic Reservation Service Provider - IATA\)](#)

[IATA \(International Airport Transport Association\)](#)

[ISO \(International Organization for Standardization 3166 Country Codes\)](#)

[ISO \(International Organization for Standardization 639 Language Codes\)](#)

[ISO \(International Organization for Standardization 8601 Date Format\)](#)

[ISO \(International Organization for Standardization 4217 Currency Format\)](#)

[ISO \(International Organization for Standardization 6709 Geographic Point Location by Coordinates\)](#)

[APEX initiative](#)

[UIC \(International Union of Railways\)](#)

## OpenTravel Message Success, Warnings & Errors Elements

Any OpenTravel message exchange requires that the participating systems have a mechanism to share data about the processing of that message and whether it was successful or not. The OpenTravel specification addresses this need by providing a consistent set of elements within all response XML schema. These elements allow the responding party to indicate if the message was processed successfully or if any errors or warnings were encountered.

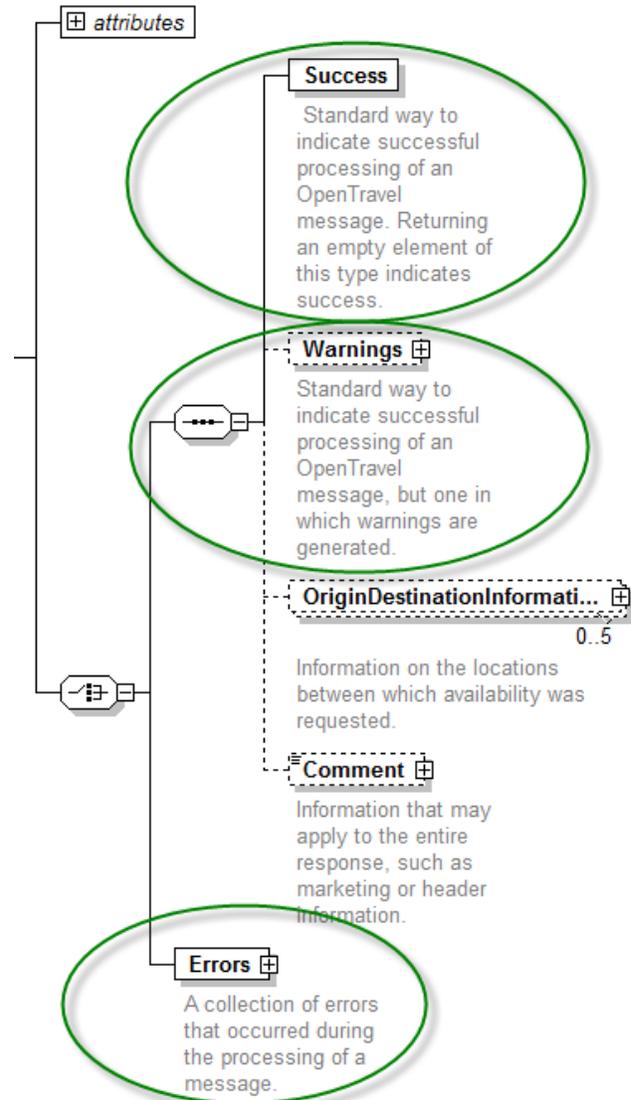
The elements are:

**Success:** an element that is not intended to contain any data. The mere presence of a success element within a response message indicates that the incoming message (request or notification) was processed successfully.

**Warnings:** This element indicates that the recipient of a message identified one or more warnings. A warning is a business-level error. A warning would be generated, for example, by a request to book a flight that does not exist, a request to book a hotel room when none are available for the dates requested or an incorrect message password or requester ID within the request message. The XML content might be syntactically valid, but the receiving application cannot fulfill the request.

**Errors:** This element indicates that an error occurred in the processing of an incoming message (request or notification). An error is defined as a critical error caused by corruption of a message in transit or a communication failure. An error would be generated, for example, by an unforeseen disruption of an application or its connection to a trading partner application. The data structure of this element is identical to the Warnings element with the exception of the NodeList attribute, which provides an XPath expression that identifies the nodes that caused the error.

Remember that the relationship of these XML schema elements is significant, because it reflects the response data structure returned to the sender of an incoming message. The root element can contain either errors or other data (success, warnings, or response information) and if errors are present, the Errors element may identify the XML content to which each error applies.



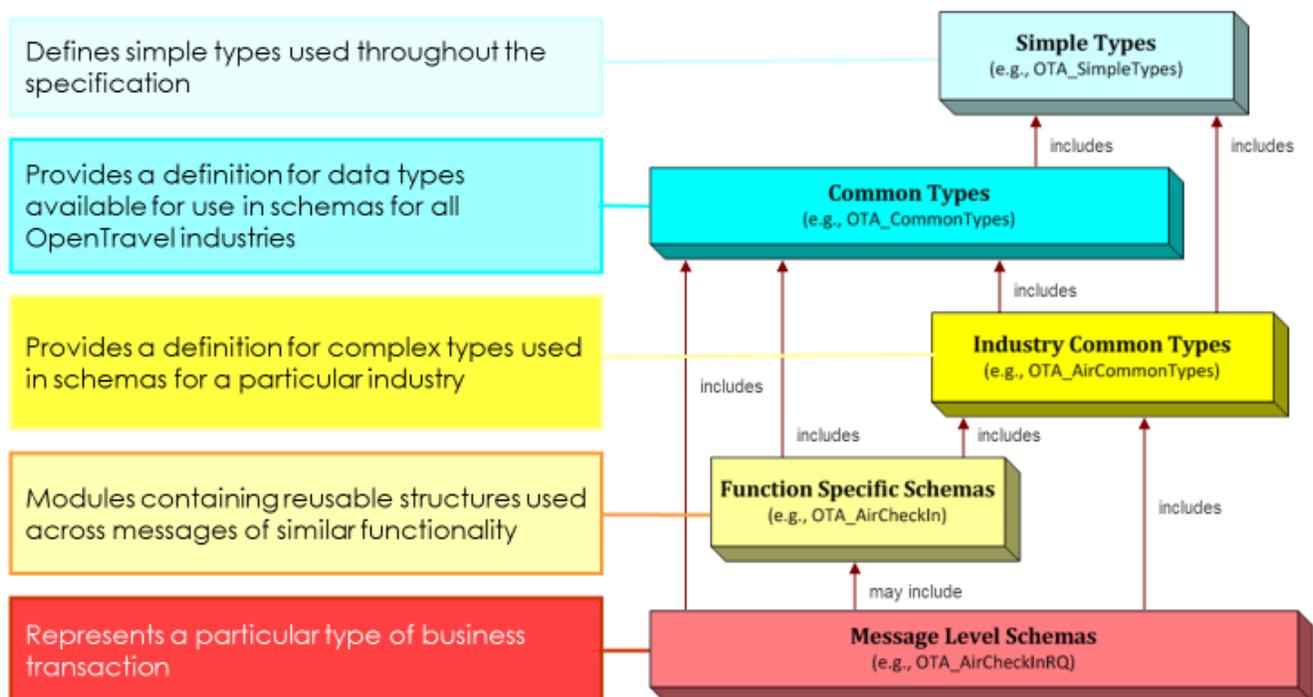
## Schema Architecture

### *Understanding OpenTravel Schema Architecture*

The OpenTravel specification is a large collection of XML schema that are organized hierarchically. This design enables an efficient method for maintaining, understanding, and implementing the specification. It also leverages proven design techniques from XML schema design as well as other software disciplines such as object orientation.

Any data exchange that uses the OpenTravel specification may require content from several messages within the OpenTravel schema hierarchy. For example, an airline reservation booking message sent from one trading partner may involve validation by an XML schema that includes several other schema. Those files may define much of the content (e.g., elements, attributes, types) that comprises that booking request message.

Accordingly, an OpenTravel implementer can expect to use a hierarchy of XML schema files, rather than a single XML schema file, for a given message exchange as shown and described below:



## *OpenTravel Message Level Schema*

Each message-level XML schema in the OpenTravel specification represents a particular type of business transaction. Some examples of message-level schema are: [OTA\\_AirBookRQ](#), [OTA\\_HotelResModifyRQ](#), and [OTA\\_CruiseBookRQ](#).

The OpenTravel organizational structure is divided into four work groups: Hospitality, Transport, Architecture and Travel Integration and this structure is reflected in the message-level XML schema.

Each message-level XML schema references other XML schema that, in turn, reference others. These message-level schema contain the declaration of the root element that appears as the primary node of any XML instance document based on that XML schema. The content contained within that root node may be defined in the message-level schema itself or in files it includes.

## *OpenTravel Industry Common Types Schema*

Because the OpenTravel specification provides numerous XML schema for each industry, all schema that pertain to a specific industry will naturally share common data types. The OpenTravel modularity model supports this need by containing XML schema files intended for use throughout the set of XML schema for each industry.

*For example*, the [OTA\\_AirBookRQ](#) and the [OTA\\_AirPriceRQ](#) schema both need to accommodate information related to an air itinerary. Rather than having each file have to define that content separately, the [OTA\\_AirCommonTypes](#) schema file provides a single definition of that data for use throughout the OpenTravel air schema. This single data definition is manifested as a complex type named `AirItineraryType`.

## *OpenTravel Common Types Schema*

The OpenTravel specification provides a common types file called [OTA\\_CommonTypes](#) that contains data structures that are referenced throughout the specification and shared by each of the respective industries.

The data types contained in this file may be referenced from message-level files, function-specific files, and industry common files.

Keep in mind that any of these files can declare elements that reference complex types, simple types, and attribute groups in the [OTA\\_CommonTypes](#) file.

## *OpenTravel Simple Types Schema*

The [OTA\\_SimpleTypes](#) XML schema serves as a module that contains simple types that are available for use throughout the specification.

The design goal of this file is identical to that of the [OTA\\_CommonTypes](#) file in that it serves as a single container for global simple types.

These types can be used at any level of the schema hierarchy. Most of the simple data types contained within this schema provide basic data structures that the other schema use to build larger collections of data.

# OpenTravel Message Transport

## Introduction

Historically, the primary work of OpenTravel has been to provide a data exchange specification for the travel industry. As use of the specification increased, however, the implementer community began expressing a need for OpenTravel to also provide guidance regarding how these schemas should be used within real-world software applications.

The OpenTravel architecture work group responded to these requests by launching projects to provide guidance about nonfunctional areas such as HTTP, SOAP, and WSDL. As is the case with the schemas, guidance documentation for these areas will be maintained over the long term to reflect changing needs and a continually evolving standards and technology market.

While OpenTravel recognizes that these standards are developed and maintained by the W3C, it also recognizes that it can help reduce efforts associated with interoperability in the travel industry by publishing implementation guidance documents that help travel companies implement web services in a consistent manner. The sections below provide guidance for using OpenTravel XML schemas in conjunction with the SOAP and HTTP messaging protocols.

## Definitions & Conventions

Each of the sections below provide guidelines regarding the use of a technology in conjunction with the OpenTravel specification. Guidelines may be denoted by the symbol '\$'. Since this document is recommended guidance and not an enforceable requirement (from the perspective of OpenTravel), the guidelines herein make use (primarily) of SHOULD nomenclature.

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this document are to be interpreted as described in [IETF document RFC 2119](#).

In other words, "MUST," "MUST NOT," "REQUIRED," "SHALL," and "SHALL NOT" identify protocol features that are not optional.

Similarly, "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" identify statements that require consideration with regard to the impact that inclusion or exclusion of the protocol feature will have on the interoperability of the system.

## State Maintenance

In general, OpenTravel messages were designed to support transactions that occur within a stateless environment. Within the messages, this is reflected by a tendency to include data in the response that was defined in the request. With this design, message senders and recipients are not forced to "remember" data between message sends.

As an example of this design philosophy, the image below depicts the OTA\_AirAvailRQ and OTA\_AirAvailRS as they appear in an XML schema editor. For each message, the highlighted element OriginDestinationInformation contains content that is similar across the request and response message.

Please note that a number of OpenTravel messages are intended for use within legacy environments that do maintain state. As a result, those schemas do not include data elements in the response that were in the request.

## SOAP Messaging

SOAP is a loosely coupled protocol for interaction between applications that may have been developed on different platforms using different languages. SOAP uses XML for payload and a wide range of underlying protocols for transport - one of them being HTTP.

Earlier proprietary protocols required client and server applications to be of the same breed, use a common protocol framework or at least run on the same platform. With the introduction of SOAP, this tight coupling between client and server applications was eliminated and applications could communicate across platform and language barriers.

Many applications are already using SOAP for the interchange of OpenTravel documents, and more are in the process of being developed. Until now there have not been any guidelines or specifications describing how OpenTravel documents should be encapsulated and transported in SOAP messages. This has prevented many existing applications from inter-operating, even when these applications conform to the SOAP and OpenTravel specifications. This section aims to bridge that gap and acts as guide to software architects and developers who want to create interoperable OpenTravel clients and services using SOAP.

### Purpose

The main purpose of this section is to provide well-defined guidelines for creating and using OpenTravel services over SOAP in a manner that ensures interoperability. The goal is that any party that develops OpenTravel-based services in accordance with this section should be able to interoperate without having to implement any changes in SOAP/OpenTravel message structure or at a protocol level. This section also attempts to outline the simplest method of transmitting OpenTravel messages over SOAP. With simplicity comes ease of implementation and in most cases also efficiency. OpenTravel recommendation of these guidelines will help to ensure wide distribution amongst implementers within the travel industry.

### The SOAP Transport Protocol

Microsoft started working on the SOAP specification as early as 1998, and a specification was first released in late 1999. This specification used a sub-set of the then in-progress XML Schema specification. SOAP was initially intended as a replacement for existing RPC protocols such as GIOP/IIOP, DCE/DCOM, RMI, and ONC, but has also developed into a document exchange transport mechanism with similarities to EDI. SOAP was submitted to W3C's XML Protocol Working Group in 2000. SOAP was released as a "Note" from W3C in May 2000.

SOAP provides a simple and extensible vehicle for interchanging data and invoking remote services using XML, as demonstrated by the skeleton SOAP structures in *Figures 1, 2 and 3*:

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <!-- Routing, security or other control data. -->
  </soap:Header>
  <soap:Body>
    <!-- RPC method call or document data. -->
  </soap:Body>
</soap:Envelope>
```

*Figure 1: Skeleton SOAP response*

The only limitation to the content of the SOAP Body element is that it must be a valid XML document. This implies that the content is a single XML element, which either represents a remote service/method or is the root element of an XML document that is being exchanged.

The Header element is optional in all SOAP messages and is used to carry information apart from the actual envelope payload – such as routing or security information.

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
```

```

    <soap:Header>
    <!-- Routing, security or other control data. -->
    </soap:Header>
    <soap:Body>
    <!-- RPC method response or document data. -->
    </soap:Body>
</soap:Envelope>

```

Figure 2: Skeleton SOAP positive response

The same limitation with regard to the content of the SOAP Body element applies to positive SOAP response messages.

```

<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
  <!-- Routing, security or other control data. -->
  </soap:Header>
  <soap:Fault>
  <faultcode>soap:Client.AppError</faultcode>
    <faultstring>Application Error</faultstring>
    <detail>
      <message>Something went wrong</message>
      <errorcode>12345</errorcode>
    </detail>
  </soap:Fault>
</soap:Envelope>

```

Figure 3: Skeleton SOAP negative response

Negative response messages contain a Fault element in place of a Body element.

### The Need for Interoperability

SOAP is already being widely used for transporting OpenTravel documents. However, due to the dual purpose of SOAP (RPC vs. messaging) and SOAP's flexibility with regards to structure, a whole range of SOAP structures are being used to transport OpenTravel data.

Many travel companies expose OpenTravel or OpenTravel -like services using SOAP, with different implementation approaches:

```

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <OTA_CancelRQ xmlns="http://www.opentravel.org/OTA/2003/05" Version="2.001">
      <POS>
        <Source ISOCountry="US" ISOCurrency="NOK" PseudoCityCode="HUR">
          <RequestorID ID="abc.123" URL="abcde..."/>
        </Source>
      </POS>
      <UniqueID ID="DGNJ6012990-389" Type="14"/>
    </OTA_CancelRQ>
  </soap:Body>
</soap:Envelope>

```

Figure 4: SOAP messaging

Using SOAP messaging, the OpenTravel payload is the only and immediate child of the SOAP Body element. This is the simplest and most efficient means of transporting OpenTravel messages over SOAP.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <acme:otaServiceCancel xmlns:acme="http://www.acme-travel.com">
      <OTA_CancelRQ xmlns="http://www.opentravel.org/OTA/2003/05" Version="2.001">
        <POS>
          <Source ISOCountry="US" ISOCurrency="NOK" PseudoCityCode="HUR">
            <RequestorID ID="abc.123" URL="abcde..." />
          </Source>
        </POS>
        <UniqueID ID="DGNJ6012990-389" Type="14" />
      </OTA_CancelRQ>
    </acme:otaServiceCancel>
  </soap:Body>
</soap:Envelope>
```

Figure 5: SOAP RPC

With SOAP RPC the immediate child of the SOAP Body element is an XML element that describes the method or function that is being invoked, in this case “otaServiceCancel”. The namespace of this element is most often defined by the service application (it is not the OpenTravel namespace).

```
Accept-Language: en
Content-Type: multipart/related; type="text/xml"; boundary="-----..."
Content-Id: soap-envelope
Content-Length: 923
SOAPAction: OTA_CancelRQ
User-Agent: Acme SOAP Client

-----=Multipart_Boundary_1136385692441_3=-----
Content-Type: text/xml
Content-Id: soap-envelope

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <acme:otaServiceCancel xmlns:acme="http://www.acme-travel.com">
      <acme:payload location="cid:OTA_CancelRQ" />
    </acme:otaServiceCancel>
  </soap:Body>
</soap:Envelope>
-----=Multipart_Boundary_1136385692441_3=-----
Content-Type: text/xml; charset="UTF-8"
Content-Id: OTA_CancelRQ

<OTA_CancelRQ xmlns="http://www.opentravel.org/OTA/2003/05" Version="2.001">
  <POS>
    <Source ISOCountry="US" ISOCurrency="NOK" PseudoCityCode="HUR">
      <RequestorID ID="abc.123" URL="abcde..." />
    </Source>
  </POS>
  <UniqueID ID="DGNJ6012990-389" Type="14" />
</OTA_CancelRQ>
```

Figure 6: SOAP RPC with attachment

All the above are taken from real-world use of SOAP for interchange of OpenTravel messages. There is nothing inherently wrong with any of these methods; they all use SOAP to carry OpenTravel messages, but they are not interchangeable.

The need for a well-defined specification in this area is obvious due to the increased use of OpenTravel messages, an increasing number of SOAP-based OpenTravel services, and the variety of SOAP structures that can be used to carry OpenTravel messages.

## Guidelines for OpenTravel Documents & Messages

This section describes a set of guidelines for a uniform method of using SOAP for interchange of OpenTravel messages. These guidelines do not, in any way, state that OpenTravel services or clients must use SOAP for transport, nor must they conform to these guidelines. These *guidelines* can be used to maximize interoperability with other SOAP-based OpenTravel clients and services. This section is not intended to contain contradiction of or repetition of what is already stated as rules or guidelines in other third party specifications:

Specification	Location
SOAP 1.1 specification	<a href="http://www.w3.org/TR/2000/NOTE-SOAP-20000508/">http://www.w3.org/TR/2000/NOTE-SOAP-20000508/</a>
SOAP 1.2 specification	<a href="http://www.w3.org/TR/soap12/">http://www.w3.org/TR/soap12/</a>
SOAP with attachments	<a href="http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211">http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211</a>
WS-I basic profile	<a href="http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html">http://www.ws-i.org/Profiles/BasicProfile-1.0-2004-04-16.html</a>
WS-Security specification	<a href="http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss">http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss</a>
XML-Signature specification	<a href="http://www.w3.org/TR/xmlsig-core/">http://www.w3.org/TR/xmlsig-core/</a>
XML-Encryption specification	<a href="http://www.w3.org/TR/xmlenc-core/">http://www.w3.org/TR/xmlenc-core/</a>
WSDL 1.1 recommendation	<a href="http://www.w3.org/TR/wsdl">http://www.w3.org/TR/wsdl</a>

### SOAP Version 1.1 and 1.2

SOAP version 1.1 was released as a “Note” from W3C in 2000. This specification had few changes from what was submitted to W3C’s XML Protocol Working Group the same year. The main addition was a fixed namespace (<http://schemas.xmlsoap.org/soap/envelope/>).

The main change in SOAP version 1.2, released as a W3C “Recommendation” in June 2003, was a change in the namespace (now <http://www.w3.org/2003/05/soap-envelope>) and a change in how the SOAP Action value was transmitted over HTTP.

```
Accept-Language: en
Content-Type: type="text/xml";
Content-Length: 194
SOAPAction: GetStockQuote
User-Agent: Acme SOAP Client

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getStockQuote Ticker="SUNW"/>
  </soap:Body>
</soap:Envelope>
```

Figure 1: SOAP 1.1 HTTP Sample

```

Accept-Language: en
Content-Type: type="text/xml"; action="GetStockQuote";
Content-Length: 193
User-Agent: Acme SOAP Client

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
<soap:Body>
  <getStockQuote Ticker="SUNW"/>
</soap:Body>
</soap:Envelope>
.
.
.

```

Figure 2: SOAP 1.2 HTTP Sample

Most SOAP toolkits/stacks support both versions of SOAP. By using a standard development kit or SOAP stack one ensures that an application will support both SOAP versions.

§ 1. OpenTravel implementers using the SOAP protocol MUST support either SOAP 1.1 or SOAP 1.2 for clients and services.

§ 2. OpenTravel implementers using the SOAP protocol SHOULD support SOAP 1.1 and SOAP 1.2 for clients and services.

## SOAP Messaging and SOAP RPC

### *Differences between SOAP Messaging and SOAP RPC*

The SOAP protocol can be used both for messaging (exchanging messages/documents) and for RPC (invoking remote services). The structure of the SOAP Envelope, Header and Body elements are the same for both paradigms, but the content of the SOAP Body element is different.

When using SOAP messaging for document exchange, the content of the SOAP Body element is a document or message that is sent from one party to another. This document or message is the only content inside the SOAP Body, and the root element of the document or message is the only immediate child element of the SOAP Body (i.e., you cannot transmit more than one document inside a single SOAP Body element). The exchanged document or message is often defined using an XML schema that is either agreed upon between the two SOAP parties or defined by a standards body (such as OpenTravel).

When using SOAP RPC the content of the SOAP Body element describes a service invocation (a remote procedure call). This description normally includes a procedure or function name and a set of parameters. The parameters can be one or more XML documents or messages, so it is possible to exchange documents and messages using SOAP RPC. The structure of the SOAP Body contents is defined on a per-service basis, and is most often described in a WSDL document that is made available by the SOAP RPC service.

### *WSDL Document/ Literal Binding*

WSDL can be used to describe both RPC and document exchange services. For document exchange services, the document/literal binding must be used (described in the [W3C WSDL 1.1 specification sections 3.3 through 3.5](#)):

```

<wsdl:definitions ...>
  ...
  <wsdl:binding name="mySoapBinding" type="impl:SOAPClientInterface">
    <!-- Use document style and not rpc. -->
    <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <!-- Define OTA_Cancel operation using OpenTravel-defined messages. -->

```

```

<wsdl:operation name="OTA_Cancel">
  <wsdlsoap:operation soapAction="OTA_Cancel"/>
  <wsdl:input name="OTA_CancelRQ">
    <wsdlsoap:body namespace="http://www.acme.com/ota" use="literal"/>
  </wsdl:input>
  <wsdl:output name="OTA_CancelRS">
    <wsdlsoap:body namespace=" http://www.acme.com/ota " use="literal"/>
  </wsdl:output>
</wsdl:operation>
</wsdl:binding>
.
.
.

```

Figure 3: WSDL Document/Literal Binding Example

## SOAP Messaging Benefits

The main benefit of using SOAP messaging for exchange of OpenTravel messages is that the content of the SOAP Body element is always a single OpenTravel message. This means a SOAP client can use the exact same message structure for any OpenTravel service that uses SOAP messaging. If using SOAP RPC, each OpenTravel service would define its own SOAP message structure (procedure name, etc.), and a client would have to use a different message structure for each SOAP RPC service.

As an example, we can consider a SOAP client that wants to issue an `OTA_AirLowFareSearchRQ` request to a set of OpenTravel services: A, B and C. If using SOAP messaging, the client could construct a single SOAP envelope containing an `OTA_AirLowFareSearchRQ` and issue it to services A, B and C. If the services were using SOAP RPC, the client would have to first determine the required SOAP RPC message structure for services A, B and C, create one SOAP request for each service (containing the `OTA_AirLowFareSearchRQ` as a parameter) and issue the request to each service.

## SOAP Messaging vs. RPC Guidelines

The overhead of using RPC is obvious, and even more so when considering the similar complexity in handling SOAP RPC response messages. But the main advantage to using SOAP messaging is ease of integration and inter-operation.

§ 3. OpenTravel implementers SHOULD use SOAP messaging for transmitting OpenTravel documents between a SOAP client and a SOAP service.

## SOAP Action URI

### SOAP Messaging and the SOAP Action URI

The SOAP Action is a URI that is transmitted with a SOAP Envelope, separate from the envelope's XML content. This URI is meant to carry the *intent* of the SOAP message. With SOAP messaging the SOAP operation (service name) is not included in the SOAP message. The SOAP Action can be used to send an operation name from the client to the service in cases where the operation is not implied by the type of document that is being sent.

## SOAP Intermediaries

The SOAP Action URI can also be very valuable to SOAP intermediaries, such as a SOAP gateway, router or proxy. Intermediaries can forward or otherwise handle a SOAP message based on its SOAP Action value, without having to parse the SOAP Envelope.

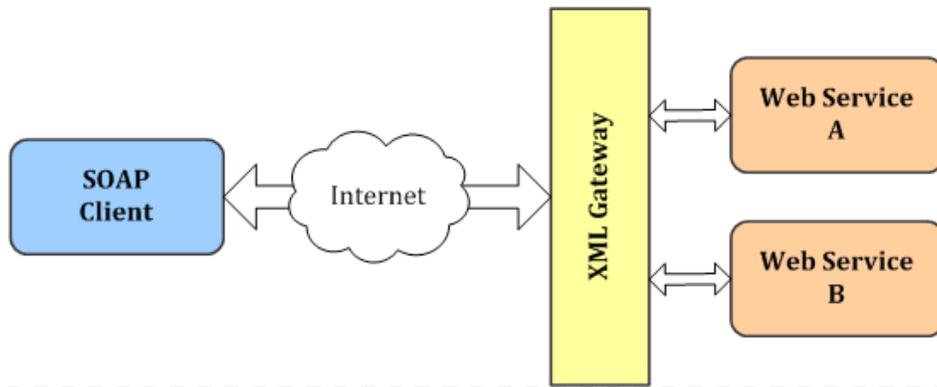


Figure 1: SOAP Intermediary

Tasks carried out by such intermediaries include:

- Routing of SOAP requests – routing based on SOAP Action value.
- XML Schema validation of SOAP request contents – XML Schema is chosen from the SOAP Action.
- XML Security token processing, such as XML Signature validation and decryption of XML Encryption elements.

### SOAP Action URI Guidelines

In scenarios where there is no SOAP intermediary there is most often no need for a SOAP Action URI. The server application can process an incoming message purely based on the root tag of the document it receives.

§ 4. Services SHOULD NOT require a SOAP Action URI to communicate with the service.

In scenarios where SOAP intermediaries are used, a SOAP Action URI may be used for in-transit processing. A SOAP server application may also choose to use the SOAP Action URI for internal routing (at an application level).

§ 5. Services and intermediaries that require a SOAP Action URI SHOULD use the OpenTravel request tag root as the SOAP Action URI. For low fares search, the SOAP Action URI would be “OTA\_AirLowFareSearchRQ”.

Some SOAP intermediaries may not allow an administrator to configure the SOAP Action URI. For this reason a SOAP Client application should allow the SOAP Action URI to be configured per service:

§ 6. Clients SHOULD support any SOAP Action URI for transmitting any OpenTravel message to a Web service.

### SOAP Envelope Content

The structure of the SOAP Envelope is well defined in the SOAP 1.1, SOAP 1.2 and WS-I basic profile specifications. WS-I basic profile section 4.1 contains a very detailed description of a properly defined SOAP Envelope. This description does not add, remove or change any of the requirements in SOAP 1.1 or 1.2, but offers the reader a better understanding of the SOAP Envelope structure and constraints than the SOAP documents.

§ 7. SOAP Envelope content SHOULD conform to the SOAP 1.1 and SOAP 1.2 specifications, as clarified in WS-I basic profile section 4.1.

### SOAP Header Content

The SOAP Header element is used to carry information that is logically separate from the SOAP payload (contents of the SOAP Body element) but that are required for the SOAP message to reach its endpoint. Such information includes:

- Security tokens
- Routing information
- Timestamps

This element is not intended to carry all or portions of a SOAP message's payload.

§ 8. SOAP Header elements in SOAP request and response messages SHOULD conform to the SOAP 1.1 and SOAP 1.2 specifications, as clarified in WS-I basic profile section 4.1.

§ 9. SOAP Header elements SHOULD NOT contain OpenTravel data.

### *SOAP Body Content*

As SOAP messaging is recommended, the content of the SOAP Body element should be a single OpenTravel request or response message. The only immediate child element of the SOAP Body should be the root element of an OpenTravel message.

§ 10. SOAP Body elements in SOAP request and response messages SHOULD conform to the SOAP 1.1 and SOAP 1.2 specifications, as clarified in WS-I basic profile section 4.1.

§ 11. All content within the SOAP Body element SHOULD be in the OpenTravel namespace, with the exception of XML-Encryption tokens that are inserted in-place of encrypted OpenTravel elements.

§ 12. The content within a SOAP Body element SHOULD be valid, well-formed XML that conforms to an OpenTravel schema.

§ 13. The only immediate child element of the SOAP Body element SHOULD be the root element of a document that is defined in an OpenTravel schema.

#### 3.2.4.13 SOAP Attachments

SOAP attachments are used to carry information that is in addition to the message payload.

§ 14. SOAP clients SHOULD support SOAP Attachments.

§ 15. SOAP services SHOULD limit the use of SOAP Attachments to images, such as hotel and vehicle images.

### *SOAP Attachments*

SOAP attachments are used to carry information that is in addition to the message payload.

§ 14. SOAP clients SHOULD support SOAP Attachments.

§ 15. SOAP services SHOULD limit the use of SOAP Attachments to images, such as hotel and vehicle images.

### *SOAP Fault vs. OpenTravel Error*

SOAP 1.1 and 1.2 define the SOAP Fault element as a vehicle for transporting error information from a SOAP service to a SOAP client. OpenTravel also defines elements for returning errors and warnings to a SOAP client. While the SOAP Fault element is capable of indicating application-level errors, as well as errors that originate in the SOAP stack, this document recommends that the SOAP Fault element be used only to communicate and transport (SOAP stack) errors.

§ 16. OpenTravel SOAP services SHOULD use SOAP Fault for SOAP-level errors.

§ 17. OpenTravel SOAP services SHOULD use OpenTravel Errors for application-level errors.

§ 18. OpenTravel SOAP clients SHOULD support both SOAP Fault and OpenTravel Error handling.

§ 19. SOAP Fault response messages SHOULD conform to the SOAP 1.1 or SOAP 1.2 specifications, as clarified in WS-I basic profile section 4.1.

## Examples

Most of the following examples use the `OTA_ReadRQ` and `OTA_ReadRS` messages. The messages are relatively small and thus suitable for short and readable examples.

### OTA\_ReadRQ - Correct

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ota:OTA_ReadRQ xmlns:ota="http://www.opentravel.org/OTA/2003/05">
      <ota:POS>
        <ota:Source ISOCountry="CA" ISOCurrency="CAD" PseudoCityCode="AERO">
          <ota:RequestorID ID="APP.001" URL="http://www.acme.com"/>
        </ota:Source>
      </ota:POS>
      <ota:UniqueID ID="105789143" Type="21"/>
    </ota:OTA_ReadRQ>
  </soap:Body>
</soap:Envelope>
```

Figure: Correct `OTA_ReadRQ` Message

This example shows a correctly formatted SOAP request containing an `OTA_ReadRQ` message. The `OTA_ReadRQ` element is the only immediate child element of the SOAP Body element. This format allows the request to be issued to any Web service that implements the `OTA_ReadRQ` message as a non-RPC service.

### OTA\_ReadRQ - Incorrect - Escaped XML

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <acme:otaReadService xmlns:acme="http://www.acme.com/ota" Request="&lt;ota:OTA_ReadRQ
xmlns:ota=&quot;http://www.opentravel.org/OTA/2003/05&quot;&gt;&lt;ota:POS&gt;&lt;ota:Source
ISOCountry=&quot;CA&quot; ISOCurrency=&quot;CAD&quot;
PseudoCityCode=&quot;AERO&quot;&gt;&lt;ota:RequestorID ID=&quot;APP.001&quot;
URL=&quot;http://www.acme.com&quot;/&gt;&lt;/ota:Source&gt;&lt;/ota:POS&gt;&lt;ota:UniqueID
ID=&quot;105789143&quot; Type=&quot;21&quot;/&gt;&lt;/ota:OTA_ReadRQ&gt;"/>
  </soap:Body>
</soap:Envelope>
```

Figure: Incorrect `OTA_ReadRQ` Message - Escaped XML

This is an RPC-type SOAP request containing a call to an “otaReadService” defined by one particular Web Service. This OpenTravel request is serialized and passed to this service as a string. This SOAP message is tied to this particular service and is not in accordance with the guidelines in this document.

### OTA\_ReadRQ - Incorrect - Wrapped in Other XML

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <acme:otaReadService xmlns:acme="http://www.acme.com/ota">
      <ota:OTA_ReadRQ xmlns:ota="http://www.opentravel.org/OTA/2003/05">
        <ota:POS>
          <ota:Source ISOCountry="CA" ISOCurrency="CAD" PseudoCityCode="AERO">
            <ota:RequestorID ID="APP.001" URL="http://www.acme.com"/>
          </ota:Source>
        </ota:POS>
        <ota:UniqueID ID="105789143" Type="21"/>
      </ota:OTA_ReadRQ>
    </acme:otaReadService>
  </soap:Body>
</soap:Envelope>
```

Figure : Incorrect OTA\_ReadRQ Message - Wrapped in Other XML

This message is also an RPC-type SOAP request. Even though the OpenTravel request is not serialized it is still tied to a particular service, thus not in accordance with the guidelines in this document.

### OTA\_ProfileReadRS - Successful, Correct

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ota:OTA_ProfileReadRS xmlns:ota="http://www.opentravel.org/OTA/2003/05">
      <ota:Success/>
      <ota:Profiles>
        <ota:ProfileInfo>
          <ota:UniqueID ID="111240578" Type="21"/>
          ...
        </ota:ProfileInfo>
      </ota:Profiles>
    </ota:OTA_ProfileReadRS>
  </soap:Body>
</soap:Envelope>
```

Figure: Correct OTA\_ProfileReadRS Successful Response

The same guideline stating that the SOAP Body element should only contain a single OpenTravel message also applies to SOAP response messages. This message is correctly formatted, as the immediate child element of the SOAP Body is an OpenTravel response message root element.

### OTA\_ProfileReadRS - Unsuccessful, Correct (Application-Level Error)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ota:OTA_ProfileReadRS xmlns:ota="http://www.opentravel.org/OTA/2003/05">
      <ota:Errors>
        <ota:Error Code="282" Status="NotProcessed" Type="10"/>
      </ota:Errors>
    </ota:OTA_ProfileReadRS>
  </soap:Body>
</soap:Envelope>
```

*Figure: Correct OTA\_ProfileReadRS Unsuccessful Response (Application-Level Error)*

This message carries an application-level error back to the client application. Application level errors should be reported using OpenTravel error elements.

### OTA\_ProfileReadRS - Unsuccessful, Correct (SOAP-Level Error)

```
<?xml version="1.0" encoding="UTF-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode>soap:MustUnderstand</faultcode>
      <faultstring>SOAP Action URI missing</faultstring>
      <faultactor>http://www.acme.com/services/ota</faultactor>
      <detail>...</detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

*Figure: Correct OTA\_ProfileReadRS Unsuccessful Response (SOAP-Level Error)*

This response message carries a SOAP-level error back to the client. Such errors should not be generated by an OpenTravel server application, but can be generated by an XML intermediary or the OpenTravel application's SOAP stack/API.

## Sample SOAP Messaging WSDL for OpenTravel

```

<wsdl:definitions
  targetNamespace="http://www.acme.com/ota"
  xmlns:impl="http://www.acme.com/ota"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:wsdlsoap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:ota="http://www.opentravel.org/OTA/2003/05">

  <!-- Import of request and response schemas. -->
  <xsd:import schemaLocation="OTA_ReadRQ.xsd"/>
  <xsd:import schemaLocation="OTA_ProfileReadRS.xsd"/>

  <!-- No additional data types should be required. -->
  <wsdl:types/>

  <!-- Define request and response message. -->
  <wsdl:message name="OTA_ReadRQ">
    <wsdl:part name="OTA_ReadRQ" element="ota:OTA_ReadRQ"/>
  </wsdl:message>
  <wsdl:message name="OTA_ProfileReadRS">
    <wsdl:part name="OTA_ProfileReadRS" element="ota:OTA_ProfileReadRS"/>
  </wsdl:message>

  <!-- Define SOAP interface. -->
  <wsdl:portType name="SOAPClientInterface">
    <wsdl:operation name="OTA_Read">
      <wsdl:input message="impl:OTA_ReadRQ" name="OTA_ReadRQ"/>
      <wsdl:output message="impl:OTA_ProfileReadRS" name="OTA_ProfileReadRS"/>
    </wsdl:operation>
  </wsdl:portType>

  <!-- Define SOAP binding for SPM. -->
  <wsdl:binding name="spmSoapBinding" type="impl:SOAPClientInterface">
    <!-- Use document style and not rpc. -->
    <wsdlsoap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <!-- Define operation using defined messages. -->
    <wsdl:operation name="OTA_Read">
      <wsdlsoap:operation soapAction="ReadProfile"/>
      <!-- Use 'literal' to include OTA XML as-is. -->
      <wsdl:input name="OTA_ReadRQ">
        <wsdlsoap:body namespace="http://localhost/services/spm/spm" use="literal"/>
      </wsdl:input>
      <wsdl:output name="OTA_ProfileReadRS">
        <wsdlsoap:body namespace="http://localhost/services/spm/spm" use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>

  <!-- Define SOAP interface with previously declared binding. -->
  <wsdl:service name="SOAPClientInterfaceService">
    <wsdl:port binding="impl:spmSoapBinding" name="spm">
      <wsdlsoap:address location="http://www.acme.com/services/ota"/>
    </wsdl:port>
  </wsdl:service>

</wsdl:definitions>

```

Figure: Sample WSDL Definition for Simple OpenTravel Service

This outlines how an OpenTravel service can be described in a WSDL document. The only data-types used in this document are types that are already defined in OpenTravel schemas. The <binding/> element declares the service as using the document/literal style to pass OpenTravel-defined messages as-is between the OpenTravel client and service.

## SOAP with Attachments Sample

```

Accept-Language: en
Content-Type: multipart/related; type="text/xml"; boundary="-----Multipart_Boundary_-----"
Content-Id: soap-envelope
Content-Length: 8967

-----Multipart_Boundary_-----
Content-Type: text/xml
Content-Id: soap-envelope

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <OTA_HotelAvailRS xmlns="http://www.opentravel.org/OTA/2003/05">
      <Success/>
      <RoomStays>
        <RoomStay xmlns="http://www.opentravel.org/OTA/2003/05">
          ...
          <BasicPropertyInfo BrandCode="DE" ChainCode="DE" ...>
            <VendorMessages>
              <VendorMessage>
                <SubSection SubTitle="HotelImage">
                  <Paragraph>
                    <URL>cid:HotelPool.jpg</URL>
                  </Paragraph>
                </SubSection>
              </VendorMessage>
              ...
            </VendorMessages>
            ...
          </BasicPropertyInfo>
        </RoomStay>
      </RoomStays>
    </OTA_HotelAvailRS>
  </soap:Body>
</soap:Envelope>
-----Multipart_Boundary_-----
Content-Type: image/jpeg
Content-Id: HotelPool.jpg

...binary image data goes here...
-----Multipart_Boundary_-----

```

*Figure: OpenTravel Message with Hotel Image Attachment*

The above example shows how a binary attachment can be returned with an OpenTravel response message. The SOAP Envelope containing the OpenTravel response is the main part of the response message. The OpenTravel message references an attachment using the 'cid:' URL prefix.

## WS-Security Token Sample

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <wsse:Security soap:actor="acme-ota-proxy" xmlns:wsse="...">
      <wsse:UsernameToken wsu:Id="morten" xmlns:wsu="...">
        <wsse:Username>morten</wsse:Username>
        <wsse:Nonce EncodingType="UTF-8">
          7h9x9UYevqMgBUzT2CXAK2oVYYU2J9PeEG0ZQniz8Tk=
        </wsse:Nonce>
        <wsse:Password Type="wsse:PasswordDigest">
          3gDnTTVAdl9NqSVA0c9sllmNJrM=
        </wsse:Password>
        <wsu:Created>2006-01-17T17:55:14Z</wsu:Created>
      </wsse:UsernameToken>
    </wsse:Security>
  </soap:Header>

  <soap:Body>
    <ota:OTA_ReadRQ xmlns:ota="http://www.opentravel.org/OTA/2003/05">
      <ota:POS>
        <ota:Source ISOCountry="CA" ISOCurrency="CAD" PseudoCityCode="AERO">
          <ota:RequestorID ID="APP.001" URL="http://www.acme.com"/>
        </ota:Source>
      </ota:POS>
      <ota:UniqueID ID="105789143" Type="21"/>
    </ota:OTA_ReadRQ>
  </soap:Body>
</soap:Envelope>
```

*Figure: OpenTravel SOAP Message with WS-Security Token*

This sample OpenTravel request contains a WS-Security username/password token. The SOAP Body contains only the OpenTravel request payload, while the security token is placed in the SOAP Header. The security token is typically processed by a SOAP proxy, such as an XML Security Gateway, while the SOAP Body content is processed by an OpenTravel Web service. The security token and OpenTravel request are kept separate, which helps disconnect the OpenTravel Web service from the security layer.

This example illustrates how standard WS-Security compliant implementations may be leveraged by OpenTravel Message producers and consumers. As with all referenced specifications, the goal is to leverage these industry standards. Please refer to the [WS-Security specification](#) for further details on this authentication method.

## XML-Signature Sample

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>
    <wsse:Security soap:actor="acme-ota-proxy" xmlns:wsse="...">
      <dsig:Signature Id="MortenJorgensen" xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
        <dsig:SignedInfo>
          <dsig:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
          <dsig:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
          <dsig:Reference URI="">
            <dsig:Transforms>
              <dsig:Transform Algorithm="http://www.w3.org/TR/1999/REC-xpath-19991116">
                <dsig:XPath xmlns:ota="http://www.opentravel.org/OTA/2003/05">
                  ancestor-or-self::ota:UniqueID
                </dsig:XPath>
              </dsig:Transform>
              <dsig:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
            </dsig:Transforms>
            <dsig:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
            <dsig:DigestValue>C33iD3LMNL0/4aSeY+DdzH76Cng=</dsig:DigestValue>
          </dsig:Reference>
        </dsig:SignedInfo>
        <dsig:SignatureValue>
          nIYgIKl7D0pUKBGPnN36g/vn1Nu38fRJ0uooKktbpFWsPQj8A75AXAJ04TYrssgrkJ0WZOFnY8h0
          d4PDAJcbgRjFusZYRh1Q3MWEMJ/9GFnLyglNTcveskWNweuUgyz56ARHHnb6MUvEzykMV4So6zaX
          6I8I130T3/r0NaD0DLE=
        </dsig:SignatureValue>
        <dsig:KeyInfo>
          <dsig:KeyName>CN=Morten Jorgensen,O=OpenJaw,...,C=IE</dsig:KeyName>
        </dsig:KeyInfo>
      </dsig:Signature>
    </wsse:Security>
  </soap:Header>

  <soap:Body>
    <ota:OTA_ReadRQ xmlns:ota="http://www.opentravel.org/OTA/2003/05">
      <ota:POS>
        <ota:Source ISOCountry="CA" ISOCurrency="CAD" PseudoCityCode="AERO">
          <ota:RequestorID ID="APP.001" URL="http://www.acme.com" />
        </ota:Source>
      </ota:POS>
    </ota:OTA_ReadRQ>
  </soap:Body>
</soap:Envelope>
```

Figure: Signed OpenTravel SOAP Message

This sample OpenTravel request has an XML Signature element, which contains a digital signature over the OpenTravel payload. Again, the SOAP Body contains only the OpenTravel request payload, while the security token is placed in the SOAP Header. The security token is typically processed by a SOAP proxy, such as an XML Security Gateway, while the SOAP Body content is processed by an OpenTravel Web service. The security token and OpenTravel request are kept separate, which helps disconnect the OpenTravel Web service from the security layer. Please refer to the [XML-Signature](#) and [WS-Security](#) specifications for further details.

## XML-Encryption Sample, Correct

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">

  <soap:Header>
    <wsse:Security soap:actor="acme-ota-proxy" xmlns:wsse="...">
      <enc:EncryptedKey xmlns:enc="..." Encoding="utf-8" MimeType="text/xml">
        <enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5">
          <enc:KeySize>256</enc:KeySize>
        </enc:EncryptionMethod>
        <dsig:KeyInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
          <dsig:KeyName> CN=Morten Jorgensen,O=OpenJaw,...,C=IE</dsig:KeyName>
        </dsig:KeyInfo>
        <enc:CipherData>
          <enc:CipherValue>
            I6kwMqpERPpbYoKa2lc/7g4kOuT/ntbpkXbRTX0VQFs6NYOcGItuvpB9qrCe4XKb
            FKsePNOMQmUQQjSszvIfzhLDst01NZaHCVRZ8FBAGiAWTH06ZPZEDc10WObcy+Y6
            hWJ4Gd0XpDTmaUc3pReil8iWZOYz9no6PjyfJDThIQI=
          </enc:CipherValue>
        </enc:CipherData>
        <enc:CarriedKeyName>session-key</enc:CarriedKeyName>
      </enc:EncryptedKey>
    </wsse:Security>
  </soap:Header>

  <soap:Body>
    <ota:OTA_ReadRQ xmlns:ota="http://www.opentravel.org/OTA/2003/05">
      <ota:POS>
        <ota:Source ISOCountry="CA" ISOCurrency="CAD" PseudoCityCode="AERO">
          <ota:RequestorID ID="APP.001" URL="http://www.acme.com"/>
        </ota:Source>
      </ota:POS>
      <enc:EncryptedData xmlns:enc="..." Encoding="utf-8" MimeType="text/xml"
        Type="http://www.w3.org/2001/04/xmlenc#Element">
        <enc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc">
          <enc:KeySize>256</enc:KeySize>
        </enc:EncryptionMethod>
        <dsig:KeyInfo xmlns:dsig="http://www.w3.org/2000/09/xmldsig#">
          <dsig:KeyName>session-key</dsig:KeyName>
        </dsig:KeyInfo>
        <enc:CipherData>
          <enc:CipherValue>
            fcU3IOeBS/wPKxJgPimX/apgGMnfKjF/hYOceAVLz6avzEn97xV/ipB1MLDtilef
            q4NJWQiwJyqWzXjCQFVEp+OTFav29m6NMc336psfZbpBXLXjNTGBtslVYZ0G214P
            FKLfua4nljlr5ve8vxn/AA==
          </enc:CipherValue>
        </enc:CipherData>
      </enc:EncryptedData>
    </ota:OTA_ReadRQ>
  </soap:Body>
</soap:Envelope>
```

Figure: Encrypted OpenTravel Sample SOAP Message

XML-Encryption tokens are the only type of security token that is inserted in-place of the SOAP payload (in this case an OpenTravel Request). XML-Encryption tokens are typically used when the message travels across one or more SOAP intermediaries or is processed by multiple Web service that need access to only certain parts of the request. Exchange of encrypted XML normally follows these steps:

A SOAP client typically creates the request as clear-text XML. The sensitive portions of the request are encrypted either by the SOAP client itself or by a proxy that resides inside the local network of the SOAP client. The SOAP

request is encrypted using the public key of the Web service it is intended for. The SOAP request can then safely travel across a public network.

The SOAP request is typically decrypted by an XML Security Gateway before it is forwarded to the Web service. The XML Security Gateway decrypts the SOAP request using the public key of the Web service.

The key point to note is that both the client and service handle OpenTravel messages in clear text, while the encryption proxy and security gateway handle the encryption and decryption processes. This allows the client and server to be implemented as if SOAP messages are transmitted in clear text.

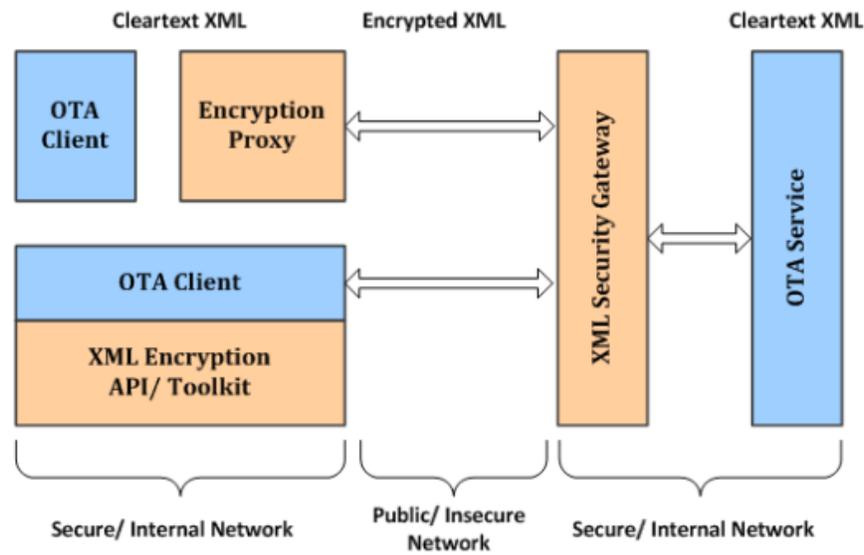


Figure: XML Encryption Scenario

Please refer to the [XML-Encryption specification](#) for further details.

## HTTP Messaging

### *Background*

#### **2001C OpenTravel Infrastructure Guidelines**

The 2001C OpenTravel Infrastructure guidelines were added to the OpenTravel specifications in the 2001C release. These guidelines included design goals, XML best practices, the concept of using flexible request/response message pairs, and the use of ebXML as a transport protocol for passing these messages. The specification of the transport protocol was not formally part of the OpenTravel specification because OpenTravel members wished to maintain flexibility with regard to which transport protocol they used.

#### **Design Goals of OpenTravel (2001C)**

In 2001, the Design Goals were stated as follows:

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

- *Openness*- OpenTravel specifications are publicly available to all organizations seeking to develop new or enhanced systems. Membership to OpenTravel is open to all organizations interested in developing these specifications.
- *Flexibility*- OpenTravel specifications provides 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*- OpenTravel has developed this specification to work with any equipment or software that can support the common standards used in the specification.
- *Security*- OpenTravel 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*- OpenTravel plans to add more services and functionality to this specification in a way that minimizes incompatibility for those implementing this or other early versions. OpenTravel work groups that develop the specifications do so with future transitions in mind.
- *International scope*- The initial specification was written in English; however, OpenTravel intends to extend later versions to provide representation in character sets supporting the Unicode standard. When possible, OpenTravel has designed data elements to meet as many global elements as possible.

### *The Need for Interoperability*

#### **Emerging Trends**

Since 2001, many new members have joined OpenTravel and want to know how to build systems that interoperate with other vendors who have implemented OpenTravel messages. Within the travel industry, many more integrators and intermediaries have emerged that wish to work with many partners. Decreasingly, the emphasis is on enabling implementation of interactions between individual pairs of trading partners and increasingly on creating services that interoperate with all interested trading partners.

Furthermore, recent OpenTravel message registrations have indicated that a simple transport protocol based on an HTTP POST is increasingly popular.

These two trends suggest that a new specification is appropriate. This section describes this specification. It provides the guidance necessary to create interoperable applications based on a reference transport protocol.

It should not be construed that companies are any less OpenTravel conformant if they choose a different transport protocol. This specification serves simply as guidance for companies wishing to make OpenTravel conformant applications that are highly interoperable.

### **New Design Goals**

This HTTP transport protocol reference section addresses the following design goals in response to emerging trends:

- *Interoperable*- Companies developing OpenTravel conformant systems should have sufficient guidance to build systems which are highly interoperable with systems built by other companies using the same OpenTravel messages for the same use cases.
- *Simple*- To be able to reach the most partners, operating from the most platforms, the protocol would do well to be as simple as possible. This also reduces the hurdles for new members to implement OpenTravel messages in their systems.
- *Optimal Performance*- The OpenTravel HTTP transport protocol should not impede performance by burdening the processing with unnecessary overhead.
- *Readily Accepted*- The OpenTravel HTTP transport protocol should reflect what member companies are already doing, thus assuring broad acceptance of the specification.

The following additional design goal is desirable, but is not directly addressed by this document:

- *Certifiable*- Companies should have some means whereby they can certify (for their own benefit) that they have done what it takes to give their system a good chance of inter-operating with other companies' systems.

### **Requirements for Interoperability**

True interoperability is a lofty goal. A reference transport specification is necessary but not sufficient to achieve it. The following are identified as important to achieving true interoperability:

- *Standard Usage Profiles*- OpenTravel messages are very flexible. However, if a set of systems are to be interoperable with each other, they must use the selected OpenTravel messages in the same way for a specific use case. OpenTravel plans to support the definition by member companies of Usage Profiles as examples of OpenTravel messages, restricted to support a specific use case. Interoperability will require that a set of Usage Profiles emerge that are generally accepted.
- *Standard Transport Protocol*- This document describes the OpenTravel Reference Transport Protocol. For purposes of interoperability, it may serve as the standard.
- *Standard Software Validation Suite*- To have any confidence that a system will interoperate with other systems, it must be exercised by a test suite to certify its features comply with the standards. The production of software is currently beyond the scope of efforts sponsored by OpenTravel.

## **OpenTravel Transport Protocol Reference: HTTP**

### **Philosophy of Interoperability**

For maximum interoperability, your system SHOULD be liberal in what it accepts and strict in what it emits.

### **Simple HTTP POST vs. ebXML**

The prior recommendations regarding ebXML are still available for companies wishing to implement that transport protocol. However, OpenTravel acknowledges that adoption of ebXML among OpenTravel member companies has been limited and that the following alternate protocol based on a simple HTTP POST represents a simpler method to create interoperable systems.

Other transport protocol references may be specified in future OpenTravel releases. Such specifications may include (but are not limited to) SOAP (Simple Object Access Protocol).

## **Standard HTTP**

Nothing in this specification should be understood to override or alter the standard definition of the HTTP protocol. The HTTP/1.0 and HTTP/1.1 versions of the HTTP protocol are described in the following documents:

HTTP/1.0 – <http://www.ietf.org/rfc/rfc1945.txt>

HTTP/1.1 – <http://www.ietf.org/rfc/rfc2616.txt>

However, the design of HTTP accommodates extensions through the use of custom headers. Custom headers that are defined for OpenTravel transactions are described below.

## HTTP Message Content

The OpenTravel transport protocol reference is based on a simple HTTP POST transaction. An HTTP POST transaction allows for an arbitrarily large amount of “content” to be sent with both the request and with the response. An OpenTravel request message is sent as the “content” in an HTTP POST request, and an OpenTravel Response message is received in response.

## HTTP Message Headers

In addition to the “content”, an HTTP message (request or response) also carries HTTP “headers.” The goal of the following description of headers is to describe maximum simplicity and interoperability using HTTP. If a system developer uses any standard HTTP-handling libraries or components, these details should be handled automatically for them.

Both clients and servers **MUST** support HTTP/1.0. They **MAY** support HTTP/1.1, but if they do, they **MUST** support the client/server negotiation that allows the server to downgrade to HTTP/1.0 if the client cannot support HTTP/1.1.

Different headers are allowed (or required) based on whether referring to the client (requestor) or server (responder) and whether using HTTP/1.0 or HTTP/1.1.

The only HTTP header **REQUIRED** for both request and response is the “Content-length” header. This allows systems that implement the HTTP protocol themselves to be written simply. If HTTP/1.1 is used, the request requires the “Host” header (a requirement of HTTP/1.1 itself). Other headers may be useful, but they are not required, and they may be safely ignored.

Header	Description	Example
Content-length	[REQUIRED] Tells how many bytes are in the “content” (after the “headers”)	Content-length: 1558
Host	[REQUIRED for HTTP/1.1 Client according to HTTP/1.1 spec] This allows a server which is hosting multiple virtual domains to know which domain the request was for. If the server is not hosting multiple virtual domains, the server <b>MAY</b> ignore this header.	Host: www.travelco.com
Connection	[OPTIONAL] Systems <b>SHOULD</b> utilize this header, but they <b>MUST</b> be robust when communicating with another server that does not recognize it. Using “Close” means that the network socket will be closed after the exchange of a single message (headers and content). “Keep-alive” means that the network socket will be kept open for a short time in case additional transactions are to be sent on the same socket. “Close” is the default for HTTP/1.0, and “Keep-alive” is the default for HTTP/1.1.	Connection: Close Connection: Keep-alive
Accept-charset	[OPTIONAL] This tells a server what character sets the client can handle. The server <b>MAY</b> ignore this if it will only be responding with ASCII text. However, if it might be responding with any non-ASCII text, it <b>MUST</b> encode the data in accordance with one of the acceptable character sets. “utf-8” is the preferred character set because XML is specified to support Unicode, and “utf-8” is widely supported (and is ASCII-compatible). If the header is supplied by the client, it <b>MUST</b> include “utf-8”. If the header is not supplied, the server should assume “utf-8”. In this way, both client and server should support “utf-8” and the content in both request and response can be un-	Accept-charset: utf-8

Header	Description	Example
	derstood to be “utf-8”.	
Accept	[OPTIONAL] This tells a server what Content-types the client can handle. The server SHOULD ignore this, because it is going to respond with an XML OpenTravel response message regardless of what the client says he can accept. The client MAY produce this header, but if so, it SHOULD include at least one of the “xml” content types.	Accept: text/xml Accept: application/xml Accept: application/xhtml+xml Accept: text/html, text/plain
Accept-language	[OPTIONAL] This tells a server what languages the client can handle.	Accept-language: en-us, en
Accept-encoding	[OPTIONAL] This tells a server whether the response content can be compressed or not. Clients and servers SHOULD implement a compressed encoding, but they MUST support the standard HTTP negotiation that allows them to interoperate with systems that do not implement it.	Accept-encoding: gzip, deflate
[other]	[OPTIONAL] Clients and servers MAY provide other headers and they MAY be ignored by the other system.	n/a

### OpenTravel Custom Header: OTA-Echo-Token

For performance reasons, in order to accommodate massive exchanges of OpenTravel messages between two systems in a high-volume B2B relationship, an optional HTTP Header, “OTA-Echo-Token”, may be used.

Without any extension to the HTTP protocol, the requesting system would send a request with a Keep-Alive header and wait for a response. Each request would have to wait for a response. This is not high enough performance for these applications.

When the “OTA-Echo-Token” header is used, the token value MUST be unique to that message and not shared by any other message sent by that system. The sending system does not necessarily expect an immediate response, but it does expect that whenever the response message is received, it will be accompanied by the same token value in an “OTA-Echo-Token” header.

Using the “OTA-Echo-Token”, it is possible to multiplex several OpenTravel message transactions asynchronously over the same TCP/IP connection. The token value correlates the request and the response.

Header	Description	Example
OTA-Echo-Token	[OPTIONAL] A token sent with a request that is expected to be echoed in a response. This is a string token with a maximum size of 128 characters.	OTA-Echo-Token: 8335013 OTA-Echo-Token: A154/125/001

A requesting system that uses this header may only be attached to a responding system that uses the header, or it will not receive the information necessary for it to function. A responding system that understands this header SHOULD be able to reply synchronously to clients that do not use the header.

### Encryption

- Encrypted communication between systems is accomplished with SSL (i.e., HTTPS).
- If encryption is required, a system MUST support HTTPS.

- Systems MAY communicate via unencrypted HTTP for transactions that do not need to be secure. HTTP is also useful for transactions that are executed across a VPN or other networking channel that is already secure for other reasons.

### **Authentication**

- Authentication of a client (requestor) to a server (responder) is achieved via HTTP Basic Authentication.
- Authentication of a server (responder) to a client (requestor) is achieved via SSL.
- If authentication is required, a system MUST support HTTP basic authentication.
- Systems MAY communicate without authentication for transactions that do not require it.

### **Other Features: Logging**

Organizations offering OpenTravel transactions MAY provide logging capability, regardless of the type of transaction in the business message (e.g., travel verbs, infrastructure verbs), and SHOULD provide logging capability for all transactions that involve implicit or explicit contracts or promises to pay. Trading partners MAY exchange event logs to provide audit trails.

# GENERIC MESSAGES

## Messages Overview

Unlike OpenTravel messages that are specific to a particular travel sub-domain (e.g., Air) other messages are generally applicable and may be used more broadly than in one domain-specific service.

This section briefly describes the generic messages that can be used across multiple travel sub-domains.

There are several OpenTravel Generic Messages that provide a wide range of functionality, including:

- [Payment validation and authorization \(including bank and credit card\)](#)
- [Reservation cancellations \(full and partial\)](#)
- [Customer profile deletions](#)
- [Sending file attachments for storage and processing by a receiving system](#)
- [Sending inventory and rates notifications](#)
- [Testing your application connectivity](#)
- [Purchasing gift certificates and other merchandise](#)
- [Retrieving reservations \(by booking id, by passenger name, etc.\)](#)
- [Requesting information in a free text type of response using a terminal message input](#)

## Messages List

- [OTA\\_AuthorizationRQ/RS](#)
- [OTA\\_CancelRQ/RS](#)
- [OTA\\_DeleteRQ/RS](#)
- [OTA\\_ErrorRQ/RS](#)
- [OTA\\_FileAttachmentNotifRQ/RS](#)
- [OTA\\_NotifReportRQ/RS](#)
- [OTA\\_PingRQ/RS](#)
- [OTA\\_PurchaseItemRQ/RS](#)
- [OTA\\_ReadRQ/OTA\\_ResRetrieveRS](#)
- [OTA\\_ScreenTextRQ/RS](#)

## OTA\_AuthorizationRQ/RS

### *Message Pair Overview*

The OpenTravel Authorization message pair verifies and validates a traveler based on information for a bank account, credit card, or driver's license. Each of these can be validated with or without an address.

The Authorization message could be used in any of the following circumstances:

- By a travel agent verifying that a customer bank account and/or credit card is valid for payment
- By a travel agent verifying that a customer bank account is valid for payment and their identification is valid
- By an online travel website verifying that the customer credit card is valid and can be used as a form of payment for the amount requested (and that the billing address for the account is accurate)
- By a travel agent verifying that the customer bank account is valid for payment on behalf of another company or party
- By an online travel website to send multiple (bundled) booking authorizations' in a single request

When bundling multiple authorizations in a single request, each Authorizations element in the response must be inspected to determine the result.

## OTA\_CancelRQ/RS

### *Message Pair Overview*

The OpenTravel Cancel message pair is used for canceling a previously made (booked) reservation. This message may be used to cancel the entire reservation or to cancel specific segments (portions) of the reservation.

A reservation ID (PNR locator, confirmation number, etc) is sent to specify which reservation should be canceled. Additionally, a supplier may require that verification information, such as customer name, credit card number, and/or phone number, be sent in order to verify that the correct reservation is being canceled.

When only specific segments in the reservation are to be canceled, either a travel segment type (e.g. air, hotel, cruise) or a segment number(s) is sent to indicate which segment(s) should be canceled.

The use of the OPTIONAL <POS> element allows an implementation to determine whether the remote user has permission to cancel the reservation.

In the Cancel Request one of the following actions must be specified:

- Initiate—indicates the initial request to cancel a reservation.
- Ignore—indicates a rollback of the request to cancel, leaving the reservation intact.
- Confirm—indicates a request to complete the cancellation.

The Cancel Response will return one of the following statuses:

- Pending—indicates the initial request to cancel a reservation is pending confirmation to complete the cancel action. Cancel rules may have been returned in the response.
- Ignored—indicates the request to cancel was rolled back, leaving the reservation intact.
- Cancelled—indicates the cancellation is complete. A cancellation ID may be returned along with the response.

The Cancel messages also provide the option for a two-step process and individual business rules that may determine how a system processes the initial cancel request. If there are no penalties involved in the cancellation,

the cancel transaction can take place and the response returns the cancellation number along with the status that the reservation has been canceled.

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 party would then resend the `OTA_CancelRQ`.

A *CancelType="Ignore"* would expect a response with the *Status "Ignored"*, thus ending the message conversation with no action being taken to cancel the reservation.

A *CancelType="Commit"* indicates a definitive instruction to process the cancellation. This message would expect the response of *Status="Cancelled"*, along with the return of a Cancellation ID, which would complete the cancellation process. The `CancelRQ` is the same message in each case, with the `CancelType` attribute indicating the action to be taken on the request.

## OTA\_DeleteRQ/RS

### *Message Pair Overview*

The OpenTravel Delete message pair uses an “action” attribute to define an operation that identifies an existing record, and removes the entire record from the database. The use of the Delete action depends upon the business rules of an organization. Alternative strategies, such as mapping a duplicate record to another by use of the UniqueID, may be considered.

The requester MAY also verify the record before deleting it to ensure the correct record has been identified prior to deleting it. In this case, the use of the Instance attribute may be useful in determining whether the record has been updated more recently than the information that is intended to be deleted. That choice, again, would be dictated by good business practices.

Steps in the Delete operation may include:

- Requester submits a Read request to view the record
- Responder returns the record for the requester to view
- Requester submits a Delete request
- Responder removes the record and returns an acknowledgment

The use of the OPTIONAL `<POS>` element allows an implementation to determine whether the remote user has permission to delete the object being read.

## OTA\_ErrorRS

### *Message Pair Overview*

In general, a system message request can fail in a number of common ways. For example the API key or user name authentication may be invalid. The location or product being accessed may not belong to the operator making the request and so on. Where appropriate, the standard OTA error response can be used.

The OpenTravel Error Response message is designed to accommodate errors that result from the parser, or from validation, before reaching the server. The set of errors that can use this error is constrained by its limited structure, to provide structural stability. This message can be used when an OpenTravel message is not processed, which may occur during a failed authentication, a failed xml parsing , or if a required field is missing and will be wrapped by the message type sent.

The `ErrorCode` types shown below in italics are currently supported by OpenTravel, while the others are reserved by the OpenTravel Specification but not implemented:

1. *An unknown error.*

2. The target business system has no implementation for the intended request.
3. The XML message has passed a low-level validation check, but that the business rules for the request message were not met.
4. The message failed authentication.
5. The security credentials in the message have expired.
6. The message failed authorization.
7. A request was sent within a message exchange that does not align to the message.
8. The target business system does not support the intended transaction-oriented operation.
9. The type of authentication requested is not recognized.
10. An element or attribute that is required in by the schema (or required by agreement between trading partners) is missing from the message.

## OTA\_FileAttachmentNotifRQ/RS

### *Message Pair Overview*

The OpenTravel File Attachment Notification message pair enables a sending system to physically transfer encoded binary files to a receiving system that will store or process them.

The request message and the encoded files are transferred together in a single encoded entity, with the request message being the first item retrieved by the receiving system. The receiving system uses the request message as a table of contents with the descriptive information it needs to decode and store the files.

The response message acknowledges the received files and returns any warnings or errors as needed.

## OTA\_NotifReportRQ/RS

### *Message Pair Overview*

The OpenTravel Notification Report message pair sends a report on whether asynchronous database updates have been processed successfully or not. This message pair currently supports hotel and vehicle but may be extended in the future to accommodate other travel sector database reports.

The responses to the database update (e.g., OTA\_HotelAvailNotifRS, OTA\_HotelRateAmountNotifRS or OTA\_VehLocDetailsNotifRS) are only used to acknowledge the fact that the message has been received (because the completion of the update in the application might take a few minutes). Moreover, in case of error, those schema do not provide the ability to point out and give details on the part that failed and this scenario is when this message pair is used.

The business flow for hotel availability and rate amount database updates is as follows:

1. A hotel provider sends a OTA\_HotelAvailNotifRQ to a booking source.
2. The booking source acknowledge the fact that the message has been received (but not yet processed) by replying with the OTA\_HotelAvailNotifRS.
3. The booking source system process the database update.
4. Once the database update is completed the booking source system may need to inform the hotel provider that updates were successful or not.
5. The booking source builds an OTA\_NotifReportRQ message and sends it to the Hotel Provider.
6. The Hotel provider acknowledges it with an OTA\_NotifReportRS message.

The business flow for vehicle location database updates is as follows:

1. A vehicle provider sends a [OTA\\_VehLocDetailsNotifRQ](#) to a recipient (e.g. a trading partner system.)
2. The recipient acknowledges the fact that the message has been received (but not yet processed) by replying with the [OTA\\_VehLocDetailsNotifRS](#).
3. The recipient source system process the database update.
4. Once the database update is completed the recipient source system may need to inform the vehicle provider that updates were successful or not.
5. The recipient source builds an OTA\_NotifReportRQ message and sends it to the vehicle provider.
6. The vehicle provider acknowledges it with an OTA\_NotifReportRS message.

## OTA\_PingRQ/RS

### *Message Pair Overview*

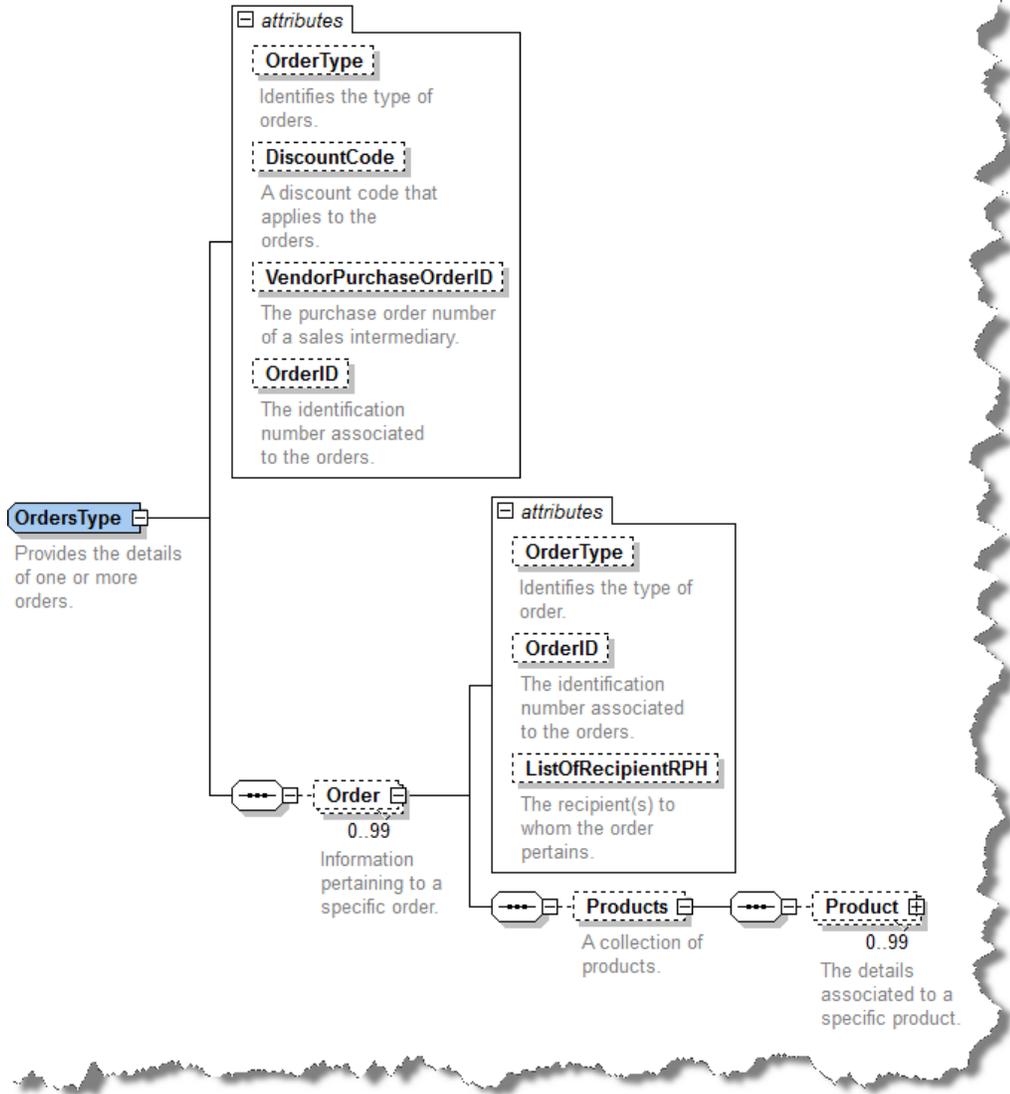
The OpenTravel Ping message pair are used to test application connectivity by sending some specific text and determining if the receiving application is able to echo back that same text. The free-text data that is passed in the request is expected to be echoed back in the response message.

## OTA\_PurchaseItemRQ/RS

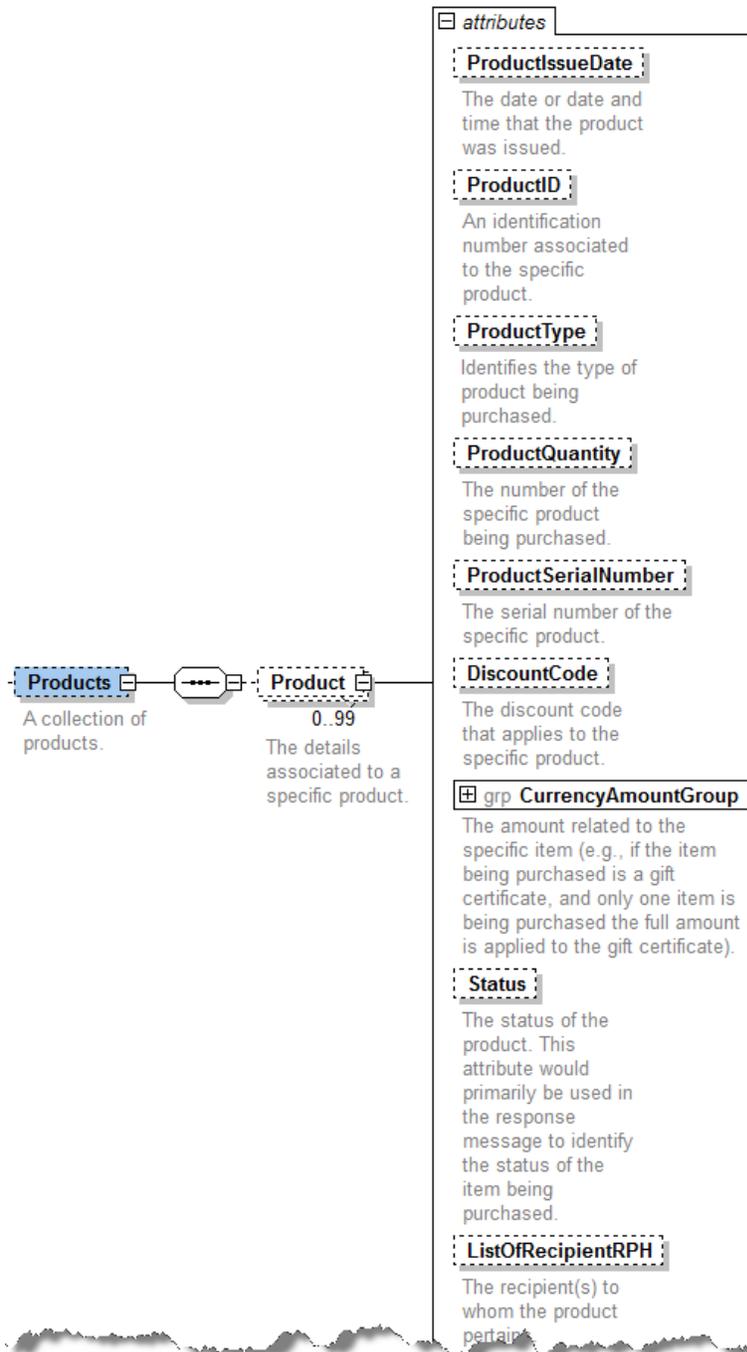
### *Message Pair Overview*

The OpenTravel Purchase Item message pair provides functionality for electronic purchases of non-inventory items (e.g. gift certificates for a hotel, drink tickets on an airline, etc.) The message pair supports the ordering process with detailed product, recipient, purchaser and payment information.

Key to this message pair are the OpenTravel *OrdersType* and *Product* complex elements defined in *OTA\_CommonTypes.xsd*. The *OrdersType* complex element contains the details of one or more repeating order(s) as shown in the diagram below:



The repeating Product complex element contains the details of one or more products that are associated with the order:



## OTA\_ReadRQ/OTA\_ResRetrieveRS

### *Message Pair Overview*

The OpenTravel Read infrastructure action defines an operation that opens an existing record and transmits information contained in that record. The Read operation enables the user to identify a particular record and retrieve its entire contents, to request a reservation when the booking id number is not known, to request a list of reservations for a traveler, and to request a list of travelers that meet specified criteria.

In the Read request, it can be specified whether the reservation itself should be returned if an exact match is found or if a list of reservations should be always be returned.

The basic operation has the following steps:

- Requester queries the database where the record resides by sending a Read request message with the record's unique identifier or with information such as traveler name, date of travel, loyalty number and/or other specified criteria.
- Responder returns the record or a list of reservations to the requester.

The response to the OTA\_ReadRQ will vary based on the type of record(s) requested and you and your trading partners' system capabilities. The following are examples of the type of records requested and the possible message response:

Request Type	Possible Response Messages
Airline Reservation	<a href="#">OTA_AirBookRS</a> OTA_ResRetrieveRS
Cruise Reservation	OTA_ResRetrieveRS
Golf Course Reservation	<a href="#">OTA_GolfCourseResRS</a> OTA_ResRetrieveRS
Global Distribution System Reservation	OTA_ResRetrieveRS
Hotel Reservation	OTA_HotelResRS OTA_ResRetrieveRS
Insurance Plan Booking	<a href="#">OTA_InsuranceBookRS</a>
Loyalty Account	<a href="#">OTA_LoyaltyAccountRS</a>
Package Tour/Holiday Booking Reservation	<a href="#">OTA_PkgBookRS</a> OTA_ResRetrieveRS
Profile	<a href="#">OTA_ProfileReadRS</a>
Tour Reservation	OTA_ResRetrieveRS

If an exact match of the reservation is not found or a list is requested, the OTA\_ResRetrieveRS message is returned with a list of reservations with enough additional information to help the requester determine which res-

ervation they wish to see. The requester may then send another Read request with the reservation identifier to retrieve a specific reservation.

## OTA\_ReviewsNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Reviews Notif message set is a push message used to push reviews based on specified criteria. Reviews are evaluations that are completed by guests after they use a product or service. The receiving system responds with an acknowledgement that the message was successfully received, or includes warnings from business processing rules or errors if the request did not succeed. This message is currently designed to handle hotel reviews, but can be easily updated to handle reviews from other travel sectors when needed.

## OTA\_ReviewsRQ/RS

### *Message Pair Overview*

The OpenTravel Reviews message set is a pull message used to request reviews that may be filtered based on criteria that is identified in the message. Reviews are evaluations that are completed by guests after they use a product or service. The response is returned containing the reviews that match the specified criteria. This message is currently designed to handle hotel reviews, but can be easily updated to handle reviews from other travel sectors when needed.

## OTA\_ScreenTextRQ/RS

### *Message Pair Overview*

The OpenTravel Screen Text message pair is used to request information in a free text type of response using a terminal message input. It allows users who do not have fully developed XML capabilities to send and receive XML messages and/or to request information for which there is no OpenTravel message functionality developed.

# AIR MESSAGES

## Messages Overview

The OpenTravel Air Messages provide a wide range of functionality, including:

- [Requesting flight availability for a city pair on a specific date for a specific number and type of passengers \(optionally constrained by availability for a specific airline, flight or booking class on a flight, all for a specific date\)](#)
- [Booking \(making a reservation\) for a specific itinerary for one or more identified passengers \(including optional pricing information\) while allowing the booking class availability and pricing to be rechecked as part of the booking process](#)
- [Flight check in from a self-service channel \(kiosks, web and mobile\)](#)
- [Requesting ticket fulfillment](#)
- [Displaying booking files on a queue](#)
- [Requesting information on fares between city pairs for a particular date or date range with no Inventory check for available seats](#)
- [Requesting updated information on the operation of a specific airline flight](#)
- [Requesting flight leg and codeshare information for a specific flight on a specific date between a city pair](#)
- [Requesting lowest fare priced itinerary options for flights between specific city pairs on certain dates for specific numbers and types of passengers](#)
- [Requesting pricing information for specific flights on certain dates for a specific number and type of passengers \(including optional information such as fare restriction preferences and negotiated fare contract codes\)](#)
- [Requesting text rules for a specific fare basis code for an airline and city pair on a specific date \(including optional information such as negotiated fare contract codes\)](#)
- [Viewing airline flight schedules](#)
- [Displaying which seats are available for a given flight, as well as their location within the aircraft](#)
- [Modifying an existing booking file \(reservation\)](#)

## Messages List

- [OTA\\_AirAvailRQ/RS](#)
- [OTA\\_AirBaggageRQ/RS](#)
- [OTA\\_AirBookRQ/RS](#)
- [OTA\\_AirCheckInRQ/RS](#)
- [OTA\\_AirDemandTicketRQ/RS](#)
- [OTA\\_AirDisplayQueueRQ/RS](#)
- [OTA\\_AirFareDisplayRQ/RS](#)
- [OTA\\_AirFlifoRQ/RS](#)
- [OTA\\_AirDetailsRQ/RS](#)
- [OTA\\_AirLowFareSearchRQ/RS](#)
- [OTA\\_AirPriceRQ/RS](#)

- [OTA AirRulesRQ/RS](#)
- [OTA AirSchedulesRQ/RS](#)
- [OTA AirSeatMapRQ/RS](#)
- [OTA AirBookModifyRQ/OTA AirBookRS](#)
- [OTA AirGetOfferRQ/RS](#)

## OTA\_AirAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Air Availability message pair contains the structure and elements of requests and responses for airline flight availability and point of sale information. The Air Availability request message requests flight availability for a city pair on a specific date for a specific number and type of passengers.

The request can also be narrowed to request availability for a specific airline, flight or booking class on a flight, all for a specific date. Optional request information can include:

- Time / Time Window
- Connecting cities
- Client Preferences (airlines, cabin, flight types etc.)

The Air Availability response message contains flight availability for a city pair on a specific date. A set of origin and destination options is returned, each of which contains one or more (connecting) flights that serve the city pair. For each flight the following information is returned:

- Origin and destination airports
- Departure and arrival date/times
- Booking Class availability
- Equipment
- Meal Information
- Codeshare information

## OTA\_AirBaggageRQ/RS

### *Message Pair Overview*

The OpenTravel Air Baggage message pair provides baggage pricing, allowance and list and/ or catalog functionality:

- Baggage Pricing
- Baggage Allowance
- Baggage List or Catalog with Pricing

Requested baggage operations may be based on a broad variety of criteria, including:

Specific request type (Baggage Allowance, Baggage Charge, Baggage Allowance and Charge and Baggage List)

- A specified airline supplier
- A specified company (associated with the traveler)
- A specific flight
- An air reservation (PNR)
- A traveler type
- Traveler loyalty benefits
- A fare group (including private and negotiated fares)

- An origin/ destination pair
- An associated offer
- A service family (product group and subgroup)
- Specific baggage information, including checked in indicator, quantity of pieces (checked bag and carry on), weight, dimensions, special item code & description, and excess baggage occurrences.

Returned information includes:

- Allowance and Charge Information
- Baggage allowance and charge by origin and destination:
- Indicator if the baggage allowance may be subject to air supplier merchandising offers
- Baggage allowance details, including weight, dimensions, special item codes & descriptions, service family and per item pricing (including currency type, amount, taxes, exchange rate details, redemption currency, pricing rules and pricing qualifiers)
- Booking and upgrade instructions
- Associated loyalty program that influenced the baggage allowance and/or pricing
- Marketing airline
- Service or bag specific fee calculation information or warnings
- Ticket box
- Airline merchandising offers that apply to baggage allowance and/or charges
- The total baggage price for the entire trip (including all passengers)
- Baggage List or Priced Catalog by origin and destination:
- Marketing airline
- Baggage detail, including maximum pieces, EMD type value, allowance method, service location, and service date)
- Airline or ATPCO service family with pricing and booking instructions and booking instructions, pricing, ticket box and offers at the product subgroup level

## OTA\_AirBookRQ/RS

### *Message Pair Overview*

The OpenTravel Air Book message pairs books a specific itinerary for one or more identified passengers. The messages contain optional pricing information, allowing the booking class availability and pricing to be re-checked as part of the booking process.

- Optional request information can include:
- Seat and meal requests
- Special Service Requests (SSR), Other Service Information (OSI), Remarks
- Fulfillment information—payment, delivery details, type of ticket desired

If the booking was successful, the Air Book response message contains the itinerary (including the directional indicator, status of the booking, and number of passengers), passenger and pricing information sent in the request, along with a booking reference number (PNR Locator) and ticketing information.

## OTA\_AirCheckInRQ/RS

## *Message Pair Overview*

The OpenTravel Air Check In message pair provides air travel check-in function specifically for the self-service channels (kiosks, web and mobile).

These messages allow customers to check-in for eligible flights using various channels (kiosks, web and mobile) and provide the ability to perform related functions such as checking baggage, issue bag tags, boarding passes and entering further passenger information (APIS data etc.)

## OTA\_AirDemandTicketRQ/RS

### *Message Pair Overview*

The OpenTravel Air Demand Ticketing message pair provides air travel ticketing functionality used for requesting ticket fulfillment.

## OTA\_AirDisplayQueueRS

### *Message Pair Overview*

The OpenTravel [OTA\\_ReadRQ](#) and Air Display Queue response messages are used to display booking files on a queue. The booking file data can be requested in full format whereby the booking file(s) on that queue will be returned or in condensed format containing queue information. This data will be used for queuing and booking reference. Additionally, the items can be removed from queue (by displaying them) or left on the respective queue.

## OTA\_AirFareDisplayRQ/RS

### *Message Pair Overview*

The OpenTravel Air Fare Display message pair allow clients to request information on fares which exist between a city pair for a particular date or date range. No Inventory check for available seats on flights is performed by the server before returning the Fare Display Response message.

The request can optionally contain information indicating that a more specific response is required. This information can include passenger information, specific flight information and information on the types of fares that the client is interested in. The response message contains repeating FareDisplayInfo elements, each of which contains information on a specific fare contract including airline, travel dates, restrictions and pricing. It can optionally return information on other types of fares that exist, but have not been included in this response.

## OTA\_AirFlifoRQ/RS

### *Message Pair Overview*

The OpenTravel Air Flight Information message pair provides updated information on the operation of a specific airline flight. The request requires the airline, flight number and departure date. The departure and arrival airport locations can be also be included.

The message response includes real-time flight departure and arrival information. The following flight operation data is included in the response:

- Departure airport
- Arrival airport

- Marketing and operating airline names, when applicable
- Flight number
- Type of equipment
- Status of current operation
- Reason for delay or cancellation
- Airport location for diversion of flight
- Current departure and arrival date and time
- Scheduled departure and arrival date and time
- Duration of flight
- Flight mileage
- Baggage claim location

## OTA\_AirDetailsRQ/RS

### *Message Pair Overview*

The OpenTravel Air Details message pair provides flight leg and codeshare information for a specific flight on a specific date between a city pair.

The Air Details Response message contains airline, arrival and departure times, equipment, meal, and duration information (total and ground) for each leg of a flight. It also contains codeshare information, on time percentage, and electronic ticketing eligibility.

## OTA\_AirLowFareSearchRQ/RS

### *Message Pair Overview*

The OpenTravel Air Low Fare Search message pair provides priced itinerary options for flights between specific city pairs on certain dates for specific numbers and types of passengers. Optional request information can include:

- Time / Time Window
- Connecting cities.
- Client Preferences (airlines, cabin, flight types etc.)
- Flight type (nonstop or direct)
- Number of itinerary options desired

The Air Low Fare Search response message contains a number of 'Priced Itinerary' options. Each includes:

- A set of available flights matching the client's request.
- Pricing information including taxes and full fare breakdown for each passenger type
- Ticketing information—ticket advisory information and ticketing time limits.
- Fare basis codes and the information necessary to make a rules entry.

## OTA\_AirPriceRQ/RS

### *Message Pair Overview*

The OpenTravel Air Price message pair provides pricing information for specific flights on certain dates for a specific number and type of passengers.

The message allows for optional information such as fare restriction preferences and negotiated fare contract codes to be included.

The pricing request contains the information necessary to perform an availability / sell from availability / price series of entries on an airline CRS or GDS.

The pricing response message contains a 'Priced Itinerary'. This includes:

- The set of flights sent in the pricing request message
- Pricing information including taxes and full fare breakdown for each passenger type
- Ticketing information
- Fare Basis Codes and the information necessary to make a Fare Rules entry.

## OTA\_AirRulesRQ/RS

### *Message Pair Overview*

The OpenTravel Air Rules message pair provides text rules for a specific fare basis code for an airline and city pair on a specific date. Optional information allows negotiated fare contract codes to be included in the request.

The response message contains a set of text (human readable) rule information paragraphs. Each paragraph is identified by a rule code.

## OTA\_AirScheduleRQ/RS

### *Message Pair Overview*

The OpenTravel Air Schedules message pair provides a customer or a third party entity with the ability to view airline flight schedules. This message pair requires the customer to specify the departure and arrival cities for a specific date. It offers flight information on airlines that provide service between the requested cities.

The Air Schedules messages could be used for the following circumstances:

- A customer wants to determine what airlines offer service to/from specific cities.
- A customer is looking for a specific flight number. By entering the arrival and departure cities and knowing the approximate arrival or departure time the customer can locate their specific flight number.
- A customer needs to determine the days of the week that service is scheduled to and from the requested destination.
- A customer wants to know the type of aircraft used to fly a route. Some customers prefer to fly on larger types of aircraft.

The Air Schedules message also contains other information that customers may be interested in, including:

- Meal service
- Duration of flight
- On-time statistics
- Smoking allowed indicator

In addition, these messages provide the foundation for electronic timetables.

## OTA\_AirSeatMapRQ/RS

### *Message Pair Overview*

The OpenTravel Air Seat Map message pair display the seats that are available for a given flight, as well as their location within the aircraft. To make a seat assignment when using an online booking product, a customer will frequently access a seat map display to determine which seats are available and then they will make a separate seating request.

These messages identify all the information necessary to request and return an available seat map for a particular flight. Types of information for the seat map request include: airline, flight number, date of travel, class of service and frequent flier status.

The response message includes: flight, aircraft and seat description information.

## OTA\_AirBookModifyRQ/OTA\_AirBookRS

### *Message Pair Overview*

The OpenTravel Air Book Modify Request/ Air Book Response message pair modify an existing booking file. The message contains all elements of the OTA\_AirBookRQ message, plus a general type of modification, e.g. name change, split, cancel or other, is indicated within the 'ModificationType' attribute.

The modification operation on the different elements is either indicated with the existing attribute 'Status' (for air segments, SSR's and seat requests) or with attribute 'Operation' of Type ActionType for the other elements (e.g. Other service information, remarks, AirTraveler Elements, etc.).

All the data to be changed is submitted in the AirBookModifyRQ element, while the AirReservation element contains all existing data. This allows the receiving system to perform a consistency check before updating the booking file. Please note that in order to keep the message small, the existing information could be omitted.

Please note that changes to a booking may result in required updates of the ticket (e.g. re-validation), implied charges for a change, altered pricing and/or the generation of some fees that may need to be collected, and these ticketing, pricing and fulfillment operations are not in scope for this message pair nor referenced in the following use cases.

The response to the OTA\_AirBookModifyRQ message is the OTA\_AirBookRS message.

## OTA\_AirGetOfferRQ/RS

### *Message Pair Overview*

The Get Offer request message provides trip and passenger characteristics to be used by an offer processing system to target relevant offers. Request criteria may be specified for the entire trip or by individual traveler and/or arranger and include any combination of the following:

- Confirmed booking —including air itinerary, traveler/arranger, purchased offers, payment, pricing and ticketing information
- Pre-booking —including origin/destination, itinerary, traveler/arranger preferences and specific flight information
- Baggage —including item type, quantity, description, origin/destination, marketing/operating carriers and traveler/itinerary association

- Seat information —including marketing classification, requested quantity and traveler/itinerary association
- Offers to include and/or exclude
- Offers that have already been presented and/or purchased

Additional supported business functionality includes:

- Offer family encoding by airline suppliers and/or ATPCO
- Detailed point of sale information
- Pricing structure flexibility, including display/ pricing currency(s), ticketing country/ city, and loyalty program redemption support
- Offer stages that specify the stage in the journey at which the ancillary offer request is being made or the offer was purchased, e.g. shopping and check-in
- Travel insurance product offers included with ancillary offers

The Get Offer response message returns relevant offers that meet the search criteria. Returned information includes:

- Pricing structure details—including exchange rates, currency overrides and accepted payment currency that apply to all offers unless override information is provided elsewhere in the message
- Priced offers—Priced offer information, including offer name and description, maximum and minimum offer quantity(s), service family, pricing details including redemption mile pricing, booking instructions, restrictions, penalties, multimedia, commissions and currency overrides

# CAR MESSAGES

## Messages Overview

There are several OpenTravel Car Messages that provide a wide range of functionality, including:

[Vehicle search with availability & rates](#) - The OpenTravel Vehicle Availability with Rates message could be used in any of the following circumstances:

A customer is performing a simple booking and will come into the branch office to pick-up and drop off the vehicle.

A customer requires a pick-up service at his/her home or office. In this case the Off Location Services element can be used to provide basic address information as well as special instructions.

A customer requires a delivery and collection service, where a car will be delivered to a specific location and the keys left in a secure place. Again, the Off Location Services element can be used to hold address information as well as special instructions.

- [Make, modify](#) and [cancel](#) reservations - The OpenTravel Vehicle Reservation (make) message pair is intended for a single reservation. The OpenTravel Vehicle Modify message pair is intended for customers to change information on an existing reservation. The OpenTravel Vehicle Cancel message pair is an extension of the generic OpenTravel cancel functionality (OTA\_CancelRQ/RS.) To cancel a reservation, the trading partner or customer provides the supplier or integrator with the exact Unique ID for the reservation. If the Unique ID is unknown, the trading partner may use the generic OpenTravel Retrieve Reservation (OTA\_ReadRQ/OTA\_ResRetrieveRS) message pair to search for an exact match.
- [Inventory reporting/ notification](#) - The OpenTravel Vehicle Rental Location Detail message pair provides detail information regarding a vendor's location based on an input location code.
- Vehicle [check in](#) and [check out](#) - The OpenTravel Vehicle Check In message pair provides the detail information related to the return of a rented vehicle to systems with a "need to know" that a vehicle has been returned and the rental agreement closed. The OpenTravel Check Out message pair provides all the data fields needed to capture the pertinent detail data related to renting a specific car to a customer, and send that information to automated systems that have a "need to know" that a rental vehicle was "checked out."
- [Vehicle location details reporting/ notification](#) - The OpenTravel Vehicle Location Details Notification message pair provides the capability for a car supplier to send location detail information to another system for the purpose of updating the other system's database.
- [Vehicle exchange](#) - The OpenTravel Vehicle Exchange message pair provides the capability to exchange a vehicle. This message differs from an OTA\_VehCheckOutRQ/RS in that a vehicle exchange is a specialized version of a check in/check out message in that it provides the ability to put the details of two vehicles inside one message.
- [Rental location information reporting](#) - The OpenTravel Vehicle Rental Location Detail message pair provides detailed information regarding a vendor's location based on an input location code.
- [Location-based search for vehicle vendors](#) - The OpenTravel Vehicle Location Search message pair provides for a search of vendor locations based on input criteria.
- [Rate plan requests & notification](#) - The OpenTravel Vehicle Rate Plan Notification message pair provides the capability for a car supplier to send rate detail information to another system for the purpose of populating the other system's database.
- Rate rules [requests & notification](#) - The OpenTravel Vehicle Rate Rule Notification message pair provides the capability for a car supplier to send rate rules information to another system for the purpose of populating the other system's database. The OpenTravel Vehicle Rate Rule message pair provides the capability for a car supplier to send rate rules information to another system for the purpose of populating the other system's database.

- [Vehicle reservation status notification](#) - The OpenTravel Vehicle Reservation Status Notification message pair is intended for updating a single reservation. Typically, a reservation message (e.g., OTA\_VehResRQ/RS), which resulted in a pending reservation, has been exchanged between two trading partners (e.g., a GDS and the car rental company) prior to this Reservation Notification message pair.

## Messages List

- [OTA\\_VehicleCommonTypes](#)
- [OTA\\_VehAvailRateRQ/RS](#)
- [OTA\\_VehCancelRQ/RS](#)
- [OTA\\_VehCheckOutRQ/RS](#)
- [OTA\\_VehCheckInRQ/RS](#)
- [OTA\\_VehLocDetailsNotifRQ/RS](#)
- [OTA\\_VehExchangeRQ/RS](#)
- [OTA\\_VehLocDetailRQ/RS](#)
- [OTA\\_VehLocSearchRQ/RS](#)
- [OTA\\_VehModifyRQ/RS](#)
- [OTA\\_VehRateNotifRQ/RS](#)
- [OTA\\_VehRateRuleNotifRQ/RS](#)
- [OTA\\_VehRateRuleRQ/RS](#)
- [OTA\\_VehResRQ/RS](#)
- [OTA\\_VehResStatusNotifRQ/RS](#)
- [OTA\\_VehRetResRQ/RS](#)
- [OTA\\_VehResNotifRQ/RS](#)

## OTA\_VehAvailRateRQ/RS

### Message Pair Overview

The OpenTravel Vehicle Availability with Rates message pair is intended for a simple reservation. This message pair assumes the customer has already performed a location search and has found a specific rental branch. This message is typically used in the following circumstances:

- A customer is performing a simple booking and will come into the branch office to pick-up and drop off the vehicle.
- A customer requires a pick-up service at his/her home or office. In this case the Off Location Services element (*OffLocService*) can be used to provide basic address information as well as special instructions.
- A customer requires a delivery and collection service, where a car will be delivered to a specific location and the keys left in a secure place. Again, the Off Location Services element can be used to hold address information as well as special instructions.

### Other Message Pair Notes & Features:

- Only one set of date/times may be specified for the availability message, e.g. multiple dates and times will require multiple messages.
- Special equipment, such as hand controls or a baby seat can be accommodated through this message pair.
- Special equipment may be associated with a specific car, rather than as part of the more general reservation.
- Special circumstances such as chauffeur-driven cars are not accommodated in this message.
- The root tag of the OTA\_VehAvailRateRQ contains standard payload attributes found in all OpenTravel 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. The attribute *ReqRespVersion* is a positive integer value that indicates the version number requested for the response message.

## OTA\_VehCancelRQ/RS

### Message Pair Overview

The OpenTravel Vehicle Cancel (reservation) message pair is an extension of the generic OpenTravel cancel functionality ([OTA\\_CancelRQ/RS](#).)

To cancel a reservation, the trading partner or customer *must* provide the supplier or integrator with the exact Unique ID for the reservation. If the Unique ID is unknown, the trading partner may use the generic OpenTravel Retrieve Reservation ([OTA\\_ReadRQ/OTA\\_ResRetrieveRS](#)) message pair to search for an exact match.

This message pair can be supports both single and two-phase approaches. In a single-phase approach, the customer simply requests to “cancel this reservation”. A two-phase approach introduces the concept of a “what if” question, e.g. “What if I cancel this reservation?”

The response to the first phase will identify any penalties, any subsequent costs, etc. The second phase is where the action is confirmed to “go ahead and complete the request” or “ignore the request I just sent.”

The purpose of the request message is indicated using the “*Type*” Attribute:

- Initiate—indicates the initial request
- Ignore—stop the request
- Confirm—to complete the modification

The state of the reservation is then indicated in the response message using the Status Attribute:

- Pending—cancellation is possible but not completed
- Ignored—cancellation ignored
- Cancelled—cancellation completed

Other Message Pair Notes & Features:

- This message pair does not require a prior Retrieve Reservation ( [OTA\\_VehResRQ/RS](#) ) message, BUT some suppliers may require a Retrieve Reservation message to be used prior to a Cancel message, so check your trading partner requirements.
- This message pair can only be used to cancel a single, specific reservation; it cannot be used to cancel multiple reservations at one time.
- The generic OpenTravel cancel response message pair ( [OTA\\_CancelRQ/RS](#) ) has been extended to provide additional details of the vehicle reservation information to the customer. This will aid the customer in understanding the consequences of a cancellation.

## OTA\_VehCheckOutRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Check Out message pair provides all the data fields needed to capture the pertinent data related to renting a specific car to a customer, and send that information to automated systems that have a “need to know” that a rental vehicle was “checked out.” This would include rental location, customer information, payment information, intended length of rental, and monetary amounts for vehicle rental, surcharges, fees, taxes, insurance, and ancillary services and equipment.

The “need to know” systems could be travel agent, car rental agency inventory management, or OpenTravel partner hosted systems.

The Vehicle Check Out response message confirms that the rental agreement is active or requests additional information.

- Vehicle Check Out messages could be used in any of the following circumstances:
- Initiate a Check Out transaction utilizing an existing reservation
- Initiate a Check Out transaction without an existing reservation, based on a vehicle request

Only one “vehicle rental” request at a time may be made with the Vehicle Check Out. The response will include any legal wording related to the rental transaction.

## OTA\_VehCheckInRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Check In message pair provides the detail information related to the return of a rented vehicle to systems with a “need to know” that a vehicle has been returned and the rental agreement closed.

The detail information includes date and time of return, condition (e.g. damage, fuel, mileage) of the vehicle and associated equipment, location of the return, and method of payment. This information is sent to systems that need to know that the vehicle can be returned to inventory, and that a rental agreement can be closed and the customer billed for the financial charges associated with renting the vehicle.

The Vehicle Check In response message confirms that the rental agreement is closed or requests additional information.

Vehicle Check In messages are typically used to initiate a check in transaction utilizing an existing rental agreement.

Only one “vehicle return” request at a time may be made with the Vehicle Check In message. The response will include any legal wording related to the rental transaction.

## OTA\_VehLocDetailsNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Location Details Notification message pair provides the capability for a car supplier to send location detail information to another system for the purpose of updating the other system’s database.

Multiple vehicle locations can be sent in one message and the action to be taken may include:

- Adding, deleting or doing a full replace on a location
- Adding, deleting, or replacing certain items of data for an existing location

## OTA\_VehExchangeRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Exchange message pair provides the capability to exchange a vehicle. This message differs from an [OTA\\_VehCheckOutRQ/RS](#) in that a vehicle exchange is a specialized version of a check in/check out message that provides the ability to put the details of two vehicles inside one message.

This transaction is part of a rental scenario and does not create a separate rental agreement. The message only accommodates one vehicle exchange at a time (e.g., group exchanges are not accommodated). Additionally, the message assumes an exchange will be made on a one-to-one basis (e.g. you won’t exchange 1 car for 2 cars).

The vehicle exchange message requires an identifying rental agreement number. The response may contain detailed charge information for the customer and any legal wording related to the rental transaction.

The vehicle exchange request message provides the data related to exchanging a specific car with another car and sending that information to automated systems that need to know that rental vehicles were exchanged. These systems could include car rental agency inventory management or OpenTravel travel partner hosted systems. This data may include, but is not limited to, contract number, rental location, reason for exchange, exchange location, incoming vehicle information, and outgoing vehicle information. Vehicle information may include incoming and outgoing vehicle asset numbers, fuel levels, and damage information, etc.

This vehicle exchange message does not change original car fuel level or conditions from the checkout. That data is available from the back-end system by recalling the original check out information.

## OTA\_VehLocDetailRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Rental Location Detail message pair provides detailed information regarding a vendor’s location based on an input location code. Only one location can be requested in each message.

The Vehicle Rental Location Detail response message returns detailed information regarding a vendor’s location. The returned information may include full address, phone number, hours of operation, delivery/collection services, seasonal dates open, customer loyalty services and hours, special program participation and credit cards accepted.

## OTA\_VehLocSearchRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Location Search message pair provides for a search of vendor locations based on input criteria. This allows a customer to find a location that is non-airport or a location when the customer does not know the airport code.

The criteria must be one or more of the following:

- A full address
- Zip/postal code
- Area code
- Area code/exchange
- Landmark name

Note that only one search may be requested in the OTA\_VehLocSearchRQ message. Multiple searches requires the sending of multiple OTA\_VehLocSearchRQ messages.

In the response, one or more vendor locations may be returned in a list format. The message includes high-level information such as specific address and number of miles from the requested criteria.

## OTA\_VehModifyRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Modify message pair is intended for customers to change information for an existing reservation.

The Vehicle Modify message pair does not require a Vehicle Retrieve Reservation ([OTA\\_VehResRQ/RS](#)) message pair to be used prior to the modify, but they may be used in conjunction with one.

This message pair can only be used for a single, specific reservation and cannot be used to change multiple reservations at one time.

A Vehicle Modify message can be a single phase or two-phase approach. For a single-phase message the client simply requests to “modify this reservation as follows...”. A two-phase message introduces the concept of a “what if” question (what if I modify this message as follows?). The response to the first phase will identify any penalties, any subsequent costs, etc. The second phase is where the action is confirmed to “go ahead and complete the request” or “ignore the request I just sent.”

The purpose of the request message is indicated using the Type Attribute:

- Initiate—indicates the initial request
- Ignore—stop the request
- Confirm—to complete the modification

The state of the response message using the Status Attribute:

- Pending—modification is possible but not completed
- Ignored—modification ignored
- Cancelled—modification completed

The Vehicle Modify message pair is typically used in the following circumstances:

- A flight has been delayed or canceled and a customer needs to update their reservation arrival.

- A customer has additional passengers and needs to change the car originally reserved.
- A customer now requires special equipment that was not on their original reservation.
- A customer has changed travel plans to fly in to a different city and needs to adjust the arrival city.
- A customer has changed travel plans to fly in or out on a different date and/or time and needs to adjust the dates and/or times. (See Use Case 1)
- A customer originally listed on the reservation is not going so the name on the reservation needs to change.
- A customer has received a promotional offer after making a reservation and wants to use it for the reservation. (See Use case 2)
- Other situations as determined by trading partners (as there are many other individual elements that could be changed but these are too numerous to list.)

## OTA\_VehRateNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Rate Plan Notification message pair allows car suppliers to send rate detail information to another system for the purpose of populating the other system's database. Multiple rates can be sent in one message and the action to be taken may include creating, deleting or updating a rate.

The request message is a "push" message. The intended partners are the car supplier systems, the CRS, a GDS, revenue management systems, etc.

## OTA\_VehRateRuleNotifRS/RS

### *Message Pair Overview*

The OpenTravel Vehicle Rate Rule Notification message pair provides the capability for a car supplier to send rate rules information to another system for the purpose of populating the other system's database. The rules are related to a rate and used to further define a rate. Multiple rules can be sent in one message and the action to be taken may include creating, deleting or updating a rule.

The request message is a "push" message. The intended partners are the car supplier systems, the CRS, the GDS and the revenue management systems.

## OTA\_VehRateRuleRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Rate Rule message pair provides the capability for a car supplier to send rate rules information to another system for the purpose of populating the other system's database. The rules are related to a rate and used to further define a rate. Multiple rules can be sent in one message and the action to be taken may include creating, deleting or updating a rule.

The request message is a "pull" message. The intended partners are the car supplier systems, the CRS, the GDS and the revenue management systems.

## OTA\_VehResRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Reservation message pair is intended for a single reservation. This message pair requires the customer to specify the location, either by a location search or by knowledge of the rental facility, and assumes the customer has already performed some kind of location search and pinpointed a specific rental branch.

An Availability with Rates ([OTA\\_VehAvailRateRQ/RS](#)) message pair may be exchanged prior to the Vehicle Reservation message pair, but is not necessary. A vendor may make a single reservation upon receiving only the reservation message.

The Vehicle Reservation messages are typically used in the following circumstances:

- A customer is performing a single booking and will come into the branch office to pick-up and drop off the vehicle.
- A customer requires a pick-up service at their home or office. In this case, the Off Location Services (*OffLocService*) element can be used to provide basic address information as well as special instructions.
- A customer requires delivery and collection service, where a car will be delivered to a specific location and the keys left in a secure place. Again, the Off Location Services (*OffLocService*) element can be used to hold address information as well as special instructions.
- Rates are already known to the customer or trading partner and only a reservation is needed to communicate the rental need. Special equipment, such as hand controls or a baby seat can be accommodated through this message pair. Special equipment can be returned related to a specific car, rather than as part of the more general reservation. Special circumstances such as chauffeur-driven cars are not accommodated in this message.

## OTA\_VehResStatusNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Reservation Status Notification message pair is intended for updating a single reservation.

Typically, a reservation message (e.g., [OTA\\_VehResRQ/RS](#)), which resulted in a pending reservation, has been exchanged between two trading partners (e.g., a GDS and the car rental company) prior to this Reservation Status Notification message pair.

## OTA\_VehRetResRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Retrieve Reservation message pair is intended for customers to display their previously made reservation(s). This message pair allows a customer to retrieve one specific reservation or receive a list of reservations that match specific criteria.

One of the following is required to complete a Vehicle Retrieve Reservation request:

A Unique ID

Customer Loyalty Number, or

Person Name.

Many companies may require a combination of these three. Other optional items are pickup information, telephone number, and vendor.

Trading partners may mandate that additional fields are mandatory. In the case where a list of reservations is retrieved, the list will provide key high-level information; such as, dates and times, pick-up location, name and type of class of the vehicle. From this list, the customer can then drill down and retrieve one specific reservation.

The Vehicle Retrieve Reservation message pair are typically used in the following circumstances:

- A customer wants to verify that all information is accurate as the reservation may have been made months ago or by a third party. The traveler can verify that the reservation was made accurately and that pertinent information has not changed from the time the reservation was made.
- A customer is on the road and does not have their itinerary for their next location. Depending on the trading partner, this customer can retrieve the reservation and see their next location or a list of locations to which they are going.
- A customer needs to modify their reservation. Depending on the trading partner, a Vehicle Retrieve Reservation ([OTA\\_VehResRQ/RS](#)) transaction may be required before the Vehicle Modify is done.
- A customer wants to modify or cancel an existing reservation, but does not have the Unique ID (reservation number). In this case, the retrieve message ([OTA\\_VehResRQ/RS](#)) could be used to retrieve a list of reservations that match the search criteria and the customer could then select a single reservation from the list on which to perform further action.

## OTA\_VehResNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Vehicle Reservation Notification message pair addresses a common business need to be able to bulk transfer entire reservations between computer systems. Other OpenTravel vehicle messages support requesting new reservations and retrieving existing reservation information in a client/server style pull operation, however, they don't support transferring bulk reservation information in a notification style push operation like this message pair does.

To simplify retransmission logic, reservations are transferred in a transactional (all or nothing) model. If the receiver sends a response message with a *Success* element, all reservation information has been successfully transferred.

If the response contains an *Error* element, none of the reservation information has been transferred and the information should be retransmitted at a later time.

Each reservation included in the request message may include the following information:

- Confirmation number
- Reservation status (new, changed, canceled)
- Itinerary information
- Vehicle selection
- Pricing information
- Booking source information
- Coverage requested
- Optional equipment requested
- Special request comments

The intended partners are reservation systems, counter level systems, fleet management systems, accounting systems, and other trading partners.

# CRUISE MESSAGES

## Messages Overview

Booking a cruise reservation typically involves searching for the cruise followed by the creation of the booking. First, the traveler decides when and where the cruise is to be taken. After the sailing has been selected the traveler decides on the fare type and accommodation costs. Finally the cruise is purchased.

In the following section, the term “traveler” will be used. This could be a travel agency, a travel partner, a GDS, or anyone the cruise line has authorized to make bookings using these messages.

To book a cruise, the traveler typically performs the following steps:

1. [Search for a sailing using the CruiseSailAvailRQ/RS message pair](#) which could be sent to one or more cruise lines. The cruise line(s) would reply with a list of sailings or a warning to inform the requester that no sailings are available that satisfy the request. Once a sailing is chosen, all subsequent messages will only be sent to the selected cruise line.
2. [Request available fares for the selected voyage with the CruiseFareAvailRQ/RS message pair](#). A list of available fares will be returned by the cruise line and the traveler selects a fare.
3. [Request the available categories using the CruiseCategoryAvailRQ/RS message pair](#). A list of available categories will be returned by the cruise line and the traveler selects a category.
4. [After the category has been selected, and any time prior to creating the booking, pricing can be requested with the CruisePriceBookingRQ/RS message pair](#). The cruise line will return all of the pricing items for both the individual traveler and the booking.
5. [Request a list of available cabins in the selected category with the CruiseCabinAvailRQ/RS message pair](#). The cruise line will return a list of available cabins from which the traveler will make a selection.
6. [Place one or more cabins on hold using the CruiseCabinHoldRQ/RS message pair](#). This takes the cabin(s) out of the cruise line’s available inventory for a short period of time, typically fifteen minutes.
7. [If the traveler decides to request one or more different cabins to be held, the CruiseCabinUnholdRQ/RS message pair](#) is used to release the specific cabin(s) currently on hold.
8. [The traveler creates a booking with the CruiseBookRQ/RS message pair](#). In the request message, the traveler advises the cruise line of all required information for all accompanying passengers. The cruise line responds with the reservation number to confirm the booking.

Alternatively, if a traveler knows the cruise line’s available voyages and fares, there would be no need to use the [CruiseSailAvailRQ/RS](#) or the [CruiseFareAvailRQ/RS](#) message pairs. The traveler would start the shopping process by requesting the available categories for a selected fare for a selected sailing.

Other business operations supported by the OpenTravel Cruise messages include:

- Cancellations - Request the penalties for both the booking and the individual travelers that would be charged if the requested cruise reservation is canceled using the [CruiseCancellationPricingRQ/RS](#) message pairs.
- Dining Options - The [CruiseDiningAvailRQ/RS](#) message pair allow requests for dining options for a selected voyage and fare code and returns a list of dinings available for a given sailing and for a selected fare.
- Fast Sells – The [CruiseFastSellRQ](#) message is a multipurpose request to access one of the following cruise provider functions: Cabin hold (maximum of 4 cabins) when a cabin number is provided; Cabin availability when a cabin number is not provided and a category is provided; Category availability when neither cabin, nor category is specified in the request; Fare availability when the fare code provided in the request is either incorrect for the sailing or not available; and, Package availability when the package is specified, but not available for the sailing.

- Cruise Information - The [CruiseInfoRQ/RS](#) message pair allows the sharing of miscellaneous structured cruise information, including: cruise ship characteristics; embark/debark time; cruise policies; and cruise line contacts.
- Cruise Itineraries - The [CruiseItineraryDescRQ/RS](#) message allows customers to request the detailed itinerary of a specific sailing.
- Cruise Payments - The [CruisePaymentRQ/RS](#) message pair provides a traveler with the ability to submit payments for an existing reservation without first retrieving the reservation.
- Cruise Packages – The [CruisePkgAvailRQ/RS](#) message pair is used to request information about the pre-cruise, post-cruise, or the inclusive-cruise packages associated with a sailing.
- PNR Updates – The [CruisePNR UpdateNotifRQ/RS](#) message pair provides a defined structure to push unsolicited update messages (such as change in departure time and cancellation for non payment) to a travel partner to enable passenger record synchronization between the cruise line company and a travel partner.
- Shore Excursions – The [CruiseShorexAvailRQ/RS](#) message pair is used to request information about shore excursions associated with a sailing (such as a sight seeing tour or a shopping expedition).
- Special Services – The [CruiseSpecialServicesRQ/RS](#) message pair is used for requests about special services offered for a given sailing/reservation ID or to request details for a specific special service.

## Messages List

- [OTA CruiseCommonTypes](#)
- [OTA CruiseBookingDocumentRQ/RS](#)
- [OTA ReadRQ/OTA CruiseBookingHistoryRS](#)
- [OTA CruiseCabinAvailRQ/RS](#)
- [OTA CruiseCabinHoldRQ/RS](#)
- [OTA CruiseCabinUnholdRQ/RS](#)
- [OTA CruiseCancellationPricingRQ/RS](#)
- [OTA CruiseCategoryAvailRQ/RS](#)
- [OTA CruiseBookRQ/RS](#)
- [OTA CruiseDiningAvailRQ/RS](#)
- [OTA CruiseFareAvailRQ/RS](#)
- [OTA CruiseFastSellRQ](#)
- [OTA CruiseInfoRQ/RS](#)
- [OTA CruiseItineraryDescRQ/RS](#)
- [OTA CruisePaymentRQ/RS](#)
- [OTA CruisePkgAvailRQ/RS](#)
- [OTA CruisePriceBookingRQ/RS](#)
- [OTA CruisePNR UpdateNotifRQ/RS](#)
- [OTA CruiseSailAvailRQ/RS](#)
- [OTA CruiseShorexAvailRQ/RS](#)
- [OTA CruiseSpecialServicesRQ/RS](#)

## OTA\_CruiseBookingDocumentRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Booking Document message pair provides the ability to request a document type and its delivery method without first having to retrieve the reservation.

The request message is the `OTA_CruiseBookingDocumentRQ` in which the user can specify the reservation ID and the type of document (e.g., cruise itinerary, air itinerary etc.) the user wants along with a delivery mode (e.g., e-mail, fax, mail etc.)

The user may optionally provide a phone number, an e-mail address or an address to where the document is to be sent. If no phone number, e-mail or address is specified then the information provided within the booking is used.

The response message `OTA_CruiseBookingDocumentRS` returns the status of each delivery request that was made.

## OTA\_ReadRQ/OTA\_CruiseBookingHistoryRS

### *Message Pair Overview*

The OpenTravel Cruise Booking History message pair provides change/service history on an existing reservation.

The request message is the [OTA\\_ReadRQ](#) with which a user can specify the reservation ID for which they want to see the change history for. The [OTA\\_ReadRQ](#) message is also used to retrieve the reservation information and an indicator that distinguishes the service history request from the request for the current reservation information.

The response message `OTA_CruiseBookingHistoryRS` returns a list of history entries for the reservation changes that have taken place.

## OTA\_CruiseCabinAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Cabin Availability message pair provides a list of available cabins for a selected category on a voyage. The voyage information is defined by the VoyageID or the departure date and duration, depending on the cruise line. The fare code and the berthed and priced category codes are retained from the respective availability message.

The Cruise Cabin Availability request message requests Cabin Availability for a given sailing with a specific mode of transportation/ gateway city pair & currency, and for a selected fare/category pair. Optional request information can include: Guest city and Inclusive package.

The Cruise Cabin Availability response message contains a list of cabins available for a given sailing with a specific mode of transportation/ gateway city pair & currency and for a selected fare/category pair. For each cabin the following information may be returned: cabin number; position in ship; ship side; category location; remark; deck name; bed options; maximum cabin occupancy; cabin size; berthed category code; priced category code; status code; category indicator; cruise package information; group code; fare code; currency code; and marketing message.

## OTA\_CruiseCabinHoldRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Cabin Hold message pair instructs a cruise line to remove a selected cabin from available inventory for a short time interval, typically fifteen minutes. This allows the user time to enter additional information which is required to complete the booking, without concern for losing the cabin.

The voyage is identified by the VoyageID or the departure date and duration, depending on the cruise line. Multiple cabins are usually allowed to be placed on hold, up to a maximum defined by the cruise lines.

The response message indicates the status of each cabin for which a hold is requested. It is conceivable in the case of a request containing multiple cabins, that the status might be different for one or more cabins. If the cabin hold request was successful, the cruise line usually provides additional information in the response, such as the acceptable forms of payment.

The cruise line also may indicate the length of time the cabin will be held before being automatically released back into inventory, as well as which data elements the cruise line requires to complete the booking, dining, insurance, and other options.

## OTA\_CruiseCabinUnholdRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Cabin Unhold message pair is used to release a hold that was previously put on a cabin, allowing the cruise line to release the cabin back into its available inventory.

The request message allows the user to release multiple cabins-from multiple voyages-to address situations where the same cabin is currently held for back-to-back cruises. The voyage is identified by the VoyageID or the departure date and duration, depending on the cruise line.

The response message indicates the status for each cabin in the request. If unsuccessful, a warning or advisory message is usually returned.

## OTA\_CruiseCancellationPricingRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Cancellation Pricing message pair is sent to a cruise line vendor to request the penalties that would be charged if the requested reservation is canceled. The response message returns the penalties for both the booking and the individual travelers.

## OTA\_CruiseCategoryAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Category Availability message pair requests a list of cabin categories for a specified voyage and fare code. The voyage information is indicated by either the VoyageID or the departure date and duration depending on the cruise line.

The Category Availability Request message requests Category Availability for a given sailing with a specific Mode of Transportation/GatewayCity pair and currency and for selected fares (depending on the cruise line). Optional request information can include: Guest ages; Guest city; and Inclusive package.

The response message returns a list of cabin categories that correspond to the requested sailing and fare. In addition to the status, the response typically contains information about the location of the category on the ship,

the maximum occupancy of the cabins, priced and berthed codes, as well as price information about the category.

## OTA\_CruiseBookRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Book message pair is used to create a reservation on the cruise line reservation system, for a specific sailing, with a specified mode of transportation/gateway city pair and currency, and a selected fare/category pair, for one or more passengers, in a specific cabin. Typically the request would be made by a Global Distribution System (GDS) or an Internet booker, or another travel broker. Prior to submitting a Create Booking request message, a cabin must have been placed on hold. The only exception to this is if a cabin is reserved as a guarantee, rather than a specific cabin.

The response message indicates if the request was successful, and returns a Reservation ID number and may also integrate the booking into a GDS' PNR based on the cruise line reply.

## OTA\_CruiseDiningAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Dining Availability message pair provides a list of dining options for a selected voyage and fare code. The voyage information will be entered via the VoyageID—or the departure date and duration—depending on the cruise line and guest count is required. Optional request information can include:

- Dining Room name
- Dining Room operation times
- Type of restaurant (fast food, café, fine dining)
- Type of cuisine (French, Thai, Vegetarian)

The Dining Availability Response message contains a list of dinings available for a given sailing and for a selected fare. For each dining the following information may be returned: Dining code; Dining status; Sitting; Meal name; Meal times; and Marketing message.

## OTA\_CruiseFareAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Fare Availability message pair provides a listing of available fares for a selected voyage. For some cruise lines, the voyage must be identified via the VoyageId; for others, the departure date and duration identify the sailing.

The guest count structure is used to communicate the number of guests that will be sailing. Although guest transportation information is optional, a prudent practice is to enter the mode of transportation and gateway city, because some fares have restrictions based on transportation.

The response message returns a list of available fares for which the traveler qualifies based upon the criteria provided.

## OTA\_CruiseFastSellRQ

### *Message Pair Overview*

The OpenTravel Cruise Fast Sell Request message requires the user to provide all mandatory sailing information. Optionally, category, price program, package code and cabin information may be provided.

The Fast Sell message is a multipurpose request to access one of the following cruise provider functions:

1. Cabin hold (maximum of 4 cabins) when a cabin number is provided,
2. Cabin availability when a cabin number is not provided and a category is provided,
3. Category availability when neither cabin, nor category is specified in the request,
4. Fare availability when the fare code provided in the request is either incorrect for the sailing or not available,
5. Package availability when the package is specified, but not available for the sailing.

The response is conditional upon inventory availability for the requested function:

- If the cabin is available, the response will be the cabin hold message ([OTA\\_CruiseCabinHoldRS](#)).
- If cabin is unavailable for hold, the response will be cabin availability ([OTA\\_CruiseCabinAvailRS](#)).
- If category is unavailable, the response will be category availability ([OTA\\_CruiseCategoryAvailRS](#)).
- If the fare is not available the response will be fare availability ([OTA\\_CruiseFareAvailRS](#)).
- If the sailing or package is not available, the response will be sailing ([OTA\\_CruiseSailAvailRS](#)) or package availability ([OTA\\_CruisePkgAvailRS](#)).

## OTA\_CruiseInfoRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Information message pair provides miscellaneous information on cruise ship statistics, embark/debark times, cruise policies, cruise line contacts, etc.

In the request message, the user can specify the type of information being requested by either a code or name.

Sailing information may be provided if the user wishes to see sailing-specific (e.g., cabin or category) details. By providing the reservation ID, consumer advice for that reservation may be returned.

The response message returns miscellaneous structured cruise information (e.g., cruise ship characteristics, embark/debark times, cruise policies and cruise line contacts).

## OTA\_CruiseItineraryDescRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Itinerary Description message pair allows customers to request the detailed itinerary of a specific sailing. The Cruise Itinerary Description request message can accept a reservation ID/ booking number or a ship and sail date to retrieve the itinerary. The response message contains the detailed itinerary in the following formats:

- Every location the ship visits in its entire voyage is listed as a port on the itinerary.
- The departure date and time are listed for each port except the disembarkation port.

- The arrival date and time are listed for each port except the embarkation port.
- The port element can also specify “At Sea”, indicating a sailing period.
- Each port listed contains an indicator to specify whether the ship docks at that specific location. This feature can be used to include scenic locations such as “The Great Barrier Reef” on the itinerary, even though the ship does not dock at the location.
- Each port listed also contains an indicator to specify whether or not shore excursions exist at that location.

## OTA\_CruisePaymentRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Booking Payment message pair provides the user with the ability to submit payments for an existing reservation without first retrieving the reservation. The user can specify the reservation ID for which the payment is being submitted along with the details of the form of payment (e.g. credit card, voucher, etc.) and the amount. Optionally, information about the agent making the payment on behalf of the customer may be provided.

The response message returns the status of each payment that was submitted.

## OTA\_CruisePkgAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Package Availability message pair is used to request information about the pre-cruise, post-cruise, or the inclusive-cruise packages associated with a sailing. A cruise package is normally a stay in a hotel for one or more days at the beginning or end of a cruise. The hotel is generally located near the port of embarkation or debarkation.

The response message contains a list of requested packages along with the associated pricing for these packages.

## OTA\_CruisePriceBookingRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Price Booking message pair provides pricing for all components of a booking. For the cruise pricing to be correct, all of the components to be purchased must be included in the Cruise Price Booking request. This includes the sailing by using either the VoyageId or the ship and sail date depending on the cruise line, the priced and berthed category, the fare code, the group code (when applicable), the air gateway, the pre and/or post land packages, the shore excursions, and anything else the cruise line sells at the time of making a reservation.

The response message returns the pricing breakdown by component for both the booking and the individual travelers.

## OTA\_CruisePNR\_UpdateNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise PNR Update Notification message pair provide unsolicited messages originating from a cruise line company to the GDS or travel partner. When a cruise booking is made, some travel partners may cache a copy of the booking information in their local application systems, however, there may be occasions

where a cruise line makes changes to the booking in their system, which would cause the records to become non-synchronized.

This could be for a number of reasons, including: Change in departure time, direct communication from the customer to the cruise line that bypasses the travel partner, change from waitlisted status, cancellation for non-payment, reinstatement of a booking, etc.

This message pair provides a defined structure to push these unsolicited update messages to the travel partner, and enable passenger record synchronization between the cruise line company and the travel partner:

- Precondition: A cruise booking by the travel partner exists, and the cruise line company has made changes.
- Post-condition: The travel partner receives the change message.

## OTA\_CruiseSailAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Sailing Availability message pair provides a listing of available voyages for: a date range, a specific geographic region and/or a specific ship code. Cruise vendors typically don't restrict the returned list of voyages to an exact request date, but will return sailings that fall within a time window around the requested date. There are two common methods in the cruise industry of identifying a sailing. Some cruise lines return the VoyageID, and other cruise lines only return the Departure Date and Duration to identify the sailing.

The Cruise Sailing Availability request message requests sailing availability for a geographical region and a specified date—for a specified number of passengers. Optional request information can include cruise lines and ship codes. The request can be narrowed to request availability for a specific cruise line or specific ship.

The Cruise Sailing Availability response message contains sailing availability for 1 or multiple cruise lines for a given region or ship—on a date and duration range. For each sailing the following types of information may be returned:

- Cruise line code, ship code, region code and status code
- Departure date, duration and port code
- Arrival date, duration and port code
- Voyage number and number of ports visited
- Maximum cabin occupancy and category location
- First and second dining services status
- Sailing indicators and free flow text
- Available modes of transportation
- Available currencies
- Cruise package information
- Registration information

This message contains similar information to a standard airline CRS or GDS sailing availability response message.

## OTA\_CruiseShorexAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Shorex Availability message pair is used to request information about shore excursions associated with a sailing. A shore excursion is normally a sightseeing tour, a shopping expedition or activity such as snorkeling or horseback riding to be taken while the ship is visiting one of the ports of call on a cruise.

The Shorex Availability request message requests shore excursions for a given sailing and currency.

The Shorex Availability response message contains shore excursion packages that are available for the given sailing/ports along with the associated pricing.

## OTA\_CruiseSpecialServiceAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Cruise Special Service Availability message pair allows a cruise vendor to offer special services to their guests, in addition to offering the cruise sailing itself (often at no charge to the customer.)

These can be categorized as:

- Occasion (birthday, anniversary, back-to-back cruise, etc.)
- Special Service (wheelchair, special dietary needs, etc.)
- Language (language translation needs)

Additional categories can be defined by the cruise vendors and special services are offered/selected at either a passenger or cabin level. The Cruise Special Service Availability Request message queries all the special services available for a given sailing/reservation ID—or details for a specific special service. The Cruise Special Service Availability Response message returns information about which special service packages are available for the given sailing.

# DESTINATION ACTIVITY MESSAGES

## Messages Overview

Travelers often want localized experiences during their vacation or journey and these may include taking guided tours, attending performances or shows, and buying tickets to local attractions. All these products, while varying greatly in makeup, are collectively termed here as “Destination Activities.”

The purpose of these messages is to enable the electronic distribution and sale of this type of product by requesting the reservation functionality of a target system and making a reservation.

Future versions of this message may support availability requests, modification, and the transfer of descriptive information, rates and inventory.

## Messages List

- OTA\_DestinationActivity.xsd
- [OTA\\_DestActivityCapabilitiesRQ/RS](#)
- [OTA\\_DestActivityResRQ/RS](#)

## OTA\_DestActivityCapabilitiesRQ/RS

### *Message Pair Overview*

The OpenTravel Destination Activity Capabilities message pair enable a “seller” system to discover the capabilities of a “target” system. Once the capabilities of a target system have been discovered, the seller system may either alter its behavior in regards to selling that target product or choose not to offer that product range.

The Destination Activity Capabilities Request message sends a request for the functional capabilities of the target system so following messages—such as the Destination Activity Reservation message ([OTA\\_DestActivityResRQ/RS](#))—may be altered depending upon the target system's level of functionality.

The Destination Activity Capabilities Response message returns a set of data representing the available functionality in the system, including:

- A *MultipleItemsInd* attribute that indicates if the target system supports receiving multiple items in one reservation call, like a shopping cart. Some target reservation systems do not support receiving multiple items in one reservation call, and in this case, if the seller system sells multiple products in its shopping cart from the one target system, one reservation call per item must be sent.
- A *CustSubAllocationInd* attribute that indicates if the target system supports having a subset of the traveler group utilized per item. For example, a family of four (husband, wife, two children) are making a reservation and details for all four family members are sent. There are three items in the cart – a family ticket to an amusement park (which all four members are using), a hop-on/hop-off bus tour (again with all four family members), and a museum ticket for only the husband and wife (and priced and sold as two adults only).

If the *CustSubAllocationInd* is true, the target system supports this arrangement and all traveler details are sent with a list of which travelers are utilizing a service specified per item. If the *CustSubAllocationInd* is false, the target system does not support this behavior and the seller system must either send two separate reservations or prohibit this functionality during a customer interaction.

A *FullCustDetailsInd* that indicates if all customer details are required (for example, if four travelers are specified then all four names must be supplied) or if only one contact name and associated information ID required. If *FullCustDetailInd* is true, the seller system must supply all traveler details for any reservation and the seller system has several options, including:

- Collecting all traveler details only when necessary when selling product for this type of target system,
- Collecting all traveler details all the time, or
- Electing not to sell product from this target system.

## OTA\_DestActivityResRQ/RS

### *Message Pair Overview*

The OpenTravel Destination Activity Reservation message pair allows a reservation to be requested/made for a destination activity. There is no requirement to determine availability prior to sending a reservation request as 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 Destination Activity Reservation Request message sends a request for a reservation to another system, and the Destination Activity Reservation Response message returns the requested reservation if the request can be processed—or includes warnings from business processing rules or errors if the request did not succeed.

# DYNAMIC PACKAGE MESSAGES

## Messages Overview

The OpenTravel Dynamic Package messages use existing OpenTravel Air and Hotel message functionality in combined messages that provide one central location for air and hotel searches to be run simultaneously. Rather than sending a message for each component, a Dynamic Package message allows one request to handle multiple component requests.

The Dynamic Package messages are functionally different from the existing OpenTravel Package messages in that rather than operating on static packages with predetermined components, the Dynamic Package messages allow the creation of travel packages with independent air, hotel and rental car selections. In addition, the OpenTravel Dynamic Package messages support “package options” that may be added to a dynamic package as either stand-alone products or included items for existing components. Examples of package options include show tickets, amusement park tickets, airport transfers, meal plan vouchers, club passes and travel insurance.

In addition to dynamic package availability and booking, the Dynamic Package messages also provide the following business functionality:

- Retrieving booked dynamic packages
- Modifying entire or components of booked dynamic packages
- Canceling entire or components of booked dynamic packages
- Trading partner notification of changes to dynamic package components

## Messages List

- [OTA\\_DynamicPkgCommonTypes.xsd](#)
- [OTA\\_DynamicPkgAvailRQ/RS](#)
- [OTA\\_DynamicPkgBookRQ/RS](#)
- [OTA\\_ReadRQ/OTA\\_DynamicPkgBookRS](#)
- [OTA\\_DynamicPkgModifyRQ/OTA\\_DynamicPkgBookRS](#)
- [OTA\\_CancelRQ/OTA\\_DynamicPkgBookRS](#)
- [OTA\\_DynamicPkgModifyNotifRQ/RS](#)

## OTA\_DynamicPkgAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Dynamic Package Availability message pair is used to obtain availability for various components of a dynamic package reservation. It can be used for the following functions:

- Component (Air, Hotel, Car, PackageOption) Availability
- Component (Air, Hotel, Car, PackageOption) Availability with pricing
- Pricing for selected un-priced flight pairs. (This is useful for determining round trip pricing for unpriced outbound/return flights)

The Dynamic Package Availability Request message requests availability based on the following information:

- Origin / Destination cities
- Specified dates

- Number and types of passengers

The Dynamic Package Availability Response message contains availability for each requested component.

## OTA\_DynamicPkgBookRQ/RS

### *Message Pair Overview*

The OpenTravel Dynamic Package Booking message pair is used to book a specific dynamic package for one or more identified travelers. The message may contain pricing information in order that prices obtained from previous requests can be verified during the booking process. Typically, the result of a Dynamic Package Booking message would be a confirmed reservation.

This message pair may also be used for confirming package prices before making a complete booking request, reducing the number of pricing errors returned during a purchase process.

## OTA\_ReadRQ/OTA\_DynamicPkgBookRS

### *Message Pair Overview*

The OpenTravel Read (reservation) and Dynamic Package Book Response message pair is used to retrieve a previously booked package reservation for review.

## OTA\_DynamicPkgModifyRQ/ OTA\_DynamicPkgBookRS

### *Message Pair Overview*

The OpenTravel Dynamic Package (reservation) Modify message pair is used to modify an existing dynamic package reservation, which may include changes to guest information, travel or accommodation arrangements.

## OTA\_CancelRQ/OTA\_DynamicPkgBookRS

### *Message Pair Overview*

The OpenTravel Cancel (reservation) Request and Dynamic Package Booking Response message pair is used to cancel a dynamic package in its entirety. Once this message is successfully run, the entire package is canceled in the supplier's system.

Please note that either the Dynamic Package Booking Response ([OTA\\_DynamicPkgBookRS](#)) or the OpenTravel Cancel (reservation) Response ([OTA\\_CancelRS](#)) messages may be used to confirm the cancellation of the package.

## OTA\_DynamicPkgModifyNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Dynamic Package Modify Notification message pair is used to notify a trading partner of modification(s) to a dynamic package and is commonly used to synchronize data between a supplier and its partner's system. This message pair is initiated by a supplier system to update partner systems with any changes that have taken place in the partner's packages, including flight changes, car rental notices (such as inventory limitations) and hotel room modifications.

The Dynamic Package Modify Notification Request is a [“push” message](#) that provides the capability for a supplier trading partner to send notifications of any changes made to another system (e.g., if a flight is canceled or moved to a later time the endpoint system needs to be alerted to the change).

The Dynamic Package Modify Notification Response message is a simple message acknowledgment that may include a booking reference number.

# GOLF MESSAGES

## Messages Overview

The OpenTravel Golf messages provide three separate request/response pairs of messages to support the functionality of requesting data from another system for finding a golf course, inquiring as to availability, and booking a tee time.

All message pairs assume a [pull model](#), where the originating system requests a specific set of data (as agreed by trading partners) and are "no-state", meaning that the originating system will initiate the transaction and expect a response from the queried system. All message responses include the request identification and responses may be returned in any order.

## Messages List

- [OTA\\_GolfCommonTypes.xsd](#)
- [OTA\\_GolfCourseAvailRQ/RS](#)
- [OTA\\_GolfFacilityInfoRQRS](#)
- [OTA\\_GolfRateRQRS](#)
- [OTA\\_GolfCourseResRQ/RS](#)
- [OTA\\_GolfCourseResModifyRQRS](#)
- [OTA\\_GolfCourseSearchRQ/RS](#)

## OTA\_GolfCourseAvailRQ/RS

### *Message Pair Overview*

The Golf Course Availability message pair provides an availability search for a golf course and golf course amenities with (optional) pricing returned.

The Golf Course Availability Request message sends a request for course availability to another system. All the elements and attributes are optional, unless otherwise stated as required by a trading partner. This message is used to request availability at a known single course for one or more potential tee times. The specific information about the golfer or golfers is necessary in order to validate booking rules and set rates.

The Golf Course Availability Response message returns the requested set of data if the request has been processed, or includes warnings from business processing rules or errors if the request did not succeed.

## OTA\_GolfFacilityInfoRQ/RS

### *Message Pair Overview*

The Golf Facility Information message pair exchanges updated golf course facility information which is used to keep trading partner database(s) and cache(s) current with golf supplier facility information.

Golf facility request information includes the ID and (optional) name of the golf course facility, which may be repeated to request information about multiple courses.

The Golf Facility Information response returns detailed golf facility information for the specified facility ID(s). Response information includes:

- Facility ID, name and associated facility (if the course is on a hotel property)
- Short and long descriptions (with optional club type)
- Contact information, including phone and website
- Multimedia, including images, descriptions and movies that describe the golf facility
- Facility features, including dress policy, golf pros, on facility dining and course designer
- Available amenities, including type, description, pricing and reservation information
- Guideline tee time pricing, including minimum, maximum and average
- Location information, including physical and geo-coding
- Hours of operation
- Course conditions, including an optional source name or URL
- Policy information
- Directions
- Course closures
- Course restrictions
- Course scorecard

Additional message functionality includes transaction processing directives that influenced search results, such as display currency.

## OTA\_GolfRateRQ/RS

### *Message Pair Overview*

The Golf Rate message pair provides a method for exchanging tee time rates.

Requested rate criteria includes:

- The name of a golf course
- A date or range of dates, including days of the week, used to filter rate results
- Applied discounts that may include promotions and corporate ID's for negotiated rates
- Summary golfer type qualifying information, including quantity, age qualifier and golfer category
- Additional rate qualifiers to filter the rate results, including rate ID, rate code, rate plan type and rate category

The response message returns relevant rate plan details by golf course including:

- Golf course code, ID, name, short name
- Prepaid rate indicator
- Available tee time rate indicator
- Rate details, including code, rate name, rate period, corporate discount number for special rates associated with a specific organization, promotion codes and customer loyalty program codes
- The date or range of dates, including days of the week, used to filter rate results
- Amenities included with a rate
- Constraints for the rate plan, including golfer types and facility(s) that the rate applies to
- Charge information associated with a cart rental
- Charge information associated with greens fee
- Policy(s) associated with the rate
- Rate category(s)
- Rate rules

## OTA\_GolfCourseResRQ/RS

### *Message Pair Overview*

The OpenTravel Golf Course Reservation message pair is used to request a reservation at a known single course for one or more potential tee times. The specific information about the golfer or golfers is required to validate booking rules and set rates. If the booking entity has the authority to take a reservation without a request (from an existing block), then the *Notification* boolean in the message will be set to "Yes".

The OpenTravel Golf Course Reservation Response message includes all rate and confirmation information. If the booking entity has the authority to make the booking and uses the request message as a notification, the response message is merely an acknowledgment of receipt of that booking.

## OTA\_GolfCourseResModifyRQ/RS

### *Message Pair Overview*

The Golf Course Reservation Modification request provides the information required to modify an existing (booked) tee time reservation for one or more golfers.

Note that the response message for this request is the OTA\_GolfCourseResRS message.

## OTA\_GolfCourseSearchRQ/RS

### *Message Pair Overview*

The OpenTravel Golf Course Search message pair allows a requester to search for golf courses based on specified criteria. Criteria is a repeating set of features that are desired in the search for a golf course. The Name and Value pair define the criteria for the search. If the requester demands that the result be filtered on a particular criterion, then the *Required* boolean is set to "Yes". If the *Required* boolean is not set to "Yes", then the response can include courses that do not meet those exact criteria. Examples would be: Name="Architect", Value="Robert Trent Jones", Required="Yes"; Name="Location", Value="Myrtle Beach", Required="Yes"; Name="Caddies", Value="Yes", Required="No"; and Name="Length", Value="6600 Yds", Required="Yes", Operation="GT".

The *Name*, *Value*, and *Required* attributes are required, but the *Operation* attribute—that represents the operation to be used as the filter when a test against a criterion value is not an equality. e.g. GT (Greater Than), LT (Less Than) —is optional.

The Golf Course Search Request message sends a request for course information to another system. All the elements and attributes are optional, unless otherwise stated as required. The requesting system may request a detailed or summary response.

The Golf Course Search Response message returns a set of data representing the course(s) that meet the requested criteria. If the criteria attribute of *Required* is "Yes", then only the courses that meet the criteria will be returned. If the *Required* attribute is "No", then a course that does not meet the specified criteria may be included in the set.

In all cases, if the criteria has been included in the request message, the comparable trait and its value will be returned in the response message—along with the basic course information and identification. The message may also include warnings from business processing rules or errors if the request did not succeed.

# HOTEL MESSAGES

## Messages List

- [OTA\\_HotelCommonTypes.xsd](#)
- [OTA\\_HotelAvailRQ/RS](#)
- [OTA\\_HotelAvailGetRQ/RS](#)
- [OTA\\_HotelAvailNotifRQ/RS](#)
- [OTA\\_HotelBookingRuleRQ/RS](#)
- [OTA\\_HotelBookingRuleNotifRQ/RS](#)
- [OTA\\_CacheChangeRQ/RS](#)
- [OTA\\_CacheChangeNotifRQ/RS](#)
- [OTA\\_HotelCommNotifRQ/RS](#)
- [OTA\\_HotelDescriptiveContentNotifRQ/RS](#)
- [OTA\\_HotelDescriptiveInfoRQ/RS](#)
- [OTA\\_HotelEventRQ/RS](#)
- [OTA\\_HotelGetMsgRQ/RS](#)
- [OTA\\_HotelInvAdjustRQ/RS](#)
- [OTA\\_HotelInvBlockRQ/RS](#)
- [OTA\\_HotelInvBlockNotifRQ/RS](#)
- [OTA\\_HotelInvCountRQ/RS](#)
- [OTA\\_HotelInvCountNotifRQ/RS](#)
- [OTA\\_HotelInvNotifRQ/RS](#)
- [OTA\\_HotelInvSyncRQ/RS](#)
- [OTA\\_HotelPostEventReportRQ/RS](#)
- [OTA\\_HotelPostEventNotifRQ/RS](#)
- [OTA\\_HotelRateAmountNotifRQ/RS](#)
- [OTA\\_HotelRatePlanRQ/RS](#)
- [OTA\\_HotelRatePlanNotifRQ/RS](#)
- [OTA\\_HotelResModifyRQ/RS](#)
- [OTA\\_HotelResModifyNotifRQ/RS](#)
- [OTA\\_HotelResNotifRQ/RS](#)
- [OTA\\_HotelResRQ/RS](#)
- [OTA\\_HotelRFP\\_MeetingRQ/RS](#)
- [OTA\\_HotelRFP\\_MeetingNotifRQ/RS](#)
- [OTA\\_HotelRFP\\_TransientRQ/RS](#)
- [OTA\\_HotelRFP\\_TransientNotifRQ/RS](#)
- [OTA\\_HotelRoomingListRQ/RS](#)

- [OTA HotelSearchRQ/RS](#)
- [OTA HotelStatsRQ/RS](#)
- [OTA HotelStatsNotifRQ/RS](#)
- [OTA HotelStayInfoNotifRQ/RS](#)
- [OTA HotelSummaryNotifRQ/RS](#)

## OTA\_HotelAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Availability Request message pair provides the ability to search for hotel products available for booking. 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. Typically, 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. A Hotel Availability Request is used in place of a Hotel Search Request when there is a need to identify availability and rate information in addition to the property list.

This specification addresses the functionality of a traditional request for availability of a property or list of properties. It allows for 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 an AAA or AARP rate, but it may not necessarily offer it at all times, which affects the availability of the rate.

The availability request can be limited to the individual property level, requiring 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. 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. An availability request with search criteria will allow a list of properties to be returned, but with greater property detail, including property info, room/rate info, availability and rules. Due to the amount of information returned for a given property, a Hotel Search Request may be more fitting when only a basic list of properties is required.

The business use cases that the Hotel Availability Request message supports are:

- 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.
- Availability/Multiple Properties based on Search Criteria—determines the availability for multiple properties identified by a Hotel Reference along with additional specified criteria.
- 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.

## OTA\_HotelAvailGetRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Availability Get message pair provide the ability for a booking source to obtain availability status from one or more specified hotel properties.

The Hotel Availability Get request message allows a booking source to search another system for detailed availability. The request message can be limited to an individual property or a collection of properties for a specified date range or it can further specify one or more rate plan(s), room type(s), rate plan/room type combinations, restrictions and revenue management qualifiers.

Based on the criteria specified in the request message, the response message contains the set of availability controls. The Hotel Availability Get response message is similar to the Hotel Availability Notify response in that it contains a complex set of controls that indicate whether the hotel has available inventory which may have surrounding rules for booking a reservation.

## OTA\_HotelAvailNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Availability Notification message 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 ([OTA\\_HotelRateAmountNotifRQ/RS](#)), which communicates the rates that apply to the availability, and a Booking Rule Notification message ([OTA\\_HotelBookingRuleNotifRQ/RS](#)), which 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 hotel guests are 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 super-set 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 *Inventory-BlockCodeType* 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 be allowed to make manual changes while processing bookings during the day, but when full optimization is done, typically during the night, this Boolean attribute deter-

mines whether to retain the changes made. This could be applied to override all status messages and is found in the Status Application Control class.

## OTA\_HotelBookingRuleRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Booking Rule request message provides the capability to request the rules and usage requirements of a rate for a specific room type or booking code for a specific hotel property. This message provides customers pertinent information which is integral to the hotel booking process. The booking rule is generally requested after the availability for a property has been requested and may be done prior to the booking in order to see the usage requirements and restrictions associated with a room type or booking code. This message may also be used to request the rules for an existing reservation by including the hotel confirmation number in the request.

The Hotel Booking Rule response message will provide the detailed rule information.

## OTA\_HotelBookingRuleNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Booking Rule Notification message 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 that communicates the status of availability at a specific hotel property, and the Rate Amount Notification message 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 to which the booking rules apply. 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 (DOW\_Pattern).

This message 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 defining 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 *AbsoluteDeadline* and/or the *AdvanceBooking* 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 *RequiredPayments* 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.

## OTA\_HotelCacheChangeRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Inventory Change message pair facilitates streamlined hotel inventory availability transactions for trading partners that cache hotel rates and availability information from hotel suppliers. The messages are designed to alleviate two significant challenges faced by both hotel suppliers and their trading partners, which are out of synch caches and rising infrastructure/support costs associated with handling a high volume of availability transactions. This new message pair is used in combination with the Hotel Availability (OTA\_HotelAvailRQ/RS) messages. In a typical scenario, two trading partners are involved, where Partner A is a hotel supplier and Partner B is an online booking service.

In step 1, Partner B sends a Hotel Inventory Change request message with their specified criteria that includes a timestamp of the last time they made a Hotel Inventory Change request and the information categories they want to update in their cache, which may be rates data, availability data or both. Depending on the trading partner agreement between the two partners, Partner B may also specify more granular search criteria in their Hotel Inventory Change request, such as chain code, rate plan codes, promotion codes and geographical areas, such as a city. Partner A processes the request message and sends a response message to Partner B with a list of hotel codes that should be updated in Trading Partner B's cache. Note that Partner A may additionally specify rate plan ID's and room types (for each returned hotel code) that should be updated in Partner B's cache. In step 2, Partner B sends an OpenTravel Hotel Availability request message to Partner A that includes a list of inventory items (that were included in the response message returned from Step 1) that require a cache update, and subsequently receives the requested information in the Hotel Availability response message from Partner A.

## OTA\_HotelInvChangeRQ/RS

### *Message Pair Overview*

This message enables a trading partner's system to request hotel inventory items that have availability and/or rate changes and therefore should be updated in a trading partner's cache. There are several methods that may be used to specify rates and availability cache parameters and search criteria in the OTA\_HotelInvChangeRQ message, including categories of hotel inventory information and the time period for the cache information to be returned.

The response message identifies hotels which have availability and/or rate changes and therefore should be updated in a trading partner's cache. The information in the response message will indicate if an OTA\_HotelAvailRQ is required to refresh the trading partner's data cache. All hotel rates and availability information in the OTA\_HotelInvChangeRS message is returned by Hotel Code.

## OTA\_HotelCacheChangeNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Cache Change Notification message set allows a hotel supplier to push their changed hotel rates and availability information to a trading partner that caches their hotel rates and availability information and have the trading partner send an acknowledgment message that they received the information.

Note that this message set is designed to allow the trading partner to receive the changed rates and availability information from the supplier so the trading partner can create a subsequent targeted Hotel Availability Request message that only requests the specific supplier data that has changed and needs to be updated in the trading partner cache.

Hotel Cache Change Notification Request information includes:

- Point of sale information for the requestor
- Unique identifier information that allows trading partners to uniquely identify each Hotel Cache Change Notification Request message for transaction tracing purposes
- Destination system information that specifies a receiving system for the message information
- Cache change information-which specifies what hotel supplier data has changed and should be updated in the trading partners system via a subsequent OTA\_HotelAvailRQ message-including:
  - Chain, brand and hotel codes
  - The start and end date of information that should be updated in the cache for the associated @HotelCode and an optional "change date mask" that indicates if there are changes in all or a portion of the returned date range; including Length of stay that indicates how many days of availability from start date should be shopped; and Full Pattern Length Of Stay to indicate which days from start date should be shopped
  - Other changed product details, including rate plan and promotion details

The Hotel Cache Change Notification Response message may return:

- A success indicator (indicating that the message was successfully received and processed)
- A business warning (indicating that some portion of the message processing generated a business condition)
- An error response (indicating why the message could not be processed)

## OTA\_HotelCommNotifRQ/RS

### *Message Pair Overview*

The Commission Notification message supports the functionality of updating other systems with commissions to be paid. The message pair 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 requesting system will send a report using the OTA\_HotelCommNotifRQ message. The responding system will acknowledge its receipt of that report using the OTA\_HotelCommNotifRS message. All message responses include the request identification. Responses may be returned in any order.

## OTA\_HotelDescriptiveContentNotifRQ/RS

### *Message Pair Overview*

The OpenTravel 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 by category, using codes agreed upon between trading partners to request more detail about a hotel.

The Hotel Descriptive Content interface enables hotel data to be accessed 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 to 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 a 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 his/her 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 responding 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.

## OTA\_HotelDescriptiveInfoRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Descriptive Content Notification message acts as a "push" message—sending information to populate a database. This message set allows an entity to request specific hotel descriptive content information. For example, a travel site wishing to update information like hours of operation would request only the specific information required and be returned only what they requested—like the pool hours or restaurant hours of operation. This message set could also be used for a Request for Information (RFI) for property specific information to help fulfill requests for data from external customers. RFI is an early step in the business negotiation process for either transient or group rates. Customers would then be able to narrow down a list of potential candidate hotels based on RFI responses.

Hotel information does not typically change on a regular schedule, thus the hotel descriptive information message allows the trading partner to request either all information stored in the hotel descriptive content message or a portion therein. The top line information available in a hotel descriptive information message is:

- Hotel Information
- Facility information
- Meeting rooms
- Guest rooms
- Restaurants
- Policies
- Area Information
- Reference points
- Attractions
- Affiliation Information
- Distribution systems
- Brands
- Loyalty programs
- Awards

- Contact Information
- Multimedia

## OTA\_HotelEventRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Event message pair may be used to pass information about an event (meeting, conference, etc.) or group of events to one or more facilities. Examples of events that may be passed include:

- A single-day event held at a facility without an accompanying room block
- A single-property event held at one hotel where the attendees are also staying at the hotel overnight
- A multi-property event held at a convention center where attendees are staying at one or more local hotels
- A series of identical events, e.g. a monthly luncheon held 12 times each year at one property, or a week-long training course held in various locations around the country

This message was created in conjunction with the [APEX initiative](#) of the Convention Industry Council (CIC), an association of meeting professional associations. The information in the message matches the APEX “Event Specification Guide”, which is a human-readable document designed to give the meeting professional both a high-level overview and a detailed description of their event or meeting.

The *EventInfo* element contains information about the event itself, while the *Site* element contains information needed for each facility or location involved in the event. For multi-property (“city-wide”) events, the OTA\_HotelEventRQ message may be sent to each property with only the *Site* element that contains information applicable to the receiving property. A site may be a hotel, convention center, restaurant, meeting facility, country club, etc.

For complex events with multiple breakout sessions and functions, the *EventDayFunction* element is used to provide detailed information about each function at a specific day, time, and location. This information is used to create the “Functional Setup Order” (FSO) for the meeting professional, which is also known as a “Banquet Event Order”, “BEO”, or “Work Order” by hotel operations teams. For events with exhibitor functions, the optional “Exhibitor Function Setup Order” replaces the standard FSO.

The *EventDay* element provides date-neutral event setup information for recurring events. For example, a two-day training session may be defined as having two sessions with lunch and dinner on Day 1, plus two sessions with breakfast and lunch on Day 2. By defining the event in the *EventDay* element relative to “Day 1” and “Day 2”, you can handle recurring events by specifying the start date of each event instance in the *EventInfo* element. The *RoomBlock* element is optional, and it will not be used for events that don’t involve an overnight stay, or for properties at which attendees are not staying overnight (e.g. a convention center). The Hotel Event Response message may echo the request document event requirements that are passed in the message while also providing the requesting system with a confirmation that the original request was received.

## OTA\_HotelGetMsgRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Get Message request/response pair of messages permits a system that normally receives notifications to ask for a re-transmission of a message.

The business model assumes that the requesting system 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 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 requesting system will send a request using the `OTA_HotelGetMsgRQ` message. The receiving system will acknowledge and respond with the `OTA_HotelGetMsgRS` message.

## OTA\_HotelInvBlockRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Inventory Block message pair allows a booking source to search another system for detailed information regarding group inventory. The request message is limited to an individual property for a specified date range and group code identifier. The message can also be used to search on rate plan(s), room type(s), and other group related information.

Based on the criteria specified in the request message, the response message contains the set of inventory controls for the group specified. The Hotel Inventory Block response message is similar to the Hotel Inventory Block Notif request message in that it contains a complex set of controls/rules relating to the group.

## OTA\_HotelInvBlockNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Inventory Block Notification message 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 ([OTA\\_HotelRatePlanNotifRQ/RS](#)) 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 the booking agent's 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 Invento-

ryBlockCodeType 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 elements contain a Reference Place Holder element (RPH) that can be used for indexing to identify a specific Inventory Block in a collection.

The <OTA\_HotelInvBlockNotifRS> returns a response to the Hotel Inventory Block Notification, indicating that the message was successfully 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\_HotelInvCountRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Inventory Count message pair provides the ability for a booking source to request inventory amounts from one or more hotel properties.

The Hotel Inventory Count request message allows a booking source to filter another system for hotel room inventory (e.g. room type codes). The request message can be limited to an individual property or a collection of properties for a specified date range and/or it can further specify room type code(s).

Based on the criteria specified in the request message, the response message contains the set of inventory count controls. The Hotel Inventory Count response message is similar to the Hotel Inventory Count Notify request in that it can contain physical inventory counts or it can be a subset/threshold of actual physical inventory.

## OTA\_HotelInvCountNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Inventory Count Notification message notifies a booking source of the amount of inventory available at a specific hotel property. It allows the Property Management System (PMS) and Central Reservation Systems (CRS) 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 Hotel Inventory Notification messages ([OTA\\_HotelInvNotifRQ/RS](#)) would be used to supply the reservation systems with descriptions of rooms in the hotel, as well as non-room products that are also subject to inventory control. 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 reservation 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, a system may carry a discrete inventory, or an inventory count in association with different rates. The super-set 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, in addition to 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 indicate 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.

## OTA\_HotelInvNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Inventory Notification message 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 database 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. In the exchange of inventory information, which is not always a simple process, the sending system and the receiving system may assign different codes to the same inventory item.

This requires the use of a translation table to link 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 and Rate Plan Notification messages. The Inventory Count Notification establishes the physical inventory count by inventory code, and a Rate Plan Notification 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.

## OTA\_HotelPostEventRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Post Event messages serve to communicate actual history compiled upon completion of one or more events. The messages may be used for city-wide, multi-facility or single facility events.

The Post Event messages may be used to communicate specific logistical details based on real usage from hosting an event at a site or sites, for example:

- to be sent by the meeting professional to potential event site(s) to provide the history of a previous event to complement a request for proposal (RFP);
- to be sent from a site upon completion of the event to the meeting professional for archival purposes and future reference;

- to request details for one or more specific events.

The Post Event messages communicate event data such as event location(s) and event date ranges, revenue, room pick-up, meeting room setup, audiovisual and catering usage.

The OTA\_HotelPostEventRQ/RS message pair is designed to obtain post event history. For example, a national sales office may request data from a number of hotels within their chain to substantiate a request for a future event.

## OTA\_HotelPostEventNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Post Event Notification message pair is designed to push post event data to another system. For example, a meeting facility may send a report containing actual event data to the event organizer following their event.

## OTA\_HotelRateAmountNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Rate Amount Notification message 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 message. 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 Notification 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 the inventory codes related to double bed rooms and king bed rooms.
- 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.
- Rate Plan 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 1st through July 1st.
- 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 au-

thorized 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.

## OTA\_HotelRatePlanRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Rate Plan message pair provides the ability to search for active rate plans and offers. The message is used primarily to ensure that two systems are in sync. It can be used in place of the Hotel Rate Plan Notif message ([OTA\\_HotelRatePlanNotifRQ/RS](#)) when a [pull model](#) is desired.

The Hotel Rate Plan Request message allows one system to query another for detailed rate plan information including availability, pricing, and viewership. By supplying different criteria (e.g., specific rate code, dates, viewership), the request message can limit the number of rate plans and/or offers returned by the response.

The Hotel Rate Plan Response returns a response with the rate plans matching the criteria.

## OTA\_HotelRatePlanNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Rate Plan Notification message 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 the booking agent's 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.

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.

## OTA\_HotelResModifyRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Reservation Modify message set handles the need for a full overlay of the reservation for the purpose of making a change to an existing booking.

The modify message can be sent independently or in conjunction with the [OTA\\_HotelAvailRQ](#) message, which was enhanced to allow a confirmation number to be sent in order to take into account the inventory being held by the existing reservation.

With this model, trading partner A may either send the OTA\_HotelResModifyRQ message following the [OTA\\_HotelAvailRQ/RS](#) or independently send the modify message if confirmation of available inventory is not needed.

## OTA\_HotelResModifyNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Reservation Modification Notification provides a request/response pair of messages to support the functionality of updating other systems with modified reservation data. The message set assumes a push model, with the sending system pushing the data to another system. The sending 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 sending system either has the authority to make the modification to the reservation, or is passing along a message from such a system. The message is a notification of a modification or cancellation to an existing reservation, and does not require the receiving system to confirm the modification, 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 sending system will send a report using the OTA\_HotelResModifyNotifRQ message. The receiving system will acknowledge its receipt of that report using the OTA\_HotelResModifyNotifRS message.

- OTA\_HotelResModifyNotifRQ – Sends a modification of a reservation to another system. All the elements and attributes are optional, unless otherwise stated as required.
- OTA\_HotelResModifyNotifRS – Returns acknowledgment that the modification to 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.

## OTA\_HotelResNotifRQ/RS

### *Message Pair Overview*

The OpenTravel 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 sending system pushing the data to another system. The sending system is typically 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 sending 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 or a modification, or cancellation before that reservation has been ended, 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 initiating system sends a report using the OTA\_HotelResNotifRQ message. The receiving system acknowledges its receipt of the 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 an acknowledgment 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.

## OTA\_HotelResRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Reservation message pair 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 Booker, or other travel service provider 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 Central Reservation Office (CRO) of the hotel chain.

The Hotel Reservation Request message is often preceded by an Availability Request message ([OTA\\_HotelAvailRQ/RS](#)). 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 returned. 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.

There is not a 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 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.

## OTA\_HotelRFP\_MeetingRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Request for Proposal Meeting (RFP - requests for proposals/ RFI - requests for information) message pair expand on the hotel message sets, to further automate a large section of hotel responses to information.

For example, an RFP/RFI is generally sent from a customer, agency, corporation, etc. to the sales team within a hotel(s) or national/regional office(s), who must then manually reply to the message(s) in various ways, such as phone, fax, email, etc. The Hotel RFP/RFI request/response message replaces the inconsistent and highly manual effort of sending the requests and the acknowledgment of receiving the request. This request/response message set is designed for Group and Meeting RFP/RFI.

## OTA\_HotelRFP\_MeetingNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel RFP Meeting Notification message pair provides functionality that allows a hotel to respond to an RFP at a later point in time, versus responding in real-time, and/or provide updated information regarding an original RFP.

The intended use case of this message is the following scenario, after receiving an [OTA\\_HotelRFP\\_MeetingRQ](#), an [OTA\\_HotelRFP\\_MeetingRS](#) message will be returned indicating when and how the hotel will respond to the request. This detailed response uses the [OTA\\_HotelRFP\\_MeetingNotifRQ](#) message.

## OTA\_HotelRFP\_TransientRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel RFP Transient message pair supports the buyer to supplier process of negotiating room rates and amenities for a given contract period. The buyer is the corporation negotiating directly with the supplier (e.g., hotel chains or individual hotels). The buyer can also be a travel agency or consortium securing competitive room rates to offer to corporation and/or the individual business traveler.

The request/response data set will generally equal each other in the first round of distribution. The second round of negotiations, if requested by the buyer, can sometimes be a subset of data from the original request. This data set will include components already defined in the Hotel Descriptive information and the Rate information. The Request is usually a corporation sending requirements to a hotelier requesting a proposal for negotiated room rates. This would be used for an initial request, a renegotiate or an accept/deny.

The response is usually sent by a hotelier. It contains various levels of details based on the Request. In many cases this will be a simple acknowledgment because human intervention is necessary to obtain full details. When these details are available, the hotel will send them in the Notif Request.

## OTA\_HotelRFP\_TransientNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel RFP Transient Notif message pair is typically used by a hotel to push detailed or updated information to an original [OTA\\_HotelRFP\\_TransientRO](#) message. The OTA\_HotelRFP\_TransientNotifRQ is a simple acknowledgment for receipt of the Notif Request.

## OTA\_HotelRoomListRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Room List message pair provides the ability to create, modify, and cancel rooming list reservations. The rooming list message accommodates both the group travel market (meetings and conventions) and the tour market (wholesale).

In the case of group or tour reservations, travelers' reservations are booked into contractually reserved space, instead of publicly available room inventory. Group room blocks are specific to a range of nights and are tied to an event at a location, whereas tour blocks are generally a set number of rooms held at a location for an extended period of time (often an annual basis).

## OTA\_HotelSearchRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Search Request message provides the ability to search for a list of hotel properties that meet specified criteria.

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 (this 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.

The search criteria must be 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. A single search request may specify a set of criterion (within a single criteria) to further narrow the list of properties returned. A single search request may also specify multiple criteria to allow an "either this, or this" scenario.

Property information returned needs to be more than just the name of the hotel chain and the hotel. It should include sufficient information to be able to select a specific property. 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 can be used to perform an availability query on a specific property or multiple properties selected from the list. This functionality is supported today by Central Reservations Systems, which are able to do detailed queries once the requester narrows his/her choice to the property level.

A wide area search can be implemented across system boundaries; outside of a single hotel chain or a GDS. 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.

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 customer 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.

Example:

An OpenTravel 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 OpenTravel offices, distance from Reagan National Airport, proximity to the King Street Metro station, etc. In addition, he/she 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 is 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.

## OTA\_HotelStatsRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Statistics message pairs provide 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 [pull model](#), described in this section, the central system requests a report using the [OTA\\_HotelStatsRQ](#) message. In this message, the report and fiscal date are identified. The receiving system (typically a PMS) responds with the [OTA\\_HotelStatsRS](#) message, which includes the report itself. All messages assume the no-state model, meaning that the sending system will initiate the transaction and expect a response from the receiving system. All message responses include the request identification. Responses may be returned in any order.

In the [push model](#), the sending system will send a report using the [OTA\\_HotelStatsNotifRQ](#) message and the receiving system will acknowledge its receipt of that report using the [OTA\\_HotelStatsNotifRS](#) message as described in this section.

[OTA\\_HotelStatsRQ](#)—Sends a request for a report to another system. All the elements and attributes are optional, unless otherwise stated as required.

[OTA\\_HotelStatsRS](#)—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.

## OTA\_HotelStatsNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Statistics message pairs provide 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](#), described in this section, the sending system sends a report using the OTA\_HotelStatsNotifRQ message and the receiving system will acknowledge its receipt of that report using the OTA\_HotelStatsNotifRS message as described in this section.

In the [pull model](#), the central system requests a report using the [OTA\\_HotelStatsRQ](#) message. In this message, the report and fiscal date are identified. The receiving system (typically a PMS) responds with the [OTA\\_HotelStatsRS](#) message, which includes the report itself. All messages assume the no-state model, meaning that the sending system will initiate the transaction and expect a response from the receiving system. All message responses include the request identification. Responses may be returned in any order.

OTA\_HotelStatsNotifRQ—Sends a report to another system. All the elements and attributes are optional, unless otherwise stated as required.

OTA\_HotelStatsNotifRS—Returns acknowledgment that the report has been successfully received, or includes Warnings from business processing rules or errors if the request did not succeed.

## OTA\_HotelStayInfoNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Stay Information Notification message pair provides 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 sending system will send a report using the OTA\_HotelStayInfoNotifRQ message. The receiving system will acknowledge its receipt of that report using the OTA\_HotelStayInfoNotifRS message. All message responses include the request identification. Responses may be returned in any order.

## OTA\_HotelSummaryNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Hotel Summary Notification message 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 specific 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/RS](#)) 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.

# TRAVEL INSURANCE MESSAGES

## Messages Overview

Travel Insurance is insurance that is intended to cover medical expenses and financial (such as money invested in nonrefundable pre-payments) and other losses incurred while traveling, either within one's own country or internationally.

Temporary travel insurance can usually be arranged at the time of the booking of a trip to cover exactly the duration of that trip, or a more extensive, continuous insurance can be purchased from travel insurance companies, travel agents or directly from travel suppliers such as cruise lines or tour operators.

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.)

OpenTravel's Travel Insurance messages provide the functionality to offer travel insurance for a variety of travelers, including individuals, couples, families and business employees. Business functionality supported through these travel insurance messages is described below.

**SEARCHING FOR TRAVEL INSURANCE:** The Insurance Plan Search message pair ([OTA InsurancePlanSearchRQ/RS](#)) provide a method for insurance agents and 3rd party vendors to obtain a list of available insurance products, as well as perform searches to find insurance products that meet certain requirements (length of coverage, number of travelers, residence/citizenship restrictions, etc) from an insurance company or intermediary.

To perform a basic product search to discover what plans are available, an insurance agent or vendor only needs to pass a form of acceptable identification to the insurance provider (insurance agency number, affiliate program id, etc.) The basic product search response returns a list of insurance products available to the selling agent, along with full descriptions of the products and insurance providers (if requested).

The Insurance Plan Search message pair also provide more advanced search functionality, including:

- Allowing vendors to pass a number of identifying codes to the insurance provider so that the provider can retrieve the plans available to that specific vendor.
- Allowing the searcher to specify preferences on different coverage's, including baggage and medical.
- Allowing the searcher to request coverage limits, including deductibles, policy limits and individual limits.
- Allowing the searcher to specify details for *each* of the trips for which insurance coverage is being searched for, including travel sectors, total trip cost, maximum trip length and covered trips (Multi-Trip plans only.)
- Allowing the searcher to provide traveler information, including birth date, gender, address, relationship to the primary insured, country of citizenship, flight accident coverage (the amount of flight accident protection (FAP) requested by the traveler if offered by the insurance plan), descriptions of covered luggage and equipment, and pre-existing conditions.

**QUOTING TRAVEL INSURANCE:** The Insurance Quote message pair ([OTA InsuranceQuoteRQ/RS](#)) provide request/response messages to an insurance vendor for prices for insurance services. The quote response returns pricing information for specific insurance plans carried by the vendor that meet the customer's requirements.

In addition to price quotations, the insurance quote response also returns to the requester details about the insurance company providing the quote, contact people/numbers if the requester needs more information, any restrictions on the policy, and booking details.

**BOOKING (PURCHASING) TRAVEL INSURANCE:** The Insurance Book message pair resemble 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. The Insurance Book response message returns to the requester the details about the insurance plan(s) booked as well as confirms the information that was sent with the insurance book request message.

## Messages List

- [OTA\\_InsuranceCommonTypes.xsd](#)
- [OTA\\_InsurancePlanSearchRQ/RS](#)
- [OTA\\_InsuranceQuoteRQ/RS](#)
- [OTA\\_InsuranceBookRQ/RS](#)

## OTA\_InsurancePlanSearchRQ/RS

### *Message Pair Overview*

The OpenTravel Insurance Plan Search message pair provide a method for insurance agents and 3rd party vendors to obtain a list of available insurance products, as well as perform searches to find insurance products that meet certain requirements (length of coverage, number of travelers, residence/citizenship restrictions, etc) from an insurance company or intermediary.

To perform a basic product search to discover what plans are available, an insurance agent or vendor only needs to pass a form of acceptable identification to the insurance provider (insurance agency number, affiliate program id, etc). The basic product search response returns a list of insurance products available to the selling agent, along with full descriptions of the products and insurance providers (if requested).

The Insurance Plan Search message also provides more advanced search functionality. In addition to identification, the searching agent may also indicate the length of a trip to be insured, specifics about the traveling individuals, benefit needs, destinations, etc. This more specific product search response would return a group of insurance products that most closely meets the requirements of the search parameters, along with full descriptions of the products and insurance providers (if requested).

## OTA\_InsuranceQuoteRQ/RS

### *Message Pair Overview*

The OpenTravel Insurance Quote message pair is equivalent to a request to an insurance vendor to provide a priced quote for insurance services. The Insurance Quote Response message returns pricing information for specific insurance plans carried by the vendor that meet the customer's requirements, as well as details about the insurance company providing the quote, contact people/numbers if the requester needs more information, any restrictions on the policy, and booking details.

## OTA\_InsuranceBookRQ/RS

### *Message Pair Overview*

The OpenTravel Insurance Book message pair resembles the Insurance Quote request message in structure and contents. The Insurance Book request is contained within the *OTA\_InsuranceBookRQ* root element and contains one or more *PlanForBookRQ* elements.

The Insurance Book Response message returns to the requester the details about the insurance plan(s) booked as well as confirms the information that was sent with the insurance book request message.

# PACKAGE TOUR/ HOLIDAY BOOKING MESSAGES

## Messages Overview

A package holiday typically contains a single “pre-defined” offering with or without a choice of basic elements such as transport and accommodation.

The business model for this concept is that allocated blocks of transport and accommodation inventory for a “season” or “brochure period”—typically “Summer” (May to October) and “Winter” (November to April)—are reserved by a tour operator from the supplier and combined into package holiday inventory items that can be set up and sold from a tour operator’s system.

The use cases covered in this document relate to a tour operator selling packages from their internal inventory stock.

A booking can contain any number of itinerary elements, such as transport, accommodation, car rental, extra products or services, special services, extras, etc. Itinerary or journey elements are distinct by type of service and product, place of delivery, date and time the service is offered and can be individually assigned to one or more of the customers involved in the booking.

The parties typically involved in the current business scenarios are comprised of Travel Agents (on behalf of customers) making inquiries and bookings with Tour Operators who publish brochures describing the package tours. The normal interaction medium is currently videotex which, due to the limited screen display size (80 characters x 25 lines), requires a considerable number of message pairs to achieve a booking. However, it is well established and extensively used, with some operators taking the majority of their bookings this way. The intention behind creating the same functionality using XML is to increase the efficiency of the booking process, extend the reach of the tour operators’ systems and expand the information available to the customer.

This document covers four scenarios—Package Availability, Package Extras, Package Costing and Package Booking:

- The Availability step checks a selected package against the supplier’s system providing full details and item-level prices
- The Extras step identifies additional optional items available with the package
- The Costing step calculates the final price for the customer’s selections
- The Booking step completes the cycle by committing the customer to paying for the holiday and the supplier to providing it

Each of these scenarios can be invoked independently of the others, subject to the necessary minimum information being supplied in the request message.

## Messages List

[OTA PkgCommonTypes.xsd](#)

[OTA PkgAvailRQ/RS](#)

[OTA PkgBookRQ/RS](#)

[OTA PkgExtrasInfoRQ/RS](#)

[OTA PkgCostRQ/RS](#)

[OTA PkgReservation](#)

## OTA\_PkgAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Package Availability Request message is designed to establish whether a specific package is available for a specific date and duration for a given number of customers (who may be subdivided by category e.g., Adult, Child etc.)

If the request is satisfied, the enquirer will be provided with a priced breakdown of the package elements. If the request is not satisfied because one or more elements of the package are not available, the enquirer may be provided with a selection of alternatives for that element.

The business use case (see below) that supports this message identifies a customer or agent (person or system acting on behalf of the customer) who requests the availability status of a specific occurrence of a package. The first step in this use case is for the enquirer to supply the details of the package, the stay and the party composition.

The steps of the use case proceed as follows:

- 1) The customer or agent requests the availability of a specific package for a date and duration for a number of passengers.
- 2) The system returns a priced package summary detailing all possible combinations of facilities (where appropriate).

The data returned at step 2 is used as the basis for the Package Booking Request. Additional data that accompanies the response message may include information, which may affect the inquirer's decision on whether to book the package, e.g., building works, unavailable facilities etc.

Where the supplier system is unable to provide costs for all combinations, it may return a basic priced summary with details of the availability of facilities from which the customer must make a choice and submit a revised request in order to get a full costing.

Possible business processing errors include one or more components of the package cannot satisfy the number of passengers for the date and duration requested. The system may return a list of possible alternative components and if the enquirer chooses one from the list as a substitute the use case will restart from step 1.

## OTA\_PkgBookRQ/RS

### *Message Pair Overview*

The OpenTravel Package Booking messages provide confirmed bookings of a package holiday whose availability may or may not have been checked.

An <ActionType> qualifier is available to modify the default "Book" request to simply return a quotation or make a provisional reservation pending authorization of payment details.

If the "Book" action request by a direct customer is satisfied, the enquirer may be requested to provide contact and payment details. On authorization of the payment details, the enquirer will be provided with a booking reference (and, optionally, invoice details for printing).

The book message pair is used to make the actual booking and respond with enough information to create a holiday invoice and itinerary.

The booking request message can have a number of states, including:

- A straight-forward booking
- An option booking, or,
- Confirmation of an option booking.

The response can therefore be either a complete booking response or an option booking response.

## OTA\_PkgExtrasInfoRQ/RS

### *Message Pair Overview*

The OpenTravel Package Extra Facilities Information message pair is used to provide information on the extra facilities available for a holiday or package.

Extra facilities cover a number of categories of holiday/package add-ons, including flight meals, transfer options, insurance, car hire and upgraded flight seats.

The fact that extra facilities are available for a given package is shown by the *ExtrasInd*="True" being present in the *Package* section of the Package Availability response message.

The selling of Extra facilities is bound by rules that are negotiated between partners, that may include:

- If one person in the party chooses an extra then all the party must have that extra as, e.g. sitting together on a flight must be bought by all of the people on the booking.
- If one person in the party chooses an extra, then everyone in the party must choose from one of a group of extras. For example a number of different flight meals are available and if one person wants a flight meal all the party must choose from one of them.
- The booking of one extra is dependent on another. For example, a child car seat can only be booked if car hire has been selected.

The *Grouping*, *ApplyTo* and *RuleCode* attributes of the extras message are used to describe these rules.

## OTA\_PkgCostRQ/RS

### *Message Pair Overview*

The OpenTravel Package Cost message pair provides final costing for the package and extras selected by the customer. These messages are expected to be used iteratively, so that several cost message pairs can be used as the customer makes selections of options or room types or extras until they are satisfied with their holiday costing. The Package Cost Response message contains full details of the holiday for clarity on what will be booked.

# TOUR/ACTIVITY MESSAGES

## Messages Overview

A tour is defined as a discrete and indivisible (for the purposes of booking) travel component which is often undertaken for “experiential” purposes. Examples include a 14-day hike of Machu Picchu, a white-water rafting trip down the Colorado River, or a 4-hour walking tour of Manhattan.

A tour is organized by a provider and is usually (but not necessarily) accompanied by a guide, to assist the consumer with the experience or journey. Tours in this context are seen as separate from package tours. A package tour is a combination of air and hotel segments (and often other transportation components), which can be seen as divisible components. With a package tour, it is sometimes possible for the purchaser to substitute one component for another (e.g. substituting hotels).

A tour as defined here has no components which can be clearly substituted by the purchaser. *Note however that the provider may create the tour using further sub-components, and may substitute components (e.g. changing rafting provider), but this is not observable to the consumer - just as airlines may change a plane for an air-leg without affecting the flight.*

Often a tour is “book-ended” by flights, transfers and hotel stays, and in that respect can become a [packaged tour](#).

## Messages List

- [OTA\\_TourActivityAvailRQ/RS](#)
- [OTA\\_TourActivityBookRQ/RS](#)
- [OTA\\_TourActivityCancelRQ/RS](#)
- OTA\_TourActivityCommonTypes
- [OTA\\_TourActivityModifyRQ](#)
- [OTA\\_TourActivityResRetrieveRQ](#)
- [OTA\\_TourActivitySearchRQ/RS](#)

## OTA\_TourActivityAvailRQ/RS

### *Message Pair Overview*

The request message is used to request availability for a day tour and/or activity selected by the customer. Note that the tour and/or activity code is known to the availability system, e.g. a unique ID and/or name is being passed in the request in addition to date/ time, customer types and counts.

The response message provides the availability of a requested tour and/or activity. Alternate tour/ activity functionality is supported for tours/ activities that don't meet exact search criteria.

## OTA\_TourActivityBookRQ/RS

### *Message Pair Overview*

The request message is used to request a reservation for a tour and/or activity.

The response message contains information for the confirmation of a reserved tour/ activity.

## OTA\_TourActivityCancelRQ/RS

### *Message Pair Overview*

Cancel a booked tour or activity.

## OTA\_TourActivityModifyRQ

### *Message Pair Overview*

Modify a booked tour or activity. Note that the response message to this request is the OTA\_TourActivityBookRS.

## OTA\_TourActivityResRetrieveRQ

### *Message Pair Overview*

Retrieve a booked tour or activity. Note that the response message to this request is the OTA\_ResRetrieveRS.

## OTA\_TourActivitySearchRQ/RS

### *Message Pair Overview*

Search for tours and/or activities matching specified criteria.

# TRAVEL ITINERARY MESSAGES

## Messages Overview

The OpenTravel Travel Itinerary message (or Passenger Name Record—PNR) is widely used to integrate, manage and service travel content—which includes: Air, Car, Hotels, Rail, and Tour & Cruise.

The following is a list of travel content information traditionally contained within the Travel Itinerary (including but not limited to):

- Personal Traveler Related Information—Name, Address, Phone, etc.
- Booked Travel Segments—Air, Car, Hotel, Tour/Cruise, etc.
- Ticketing, Pricing & Form of Payment Information
- Special Service Request and Remark Details
- Travel Itinerary or PNR Synchronization
- Complete Travel Itinerary Book Request
- Travel Itinerary Update/Modify
- Travel Itinerary Cancel/Ignore

## OTA\_TravelItineraryReadRQ/OTA\_TravelItineraryRS

### *Message Pair Overview*

The OpenTravel Travel Itinerary message(s) are a combination of existing, OpenTravel schema components or fragments) with some additional components included to make one larger XML Schema that includes a Travel Itinerary.

The OTA\_TravelItineraryRS message model builds upon the component assets via loosely coupling associations between them (either hierarchically or by attribute references) to maximize both flexibility and re-usability.

A request to read (retrieve) a Travel Itinerary is issued using the OTA\_TravelItineraryReadRQ message with a unique ID referencing the itinerary.

The response to a read request, OTA\_TravelItineraryRS may contain a <SpecialServices> element at both the itinerary item level and higher itinerary level. Such hierarchy provides a clear means to associate a specific service element with one item or to multiple items.

This contextual association pattern also applies to the pricing elements <ItemPricing> and <ItineraryPricing>, which directly relate to specific item pricing or subtotal pricing of items respectively. Additionally, content which is related to (e.g., not reserved or owned by) a specific OpenTravel travel itinerary message can be referenced via the <AssocItem> element. This element is modeled much like itinerary items where, unique patterns of contextually related service and pricing options can be applied.

# RAIL MESSAGES

## Messages Overview

The OpenTravel Rail messages provide search, availability, booking and reservation retrieval after a booking has been made.

## Messages List

- [OTA\\_RailCommonTypes.xsd](#)
- [OTA\\_RailAvailRQ/RS](#)
- [OTA\\_RailBookRQ/RS](#)
- [OTA\\_RailConfirmBookingRQ/RS](#)
- [OTA\\_RailFareQuoteRQ/RS](#)
- [OTA\\_RailIgnoreBookingRQ/RS](#)
- [OTA\\_RailPaymentRQ/RS](#)
- [OTA\\_RailPriceRQ/RS](#)
- [OTA\\_ReadRQ/ OTA\\_RailResRetrieveSummaryRS](#)
- [OTA\\_ReadRQ/ OTA\\_RailResRetrieveDetailRS](#)
- [OTA\\_RailScheduleRQ/RS](#)
- [OTA\\_RailShopRQ/RS](#)

## OTA\_RailAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Rail Availability message pair provides the ability to request rail services between two station pairs on a specific date, for a specific number of passengers of a particular passenger type. The request can be narrowed to request availability for a specific train number, and can include fares (where applicable) or just scheduled service. Optional request information can include:

- Time / Time Window
- Time / Time Window for return journey
- Connecting cities / Number of connections
- Fare Types
- Discount or Promotional codes that may apply to the fare
- Client Preferences (class of service, sleeper cars)
- Maximum number of responses desired

The Rail Availability Response message contains train availability for a station pair on a specific date. A set of origin and destination options is returned, each of which contains one or more (connecting) trains that serve the city pair. The message is intended to provide all the information necessary to travelers to make informed accurate selections prior to booking. Each option may contain the following:

- Class codes for the class of service or amenities
- Special vendor comments for the city pair
- Whether or not seats can be selected
- On time percentage for the train
- Number and type of passengers
- Fare for each passenger type
- Total Fare for all passengers
- Currency

Due to the wide range of capabilities and requirements in various rail inventory management systems, a number of optional details that relate to the fare rules and restrictions may be returned.

## OTA\_RailBookRQ/RS

### *Message Pair Overview*

The Rail Book request message is used to request that a reservation be created for one or more rail journeys between specified locations on specific date(s) for a specific number and type of passengers.

NOTE that a subsequent OTA\_RailConfirmBookingRQ message is required to acquire a PNR for the reservation, or an OTA\_RailIgnoreBookingRQ message is required to release the reservation (and associated inventory) from the trading partner system.

### **Use of the Messages in Transactional Queue Implementations**

In a transactional queue implementation, one or more provisional booking message(s) are typically added to a provisional booking queue using an OTA\_RailBookRQ/RS message and can be subsequently grouped into a set that can be applied as one transaction for either removal from the queue by a cancellation request using the

OTA\_RailIgnoreBookingRQ/RS message; or a final booking confirmation using the OTA\_RailConfirmBookingRQ/RS message.

## OTA\_RailConfirmBookingRQ/RS

### *Message Pair Overview*

The Rail booking messages support both transactional queued and non-transactional queued provisional booking message implementations. Note that a “provisional” booking is a reservation that may have been processed successfully using the OTA\_RailBookRQ message, but has not been confirmed in the trading partners system as committed or cancelled.

### **Use of the Messages in Transactional Queue Implementations**

In a transactional queue implementation, one or more provisional booking message(s) are typically added to a provisional booking queue using an OTA\_RailBookRQ/RS message and can be subsequently grouped into a set that can be applied as one transaction for either removal from the queue by a cancellation request using the OTA\_RailIgnoreBookingRQ/RS message; or a final booking confirmation using the OTA\_RailConfirmBookingRQ/RS message as shown in the example below. Note in this example the “provisional booking queue” is only shown for the rail supplier.

The OTA\_RailConfirmBookingRS message returns a PNR Locator list which is contained within the <Confirmation> element in the message. Within the PNR Locator list, the rail supplier may return only unique ID(s) for the confirmed reservations -OR- the unique ID (s) and corresponding reservation details. Note that in a transactional queued implementation, *all reservations in a submitted set* must be processed successfully or a <Success> element *will not* be returned in the response message. For example, if there are 5 provisional booking message ID's submitted in the request message and one or more fail during the COMMIT process, then the entire transaction—including all 5 of the provisional booking messages—are NOT considered to be COMMITTED and no PNR information will be returned.

### *OTA\_RailConfirmBookingRQ Request Message Overview*

The Rail Confirm Booking RQ message is the request message to commit one or more provisionally booked messages that have been processed successfully from an OTA\_RailBookRQ message but are not yet confirmed in the trading partners system. One or more unique ID's of provisional bookings are specified in the <UniqueID> element within the message. Successful processing of this message results in PNR creation.

### *OTA\_RailConfirmBookingRS Request Message Overview*

The Rail Confirm Booking RS message is the response message for an OTA\_RailConfirmBookingRQ that commits one or more provisionally booked messages that have been processed successfully from an OTA\_RailBookRQ message but are not yet confirmed in the trading partners system as booked. If the request message has been processed successfully, the Confirmation element in this response message contains the PNR Locator list that contains only unique ID(s) –OR- unique ID(s) with corresponding reservation information.

## OTA\_RailFareQuoteRQ/RS

### *Message Pair Overview*

The OpenTravel Rail Fare Quote retrieve request/response message set is used to request and receive rail fare option information based on:

- Qualifying fare options (including train type, origin/destination locations, arrival/departure times and/or train number and network codes)
- Up to 9 fare basis codes
- Traveler preferences (including accommodation and class codes)

- Rate qualifying information (including travel purpose, promotion codes and other discounts that may affect the fare)
- Passenger category information (by passenger quantity, including gender, occupation, passenger qualifying codes, rate qualifiers and disability requirements)
- Rail Fare Quote Request information includes:
  - Qualifying fare options (including train type, origin/destination locations, arrival/departure times and/or train number and network codes)
  - Up to 9 fare basis codes
  - Traveler preferences (including accommodation and class codes)
- Rate qualifying information (including travel purpose, promotion codes and other discounts that may affect the fare)
- Passenger category information (by passenger quantity, including gender, occupation, passenger qualifying codes, rate qualifiers and disability requirements)

The Rail Fare Quote Response may return:

- A success indicator with rail fare quote details
- A business warning with or without rail fare quote details
- An error response (indicating why the message could not be processed) with no rail fare quote details as the request has not been processed

If successfully processed, the Rail Fare Quote Response information includes:

- Origin and destination location for the entire fare quote
- Fare owner details
- Fares details including:
  - Train type
  - Start and end dates for the fare quote
  - Blackout dates indicator
  - Class codes for inventory controlled items
  - Seat, berth and compartment class details
- Up to 99 origin/destination pairs associated with the fare quote
- Fare basis code(s) and associated fare amount(s)
- Terms and conditions associated with the fare quote
- Notes related to all fare information returned

## OTA\_RailIgnoreBookingRQ/RS

### *Message Pair Overview*

The OpenTravel Rail Book message pair requests a train reservation on a specific rail service provider for travel between two or more stations on specific dates for a specific number and type of passengers in specific classes of service. The optional request information can include:

- Train number
- Departure date and time
- Seat Type, including the direction the seat faces

- Traveler name(s)
- Rate type
- Form of payment
- Delivery address

The Rail Book Response message validates whether or not the booking was successful, provides warning information regarding the booking and itinerary elements including a rail reservation number or “PNR”.

## OTA\_RailPaymentRQ/RS

### *Message Pair Overview*

The Rail Payment message pair is used to submit a payment for a rail reservation. This message is typically used in some combination with the OTA\_RailBookRQ/RS, OTA\_RailConfirmBookingRQ/RS and OTA\_RailIgnoreBookingRQ/RS messages.

### **OTA\_RailPaymentRQ Request Message Overview**

The Rail Payment RQ message is the request message to provide payment for a booked reservation and includes the unique ID associated with the reservation, e.g. a confirmation number and PNR record locator. It allows multiple forms of payment to be used to pay for a single reservation. A variety of payment forms are accepted, including debit and credit cards, exchanged tickets, cash, vouchers, bank drafts and loyalty redemption. In addition to reservation payment information, commission information may also be exchanged, including prepaid amounts, percentages and payment frequency.

### **OTA\_RailPaymentRS Request Message Overview**

The Rail Payment RS message returns the reservation confirmation number that uniquely identifies the reservation, e.g. confirmation number and PNR record locator, if the payment and commission information has been successfully processed. If payment has not been processed successfully, it may return one or more business errors that provide details on why the payment or commission could not be processed.

## OTA\_RailPriceRQ/RS

### *Message Pair Overview*

The OpenTravel Rail Price retrieve request/response message set is used to request and receive rail itinerary pricing information for a specified origin/destination pair –and- specified passenger information, including passenger type, rate qualifier(s) and other passenger details.

Rail Price Request information includes:

- Rail itinerary information, including:
  - Origin/Destination location details
  - Train segment information, including a collection of train segments that provide journey information from origin to destination. Each segment has full details on one specific train segment, including the origin and destination locations for this segment, related travelers' request, etc.

NOTE that a rail pricing request may include Rail Itinerary OR Booking Reference information.

- Booking Reference information - The PNR address of the booking which needs to be priced, optionally including train segments that have already been booked and a fare basis code to be used to price the train segments in the PNR.
- NOTE that a rail pricing request may include Rail Itinerary OR Booking Reference information.

- Passenger information – Including specific passenger information, including passenger type, rate qualifier and other passenger details.
- Rate qualifier information for ALL travelers - Including as travel purpose, promotion codes and rate category that may affect the fare.
- Payment form information - Information about the form(s) of payment to be used to pay for this reservation. The element repeats to allow for multiple forms of payment to be used, if so required.

The Rail Price Response may return:

- A success indicator with rail price details
- A business warning with or without rail price details
- An error response (indicating why the message could not be processed) with no rail price details as the request has not been processed

If successfully processed, the Rail Price Response information includes:

- A priced itinerary with some or all of the following information:
- Pricing information for a rail itinerary - Including the total price for the rail itinerary and pricing details, e.g. summary or detailed basic fare, discounts, service charges, fare adjustments, fees and taxes.
- Journey information from one specific origin to one specific destination, including train segment and pricing details.
- Passenger information - Including occupation, age qualifier, disability, loyalty program, personal identification and contact information.
- Notes associated with the priced itinerary

## OTA\_ReadRQ/ OTA\_RailResRetrieveDetailRS

### *Message Pair Overview*

This message returns detailed information about existing rail reservation(s) **for an exact match on a read request.**

Detailed reservation information in this message:

- Full rail itinerary details, including origin and destination location details (including departure and arrival station, marketing and operating companies, equipment, train identification and classification, inventory class codes, accommodation details, ancillary services, charges and vendor messages).
- Summary Passenger information, including: occupation; gender; age category; ADA requirements; applicable rate qualifiers and discounts; and other information such as height and passenger class.
- Detailed Passenger information, including: birthdate; identification details; name; address; telephone; and loyalty program details.
- Payment rules information, including specific payment rules associated with the reservation.
- Fulfillment details, including information about the company that will (or has) fulfilled/ issued the rail ticket or reservation.

## OTA\_ReadRQ/ OTA\_RailResRetrieveSummaryRS

### *Message Pair Overview*

This message returns a list of summary reservation(s) information when:

- An exact match on a read was not found, OR,
- Summary reservation response information was specifically requested in the OTA\_RailReadRQ/@ResInfoLevel attribute.
- Summary reservation information in this message includes:
  - Reservation booking reference, date and status.
  - Origin and destination summary information.
  - Summary Passenger information, including: occupation; gender; age category; ADA requirements; applicable rate qualifiers and discounts; and other information such as height and passenger class.

## OTA\_RailScheduleRQ/RS

### *Message Pair Overview*

The OpenTravel Rail Schedule message pair facilitates exchanging rail schedule information between a rail supplier and trading partners. The Rail Schedule message pair is typically used as a precursor to Rail Availability (OTA\_RailAvailRQ/RS) messages and may be used to reduce transaction load against a rail rates and availability system when inventory availability and fares are not required as shown in the scenario below.

A rich set of rail schedule information may be exchanged that includes: origin & destination; train segment details for a collection of one or more legs that are defined as a single train number; departure and arrival station details; marketing and operating train company information; the type of equipment used for the train journey; on time information for the train; classes of service codes and special comments for the leg of the journey.

#### **OTA\_RailScheduleRQ Request Message Overview**

The Rail Schedule request message requests rail schedules for a city pair. Optional request information can include: departure time and flexible time windows; connecting cities; and passenger preferences (such as specific trains, cabin and seat types, train types etc.) The flexibility of the message allows the request to be narrowed to schedules for a specific train.

#### **OTA\_RailScheduleRS Response Message Overview**

The Rail Schedule Response message contains rail schedules for a city pair. A set of OriginDestinationOptions are returned, each of which contains one or more (connecting) trains that serve a city pair. For each train, the following information may be returned: origin and destination railway stations; departure and arrival times; station information including days and hours of operation, and station staffing and self-serve technology options. This message contains similar information to the OTA\_RailAvailRS message except it does not contain inventory availability and fares.

## OTA\_RailShopRQ/RS

### *Message Pair Overview*

The message pair, OTA\_RailShopRQ/RS, specifies search criteria for which rail pricing and availability should be searched for. Search criteria may include:

- Information about the outbound (and optional return and connecting locations) between which availability and low fares available are to be checked
- Passenger type(s) and category(s)
- Global preferences for all origin and destination locations for both outbound and return trips (note that these preferences may be overridden elsewhere in the message) and include rail operator, transport modes, accommodations and amenities
- Specific train number/ network code combination or just a network code
- Rate qualifier(s) to specify the type of fares of interest to the traveler, along with any discount number or promotional codes that may affect the fare
- Fare basis code(s) that apply to the whole trip

Additional message functionality includes specifying detail point of sale information for the message initiator.

The Rail Shop response message contains pricing and availability details for the requested search criteria, including origin/ destination location, accommodations and departure/ return dates and times. For each specified O/D pair, train options may be specified that include:

- Origin/Destination location code and optional code context
- Journey segment availability detail information, which may be for a train or a bus segment, and includes:
- Specific inventory-controlled service class code or detailed accommodation information
- Class and passenger type fares
- Other service-related information including reservation class, reservation type, different class codes and auto train vehicle type
- Pricing details may be specified at the origin/ destination pair and/ or segment level, and include basic fare, alternate currency with exchange rate details, terms and conditions, adjustments, discounts, fees, taxes and ancillary charges.

# LOYALTY MESSAGES

## Messages Overview

Many companies in the travel industry offer loyalty programs. In the past, many companies managed their own loyalty programs, but now there are specialized companies whose sole business is to manage loyalty programs.

The OpenTravel Loyalty messages allow the travel industry to communicate with the loyalty industry. All currently defined verticals in the OpenTravel can use this message pair.

Within the loyalty services industry, certificates are frequently granted to consumers for use in purchasing products and/or services from participating businesses. These certificates can be given a variety of loyalty point values and can be issued in a variety of formats (e.g., electronic certificates, paper certificates).

The OpenTravel Loyalty messages provide the following business functionality:

- Sending enrollment information to a loyalty service provider to create a new customer account
- Communicating with a loyalty service provider to generate redemption certificates for customers
- Notifying a loyalty service provider that a customer has redeemed an existing redemption certificate
- Requesting account information for customers enrolled in a loyalty program from a loyalty service provider

## Messages List

- [OTA\\_LoyaltyCommonTypes.xsd](#)
- [OTA\\_LoyaltyAccountCreateRQ/OTA\\_LoyaltyAccountRS](#)
- [OTA\\_LoyaltyCertificateCreateRQ/RS](#)
- [OTA\\_LoyaltyCertificateCreateNotifRQ/RS](#)
- [OTA\\_LoyaltyCertificateRedemptionRQ/RS](#)
- [OTA\\_LoyaltyAccountRS](#)

## OTA\_LoyaltyAccountCreateRQ/OTA\_LoyaltyAccountRS

### *Message Pair Overview*

The OpenTravel Loyalty Account Create message pair allows businesses to send enrollment information to their loyalty service provider to create a new account for one of their customers.

This message pair is based on the profile structure ([OTA\\_Profile.xsd](#)) with extensions for information that pertains only to loyalty account creation.

In the response message the newly created account information (e.g., membership ID) is returned.

## OTA\_LoyaltyCertificateCreateRQ/RS

### *Message Pair Overview*

The OpenTravel Loyalty Certificate Create message pair allows businesses to communicate with their loyalty service provider to generate redemption certificates for their customers. Whereas this “Create” message pair is for on-line information exchange, the companion Loyalty Certificate Create “Notif” message pair is for a [batch “push”](#) transmission.

## OTA\_LoyaltyCertificateCreateNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Loyalty Certificate Create Notif message pair allows businesses to communicate with their loyalty service provider to generate redemption certificates for their customers. Whereas this “Notif” message pair is for a batch [“push”](#) transmission, the companion Loyalty Certificate “Create” message pair is for on-line information exchange.

## OTA\_LoyaltyCertificateRedemptionRQ/RS

### *Message Pair Overview*

The OpenTravel Loyalty Certificate Redemption message pair allows businesses to notify their loyalty service provider, or loyalty service providers to notify businesses, that a customer has redeemed an existing redemption certificate.

## OTA\_ReadRQ/OTA\_LoyaltyAccountRS

### *Message Pair Overview*

The OpenTravel Read Request and Loyalty Account Response message pair allow businesses to request account information for customers enrolled in their loyalty program (from their loyalty service provider.) The generic [OTA\\_ReadRQ](#) message is used to request the loyalty account information. In response, the loyalty service provider returns a message containing the customers account information. The Loyalty Account Response message is based on the [OTA\\_Profile.xsd](#) with extensions for the information that pertains specifically to the loyalty account.

# PROFILE MESSAGES

## Messages Overview

The OpenTravel Profile messages define the detailed business content of a customer profile from a travel industry perspective. This specification provides a set of common messages for transmitting customer profile data that customers provide to travel services to create these profiles, and for the exchange of profile information between travel services within the industry.

A profile includes basic information about a customer or a company for identification as well as financial transactions, memberships and contacts. The profile also defines collections of preferences for specific types of travel including key travel support services such as travel agencies and insurance. Profiles contain information about organizational affiliations, and identify certifications and alliances held by companies in their business relationships. No supplier pricing information is included, nor is data on the travel policies or requirements of an organization addressed in this specification.

The OpenTravel profile messages provide the following business functionality:

- Creating new customer profiles
- Updating customer profiles
- Merging duplicate customer profiles
- Searching for and retrieving customer profiles

## Messages List

- [OTA\\_Profile.xsd](#)
- [OTA\\_ProfileCreateRQ/RS](#)
- [OTA\\_ProfileMergeRQ/RS](#)
- [OTA\\_ProfileModifyRQ/RS](#)
- [OTA\\_ReadRQ/OTA\\_ProfileReadRS](#)

## OTA\_ProfileCreateRQ/RS

### *Message Pair Overview*

The OpenTravel Profile Create message pair define an operation that generates a new record with a unique identifier. The sequence follows these steps:

- Requester sends a Create request along with the initial data, and optionally a unique identifier
- Responder creates a new record and assigns a unique identifier (e.g., a Profile ID or Reservation ID)
- Responder responds with a message providing a unique identifier for the new record created

## OTA\_ProfileMergeRQ/RS

### *Message Pair Overview*

The OpenTravel Profile Merge message pair provides the functionality to merge a number of existing profiles into one and is typically used to remove duplicate customer profiles. A profile merge may consist of up to three different operations on the system:

- An update of the profile which should remain in the system (the consolidated profile).
- Deletion of the profiles which should be merged into the consolidated one (the obsolete profile(s)).
- Updates of any business objects referenced in any of the obsolete profiles, e.g. reservations.

## OTA\_ProfileModifyRQ/RS

### *Message Pair Overview*

The OpenTravel Profile Modify message pair handles the need for a full overlay of the profile for the purpose of making a change to an existing profile. Note that all profile information can be modified.

## OTA\_ReadRQ/OTA\_ProfileReadRS

### *Message Pair Overview*

The OpenTravel Read Request and Profile Read Response message pair handles a profile search operation using a number of search criteria. Examples of search criteria would be: guest/company name, address, e-mail address, telephone number, birth date of a guest, etc. The OpenTravel Profile Read response message is used to convey the response.

# GROUND TRANSPORTATION MESSAGES

## Messages Overview

The OpenTravel Ground Transportation schema is designed to support XML distribution for the personal ground transportation sector (that includes limousine, black car and personal shuttle service) and covers the scope of web services required to implement a modern personal ground transportation reservation system.

The 2011A OpenTravel Publication introduced three new Ground Transportation message sets:

- OTA\_GroundAvailRQ/RS – Check ground transfer service availability
- OTA\_GroundBookRQ/RS – Create a ground transportation service reservation (booking)
- OTA\_GroundResRetrieveRQ/RS – Retrieve a ground transportation service reservation

The OpenTravel ground transportation messages provide the following business functionality:

- Extensive search, availability and booking details including:
  - Availability Request:
    - Type of service-including airport and simple transfers
    - Date and time of required service
    - Pickup, stops and drop-off details
    - Passenger and baggage quantities
    - Special requirements-including child car seats, fuel efficient vehicle, meet and greet, pet friendly and disability equipped vehicle request
    - Rate qualifying information-including promotions and corporate accounts
    - Vehicle type-including limousine and luxury sedan.

Availability Response:

- Service information (including service type, vehicle type, disability equipped vehicle indicator and recommended maximum passengers and baggage)
- Vehicle make and model information
- Rate qualifying information associated with the vehicle and/or ground transportation service
- Restrictions that apply to the vehicle and/or ground transportation service, such as advanced booking, guarantees and cancellation/ modification penalty indicators
- Service charges, including taxes, minimum and maximum charges, and service charge calculation details
- Fees that may be applied in addition to the Ground Transportation Service charges
- Estimated total charges
- Service availability reference ID/number
- Payment rules that are applicable to a specific vehicle or ground service type
- Booking Request/ Response:
  - Cancellation policy details
  - Airport transfer flight information
  - Payment and guarantee information
- Reservation retrieve request information includes:

- Point of sale information for the requester
- A choice between entering a known ground transportation service reservation identifier used to return one exact matching reservation -OR- basic search criteria used to retrieve one or more matching reservations
- A reference ID for the existing ground transportation reservation
- Search criteria that can be used to retrieve an existing ground transportation reservation if the reservation ID/reference number is not known, including passenger name/contact information and rate qualifiers associated with the reservation

## Messages List

- [OTA\\_GroundCommonTypes.xsd](#)
- [OTA\\_GroundAvailRQ/RS](#)
- [OTA\\_GroundBookRQ/RS](#)
- [OTA\\_GroundResRetrieveRQ/RS](#)
- [OTA\\_GroundCancelRQ/RS](#)
- [OTA\\_GroundModifyRQ](#)
- [OTA\\_GroundResRetrieveRQ/RS](#)
- [OTA\\_GroundResNotifRQ/RS](#)

## Implementation Best Practices

1. [OTA\\_GroundAvailRQ/@MaxResponses](#)  
This attribute may be used to specify the maximum amount of search results to be returned by the ground transportation service supplier.
2. [OTA\\_GroundAvailRQ/RateQualifier/@RateCategory](#);  
[OTA\\_GroundAvailRS/GroundServices/GroundService/RateQualifier/@RateCategory](#);  
[OTA\\_GroundBookRS/Reservation/RateQualifier/@RateCategory](#);  
[OTA\\_GroundBookRQ/RateQualifier/@RateCategory](#)  
Due to legal requirements in some states and countries, it is NOT recommended to use this field for published ground transportation service rates.
3. [OTA\\_GroundAvailRS/GroundServices/GroundService/ServiceInfo/@MaximumPassengers](#);  
[OTA\\_GroundAvailRQ/PassengerInfo/@MaximumPassengers](#);  
[OTA\\_GroundBookRQ/PassengerInfo/@MaximumPassengers](#)  
It is recommended that the quantity specified in this field equal the quantity of available passenger seat belts in the Ground Service vehicle.
4. [OTA\\_GroundAvailRS/GroundServices/GroundService/ServiceCharges](#);  
[OTA\\_GroundBookRS/Reservation/ServiceCharges](#)  
Non-base charges, such as gratuity, are typically specified in the Fees element and NOT this element.
5. [OTA\\_GroundAvailRS/GroundServices/GroundService/ServiceCharges/ServiceCharge/TaxAmounts](#);  
[OTA\\_GroundBookRS/Reservation/ServiceCharges/ServiceCharge/TaxAmounts](#)  
It is recommended that taxes that apply to base services charges be specified here (versus in the [ServiceCharge/@Amount](#) attribute) for customer transparency.
6. [OTA\\_GroundAvailRS/GroundServices/GroundService/TotalCharge](#);  
[OTA\\_GroundBookRS/Reservation/TotalCharge](#)

This element typically contains totaled Service Charge and additional Fee amounts.

7. OTA\_GroundAvailRS/GroundServices/GroundService/Reference; OTA\_GroundBookRQ/Reference

This element contains unique availability response information (e.g. ID/number) that may be used as a reference number in a booking request so availability service details do not need to be replicated in the booking request message.

8. OTA\_GroundAvailRS/GroundServices/GroundService/PaymentRules;  
OTA\_GroundBookRQ/PaymentRules

Payment rule information should be specified if there are unique payment rules that apply to a specific ground transportation service or vehicle type.

## OTA\_GroundAvailRQ/RS

### *Message Pair Overview*

The OpenTravel Ground Transportation availability request message queries a suppliers system for service availability that matches specified search criteria. Search criteria options and response information include:

- Type of service-including airport and simple transfers
- Date and time of required service
- Pickup, stops and drop-off details
- Passenger and baggage quantities
- Special requirements-including child car seats, fuel efficient vehicle, meet and greet, pet friendly and disability equipped vehicle request
- Rate qualifying information-including promotions and corporate accounts
- Vehicle type-including limousine and luxury sedan.
- Promotions and corporate rate plans
- The OpenTravel Ground Transportation availability request message queries a suppliers system for service availability that matches specified search criteria. Search criteria options include:
- Type of service-including airport and simple transfers
- Date and time of required service
- Pickup, stops and drop-off details
- Passenger and baggage quantities
- Special requirements-including child car seats, fuel efficient vehicle, meet and greet, pet friendly and disability equipped vehicle request
- Rate qualifying information-including promotions and corporate accounts
- Vehicle type-including limousine and luxury sedan.

The response message includes:

- Service information (including service type, vehicle type, disability equipped vehicle indicator and recommended maximum passengers and baggage)
- Vehicle make and model information
- Rate qualifying information associated with the vehicle and/or ground transportation service
- Restrictions that apply to the vehicle and/or ground transportation service, such as advanced booking, guarantees and cancellation/ modification penalty indicators
- Service charges, including taxes, minimum and maximum charges, and service charge calculation details
- Fees that may be applied in addition to the Ground Transportation Service charges
- Estimated total charges
- Service availability reference ID/number
- Payment rules that are applicable to a specific vehicle or ground service type

## OTA\_GroundResRetrieveRQ/RS

### *Message Pair Overview*

The OpenTravel Ground Transportation reservation retrieve request/response message set is used to retrieve an existing Ground Transportation reservation. Note that this message set allows a reservation to be queried against based on a (valid) unique reservation identifier or reservation search criteria.

Ground Transportation Reservation Retrieve Request information includes:

- Point of sale information for the requester
- A choice between entering a known ground transportation service reservation identifier used to return one exact matching reservation -OR- basic search criteria used to retrieve one or more matching reservations:
- A reference ID for the existing ground transportation reservation
- Search criteria that can be used to retrieve an existing ground transportation reservation if the reservation ID/reference number is not known, including passenger name/contact information and rate qualifiers associated with the reservation
- The Ground Transportation Retrieve Reservation Response may return:
- A success indicator with existing reservation details for an exact match for one reservation (typically when a valid reservation reference ID/number has been specified)
- A success indicator with existing reservation details for a list of (multiple) reservations that match any portion of the criteria specified in the Ground Transportation Reservation Retrieve request message
- A business warning with or without existing reservation details
- An error response (indicating why the message could not be processed) with no reservation details as the reservation retrieve request has not been processed

If successfully processed, the Ground Transportation Retrieve Reservation Response information includes:

- A confirmation number for the ground transportation service reservation
- Information about a primary passenger (who is the contact for the reservation) and additional passengers associated with the reservation
- Information about the locations associated with the reservation, including pickup, interim stop(s) and drop-off details
- Vehicle make and model information associated with the reservation
- Rate qualifying information associated with the vehicle and/or ground transportation service reservation
- Service charges, including taxes, minimum and maximum charges, and service charge calculation details
- Fees that may be applied in addition to the Ground Transportation Service charges
- Estimated total charges
- Restrictions that apply to the vehicle and/or ground transportation service reservation
- Payment rules that are applicable to the ground service reservation
- Comments associated with the reservation

## OTA\_GroundBookRQ/RS

### *Message Pair Overview*

The OpenTravel Ground Transportation Reservation (Booking) message set includes the information required to create one or more ground transportation reservation(s).

Ground Transportation Reservation Request information includes:

- Point of sale information for the requester
- Availability reference ID (A reference ID that was returned in a prior availability response message and may be used to reduce the amount of duplicate information in this request)
- Service locations, that include pickup, stop(s) and drop-off details
- Primary passenger and additional passenger contact information
- General passenger information that may determine the class of vehicle required
- Passenger preferences (including service type, vehicle type, ADA equipped vehicle indicator and recommended maximum passengers and baggage)
- Rate qualifying information associated with the vehicle and/or ground transportation service
- Payment rules that are applicable to a specific vehicle or ground service type
- An option to pass an event manifest (e.g. multiple passenger list) for supplier processing and future car/service assignment
- Payment information that may apply to all reservations as well as individual per-reservation payment information

The OpenTravel Ground Transportation Booking Response message may return:

- A success indicator with reservation booking details
- A business warning with or without reservation booking details
- An error response (indicating why the message could not be processed) with no reservation details as the booking request has not been processed and/or confirmed
- If successfully processed, the Ground Transportation Booking Response information includes:
  - A confirmation number for the successfully booked ground transportation service reservation(s)
  - Information about a primary passenger (who is the contact for this reservation) and additional passengers associated with the reservation
  - Information about the locations associated with the reservation, including pickup, interim stop(s) and drop-off details
  - Vehicle make and model information associated with this reservation
  - Rate qualifying information associated with the vehicle and/or ground transportation service reservation
  - Service charges, including taxes, minimum and maximum charges, and service charge calculation details
  - Fees that may be applied in addition to the Ground Transportation Service charges
  - Estimated total charges
  - Restrictions that apply to this vehicle/ ground transportation service reservation
  - Payment rules that are applicable to this ground service reservation
  - Comments associated with this reservation

## OTA\_GroundCancelRQ/RS

### *Message Pair Overview*

The OpenTravel ground cancellation message pair requests the cancellation of one confirmed ground reservation and responds with cancellation confirmation information.

The Ground Cancel request message requests the cancellation of one confirmed ground reservation. The CancelType (e.g. Cancel or Hold) and a minimum of one unique ID's (typically the booking confirmation number of the reservation to be canceled) are required.

The Ground Cancel response message acknowledges that the ground reservation cancel message was received and processed successfully and/or has returned business warnings. If processed successfully, one or more Cancellation ID's will be returned with optional cancellation rules information and other details about the canceled reservation.

## OTA\_GroundModifyRQ/OTA\_GroundBookRS

### *Message Pair Overview*

The OpenTravel Ground Transportation modify reservation (booking) request message includes the information required to change an existing ground transportation reservation. Note that the response for this modification is the OTA\_GroundBookRS message.

The reference ID that identifies a reservation is required in the request message. This identifies the reservation that is being modified. This message allows for modifications to data within the reservation which include such items as passenger, location and payment information.

The response to the OTA\_GroundModifyRQ message is the OTA\_GroundBookRS message. Reservation response information includes:

- A confirmation number for the successfully modified ground transportation service reservation
- Information about a primary passenger (who is the contact for this reservation) and additional passengers associated with the reservation
- Information about the locations associated with the reservation, including pickup, interim stop(s) and drop-off details
- Vehicle make and model information associated with this reservation
- Rate qualifying information associated with the vehicle and/or ground transportation service reservation
- Service charges, including taxes, minimum and maximum charges, and service charge calculation details
- Fees that may be applied in addition to the Ground Transportation Service charges
- Estimated total charges
- Restrictions that apply to this vehicle/ ground transportation service reservation
- Payment rules that are applicable to this ground service reservation
- Comments associated with this reservation

## OTA\_GroundResRetrieveRQ/RS

### *Message Pair Overview*

The OpenTravel Ground Transportation Reservation Retrieve request is used to specify criteria to retrieve an existing ground transportation reservation, which may include a unique ground transportation reservation identifier (which if valid will return one matching reservation) or general search criteria for a reservation (which if valid may return one or more reservations that matched the search criteria.)

Reservation retrieve request information includes:

- Point of sale information for the requestor
- A choice between entering a known ground transportation service reservation identifier used to return one exact matching reservation -OR- basic search criteria used to retrieve one or more matching reservations
- A reference ID for the existing ground transportation reservation
- Search criteria that can be used to retrieve an existing ground transportation reservation if the reservation ID/reference number is not known, including passenger name/contact information and rate qualifiers associated with the reservation

The Ground Transportation Reservation Retrieve response is used to return either an exact match for one reservation (typically when a valid reservation reference ID/number has been specified) -OR- a list of reservations that match the criteria specified in the Ground Transportation Reservation Retrieve request message. This message may include a success element (and/or business warnings) with reservation details or an error message indicating why the message could not be processed. If reservation search criteria has been specified, the Reservations element may contain multiple matching reservations.

Returned information includes:

- A confirmation number for the ground transportation service reservation
- Information about a primary passenger (who is the contact for the reservation) and additional passengers associated with the reservation
- Information about the locations associated with the reservation, including pickup, interim stop(s) and drop-off details
- Vehicle make and model information associated with the reservation
- Rate qualifying information associated with the vehicle and/or ground transportation service reservation
- Service charges, including taxes, minimum and maximum charges, and service charge calculation details
- Fees that may be applied in addition to the Ground Transportation Service charges
- Estimated total charges
- Restrictions that apply to the vehicle and/or ground transportation service reservation
- Payment rules that are applicable to the ground service reservation
- Comments associated with the reservation

## OTA\_GroundResNotifRQ/RS

### *Message Pair Overview*

The OpenTravel Ground Transportation reservation (booking) notification request message is designed to be used as an unsolicited push notification message between IT systems and/or trading partners. This message includes the information required to create or store details for one ground transportation reservation that may have multiple segments associated with it.

Exchanged reservation information includes:

- Confirmation number(s) for the successfully booked ground transportation service reservation
- Information about the primary passenger (who is the contact for this reservation) and additional passengers associated with the reservation
- Information about the locations associated with the reservation, including pickup, interim stop(s) and drop-off details
- Vehicle make and model information associated with the reservation
- Rate qualifying information associated with the vehicle and/or ground transportation service reservation
- Service charges, including taxes, minimum and maximum charges, and service charge calculation details
- Fees that were applied in addition to the Ground Transportation Service charges
- Estimated total charges
- Restrictions that apply to the ground transportation service reservation
- Payment rules that were applicable to the ground service reservation
- Comments associated with the reservation

The response message simply acknowledges that the ground reservation notification message was received and processed successfully and/or has returned business warnings.

# VACATION RENTAL MESSAGES

## Messages Overview

The OpenTravel Vacation Rental schema is designed to support XML distribution for the ...

The 2013A OpenTravel Publication introduced XXX Vacation Rental message sets:

- TBC

The OpenTravel Vacation Rental messages provide the following business functionality:

- TBC

## Messages List

- OTA\_VacRentalCommonTypes.xsd
- [TBC](#)

## Implementation Best Practices

1. TBC

OTA\_XXXXXXRQ/RS

*Message Pair Overview*

TBC...