



1

2

## 2001C Message Specification

3

(Draft Copy - Public Review)

4

5

6

7

8

### Abstract

9

This document presents the specifications for the Air Availability RQ/RS message pairs, the Package Tours RQ/RS message pairs, and the Golf Tee Times RQ/RS message pairs.

10

11

Section 1 - The Air Working Group presents the Air Availability section with descriptions and definitions of XML data components.

12

13

Section 2 – This section presents the Package Tours RQ/RS message pairs.

14

Section 3 – This section presents the Golf Tee Times RQ/RS message pairs.

15

16

17

18

19

20

21

22

23

24

25

26

**OpenTravel Alliance, Inc.**

27

333 John Carlyle Street, Suite 600

28

Alexandria, Va. 22314 USA

29

+1 703-548-7005

30

Fax +1 703-548-1264

31

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

32

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

33

34

35

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

36  
37  
38  
39  
40  
41  
42

**OpenTravel™ Alliance, Inc.**  
**License Agreement**  
**Copyright, OpenTravel Alliance (OTA), 2001.**  
**All rights reserved, subject to the User License set out below.**

**Authorization to Use Specifications and documentation**

43 **IMPORTANT:** The OpenTravel™ Alliance (“OTA”) Message Specifications (“Specifications”),  
44 whether in paper or electronic format, are made available subject to the terms stated below.  
45 Please read the following carefully as it constitutes a binding Agreement, based on mutual  
46 consideration, on you and your company as licensee (“You”).  
47

48 **1. Documentation.** OTA provides the Specifications for voluntary use by individuals,  
49 partnerships, companies, corporations, organizations, and other entities at their own risk.  
50 The Specifications and any OTA supplied supporting information, data, or software in  
51 whatever medium in connection with the Specifications are referred to collectively as the  
52 “Documentation.”

53 **2. License Granted.**

54 **2.1.** OTA holds all rights, including copyright, in and to the Documentation. OTA  
55 grants to You this perpetual, non-exclusive license to use the Documentation,  
56 subject to the conditions stated below. All use by You of the Documentation is  
57 subject to this Agreement.

58 **2.2.** You may copy, download, and distribute the Documentation and may modify the  
59 Documentation solely to allow for implementation in particular contexts. You may  
60 bundle the Documentation with individual or proprietary software and/or  
61 sublicense it in such configurations.

62 **2.3.** You must reference, in a commercially reasonable location, the fact that the OTA  
63 Documentation is used in connection with any of your products or services, in  
64 part or in whole, whether modified or not, and You may include truthful and  
65 accurate statements about Your relationship with OTA or other use of the  
66 Documentation.

67 **2.4.** However, you may not change or modify the Specification itself, develop a new  
68 standard or specification from the Documentation, or state or imply that any  
69 works based on the Documentation are endorsed or approved by OTA.

70 **2.5.** You must include the OTA copyright notice in connection with any use of the  
71 Documentation. Any uses of the OTA name and trademarks are subject to the  
72 terms of this Agreement and to prior review and approval by OTA.

73 **2.6.** Nothing in this Agreement shall be interpreted as conferring on You or any other  
74 party any other interest in or right to the Documentation. Nothing in this  
75 Agreement shall be interpreted as in any way reducing or limiting OTA’s rights in  
76 the Documentation.

77 **3. LIABILITY LIMITATIONS. THIS AGREEMENT IS SUBJECT TO THE FOLLOWING**  
78 **LIABILITY LIMITATIONS:**

79 **3.1.** ANY DOCUMENTATION PROVIDED PURSUANT TO THIS NON-EXCLUSIVE  
80 LICENSE AGREEMENT IS PROVIDED “AS IS” AND NEITHER OTA NOR ANY  
81 PERSON WHO HAS CONTRIBUTED TO THE CREATION, REVISION, OR  
82 MAINTENANCE OF THE DOCUMENTATION MAKES ANY  
83 REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING  
84 BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY OR FITNESS  
85 FOR ANY PARTICULAR PURPOSE.

86 **3.1.** Neither OTA nor any person who has contributed to the creation, revision or  
87 maintenance of the documentation makes any representations or warranties,  
88 express or implied, that the use of the documentation or software will not infringe

89 any third party copyright, patent, patent application, trademark, trademark  
90 application, or other right.

91 **3.2.** Neither OTA nor any person who has contributed to the creation, revision, or  
92 maintenance of the documentation shall be liable for any direct, indirect, special  
93 or consequential damages or other liability arising from any use of the  
94 documentation or software. You agree not to file a lawsuit, make a claim, or take  
95 any other formal or informal action against OTA based upon Your acquisition,  
96 use, duplication, distribution, or exploitation of the Documentation.

97 **3.3.** The foregoing liability limitations shall apply to and be for the benefit of OTA, any  
98 person who has contributed to the creation, revision or maintenance of the  
99 documentation, and any member of the board of directors, officer, employee,  
100 independent contractor, agent, partner, or joint venturer of OTA or such person.

101 **4. No Update Obligation.** Nothing in this Agreement shall be interpreted as requiring OTA to  
102 provide You with updates, revisions or information about any development or action affecting  
103 the Documentation.

104 **5. No Third Party Beneficiary Rights.** This Agreement shall not create any third party  
105 beneficiary rights.

106 **6. Application to Successors and Assignees.** This Agreement shall apply to the use of the  
107 Documentation by any successor or assignee of the Licensee.

108 **7. Term.** The term of this license is perpetual subject to Your compliance with the terms of this  
109 Agreement, but OTA may terminate this License Agreement immediately upon your breach  
110 of this Agreement and, upon such termination you will cease all use duplication, distribution,  
111 and/or exploitation of the Documentation in any manner.

112 **8. Interpretation and Choice of Forum.** The law of the Commonwealth of Virginia and any  
113 applicable Federal law shall govern this Agreement. Any disputes arising from or relating to  
114 this Agreement shall be resolved in the courts of the Commonwealth of Virginia, including  
115 Federal courts. You consent to the jurisdiction of such courts and agree not to assert before  
116 such courts any objection to proceeding in such forum.

117 **9. Acceptance.** Your acceptance of this License Agreement will be indicated by your  
118 affirmative acquisition, use, duplication, distribution, or other exploitation of the  
119 Documentation. If you do not agree to these terms, please cease all use of the  
120 Documentation now.

121 **10. Questions.** Questions about the Agreement should be directed to [www.opentravel.org](http://www.opentravel.org).

# Table of Contents

122

123

124

125	<b>Section 1. Flight Availability and Point Of Sale .....</b>	<b>5</b>
126	1.1. Existing EDI-based systems .....	5
127	1.2. XML in place of EDI.....	5
128	1.3. Flight Availability.....	6
129	1.4. Root verbs.....	6
130	1.5. <OTA_AirAvailRQ> .....	6
131	1.6. <OTA_AirAvailRS>.....	9
132	1.7. Sample Messages .....	11
133	<b>Section 2. – Package Tours/Holiday Bookings .....</b>	<b>15</b>
134	2.0. Package Availability Request.....	15
135	2.1. Package Availability Use Case.....	16
136	2.2. Package Availability Messages.....	16
137	2.3. Create Booking.....	26
138	2.4. Create Booking Use Case.....	26
139	2.5. Create Booking Messages.....	27
140	<b>Section 3. – Golf Tee Times.....</b>	<b>33</b>
141	3.1. Golf Course Search Request Message (OTA_CourseSearchRQ) .....	34
142	3.2. Golf Course Search Response Message (OTA_CourseSearchRS) .....	34
143	3.3. Course Availability Request Message (OTA_CourseAvailRQ).....	35
144	3.4. Course Availability Response Message (OTA_CourseAvailRS) .....	37
145	3.5. Course Tee Time Reservation Request Message (OTA_CourseResRQ) .....	39
146	3.6. Course Tee Time Reservation Response Message (OTA_CourseResRS) .....	41
147	3.7. Sample XML Message for Course Search Request.....	41
148	3.8. Sample XML Message for Course Search Response .....	42
149	3.9. Sample XML Message for Course Availability Request .....	43
150	3.10. Sample XML Message for Course Availability Response .....	43
151	3.11. Sample XML Message for Course Reservation Request.....	43
152	3.12. Sample XML Message for Course Reservation Response .....	44
153	3.13. Revision History .....	44

154 **Section 1. Flight Availability and Point Of Sale**

155  
156  
157  
158  
159  
160  
161  
162

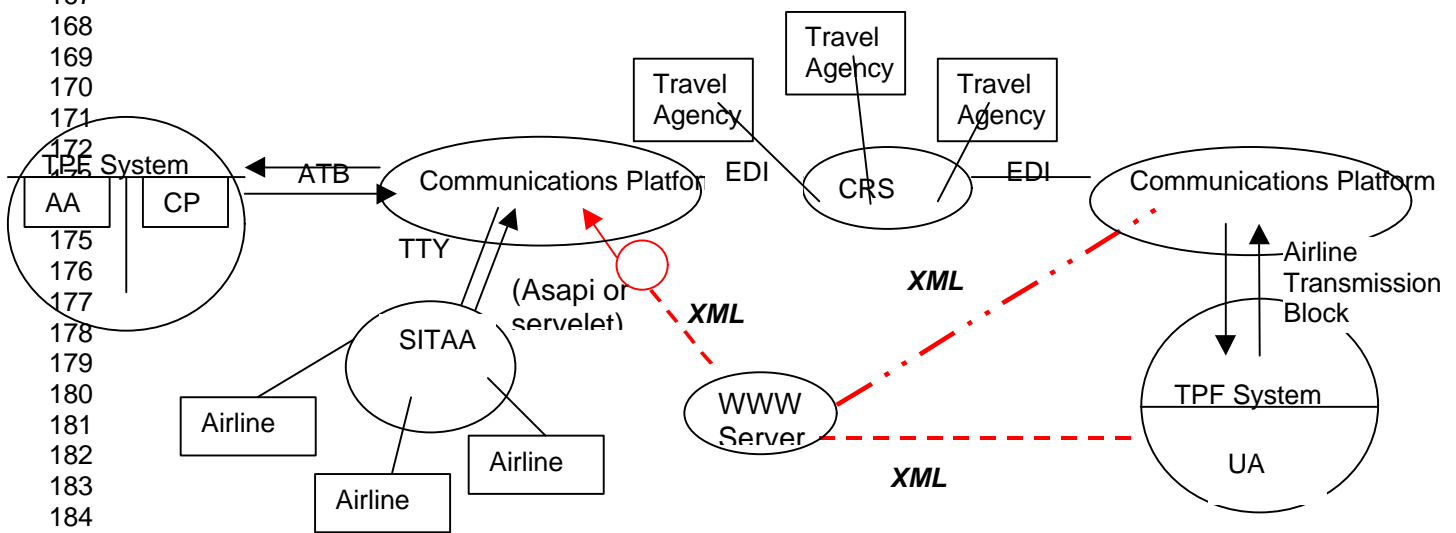
This specification addresses the structure, elements, and context of requests and responses for airline flight availability and point of sale information. It is intended to provide a simple interface to Air Reservation Systems for querying availability in 3 scenarios:

1. General availability request for multiple available flight options from multiple carriers.
2. Specific availability request for multiple flight options from a single carrier.
3. Specific availability request for a specific flight from a specific carrier.

163 **1.1. Existing EDI-based systems**

164  
165  
166  
167

It is presumed that airlines currently use EDI as the “common language” to exchange data. The example below shows how XML can be used in place of EDI when two suppliers communicate:



168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185

A partitioned TPF System serving multiple airlines (AA and CP here) sends an airline transmission block, which is proprietary to each carrier system. These transmission blocks carry messages in uncommon formats. The communications platform acts as a translator, and sends messages to other entities into a commonly agreed upon language with specific structures and business rules. This language, currently in use by the airline industry, is called EDI.

186  
187  
188  
189  
190  
191  
192

Translated messages move from the communications platform to external entities such as CRSSs and SITA, who in turn distribute them to multiple airlines. The diagram above shows UA as the eventual supplier recipient; the UA communications platform translates the EDI message into the language used by its TPF system.

193  
194  
195  
196  
197  
198  
199  
200  
201

EDI is currently dependent upon dedicated lines and third-party distributors, both of which add expense to the data transmission process. XML enables messages to be transmitted from suppliers to travel agents, other suppliers and third parties over the Internet, enabling more flexibility in communication and lowering communications costs.

202 **1.2. XML in place of EDI**

203  
204  
205

The ideal concept would be to send ATBs directly from a TPF system to the communications platform using XML, which would not require translation at the communications platform level. If a carrier is not currently supporting EDI, then creating the ability to send and receive XML

206 messages directly into its TPF system could fairly straightforward; some airlines are currently  
207 developing this capability.

208  
209 A faster solution requiring less investment for EDI-translating systems would be to add an  
210 additional block to the communications platform to convert airline transmission blocks into XML.  
211 If a carrier prefers to outsource its XML services, the communications layer can continue to  
212 export its data blocks in EDI, but can designate the XML outsourcer as a recipient. The  
213 outsourcer could then translate and distribute the XML messages as necessary. In all three  
214 cases, the XML would be sent to a www server, which in turn sends the XML message to an  
215 asapi or servlet that sends the XML-formatted messages to the participating carrier for  
216 translation back into the TPF system.

217  
218 Systems may be slowed down with each additional translation layer, so minimization of stops  
219 along the way is ideal, and can possibly be achieved by reaching agreements with trading  
220 partners to establish direct Internet links.

221

### 222 **1.3. Flight Availability**

223

224 The airline flight availability messages listed in this section are designed to allow a system to  
225 query a supplier system using XML as the common language rather than EDI. The querying  
226 system identifies itself, requests that the supplier system search for airline flights and flight  
227 services, and then the supplier may return which flights meet the defined criteria. Presumably, an  
228 availability request is made with the intent to ultimately book a flight reservation.

229

### 230 **1.4. Root verbs**

231

232 The single OTA\_AirAvailRQ schema definition supports three types of flight availability requests:  
233 specific flight, specific airline, and multiple airline. A specific flight request seeks availability  
234 information on particular flights on particular dates. A specific airline request is more general,  
235 seeking availability for a journey that might occur during a range of times or dates. A multiple  
236 airline request is one that combines one or more dates or segments that are not normally priced  
237 for sale as a unit.

238

239 The following two schemas are defined:

240

241 OTA\_AirAvailRQ – The request message. Seeks availability of flights on particular date.

242 OTA\_AirAvailRS – The response message. Determines availability within the constraints of  
243 specified criteria for flights on particular dates.

244

### 245 **1.5. <OTA\_AirAvailRQ>**

246

247 This request seeks availability of flights on particular dates for an origin–destination pair from one  
248 or more suppliers. The request consists of the following child elements:

249

Element	MinOccurs	MaxOccurs
“POS”	1	1
“OriginDestinationInformation”	1	1
“SpecificRequestInformation”	0	1
“TravelPreferences”	0	1
“TravellInformation”	0	1

250

251 The OTA\_AirAvailRQ element contains 2 attributes:  
 252

Attribute	Type	Use	Choices	Default
"DesiredOptionQuantity"	xs:positiveInteger	optional		10
"DirectFlightsOnly"	xs:positiveInteger	optional		False

253 **1.5.1. <POS>**

254 This is the standard OTA point of sale information element. It is detailed in the 2001C  
 255 Specification schema fragments.

256 **1.5.2. <OriginDestinationInformation>**

257 This element contains the information relating to the travel origin and destination, plus information  
 258 on times and dates of travel and connection points. It may contain the following child elements:  
 259

Element	MinOccurs	MaxOccurs
"OriginLocation"	1	1
"DestinationLocation"	1	1
"TravelDateTime"	1	1
"Connection"	0	1

260  
 261

262 The OriginDestinationInformation element has no attributes.

263 **1.5.3. <OriginLocation>, <DestinationLocation>**

264 OriginLocation and DestinationLocation are elements of type LocationType, a complex Type  
 265 based on xs:string which has 2 attributes:  
 266

Attribute	Type	Use	Choices	Default
"LocationCode"	xs:string	Required		
"CodeContext"	xs:string	Optional	"IATA", "Internal"	"IATA"

267

268 **1.5.4. <TravelDateTime>**

269 TravelDateTime is an element of type TravelDateTime, a complexType with 4 attributes:

Attribute	Type	Use	Choices	Default
"TravelDate"	xs:dateTime	Required		
"TravelTimeIndicator"	xs:string	Optional	"DepartureTime", "ArrivalTime"	"DepartureTime"
"TimeWindowBefore"	xs:duration	Optional		
"TimeWindowAfter"	xs:duration	Optional		

270

271 **1.5.5. <Connection>**

272 Connection is an element of type ConnectionType. It has one child element, ConnectionLocation  
 273 which is of type LocationType:

Element	MinOccurs	MaxOccurs
"ConnectionLocation"	1	1

274

275 Connection has a single attribute which indicates the preference level of the requested  
 276 connection:

Attribute	Type	Use	Choices	Default
"PreferLevel"	PreferLevelType	Optional	"Only",	"Preferred"

“Acceptable”,  
“Preferred”

277

### 278 1.5.6. <SpecificRequestInformation>

279 The SpecificRequestInformation element is used to restrict the request to a specific airline,  
280 specific flight or specific booking class. It has 3 child elements and no attributes:

281

Element	MinOccurs	MaxOccurs
“FlightNumber”	0	1
“Airline”	0	1
“BookingClassPref”	0	1

282

283 FlightNumber is of type xs:positiveInteger.

284

### 285 1.5.7. <Airline>

286 Airline is an element of type CompanyName, a complex type based on xs:string with 2 attributes:

Attribute	Type	Use	Choices	Default
“CompanyCode”	xs:string	required		
“CodeContext”	xs:string	optional	“IATA”, “internal”	“IATA”

287

### 288 1.5.8. <BookingClassPref>

289 BookingClassPref is an element with 2 attributes:

Attribute	Type	Use	Choices	Default
“BookingClassCode”	ResBookDesigC odeType	required	Single alphabet character	
“CodeContext”	PreferLevelType	optional	“Only”, “Unacceptable”, “Prefeered”	“Preferred”

290

### 291 1.5.9. <TravelPreferences>

292 The TravelPreferences element holds preference information related to the flight search. The  
293 child elements are based on the Air Preferences section of the OTA 2001A Profiles Specification:

294

Element	MinOccurs	MaxOccurs
“VendorPref”	0	Unbounded
“FlightTypePref”	0	Unbounded
“AirFareRestrictPref”	0	Unbounded
“AirEquipPref”	0	Unbounded
“AirCabinPref”	0	Unbounded
“AirSrvcClassPref”	0	Unbounded
“TicketDistribPref”	0	Unbounded

295

296 These elements are documented fully in the 2001A Specification.

297

### 298 1.5.10. <TravelerInformation>

299 The TravelerInformation element allows the type and number of proposed travelers to be included  
300 in the Availability Request message. It contains a single element and no attributes:

301

Element	MinOccurs	MaxOccurs
"PassengerTypeQuantity"	1	Unbounded

302

### 303 1.5.11. <PassengerTypeQuantity>

304 PassengerTypeQuantity is an element with 2 attributes used to specify a Passenger Type Code  
 305 (PTC), plus the number of passengers who match that PTC:

306

Attribute	Type	Use	Choices	Default
"PassengerTypeCode"	ThreeLetterCodeType	Required		
"Quantity"	xs:positiveInteger	Required		

307

## 308 1.6. <OTA\_AirAvailRS>

309

310 This response message details availability of flights on particular dates for an origin–destination  
 311 pair from one or more suppliers. The response consists of the following child elements:

312

Element	MinOccurs	MaxOccurs
"OTA_SucceedDef"	0	1
"OTA_FailureDef"	0	1
"OriginDestinationOption"	0	Unbounded

313

314 The OTA\_AirAvailRS element has no attributes. Either OTA\_SucceedDef or OTA\_FailureDef  
 315 must be present. They are defined in .....

316

### 317 1.6.1. <OriginDestinationOption>

318 An OriginDestinationOption element is the container for a set of connected flight segments that  
 319 originate at the OriginLocation specified in the OTA\_AirAvailRQ message and terminate at the  
 320 DestinationLocation.

321

322 It has no attributes and 1 child element:

323

Element	MinOccurs	MaxOccurs
"FlightSegment"	1	Unbounded

324

### 325 1.6.2. <FlightSegment>

326 The FlightSegment Element represents a single flight with a single flight number. It can include  
 327 stopovers.

328 It has the following child elements:

Element	MinOccurs	MaxOccurs
"DepartureAirport"	1	1
"ArrivalAirport"	1	1
"DepartureDateTime"	1	1
"ArrivalDateTime"	1	1
"Cabin"	0	Unbounded
"MarketingAirline"	1	1
"OperatingAirline"	0	1
"Equipment"	0	1
"Comment"	0	Unbounded

329

330 It has the following attributes:

Attribute	Type	Use	Choices	Default
"SequenceNumber"	xs:integer	Required		
"FlightNumber"	xs:positiveInteger	Required		
"StopQuantity"	xs:nonNegativeInteger	Optional		
"JourneyDuration"	xs:duration	Optional		
"SmokingInd"	SmokingIndType	Optional	"Yes", "No"	"No"
"OnTimeRate"	OnTimeRateType	Optional	Percentage(1 to 90)	
"Ticket"	TicketType	Optional	"Paper", "eTicket"	"Paper"

331

332 *SequenceNumber* is a reference place holder to identify this particular flight segment and must be  
 333 unique within this message.

334

### 335 1.6.3. <DepartureAirport>,<ArrivalAirport>

336

337 DepartureAirport and ArrivalAirport are elements of type LocationType, a complex Type based on  
 338 xs:string which has 2 attributes:

### 339 1.6.4. <DepartureDateTime>, <ArrivalDateTime>

340 DepartureDateTime and ArrivalDateTime are elements of type xs:dateTime. They have no  
 341 attributes.

### 342 1.6.5. <Cabin>

343 The Cabin element is a construct for holding a set of Booking classes, plus the seat availability for  
 344 each class within a particular aircraft cabin (First, Business or Economy). It also contains  
 345 attributes to identify the cabin and give meal information for the cabin.

346

Element	MinOccurs	MaxOccurs
"BookingClassAvail"	0	Unbounded

347

Attribute	Type	Use	Choices	Default
"CabinType"	CabinType	Required	"First", "Business", "Economy"	
"MealCode"	MealCodeType	Optional	Single Alphabet Character	

348

349

### 350 1.6.6. <MarketingAirline>, <OperatingAirline>

351 MarketingAirline and OperatingAirline are both elements of type CompanyName, a complex type  
 352 based on xs:string with 2 attributes:

353

Attribute	Type	Use	Choices	Default
"CompanyCode"	xs:string	Required		
"CodeContext"	xs:string	Optional	"IATA", "Internal"	"IATA"

354

355 Company Code in both cases should be a 2 letter airline code.

### 356 1.6.7. <Equipment>

357 Equipment is an element of type EquipmentType, a complex type based on xs:string with 1  
 358 attribute:

359

Attribute	Type	Use	Choices	Default
"AirEquipType"	xs:string	Required	3 letter aircraft type	

360

### 361 1.6.8. <Comment>

362 Comment is an element of type xs:string for storing any text information relating to this flight  
363 segment.

364

### 365 1.6.9. <BookingClassAvail>

366 BookingClassAvail is an element of type BookingClassAvailabilityType, a complex type based on  
367 xs:string with 2 attributes:

Attribute	Type	Use	Choices	Default
"ResBookDesigCode"	ResBookDesigCode	Required	Single Alphabet Character	
"Availability"	xs:string	Required		

368

369 While Availability will generally be a numeric value indicating the number of seats available for a  
370 specific booking class, it is defined as a string to allow non-numeric indicators to be included such  
371 as waitlisted, available, etc. It is envisaged that separate attributes will be defined for these in the  
372 future.

## 373 1.7. Sample Messages

### 374 1.7.1. OTA\_AirAvailRQ

375

#### 376 1.7.1.1. General Availability Request

377

378

379

380 This is a general availability request for flights between London and Los Angeles for 2 adults and  
381 1 child.

382

```
383 <OTA_AirAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
384 xsi:noNamespaceSchemaLocation="OTA_AirAvailRQ.xsd" DesiredOptionQuantity="10" DirectFlightsOnly="false">
385   <POS>
386     <Source AgentSine="BSIA1234PM" PseudoCityCode="2U8" ISOCountry="US" ISOCurrency="USD">
387       <Uniquelid URL=" http://provider1.org/OTAEngine/" Type=" OTA_AirAvailRQ" Id="ID0507G4325"
388 Instance="2001-06-03T13:09:21"/>
389     </Source>
390   </POS>
391   <OriginDestinationInformation>
392     <OriginLocation LocationCode="LON" CodeContext="IATA"/>
393     <DestinationLocation LocationCode="LAX" CodeContext="IATA"/>
394     <TravelDateTime TravelDate="2001-09-11T09:00:00" TravelTimeIndicator="DepartureTime"
395 TimeWindowBefore="P3D" TimeWindowAfter="P3D"/>
396   </OriginDestinationInformation>
397   <TravelPreferences SmokingInd="No" MaxStopsQuantity="1">
398     <VendorPref PreferLevel="Preferred">
399       <VendorName CompanyCode="BA" CodeContext="IATA"/>
400     </VendorPref>
401     <FlightTypePref PreferLevel="Preferred">Direct</FlightTypePref>
402     <AirEquipPref PreferLevel="Preferred" AirEquipType="757"/>
403     <AirCabinPref PreferLevel="Preferred" CabinType="Economy"/>
404     <TicketDistribPref PreferLevel="Only" DistribType="Mail" TicketTime="P5D"/>
405   </TravelPreferences>
```

```

406     <TravelerInformation>
407         <PassengerTypeQuantity PassengerTypeCode="ADT" Quantity="2"/>
408         <PassengerTypeQuantity PassengerTypeCode="CNN" Quantity="1"/>
409     </TravelerInformation>
410 </OTA_AirAvailRQ>
411

```

### 1.7.1.2. Specific Airline Availability Request

This is a specific airline availability request for flights between Miami and Los Angeles on American Airlines for 1 adult with a preferred connection in Dallas.

```

417 <OTA_AirAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
418 xsi:noNamespaceSchemaLocation="OTA_AirAvailRQ.xsd" DesiredOptionQuantity="10" DirectFlightsOnly="false">
419     <POS>
420         <Source AgentSine="BSIA1234PM" PseudoCityCode="2U8" ISOCountry="US" ISOCurrency="USD"
421 CarrierCode="AA" AirportCode="IAD" FirstDepartPoint="IAD" ERSPUserId="12345432">
422             <Uniqueld URL=" http://provider1.org/OTAEngine/" Type=" OTA_AirAvailRQ" Id="ID0507G4325"
423 Instance="2001-06-03T13:09:21"/>
424             <BookingChannel Type="GDS" Primary="1"/>
425         </Source>
426     </POS>
427     <OriginDestinationInformation>
428         <OriginLocation LocationCode="MIA" CodeContext="IATA"/>
429         <DestinationLocation LocationCode="LAX" CodeContext="IATA"/>
430         <TravelDateTime TravelDate="2001-09-11T09:00:00" TravelTimeIndicator="DepartureTime"/>
431         <Connection PreferLevel="Preferred">
432             <ConnectionLocation LocationCode="DFW" CodeContext="IATA"/>
433         </Connection>
434     </OriginDestinationInformation>
435     <SpecificRequestInformation>
436         <Airline CompanyCode="AA" CodeContext="IATA"/>
437     </SpecificRequestInformation>
438
439     <TravelerInformation>
440         <PassengerTypeQuantity PassengerTypeCode="ADT" Quantity="1"/>
441     </TravelerInformation>
442 </OTA_AirAvailRQ>
443

```

### 1.7.1.3. Specific Flight Availability Request

This is a specific flight request for availability in booking class Q on American Airlines flight 123 between JFK and Los Angeles for 1 adult.

```

449 <OTA_AirAvailRQ xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
450 xsi:noNamespaceSchemaLocation="OTA_AirAvailRQ.xsd">
451     <POS>
452         <Source AgentSine="BSIA1234PM" PseudoCityCode="2U8" ISOCountry="US" ISOCurrency="USD"
453 CarrierCode="AA" >
454             <Uniqueld URL=" http://provider1.org/OTAEngine/" Type=" OTA_AirAvailRQ" Id="ID0507G4325"
455 Instance="2001-06-03T13:09:21"/>
456         </Source>
457     </POS>
458     <OriginDestinationInformation>
459         <OriginLocation LocationCode="JFK" CodeContext="IATA"/>
460         <DestinationLocation LocationCode="LAX" CodeContext="IATA"/>
461         <TravelDateTime TravelDate="2001-09-11T09:00:00"/>
462     </OriginDestinationInformation>
463     <SpecificRequestInformation>
464         <Airline CompanyCode="AA" CodeContext="IATA"/>
465         <FlightNumber>123</FlightNumber>
466         <BookingClassPref BookingClassCode="Q" PreferLevel="Only"/>
467     </SpecificRequestInformation>
468     <TravelerInformation>
469         <PassengerTypeQuantity PassengerTypeCode="ADT" Quantity="1"/>

```

```
470     </TravelerInformation>
471 </OTA_AirAvailRQ>
472
```

## 473 1.7.2. OTA\_AirAvailRS

474

### 475 1.7.2.1. Successful Response

476

477 This is a successful response to a general availability request. The specific requests have exactly  
478 the same format, but would not return options outside the restrictions specified in the request.

479

```
480 <?xml version="1.0" encoding="UTF-8"?>
481 <OTA_AirAvailRS xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
482 xsi:noNamespaceSchemaLocation="OTA_AirAvailRS.xsd">
483   <OriginDestinationOption>
484     <FlightSegment SequenceNumber="1" FlightNumber="212" StopQuantity="0"
485 JourneyDuration="P0Y0M0DT12H30M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">
486       <DepartureAirport LocationCode="LHR" CodeContext="IATA"/>
487       <ArrivalAirport LocationCode="LAX" CodeContext="IATA"/>
488       <DepartureDateTime>2001-09-11T10:30:00+00:00</DepartureDateTime>
489       <ArrivalDateTime>2001-09-11T11:14:00+08:00</ArrivalDateTime>
490       <Cabin CabinType="First" MealCode="L">
491         <BookingClassAvail ResBookDesigCode="F" Availability="9"/>
492       </Cabin>
493       <Cabin CabinType="Business" MealCode="L">
494         <BookingClassAvail ResBookDesigCode="C" Availability="7"/>
495         <BookingClassAvail ResBookDesigCode="J" Availability="7"/>
496       </Cabin>
497       <Cabin CabinType="Economy" MealCode="S">
498         <BookingClassAvail ResBookDesigCode="Q" Availability="0"/>
499         <BookingClassAvail ResBookDesigCode="V" Availability="4"/>
500         <BookingClassAvail ResBookDesigCode="N" Availability="1"/>
501         <BookingClassAvail ResBookDesigCode="Y" Availability="7"/>
502       </Cabin>
503       <MarketingAirline CompanyCode="BA" CodeContext="IATA"/>
504       <Equipment AirEquipType="744"/>
505     </FlightSegment>
506   </OriginDestinationOption>
507   <OriginDestinationOption>
508     <FlightSegment SequenceNumber="2" FlightNumber="112" StopQuantity="0"
509 JourneyDuration="P0Y0M0DT12H00M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">
510       <DepartureAirport LocationCode="LHR" CodeContext="IATA"/>
511       <ArrivalAirport LocationCode="LAX" CodeContext="IATA"/>
512       <DepartureDateTime>2001-09-11T10:20:00+00:00</DepartureDateTime>
513       <ArrivalDateTime>2001-09-11T11:13:50+08:00</ArrivalDateTime>
514       <Cabin CabinType="First" MealCode="L">
515         <BookingClassAvail ResBookDesigCode="F" Availability="6"/>
516       </Cabin>
517       <Cabin CabinType="Business" MealCode="L">
518         <BookingClassAvail ResBookDesigCode="C" Availability="4"/>
519         <BookingClassAvail ResBookDesigCode="J" Availability="4"/>
520       </Cabin>
521       <Cabin CabinType="Economy" MealCode="S">
522         <BookingClassAvail ResBookDesigCode="Q" Availability="4"/>
523         <BookingClassAvail ResBookDesigCode="V" Availability="4"/>
524         <BookingClassAvail ResBookDesigCode="N" Availability="2"/>
525         <BookingClassAvail ResBookDesigCode="Y" Availability="6"/>
526       </Cabin>
527       <MarketingAirline CompanyCode="UA" CodeContext="IATA"/>
528       <Equipment AirEquipType="777"/>
529     </FlightSegment>
530   </OriginDestinationOption>
531 </OriginDestinationOption>
```

532 <FlightSegment SequenceNumber="3" FlightNumber="22" StopQuantity="0"  
533 JourneyDuration="P0Y0M0DT6H30M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">  
534 <DepartureAirport LocationCode="LHR" CodeContext="IATA"/>  
535 <ArrivalAirport LocationCode="JFK" CodeContext="IATA"/>  
536 <DepartureDateTime>2001-09-11T09:30:00+00:00</DepartureDateTime>  
537 <ArrivalDateTime>2001-09-11T11:35:00+05:00</ArrivalDateTime>  
538 <Cabin CabinType="First" MealCode="L">  
539 <BookingClassAvail ResBookDesigCode="F" Availability="9"/>  
540 </Cabin>  
541 <Cabin CabinType="Business" MealCode="L">  
542 <BookingClassAvail ResBookDesigCode="C" Availability="7"/>  
543 <BookingClassAvail ResBookDesigCode="J" Availability="7"/>  
544 </Cabin>  
545 <Cabin CabinType="Economy" MealCode="S">  
546 <BookingClassAvail ResBookDesigCode="Q" Availability="0"/>  
547 <BookingClassAvail ResBookDesigCode="V" Availability="4"/>  
548 <BookingClassAvail ResBookDesigCode="N" Availability="1"/>  
549 <BookingClassAvail ResBookDesigCode="Y" Availability="7"/>  
550 </Cabin>  
551 <MarketingAirline CompanyCode="BA" CodeContext="IATA"/>  
552 <Equipment AirEquipType="767"/>  
553 <Comment>Departs Terminal 4</Comment>  
554 </FlightSegment>  
555 <FlightSegment SequenceNumber="4" FlightNumber="4632" StopQuantity="0"  
556 JourneyDuration="P0Y0M0DT4H30M" SmokingInd="No" OnTimeRate="90" Ticket="Paper">  
557 <DepartureAirport LocationCode="JFK" CodeContext="IATA"/>  
558 <ArrivalAirport LocationCode="LAX" CodeContext="IATA"/>  
559 <DepartureDateTime>2001-09-11T14:00:00+05:00</DepartureDateTime>  
560 <ArrivalDateTime>2001-09-11T16:35:00+08:00</ArrivalDateTime>  
561 <Cabin CabinType="Business" MealCode="L">  
562 <BookingClassAvail ResBookDesigCode="C" Availability="7"/>  
563 <BookingClassAvail ResBookDesigCode="J" Availability="7"/>  
564 </Cabin>  
565 <Cabin CabinType="Economy" MealCode="S">  
566 <BookingClassAvail ResBookDesigCode="Q" Availability="0"/>  
567 <BookingClassAvail ResBookDesigCode="V" Availability="0"/>  
568 <BookingClassAvail ResBookDesigCode="N" Availability="0"/>  
569 <BookingClassAvail ResBookDesigCode="Y" Availability="7"/>  
570 </Cabin>  
571 <MarketingAirline CompanyCode="BA" CodeContext="IATA"/>  
572 <OperatingAirline CompanyCode="US" CodeContext="IATA"/>  
573 <Equipment AirEquipType="757"/>  
574 </FlightSegment>  
575 </OriginDestinationOption>  
576  
577 <OTA\_SucceedDef>  
578 <Success/>  
579 </OTA\_SucceedDef>  
580 </OTA\_AirAvailRS>

581 **Section 2. – Package Tours/Holiday Bookings**

582

583 A **package holiday** usually consists of a single “pre-defined” offering with or without a choice of  
584 basic elements such as transport and accommodation. The business model for this concept is  
585 that allocated blocks of transport and accommodation inventory for a ‘season’ or ‘brochure period,  
586 typically ‘Summer’ (May to October) and ‘Winter’ (November to April) are reserved by a tour  
587 operator from the supplier. These are combined into package holiday inventory items, and set up  
588 and sold from the tour operator’s system. Notification to the original supplier of the take-up of  
589 individual inventory items takes place a short period before departure of the customers. The use  
590 cases covered in this document relate to the selling by the tour operator of the packages from  
591 their internal inventory stock.

592

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

597

598 The parties involved in the current business interactions comprise Travel Agents (on behalf of  
599 customers) making enquiries and bookings with the Tour Operators who publish brochures  
600 describing the package tours on offer. The normal interaction medium is videotex which, due to  
601 the limited screen display size (80 characters x 25 lines), requires a considerable number of  
602 message pairs to achieve a booking. However, it is well-established and extensively used, with  
603 some operators taking the majority of their bookings this way.

604

605 This document covers two scenarios – Package Availability and Package Booking. The  
606 Availability phase checks a selected package against the supplier’s system and provides full  
607 details and costings and the Booking phase completes the cycle by committing the customer to  
608 paying for the holiday and the supplier to providing it. Each scenario can be invoked  
609 independently of the others, subject to the necessary minimum information being supplied in the  
610 request message.

611

<b>Package Availability</b> comprises the following messages .:	
Package Availability Request.	OTA_PkgAvailRQ
Package Availability Response	OTA_PkgAvailRS
<b>Package Booking</b> comprises the following messages :	
Package Booking Request.	OTA_PkgBookRQ
Package Booking Response	OTA_PkgBookRS

612

613 **2.0. Package Availability Request**

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

617

618 If the request is satisfied, the enquirer will be provided with a priced breakdown of the package  
619 elements.

620

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

623

624 **2.1. Package Availability Use Case**

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

629  
630 The steps of the use case proceed as follows:

631

- 632 • The Customer or agent requests the availability of a specific package for a date and duration  
633 for a number of passengers.
- 634 • The system returns a priced package summary detailing all possible combinations of facilities  
635 (where appropriate).

636

637 The data returned at Step 2 is used as the basis for the Package Booking Request.

638

639 Additional data that accompanies the response message may include information which may  
640 affect the enquirer's decision on whether to book the package, e.g. building works, unavailable  
641 facilities etc.

642

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

646

647 Possible business processing errors include:

648

- 649 • One or more components of the package cannot satisfy the number of passengers for the  
650 date and duration requested. The system may return a list of possible alternative  
651 components and if the enquirer chooses one from the list as a substitute the use case will  
652 restart from step 1.

653

654 **Scenario**

655

656 A customer wants to know if the package consisting of the Hotel Miramar in Alcudia Majorca  
657 travelling on a specific return flight pair between London Gatwick and Palma is available for 2  
658 adults and 2 children for 14 nights from 02 October 2001. The supplier system responds  
659 with information of the flight, the hotel and prices for single, twin, triple bedded rooms for  
660 variable valid occupancies which could be generated by 2 adults and 2 children (e.g. one twin  
661 with two extra beds or a single for 1 adult with a triple for 1 adult and two children etc).

662 **2.2. Package Availability Messages**

663 The specification for Package Availability provides a request/response pair of messages to  
664 support the functionality of enquiring for the availability of a package and its facilities, together  
665 with responses to cover the functionality of providing suitable alternatives should the requested  
666 package not be available.

667

668 The following action verbs are used as root elements of a payload document to obtain package  
669 availability data:

670

- 671 • **OTA PkgAvailRQ** – Provides the identification criteria for the package and customer  
672 requirements.
- 673 • **OTA PkgAvailRS** - Returns a priced summary of the requested package or Warnings, or  
674 Errors if the request did not succeed.

675

## 676 2.2.1. Package Availability Request

677 The <OTA\_PkgAvailRQ> message identifies the required package and availability criteria to be  
678 met.

679  
680 The root element of the <OTA\_PkgAvailRQ> contains the standard payload attributes found in all  
681 OTA payload documents as well as the attribute <ReqRespVersion> that requests a specific  
682 version of the response message. As this is the first publication of the OTA package availability  
683 message set, currently the only valid value is "1". The other components are <POS> which  
684 identifies the requester, and <PackageRequest> which identifies the package required and the  
685 party makeup.

### 686 687 **Package Request**

688  
689 <PackageRequest> identifies the package to be satisfied and defines the date, duration and  
690 customers by age category.

691 The components of <PackageRequest> are:

- 692
- 693
- 694 • *PackageType* – describes a type of TTL product, e.g. flight only, air package etc.
- 695 • *TourCode* – the supplier's code to identify the tour or accommodation arrangement for the  
696 package.
- 697 • *Travel Code* - the supplier's code to identify the travel arrangement for the package.
- 698 • *CompanyCode* – identifies the supplier of the package.
- 699 • *BrochureCode* – identifies the brochure in which the package is offered (Optional).
- 700 • *BoardCode* – identifies the preferred board terms (Optional).
- 701 • *SKU* (Stock Keeping Unit) – an alternative method of identifying the full tour package
- 702 • *URL* (Uniform Resource Locator) – another alternative method of identifying the full tour  
703 package

704 **DateRange** – defines the start date and duration of the required package.

705 **InventoryItems** – a collection of <InventoryItem> elements. Each <InventoryItem> can  
706 be an <AccommodationSegment> or a <TravelSegment>.

707 **CustomerCounts** – the number of customers in the relevant age categories (e.g. Adult, Child  
708 etc.)

709 **TPA\_Extensions** – allows for bi-laterally-agreed additions to the base message

710

### 711 **Sample Messages**

712

713 The following sample messages address various ways to use the Package Availability Request  
714 message to establish the availability of a package holiday:

715

716 Example 1 – A standard package request

717

```
718 <?xml version="1.0" encoding="UTF-8"?>
719 <OTA_PkgAvailRQ xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
720 xsi:schemaLocation="http://www.dummy.com OTA_PkgAvailRQ.xsd" TimeStamp="2001-04-08T21:10:30"
721 EchoToken="7656" Target="Production" Version="1">
722   <POS>
723     <Source AgentSine="A1234">
724       <UniqueId URL="www.agent.com" Type="Profile" Id="a1234"/>
725       <BookingChannel Type="Internet"/>
726     </Source>
727   </POS>
728   <PackageRequest TourCode="ADABA" TravelCode="GAD20" PromotionCode="ABC123">
729     <DateRange StartDate="2001-10-02" Duration="P14D"/>
730     <CustomerCounts>
731       <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
732       <CustomerCount AgeQualifyingCode="Child" Count="2"/>

```

```

733         <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
734     </CustomerCounts>
735 </PackageRequest>
736 </OTA_PkgAvailRQ>

```

## 737 Example 2 – A one-way flight only request

```

738
739
740 <?xml version="1.0" encoding="UTF-8"?>
741 <OTA_PkgAvailRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
742 instance" xsi:schemaLocation="http://www.opentravel.org/OTA
743 OTA_PkgAvailRQ.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1">
744     <POS>
745         <Source AgentSine="A1234">
746             <Uniqueld URL="www.agent.com" Type="Profile" Id="a1234"/>
747             <BookingChannel Type="Internet"/>
748         </Source>
749     </POS>
750     <PackageRequest PackageType="Flight Only" CompanyCode="COS" PromotionCode="ABC123">
751         <DateRange StartDate="2001-10-02"/>
752         <InventoryItems>
753             <InventoryItem>
754                 <Travel>
755                     <AirSegment TravelCode="GAD20" DirectionInd="One Way"/>
756                 </Travel>
757             </InventoryItem>
758         </InventoryItems>
759         <CustomerCounts>
760             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
761         </CustomerCounts>
762     </PackageRequest>
763 </OTA_PkgAvailRQ>
764

```

### 765 2.2.2. Package Availability Response - Success

766 The <OTA\_PkgAvailRS> message returns a priced summary of the requested package or valid  
767 alternatives if any item of inventory cannot be satisfied. The components of a successful  
768 response are as follows:

- 769 • *OTA\_v2ent* – indicates whether the original request was successful or unsuccessful and  
770 includes the standard attributes of the root tag of all OTA messages.
- 771 **Package** – provides complete details of the requested package. Its components are:
  - 772 • *SKU* (Stock Keeping Unit) – the supplier's identification code for the full tour package
  - 773 • *URL* (Uniform Resource Locator) – an alternative method of identifying the full tour  
774 package
  - 775 • *ProductType* – identifies the type of product, e.g. Air Inclusive; Flight Only;  
776 Accommodation Only
  - 777 • *AvailableQty* – the number of seats or beds remaining for sale.
- 778 **DateRange** – provides the start date and overall duration of the package
- 779 **PackageSegments** – A collection of <PackageSegment> elements making up a package  
780 itinerary.
- 781 **Prices** – a collection of package price elements and associated price rules
- 782 **Extras** – a collection of optional items which may be added to a basic package, e.g.  
783 excursions, car hire etc.
- 784 **AcceptedPaymentForms** - provides a list of the methods which may be used to pay for the  
785 holiday.

#### 786 **Package Segment**

787 The components of <PackageSegment> are:

- 788 • *ItinerarySequence* – The position of this item in the package itinerary

- 791 • *ChronologicalSequence* - The position of this item in the package in time sequence. Note: it  
792 is possible for more than one inventory item to have the same ChronologicalSequence value  
793 e.g. Car Hire and Hotel Stay.  
794

795 and any one of the following elements:  
796

797 **DestinationInformation** - contains essential information regarding the resort or area of the  
798 chosen property which is appropriate to the enquirer's stay and could affect their decision to  
799 book. The components of <DestinationInformation> are:

- 800 • *ResortName* – describes the resort or area containing the requested property.
- 801 **Information** – defines the information text together with the period of applicability.

802  
803 **AccommodationInformation** - summarises a property stay with associated facilities and  
804 essential information. The components of <AccommodationInformation> are:

- 805 • *RPH* (Reference Place Holder) – an index code to identify an instance in a collection of  
806 like items.
- 807 • *HotelName* – describes the requested property.
- 808 • *ResortName* – describes the resort or area of the requested property.

809 **DateRange** – defines the start date and duration of the property stay.

810 **RoomProfiles** – defines a set of room profiles identifying room types and occupancies.

811 **MealPlans** – defines the choice of board terms for the stay.

812 **Information** – defines the information text together with the period of applicability.  
813

814 **TransportInformation** - summarises a travel segment and provides essential information when  
815 appropriate. The components of <TransportInformation> are:

816 **Travel** – defines a single travel leg.

817 **Information** – defines the information text together with the period of applicability.  
818

### 819 **Travel**

820 The <Travel> element describes a travel arrangement which satisfies the customer's criteria.

821 Only <AirSegment> is currently defined; other transport segments are planned e.g.

822 <RailSegment>, <FerrySegment>.  
823

824 The components of an <AirSegment> are:

- 825 • *RPH* (Reference Place Holder) – an index code to identify an instance in a collection of like  
826 items.
- 827 • *Type* – Identifies whether the segment is Charter or Scheduled.
- 828 • *TravelDateTimeAttributes*:
  - 829 • *PkgDepartureDay* – defines the day when the travel segment begins e.g. 'Monday' etc.
  - 830 • *PkgCheckIn* – defines check-in date and time.
  - 831 • *PkgDeparture* – defines the take-off date and time.
  - 832 • *PkgArrival* – defines the landing date and time.
- 833 • *TravelCode* – the supplier's code to identify the travel arrangement for the package.
- 834 • *Duration* – a length of time.
- 835 • *CarrierCode* – a code representing the transport operator.
- 836 • *CarrierName* - the name of the transport operator.
- 837 • *Serviceld* – the transport operator's code to identify a travel segment.

838 **DepartureAirport** – identifies the airport where the travel segment begins.

839 **ArrivalAirport** – identifies the airport where the travel segment ends.

840 **Supplement** – the supplement payable for this travel item.

841 **SeatAvailability** – a collection of seat availability by class.  
842

843 **Sample Message - success**  
844

```

845 <?xml version="1.0" encoding="UTF-8"?>
846 <OTA_PkgAvailRS xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
847 xsi:schemaLocation="http://www.dummy.com OTA_PkgAvailRS.xsd" EchoToken="7656" TimeStamp="2001-04-
848 08T21:10:30" Target="Production" Version="1" SequenceNbr="3">
849   <OTA_SucceedDef>
850     <Success/>
851   </OTA_SucceedDef>
852   <Package>
853     <DateRange Start="2001-10-02" Duration="P14D"/>
854     <PackageSegments>
855       <PackageSegment>
856         <TransportInformation>
857           <Travel>
858             <AirSegment RPH="1" DepartureDay="Tue" Departure="2001-10-02T05:00" Arrival="06:45"
859 TravelCode="GAD20">
860               <DepartureAirport Code="LGW">London Gatwick</DepartureAirport>
861               <ArrivalAirport Code="PMI">Palma</ArrivalAirport>
862               <AvailableSeats>
863                 <SeatAvailability SeatQty="4"/>
864               </AvailableSeats>
865             </AirSegment>
866           </Travel>
867           <Information>N.B this flight is non-smoking</Information>
868         </TransportInformation>
869       </PackageSegment>
870       <PackageSegment>
871         <DestinationInformation ResortName="Alcudia">
872           <Information FirstDate="2001-10-08" LastDate="2001-10-15">Please note that the October festival
873 in Alcudia is very noisy</Information>
874         </DestinationInformation>
875       </PackageSegment>
876       <PackageSegment>
877         <AccommodationInformation RPH="2" HotelName="Hotel Miramar" ResortName="Alcudia">
878           <DateRange StartDate="2001-10-02" Duration="P14D"/>
879           <RoomProfiles>
880             <RoomProfile Code="TWIN" Description="Twin room with two extra beds and cot, bath, shower,
881 WC" Qty="1" CotQty="1">
882               <CustomerCounts>
883                 <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
884                 <CustomerCount AgeQualifyingCode="Child" Count="2"/>
885                 <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
886               </CustomerCounts>
887             </RoomProfile>
888           </RoomProfiles>
889           <MealPlans>
890             <MealPlan>
891               <BoardTerms Code="HB" Qty="4">Half Board</BoardTerms>
892             </MealPlan>
893           </MealPlans>
894           <Information>The outdoor swimming pool will be closed for the winter from 14
895 October</Information>
896         </AccommodationInformation>
897       </PackageSegment>
898     </PackageSegment>
899     <TransportInformation>
900       <Travel>
901         <AirSegment RPH="3" DepartureDay="Tue" Departure="2001-10-16T07:45" Arrival="09:30"
902 TravelCode="GAD20">
903           <DepartureAirport Code="PMI">Palma</DepartureAirport>
904           <ArrivalAirport Code="LGW">London Gatwick</ArrivalAirport>
905           <AvailableSeats>
906             <SeatAvailability SeatQty="4"/>
907           </AvailableSeats>
908         </AirSegment>
909       </Travel>
910       <Information>N.B this flight is non-smoking</Information>
911     </TransportInformation>
912   </PackageSegment>
913 </PackageSegments>
914 </Package>

```

```

915 <AcceptedPaymtForms>
916   <AcceptedPaymtForm PaymtMethodType="CreditCard" PaymtFormUseType="Guarantee">
917     <PaymtCardCodes>
918       <PaymtCardCode Type="AMEX"/>
919       <PaymtCardCode Type="VISA"/>
920       <PaymtCardCode Type="MASTERCARD"/>
921     </PaymtCardCodes>
922   </AcceptedPaymtForm>
923   <AcceptedPaymtForm PaymtMethodType="CreditCard" PaymtFormUseType="Deposit">
924     <PaymtCardCodes>
925       <PaymtCardCode Type="AMEX" Rule="A 1% surcharge is applied to American Express payments."/>
926       <PaymtCardCode Type="VISA"/>
927       <PaymtCardCode Type="MASTERCARD"/>
928     </PaymtCardCodes>
929   </AcceptedPaymtForm>
930   <AcceptedPaymtForm PaymtMethodType="CreditCard" PaymtFormUseType="Paymt">
931     <PaymtCardCodes>
932       <PaymtCardCode Type="AMEX" Rule="A 1% surcharge is applied to American Express payments."/>
933       <PaymtCardCode Type="VISA"/>
934       <PaymtCardCode Type="MASTERCARD"/>
935     </PaymtCardCodes>
936   </AcceptedPaymtForm>
937   <AcceptedPaymtForm PaymtMethodType="DebitCard" PaymtFormUseType="Paymt">
938     <PaymtCardCodes>
939       <PaymtCardCode Type="DELTA"/>
940       <PaymtCardCode Type="SWITCH"/>
941     </PaymtCardCodes>
942   </AcceptedPaymtForm>
943 </AcceptedPaymtForms>
944 </OTA_PkgAvailRS>
945

```

### 946 2.2.3. Package Availability Response - Warning

947 Where an item of Inventory cannot satisfy the request but alternatives are available, the  
948 <OTA\_PkgAvailRS> message may return a warning together with details of these alternatives.

949

950 Components of an <OTA\_PkgAvailRS> message with warnings are as follows:

951

- 952 • *OTA\_v2ent* – indicates whether the original request was successful or unsuccessful and  
953 includes the standard attributes of the root tag of all OTA messages.

954 **Package** – describes the package holiday.

955 **TravelChoices** – a collection of <TravelItem> elements describing alternative travel  
956 arrangements, supplied when the original travel item cannot satisfy the request. The  
957 components of a <TravelItem> element are:

- 958 • *AccomOKFlag* – indicates whether or not the requested accommodation part of the  
959 package is available.

960 **TravelDetail** – provides details of travel arrangements by leg or:

961 **TravelJourney** – provides details of round-trip travel arrangements.

962 **Supplements** – a collection of <Supplement> elements defining additional charges over the  
963 base travel arrangement included in the package.

964 **Information** – defines the information text together with the period of applicability.

965 **AccommodationChoices** – a collection of <AvailableProperty> elements describing  
966 alternative accommodation items, supplied when the original accommodation item cannot satisfy  
967 the request. The components of an <AvailableProperty> element are:

- 968 • *SKU* (Stock Keeping Unit) – the supplier's identification code for the property.
- 969 • *URL* (Uniform Resource Locator) – an alternative method of identifying the full tour  
970 package
- 971 • *CheckInDateTime* - Date and (optionally) time of checkin (also known as Arrival  
972 Date/Time).
- 973 • *Duration* – the length of the stay.

- 974 • *CheckOutDateTime* - Normal Date and (optionally) time of checkout (also known as  
975 Departure Date/Time).  
976 • *LastCheckOutDateTime* - Latest Date and (optionally) time of checkout.  
977 • *MaxChildAge* – the maximum age to qualify for child prices or reductions.  
978 • *TravelOKFlag* – indicates whether or not the requested travel part of the package is  
979 available.  
980 **Property** – provides key details for the property.  
981 **Resort** – identifies the resort containing the property  
982 **AvailableRooms** – a collection of <Room> elements describing rooms available for booking.  
983 **AccommodationClass** – identifies the national and supplier ratings for the property.  
984 **RoomPrices** – a collection of <RoomPrice> elements setting out occupancy-level prices.  
985 **AvailableMealPlans** – a collection of <MealPlan> elements describing meal arrangements  
986 available for booking.  
987 **Sourceldentification** – indicates where details can be found, e.g. Brochure and Page  
988 Number.  
989 **FacilityChoices** – a list of accommodation facilities from which the customer must choose in  
990 order for a successful request to be processed further. The components of  
991 <FacilityChoices> are:  
992 **AvailableRooms** – a collection of <Room> elements describing rooms available for booking.  
993 **RoomPrices** – a collection of <RoomPrice> elements setting out occupancy-level prices.  
994 **AvailableMealPlans** – a collection of <MealPlan> elements describing meal arrangements  
995 available for booking.  
996

### 997 **AvailableRooms**

998 <AvailableRooms> comprises a multiple occurrence of <Room> to define the room  
999 arrangements available for the requested stay at the property. <Room> is a  
1000 <RoomInventoryType>, which describes a type of room/cabin/apartment etc which is available  
1001 for the requested stay at the property.  
1002

1003 The components of <RoomInventoryType> are:

- 1004 • *RPH* (Reference Place Holder) – an index code to identify an instance in a collection of like  
1005 items
- 1006 • *Code* – the code identifying this room type e.g. 'Single', 'Twin' etc.
- 1007 • *Description* – a full description of the room (optional).
- 1008 • *AvailableQty* – the quantity of this room type available for booking (optional).
- 1009 • *MinOccupancy* – the minimum number of occupants for this room type (optional).
- 1010 • *MaxOccupancy* – the maximum number of occupants for this room type (optional).
- 1011 • *CotQty* – the number of cots which this room type can hold (optional).
- 1012 • *FreeChildFlag* – indicates whether the room is available when the booking has a free child  
1013 (optional).
- 1014 • *RoomId* – identifies a specific room or accommodation unit (optional).

1015 **Supplement** – the supplement payable for a room type.  
1016

### 1017 **RoomPrices**

1018 <RoomPrices> comprises a multiple occurrence of <RoomPrice> to define the prices for each  
1019 available room type by occupancy profile. The components of <RoomPrice> used in this  
1020 section are:

- 1021 • *RPH* (Reference Place Holder) – an index code to identify the room type in the package  
1022 element.
- 1023 • *Code* – the Room Code.
- 1024 • *Description* – a short description of the room type.

1025 **CustomerCounts** – a collection of customer counts by age qualification e.g. Adults; Child.

1026 **ItemPrice** – the unit price (e.g. per Adult) and the extended price (the unit price multiplied by the  
1027 customer count for this age qualification) for each age qualification.  
1028 **ProfilePrice** – the sum of the extended prices for this room’s occupancy profile.  
1029

### 1030 **AvailableMealPlans**

1031 <AvailableMealPlans> comprises a multiple occurrence of <MealPlan> to define the board  
1032 arrangements available for the requested stay at the property

1033  
1034 The components of <MealPlan> are:

- 1035 • **BoardTerms** – the <BoardCode> identifying this meal plan e.g. “BB” (Bed & Breakfast), “HB”  
1036 (Half Board) etc. and a <Qty> identifying the number of people taking the plan (if  
1037 <CustomerCounts> is not necessary)

1038 **CustomerCounts** – the number of customers by age category. Only necessary when the  
1039 supplement is variable by age category.

1040 **Supplement** – the supplement payable for this meal plan.

1041

### 1042 **Sample Message – Warning**

1043

```
1044 <?xml version="1.0" encoding="UTF-8"?>
1045 <OTA_PkgAvailRS xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
1046 xsi:schemaLocation="http://www.dummy.com OTA_PkgAvailRS.xsd" EchoToken="7656" TimeStamp="2001-04-
1047 08T21:10:30" Target="Production" Version="1" SequenceNbr="3">
1048   <OTA_SucceedDef>
1049     <Warnings>
1050       <Warning Type="BizRule">Please select room(s) and meal plan(s) from the accompanying lists</Warning>
1051     </Warnings>
1052   </OTA_SucceedDef>
1053   <Package>
1054     <DateRange Start="2001-10-02" Duration="P14D"/>
1055     <PackageSegments>
1056       <PackageSegment>
1057         <TransportInformation>
1058           <Travel>
1059             <AirSegment RPH="1" DepartureDay="Tue" Departure="2001-10-02T05:00" Arrival="06:45"
1060 TravelCode="GAD20">
1061               <DepartureAirport Code="LGW">London Gatwick</DepartureAirport>
1062               <ArrivalAirport Code="PMI">Palma</ArrivalAirport>
1063               <AvailableSeats>
1064                 <SeatAvailability SeatQty="4"/>
1065               </AvailableSeats>
1066             </AirSegment>
1067           </Travel>
1068           <Information>N.B this flight is non-smoking</Information>
1069         </TransportInformation>
1070       </PackageSegment>
1071       <PackageSegment>
1072         <DestinationInformation ResortName="Alcudia">
1073           <Information FirstDate="2001-10-08" LastDate="2001-10-15">Please note that the October festival
1074 in Alcudia is very noisy</Information>
1075         </DestinationInformation>
1076       </PackageSegment>
1077       <PackageSegment>
1078         <AccommodationInformation RPH="2" HotelName="Hotel Miramar" ResortName="Alcudia">
1079           <DateRange StartDate="2001-10-02" Duration="P14D"/>
1080           <Information>The outdoor swimming pool will be closed for the winter from 14
1081 October</Information>
1082         </AccommodationInformation>
1083       </PackageSegment>
1084     </PackageSegment>
1085     <TransportInformation>
1086       <Travel>
1087         <AirSegment RPH="3" DepartureDay="Tue" Departure="2001-10-16T07:45" Arrival="09:30"
1088 TravelCode="GAD20">
1089           <DepartureAirport Code="PMI">Palma</DepartureAirport>
```

```

1090         <ArrivalAirport Code="LGW">London Gatwick</ArrivalAirport>
1091         <AvailableSeats>
1092             <SeatAvailability SeatQty="4"/>
1093         </AvailableSeats>
1094     </AirSegment>
1095 </Travel>
1096     <Information>N.B this flight is non-smoking</Information>
1097 </TransportInformation>
1098 </PackageSegment>
1099 </PackageSegments>
1100 </Package>
1101 <FacilityChoices>
1102 <AvailableRooms>
1103     <Room RPH="1" Code="SINGLE" Description="Single Room" AvailableQty="2" MinimumOccupancy="1"
1104 MaximumOccupancy="1">
1105         <Supplement CurrencyCode="GBP">21.00</Supplement>
1106     </Room>
1107     <Room RPH="2" Code="TWIN" Description=" Twin Room" AvailableQty="5" MinimumOccupancy="2"
1108 MaximumOccupancy="4" MaximumAdults="3" CotQty="2" FreeChildFlag="true"/>
1109 </AvailableRooms>
1110 <RoomPrices BoardTerms="HB">
1111     <RoomPrice RPH="1" Code="SINGLE">
1112         <CustomerCounts>
1113             <CustomerCount AgeQualifyingCode="Adult" Count="1"/>
1114         </CustomerCounts>
1115         <ItemPrice Description="Adult Price">
1116             <UnitPrice>371.00</UnitPrice>
1117         </ItemPrice>
1118         <ProfilePrice>371.00</ProfilePrice>
1119     </RoomPrice>
1120     <RoomPrice RPH="2" Code="TWIN">
1121         <CustomerCounts>
1122             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1123         </CustomerCounts>
1124         <ItemPrice Description="Adult Price">
1125             <UnitPrice>350.00</UnitPrice>
1126         </ItemPrice>
1127         <ProfilePrice>700.00</ProfilePrice>
1128     </RoomPrice>
1129     <RoomPrice RPH="2" Code="TWIN">
1130         <CustomerCounts>
1131             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1132             <CustomerCount AgeQualifyingCode="Child" Count="1"/>
1133         </CustomerCounts>
1134         <ItemPrice Description="Adult Price" Qty="2">
1135             <UnitPrice>350.00</UnitPrice>
1136             <ExtendedPrice>700.00</ExtendedPrice>
1137         </ItemPrice>
1138         <ItemPrice Description="First Child">
1139             <UnitPrice>250.00</UnitPrice>
1140             <ExtendedPrice>250.00</ExtendedPrice>
1141         </ItemPrice>
1142         <ProfilePrice>950.00</ProfilePrice>
1143     </RoomPrice>
1144     <RoomPrice RPH="2" Code="TWIN">
1145         <CustomerCounts>
1146             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1147             <CustomerCount AgeQualifyingCode="Child" Count="2"/>
1148         </CustomerCounts>
1149         <ItemPrice Description="Adult Price" Qty="2">
1150             <UnitPrice>350.00</UnitPrice>
1151             <ExtendedPrice>700.00</ExtendedPrice>
1152         </ItemPrice>
1153         <ItemPrice Description="First Child">
1154             <UnitPrice>200.00</UnitPrice>
1155             <ExtendedPrice>200.00</ExtendedPrice>
1156         </ItemPrice>
1157         <ItemPrice Description="Second Child">
1158             <UnitPrice>250.00</UnitPrice>
1159             <ExtendedPrice>250.00</ExtendedPrice>

```

```

1160     </ItemPrice>
1161     </ProfilePrice>1150.00</ProfilePrice>
1162 </RoomPrice>
1163 <RoomPrice RPH="2" Code="TWIN">
1164     <CustomerCounts>
1165         <CustomerCount AgeQualifyingCode="Adult" Count="3"/>
1166     </CustomerCounts>
1167     <ItemPrice Description="Adult Price" Qty="3">
1168         <UnitPrice>308.00</UnitPrice>
1169     </ItemPrice>
1170     </ProfilePrice>924.00</ProfilePrice>
1171 </RoomPrice>
1172 </RoomPrices>
1173 <RoomPrices BoardTerms="FB">
1174     <RoomPrice RPH="1" Code="SINGLE">
1175         <CustomerCounts>
1176             <CustomerCount AgeQualifyingCode="Adult" Count="1"/>
1177         </CustomerCounts>
1178         <ItemPrice Description="Adult Price">
1179             <UnitPrice>611.00</UnitPrice>
1180         </ItemPrice>
1181         </ProfilePrice>611.00</ProfilePrice>
1182     </RoomPrice>
1183     <RoomPrice RPH="2" Code="TWIN">
1184         <CustomerCounts>
1185             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1186         </CustomerCounts>
1187         <ItemPrice Description="Adult Price" Qty="2">
1188             <UnitPrice>490.00</UnitPrice>
1189         </ItemPrice>
1190         </ProfilePrice>980.00</ProfilePrice>
1191     </RoomPrice>
1192     <RoomPrice RPH="2" Code="TWIN">
1193         <CustomerCounts>
1194             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1195             <CustomerCount AgeQualifyingCode="Child" Count="1"/>
1196         </CustomerCounts>
1197         <ItemPrice Description="Adult Price" Qty="2">
1198             <UnitPrice>490.00</UnitPrice>
1199             <ExtendedPrice>980.00</ExtendedPrice>
1200         </ItemPrice>
1201         <ItemPrice Description="First Child" Qty="1">
1202             <UnitPrice>390.00</UnitPrice>
1203             <ExtendedPrice>390.00</ExtendedPrice>
1204         </ItemPrice>
1205         </ProfilePrice>1370.00</ProfilePrice>
1206     </RoomPrice>
1207     <RoomPrice RPH="2" Code="TWIN">
1208         <CustomerCounts>
1209             <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1210             <CustomerCount AgeQualifyingCode="Child" Count="2"/>
1211         </CustomerCounts>
1212         <ItemPrice Description="Adult Price" Qty="2">
1213             <UnitPrice>490.00</UnitPrice>
1214             <ExtendedPrice>980.00</ExtendedPrice>
1215         </ItemPrice>
1216         <ItemPrice Description="First Child">
1217             <UnitPrice>340.00</UnitPrice>
1218             <ExtendedPrice>340.00</ExtendedPrice>
1219         </ItemPrice>
1220         <ItemPrice Description="Second Child">
1221             <UnitPrice>390.00</UnitPrice>
1222             <ExtendedPrice>390.00</ExtendedPrice>
1223         </ItemPrice>
1224         </ProfilePrice>1710.00</ProfilePrice>
1225     </RoomPrice>
1226     <RoomPrice RPH="2" Code="TWIN">
1227         <CustomerCounts>
1228             <CustomerCount AgeQualifyingCode="Adult" Count="3"/>
1229         </CustomerCounts>

```

```

1230     <ItemPrice Description="Adult Price" Qty="3">
1231         <UnitPrice>448.00</UnitPrice>
1232     </ItemPrice>
1233     <ProfilePrice>1344</ProfilePrice>
1234 </RoomPrice>
1235 </RoomPrices>
1236 <AvailableMealPlans>
1237     <MealPlan>
1238         <BoardTerms Code="HB">Half Board</BoardTerms>
1239     </MealPlan>
1240     <MealPlan>
1241         <BoardTerms Code="FB">Full Board</BoardTerms>
1242         <Supplement CurrencyCode="GBP" AgeQualifyingCode="Adult">140.00</Supplement>
1243         <Supplement CurrencyCode="GBP" AgeQualifyingCode="Child">105.00</Supplement>
1244     </MealPlan>
1245 </AvailableMealPlans>
1246 </FacilityChoices>
1247 </OTA_PkgAvailRS>
1248

```

### 1249 **2.3. Create Booking**

1250 The Create Booking messages are designed to make a confirmed booking of a package holiday  
1251 whose availability may or may not have been checked. An <ActionType> qualifier is available to  
1252 modify the default 'Book' request to simply return a Quotation or make a provisional reservation  
1253 pending authorisation of payment details.

1254 If the 'Book' action request is satisfied, the enquirer will be requested to provide contact and  
1255 payment details. On authorisation of the payment details the enquirer will be provided with a  
1256 Booking Reference (and, optionally, Invoice details for printing).

### 1258 **2.4. Create Booking Use Case**

1259 The business use case that supports this message identifies a customer or agent (person or  
1260 system acting on behalf of the customer) who requires to book a specific occurrence of a  
1261 package. The first step in this use case is for the traveller or requesting party to supply the  
1262 package and party details.

1263 The steps of the use case proceed as follows:

- 1264 1. The Customer or agent requests the creation of a booking for a specific package for a date  
1265 and duration for a number of passengers, together with contact details.
- 1266 2. The system reserves the necessary capacity and confirms the details of the booking including  
1267 costs.
- 1268 3. When necessary the customer or agent provides payment details and the supplier obtains  
1269 payment authorisation.
- 1270 4. The supplier creates a booking entity and provides the customer or agent with the Booking  
1271 Reference and optionally the data to produce a written confirmation.

1272 Possible business processing errors include:

- 1273 • One or more elements of the package cannot satisfy the number of passengers for the date  
1274 and duration requested in which circumstance the use case will revert to the Package  
1275 Availability response message as described in the Package Availability scenario in section 1  
1276 of this document.
- 1277 • Payment authorisation is refused.

#### 1280 Scenario

1281 A customer wishes to book the package consisting of the Hotel Miramar in Alcudia Majorca  
1282 travelling on a specific return flight pair between London Gatwick and Palma for 2 adults and  
1283 2 children and one infant for 14 nights from 02 October 2001.

1284 **2.5. Create Booking Messages**

1285 The OTA version 2001B specification of Create Booking provides a set of messages to support  
1286 the functionality of creating a package booking.

1287 The syntax of the action verbs that are the root elements of a payload document used to create a  
1288 booking are as follows:

- 1289 • **OTA\_PkgBookRQ** – Provides the identification criteria for the package components,  
1290 passenger and contact details.
- 1291 • **OTA\_PkgBookRS** – Provides complete details of the booking with costs and methods of  
1292 payment.

1293 **2.5.1. Create Booking Request**

1294 The OTA\_PkgBookRQ message identifies the required package and supplies passenger details  
1295 for the booking.

1296 The root element of the <OTA\_PkgBookRQ> contains the standard payload attributes found in all  
1297 OTA payload documents as well as the attribute <ReqRespVersion> that requests a specific  
1298 version of the response message. As this is the first publication of the OTA package availability  
1299 message set, currently the only valid value is "1".

1300

1301 The other components are:

- 1302 • *ActionType* identifies the action to be taken – Book, Hold, Quote or Confirm.

1303 **POS** – identifies the requestor.

1304 **Uniqueld** – identifies a provisional reservation being retrieved for confirmation.

1305 **PackageRequest** – identifies the package to be satisfied and defines the date, duration and  
1306 customers by age category.

1307 **ContactDetail** – Details of the person to whom correspondence regarding a booking will be  
1308 directed.

1309 **PassengerListItems** – a collection of <PassengerListItem> elements each providing identity  
1310 details of a single passenger. The components of a <PassengerListItem> are:

- 1311 • *PassengerRPH* – the index used to identify a passenger.
- 1312 • *Gender* – A code indicating the gender of a passenger.
- 1313 • *Nationality* – A code to identify the country of origin of a person by birth or naturalisation.
- 1314 • *Age* – the age of a passenger, usually required as at the end of the stay.
- 1315 • *BirthDate* – the date of birth of a passenger.

1316 **PersonName** – identifies an individual passenger by name.

1317 **SpecialNeeds** – a collection of <SpecialNeed> items.

1318 **PersonalInsuranceItems** – a collection of <PersonalInsuranceItem> elements which  
1319 identifies the type of personal insurance and number of customers covered. The components of a  
1320 <PersonalInsuranceItem> are:

- 1321 • *Code* – A coded identification of a type of insurance.

1322 **CustomerCounts** – the number of customers in the relevant age categories (e.g. Adult,  
1323 Child etc.).

1324 **OwnInsuranceChoice** – a collection of <OwnInsuranceChoice> elements providing details of  
1325 the customer's own insurance policy which is being used in place of the supplier's offering. The  
1326 components of <OwnInsuranceChoice> are:

- 1327 • *InsuranceCompany* – the name of the customer's chosen insurance company.
- 1328 • *PolicyNumber* – a reference by which an issued policy is recognised.

1329 **CustomerCounts** – the number of customers in the relevant age categories (e.g. Adult, Child  
1330 etc.)

1331 **PaymentDetails** – a collection of <PaymentDetail> elements which define a monetary amount  
 1332 and a method of payment. This will commonly be the only element, apart from the Uniqueld,  
 1333 supplied for a 'Confirm' action. The components of <PaymentDetail> are:  
 1334 **PaymentForm** – defines the method of payment. See OTA\_Profile for description.  
 1335 **Amount** – the monetary value to be paid with the method of payment.  
 1336

1337 **Sample Message – Book**

```

1338
1339 <?xml version="1.0" encoding="UTF-8"?>
1340 <OTA_PkgBookRQ xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
1341 xsi:schemaLocation="http://www.dummy.com OTA_PkgBookRQ.xsd" TimeStamp="2001-04-08T21:10:30"
1342 EchoToken="7656" Target="Production" Version="1" ActionType="Book">
1343   <Uniqueld Type="Reservation"/>
1344   <InventoryItems>
1345     <InventoryItem>
1346       <Travel>
1347         <AirSegment RPH="1" DepartureDate="2001-10-02" TravelCode="GAD20">
1348           <AvailableSeats>
1349             <SeatAvailability SeatQty="4"/>
1350           </AvailableSeats>
1351         </AirSegment>
1352       </Travel>
1353     </InventoryItem>
1354     <InventoryItem>
1355       <Accommodation RPH="2">
1356         <HotelReference HotelCode="ADABA"/>
1357         <DateRange StartDate="2001-10-02" Duration="P14D"/>
1358         <RoomProfiles>
1359           <RoomProfile RPH="01" Code="TWIN" Qty="1" CotQty="1">
1360             <CustomerCounts>
1361               <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1362               <CustomerCount AgeQualifyingCode="Child" Count="2"/>
1363               <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
1364             </CustomerCounts>
1365             <PassengerRPHs>
1366               <PassengerRPH RPH="1"/>
1367               <PassengerRPH RPH="2"/>
1368               <PassengerRPH RPH="3"/>
1369               <PassengerRPH RPH="4"/>
1370               <PassengerRPH RPH="5"/>
1371             </PassengerRPHs>
1372           </RoomProfile>
1373         </RoomProfiles>
1374         <MealPlans>
1375           <MealPlan>
1376             <BoardTerms Code="HB" Qty="4"/>
1377           </MealPlan>
1378         </MealPlans>
1379       </Accommodation>
1380     </InventoryItem>
1381     <InventoryItem>
1382       <Travel>
1383         <AirSegment RPH="3" DepartureDate="2001-10-16" TravelCode="GAD20">
1384           <AvailableSeats>
1385             <SeatAvailability SeatQty="4"/>
1386           </AvailableSeats>
1387         </AirSegment>
1388       </Travel>
1389     </InventoryItem>
1390   </InventoryItems>
1391   <ContactDetail>
1392     <Name>
1393       <NamePrefix>Mr</NamePrefix>
1394       <GivenName>S</GivenName>
1395       <Surname>Corcoran</Surname>
1396       <PrivacyDetails/>
1397     </Name>
1398   </ContactDetail>

```

```

1399 <PassengerListItems>
1400   <PassengerListItem RPH="01" Gender="M" Nationality="GB" BirthDate="1972-02-27">
1401     <Name>
1402       <NamePrefix>Mr</NamePrefix>
1403       <GivenName>Simon</GivenName>
1404       <Surname>Corcoran</Surname>
1405       <PrivacyDetails/>
1406     </Name>
1407   </PassengerListItem>
1408   <PassengerListItem RPH="02" Gender="F" Nationality="US" Age="24">
1409     <Name>
1410       <NamePrefix>Mrs</NamePrefix>
1411       <GivenName>A</GivenName>
1412       <Surname>Corcoran</Surname>
1413       <PrivacyDetails/>
1414     </Name>
1415   </PassengerListItem>
1416   <PassengerListItem RPH="03" Gender="F" Nationality="GB" Age="8">
1417     <Name>
1418       <NamePrefix>Miss</NamePrefix>
1419       <GivenName>B</GivenName>
1420       <Surname>Corcoran</Surname>
1421       <PrivacyDetails/>
1422     </Name>
1423   </PassengerListItem>
1424   <PassengerListItem RPH="04" Gender="M" Nationality="GB" Age="4">
1425     <Name>
1426       <NamePrefix>Mstr</NamePrefix>
1427       <GivenName>C</GivenName>
1428       <Surname>Corcoran</Surname>
1429       <PrivacyDetails/>
1430     </Name>
1431   </PassengerListItem>
1432   <PassengerListItem RPH="05" Gender="F" Nationality="GB" Age="1">
1433     <Name>
1434       <NamePrefix>Inf</NamePrefix>
1435       <GivenName>D</GivenName>
1436       <Surname>Corcoran</Surname>
1437       <PrivacyDetails/>
1438     </Name>
1439   </PassengerListItem>
1440 </PassengerListItems>
1441 <OwnInsuranceChoices>
1442   <OwnInsuranceChoice InsuranceCompany="Home and Colonial" PolicyNmbr="1453891762">
1443     <CustomerCounts>
1444       <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
1445       <CustomerCount AgeQualifyingCode="Child" Count="2"/>
1446       <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
1447     </CustomerCounts>
1448   </OwnInsuranceChoice>
1449 </OwnInsuranceChoices>
1450 </OTA_PkgBookRQ>
1451

```

### Sample Message – Confirm

```

1452
1453
1454 <OTA_PkgBookRQ xmlns="http://www.opentravel.org/OTA" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1455 xsi:schemaLocation="http://www.opentravel.org/OTA
1456 OTA_PkgBookRQ.xsd" TimeStamp="2001-04-08T21:10:30" EchoToken="7656" Target="Production" Version="1"
1457 ActionType="Confirm">
1458   <UniqueId Type="Reservation" Id="AA12345"/>
1459   <PaymentDetails>
1460     <PaymentDetail>
1461       <CreditCard CardType="Credit" CardNumber="4444 1234 5678 5532" ExpireDate="1001">
1462         <CardHolderName>
1463           <NamePrefix>Mr</NamePrefix>
1464           <GivenName>S</GivenName>
1465           <Surname>Corcoran</Surname>
1466           <PrivacyDetails/>
1467         </CardHolderName>

```

```

1468     <CardIssuerName IssuingBankId="VISA"/>
1469     <Address>
1470         <AddressLine>
1471             <AddressLine>10 This Road</AddressLine>
1472             <AddressLine>Holborn</AddressLine>
1473         </AddressLine>
1474         <CityName PostalCode="EC4 1HJ">London</CityName>
1475         <CountryName CountryCode="UK">United Kingdom</CountryName>
1476         <PrivacyDetails/>
1477     </Address>
1478     <PrivacyDetails/>
1479 </CreditCard>
1480 <PrivacyDetails/>
1481 <Amount CurrencyCode="GBP">1456.00</Amount>
1482 </PaymentDetail>
1483 </PaymentDetails>
1484 </OTA_PkgBookRQ>
1485

```

## 1486 2.5.2. Create Booking Response

1487 The OTA\_PkgBookRS message provides full details of the requested booking.

1488 The components of OTA\_PkgBookRS are as follows:

- 1489 • *OTA\_v2ent* – indicates whether the original request was successful or unsuccessful and
- 1490 includes the standard attributes of the root tag of all OTA messages.
- 1491 **PackageReservation** – details the booking as recorded by the supplier. The components of
- 1492 <PackageReservation> are:
- 1493 **POS** – identifies the requestor.
- 1494 **Uniqueld** – identifies a provisional reservation being retrieved for confirmation.
- 1495 **InventoryItems** – a collection of <InventoryItem> elements. Each <InventoryItem>
- 1496 can be an <AccommodationSegment> or a <TravelSegment>.
- 1497 **ContactDetail** – Details of the person to whom correspondence regarding a booking will be
- 1498 directed.
- 1499 **PassengerListItems** – a collection of <PassengerListItem> elements.
- 1500 **PersonalInsuranceItems** – a collection of <PersonalInsuranceItem> elements.
- 1501 **OwnInsuranceChoice** – a collection of <OwnInsuranceChoice> elements.
- 1502 **InvoiceDetail** – a list of all the items making up a priced booking.
- 1503 **InformationItems** – a collection of the information required for individual inventory items.

### 1504 **Invoice Detail**

1505 **Invoice Detail** is a list of all the items making up a priced booking. The components of

1506 <InvoiceDetail> are:

- 1507 • *BalanceDueDate* – the date on which the BalanceAmount is payable by the customer.
- 1508 **CostingItems** – is a collection of <CostingItem> elements which define an individually priced
- 1509 component of a priced booking. The components of <CostingItem> are:
- 1510 • *Description* – a description of the priced item.
- 1511 • *Quantity* – the number of units required.
- 1512 **UnitPrice** – the cost of one unit of the component.
- 1513 **ExtendedPrice** – unit price times quantity..
- 1514 **GrossAmount** – defines the total amount payable by the customer.
- 1515 **DepositAmount** – defines the amount payable at the time of booking to secure the booking.
- 1516 **DiscountCommission** – defines the fee earned by the agent for the booking.
- 1517 **TaxItems**– defines the tax amount(s) payable on an invoice.
- 1518 **NetAmount** – the amount payable by the agent, i.e. GrossAmount less Discount/Commission
- 1519 (optional).
- 1520 **BalanceDueAmount** – the amount remaining to be paid by the customer i.e. GrossAmount less
- 1521 DepositAmount.

1522

1524  
1525  
1526  
1527  
1528  
1529  
1530  
1531  
1532  
1533  
1534  
1535  
1536  
1537  
1538  
1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575  
1576  
1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591  
1592

## Sample message

```
<?xml version="1.0" encoding="UTF-8"?>
<OTA_PkgBookRS xmlns:xsi="http://www.w3.org/2000/10/XMLSchema-instance"
xsi:schemaLocation="http://www.dummy.com OTA_PkgBookRS.xsd" EchoToken="7656" TimeStamp="2001-04-
08T21:10:30" Target="Production" Version="1" SequenceNbr="3">
  <OTA_SucceedDef>
    <Success/>
  </OTA_SucceedDef>
  <PackageReservation>
    <Uniqueld Type="Reservation" Id="AA12345"/>
    <InventoryItems>
      <InventoryItem ChronologicalSequence="1" ItinerarySequence="1">
        <Travel>
          <AirSegment RPH="1" DepartureDay="Tue" Departure="2001-10-02T05:00" Arrival="06:45"
TravelCode="GAD20">
            <DepartureAirport Code="LGW">London Gatwick</DepartureAirport>
            <ArrivalAirport Code="PMI">Palma</ArrivalAirport>
            <AvailableSeats>
              <SeatAvailability SeatQty="4"/>
            </AvailableSeats>
          </AirSegment>
        </Travel>
      </InventoryItem>
      <InventoryItem ChronologicalSequence="2" ItinerarySequence="2">
        <Accommodation RPH="2" HotelName="Hotel Miramar" ResortName="Alcudia">
          <HotelReference HotelCode="ADAB"/>
          <DateRange StartDate="2001-10-02" Duration="P14D"/>
          <RoomProfiles>
            <RoomProfile Code="TWIN" Description="Twin room with two extra beds and cot, bath, shower,
WC" Qty="1" CotQty="1">
              <CustomerCounts>
                <CustomerCount AgeQualifyingCode="Adult" Count="2"/>
                <CustomerCount AgeQualifyingCode="Child" Count="2"/>
                <CustomerCount AgeQualifyingCode="Infant" Count="1"/>
              </CustomerCounts>
            </RoomProfile>
          </RoomProfiles>
          <MealPlans>
            <MealPlan>
              <BoardTerms Code="HB" Qty="4">Half Board</BoardTerms>
            </MealPlan>
          </MealPlans>
        </Accommodation>
      </InventoryItem>
      <InventoryItem ChronologicalSequence="3" ItinerarySequence="3">
        <Travel>
          <AirSegment RPH="3" DepartureDay="Tue" Departure="2001-10-16T07:45" Arrival="09:30"
TravelCode="GAD20">
            <DepartureAirport Code="PMI">Palma</DepartureAirport>
            <ArrivalAirport Code="LGW">London Gatwick</ArrivalAirport>
            <AvailableSeats>
              <SeatAvailability SeatQty="4"/>
            </AvailableSeats>
          </AirSegment>
        </Travel>
      </InventoryItem>
    </InventoryItems>
    <ContactDetail>
      <Name>
        <NamePrefix>Mr</NamePrefix>
        <GivenName>S</GivenName>
        <Surname>Corcoran</Surname>
        <PrivacyDetails/>
      </Name>
    </ContactDetail>
    <PassengerListItems>
      <PassengerListItem RPH="01" Gender="M" Nationality="GB" BirthDate="1972-02-27">
        <Name>
```

```

1593         <NamePrefix>Mr</NamePrefix>
1594         <GivenName>Simon</GivenName>
1595         <Surname>Corcoran</Surname>
1596         <PrivacyDetails/>
1597     </Name>
1598 </PassengerListItem>
1599 <PassengerListItem RPH="02" Gender="F" Nationality="US" Age="24">
1600     <Name>
1601         <NamePrefix>Mrs</NamePrefix>
1602         <GivenName>A</GivenName>
1603         <Surname>Corcoran</Surname>
1604         <PrivacyDetails/>
1605     </Name>
1606 </PassengerListItem>
1607 <PassengerListItem RPH="03" Gender="F" Nationality="GB" Age="8">
1608     <Name>
1609         <NamePrefix>Miss</NamePrefix>
1610         <GivenName>B</GivenName>
1611         <Surname>Corcoran</Surname>
1612         <PrivacyDetails/>
1613     </Name>
1614 </PassengerListItem>
1615 <PassengerListItem RPH="04" Gender="M" Nationality="GB" Age="4">
1616     <Name>
1617         <NamePrefix>Mstr</NamePrefix>
1618         <GivenName>C</GivenName>
1619         <Surname>Corcoran</Surname>
1620         <PrivacyDetails/>
1621     </Name>
1622 </PassengerListItem>
1623 <PassengerListItem RPH="05" Gender="F" Nationality="GB" Age="1">
1624     <Name>
1625         <NamePrefix>Inf</NamePrefix>
1626         <GivenName>D</GivenName>
1627         <Surname>Corcoran</Surname>
1628         <PrivacyDetails/>
1629     </Name>
1630 </PassengerListItem>
1631 </PassengerListItems>
1632 <InvoiceDetail>
1633     <CostingItems>
1634         <CostingItem Description="Basic Holiday Price" Qty="2">
1635             <UnitPrice>546.00</UnitPrice>
1636             <ExtendedPrice>1092.00</ExtendedPrice>
1637         </CostingItem>
1638         <CostingItem Description="First Child" Qty="1">
1639             <UnitPrice>150.00</UnitPrice>
1640             <ExtendedPrice>150.00</ExtendedPrice>
1641         </CostingItem>
1642         <CostingItem Description="Second Child" Qty="1">
1643             <UnitPrice>223.00</UnitPrice>
1644             <ExtendedPrice>223.00</ExtendedPrice>
1645         </CostingItem>
1646     </CostingItems>
1647     <GrossAmount>1465.00</GrossAmount>
1648     <DepositAmount>300.00</DepositAmount>
1649     <BalanceDueAmount>1165.00</BalanceDueAmount>
1650 </InvoiceDetail>
1651 </PackageReservation>
1652 <InformationItems>
1653     <InformationItem ItinerarySequence="2" Type="Resort" Name="Alcudia">
1654         <Information FirstDate="2001-10-02" Duration="P14D" >Please note that the October festival in Alcudia is
1655 very noisy</Information>
1656     </InformationItem>
1657 </InformationItems>
1658 </OTA_PkgBookRS>
1659
1660

```

### 1661 **Section 3. – Golf Tee Times**

1662 The OTA version 2001C specification of Golf Tee Times provides three separate  
1663 request/response pairs of messages to support the functionality of requesting data from another  
1664 system in the process of finding a golf course, inquiring as to availability, and booking a tee time.  
1665 All message sets assumes a pull model, where the originating system requests a specific set of  
1666 data (as agreed by trading partners).

1667  
1668 A system through which an interested party initiates a booking process will request a list of  
1669 courses that meet the specified qualifiers using the **OTA\_CourseSearchRQ** message. In this  
1670 message, the desired criteria date is identified. The receiving system (typically a Golf Course Tee  
1671 Sheet System or a consolidator's system) responds with the **OTA\_CourseSearchRS** message,  
1672 which includes either summary or detailed information about the courses that meet the requested  
1673 criteria. Where the flag **DetailResponse** is set to "Yes", all traits of the course(s) meeting the  
1674 criteria are returned. Where the flag is set to "No", only those traits matching the requested  
1675 criteria and the basic course information are returned. All messages assume the no-state,  
1676 meaning that the querying system will initiate the transaction and expect a response from the  
1677 queried system. All message responses include the request identification. Responses may be  
1678 returned in any order.

1679  
1680 **OTA\_CourseSearchRQ** – Sends a request for course information to another system. All the  
1681 elements and attributes are optional, unless otherwise stated as required. The requesting system  
1682 may request a detailed or summary response.

1683  
1684 **OTA\_CourseSearchRS** - Returns a set of data representing the course(s) that meet the  
1685 requested criteria. Where the criteria attribute of **Required** is "Yes" then only those courses that  
1686 meet those criteria will be returned. Where the **Required** attribute is "No" then a course that does  
1687 not meet that criteria may be included in the set. In all cases, where the criteria has been  
1688 included in the request, the comparable trait and its value will be returned, along with the basic  
1689 course information and identification. The message may also include **Warnings** from business  
1690 processing rules or **Errors** if the request did not succeed.

1691  
1692 **OTA\_CourseAvailRQ** – Sends a request for a report to another system. All the elements and  
1693 attributes are optional, unless otherwise stated as required.

1694  
1695 **OTA\_CourseAvailRS** - Returns the requested set of data if the request can be processed, or  
1696 includes **Warnings** from business processing rules or **Errors** if the request did not succeed.

1697  
1698 **OTA\_CourseResRQ** – Sends a request for a reservation to another system. All the elements and  
1699 attributes are optional, unless otherwise stated as required.

1700  
1701 **OTA\_CourseResRS** - Returns the requested reservation if the request can be processed, or  
1702 includes **Warnings** from business processing rules or **Errors** if the request did not succeed.

1703  
1704

#### 1705 **Note:**

1706 All the elements are indicated in **Bold**.

1707 All the attributes are indicated in *italics*.

1708

1709 Example:

1710 **Element** – Element Description

1711 • *Attribute* – Attribute Description

1712 **3.1. Golf Course Search Request Message (OTA\_CourseSearchRQ)**

1713

1714 The root tag of OTA\_CourseSearch> contains the standard payload attributeGroup  
1715 OTA\_PayloadStdAttributes, found in all OTA payload documents.

1716

1717 **OTA\_CourseSearchRQ** - This is the main Request element.

1718

1719 • *OTA\_PayloadStdAttributes*- includes standard attributes of the root tag of all OTA messages  
1720 as defined in the OTA Version 2 Infrastructure Document.

1721

1722 • *GolfCourseID* - If a GolfCourseID is included in the request, then the response will be for only  
1723 that golf course. If no GolfCourseID is included, the response will include all golf courses that  
1724 are deemed to meet the requested criteria.

1725

1726 • *DetailResponse* - If the requestor desires a response that includes all the traits associated  
1727 with the golf course(s) in the response, this boolean is set to "Yes". In that case, all the traits  
1728 of the course will be returned. Where the DetailResponse is set to "No", the response will be  
1729 required to send only the traits that match the criteria being sent.

1730

1731 **Criteria** - Criteria is a repeating set of features that are desired in the search for a golf course.  
1732 The Name and Value pair define the criteria for the search. If the requestor demands that the  
1733 result be filtered on a particular criterion, then the Required boolean is set to "Yes". If the  
1734 Required boolean is not set to "Yes", then the response can include courses that do not meet  
1735 those exact criteria. Examples would be:

1736

Name="Architect", Value="Robert Trent Jones", Required="Yes".

1737

Name="Location", Value="Myrtle Beach", Required="Yes".

1738

Name="Caddies", Value="Yes", Required="No".

1739

Name="Length", Value="6600 Yds", Required="Yes", Operation=">".

1740

The Name, Value, and Required attributes are required, but the Operation is

1741

optional.

1742

1743 • *Name* - A code representing the specific Criterion on which the search is to filter.

1744

• *Value* - The value of the Criterion.

1745

• *Required* - A flag establishing if this criterion must be met in a successful  
1746 candidate. If "Yes", then all responses must meet the specified criterion.

1747

• *Operation* - Where a test against a criterion value is not an equality, this represents  
1748 the operation to be used as the filter. Examples would be GT (Greater Than), LT  
1749 (Less Than), etc

1750

1751 **3.2. Golf Course Search Response Message (OTA\_CourseSearchRS)**

1752 The root tag of <OTA\_CourseSearchRS> uses the external *include* <OTA\_v2ent.xsd> that  
1753 defines the root element standard attributes found in all the OTA payload documents, and the  
1754 response options of returning the indication of Success, Warning or Errors in processing the  
1755 request. The response message may include Warnings from business processing rules or Errors  
1756 if the request did not succeed.

1757

1758 **OTA\_CourseSearchRS** - Main Response element.

1759

• *OTA\_PayloadStdAttributes* - See above

1760

Success

1761

1762

1763

1764

**Warnings**



1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871  
1872

**OTA\_CourseAvailRQ** - The OTA\_CourseAvailRQ 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. It is the primary request element.

- *OTA\_PayloadStdAttributes* - See above
- *GolfCourseID* - A unique identifier assigned to the golf course

**GolfCourseTeeTime** - The GolfCourseTeeTime attributes include the range of dates and times for which the round (or rounds) are being requested, as well as information relating to the number of golfers, the number of holes requested, and the number of times. It also includes the maximum price permitted in the response

- *StartDate* - The first date for which the availability request is valid
- *EndDate* - The last date for which the availability request is valid. If the request is for one day only, this will be the same as the StartDate
- *StartTime* - The earliest time for which the availability request is valid
- *EndTime* - The latest start time for which the availability request is valid. The StartTime and EndTime refer to each day of the period represented by the StartDate and EndDate. Where the start and end time varies by day, separate requests must be sent
- *NumberOfGolfers* - The number of golfers covered by this request.
- *NumberOfHoles* - The number of holes that the golfers wish to play within one round. If this differs by day, separate messages must be sent.
- *NumberOfTimes* - The number of Tee Times covered by this request. For example, if a foursome wants to play two separate rounds within one day, the number of tee times would be 2
- *MaxPrice* - The highest price acceptable for the requested rounds. Where a number of courses are represented by a system, only those rounds that are at or below the highest price should be returned
- *Currency* - The ISO currency code in which the price is to be quoted, and in which the max price is stated

**RateQualifier** - The RateQualifier is a code or set of repetitive codes that represent a business agreement between trading partners, or a request for a rate that might be associated with the requesting party. For example, if the requestor is a wholesaler or reseller, the RateQualifier will be a code associated with that status. If the individual golfer is a member of an organization that typically gets a discount (such as NGCOA, PGA, USGA, etc.) then that code would be included in the rate qualifier.

**Memberships** - Memberships are associated with one or more of the golfers for whom the specific request for availability is being made. Some memberships may permit booking rules that do not apply to other golfers. For example, a reciprocal membership may allow booking farther in advance than the standard golfer is permitted. NOTE: a definition of Memberships should be reflected in the OTA\_Profile schema.

**Membership** - An individual membership.

- *Type* - The type of membership, using a code to represent the specific entity. Examples would be PGA, NGCOA, USGA, etc.
- *ID* - The Identification of the party within the stated membership

1873  
1874  
1875  
1876  
1877  
1878  
1879  
1880  
1881  
1882  
1883  
1884  
1885  
1886  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925

### 3.4. Course Availability Response Message (OTA\_CourseAvailRS)

The root tag of OTA\_StatisticsNotifRS uses the external *include* OTA\_v2ent.xsd that defines the root element standard attributes found in all the OTA payload documents, and the response options of returning the indication of Success, Warning or Errors in processing the request. The response message may include Warnings from business processing rules or Errors if the request did not succeed.

**OTA\_CourseAvailRS** - Main Response element.

*OTA\_PayloadStdAttributes*

#### Success

#### Warnings

##### Warning

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

#### Errors

##### Error

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

**GolfCourseTeeTime** - The GolfCourseTeeTime attributes include the range of dates and times for which the round (or rounds) are being requested, as well as information relating to the number of golfers, the number of holes requested, and the number of times. It also includes the maximum price permitted in the response

- *StartDate* - The first date for which the availability request is valid
- *EndDate* - The last date for which the availability request is valid. If the request is for one day only, this will be the same as the StartDate
- *StartTime* - The earliest time for which the availability request is valid
- *EndTime* - The latest start time for which the availability request is valid. The StartTime and EndTime refer to each day of the period represented by the StartDate and EndDate. Where the start and end time varies by day, separate requests must be sent
- *NumberOfGolfers* - The number of golfers covered by this request.
- *NumberOfHoles* - The number of holes that the golfers wish to play within one round. If this differs by day, separate messages must be sent.
- *NumberOfTimes* - The number of Tee Times covered by this request. For example, if a foursome wants to play two separate rounds within one day, the number of tee times would be 2

1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936  
1937  
1938  
1939  
1940  
1941  
1942  
1943

- *MaxPrice* - The highest price acceptable for the requested rounds. Where a number of courses are represented by a system, only those rounds that are at or below the highest price should be returned
- *Currency* - The ISO currency code in which the price is to be quoted, and in which the max price is stated
- *ConfID* – A confirmation Identification issued by the supplier’s authority.

**Fee** - The Fee is the actual amount to be charged for the associated round of golf. It includes the RateQualifier and a description of that rate, as well as the amount and currency

- *RateQualifier* - A code representing the basis for which the rate was quoted. If a discount or net rate was given, this indicates the reason for the discount or net rate
- *Name* - A description of the rate
- *Amount* - The amount of the rate.
- *Currency* - The ISO currency code for the currency in which the rate is quoted

1944 **3.5. Course Tee Time Reservation Request Message**  
1945 **(OTA\_CourseResRQ)**

1946 The root tag of OTA\_CourseResRQ uses the external *include* <OTA\_v2ent.xsd> that defines the  
1947 root element standard attributes found in all the OTA payload documents  
1948

1949 **OTA\_CourseResRQ** - The OTA\_CourseResRQ message is used to request a reservation at a  
1950 known single course for one or more potential tee times. The specific information about the golfer  
1951 or golfers is necessary in order to validate booking rules and set rates. Where the booking entity  
1952 has the authority to take a reservation without a request (from an existing block) then the  
1953 Notification boolean will be set to "Yes". It is the main Request element.  
1954

1955 *OTA\_PayloadStdAttributes* - See above  
1956

1957  
1958 **GolfCourseRes** - A specific reservation for a tee time for an individual or a number of golfers  
1959

- 1960
- 1961 • *GolfCourseID* - The Unique Identifier associated with the golf course
  - 1962 • *Notification* - Where the booking entity has the authority to take a reservation without a  
1963 request (from an existing block) then the Notification boolean will be set to "Yes". In this  
1964 case, the represented reservation will be considered to have been made, and this  
1965 message is a notification of the reservation. The supplier has contractually agreed to  
1966 accept such a reservation, and cannot deny the notification.
  - 1967 • *Id* - A unique identifier assigned to this reservation
  - 1968 • *Status* - The status of this reservation transaction
  - 1969 • *StatusMessage* - A description or message associated with the status
  - 1970 • *RequestorResId* - The unique Identifier used by the party booking the reservation
  - 1971 • *ResponderResConfId* - The unique identifier assigned by the supplier.
  - 1972 • *CancellationPolicy* - An explanation of the cancellation policy in effect for this reservation
  - 1973 • *CancellationWindow* - The final time at which a cancellation may be made
  - 1974 • *CancellationCharge* - The charge to be applied in the event of cancellation

1975  
1976 **Round** - A single round of golf that may include one or more golfers. If there is more  
1977 than one round, each round will be represented separately  
1978

- 1979
- 1980 • *RoundId* - A unique identifier applied to this round
  - 1981 • *PlayDate* - The date for which the round is reserved
  - 1982 • *PlayTime* - The time for which the round is reserved
  - 1983 • *NumberOfGolfers* - The number of golfers represented in this round
  - 1984 • *PackagesId* - An identification of a package of which this round is a part
  - 1985 • *StartingTee* - The tee on which the round will start. This could be a hole number, or a  
1986 course designation where there are multiple courses.
  - 1987 • *NumberOfCarts* - The number of carts included within the reservation
  - 1988 • *TotalAmount* - The Total Amount for this round
  - 1989 • *Currency* - The ISO currency code in which the TotalAmount is represented.

1990 **Charge** - The actual pricing and costing information about the charges being applied  
1991 to the round. One or more Charge may apply to a single round  
1992

- 1993
- 1994 • *Id* - A unique identifier representing the charge
  - 1995 • *Description* - A description of the charge
  - *Quantity* - The quantity associated with this charge
  - *UnitPrice* - The unit price for this charge

1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029

- *UnitCost* - A unit cost associated with this charge. The cost to the booking party may be different from the charge to the golfer
- *Amount* - Total amount extended
- *BillingType* - A code representing the method of billing between parties

**Golfer** - Information about the individual golfer for which the reservation is being requested

- *Gender* – The Gender (if known) of the golfer.
- *DateOfBirth* – The Date of Birth (if known) of the golfer.

**Memberships** – a collection of membership information for the golfer.

**Membership** – information about a specific membership.

- *Code* – The code representing the membership.
- *ID* – The identification of this member within the membership group.

**PaymentForm** – See OTA\_Profile for description.

**PersonName** – See OTA\_Profile for description

**Address** – See OTA\_Profile for description

**Telephone** – See OTA\_Profile for description

**Email** – See OTA\_Profile for description

**RateQualifier** – A single or multiple codes. A RateQualifier notes a criterion that was used for setting the actual charge as applied to the round. For example, if the requestor is a wholesaler or reseller, the RateQualifier will be a code associated with that status. If the individual golfer is a member of an organization that typically gets a discount (such as NGCOA, PGA, USGA, etc.) then that code would be included in the rate qualifier. Twilight, weekday, local, hotel guest, etc. will also be reflected here.

2030 **3.6. Course Tee Time Reservation Response Message**  
2031 **(OTA\_CourseResRS)**

2032 The root tag of OTA\_CourseResRS uses the external *include* OTA\_v2ent.xsd that defines the  
2033 root element standard attributes found in all the OTA payload documents, and the response  
2034 options of returning the indication of Success, Warning or Errors in processing the request. The  
2035 response message may include Warnings from business processing rules or Errors if the request  
2036 did not succeed.

2037  
2038 **OTA\_CourseResRS** - The OTA\_CourseResRS message is the response to a reservation  
2039 request at a known single course for one or more tee times. The specific information about the  
2040 golfer or golfers is optional where that data has been supplied in the request. Where the booking  
2041 entity has the authority to take a reservation without a request (from an existing block) and the  
2042 Notification boolean has been set to "Yes", only the attributes of GolfCourseRes are required. All  
2043 elements are optional. It is the main Response element.

2044  
2045 *OTA\_PayloadStdAttributes* - See above

2046  
2047 **Success**

2048  
2049 **Warnings**

2050 **Warning**

- 2051 • *Type* - Type of warning. Valid values: (Unknown | NoImplementation  
2052 | BizRule | Authentication | AuthenticationTimeout | Authorization |  
2053 ProtocolViolation | TransactionModel | AuthenticationModel |  
2054 ReqFieldMissing )
- 2055 • *Code* - The code assigned to the warnin
- 2056 • *DocURL*>
- 2057 • *Status*
- 2058 • *Tag*

2059 **Errors**

2060 **Error**

- 2061 • *Type* - Type of error. Valid values: (Unknown |NoImplementation | BizRule |  
2062 Authentication | AuthenticationTimeout | Authorization | ProtocolViolation |  
2063 TransactionModel | AuthenticationModel | ReqFieldMissing )
- 2064 • *Code* - The code assigned to the warning
- 2065 • *DocURL*
- 2066 • *Status*
- 2067 • *Tag*

2068  
2069 **GolfCourseRes** - A specific reservation for a tee time for an individual or a number of golfers.  
2070 See OTA\_CoursResRQ for description.

2071 **3.7. Sample XML Message for Course Search Request**

2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080

```
<?xml version="1.0" encoding="UTF-8"?>  
<OTA_CourseSearchRQ EchoToken="12345" TimeStamp="1999-05-31T13:20:00-05:00" Target="Production"  
Version="2001C" SequenceNmbr="123456" GolfCourseID="" DetailResponse="Yes">  
  <Criteria Name="Architect" Value="Robert Trent Jones" Required="Yes"/>  
  <Criteria Name="Singles Confirmed" Value="Yes"/>  
  <Criteria Name="ADA Challenged" Value="Wheelchair"/>  
  <Criteria Name="Slope" Value="110" Operation="LessThan" Required="Yes"/>  
</OTA_CourseSearchRQ>
```

2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111

### 3.8. Sample XML Message for Course Search Response

```
<?xml version="1.0" encoding="UTF-8"?>  
<OTA_CourseSearchRS EchoToken="12345" TimeStamp="1999-05-31T13:20:00-05:00" Target="Production"  
Version="2001C" SequenceNmbr="123456" Detail="Yes">  
  <GolfCourse Id="FL1234" Name="Sea Grass Golf Resort" City="Jupiter" State="FL" Country="USA"  
PostalCode="21921" County="Palm Beach" Province="" PhoneNumber="444 444-4444">  
  <Trait Name="Architect" Value="Robert Trent Jones"/>  
  <Trait Name="Singles Confirmed" Value="Yes"/>  
  <Trait Name="ADA Challenged" Value="Wheelchair"/>  
  <Trait Name="Slope" Value="110"/>  
  <Trait Name="Metal Spikes" Value="No"/>  
  <Trait Name="Caddies Available" Value="No"/>  
  <Trait Name="Yardage" Value="6345"/>  
  <Trait Name="Personal Carts Permitted" Value="No"/>  
  <Trait Name="Fivesome" Value="No"/>  
  <Trait Name="GrassType" Value="Bermuda"/>  
</GolfCourse>  
  <GolfCourse Id="FL4321" Name="Beach Side Golf Resort" City="Palm Beach Gardens" State="FL" Country="USA"  
PostalCode="21932" County="Palm Beach" Province="" PhoneNumber="555 555-5555">  
  <Trait Name="Architect" Value="Jack Nicklaus"/>  
  <Trait Name="Singles Confirmed" Value="Yes"/>  
  <Trait Name="ADA Challenged" Value="Blind"/>  
  <Trait Name="Slope" Value="112"/>  
  <Trait Name="Metal Spikes" Value="Yes"/>  
  <Trait Name="Caddies Available" Value="Yes"/>  
  <Trait Name="Yardage" Value="7102"/>  
  <Trait Name="Fivesome" Value="Yes"/>  
  <Trait Name="GrassType" Value="Rye"/>  
</GolfCourse>  
</OTA_CourseSearchRS>
```

2112 **3.9. Sample XML Message for Course Availability Request**

```
2113
2114 <?xml version="1.0" encoding="UTF-8"?>
2115 <OTA_CourseAvailRQ EchoToken="12345" TimeStamp="1999-05-31T13:20:00-05:00" Target="Production"
2116 Version="2001C" SequenceNmbr="123456" GolfCourseID="FL1234">
2117 <GolfCourseTeeTime StartDate="2001-10-31" EndDate="2001-10-31" StartTime="13:00:00" EndTime="14:30:00"
2118 NumberOfGolfers="4" NumberOfHoles="18" NumberOfTimes="1" MaxPrice="80">
2119 <RateQualifier>NGCOA</RateQualifier>
2120 <Memberships/>
2121 </GolfCourseTeeTime>
2122 </OTA_CourseAvailRQ>
```

2123 **3.10. Sample XML Message for Course Availability Response**

```
2124
2125 <?xml version="1.0" encoding="UTF-8"?>
2126 <OTA_CourseAvailRS EchoToken="12345" TimeStamp="1999-05-31T13:20:00-05:00" Target="Production"
2127 Version="2001C" SequenceNmbr="123456" GolfCourseID="FL1234">
2128 <GolfCourseTeeTime Date="2001-10-31" Time="13:36:00" NumberOfGolfers="4" NumberOfHoles="18"
2129 NumberOfTimes="1" NumberOfCarts="2" StartingTee="1" CancellationPolicy="48 Hours" Resellable="No" Capacity="5"
2130 CapacityAvailable="1" Price="81" CurrencyId="USD" ConfID="314253">
2131 <Fee RateQualifier="NGCOA" Name="GreensFee" Amount="70" Currency="USD"/>
2132 <Fee RateQualifier="NGCOA" Name="CartFee" Amount="11" Currency="USD"/>
2133 </GolfCourseTeeTime>
2134 </OTA_CourseAvailRS>
```

2135 **3.11. Sample XML Message for Course Reservation Request**

```
2136
2137 <?xml version="1.0" encoding="UTF-8"?>
2138 <OTA_CourseResRQ EchoToken="12345" TimeStamp="1999-05-31T13:20:00-05:00" Target="Production"
2139 Version="2001C" SequenceNmbr="123456" GolfCourseID="FL1234">
2140 <GolfCourseRes Id="123456" Status="New" StatusMessage="New Reservation Request"
2141 RequestorResId="AC4132">
2142 <Round RoundId="23456" PlayDate="2002-02-10" PlayTime="11:06:00" NumberOfGolfers="2" PackageId=""
2143 StartingTee="1" NumberOfCarts="1" TotalAmount="140.00" CurrencyId="USD">
2144 <Charge Id="Std" Description="Standard Rate - Tourist" Quantity="2" UnitPrice="70" UnitCost="63.34"
2145 Amount="140.00" BillingType="Net"/>
2146 <Golfer FirstName="Bobby" LastName="Jones" Address1="123 Augusta Lane" Address2="" City="Atlanta"
2147 State="GA" PostalCode="23456" Country="USA" PhoneCountry="1" Phone="333-444-5555"
2148 Email="golfer@myswing.com" Gender="M" DateOfBirth="1923-06-03" RateQualifier="NGCOA">
2149 <Membership type="NGCOA" ID="4123546"/>
2150 <CreditCard Number="31452343568762231" ExpirationDate="05/04" CardTypeId="VS"
2151 NameOnCard="Bobby Jones"/>
2152 <Memberships/>
2153 </Golfer>
2154 <RateQualifier>NGCOA</RateQualifier>
2155 </Round>
2156 </GolfCourseRes>
2157 </OTA_CourseResRQ>
2158
```

